AICS '05

AICS '05: Proceedings of the 16th Irish Conference on Artificial Intelligence and Cognitive Science

edited by
Norman Creaney

Published by the University of Ulster,
Cromore Road, Coleraine, BT52 1SA
August 2005.

ISBN: 1-85923-197-7

Norman Creaney (Ed.)
School of Computer and Information Engineering
Faculty of Engineering
University of Ulster at Coleraine
Cromore Road, Coleraine
BT52 1SA

Email: n.creaney@ulster.ac.uk
URL: http://www.ulster.ac.uk/aics05

Preface

This book contains the collection of papers accepted for the 16th Irish Conference on Artificial Intelligence and Cognitive Science (AICS '05). The conference was hosted by the School of Computing and Information Engineering of the University of Ulster at Coleraine, on 7th-9th September 2005; and was held in Flowerfield Arts Centre, Portstewart, Northern Ireland.

The AICS conference has taken place annually since 1988 and provides a forum for the exchange of ideas and the presentation of results relating to research conducted both in Ireland and worldwide. There were oral presentation sessions on: conversational recommender systems, cognitive science, image processing & intelligent multi-media, natural language processing, information retrieval, natural language and intelligent multi-media, coherence and trust, constraint satisfaction, evolutionary computing & neural networks, and machine learning. As in previous years AICS '05 included a poster session, and the papers relating to this session are also included in these proceedings.

September 2005 Norman Creaney

Programme Committee

Michaela Black (University of Ulster)
Derek Bridge (University College Cork)
Ken Browne (University College Cork)
Ruth Byrne (Trinity College Dublin)
Arthur Cater (University College Dublin)
Darryl Charles (University of Ulster)
Rem Collier (University College Dublin)
Brian Crean (Galway-Mayo IT)
Norman Creaney (University of Ulster)
Fred Cummins (University College Dublin)
Padraig Cunningham (Trinity College Dublin)
Kevin Curran (University of Ulster)
John Dunnion (University College Dublin)
Malachy Eaton (University of Limerick)
Colin Fyfe (University of Paisley)
Josephine Griffith (NUI Galway)
Ray Hickey (University of Ulster)
Liadh Kelly (Athlone IT)
Michael Madden (NUI Galway)
Eleni Mangina (University College Dublin)
Paul Mc Kevitt (University of Ulster)
David McSherry (University of Ulster)
Kevin McCarthy (University College Dublin)
Lorraine McGinty (University College Dublin)
Stephen McGlinchey (University of Paisley)
Michael McTear (University of Ulster)
Conor Muldoon (University College Dublin)
Eamonn Newman (University College Dublin)
Greg O'Hare (University College Dublin)
Gearóid. O'Néill (University of Limerick)
Diarmuid O'Donoghue (NUI Maynooth)
Michael O'Grady (University College Dublin)
Ciaran O'Leary (Dublin IT)
Michael O'Neill (University of Limerick)
Ian O'Neill (Queens University Belfast)
Colm O'Riordan (NUI Galway)
Barry O'Sullivan (University College Cork)
Paul Piwek (University of Brighton)
James Reilly (University College Dublin)
Stephen Sheridan (IT Blanchardstown)
Barry Smyth (University College Dublin)
Richard Sutcliffe (University of Limerick)
Marc van Dongen (University College Cork)
Josef van Genabith (Dublin City University)
Carl Vogel (Trinity College Dublin)
Ray Walshe (Dublin City University)

Acknowledgements

The AICS '05 Programme Committee would like to gratefully acknowledge the support that was received from Coleraine Borough Council; and we thank Robert Burke for his striking photograph of the Giant's Causeway. I would also like to express a personal thank you to Lorraine McGinty and Brian Crean, the organisers of AICS '04, for their help and advice; to David McSherry for his assistance in preparing the programme; to Bilal Al Momani for organising the poster session; and to all the reviewers for the thorough and timely feedback they provided to the authors. Finally, I would like to thank Pauleen Marshall for the help and support she provided throughout the process of organising AICS '05.

Additional Reviewers

Arnaud Lallouet (University of Orléans)
Ian Miguel (University of St Andrews)
Adrian Moore (University of Ulster)
Philip Morrow (University of Ulster)
Bryan Scotney (University of Ulster)

Local Organising Committee

Michaela Black
Dave Bustard
Darryl Charles
Norman Creaney (Conference Chair)
Ray Hickey
Pauleen Marshall (Conference Secretary)
Michael McNeill
David McSherry
Bilal Al Momani (Poster Session Coordinator)
Bryan Scotney

Table of Contents

Invited Speakers

Artificial Intelligence or Natural Stupidity: An Exercise in Practical Deception
Noel Sharkey *3*

Autonomic Computing for Pervasive ICT: a Whole-System Perspective
Mark Shackleton *5*

Session 1: Conversational Recommender Systems

Incremental Nearest Neighbour with Default Preferences
D McSherry *9*

Dynamic Critiquing: An Analysis of Cognitive Load
K McCarthy, L McGinty & B Smyth *19*

Diversity-Enhanced Conversational Collaborative Recommendations
D Bridge & JP Kelly *29*

Session 2: Cognitive Science

How People Reason Under Uncertainty: A Computational Model of Probability Judgement
FJ Costello *41*

The Conductor Model of Online Speech Production
F Cummins *51*

Are Mixtures of Experts Psychologically Plausible?
S Helie, G Giguere, D Cousineau & R Proulx *61*

Session 3.1: Image Processing and Intelligent Multi-Media

Song Form Intelligence for Streaming Music across Wireless Bursty Networks
J Doherty, K Curran & P Mc Kevitt *73*

Satellite Image Classification - A Contextual Evidence-based Approach
BM Al Momani, SI McClean & PJ Morrow *83*

Multi-Class and Single-Class Classification Approaches to Vehicle Model Recognition from Images
DT Munroe & MG Madden *93*

Session 3.2: Natural Language Processing

The Role of Experience in the Interpretation of Noun-Noun Combinations
P Maguire & A Cater *105*

Identifying Semantic Equivalence for Multi-Document Summarisation
E Newman, N Stokes, J Dunnion & J Carthy *115*

HybridTrim: A Hybrid Approach to News Headline Generation
R Wang, N Stokes, W Doran, J Dunnion & J Carthy *125*

Session 4.1: Information Retrieval

Evolving Co-occurrence Based Query Expansion Schemes in Information Retrieval using Genetic Programming
R Cummins & C O'Riordan *137*

Probability-Based Fusion of Information Retrieval Result Sets
D Lillis, F Toolan, A Mur, L Peng, R Collier & J Dunnion *147*

Session 4.2: Natural Language and Intelligent Multi-Media

Integrating Visual & Linguistic Salience for Reference Resolution
J Kelleher *159*

Presenting Temporal Relations of Virtual Human Actions by Multiple Animation Channels
M Ma & P Mc Kevitt *169*

Session 5.1: Coherence and Trust

Modeling Trust in Collaborative Web Search
P Briggs & B Smyth *181*

Coherence Measures and their Relation to Inconsistent Knowledge Bases
DH Glass *191*

Session 5.2 Constraint Satisfaction

Robust Constraint Solving using Multiple Heuristics
A Vidotto, KN Brown & JC Beck *203*

CSP Heuristics Categorised with Factor Analysis
RJ Wallace.....213

Session 6: Evolutionary Computing and Neural Networks

Applying Cultural Learning to Sequential Decision Task Problems
D Curran & C O'Riordan 225

Autonomous Dynamics in a Dense Associative Network for Thought Processes
C Gros 235

The Harmonic Topographic Map
M Pena & C Fyfe..... 245

Session 7: Machine Learning

An Assessment of Case Based Reasoning for Short Text Message Classification
M Healy, SJ Delany & A Zamolotskikh 257

Information Extraction from Calls for Papers with Conditional Random Fields and Layout Features
KM Schneider..... 267

Investigation into the use of PCA with Machine Learning for the Identification of Narcotics based on Raman Spectroscopy
T Howley, MG Madden, ML O'Connell & AG Ryder 277

Separating Heuristic Search and Statistical Inference in Decision Tree Learning: The ID3* Algorithm
RJ Hickey 287

Poster Session

Approaches to Developing an Intelligent Multimedia Distributed Platform Hub
GG Campbell, T Lunney & P Mc Kevitt 299

Lexical Functional Grammar Constraints and Concurrent Constraint Programming
P Hancox 309

Enhancing Information Retrieval Interfaces with Information Foraging
C Hoare, H Sorensen 319

Agent Cooperation using Simple Fixed Bias Tags and Multiple Tags
E Howley & C O'Riordan 329

Decision Tree Learning based approach for Traffic Classification in High Speed Networks
P Kulkarni, N Mohammed, S McClean, M Black, G Parr & B Scotney *339*

Emergence of Cooperative Societies in Structured Multi-Agent Systems
M Moran & C O'Riordan..... *349*

A HOTAIR Scalability Model
A Mur, L Peng, R Collier, D Lillis, F Toolan & J Dunnion..... *359*

Reuse and Arbitration in Diverse Societies of Mind
C O'Leary..... *369*

Improving Incremental Critiquing
M Salamo, J Reilly, L McGinty & B Smyth *379*

A Framework for the Automatic Descrtiption of Musical Structure using MPEG-7 Audio
E Smyth, K Curran, P Mc Kevitt & T Lunney *389*

Scheduling with Uncertain Release Dates by Probabilistic Sampling
CW Wu, KN Brown & JC Beck..... *397*

Author Index 409

Invited Speakers

Artificial Intelligence or Natural Stupidity: an Exercise in Practical Deception

Noel Sharkey

Professor of Computer Science, EPSRC Senior Media Fellow,
Department of Computer Science, University of Sheffield,
Regent Court, 211 Portabello Road, Sheffield, S1 4DP, England.
noel@dcs.shef.ac.uk

Since the time of the ancients, machines have been designed to fill the public with awe and wonder. This was achieved by keeping the mechanisms of artificial creatures hidden. As the public grew wiser, cleverer tricks were required. By 200 BC it took self-movement from water powered automata to draw attention and now we have computers and robots. All creatures can be fooled by a bit of camouflage and illusion and humans are no exception. Just as the fisherman uses a lure to catch fish, so the AI worker can pick out features that deceive humans into believing animacy. Robots are already moving into the household as vacuum cleaners and pet-like companions. As this happens more and more, we will need to use more and more sophisticated deception. This talk takes a lively tour through the history of "intelligent" mechanisms from ancient to modern and proposes a new understanding of our role as natural magicians.

Autonomic Computing for Pervasive ICT: a Whole-System Perspective

Mark Shackleton

Future Technologies Group, Orion Building, Floor 1, PP12,
Adastral Park, Martlesham Heath, IP5 3RE, England
mark.shackleton@bt.com

It is unlikely that we can expect to apply traditional centralised management approaches to large-scale pervasive computing scenarios. Approaches that require manual intervention for system management will similarly not be sustainable in the context of future deployments considering their scale and their dynamic (or mobile) nature. This situation motivates the need to apply "autonomic" techniques to system management, where the behaviour of whole systems results from the inherent properties that have been engineered in, i.e. such systems need to be adaptive, reliable and self-managing at the "whole system" level. In this talk we outline a number of design principles that can be applied to create systems that are autonomic in their operation. We focus particularly on generating (and analysing) global system behaviour that arises from the carefully designed interactions of the system components, rather than on the individual behaviour of the components themselves. The design heuristics that we derive (which are often nature-inspired) are illustrated in the context of a number of examples that show how the use of the appropriate principles can lead to the inherent global behaviours that we desire. The result is self-managing, self-repairing systems that can be easily deployed, thus reducing total cost of ownership and increasing overall system reliability.

Conversational Recommender Systems

Incremental Nearest Neighbour with Default Preferences

David McSherry

School of Computing and Information Engineering, University
of Ulster, Coleraine BT52 1SA, Northern Ireland
dmg.mcsherry@ulster.ac.uk

Abstract. Incremental nearest neighbour (iNN) is a case-based reasoning (CBR) approach to product recommendation that aims to minimise the length of recommendation dialogues while ensuring that the dialogue is terminated only when the outcome is certain to be the same regardless of the user's preferences with respect to any remaining attributes. In this paper, we examine the potential role of *default* preferences in the retrieval of recommended cases. As we demonstrate in the domain of digital cameras, default preferences have the potential to dramatically increase recommendation efficiency, particularly when combined with iNN's goal-driven approach to elicitation of personal preferences and ability to recognise when the recommendation dialogue can be terminated without loss of solution quality.

1 Introduction

In CBR recommender systems, cases representing available products are retrieved in response to a query representing the user's requirements. In approaches related to conversational CBR [1], a query is incrementally elicited in a recommendation dialogue with the user, often with the aim of minimising the number of questions the user is asked before an acceptable product is retrieved [2,3]. Recent research has also highlighted the importance of recommender systems having the ability to explain their reasoning [4,5], and recognise when the dialogue can be terminated without loss of solution quality [3].

Incremental nearest neighbour (iNN) is an approach to recommendation designed to address these issues in a mixed-initiative recommender system called *Top Case* [3,5]. It combines a *goal-driven* approach to selecting the most useful question at each stage of the recommendation dialogue with a mechanism for ensuring that the dialogue is terminated only when it is certain that the outcome will be the same regardless of the user's preferences with respect to any remaining attributes. Another important benefit is the ease with which the recommendation process can be explained. For example, Top Case can explain the relevance of any question the user is asked in terms of its ability to discriminate between competing cases. Recommendations based on incomplete queries can also be justified on the grounds that the user's unknown preferences cannot affect the outcome.

In this paper, we examine the potential role of *default* preferences in iNN and other approaches to the retrieval of recommended cases. In a recommender system for personal computers (PCs), for example, it is reasonable to assume that most users

would prefer to maximise processor speed, memory, and hard disk size, while minimising price. It should therefore be necessary to ask the user only about preferences that are likely to differ between individuals, such as make, PC type, and screen size. As well as reducing dialogue length, avoiding unnecessary questions may also be beneficial in terms of solution quality, as providing preferred values for attributes like processor speed may require technical knowledge that the user is lacking.

However, a common approach to the assessment of similarity with respect to numeric attributes is one in which (a) the user is asked to specify a preferred value and (b) values that are closest to the preferred value receive the highest similarity scores. While this makes good sense for attributes like screen size in the PC domain, the idea that a user who specifies £500 as an "ideal" price would prefer to pay £510 than £480 seems unrealistic. Instead, attributes whose values most users would prefer to maximise or minimise are sometimes treated as *more-is-better* (MIB) or *less-is-better* (LIB) attributes in the retrieval process [6-8]. However, there may be no benefit in terms of recommendation efficiency. For example, it is typical in the case of a LIB attribute for a preferred maximum to be elicited, and for the highest similarity score to be assigned to any value that is below the preferred maximum.

In this paper, we present an approach to retrieval based on default preferences in which there is no need to elicit preferred values for LIB or MIB attributes. Instead, the preferred value of a LIB attribute is assumed to be the lowest value of the attribute in the case base, while the preferred value of a MIB attribute is assumed to be the highest value.

In Section 2, we use a well-known case base in the digital camera domain to illustrate the iNN process in Top Case. In Section 3, we present our approach to retrieval based on default preferences and show how it can be combined with iNN's goal-driven approach to the elicitation of personal preferences. In Section 4, we investigate the potential benefits of default preferences in terms of recommendation efficiency. Related work is discussed in Section 5 and our conclusions are presented in Section 6.

2 Recommendation in Top Case

An initial query, if any, entered by the user is incrementally *extended* in Top Case by asking the user about her preferences with respect to attributes not mentioned in her initial query. In the goal-driven approach to attribute selection that characterises iNN, an important role is played by the concept of *case dominance* [3]. In the following definition, *Sim* is the similarity measure on which retrieval is based, and a given query is considered to be an extension of itself.

Definition 1. *A given case* C_1 *dominates another case* C_2 *with respect to a query Q if* $Sim(C_1, Q^*) > Sim(C_2, Q^*)$ *for all possible extensions Q* of Q.*

One reason for the importance of case dominance is that any case that is dominated by another case with respect to the current query can be eliminated; clearly it can never emerge as the most similar case regardless of the user's preferences with respect to any remaining attributes. It can also be seen that if any

case dominates all the other cases, then there is no need for the user's query to be further extended, as the outcome is bound to be the same regardless of the user's preferences with respect to any remaining attributes.

McSherry [3] uses the triangle inequality to show that a given case C_1 dominates another case C_2 with respect to a query Q over a subset A_Q of the case attributes A if and only if:

$$Sim(C_1, Q) - Sim(C_2, Q) > \sum_{a \in A - A_Q} w_a (1 - sim_a(C_1, C_2)) \quad (1)$$

where for each $a \in A$, w_a is the importance weight assigned to a. The cases dominated by a given case can thus be determined with a computational effort that increases only linearly with the size of the case base.

At each stage of the recommendation dialogue, the attribute selected by Top Case is the one with the potential to maximise the number of cases dominated by the case it has selected as a *target* case. Starting with the case that is most similar to the user's initial query, or a randomly selected case if there is no initial query, the target case is continually revised as the user's preferences are elicited.

The recommendation dialogue continues until it is certain that the outcome will be the same regardless of the user's preferences with respect to any remaining attributes. At this stage, the user is informed that the target case has been confirmed as the recommended case. Any cases that equal the current similarity of the target case are also recommended.

The example case base that we use to demonstrate the recommendation process is McCarthy *et al.*'s [9] digital camera case base, which contains the descriptions of over 200 digital cameras. Case attributes and weights assigned to them in our experiments are make (9), price (8), format (7), resolution (6), optical zoom (5), digital zoom (1), weight (4), storage type (2), and memory (3). Fig. 1 shows an example recommendation dialogue in Top Case based on the digital camera case base. Minor editing in the interest of clarity includes the format of the user's initial query. Though available in the case base as part of a retrieved camera's description, the specific model plays no part in the recommendation process and is not shown in the example dialogue because of the limitations of space.

In the example dialogue, the user has declined to enter an initial query. Having assumed the initiative, Top Case randomly selects Case 169, a compact *Casio* camera, as its initial target case, and shows the user this case and one of the 209 cases that are not currently dominated by the target case. Top Case now selects *make* as the most useful attribute for confirming the target case and asks the user what make she prefers. When the user asks why this question is relevant, Top Case explains that if the preferred make is *Casio*, this will eliminate 102 cases, including the *Canon* camera that the user was shown in the previous cycle. That is, if the preferred make is *Casio*, the *Canon* can never equal the similarity of the target case regardless of the user's preferences with respect to the remaining attributes.

Later in the example dialogue, the target case has changed to Case 40 in light of the user's preferences for an ultra-compact Sony costing in the region of 500 euro. It remains unchanged following elicitation of the user's preference with respect to optical zoom, and now dominates all but five of the other cases in the case

Top Case: The target case is:
Case 169: casio, 265, compact, 1.2, 0, 2, 320, compact flash, 8 (0)
Other competing cases (209) include:
Case 210: canon, 459, compact, 5, 3.6, 4.1, 300, compact flash, 32 (0)
What is the preferred make?

User: **why**

Top Case: *Because if make = casio this will eliminate 102 cases, including Case 210.* What is the preferred make?

User: sony

Top Case: The target case is:
Case 33: sony, 875, compact swivel, 8, 7.1, 2, 906, memory stick, 0 (0.2)
Other competing cases (208) include:
Case 43: sony, 304, compact, 5, 3, 4, 259, memory stick, 16 (0.2)
What is the preferred format?

User: ultra compact

Top Case: The target case is:
Case 27: sony, 376, ultra compact, 5, 3, 4, 210, memory stick, 32 (0.36)
Other competing cases (14) include:
Case 41: sony, 184, ultra compact, 2, 0, 0, 118, memory stick, 8 (0.36)
What is the preferred price?

User: 500

Top Case: The target case is:
Case 40: sony, 455, ultra compact, 5, 4, 4, 298, memory stick, 32 (0.53)
Other competing cases (10) include:
Case 30: sony, 454, ultra compact, 5, 3, 4, 200, memory stick, 32 (0.53)

Top Case: What is the preferred optical zoom?

User: 4

Top Case: The target case is:
Case 40: sony, 455, ultra compact, 5, 4, 4, 298, memory stick, 32 (0.64)
Other competing cases (5) include:
Case 35: sony, 249, ultra compact, 2, 0, 0, 200, memory stick, 8 (0.59)

Top Case: What is the preferred resolution?

User: **why**

Top Case: *Because if resolution = 5 this will confirm Case 40 as the recommended case*.
What is the preferred resolution?

User: 5

Top Case: The recommended case is:
Case 40: sony, 455, ultra compact, 5, 4, 4, 298, memory stick, 32 (0.78)

User: **why** 40

Top Case: *Case 40 differs from your query only in price and is highly recommended no matter what digital zoom, weight, storage type, or memory you prefer*

Fig. 1. Example recommendation dialogue in Top Case

base. When asked to explain the relevance of resolution, the most useful of the remaining attributes at this stage, Top Case tells the user that a preferred resolution of 5 mega-pixels would be enough to confirm Case 40 as the recommended case. The question is now repeated and the user tells the system that 5 mega-pixels is in fact the preferred resolution.

As the outcome is now certain to be the same regardless of the user's preferences with respect to the remaining attributes, the dialogue can be safely terminated with Case 40 as the recommended case. When the user asks why Case 40 is recommended, Top Case points out that it differs from her query only in price and is highly recommended no matter what digital zoom, weight, storage type, or memory she prefers.

3 Retrieval Based on Default Preferences

Our approach to retrieval based on default preferences involves the construction of a default query $Q_{default}$ in which the preferred value of a LIB attribute is assumed to be the minimum value of the attribute in the case base. Similarly, the preferred value of a MIB attribute is assumed to be the maximum value in the case base. The default query also includes a preferred value for any nominal attribute with a value that most users can be assumed to prefer. Only attributes for which default preferences can reasonably be assumed are included in the default query.

In the digital camera case base, for example, we assume that most users would prefer to minimise price and weight while maximising memory, resolution, optical zoom, and digital zoom. So our default query in the digital camera domain includes default preferences for price (106 euro), memory (64 Mb), resolution (13.7 mega-pixels), optical zoom (10×), digital zoom (8×), and weight (100 gm). For example, 106 euro is the minimum price in the case base, while 64 Mb is the maximum memory. However, as no obvious assumptions can be made about the preferred make, format, or storage type, these attributes do not appear in the default query.

The question now arises of how to assess the similarity of a given case with respect to any MIB or LIB attributes in the default query. The similarity measure that we use for all numeric attributes is the same regardless of whether or not they are in the default query. As often in practice, we define the similarity of a given case C to a query Q with respect to a numeric attribute a to be:

$$sim_a(C,Q) = 1 - \frac{\left|\pi_a(C) - \pi_a(Q)\right|}{\max(a) - \min(a)} \tag{2}$$

where $\max(a)$ and $\min(a)$ are the maximum and minimum values of a in the case base, $\pi_a(C)$ is the value of a in C, and $\pi_a(Q)$ is the preferred value of a.

In the case of a LIB or MIB attribute, however, there is no need for a preferred value to be elicited from the user as a preferred value is provided in the default query. For example, it follows from Equation 2 that for any case C and LIB attribute a:

$$sim_a(C,Q_{default}) = 1 - \frac{\pi_a(C) - \min(a)}{\max(a) - \min(a)} = \frac{\max(a) - \pi_a(C)}{\max(a) - \min(a)} \tag{3}$$

Our approach to handling default preferences can be used to complement CBR approaches to the retrieval of recommended cases in various ways. In NN retrieval, for example, the default query can be combined with a standard query representing the *personal* preferences of the user. An obvious benefit in the digital camera domain is a 67% reduction in query length relative to full-length queries in which the user is required to specify preferred values for all attributes in the case base. The ability to *kick start* the recommendation process by presenting the case that is most similar to the default query may also be beneficial in critiquing approaches [9-12], particularly when the user declines to enter an initial query. However, we focus in this paper on the potential benefits of retrieval based on default preferences as a starting point for the elicitation of personal preferences in iNN.

Fig. 2 shows an example recommendation dialogue that we use to demonstrate the use of default preferences in Top Case. Our hypothetical user is the same as in Fig. 1. Once again, the user has declined to enter an initial query which would otherwise be combined with the default query as the basis for the retrieval of an initial target case. In the absence of an initial query, the case that is most similar to the default query is retrieved and used as the initial target case in iNN's goal-driven approach to elicitation of the user's personal preferences.

Top Case: The target case is:
Case 76: nikon, 864, SLR-Like, 8, 8, 4, 512, compact flash, 64 (0.46)
Other competing cases (207) include:
Case 1: canon, 879, SLR-Like, 8, 7.1, 3.2, 640, compact flash, 64 (0.44)
What is the preferred make?

User: **why**

Top Case: *Because if make = nikon this will eliminate 193 cases, including Case 1*
What is the preferred make?

User: sony

Top Case: The target case is:
Case 40: sony, 455, ultra compact, 5, 4, 4, 298, memory stick, 32 (0.58)
Other competing cases (33) include:
Case 29: sony, 336, compact, 5, 3, 4, 236, memory stick, 32 (0.57)
What is the preferred format?

User: **why**

Top Case: *Because if format = ultra compact this will confirm Case 40 as the recommended case*

Top Case: What is the preferred format?

User: ultra compact

Top Case: The recommended case is:
Case 40: sony, 455, ultra compact, 5, 4, 4, 298, memory stick, 32 (0.73)

User: **why** 40

Top Case: *Case 40 matches your query exactly and is highly recommended no matter what storage type you prefer*

Fig. 2. Example recommendation dialogue in Top Case based on iNN with default preferences

With default preferences for 6 of the 9 attributes in the case base, the number of questions that the user can be asked is reduced to a maximum of 3. In fact, a recommendation is reached by Top Case after the user has been asked only two questions. The camera recommended by Top Case on this occasion, an ultra-compact *Sony*, is the same as the one it recommended in the example dialogue in Fig. 1. Of course, its recommendation might well have been different, since the default preferences used by Top Case in the second dialogue are quite different from the preferred values elicited from the user in the first dialogue.

4 Empirical Results

Based on a simple *leave-one-out* approach, our first experiment examines the performance of Top Case on the digital camera case base in the absence of default preferences. First we temporarily remove each case, use its description to represent the preferences of a simulated user in a Top Case dialogue based on the remaining cases, and record the number of questions asked before a recommendation is made. As in the example Top Case dialogue in Fig. 1, no use is made of default preferences in this experiment, so the simulated user may be asked up to 9 questions.

Our second experiment, again on the digital camera case base, examines the effects on recommendation efficiency of retrieval based on default preferences combined with iNN's goal-driven approach to the elicitation of personal preferences. Again we use a leave-one-out approach, with the description of a left-out case now providing the *personal* preferences of a simulated user with respect to make, format, and storage type, and with default preferences for the other six attributes. As in the example dialogue in Section 3, each simulated dialogue begins with the retrieval of the case that is most similar to the default query:

price = 106, memory = 64, resolution = 13.7, optical zoom = 10, digital zoom = 8, weight =100

Also as in Section 3, the retrieved case is used as an initial target case in Top Case, and a starting point for the elicitation of personal preferences. Again we record the number of questions asked before a recommendation is made by Top Case. With default preferences for 6 of the 9 attributes, the maximum number of questions that can be asked in this experiment is 3.

Observed dialogue lengths in the two experiments are summarised in Fig. 3. Relative to a recommendation dialogue of the longest possible length, iNN without default preferences has reduced dialogue length by up to 67% and by 37% on average. More striking, though, is the reduction in dialogue length of 73% on average for iNN when combined with default preferences.

As noted in Section 3, our approach to handling default preferences can also be used in the basic NN approach to the retrieval of recommended cases. The impact on recommendation efficiency in NN is easily determined if we assume that the user is required to specify preferred values for any attributes for which preferred values are not provided by default. In Table 1, we summarise the gains in recommendation efficiency on the digital camera case base provided by default preferences in NN and iNN. Reductions in dialogue (or query) length are also shown as percentages of the maximum dialogue (or query) length.

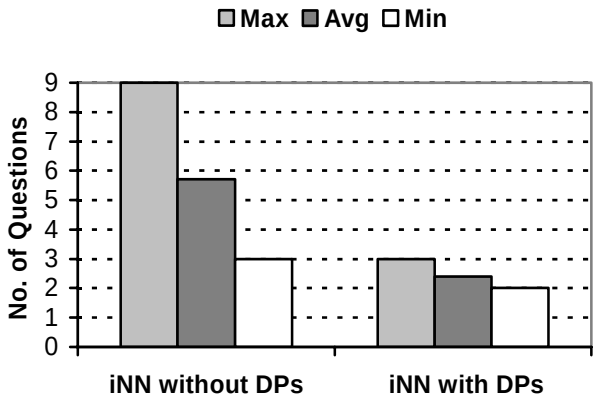


Fig. 3. Lengths of recommendation dialogues in iNN with and without default preferences (DPs) in the digital camera case base

The most striking feature of the results when viewed in this way is the dramatic reduction (67%) in the length of NN queries provided by default preferences. The *additional* reduction of 6% provided by iNN's goal-driven attribute selection strategy, though certainly beneficial, is relatively modest. The combined effect of default preferences and iNN is to reduce dialogue length to a maximum of 2 or 3 questions. As might be expected, iNN's contribution to recommendation efficiency is much greater in the absence of default preferences, again with an average reduction in dialogue (or query) length of 37%.

Table 1. Lengths of NN queries and iNN dialogues with and without default preferences (DPs)

	Without DPs	*With DPs*
NN	9	3 (67%)
iNN	5.7 (37%)	2.4 (73%)

5 Related Work

The importance of default preferences is recognised in Linden *et al.*'s [12] *Automated Travel Assistant*, a flight recommender that interactively builds flight itineraries from real-time airline information. However, retrieval is not based on similarity, and there are no mechanisms for selecting the most useful attribute on each recommendation cycle or recognising when the dialogue can be safely terminated as in iNN. There is also no evaluation of the impact of default preferences on recommendation efficiency.

In existing approaches to retrieval based on MIB and LIB attributes, it is typical for the highest similarity score to be assigned to any value that is above the preferred minimum for a MIB attribute or below the preferred maximum for a LIB attribute [6,7]. However, it does not seem realistic to assume, for example, that all values of a MIB attribute above the preferred minimum are equally preferred [8]. Another problem is that the attribute's discriminating power may be greatly reduced if the user chooses a "modest" value that is exceeded by most cases.

In compromise-driven retrieval (CDR), preferred maximum and minimum values for LIB and MIB attributes are used to identify *compromises* that the user may be prepared to make, but make no contribution to a retrieved case's similarity [8]. For example, the following measure is used to assess the similarity of a given case with respect to a LIB attribute:

$$sim_a(C,Q) = \frac{\max(a) - \pi_a(C)}{\max(a) - \min(a)} \quad (4)$$

Though not explicitly defined in terms of a preferred value, this measure is in fact equivalent to the standard similarity measure for numeric attributes (Equation 2) with the minimum value of the attribute as the preferred value as in Equation 3. Similarly, our assumption that the preferred value of a MIB attribute is the maximum value in the case base reduces the standard similarity measure for numeric attributes to the one used for MIB attributes in CDR.

A known limitation of similarity-based retrieval, and one to which iNN is not immune, is that the most similar case is not necessarily the one that is most acceptable to the user [8,13]. Approaches to extending recommendation dialogues beyond an initially unsuccessful recommendation include *critiquing* approaches to the elicitation of user feedback [9-12], exploring *compromises* that the user may be prepared to make [8], and *referral* of the user's query to other recommender agents in search of more acceptable cases that might be available elsewhere [13]. The potential role of default preferences in these approaches to recovery from an initial recommendation failure is one of the issues we propose to investigate in future research.

6 Conclusions

In combination with iNN's goal-driven approach to the elicitation of personal preferences, the potential benefits of retrieval based on default preferences include a dramatic reduction in the length of recommendation dialogues. For example, dialogue length was reduced to a maximum of two or three questions in our experiments on the digital camera case base. Avoiding questions requiring technical knowledge that users may be lacking (e.g. optical/digital zoom) may also be beneficial in terms of solution quality. Also in the context of iNN, showing the user the most promising case based on default preferences provides a natural starting point for the elicitation of personal preferences.

We have also argued that the potential benefits of retrieval based on default preferences are not limited to iNN. In the basic NN approach to retrieval of recommendation cases, we have shown that default preferences have the potential to

reduce query length by 67% in the digital camera domain. In critiquing algorithms, an initial recommendation based on default preferences may be a useful starting point for the elicitation of user feedback, particularly when the user declines to enter an initial query.

Acknowledgement. Thanks to Kevin McCarthy and his co-authors for providing the digital camera case base [9] used to illustrate the ideas presented in this paper.

References

1. Aha, D.W., Breslow, L.A., Muñoz-Avila, H.: Conversational Case-Based Reasoning. Applied Intelligence **14** (2001) 9-32
2. Kohlmaier, A., Schmitt, S., Bergmann, R.: A Similarity-Based Approach to Attribute Selection in User-Adaptive Sales Dialogues. In: Aha, D.W., Watson, I. (eds.) Case-Based Reasoning Research and Development. LNAI, Vol. 2080. Springer-Verlag, Berlin Heidelberg (2001) 306-320
3. McSherry, D.: Increasing Dialogue Efficiency in Case-Based Reasoning Without Loss of Solution Quality. Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (2003) 121-126
4. Herlocker, J.L., Konstan, J.A., Riedl, J.: Explaining Collaborative Filtering Recommendations. Proceedings of the ACM Conference on Computer Supported Cooperative Work (2000) 241-250
5. McSherry, D.: Explanation in Recommender Systems. In: Gervás, P., Gupta, K. M. (eds.) Proceedings of the ECCBR 2004 Workshops, Universidad Complutense Madrid (2004) 125-134
6. Bergmann, R., Breen, S., Göker, M., Manago, M., Wess, S.: Developing Industrial Case-Based Reasoning Applications: The INRECA Methodology. Springer-Verlag, Berlin Heidelberg New York (1999)
7. Wilke, W., Lenz, M., Wess, S.: Intelligent Sales Support with CBR. In: Lenz, M., Bartsch-Spörl, B., Burkhard, H.-D., Wess, S. (eds.) Case-Based Reasoning Technology. Springer-Verlag, Berlin Heidelberg New York (1998) 91-113
8. McSherry, D.: Similarity and Compromise. In: Ashley, K.D., Bridge, D.G. (eds.) Case-Based Reasoning Research and Development. LNAI, Vol. 2689. Springer-Verlag, Berlin Heidelberg New York (2003) 291-305
9. McCarthy, K., Reilly, J., McGinty, L., Smyth, B.: Experiments in Dynamic Critiquing. Proceedings of the International Conference on Intelligent User Interfaces (2005) 175-182
10. Bridge, D., Ferguson, A.: An Expressive Query Language for Product Recommender Systems. Artificial Intelligence Review, **18** (2002) 269-307
11. Burke, R.: Interactive Critiquing for Catalog Navigation in E-Commerce. Artificial Intelligence Review, **18** (2002) 245-267
12. Linden, G., Hanks, S., Lesh, N.: Interactive Assessment of User Preference Models: The Automated Travel Assistant. Proceedings of the Sixth International Conference on User Modeling (1997) 67-78
13. McSherry, D.: Conversational CBR in Multi-Agent Recommendation. IJCAI-05 Workshop on Multi-Agent Information Retrieval and Recommender Systems (2005)

Dynamic Critiquing: An Analysis of Cognitive Load*

Kevin McCarthy, Lorraine McGinty, and Barry Smyth

Adaptive Information Cluster, Smart Media Institute,
School of Computer Science and Informatics,
University College Dublin (UCD),
Belfield, Dublin 4, Ireland.
{kevin.mccarthy, lorraine.mcginty, barry.smyth}@ucd.ie

Abstract. Conversational recommender systems solicit feedback from users in order to objectively inform the recommendation process. Ideally the user is presented with suitable products/services as promptly, as possible. Efficiency is key, and normally, this is measured in terms of the session length (i.e., the number of recommendation cycles with the user). In this paper we argue that it is also important to understand the effort required of the user during these interactions. Cognitive load refers to the level of effort associated with thinking and reasoning. We will look at the cognitive load implications, as measured by interaction time, of a critiquing conversational recommender which uses dynamically generated *compound critiques*. In particular, we find two interesting results. First, on a cycle-by-cycle basis the dynamic critiquing approach places a greater cognitive cost burden than that for the unit critiquing approach. Secondly, and arguably more importantly, the reverse is true when we look at overall session performance – that is, the dynamic critiquing approach outperforms the unit critiquing variation. We demonstrate these in relation to results obtained in a recent real-user trial.

1 Introduction

Recommender systems are important when it comes to helping users to navigate through complex product spaces [1–3]. In particular they provide useful assistance to users even when user requirements are initially unclear. For example, conversational recommender systems aim to refine a user's initial requirements by presenting a sequence of recommendations and inviting the user to provide feedback on each suggestion. This feedback can be provided in different ways, and is often dependant on the recommendation setting in question. Previous work by [4] describes four distinct forms of feedback - value elicitation, critiquing, preference-based, and ratings-based feedback - and investigates how they rank in terms of reducing recommendation session length. However, this work does point out that recommendation session length is not the only influencing criterion

* This material is based on works supported by Science Foundation Ireland under Grant No. 03/IN.3/I361.

when it comes to assessing recommender performance, and selecting appropriate feedback strategies. For instance, the *cognitive cost* to the user of providing feedback is another, albeit lesser investigated, issue. This is the issue under scrutiny in this paper.

In this paper we are interested in a particular form of feedback, called critiquing [5–7]. Critiquing-based recommender systems expect the user to provide feedback in the form of directional feature-level preferences for a product. For example, in a holiday recommender a user might indicate that they are looking for a holiday that is "*closer to home*", this is a unit critique over the *location* feature of a holiday suggestion. In this way, the standard model of critiquing requires the user to apply a *single* critique to a *single* feature (*unit critiquing*), but recently this model has been extended to cover multiple features, by supporting dynamically generated *compound critiques*.

Section 2 describes the unit and compound critiquing ideas in further detail, and discusses some of the advantages we have previously reported in relation to the notion of dynamically generating compound critiques. A common question we have been faced with in the past has been in relation to the cognitive load impact associated with this idea. For example, how much of an increase in the user's cognitive load is associated with choosing a compound critique over a unit critique? We evaluate this issue in Section 4, such that developers of critique-guided recommender systems can better understand the cost-benefit characteristics associated with using unit and compound critiques. We propose to estimate cognitive load by measuring the amount of time that it takes a user to interact with the recommender system, and we use this to compare the cognitive load characteristics of standard (unit) critiquing and dynamic compound critiquing. In particular, we demonstrate that while the dynamic critiquing approach presents with a higher cognitive load at the level of a single recommendation cycle, its ability to reduce overall average session length has a positive impact on the overall cognitive load at the level of the recommendation session. Importantly, we demonstrate these ideas using results obtained in a recent live-user trial.

2 A Review of Critiquing-Based Recommender Systems

Critiquing, as a form of feedback, is best known by association with the FindMe recommender systems [1, 5]. The original motivation for critiquing included the need for a type of feedback mechanism that was simple for users to understand and apply, and yet informative enough to focus the recommender system. For instance, the Entrée restaurant recommender [1] presents users with a fixed set of directional critiques in each recommendation cycle. In this way users can easily request to see further suggestions that are different from the current recommendation, in terms of some specific feature. For example, the user may request another restaurant that is *cheaper* by critiquing its *price*. Critiques of this type are what we call *unit* critiques. This ultimately limits the ability of the recommender to narrow its focus, because it is guided by only single-feature

preferences from cycle to cycle. An alternative strategy is to consider critiques that operate over multiple features, what we call *compound* critiques. This idea of compound critiques is not novel; the seminal work of Robin Burke [5] refers to critiques for manipulating multiple features. For instance, Car Navigator, a car recommender from the FindMe family, offers both static unit critiques and static compound critiques to the user. An example of one such compound critique is the *sportier* critique, which operates over a number of different car features; *engine size, acceleration* and *price* are all increased. Obviously compound critiques have the potential to improve recommender efficiency because they allow the recommender to focus on multiple feature constraints in each cycle. However, in the past when compound critiques have been used they have been hard-coded by the system designer so that the user is presented with a fixed set of compound critiques in each recommendation cycle. These compound critiques may, or may not, be relevant depending on the cases that remain at a given point in time.

Fig. 1. Screenshot of our prototype dynamic critiquing recommender.

Recently, we have argued the need for a more dynamic approach to critiquing whereby compound critiques are generated, *on-the-fly*, for each recommendation cycle [8, 9]. This *dynamic critiquing* approach generates a set of compound critiques to present to the user in each recommendation cycle. Because these compound critiques are informed by the remaining products, the recommender system will *always* return a recommendation for one of these. In previous work [8, 9] we have described how these compound critiques are generated, selected and presented to the user. The compound critiques generated by the dynamic critiquing approach are always applicable to the current recommendation. We have also developed methods such as critique profiling and introduced diversity in order to enhance the beneficial effects of our dynamically generated compound critiques. Figure 1 shows a screenshot of a digital camera recommender system that we have developed to showcase and evaluate the dynamic critiquing approach. It shows a recommended case, its unit critiques and three relevant compound critiques. From here the user can select a critique to inform the next recommendation, terminating their session when they see a satisfactory camera.

In previous publications we have looked at improving recommender efficiency by reducing the number of conversational cycles. We found in both artificial and on-line evaluations [8–10] that session reductions of up to 69% can be obtained when the user chooses dynamically generated compound critiques. However, the compound critiques we generate change from cycle to cycle, unlike their *unit* counterparts. For example, Entrée uses 7 fixed unit critiques in each cycle and the same critique options appear in the same positions every cycle. Our approach presents unit critiques in a similar way to Entrée, however the compound critiquing part of the interface changes dynamically during each cycle as do the presented compound critiques. How much of an increase in the user's cognitive load is associated with choosing a compound critique over a unit critique, when we consider that the compound critiques are dynamically generated? We will try to answer this question in evaluating the results of a recent live-user trial.

3 Cognitive Load in Conversational Recommenders

The issue of cognitive load in the context of conversational recommender systems is one that has not been explored in great detail to date. The vast majority of recommender systems make cognitive demands on users in some way. Limiting factors include: (1) users rarely have a clear understanding of what product they are looking for, (2) users may not be proficient in the required product description knowledge, (3) users may be unable to effectively compare presented suggestions, and/or (4) are often incapable of retaining learned information in short-term memory.

Assessing cognitive load implications associated with recommendation scenarios of this kind is a very difficult task. Related research has investigated the prospects of using physiological methods, such as pupil size, to measure a user's cognitive load [11]. Despite early optimism, empirical evaluations have demon-

strated that behavioural indicators, such as reading speed, are actually more accurate.

	Air Durham (men) $100 Buy It!	Air Structure Triax (men) $90 Buy It!	Air Span Triax (men) $85 Buy It!
Surface	Road Shoe	Road Shoe	Road Shoe
Midsole	PU midsole	Phylon midsole	Phylon midsole
Width	Standard	Standard	Standard
Motion Control	★★★★	★★★★	★★★★
Feel: Impact Protection	★★★★	★★★★	★★★★
Feel: Responsiveness	★★★★	★★★★	★★★★
Breathability	★★★★	★★★★	★★★★
Water Resistance	No	No	No
Weight	14.5 ounces	12.6 ounces	12 ounces

Fig. 2. Screenshot from the Nike Advisor site, Nike.com, illustrating the cognitive cost burden associated with the case evaluation level.

Accordingly, evaluations of the cognitive load impact of recommender techniques and technologies have tended to examine a user's willingness to interact and speed of interaction as a measure of this cost. In a typical recommendation cycle there are two cognitive task levels that contribute to this cost. They are, (1) cognitive load the user experiences at the *case evaluation level*, and (2) cognitive load they tolerate at the *feature interaction level*. The case evaluation level of cognitive load is best demonstrated when k cases are presented to the user as in the example in Figure 2. There is a comparative cognitive cost because the user has to compare the presented cases to each other before giving feedback.

Our research looks at the cognitive burden placed on users of conversational recommender systems at the feature interaction level. As mentioned earlier, our digital camera recommender uses critiquing as its user feedback mechanism. Only one suggestion is presented in each recommendation cycle so there is no associated cost at the case evaluation level. As shown in Figure 1, the user first examines the relevant unit and compound critiques (i.e., each made up of 3 unit critiques), before providing their feedback. To date our evaluations have concentrated predominantly on the efficiency benefits of our dynamic critiquing approach, in terms of the session length reductions realised. While results have been positive we have been challenged as to whether there are significant cognitive load issues at the feature interaction level. That is, are there cost implications enforced on the user as a result of having to *breakdown* and *understand* the critique options offered. More importantly, these compound critiques change dynamically, and this could lead to longer overall session times. It could be the case that the cognitive cost (i.e., time spent) encountered at the feature interaction level far outweighs the benefit of shorter recommendation session lengths – for example, a unit critiquing user who finds their product in 10 cycles and

takes 2 minutes to do so is better off than a compound critiquing user who finds the same product in 5 cycles but takes 6 minutes to do so.

Obviously if the users in our experiment who use the compound critiques to reduce their system interactions take a considerably longer duration of time to be recommended the product they were looking for, then dynamic critiquing and the use of compound critiques is infeasible in a real system. In our evaluation in Section 4, instead of looking at the number of interactions with the system as a measurement of efficiency we analyse how the time taken for a user to respond with feedback can be used to determine how reductions in interaction cycles correlate with increases in cognitive load.

Some timing analysis has been performed in relation to recommendation efficiency. For example, Smyth and Cotter [12, 13] look at the efficiency of personalized mobile portals. They argued that the efficiency benefit comes not only from the fact that the user has fewer interaction cycles with the system (clicks and scrolls in this case), but also that the user ends up spending less time on unnecessary navigation. The fact that users spend less time navigating to what they want is very important in the mobile domain as users are often charged according to the amount of time they spend connected to the service.

Cognitive load analysis, as measured by interaction time, can only be carried out in the context of live-user evaluations. We have recently carried out such a trial but to date have only reported results that comment on the efficiency benefits of our dynamic critiquing approach (and its extensions) in terms of recommendation cycles. The key contribution of this paper is the cognitive analysis of this data to determine whether or not dynamic compound critiques place an unacceptable burden on the user in terms of their cognitive load.

4 Evaluation

We are especially interested in the response time of users utilizing the compound critiques in our system as a measure of feature interaction level cognitive load. Firstly, we report on the average overall session time and corresponding session lengths for low and high frequency compound critique users. We wish to see if high frequency users suffer from longer session times, therefore telling us if there is a cognitive load trade off by having a shorter number of conversational cycles. Then we look at a break down of the low and high frequency users' time results for both unit and compound critiques. This should give us some insight into how low and high frequency compound critique users differ in how they use the system with respect to cognitive time involved at the feature-interaction level.

4.1 Setup

Users for our trial were made up of both undergraduate and postgraduate students from the School of Computer Science and Informatics at University College Dublin. The participants were asked to use our Digital Camera recommender system (see Figure 1) during December 2004. They were asked to use the system

as if they were purchasing a new digital camera and they were presented with a very brief introductory tutorial to explain the interface and the different forms of feedback. In total 45 different users led to 55 unique recommendation sessions, which were closely monitored and logged. The average overall session length was just over 21 cycles with an average total elapsed time of just over 105 seconds. In addition, users tended to select compound critiques in about 26% of their cycles and unit critiques in the remaining cycles; these application frequencies are similar to those previously reported [8, 9].

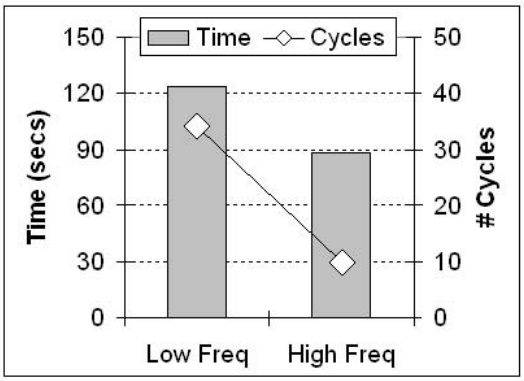


Fig. 3. Cognitive time with efficiency results.

4.2 Results

To assess the relative impact of unit and compound critiques we divided the sessions into two equal groups: *low frequency* sessions included all those sessions whose compound critique application frequency was less than the median frequency, with *high frequency* sessions making up the remainder. On average users with low frequency sessions applied compound critiques only about 7% of the time compared to 43% of the time for high frequency users. Next we computed the average session length and time for these low and high frequency sessions. The results are presented in Figure 3 and clearly indicate an advantage to the high frequency sessions, which are seen to have 71% fewer cycles per session (10 *vs.* 34) and be 28% shorter in time (88.7 seconds *vs.* 123.8 seconds per session) when compared to the low frequency sessions. In other words, when users availed of compound critiques more frequently they benefited from fewer recommendation cycles and shorter recommendation session durations.

By looking at the timing of individual cycles within a session it was possible to gain an understanding of how long it took users to interpret and select between unit and compound critiques – the cognitive load at a feature interaction level.

To do this we compared the average time a user took during a cycle where a unit critique was selected, to the average time taken when a compound critique was selected. In Figure 4 we see that over all sessions, the average time for unit critique cycles is 3.99 seconds, compared to 11.75 seconds for cycles where compound critiques are chosen. This is consistent with the notion that compound critiques are likely to be more time consuming to interpret than the less complex unit critiques; typically, a compound critique is made up of about 3 separate unit critiques and we see here that cycles where compound critiques are selected are almost 3 times as long as unit critique cycles.

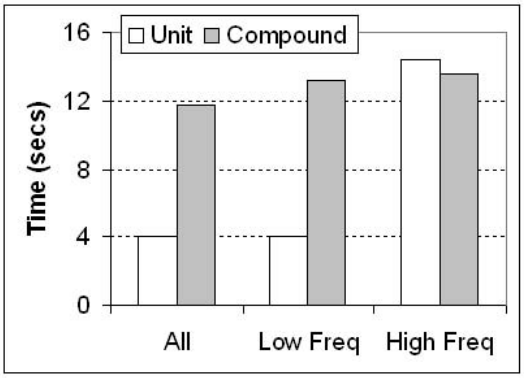


Fig. 4. Unit and compound critiques cognitive time results.

However, this is not the complete picture. In Figure 4 we also show the average time for unit and compound critique cycles within the low frequency and high frequency sessions. The low frequency results are similar to the overall results with unit critique cycles taking about 4 seconds and compound critique cycles taking about 13 seconds. However, the high frequency cycles are very different with unit critique cycles taking over 14 seconds. In fact they are more time consuming than the cycles where compound critiques are chosen. This can be explained by a fundamental difference in the behaviour of low frequency and high frequency users. Remember that the former had an average compound critique application frequency of only 7% (compared to more than 40% for the latter). The application data suggests that the low frequency users were very rarely selecting compound critiques and the timing data indicates that when they were selecting unit critiques they were doing so without due consideration of the compound critique options that were available; this is useful because it suggests that the average unit critique time is indicative of just the time taken to interpret unit critiques. In contrast, the average cycle time for the high frequency users is roughly the same regardless of whether a unit critique or a compound critique is finally chosen. This indicates that these users were at least giving

consideration to both unit and compound critiques during each cycle, and so even when a unit critique was chosen they were still taking the time to reflect on the compound critique options.

5 Conclusions

This paper has described the results of a recent live-user trial to evaluate the relative cognitive load merits of unit and compound critiquing in a conversational recommender system. For the first time we have focused not only on the number of cycles in a recommendation session, but also on the elapsed response time as a measure of cognitive load. Because of this we have been able to gain a better understanding of key differences in the way that users interact with a critiquing-based recommender system. We have clarified that there is a significant increase in the cognitive load associated with the use of compound critiques, when compared to users who focused almost exclusively on unit critiques. However this additional cognitive investment is worthwhile because it translates into sessions that are shorter, both in terms of their elapsed time (28% reduction) and their number of cycles (71% reduction).

It is also interesting to note how users use the system. The low frequency compound critique users very rarely utilize the compound critiques (only about 7% of the time). However, when they did take the time to use them, they responded in almost exactly the same time as the high frequency users for compound critiques. This shows us that the results show a consistent cognitive load across both groups of users for compound critiques. If we also look at the cognitive load times for unit critiques we see there is a huge difference, and as mentioned, this can be attributed to the fact that low frequency users are not even considering the compound critiques on offer.

The results reported in this paper tell us that even if there is a larger cognitive load at the feature interaction level, associated with cycles where compound critiques are considered, the end result is that the cognitive increase is still outweighed by the recommendation cycle reduction possible. Users will find the product they are looking for more quickly and more efficiently with less overall cognitive effort and time consumed if they utilize compound critiques.

As a future work discussion, if we refer again to the screenshot shown in Figure 1, we can see that the unit critiques are placed right beside the feature values. When a user is considering a unit critique they can see the feature value they will be critiquing. It is inherently more difficult for a user to understand a compound critique as the user does not have the benefit of being able to see the feature values beside the unit critiques that make up the compound critique. Also it is a well-known fact in interface design that if toolbar icons, or in our case the unit critiques, do not change position or value from interaction to interaction, that the user will come to subconsciously know that they are available. This learning ability is not available to the compound critiques as they are generated dynamically during each cycle of the recommendation process. It

would be interesting to consider how these factors affect the cognitive load results presented here. However such an evaluation is beyond the scope of this paper.

References

1. Burke, R., Hammond, K., Young, B.: The FindMe Approach to Assisted Browsing. Journal of IEEE Expert **12(4)** (1997) 32–40
2. Schafer, J.B., Konstan, J.A., Riedl, J.: E-Commerce Recommendation Applications. Data Mining and Knowledge Discovery **5** (2001) 115–153
3. Shimazu, H.: ExpertClerk : Navigating Shoppers' Buying Process with the Combination of Asking and Proposing. In Nebel, B., ed.: Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01), Morgan Kaufmann (2001) 1443–1448 Seattle, Washington, USA.
4. Smyth, B., McGinty, L.: An Analysis of Feedback Strategies in Conversational Recommender Systems. In Cunningham, P., ed.: Proceedings of the Fourteenth National Conference on Artificial Intelligence and Cognitive Science (AICS-2003). (2003) Dublin, Ireland.
5. Burke, R., Hammond, K., Young, B.: Knowledge-based Navigation of Complex Information Spaces. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence, AAAI Press/MIT Press (1996) 462–468 Portland, OR.
6. Faltings, B., Pu, P., Torrens, M., Viappiani, P.: Designing Example-Critiquing Interaction. In: Proceedings of the International Conference on Intelligent User Interface(IUI-2004), ACM Press (2004) 22–29 Funchal, Madeira, Portugal.
7. Sherin, S., Lieberman, H.: Intelligent Profiling by Example. In: Proceedings of the International Conference on Intelligent User Interfaces (IUI 2001), ACM Press (2001) 145–152 Santa Fe, NM, USA.
8. McCarthy, K., Reilly, J., McGinty, L., Smyth, B.: On the Dynamic Generation of Compound Critiques in Conversational Recommender Systems. In Bra, P.D., ed.: Proceedings of the Third International Conference on Adaptive Hypermedia and Web-Based Systems (AH-04), Springer (2004) 176–184 Eindhoven, The Netherlands.
9. Reilly, J., McCarthy, K., McGinty, L., Smyth, B.: Dynamic Critiquing. In Calero, P.A.G., Funk, P., eds.: Proceedings of the European Conference on Case-Based Reasoning (ECCBR-04), Springer (2004) 763–777 Madrid, Spain.
10. McCarthy, K., McGinty, L., Smyth, B., Reilly, J.: On the Evaluation of Dynamic Critiquing: A Large-Scale User Study. In Veloso, M., Kambhampati, S., eds.: Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI'05), AAAI Press (2005) Pittsburgh, PA, USA.
11. Schultheis, H., Jameson, A.: Assessing cognitive load in adaptive hypermedia systems: Physiological and behavioral methods. In Bra, P.D., ed.: Proceedings of the Third International Conference on Adaptive Hypermedia and Web-Based Systems (AH 2004), Springer (2004) 225–234 Eindhoven, The Netherlands.
12. Smyth, B., Cotter, P.: The Plight of the Navigator: Solving the Navigation Problem for Wireless Portals. In Bra, P.D., Brusilovsky, P., Conejo, R., eds.: Proceedings of the Second International Conference on Adaptive Hypermedia and Web-Based Systems (AH 2002), Springer (2002) 328–337 Malaga, Spain.
13. Smyth, B., Cotter, P.: Personalized adaptive navigation for mobile portals. In van Harmelen, F., ed.: Proceedings of the 15th Eureopean Conference on Artificial Intelligence (ECAI 2002). (2002) 608–612 Lyon, France.

Diversity-Enhanced Conversational Collaborative Recommendations

Derek Bridge and John Paul Kelly

Department of Computer Science,
University College, Cork
d.bridge@cs.ucc.ie/jpk2@student.cs.ucc.ie

Abstract. In conversational collaborative recommender systems, user feedback influences the recommendations. We summarise the seminal work in this field [5] and make precise a variant in which the likes and dislikes of other users in the system are distinguished when matching against the active user's short-term positive and negative profiles. But the major innovation that we report is our mechanism for enhancing the diversity of the recommendations made by collaborative recommenders. Significantly, we increase diversity using collaborative data only. In our experiments with diversity-enhanced recommendations, users find target items in many fewer recommendation cycles.

1 Introduction

Recommender systems suggest products, services or information sources to their users. They differ in the way they find the items they recommend:

Content-based systems: The system stores a description of each available item. A user describes the item that she wants as a query or she describes the kinds of items that she likes as entries in a user profile. The system compares the user's descriptions against the store of item descriptions and recommends items that match.

Collaborative systems: Item descriptions are not used. A user's profile stores user opinions against item identifiers. The system compares other users with the active user and recommends items that were liked by users whose profiles are similar to the active user's profile.

Recommender systems differ also by the extent to which they engage in dialogue with the user:

Single-shot systems: In response to a user request (and, where appropriate, submission of a user query), the system delivers a set of recommendations to the user. Each request is treated independently of previous ones.

Conversational systems: Users elaborate their requirements over the course of an extended dialogue. On receiving a set of recommendations, the user might refine her query; or she might supply feedback on the recommended items. Her feedback influences the next set of recommendations.

Table 1. A ratings matrix

	Ann	**Bob**	**Col**	**Deb**	**Edd**	**Flo**
Cape Fear	⊥	⊥	3	5	5	5
Naked Gun	3	2	⊥	2	4	⊥
Aliens	⊥	5	⊥	⊥	2	4
Taxi Driver	⊥	⊥	3	4	3	⊥

Conversational systems can more easily adapt their recommendations to the user's short-term interests. By dint of mood changes or other special circumstances, short-term interests may not coincide with long-term interests.

There is a mature body of research on conversational *content-based* systems, but research into *collaborative* systems has focused on single-shot recommenders. The work of Rafter & Smyth [5] is a recent exception. In Sects. 3, 4 and 5, we describe conversational collaborative recommenders of increasing sophistication, starting from the one in [5]. The results of an empirical comparison are reported in Sect. 6. But first, in Sect. 2, we summarise the operation of collaborative recommenders.

2 Collaborative Recommenders

In a collaborative recommender, given m items, $I = \{i : 1 \dots m\}$, and n users, $U = \{u : 1 \dots n\}$, preferences are represented using a $m \times n$ matrix of ratings $r_{i,u}$. Note that it is possible and common that $r_{i,u} = \perp$, signalling that the user has not yet rated that item. An example of a ratings matrix for movies is shown as Table 1. Each column in the matrix is a user's long-term profile. We will write u^{LT} for the item identifiers that have non-$\perp$ ratings in user u's long-term profile. For example, $\text{Bob}^{LT} = \{\text{Naked Gun}, \text{Aliens}\}$.

There are many ways of building collaborative recommenders. Here we describe just the one we have implemented; for details, see [2]:

- The similarity $w_{u_a,u}$ between the active user u_a and each other user, $u \neq u_a$, is computed using Pearson Correlation, $\text{correl}(u_a, u)$, over their co-rated items, devalued in the case of users who have co-rated fewer than 50 items by a significance weight. For later parts of this paper, it is important to appreciate that $\text{correl}(u_a, u)$ and hence $w_{u_a,u}$ can be positive, zero or negative.
- Next, the *nearest neighbours* of u_a are selected, i.e. the N (in our case, 20) other users u for whom $w_{u_a,u}$ is highest.
- For each item i that has not been rated by u_a but has been rated by at least one of the neighbours, u_a's rating for i is predicted, p_{i,u_a}, essentially as a weighted average of the neighbours' ratings for item i.
- These items are then sorted into descending order of p_{i,u_a}. This is the order in which items will be recommended. For example, if in a single-shot system we want to recommend three items, then the first three items in this sorted list are selected.

3 The RS-CCR$^+$ and RS-CCR$^\pm$ Systems

RS-CCR$^+$ and RS-CCR$^\pm$ are our designations of the very basic conversational collaborative recommender systems described in [5]. Note, however, that the system with which Rafter & Smyth do their experiments is closer to the one we describe in the next section (B.Smyth, personal communication 2005). In all these systems, the active user has a long-term profile (based on a column in the ratings matrix), u_a^{LT}, as do all other users. But, for the duration of her interaction with the system, the active user also has two short-term profiles, $u_a^{ST^+}$ and $u_a^{ST^-}$.

Initially, the short-term profiles are empty and the first set of k (typically, three) recommendations is made in the fashion described in Sect. 2. At this point, and indeed whenever the system recommends items to the user, the system solicits user feedback. The user's feedback takes one of two forms:

- She can indicate which recommended item best matches her short-term interests. If she does, the selected item's identifier is added to her short-term positive profile, $u_a^{ST^+}$. Nothing is done with the other items.
- She can indicate that none of the recommended items adequately meets her short-term interests. If she does, the identifiers of all the recommended items are added to her short-term negative profile, $u_a^{ST^-}$.

The system then recommends another set of items. New recommendations never repeat ones made previously in this dialogue. But, additionally, through the way it computes user similarity, the system attempts to steer new recommendations towards the kind of items in $u_a^{ST^+}$ and away from the kind of items in $u_a^{ST^-}$; see below for details. This recommendation-feedback cycle continues until either the user finds an item she wishes to consume, she abandons the dialogue having found no such item, or the system can make no fresh recommendations.

It remains to say how $u_a^{ST^+}$ and $u_a^{ST^-}$ influence subsequent recommendations. When finding neighbours, users whose long-term profiles contain items in the active user's short-term profiles will receive a boost:

- In RS-CCR$^+$, the more a user's long-term profile overlaps with the active user's short-term positive profile, the greater the boost:

$$w_{u_a,u} =_{\text{def}} \text{correl}(u_a, u) \times \text{overlap}(u_a^{ST^+}, u^{LT}) \quad (1)$$

- In RS-CCR$^\pm$, overlaps with the active user's short-term positive and negative profiles are combined by H, the harmonic mean:

$$w_{u_a,u} =_{\text{def}} \text{correl}(u_a, u) \times H(\text{overlap}(u_a^{ST^+}, u^{LT}), \text{overlap}(u_a^{ST^-}, u^{LT})) \quad (2)$$

Given that $u_a^{ST^+}$, $u_a^{ST^-}$ and u^{LT} are simply sets of item identifiers, the overlap function is defined as the size of the intersection of its two arguments (R.Rafter, personal communication 2004). If either intersection is empty, as they quite commonly will be, then in Equations (1) and (2) $\text{correl}(u_a, u)$ will be multiplied

by zero, making $w_{u_a,u}$ also zero. This is undesirable. Hence, Rafter & Smyth do not use the overlap when it is zero (B.Smyth, personal communication 2005) and, with the same effect, in our implementations of RS-CCR$^+$ and RS-CCR$^\pm$ we use the following:

$$\text{overlap}(A, B) =_{\text{def}} \max(1, |A \cap B|) \tag{3}$$

which defaults to 1 when the intersection is empty. (Other definitions are possible without making any major difference to the results, e.g. $1 + |A \cap B|$.)

4 The CCR$^+$ and CCR$^\pm$ Systems

The idea in conversational collaborative recommending is that the selection of nearest neighbours is "... directed towards users that have liked the items in the target user's [short-term positive profile], and towards users that have disliked items in the target user's [short-term negative profile]." [5, p.152] However, in the systems we have described in Sect. 3 the items in the active user's short-term positive profile $u_a^{ST^+}$ are compared with *all* the items in the other user's long-term profile u^{LT}, irrespective of whether the other user liked them or not. Similarly, items in $u_a^{ST^-}$ are compared with the whole of u^{LT}, rather than just those members of u^{LT} that u disliked.

To remove this weakness, the systems that we designate CCR$^+$ and CCR$^\pm$ partition u^{LT} into two: the likes and the dislikes. The likes (the long-term positive profile) we denote by u^{LT^+} and this set is compared with $u_a^{ST^+}$. The dislikes (the long-term negative profile) we denote by u^{LT^-} and this set is compared with $u_a^{ST^-}$. For example, in the MovieLens dataset, whose rating scale is 1–5, u^{LT^+} contains items rated 3 or above; u^{LT^-} contains items rated below 3. Before we give the new definitions of $w_{u_a,u}$, there is another issue to resolve.

Recall that $\text{correl}(u_a, u)$ can be negative. On occasion, there may be so few positively correlated neighbours that negatively correlated users are among those with the highest values for $\text{correl}(u_a, u)$. When this is the case, Equations (1) and (2) will multiply this negative number by the overlap, which is positive. Far from boosting the similarity of a user with high overlap, the resulting value for $w_{u_a,u}$ will be a negative number of greater magnitude and so the user will be less likely to be a neighbour. To obtain proper boosting behaviour, we have chosen to *add* overlap values to $\text{correl}(u_a, u)$. (We tried some better-motivated schemes, but they worked less well.)

Taking both the above ideas into account, we define $w_{u_a,u}$ as follows:

– In CCR$^+$:

$$w_{u_a,u} =_{\text{def}} \text{correl}(u_a, u) + \text{overlap}(u_a^{ST^+}, u^{LT^+}) \tag{4}$$

– In CCR$^\pm$:

$$w_{u_a,u} =_{\text{def}} \text{correl}(u_a, u) + \text{overlap}(u_a^{ST^+}, u^{LT^+}) + \text{overlap}(u_a^{ST^-}, u^{LT^-}) \tag{5}$$

Rafter & Smyth have also addressed the two issues we have discussed in this section (B.Smyth, personal communication 2005). Like us, they partition u^{LT} into likes and dislikes, comparing the former with $u_a^{ST^+}$ and the latter with $u_a^{ST^-}$. Their way of overcoming the problem of negative values of correl(u_a, u) is to exclude such users u from the set of neighbours. We chose our approach because excluding negatively correlated users will (slightly) narrow the set of items that may be recommended and we felt that this was undesirable given that the goal of the system is to make recommendations that do not necessarily reflect the user's normal long-term interests. However, we suspect that this difference of detail results in only marginal differences in recommendations.

In any case, CCR$^+$ and CCR$^\pm$ perform only slightly better than RS-CCR$^+$ and RS-CCR$^\pm$(see Sect. 6). A more radical innovation is needed.

5 The CCR$^+$-Div and CCR$^\pm$-Div Systems

This section introduces the CCR$^+$-Div(b, k) and CCR$^\pm$-Div(b, k) systems. In their names, Div indicates a concern for the diversity of recommendations; b and k are parameters, which are explained below.

For content-based recommender systems, the argument has been convincingly made that items should be selected for *diversity* (relative to each other) as well as *similarity* (to the query or the user's profile) [7]. Too much similarity between the recommended items (e.g. three Woody Allen movies) can be undesirable. But, when recommendations are diverse, if the user is not satisfied with the most highly recommended item, for example, the chances of her being satisfied with one of the alternative recommendations is increased.

There is a body of research that addresses diversity for content-based recommenders, e.g. [7, 1, 4]. It is only now that we are seeing the first work that attempts to improve the diversity of the items recommended by collaborative recommenders. Specifically, apart from our own work, we are aware only of Ziegler's work on book recommendations [8]. Neglect of diversity may be because collaborative recommenders can provide *serendipitous* recommendations [2]. Serendipitous recommendations are pleasing recommendations for unexpected items; on occasion, they may increase diversity. However, we hypothesise that a more direct concern for diversity may be important, especially in *conversational* collaborative systems.

To investigate this, we implemented the Bounded Greedy selection algorithm (henceforth BG) from [7]. To recommend k items, BG finds bk items. In [7], these are the bk items that are most similar to the query (content-based recommending). Here, they are the bk items with the highest prediction values p_{i,u_a} (where neighbours are computed by the CCR$^+$ or CCR$^\pm$ systems). From these bk items, BG selects k to recommend to the user. It selects the k in a greedy fashion, based on ones selected so far; see Algorithm 1.

In the algorithm, the quality of item i relative to the result set so far R is defined as follows:

$$Quality(i, R) =_{\text{def}} (1 - \alpha) \times p_{i,u_a} + \alpha \times RelDiv(i, R) \quad (6)$$

Algorithm 1 The Bounded Greedy selection algorithm. Adapted from [7].

Candidates ← *bk* items recommended by CCR^+ (or $\text{CCR}^\pm$)
$R \leftarrow \{\}$
for $j \leftarrow 1$ to k **do**
  best ← the $i \in$ *Candidates* for which *Quality*(i, R) is highest
  insert *best* into R
  remove *best* from *Candidates*
end for
return R

α is a factor that allows the importance of the predicted rating and diversity to be changed; we use $\alpha = 0.5$. Diversity relative to the result set so far is defined as the average distance between i and the items already inserted into R:

$$\mathit{RelDiv}(i, R) =_{\text{def}} \begin{cases} 1 & \text{if } R = \{\} \\ \frac{\sum_{j \in R} \text{dist}(i,j)}{|R|} & \text{otherwise} \end{cases} \quad (7)$$

This leaves the issue of how to measure distance *between items* in Equation (7). In [7], the distance between items is the inverse of the *content-based* similarity. If item descriptions are available, the same approach can be used to enhance the diversity of collaborative recommendations. Ziegler, for example, uses taxonomic knowledge in his system [8]. But we choose to proceed on the assumption that item descriptions are not available. We enhance diversity using a measure of distance that is calculated using *collaborative data only*.

The intuition behind our approach to distance is that the community of users who have rated item i have a certain set of tastes. The more the membership of the community who rated item i differs from the membership of the community who rated item j, the more likely i and j satisfy different tastes and are different kinds of items.

In detail, then, we compute $\text{dist}(i, j)$ as follows:

- CCR^+(or $\text{CCR}^\pm$) will already have found u_a's N nearest neighbours.
- For both i and j, we create bit vectors I and J of length N. Digit d in vector I is set if neighbour d has a non-$\perp$ rating for item i; similarly for bits in J.
- $\text{dist}(i, j)$ is computed as the Hamming distance between I and J, i.e. a count of the number of positions in which their bits differ.

Fig. 1 illustrates this process; it shows Naked Gun to be more different from Cape Fear than Taxi Driver is. In the figure, we take N, the number of nearest neighbours, to be 3, and we assume these are Ann, Col and Deb. We take their ratings from Table 1 and set bits to show who rated what.

There are other ways of computing distances between items, even using just collaborative data. We mention two alternatives and defend our own choice:

- We could compute the (inverse of) Pearson correlation between *rows* in the ratings matrix, Table 1. Some collaborative recommenders work on this basis, e.g. [6]. An advantage would be that item-item (dis-)similarities could

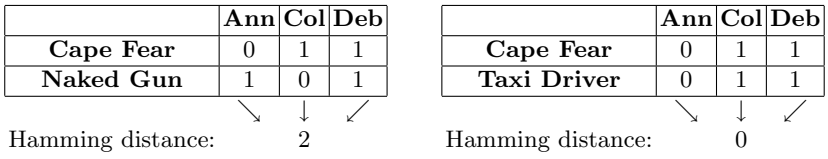


Fig. 1. Hamming distances.

then be computed in advance and cached, needing recalculation only when a new rating arrives. However, our approach restricts attention to the nearest neighbours (which are not known until recommendation time), making item distances sensitive to the active user's long- and short-term preferences.

– Even so, with attention restricted to the nearest neighbours, we could still have used (inverse) Pearson correlation, instead of Hamming distance. The former has the advantage of being sensitive to the actual ratings (the numeric values). However, the BG algorithm requires a very large number of distances to be computed.[1] Hamming distance proves effective (see the next section) while being cheaper to compute.

6 Empirical Evaluation

We adopt Rafter's & Smyth's methodology [5], but our datasets differ. They select the 2100 largest user profiles from the '1 Million MovieLens Dataset'; the average profile size for the 2100 users is 355 ratings. We use the entire '100K MovieLens Dataset', which contains profiles for 943 users; the average profile size is 106 ratings, which we think is more realistic.[2]

One hundred user profiles are selected at random and removed from the dataset. Each of these will act in turn as an (artificial) active user. The item that the user is seeking is obtained through the leave-one-out methodology, i.e. given the active user's long-term profile, each item in turn is withheld and treated as the target item. Sets of 3 recommendations are made to the user until either the target item is one of the recommended items, there have been 100 recommendation cycles, or no further recommendations can be made to this user, whichever comes soonest. If the target item is recommended within 100 cycles, the number of items recommended is recorded. Results are subjected to 3-fold cross-validation, with a different 100 active users in each fold.

In each recommendation cycle, the (artificial) user's feedback needs to be simulated. For each movie, the MovieLens datasets record a set of genres, which allows a simple-minded content-based approach. If the target item's set of genres is G_t and a recommended item's set of genres is G_r, we compute $\frac{|G_t \cap G_r|}{|G_t \cup G_r|}$. If all recommended items score zero, then none is taken to match the user's short-term

[1] $\frac{(2b(k-1)-k-1)k}{2}$ of them, in fact, for each set of k recommendations!

[2] We are grateful to the GroupLens project team for making their data available.

interests, so all the items are inserted into $u_a^{ST^-}$; otherwise, the highest-scoring item (with random tie-breaking) is taken to match the user's short term-interests, so this item is inserted into $u_a^{ST^+}$ and nothing is done with the others.

Fig. 2a shows, as a percentage of 34759 dialogues, how often the target item was found. In addition to RS-CCR$^+$, RS-CCR$^\pm$, CCR$^+$, CCR$^\pm$, CCR$^+$-Div(3, 15) and CCR$^\pm$-Div(3, 15), we show the results for SS-CR, a single-shot recommender (Sect. 2). We regard SS-CR as successful if the target item is among all the possible recommendations it can make to the active user. The other systems are successful if the target item is recommended within 100 cycles of 3 recommendations each. Unsurprisingly, SS-CR has by far the highest success rate; encouragingly, the diversity-enhanced systems, CCR$^+$-Div(3, 15) and CCR$^\pm$-Div(3, 15), have higher success rates than the others.

Fig. 2b shows, for each system's successful trials, how many items are recommended, on average, before the system recommends the target item. The diversity-enhanced systems recommend 20 fewer items than the best of the others. However, all the systems recommend, on average, over 100 items before they reach the target. This would clearly not be acceptable in practice. In defence, we note that the experimental methodology is severe: real users might be satisfied with any one of a set of items, whereas in the experiments there is a single target item each time. Furthermore, the simulated user feedback is so crude that it can sidetrack the conversational recommenders, making them on occasion uncompetitive even with the single-shot system.

Figs. 2c and 2d compare each system with SS-CR (when both are successful). We see (Fig 2c) that the diversity-enhanced systems make fewer recommendations than SS-CR nearly 80% of the time; the other systems are competitive with SS-CR less than 40% of the time. Then in Fig. 2d we show winning and losing margins. The figure shows, for example, that, when RS-CCR$^\pm$ wins against SS-CR, it makes on average 52 fewer recommendations and, when RS-CCR$^\pm$ loses against SS-CR, it makes on average 40 more recommendations. By this measure, RS-CCR$^\pm$ and RS-CCR$^+$ win by most when they win, but they also lose by most when they lose.

Fig. 2e compares the diversity-enhanced systems (with different values for b) with systems that choose k products *at random* from the bk products that have the highest predicted ratings (designated CCR$^+$-Rnd(b, k) and CCR$^\pm$-Rnd(b, k)). This allows us to see that our diversity-enhancement mechanism is making a systematic improvement. We also note that higher values of b have the greatest advantage.

Finally, in Fig. 2f, we compute for each system the average diversity (all-pairs distance) of each set of items it recommends, averaged over all such sets. CCR$^+$-Div(b, k) and CCR$^\pm$-Div(b, k) have the best values but, of course, this has to be taken with a pinch of salt, because it evaluates these two systems with exactly the measure that they seek algorithmically to maximise! All the values may seem low but this is a facet of the averaging; some of the individual recommendation sets may be quite diverse.

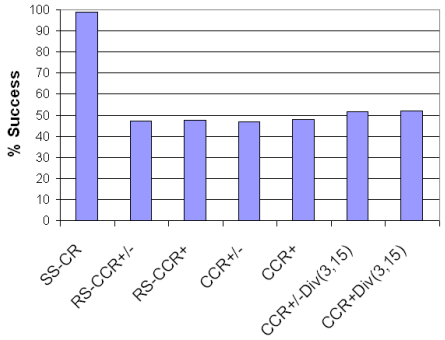


Fig. 2a. Success rates

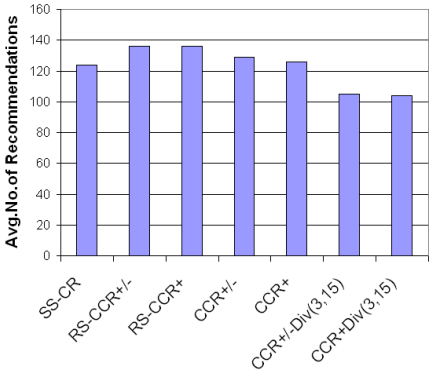


Fig. 2b. Avg. no. of recommendations

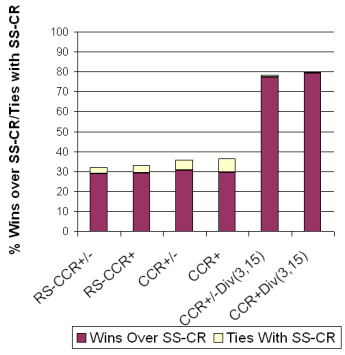


Fig. 2c. % Wins over SS-CR/Ties with SS-CR

	Win	*Lose*
RS-CCR+/-	52	40
RS-CCR+	55	40
CCR+/-	48	29
CCR+	46	24
CCR+/-Div(3,15)	50	31
CCR+Div(3,15)	48	28

Fig. 2d. Winning and losing margins

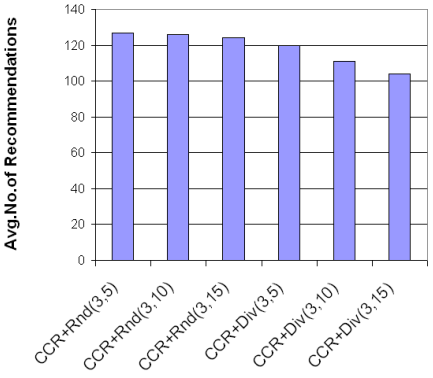


Fig. 2e. Avg. no. of recommendations

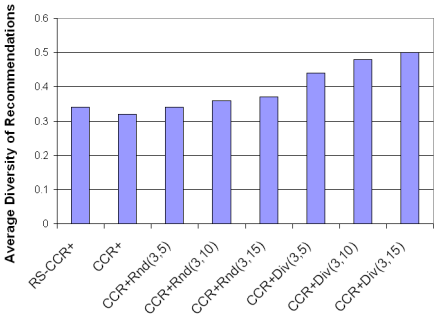


Fig. 2f. Avg. diversity of recommendations

Fig. 2. Empirical results

7 Conclusions

Building on the seminal work reported in [5], we have developed a number of conversational collaborative recommender systems. In all these systems, the selection of neighbours is guided by overlap with the active user's short-term positive and negative profiles. In CCR^{+}-Div(b, k) and $\text{CCR}^{\pm}$-Div(b, k), we introduce for the first time an explicit mechanism that uses collaborative data only to enhance the diversity of recommendations made by (conversational) collaborative recommender systems. Experimental results are hugely improved, and we show that our diversity mechanism makes a systematic difference over a random selection.

Conversational collaborative recommenders are a new line of research, and enhancing the diversity of their recommendations is a new departure too. Future work could include: seeking better-motivated ways of boosting similarity; and more systematic investigation of good values for α, b and k; We would like to see an empirical comparison of different ways of computing item distance (some of which were mentioned in Sect. 5). It would be particularly interesting to compare approaches that use purely collaborative data with those that use content-based data. We would also like to investigate the role of diversity over the course of the dialogue. Diversity can be helpful in early cycles, when the user is exploring the space and making her short-term interests known; but in later cycles, when the user is homing in on a suitable item, diversity may be less appropriate [3].

References

1. D. Bridge and A. Ferguson. Diverse product recommendations using an expressive language for case retrieval. In S. Craw and A. Preece, editors, *Procs. of the 6th European Conference on Case-Based Reasoning*, pages 43–57. Springer, 2002.
2. J. L. Herlocker. *Understanding and Improving Automated Collaborative Filtering Systems*. PhD thesis, University of Minnesota, 2000.
3. L. McGinty and B. Smyth. On the role of diversity in conversational recommender systems. In K. Ashley and D. Bridge, editors, *Procs. of the 5th International Conference on Case-Based Reasoning*, pages 276–290. Springer, 2003.
4. D. McSherry. Diversity-conscious retrieval. In S. Craw and A. Preece, editors, *Procs. of the 6th European Conference on Case-Based Reasoning*, pages 219–233. Springer, 2002.
5. R. Rafter and B. Smyth. Towards conversational collaborative filtering. In L. McGinty and B. Crean, editors, *Procs. of the 15th Artificial Intelligence and Cognitive Science Conference*, pages 147–156, 2004.
6. B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Procs. of the Tenth International World Wide Web Conference*, pages 285–295, 2001.
7. B. Smyth and P. McClave. Similarity vs. diversity. In D. W. Aha and I. Watson, editors, *Procs. of the 4th International Conference on Case-Based Reasoning*, pages 347–361. Springer, 2001.
8. C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *Procs. of the 14th International World Wide Web Conference*, 2005.

Cognitive Science

How People Reason under Uncertainty: A Computational Model of Probability Judgement

Fintan J. Costello (fintan.costello@ucd.ie)

Department of Computer Science, University College Dublin
Belfield, Dublin 6, Ireland

Abstract. This paper describes a computational model of how people make complex probability judgments by combining simpler judgments using the logical operations of conjunction (AND) and disjunction (OR). This model explains the occurrence of a reliable error in people's judgments of conjunction probability (the *conjunction fallacy*, where a conjunction *A AND B* is judged more likely than one of its constituents *A* or *B*) and in their judgement if disjunction probability (the *disjunction fallacy*, where a disjunction *A OR B* is judged *less* likely than one of its constituents *A* or *B*). Two experiments tested this model by asking people to judge the likelihood of different everyday weather events and the likelihood of conjunctions and disjunctions of those events. In both experiments the model was able to accurately predict the occurrence of these fallacies in both conjunctions and disjunctions. The approach to conjunction and disjunction implemented in this model may provide a useful tool for AI models of reasoning under uncertainty, allowing them to combine probabilities in a way that reflects human reasoning.

1 Introduction

The ability to reason under uncertainty (that is, to reason about probable, rather than certain, conclusions) is a central part of human thinking. Such reasoning can be seen in the most commonplace examples of everyday thought ("What's the likelihood of rain today? Should I bring an umbrella?") and in the most challenging and important areas of reasoning ("What's the probability that this patient is suffering from pneumothorax? Should I operate immediately?": pneumothorax occurs when a section of lung tissue gives way allowing air into the chest, and requires immediate insertion of a tube into the ribcage to evacuate air). In this paper I describe a simple computational model that aims to describe how people reason with probabilities. This model is specifically intended to explain how people put probability judgments together in conjunctions (how people AND probabilities) and in disjunctions (how people OR probabilities). This model should be useful for AI reasoning systems intended to mimic human reasoning about conjunctions and disjunctions. This model should also be useful for systems intended to spot possible errors in human reasoning: the model is designed to explain some common errors that people make in their judgements of conjunctive and disjunctive probability.

A fundamental rule of probability is that a conjunction of two events *A AND B* cannot be more probable than either of its constituent events. This rule is an unavoidable consequence of the fact that, for *A AND B* to happen, *A* has to happen and *B* has to happen. This requirement is so straightfoward and so obvious that we would expect people to follow this rule in their judgements of conjunctive probabities. However, in a influential paper looking at how people carried out the operation of conjunction (the AND operation) for judgments of probability, Tversky and Kahnemann [1] found that for some conjunctions people reliably deviate from this rule, judging a conjunction to be more likely than one or other of its constituents (committing a 'conjunction fallacy'). This conjunction fallacy has been confirmed in a number of studies [2, 3].

The standard approach to conjunction assumes that the probability of a conjunction *A AND B* is computed by first obtaining the probability of *A* and the probability of *B*, and then combining those two probabilities using some function. To account for the conjunction fallacy we need to provide a function which will respond differently to different conjunctions, giving some conjunctions a lower probability than their constituents, but giving other conjunctions a higher probability than one or both constituents. A number of researchers, particularly in the area of fuzzy logic, have proposed various conjunctive functions, such as Product, Average, Sum, or Minimum (see [4, 5]). However, none of these can respond in different ways to different conjunctions.

In this paper I propose a simple model of conjunctive probability which can respond in different ways to different conjunctions, in some cases producing correct conjunction responses and in other cases producing conjunction fallacy responses. This model is an extension of standard probability theory. I first describe this model and its account for the conjunction fallacy. I then describe two experiments testing this model by comparing with people's judgements of probability in an everyday domain: that of estimating the likelihood of different types of weather. In these experiments I assess the model's ability to account for the occurence of the conjunction fallacy in people's responses. I also examine the occurrence of *disjunction fallacy* responses in these experiments. A disjunction fallacy arises whenever people judge a disjunction *A OR B* to be *less* probable than any of its constituents *A* or *B*. In both experiments, the model was able to accurately predict both the occurrence of the conjunction fallacy and the disjunction fallacy in people's responses. The model seems to give a simple unified account for both these characteristic errors in people's probabilistic reasoning.

1.1 AND, OR, and NOT in probability theory

Perhaps the simplest way to produce a continuous-valued logic for people's judgments of likelihood is to use standard probability theory equations for ANDing and ORing independent probabilities: equations 1, 2, and 3.

$$P(NOT\ A) = 1 - P(A), \tag{1}$$

$$P(A\ AND\ B) = P(A) \times P(B), \tag{2}$$

$$P(A\ OR\ B) = 1 - P(NOT\ A) \times P(NOT\ B). \tag{3}$$

These equations assume that the probability of a given event A occurring falls in a range from 0 (certain not to happen) up to 1 (certain to happen). The probability of A not occurring is then simply 1 minus the probability of A occurring. The probability of both A and B occurring is simply the product of the probabilities of A and B. Finally, the probability of A or B occurring is 1 minus the probability of not A occurring and not B occurring.

These equations make sense as a logic for graded probability. A problem, however, is that the product function representing AND (Equation 2) is unable to account for conjunction fallacy responses. With constituent probabilities limited to the range from 0 to 1, this product function always produces the same relationship between conjunctive probability and constituent probabilities: a 'less than' relationship where the probability of a conjunction is always less than or equal to the probability of both constituents of that conjunction. In the next section I describe a simple extension that allows the production of various different relationships between conjunctive and constituent probabilities.

1.2 Relationships between constituent and conjunction probabilities

The product function for conjunction will always produce conjunction likelihood ratings that are less than constituent ratings, as long as we assume that likelihood ratings (like probabilities) are limited to the range 0 to 1. Interestingly, however, if we allow ratings scores to move above 1, the product function will produce three different relations between constituent likelihood values and conjunctive values. If the values for both constituents A AND B are below one, the conjunctive value ($A \times B$) is lower than both. If one value is above 1 and the other is below 1, however, the conjunctive value $A \times B$ will fall between both values (if $A = 0.9$ and $B = 1.1$, $A \times B = 0.99$, higher than A but lower than B). Finally, if both constituent scores are above 1, the conjunctive score will be higher than both values (if $A = 1.1$ and $B = 1.2$, $A \times B = 1.32$, higher than both).

This observation leads to a new proposal as to why people sometimes produce the conjunction fallacy in judgments of likelihood: they are making use of the product function for forming conjunctions of constituent judgments, but some of those constituent judgments are higher than 1. In any conjunction where one or both constituent have likelihood judgments greater than 1, the conjunction fallacy will occur. The following equations formalise this idea in an 'offset' version of the standard probabilistic logic. This 'offset' model makes a distinction between people's probability judgments and their responses, and assumes that their responses are offset so that some fall above 1 and others below 1.

$$R(A) = P(A) + s, \tag{4}$$

$$R(NOT\ A) = P(NOT\ A) + s, \tag{5}$$

$$R(A\ AND\ B) = R(A) \times R(B), \tag{6}$$

$$R(A\ OR\ B) = 1 - R(NOT\ A) \times R(NOT\ B). \quad (7)$$

In these equations, the likelihood judgment for item A is equal to the probability of that item plus a offset factor s. This offset factor moves some likelihood judgments over the 1 boundary. Similarly, the likelihood judgment for $NOTA$ is the probability of $NOTA$, offset by the factor s. Importantly for the current proposal, the likelihood judgment for a conjunction $A\ AND\ B$ is simply equal to the product of the offset likelihood judgments for A and B. In some cases one of these constituent judgments will will be above 1, producing conjunction fallacy responses. Finally, the likelihood judgment for a disjunction will be as in the standard probability approach, but making use of the offset likelihood judgments for $NOTA$ and $NOTB$, rather than the standard probability judgments.

In the next section I describe an experiment which tests this 'offset' model in two ways. This experiment first examines the extent to which the equation for AND (Equation 6) above can account for people's pattern of response for conjunctive judgments, and for the occurrence of conjunction fallacy responses. Next, the experiment examines the extent to which the derived equation for OR (Equation 7) can account for people's pattern of response disjunctive likelihood judgments and for the occurence of any *disjunction* fallacy responses.

2 Experiment 1

This experiment asked participants to rate the likelihood of various different single weather events ('cold', 'sunny','windy') and different conjunctions and disjunctions ('cold and windy', 'cold or windy') of those events. Weather events were used because people frequently need to assess the likelihood of such events in their everyday lives (deciding whether they need a coat when going out, whether they should bring sunglasses, and so on), and so the assessment of likelihood in the experiment should be a natural task for participants to carry out.

The aim in this experiment was to examine the relationships that can hold between people's judgments of likelihood for complex weather events (conjunctions and disjunctions) and their judgments of likelihood for the constituents of those complex events. Three different relationships were examined: the complex event being more likely than both constituent events (a 'max' response); the complex event being more likely than one constituent event, but less likely than the other (a 'mid' response); and the complex event being judged less likely than both constituents (a 'min' response). The experiment compares the frequency of these different types of response for conjunctions and disjunctions with the occurrence of these responses as predicted by the 'offset' model.

2.1 Materials and procedure

Twenty-four familiar types of weather event were selected and formed into 12 pairs. For each pair, a conjunctive event was generated by ANDing the two constituents and a disjunctive event was generated by ORing the constituents (see Table 1 below for a full list). These 24 single events and 24 complex events (12

conjunctions and 12 disjunctions) were printed two per page in a questionnaire given to participants. Events were in a different random order in each questionnaire. Each event was accompanied by a a request to the participant to 'rate the following statement on the corresponding scale according to the likelihood with which you believe the weather type may occur many times in Ireland over the year'. This request was followed by a 7-point rating scale going from 'highly unlikely' (-3) to 'highly likely'(+3). The cover page of this questionnaire contained two worked examples using different weather events to demonstrate the task. A total of 16 participants were given these questionnaires, which took between 20 and 40 minutes to complete.[1]

2.2 Results

To examine the occurrence of 'max', 'mid' and 'min' responses for conjunctions in the experiment, each participant's conjunction and constituent likelihood judgments were individually compared. In total there were 192 distinct conjunction likelihood judgments in the experiment (16 participants $\times$ 12 conjunctions). The most frequent pattern across these judgments was a logically correct 'conjunction less than constituents' ('conjunction min') response, which occured in 113 cases (59% of the total). The two other fallacious response types were rarer: a 'conjunction mid' response was given in 38 cases (20%) , and a 'conjunction max' response in 34 cases (18%), giving a total of 72 cases (38%) of fallacious responses overall. In the remaining 3% of cases the conjunction likelihood was equal to both constituent likelihoods: all three likelihood rating were the same.

A similar count of 'max', 'mid' and 'min' responses for *disjunctions* in the experiment was also carried out. The logically correct 'disjunction max' response occured in 75 cases (39%). The fallacious 'disjunction mid' and 'disjunction min' responses occured in 72 cases (38%) and 35 cases (18%) respectively, giving a total of 107 (56%) cases in which people made fallacious judgments. In the remaining 5% of cases all three likelihood rating were the same.

2.3 Comparison with the offset model

The 7-point likelihood rating scale used by participants in the experiment went from 'highly unlikely' (-3) to 'highly likely'($+3$). To apply the offset model to this data, the average likelihood ratings produced for participants for single weather events were first transformed so they fell onto the 0 to 1 range for probabilities. This transformation was carried out by simply increasing each rating value by 3 and then dividing by 6. Under this transformation -3 (the lowest point on the rating scale, marked 'highly unlikely', became $((-3)+3)/6 = 0$, and $+3$ (the highest point on the rating scale, marked 'highly likely', became $((+3) + 3)/6 = 1$. After this transformation a value for the offset parameter s was selected and added to all transformed single-event likelihoods, and these offset single-event likelihood scores were used as input for the equations for

[1] I would like to thank John Phelan for running this experiment.

Table 1. Response patterns given by at least 50% of participants to conjunctions and disjunctions in Expt 1, with response patterns computed by offset model.

	response patterns for conjunction			response patterns for disjunction	
conjunction	observed	computed	disjunction	observed	computed
cold and windy	min (correct)	min (correct)	cold or windy	max (correct)	max (correct)
frost and drizzle	min (correct)	min (correct)	frost or drizzle		
fair and hot	min (correct)	min (correct)	fair or hot	mid (fallacy)	mid (fallacy)
rain and thunder	mid (fallacy)	mid (fallacy)	rain or thunder	mid (fallacy)	mid (fallacy)
rough winds and hail			rough winds or hail	max (correct)	max (correct)
warm and humid	min (correct)	min (correct)	warm or humid	max (correct)	mid (fallacy)
wet and bright	min (correct)	min (correct)	wet or bright		
overcast and calm			overcast or calm	mid (fallacy)	mid (fallacy)
showers and sunny	min (correct)	min (correct)	showers or sunny	mid (fallacy)	mid (fallacy)
sleet and sunshine	min (correct)	min (correct)	sleet or sunshine		
gale winds and hazy	min (correct)	min (correct)	gale winds or hazy		
icy and cloudy	min (correct)	min (correct)	icy or cloudy		

AND (Equation 6) and OR (Equation 7). Using this procedure, the offset model computed values for each of the 12 event conjunctions which participants rated in the experiment, and each of the 12 event disjunctions. A conservative offset parameter value of $s = 0.1$ was used; this value was considered conservative because adding this value to all 24 single-event likelihoods resulted the likelihood of only one event moving above 1.

To compare the model's predicted responses with those seen in the experiment, I counted the number of participants who gave the response patterns 'min', 'mid', or 'max' for each conjunction and disjunction in the experiment. If a given response pattern was produced by 50% or more of participants for any given conjunction or disjunction, that was taken to be the dominant response pattern for that item. These dominant response patterns from the experimental data were then compared with the predicted response patterns computed by the offset model (see Table 1). Note that the blank entries in the response columns in Table 1 represent cases where there was no dominant response for that conjunction or disjunction. The model's responses are not compared with participants' responses in these cases.

As Table 1 shows, there was a close association between the response patterns predicted by the model and the dominant response patterns in the experiment. For example, there were 10 conjunctions for which the response 'min' was dominant; the model also produced that response for those conjunctions. There was one conjunction ('thunder and rain') for which the response 'mid' was dominant (a conjunction fallacy response, in that the conjunction likelihood is higher than one of the constituent likelihoods). The model also produced that response for that conjunction. In total there were 17 cases where there was a dominant

response for a conjunction or disjuction; in 16 out of that 17 the model produced the same response ($p < .01, binomial$). The model thus seemed to be able to mirror people's production of both logically correct responses ('min' for conjunctions; 'max' for disjunctions) and for both conjunction and disjunction fallacies ('mid' responses in both cases).

To further investigate the aggeement between the model's computed conjunction and disjunction scores and participants' responses in the experiment, the models scores were compared with the average likelihood scores for conjunction and disjunctions in the experiment. For disjunctions there was a significant correlation between observed average likelihoods and those computed from constituent scores by the offset model ($r = 0.72, \%var = 0.52, p < .01$). For conjunctions, however, the correlation was less significant ($r = 0.61, \%var = 0.37, p < .05$). Given the model's good account of the occurence of the different sorts of conjunction and disjunction response patterns in the experiment, these relatively low correlations are surprising. One possible explanation for these relatively low correlations comes from the fact that participant's likelihood judgments for single events and for conjunctions and disjunctions in the experiment fall into quite a narrow range; most single and complex events were judged likely by most participants; few events were judged unlikely. To address this possibility the next experiment repeated the task of the current experiment, but using a set of single events that were distributed more evenly across the range of different likelihoods.

3 Experiment 2

As before, this experiment asked participants to rate the likelihood of various different single weather events and the likelihood of conjunctions and disjunctions of those events. In this experiment, single events were selected to have a range of different likelihoods, from highly unlikely to highly likely.

3.1 Materials and procedure

From the set of 24 single events used in Experiment 1, two sets of 4 single events were selected so that each set of events included events of a range of different likelihoods. By combining each single event from one set with every single event in the other set, a collection of 16 event pairs were constructed. For each pair, a conjunctive event was generated by ANDing the two constituents and a disjunctive event was generated by ORing the constituents. There were thus 8 single events, 16 conjunctive complex events, and 16 disjunctive complex events, used as materials in the experiment. Single and complex events were presented to experimental participants on a standard web browser. Each participants saw events in a different random order. The format and instructions used were as in Experiment 1. 21 participants were given these web-based questionnaires, which typically took between 20 and 40 minutes to complete.

Table 2. Response patterns given by at least 50% of participants (or, for fallacious responses, by the highest proportion of participants) to conjunctions and disjunctions in Expt 2, with response patterns computed by offset model.[2]

	response patterns for conjunction			Response patterns for disjunction	
conjunction	observed	computed	disjunction	observed	computed
rain and cloudy	equal (correct)	all equal (correct)	rain or cloudy	equal (correct)	all equal (correct)
windy and rain	equal (correct)	all equal (correct)	windy or rain	equal (correct)	all equal (correct)
rain and sunny	min (correct)	mid (fallacy)	rain or sunny		
thunder and rain[a]	mid (fallacy)	mid (fallacy)	thunder or rain		
cold and cloudy	equal (correct)	all equal (correct)	cold or cloudy	equal (correct)	all equal (correct)
windy and cold	equal (correct)	all equal (correct)	windy or cold	equal (correct)	all equal (correct)
cold and sunny	min (correct)	min (correct)	cold or sunny		
thunder and cold	min (correct)	min (correct)	thunder or cold[b]	mid (fallacy)	mid (fallacy)
frost and cloudy	min (correct)	mid (fallacy)	frost or cloudy		
windy and frost	min (correct)	mid (fallacy)	windy or frost		
frost and sunny	min (correct)	min (correct)	frost or sunny	max (correct)	max (correct)
thunder and frost	min (correct)	min (correct)	thunder or frost	max (correct)	max (correct)
sleet and cloudy			sleet or cloudy		
windy and sleet[a]	mid (fallacy)	mid (fallacy)	windy or sleet[b]	mid (fallacy)	mid (fallacy)
sleet and sunny	min (correct)	min (correct)	sleet or sunny	max (correct)	max (correct)
thunder and sleet	min (correct)	min (correct)	thunder or sleet	max (correct)	max (correct)

[a] Fallacious 'conjunction mid' response produced by 43% of participants.

[b] Fallacious 'disjunction mid' response producedby 38% or participants.

3.2 Results

As in Experiment 1, each participant's judgments of likelihood for conjunctions and disjunctions were examined individually. In total there were 336 distinct conjunction likelihood judgments in the experiment (21 participants × 16 conjunctions). The most frequent pattern across these conjunctive judgments was a logically correct ' min' response, occuring in 173 cases (51%). The two fallacious response types were rarer: in 64 cases (19%) a 'mid' response was given, and in 36 cases (11%) a 'max' response occured, giving a total of 100 (30%) cases in which people made fallacious judgments. In 63 cases (19%) the conjunction likelihood was exactly equal to both constituent likelihoods: all three likelihood rating were the same. Apart from these 'all equal' responses, the pattern here was similar to that seen in Experiment 1 (where logically correct 'conjunction min' responses also dominated).

[2] No fallacious response was produced by at least 50% of participants in Experiment 2. To allow comparison with the offset model, the most frequent fallacious responses for conjunctions and disjunctions are included in this table, even though they were produced by less than 50% of participants.

Disjunction responses were also similar to those from Experiment 1 (apart from an increase in 'all equal' responses). The logically correct 'max' response occured in 120 cases (36%). The fallacious 'mid' and 'min' responses occured in 66 cases (20%) and 41 cases (12%) respectively, giving a total of 107 (32%) fallacious judgments. Finally, in 109 cases (32%) an 'all equal' response was returned.

3.3 Comparison with the offset model

As before, to apply the model the average likelihood ratings produced for participants for single weather events were transformed onto the range 0 to 1 by increasing each rating value by 3 and then dividing by 6. A value for the offset parameter s was selected (a conservative value of 0.06 was used) and added to all these single-event likelihoods, and these offset single-event likelihood scores were used as input for the equations for AND and OR (Equations 6 and 7).

In this experiment, unlike in Experiment 1, there was a significant proportion of responses where both constituent event likelihoods and conjunctive or disjunctive event likelihood were all equal. Since the offset model was computing conjunctive and disjunctive likelihoods on the basis of average likelihood scores for constituent events, the model was extremely unlikely to produce a response in which both offset constituent scores and computed conjunction (or disjunction) likelihood were all exactly equal. To attempt to account for the 'all equal' responses in the experiment, a second parameter was added to the model, a distance parameter d such that if any likelihoods computed by the model fell with d of each other, they would be taken to be equal. This distance parameter would give the model a chance to account for the 'all equal' responses seen in the experiment. A value of $d = 0.05$ was chosen for this parameter.

To compare the model's predicted responses with those seen in the Experiment, I counted the number of participants who gave the response patterns 'min', 'mid', or 'max' for each conjunction and disjunction in the experiment. If a given response pattern was produced by 50% or more of participants for any given conjunction or disjunction, that was taken to be the dominant response for that item. There were no cases where a 'mid' response was produced by 50% or more of participants for either a conjunction or a disjunction. Without such responses there would be no cases of the conjunction fallacy or the disjunction fallacy against which the model's predictions could be compared. To avoid this problem, the conjunctions for which the 'mid' response was most frequent, and the disjunctions for which the 'mid' response was most frequent, were also included in the set of dominant responses (see Table 2). Again, the blank entries in the response columns in Table 2 represent cases where there was no dominant response for that conjunction or disjunction. The model's and participants' responses are not compared in these cases.

There was a close association between the response patterns predicted by the model and the dominant response patterns in the experiment. In total there were 25 cases where there was a dominant response for a conjunction or disjunction; the model produced the same response for 22 out of that 25 ($p < .01, \mathit{binomial}$).

The model's computed scores were compared with the average likelihood scores for conjunction and disjunctions in the experiment, giving a significant correlation both for conjunctions ($r = 0.85, \%var = 0.72, p < .01$) and disjunctions ($r = 0.92, \%var = 0.85, p < .01$). In both cases these correlations are higher than those seen in Experiment 1, and confirm the model's account for the experimental data.

4 Conclusions

Both the conjunction fallacy and disjunction fallacy have been used as evidence that people do not think logically; and indeed, both conjunction fallacy and disjunction fallacy responses must be seen as logically incorrect. From some perspectives, this is very worrying: if even our mental mechanisms for conjunction and disjunction (two of the simplest possible operations) are not logically consistent and correct, how can we rely on our more complex thought processes? The offset model of continuous-valued logic described here may address these concerns. In this model, our mechanisms for both conjunction and disjunction are logically justifiable and consistent (the equations for AND and for OR are simply transferred directly from probability theory). Errors in reasoning, such as conjunction and disjunction fallacies, do not arise because our mental operations for conjunction or disjunction are illogical. Rather they arise because of 'offset' of the inputs to these operations.

This model should be useful for AI reasoning systems in two ways. First, by making use of the 'offset' model for AND, OR and NOT, systems that are designed to mimic human reasoning in some way (natural-language-processing systems, for example, or systems which model human-computer interaction) may be able to give a closer match to people's responses and judgments. Second, systems that are designed to advise human users in some way (medical diagnosis systems, for example, or automated tutoring systems) may, by using this model, be able to spot situations in which users are likely to be making logical errors and to point out these errors. Given the simplicity of the 'offset' idea, it should be simple to apply this approach to AI systems in these areas. However, more work is needed to assess the contribution the 'offset' model would make in these domains.

References

1. Tversky, A. and Kahneman, D.: Extension versus intuititve reasoning: The conjunction fallacy in probability judgment. Psychological Review **90** (1983) 293-315.
2. Stolarz-Fantino, S., Fantino E., Zizzo, D.J.: The conjunction effect: New evidence for robustness. American Journal of Psychology **116(1)** (2003) 15-34.
3. Sides, A., Osherson, D., Bonini, N.: On the reality of the conjunction fallacy. Memory & Cognition **30(2)** (2002) 191-198.
4. Osherson, D. N., Smith E. E. : Gradedness and conceptual combination. Cognition **12(3)** (1982) 299-318.
5. Hajek, P.: Metamathematics of Fuzzy Logic. (1988) Kluwer, Amsterdam.

The Conductor Model of Online Speech Modulation

Fred Cummins

Department of Computer Science,
University College Dublin
fred.cummins@ucd.ie

Abstract. Two observations about prosodic modulation are made. Firstly, many prosodic parameters co-vary when speaking style is altered, and a similar set of variables are affected in particular dysarthrias. Second, there is a need to span the gap between the phenomenologically simple space of intentional speech control and the much higher dimensional space of manifest effects. A novel model of speech production, the Conductor, is proposed which posits a functional subsystem in speech production responsible for the sequencing and modulation of relatively invariant elements. The ways in which the Conductor can modulate these elements are limited, as its domain of variation is hypothesized to be a relatively low-dimensional space. Some known functions of the cortico-striatal circuits are reviewed and are found to be likely candidates for implementation of the Conductor, suggesting that the model may be well grounded in plausible neurophysiology. Other speech production models which consider the role of the basal ganglia are considered, leading to some observations about the inextricable linkage of linguistic and motor elements in speech as actually produced.

1 The Co-Modulation of Some Prosodic Variables

It is a remarkable fact that speakers appear to be able to change many aspects of their speech collectively, and others not at all. I focus here on those prosodic variables which are collectively affected by intentional changes to speaking style, and argue that they are governed by a single modulatory process, the 'Conductor'. In the following section, the Conductor is independently motivated by considering the nature of intentional control of complex action. Converging evidence for the Conductor comes from consideration of the role of cortico-striatal circuits, known to be critical to the online control of action sequences. The Conductor is intended to be a small step towards a biologically plausible account of speech production [23].

In several studies in which subjects mimic other speakers, both Zetterholm [36, 37] and Wretling and Eriksson [35, 14] found that mimics were able to produce reasonable matches to target of such global prosodic parameters as pitch range, global tempo, and phrase size, while many of the fine details of their speech, evident at segmental or subsegmental level, were relatively unchanged,

or changed in ways which were not systematically related to the target. These global prosodic variables may not be independent of one another.

In my own work, I have found that a similar group of variables are affected when two people read aloud at the same time. Speakers in Synchronous Speech experiments have little or no difficulty in reading a prepared text aloud and in synchrony with a co-speaker [9]. In this paradigm, subjects are allowed to read a short text through, and are then given a start signal by the experimenter. Typically, subjects find the task of reading in tight synchrony with another speaker to be a natural one, and their performance is good from the outset, and does not improve much with practice [10]. To satisfy the task goals, they modify their speech rate, inhibit their natural expressive intonation, and produce a rather 'vanilla' form of speech which is, presumably, maximally predictable for their co-speakers. Comparison of speech produced when reading alone and together with another person reveals that there are no clear differences in the relative duration of speech elements across the two conditions [11]. The conditions differ, however, in that temporal variability across speakers is greatly reduced in the synchronous condition for macroscopic intervals, such as phrases and pauses, but unaffected for smaller ones, such as syllables and segments. Pitch variation is also reduced in synchronous speech, and of course the task requirements demand that speakers match their phrase on- and off-sets rather exactly.

A similar bag of variables are communally affected in several motor speech disorders, notably those involving damage to the basal ganglia, as in Parkinson's Disease (PD). The hypokinetic dysarthria typical of this syndrome is characterized (among other things) by difficulty in the initiation of speech, a greatly reduced intonational contour, and altered rhythm, often manifested as rapid but inappropriately modulated syllable sequences [8, 19]. The speech problems experienced by sufferers of PD are clearly related to general motor difficulties, which likewise present as difficulty in initiating action, and disturbed fluency or rhythm once action gets underway.

In a recent thesis, Tyrone [32] argued that dysarthria is a feature of sign language as well as spoken language. Deaf subjects were found to exhibit sign dysarthria in the absence of severe impairment of simple, non-sequenced movements. She concluded that the similarities in vocal and signed dysarthria were rooted in their related demands on the sequencing of complex coordinated movements, rather than in language *per se*. This interpretation receives support from the nature of the difficulties PD patients exhibit in other non-linguistic motor tasks.

One might summarize the variables which are collectively affected by intentional stylistic variation (mimicry, synchronous speech) and by unintentional pathology (hypokinetic dysarthria) as those related to the fluent modulation of speech. Rhythmic modulation, phrase initiation and ending, and intonational variation together make up a set of prosodic variables one might group together under the term of convenience of 'phrasing'. It has been notoriously difficult to cleanly separate linguistic and paralinguistic elements to prosody. As we shall see below, there appears to be considerable overlap in the brain circuits supporting

the modulation of speech in response to a specific speaking situation and those responsible for syntactic sequencing, and so a clean separation of prosody into linguistic and non-linguistic components may not be possible in principle.

2 The Route from Simple Intent to Multiple Effects

If asked to modify one's speech, e.g. by speaking rapidly, or in a very different style, the subjective impression is one of making a relatively simple change. While subjective impressions are not especially trustworthy indicators of mental activity, it is nonetheless striking that a radical change to speech style (yelling, calming voice, comedic variation) is achieved without much conscious detail—one simply shifts from one's regular voice to an altered form—yet the measurable effects are many and varied. In particular, the bag of variables previously grouped under the label 'phrasing' are all going to be affected, yet one does not have the impression of independent variation of each of a host of parameters. (Of course, any given stylistic modification may affect other variables as well, but the phrasing variables identified here are typically affected together.) Rather, these variables collectively characterize specific speaking styles. There is therefore an explanatory gap to be bridged between the subjective experience of a relatively low-dimensional space of intentional speech modification and the observed higher dimensional space of manifest effects.

We have observed that the prosodic variables which collectively constitute the hallmarks of many speaking styles are not independent, but are modulated together. This suggests that a full account of speech production ought to capture their mutual dependence. In what follows, I sketch a preliminary model that does just that. The model draws heavily on an analogy for its initial form, but it will be demonstrated that there is a wealth of neurological evidence, and several relevant and related models, which together suggest that the model is a substantial first step towards a neurobiologically plausible model of online speech production.

3 The Conductor Model of Online Modulation of Speech

The starting point for the present model is an analogy with the conductor of an orchestra. The conductor does not play any instrument herself. In her absence, it may be even possible for an orchestra to get through a musical composition, but the performance will lack the coherence and emotional import of a well-conducted one. The conductor is partly responsible for the sequencing of the individual phrases, but her role most critically affects the temporal and expressive modulation of the individual parts which contribute to the musical whole. Critically, the conductor does not interfere in the high dimensional space of instrument control. Her signals to the individual players are relatively abstract, being restricted to a few dimensions of temporal sequence, relative intensity and their dynamics. (Musical) phrase initiation, cessation and pausing, continuous tempo variation, accentual prominence, are all controlled by the conductor in an

abstract fashion, unencumbered by the differences involved between fingering an oboe and bowing a viola.

One can likewise posit a neurological system which does not, itself, contain detailed instructions for making individual gestures or gesture constellations required for speaking, but which is responsible for sequencing such constellations, and ensuring that they are appropriately modulated, as required by the speaking style employed. The observations made above suggest that this process would affect macroscopic durations, intensity modulation and intonational variation (range, and perhaps accent height). I will refer to this hypothetical process as the 'Conductor'.

In this view of speech production, elements are retrieved from some source, and are sequenced and modulated during online production. The retrieved elements themselves contain the gesture-specific information required for production. The conductor is responsible for the temporal sequencing of these gestures, including the responsibility for ensuring that such sequencing is fluent and context sensitive. The conductor is also responsible for the affective modulation of the units sequenced, that is, intensity and pitch modulation which is not specified by the concatenated units, but is a function of the specific communicative situation, including speaker, and listener-oriented constraints. This modulation is relativly abstract, and may be thought of as akin in some respects to continuous variation along the hypo-hyper axis of variation, as in Lindblom's H&H theory of speech production [25].

The model is agnostic about the exact nature of the elements sequenced, but we note that they can hardly be much larger than syllables, or much smaller than segments. The collection of gestures which are phased with respect to the syllable nucleus in Articulatory Phonology provides a plausible candidate unit size [5, 6] which may serve for initial development of the model. (As an aside, it is interesting to ask what size the units sequenced by a conductor are, or, indeed, whether the question is meaningful.)

4 Implementing the Conductor Within a Production Model

The framework of Articulatory Phonology (AP), and its implementation using Task Dynamics, provides an initial insight into how the Conductor might operate during production. Some recent work within AP has sought to incorporate abstract gestures, which are, themselves, not tied to specific articulators. A 'prosodic-gesture' or 'π-gesture' is employed to modulate the temporal unfolding of a group of physical gestures which are linked to specific articulators [7]. The AP model allows at least two distinct modulation options here: the stiffness of individual gestures, or the clock-rate which underlies the dynamics of all gestures. Although current opinion seems to favour the latter as a modulation mechanism (see also [28]) it is probably too early to be dogmatic about the exact method of modulation employed. Modulation of these parameters alone brings with it changes to the relative alignment (and hence the fluent context-conditioned se-

quencing) of elements, and also has consequences for the extent of the resultant gestures, as demonstrated in Byrd and Saltzman (2003). This model thus provides a natural framework for the future development of the Conductor model.

Articulatory Phonology is not the only framework in which a process akin to the Conductor could be implemented. It is also the case that the adoption of the AP framework requires a commitment to several choices which are not necessary elements of the Conductor model. For example, the proposed π-gestures in AP are constrained to affect all concurrent gestures similarly. This is perhaps a sensible requirement, but it is not a necessary consequence of the Conductor model. In addition, the simple second-order dynamic associated with individual gestures within AP constrains the number of possible variables which could be affected by the Conductor, effectively limiting them to stiffness modulation and time warping. Other approaches to gestural modelling might provide a different set of potential implementation variables, and each such set will impose different limits on the effects which a relatively abstract and non-specific Conductor can bring about. But it is a strong contention of the present model that any plausible account of speech production must allow this kind of abstract, gesture-independent modification by an external process.

One clear responsibility of the Conductor is the regulation of speaking tempo. It has been repeatedly observed that the bulk of tempo variation in speech production is effected by adjusting the duration and relative frequency of pauses [31]. That is, it is the initiation and cessation of individual phrases which underlies most of the perceived tempo of speech, not any direct modification of the internal details of segments or syllables. This is clearly compatible with a Conductor process whose primary task is the fluent sequencing of relatively invariant units in production. Further tempo modulation, corresponding to changes in articulation rate, can be achieved by varying the above stiffness and clock variables.

5 A Neurophysiological Basis for the Conductor

I propose that there is, in fact, a neurophysiological system which implements the Conductor process. The proposed role of the Conductor seems entirely compatible with our current knowledge of the role of specific circuits originating in motor and pre-motor cortex, extending by a variety of routes through the basal ganglia, onwards through the thalamus and back to cortex. There are several parallel circuits known to exist, and they include both direct and indirect paths through the basal ganglia [12, 17, 3].

Proposed functions of these cortico-striatal loops, as they are sometimes called, include the selection of some actions and inhibition of others, the rule-based sequencing of actions, and the coordination of action sequences into fluent wholes [3]. The role of these circuits in speech production has been the focus of some empirical and modelling work [23, 33], as discussed below.

Connectivity between the individual stages of the cortico-striatal loops suggests a funnelling of information, or dimensionality reduction between the cerebral cortex and the basal ganglia. A recent model, the Reinforcement Driven

Dimensionality Reduction model of Bar-Gad, Bergman and colleagues has made the postulate explicit that the basal ganglia are compressing cortical information using optimal information extraction methods [2, 3]. This dimensionality reduction which appears to take place suggests that if the basal ganglia are modulating the sequencing and execution of individual components, they are doing so in a lower dimensional space than that which specifies the execution of each individual component. In short, the funneling taking place at the basal ganglia appears to be *prima faciae* suited to implementing a low-dimensional control signal which modulates the individual motoric components in relatively non-specific fashion, as envisioned by the Conductor model.

The issue of low-dimensional control over a complex, high-dimensional system addresses both issues raised at the outset. It accords well with the intuition that control is relatively abstract and goal-directed, and does not involve detailed and disjoint control over the myriad of variables affected by a change in style. This is a solution which addresses the infamous 'degrees of freedom' problem noted by Bernstein [4], and is similar to action-theoretic approaches to skilled action, in which task-specific goals are defined in a relatively low-dimensional space, and they cause multiple, mutually yoked effects in effector space [20, 21]. It also follows that low-dimensional control of a higher-dimensional system will have, as a necessary consequence, the co-variation of very many variables in the more complex system.

A caveat is in order, before the hypothesized Conductor is identified with specific neural circuits. Although the cortico-striatal loops are clearly implicated in rule-based sequencing, context-conditioned action modulation and fluency, all of which suggest a pivotal role in speech production, there are several such circuits, which may differ greatly in their relative contributions, and the parallel circuits may not be entirely separate. The circuits are also implicated in other, rather distinct activities, such as reward-based action. Matsumoto and colleagues [27] have shown that CS-loops may be essential to the *acquisition* of smooth movement patterns, but that they may not be essential to their *execution*, though this evidence is based on two primates only. And similar circuits linking cortex, thalamus and the cerebellum are also regularly implicated in the fluid control of action. Indeed the cerebellar loops may jointly regulate fluent action sequencing in tandem with the cortico-striatal loops [30]. The insula has also been implicated in the coordination of speech articulation [13].

6 Relation to Other Models of Speech Production

The view of speech production sketched herein suggests some answers to rather fundamental issues in modelling speech production. It is assumed that relatively context-insensitive representations are available for sequencing and for context-specific modulation by the Conductor. This rather weak claim accords with most views of the process of speech production, and is thus relatively uncontroversial. However, the Conductor models assumes that those forms which are available for sequencing are already specified in a form suitable for online modulation

by the Conductor. I have suggested that the syllable representations employed within Articulatory Phonology suggest themselves as possible units. One reason this is so is that a gestural specification of linguistic forms provides some rather obvious channels for modulation to be effected, via time warping or stiffness change. More conventional phonological representations which employ timeless, abstract symbols pose huge problems of translation into some form suitable for production [16].

The proposed Conductor model is not at all incompatible with some existing models of speech production. Firstly, the abstract process of modulation through time warping or stiffness variation suggested here is one possible way of implementing Lindblom's continuum of Hyper- and Hypo-speech [25]. The H&H model emphasizes the fact that speech production is adaptive and finely modulated, so as to respect both speaker and listener-oriented constraints. The modulation envisaged within this model is responsible for a myriad of kinematic effects, but these are understood to derive from a much simpler, low-dimensional control space.

Lieberman [23, 24] has developed a theory of the evolution of speech, in which speech production is based around what he calls "Functional Language Systems", implemented by distributed networks within the brain. The cortico-striatal loops which are implicated by the Conductor model are here hypothesized to underlie sequencing of both speech/motor elements and syntax.

Ullman [33, 34] has developed a model in which declarative and procedural elements are fundamentally separated. The declarative elements correspond roughly to lexical units, while the procedural systems are responsible for their sequencing. He explicitly identifies the cortico-striatal circuits, along with the cortico-cerebellar circuits previously mentioned, as supporting the procedural system. The sequencing of elements treated in Ullman's model refers to syntax, rather than the 'phonetic' sequencing discussed above. Indeed, there is good reason to think that the sequencing abilities of the cortico-striatal circuits might serve both purposes: the physical stringing together of units into a fluid sequence of sounds, and the rule-based serial ordering of units retrieved from the lexicon, and ordered in accordance with the rules of a grammar. Some general notes on sequencing now follow.

7 On Sequencing

The basal ganglia and associated circuits are phylogenetically old, going back at least to the common ancestor of the human and the frog. In rats, cortico-striatal loops (CS-loops) are critically implicated in grooming behaviour where such grooming consists of syntactically well-formed sequences of highly practiced actions [1]. Neuronal activity in the CS-loops is not a function of the individual movements (which may occur within our outside of syntactically governed sequences), but of the syntactic sequence itself. Rat grooming syntax is not hierarchically complex, but involves sequencing of specific action types. Graybiel [18] has argued that one role of the basal ganglia in sequence learning includes

the recoding of action elements into higher-order units: a form of 'chunking' for action. Fentress [15] has demonstrated that the acquisition of the adult grooming pattern in mice is more than just learning to string appropriate movements together. Baby mice learn individual grooming strokes in isolation, and then have to learn to integrate them within a fluent action sequence. Initial attempts to generate a fluent sequence appear to result in a temporary 'unlearning' of the individual parts, as the sequencing itself is mastered. This again points to a clean separation of the problem of fluent sequencing from that of the execution of individual actions in isolation. It also suggests that part of what is being mastered is the hierarchical organization of action sequences, and not just a linear ordering. The hierarchical nature of sequential action in humans has also been demonstrated by Rosenbaum [29].

It may appear as if two entirely separate roles for the CS-loops are being sugggested. On the one hand, they are clearly implicated in syntactic sequencing. This is the domain of formal linguistics, and is typically considered to be entirely disjoint from the messier business of producing sounds. On the other, the same circuits are suggested to be responsible for the fluent production of context-conditioned speech.

Perhaps the separation of disciplines typically enshrined in our academic departments and professional societies may not adequately reflect the partition of labour as embodied in real brains [23, 24]. If language is not to be considered as miraculous, it must indeed be based on cognitive abilities which precede it phylogenetically. The close association of syntactic sequencing and the fluent sequencing of complex skilled action was famously pointed out by Karl Lashley [22]. The convergence of behavioural and neurophysiological evidence sketched above seems to suggest that we may be within sight of an account of language which is credible diachronically in evolutionary terms, and synchronically in neurophysiological terms. Despite the difficulties this may pose for trade unions in Universities, it is surely to be welcomed.

References

1. J. Wayne Aldridge, Kent C. Berridge, Mark Herman, and Lee Zimmer. Neuronal coding of serial order: syntax of grooming in the neostriatum. *Psychological Science*, 4(6):391–395, 1993.
2. I. Bar-Gad, G. Havazelet-Heimer, J. A. Goldberg, E. Rupping, and H. Bergman. Reinforcement-driven dimensionality reduction—a model for information processing in the basal ganglia. *J. Basic and Clin. Physiol. Pharm.*, 11(4):305–320, 2000.
3. I. Bar-Gad, G. Morris, and H. Bergman. Information processing, dimensionality reduction and reinforcement learning in the basal ganglia. *Progress in Neurobiology*, 71:439–473, 2003.
4. N. Bernstein. *The Coordination and Regulation of Movements*. Pergamon Press, London, 1967.
5. Catherine Browman and Louis Goldstein. Some notes on syllable structure in articulatory phonology. In Osamu Fujimura, editor, *Articulatory Organization: Phonology in Speech Perception*. S. Karger, Basel, 1988.

6. Catherine P. Browman and Louis Goldstein. Articulatory phonology: An overview. *Phonetica*, 49:155–180, 1992.
7. Dani Byrd and Elliot Saltzman. The elastic phrase: modeling the dynamics of boundary-adjacent lengthening. *Journal of Phonetics*, 31(2):149–180, 2003.
8. G. J. Canter. Speech characteristics of patients with parkinson's disease: I. intensity, pitch, and duration. *Journal of Speech and Hearing Disorders*, 28(3):221–229, 1963.
9. Fred Cummins. On synchronous speech. *Acoustic Research Letters Online*, 3(1):7–11, 2002.
10. Fred Cummins. Practice and performance in speech produced synchronously. *Journal of Phonetics*, 31(2):139–148, 2003.
11. Fred Cummins. Synchronization among speakers reduces macroscopic temporal variability. In *Proceedings of the 26th Annual Meeting of the Cognitive Science Society*, pages 304–309, 2004.
12. M. R. DeLong. Overview of basal ganglia function. In Mano et al. [26], pages 65–70.
13. Nina F. Dronkers. A new brain region for coordinationg speech articulation. *Nature*, 384:159–161, 1996.
14. Anders Eriksson and Pär Wretling. How flexible is the human voice?–a case study of mimicry. In *Proceedings of EUROSPEECH*, volume 2, pages 1043–1046, Rhodes, Greece, 1997.
15. John C. Fentress. Hierarchical motor control. In *Psychobiology of Language*, pages 40–61. MIT Press, 1983.
16. Carol A. Fowler, Philip Rubin, Robert Remez, and Michael Turvey. Implications for speech production of a general theory of action. In B. Butterworth, editor, *Language Production*, pages 373–420. Academic Press, San Diego, CA, 1981.
17. Ann M. Graybiel. The basal ganglia. *Trends in Neuroscience*, 18(2):60–62, 1995.
18. Ann M. Graybiel. The basal ganglia and chunking of action repertoires. *Neurobiology of Learning and Memory*, 70:119–136, 1998.
19. L. Hammen, Vicki and Kathryn M. Yorkston. Speech and pause characteristics following speech rate reduction in hypokinetic dysarthria. *Journal of Communication Disorders*, 29:429–445, 1996.
20. Katherine S. Harris. Action theory as a description of the speech process. In Herman F. M. Peters and Wouter Hulstijn, editors, *Speech Motor Dynamics in Stuttering*, chapter 2, pages 25–39. Springer, New York, 1987.
21. J. A. S. Kelso, K. G. Holt, P. N. Kugler, and M.T. Turvey. On the concept of coordinative structures as dissipative structures: II. Empirical lines of convergence. In G.E. Stelmach and J. Requin, editors, *Tutorials in Motor Behavior*. North-Holland, 1980.
22. Karl S. Lashley. The problem of serial order in behavior. In L. A. Jefress, editor, *Cerebral Mechanisms in Behavior*, pages 112–136. John Wiley and Sons, New York, NY, 1951.
23. Philip Lieberman. *Human Language and Our Reptilian Brain: The Subcortical Bases of Speech, Syntax, and Thought*. Harvard University Press, 2000.
24. Philip Lieberman. On the nature and evolution of the neural bases of human language. *Yearbook of Physical Anthropology*, 45:36–62, 2002.
25. Björn Lindblom. Explaining phonetic variation: a sketch of the H&H theory. In William J. Hardcastle and A. Marchal, editors, *Speech Production and Speech Modelling*, pages 403–439. Kluwer Academic, Dordrecht, 1990.
26. N. Mano, I. Hamada, and M. R. DeLong, editors. *Role of the Cerebellum and Basal Ganglia in Voluntary Movement*. Elsevier, 1993.

27. N. Matsumoto, T. Hanakawa, S. Maki, A. M. Graybiel, and M. Kimura. Nigrostriatal dopamine system in learning to perform sequential motor tasks in a predictive manner. *Journal of Neurophysiology*, 82:978–998, 1999.
28. Robert Port and Fred Cummins. The English voicing contrast as velocity perturbation. In J. Ohala, T. Nearey, B. Derwing, M. Hodge, and G. Wiebe, editors, *Proceedings of the 1992 International Conference on Spoken Language Processing*, pages 1311–1314. University of Alberta, 1992.
29. David A. Rosenbaum, Sandra B. Kenny, and Marcia A. Derr. Hierarchical control of rapid movement sequences. *Journal of Experimental Psychology: Human Perception and Performance*, 9(1):86–102, 1983.
30. W. T. Thatch, J. W. Mink, H. P. Goodkin, and J. G. Keating. Combining versus gating motor programs: differential roles for cerebellum and basal ganglia. In Mano et al. [26], pages 235–245.
31. Jürgen Trouvain. *Tempo Variation in Speech Production*. PhD thesis, Institut für Phonetik, Universität des Saarlandes, 2004. Published as Forschungsbericht Nr. 8.
32. Martha Ellen Tyrone. *An Investigation of Sign Dysarthria*. PhD thesis, The City University, London, 2004.
33. M. T. Ullman, S. Corkin, M. Coppola, G. Hickok, J. H. Growdon, and W. J. Koroshetz. A neural dissociation within language: evidence that the mental dictionary is part of declarative memory, and that grammatical rules are procesed by teh procedural system. *J. Cognitive Neuroscience*, 9(2):266–276, 1997.
34. Michael T. Ullman. Contributions of memory circuits to language: the declarative/procedural model. *Cognition*, 92:231–270, 2004.
35. Pär Wretling and Anders Eriksson. Is articulatory timing speaker specific? – evidence from imitated voices. In *Proc. FONETIK 98*, pages 48–52, 1998.
36. Elizabeth Zetterholm. Intonation pattern and duration differences in imitated speech. In *Proc. Speech Prosody 2002*, pages 731–734, Aix-en-Provence, 2002.
37. Elizabeth Zetterholm. The same but different: three impersonators imitate the same target voices. In *Proc of 15th ICPHS*, Barcelona, 2003.

Are Mixtures-of-Experts Psychologically Plausible?

Sébastien Hélie[1], Gyslain Giguère[1], Denis Cousineau[2], and Robert Proulx[3]

[1] Université du Québec À Montréal, Computer Science, C.P. 8888 Succ. Centre-Ville, Montréal, H3C 3P8, Canada
{Helie.Sebastien, Giguere.Gyslain}@courrier.uqam.ca
[2] Université de Montréal, Psychology, C.P. 6128 Succ. Centre-Ville, Montréal, H3C 3J7, Canada
Denis.Cousineau@umontreal.ca
[3] Université du Québec À Montréal, Psychology, C.P. 8888 Succ. Centre-Ville, Montréal, H3C 3P8, Canada
Proulx.Robert@uqam.ca

In this paper, we explore the psychological plausibility of mixture-of-experts models. This type of models is referred to in psychology as the knowledge partitioning theory (KP). Over the years, it was argued that: 1) KP is a necessary part of function learning, 2) the usefulness of KP is proportional to task difficulty, and 3) the experts used by humans to perform function learning tasks are always linear. In the present study, these statements were tested by modifying the test display. The results show that increasing the difficulty of stimulus estimation unexpectedly resulted in non-linear KP. Also, adding less useful information to the display resulted in a smaller proportion of partitioning participants. We conclude that mixture-of-experts are adequate psychological models for KP, but that the linearity and ubiquity claims need to be weakened.

1 Introduction

The main goal of any intelligent agent is to adapt to its environment. This is often accomplished by finding contextual cues which are informative about the action to be performed next. One way of achieving such adaptation is to use several processes (or *experts*), each associated to a particular context. If enough experts are available to adequately cover the entire space of action, no search is necessary: a network can be trained to gate each situation to the correct expert, which computes the best action according to context. This type of architecture is called a mixture-of-experts [1, 2]. This class of models is particularly effective in situations where different (even contradictory) responses are appropriate according to situations [2] (such as multispeaker vowel recognition [1]).

One task accomplished by humans in order to adapt to their ever changing environment is categorization. In the particular case where exemplars (inputs) and categories (outputs) are continuous, one usually extracts the function relating the input to the output (function learning), instead of performing standard associative learning. Simple examples of function learning includes estimating the distance of a

moving object according to its size on the retina, or how long you can stay in the sun before you burn.

1.2 Function learning and its application to forest fires

Another situation where learning a function is necessary is when one must estimate the speed of spread of forest fires [3, 4]. When the slope of the terrain and the wind direction are in opposition, a forest fire spreads uphill at a speed negatively related to wind speed, unless the wind becomes strong enough to overcome the fire's natural propensity to spread uphill. From this moment on, the fire will spread downhill at a speed positively related to the increasing wind speed. Overall, the function relating speed of spread to wind speed is a concave quadratic function where the vertex indicates the point at which the force applied by the wind overpowers the tendency of the fire to spread uphill.

Another important aspect of firefighting is the use of back-burning fires to control the reach of the to-be-controlled fire. Back-burning fires are lit and managed by firefighters to starve the to-be-controlled fire of fuel. Usually, a back-burner is lit when the wind speed is low; otherwise the firefighters might lose control of this second fire.

The type of fire (back-burning, to-be-controlled) is an important cue which facilitates the estimation of a fire's spreading speed: instead of considering a quadratic function to determine the propagation of the fire, firefighters can use two linear functions: a decreasing function associated to the back-burning context and an increasing one associated to the to-be-controlled context. This two-stage decision process (cue identification and response selection) is equivalent to a mixture-of-experts architecture [1, 2] which identifies the context first and uses a different, linear, expert accordingly. In [3], experienced firefighters were shown to use this two-stage strategy. Lewandowsky and Kirsner argued that the association between the context (type of fire) and the linear functions had been learnt through their many years of experience. Accordingly, it was argued that this two-stage process was ubiquitous in expertise: this theory was called knowledge partitioning (KP).

To test the KP theory, another experiment was designed to assess whether novices would also use this strategy in a function learning task [4]. In this second experiment, the participants were taught basic firefighting background knowledge and trained in a standard function learning experiment using a concave quadratic function. Every stimulus (wind speed) was also accompanied by a context label, which was systematically associated to a different half of the function during training. This manipulation aimed at recreating the bias present in experienced firefighters' knowledge, for which back-burners are usually encountered in low wind speed situations and to-be-controlled fires in high wind speed conditions. At test, every stimulus was presented twice: once as a back-burner and once as a to-be-controlled fire. Results showed that participants easily achieved the task but, more importantly, that while spreading speeds were almost perfectly estimated when wind speeds appeared in their usual context, they were systematically underestimated when shown in the unusual context. Hence, participants' gave dramatically different responses to identical stimuli presented in different contexts, which supports the KP theory.

This support for the KP theory constitutes empirical evidence in favor of the psychological plausibility of mixture-of-experts models [1, 2]. In particular, one such model was proposed to explain human performance: POpulation of Linear Experts (POLE) [5]. In POLE, when a stimulus is encountered, a gating mechanism directs it to the correct expert, which represents one of many linear functions with different slopes and intercepts. There are enough experts to cover the entire stimulus space and only the gating system has adjustable weights. Once an expert is chosen, it computes the answer accordingly.

This model [5] possesses three important properties. First, POLE accounts for all past results in the function learning literature by using KP. Second, experts do not blend together. Therefore, the system always commits to a cue-value and chooses an expert accordingly (KP). Third, each expert represents a linear relationship between the stimulus and the response. These properties and the preceding empirical results were used to make certain claims concerning the generality and properties of KP [4, 5]: 1) KP is always used when the association between the context and a part of the function is systematic, 2) the usefulness of KP is proportional to task difficulty, and 3) humans always partition into *linear* subfunctions to achieve complex function learning tasks.

In the present study, we empirically tested the preceding claims related to KP in general [3, 4] and to POLE in particular [5]. Human data were collected by altering Lewandowsky et al.'s [4] experimental settings. A first group performed the same task as Lewandowsky et al. [4]. A second group was trained in the same task with smaller stimuli: this task was hypothesized to be more difficult and should lead to an equal or higher proportion of partitioners. The third group was trained with settings identical to the second, except that information was added to the display (constant visual markers). Because the necessary conditions for finding KP were present in this condition (e.g. a systematic association between particular values and a context), we should find as many partitioners as in the small stimuli condition.

2 Experiment

2.1 Method[1]

2.1.1 Participants

Fifty-four undergraduate students from the Université de Montréal participated in this experiment. Eighteen participants were trained in a reproduction of Lewandowsky et al. [4] (control group), eighteen were trained with small stimuli (small stimuli group), and the remaining participants were trained with small stimuli and supplemental

[1] This experiment is an extension of Lewandowsky et al.'s Experiment 1, systematic condition [4]. Therefore, this section bears on their original methodology. However, it was brought to our attention that our participants were given more extensive background knowledge. Nevertheless, performance was not qualitatively different, as shown by the results from the control group.

information (extra information group). In each group, six participants were assigned to the complete condition, six to the left-only condition, and the remaining six to the right-only condition. Participants in the complete conditions received 7$ as compensation for their time, and those in the left-only or right-only conditions received 5$. The experiment was conducted in French.

2.1.2 Material

Participants were tested individually. All instructions and stimuli were presented on 43 cm (17 inch) monitors connected to PCs. Participants were positioned approximately 60 cm away from the monitor. The experimental task was programmed using Sun Microsystems' Java J2SDK1.4.1. The program was used to present the material and record the participants' answers.

2.1.3 Stimuli

Participants were expected to learn a concave quadratic function in which the fire's spreading speed (F) was related to wind speed (W) in the subsequent manner: $F(W) = 24.2 - 1.8W + 0.05W^2$. Wind direction always opposed slope, and the vertex of the function ($W = 18$) represented the point at which the force of the wind balanced the effect of the slope. To the left of that point, fire speed decreased with increasing wind speed, without changing the direction of the fire spread. Lewandowsky et al. [4] referred to these fires as "slope-driven". To the right of the vertex, fires were "wind-driven" and their speed increased as a function of wind speed. During training, 36 stimuli were used, ranging from wind speeds of 0 to 36, omitting the vertex of the function. At test, the omitted wind speed of 18 was included, resulting in a total of 37 transfer stimuli.

On each trial, a horizontal arrow, whose length was proportional to a particular wind speed (henceforth referred to as the stimulus), was shown at the top of the display. The minimal arrow length, associated with the value 0, was approximately 5.8 cm for the small stimuli and extra information groups and 0.7 cm for the control group. The maximal length, associated with the value 36, was approximately 26 cm for the small stimuli and extra information groups and 31 cm for the control group. Thus, in the small stimuli and the extra information groups, the shortest arrow occupied 1/6 of the display and the longest 5/6. In the control group, the arrows spanned the entire monitor. No numerical values for wind or fire speed were shown. Participants were to consider each fire in a context represented both by a color-coded verbal label and arrow (blue for *Back-burning* and red for *Firefighting*). In the extra information group, visual markers were added to the display to indicate the minimal and maximal possible stimulus lengths. The markers were the only difference between the small stimuli group and the extra information group.

Participants were asked to predict the speed of the fire (notwithstanding its direction of spread) by moving a sliding pointer along a 23.3 cm-scale positioned in the left part of the display. The scale was labeled *slow* at the bottom and *fast* at the top, without any incremental values or tick marks.

After each training trial, the participant's response was followed by a feedback arrow. The arrow was located next to the response scale to indicate the correct speed of spread. Also, a message appeared in a rectangle at the bottom center of the screen

to encourage the participant to perform better (yellow rectangle) or to indicate that the response was satisfying (green rectangle). Predictions deviating by 5 or more units (approximately 7.2 cm) from the correct answer were accompanied by the former (yellow message) while acceptable performances were accompanied by the latter (green message). Participants were required to acknowledge feedback by a mouse click. The inter-stimulus interval (ISI) was 2 seconds, and the textual context-label always preceded the stimulus by 1 second. At test, feedback was absent.

2.1.4 Procedure

The procedure was identical for all groups and varied according to conditions. In all conditions, each stimulus was presented five times during training. Hence, there was a total of 180 trials for the complete conditions, but only 90 trials for the left-only and right-only conditions (because training was restrained to one half of the function). In all conditions, 90% of fire speeds occurred in their respective contexts, and the remaining 10% were presented in the opposite context. However, in the left-only and the right-only conditions, all stimuli were presented in the same context (back-burning for left-only and firefighting for right-only). All magnitudes were presented once within each block of 36 trials (18 for the left-only and the right-only conditions), except during the first block, where magnitudes were presented in a blocked manner.

After completion of the training trials, participants in all conditions completed the same transfer test. The transfer test involved predicting the fire speed of all stimuli in both contexts.

2.2 Results

2.2.1 Training

The participants' *Absolute Deviation from Function* (ADF) was used to evaluate learning[2]. Learning curves are shown in Fig. 1. As seen, participants in all conditions from all groups improved their ADF and were thus able to learn the function. Also, Fig. 1 suggests no effects of groups or conditions.

A Group (small stimuli vs. extra information vs. control) × Condition (complete, left-only, right-only) × Block (5, repeated measures) ANOVA was performed on the participants' ADF to corroborate what Fig. 1 hinted. First, the participants were able to diminish their ADF with practice: The mean ADF was 2.33 in the first block and diminished to 1.74 in the fifth block ($F(4, 176) = 14.32$, $p < .01$). However, this effect must be interpreted with care, because the Block × Group interaction was significant ($F(8, 176) = 10.24$, $p < .01$). Thus, the group effect was further decomposed within each block. The groups significantly differed in the first block of training ($F(2, 44) = 28.42$, $p < .01$) but were similar in all other blocks (all $F(2, 44) < 1.63$, $p > .05$). *Tukey A post hoc comparisons* showed that the control group was significantly better than the other two at the beginning of the

[2] The performance of one participant from the small stimuli group, complete condition, deteriorated with practice ($F(4, 175) = 2.54$, $p < .05$). Therefore, this participant was excluded from the following analyses.

task (both differences > 0.96, $p < .01$). However, as suggested by the absence of group effect in the remaining blocks, this difference disappeared with training.

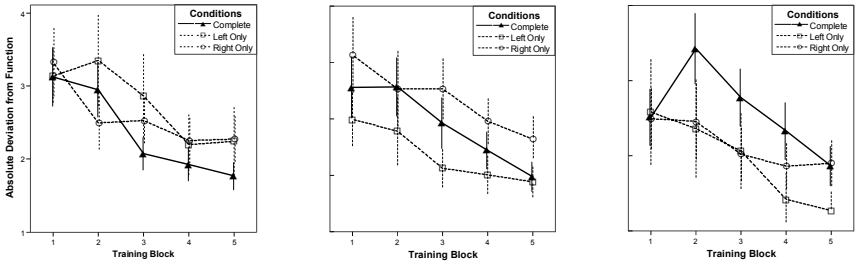


Fig. 1. Participants' ADF during the training phase. The left panel shows performance of participants in the extra information group, the middle panel participants in the small stimuli group and the right panel shows the control group

2.2.2 Group performance at test

KP can be detected experimentally by a difference in responses to a given stimulus in different contexts [4]. Fig. 2 shows transfer performances for participants trained in the complete conditions. As seen in the left panel, participants trained with extra information learned the function quite well. Surprisingly, answers in both contexts matched the quadratic function and were not affected by context.

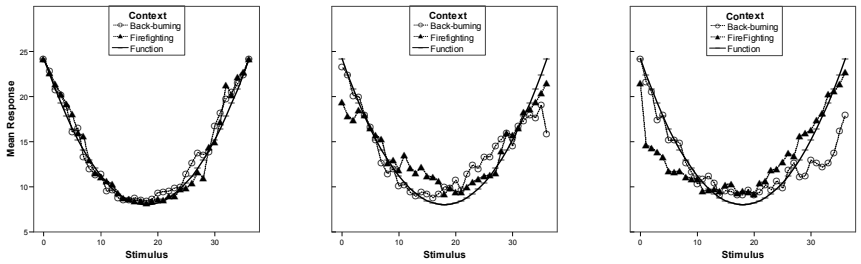


Fig. 2. Mean responses at test in each context. Panels represent the same groups as in Fig. 1

Responses of participants trained with small stimuli are shown in the middle panel. As expected, their responses at test were affected by the context (compare with the left panel), but in an unexpected way. In comparison, the deviations found by Lewandowsky et al. [4] were systematic: low wind speeds resulted in an underestimation of the speed of fire spread in the firefighting context and high wind speeds accordingly resulted in underestimations in the back-burning context. This is exactly the pattern of results found in the control group (see the rightmost panel). In the middle panel, the underestimations are present (to a lesser extent), but mid-range wind speeds were overestimated.

A better way to highlight the difference in responses to a given stimulus is to compute the signed differences [4]. A signed difference is computed by subtracting the answer given at test to each stimulus in the back-burning context from the answer given to the same stimulus in the firefighting context. Signed differences randomly aggregated around zero would suggest the absence of partitioning, while signed differences systematically deviating in one direction would indicate the presence of partitioning.

The left panel of Fig. 3 confirms that participants trained with extra information are not partitioning their knowledge: their signed differences are randomly aggregated around the abscissa. Participants in the control group did show the expected pattern of results: signed differences are negative to the left of the vertex and positive to the right. The signed differences of participants trained with small stimuli are more intriguing (middle panel): they are substantially deviating from the abscissa in a sine-like way.

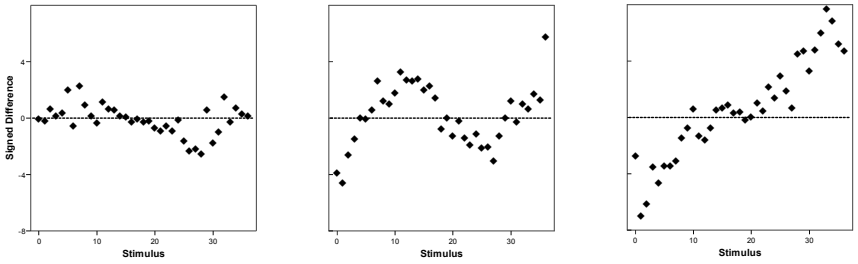


Fig. 3. Signed differences of the participants at test. Panels represent the same groups as in Fig. 1

2.2.3 Individual results at test

Considering that Lewandowsky et al. [4] found important individual differences relating to KP, it is relevant to verify if the effects found in section 2.2.2 were representative of the entire groups of participants. A novel, statistical way of classifying the participants as partitioners (P) or non-partitioners (NP) is to individually plot their signed differences and estimate the best-fitting linear model using a linear regression. A slope which is significantly different from zero suggests a systematic effect of context, namely KP. On the other hand, a slope of zero suggests no clear effect of context. Table 1 shows the slope and intercept individually estimated for each participant.

Table 1 shows that all but one participant fit a model with an absolute slope of 0.05 or less in the group trained with additional information. Because these slopes did not differ significantly from zero (p = .05), these participants were classified as NPs. The exact opposite was true of participants in the small stimuli group: All but one participant had an absolute slope greater than 0.20. Hence, these participants were classified as Ps. In the control group, the best-fitting slope of two of the six participants was smaller than 0.10: these slopes did not significantly differ from

zero (p = .05) and these participants were classified as NPs. The four remaining participants were classified as Ps.

The proportion of Ps in the small stimuli group significantly differed from the proportion of Ps in the extra information group according to a binomial test ($B(5, 1/6) = 4$, $p < .01$). The proportion of Ps in the small stimuli group (80%) is well in range with past literature[3] while the proportion of Ps in the group trained with extra information (16.7%) is below past results. The proportion of Ps in the control group (67%) is similar to Lewandowsky et al.'s results [4] and does not significantly differ from the small stimuli group ($B(6, 4/5) = 4$, $p > .05$). However, this proportion of Ps differs from the proportion found in the extra information group ($B(6, 1/6) = 4$, $p < .01$).

Table 1. Estimated Parameters for the Best-Fitting Linear Models

		Estimated			
	Participant	Slope	Intercept	r^2	Classification
Extra Information					
	110	-0.03	0.63	0.02	NP
	111	0.00	-0.13	0.00	NP
	112	-0.25	5.48	0.34	P
	120	-0.02	0.37	0.01	NP
	121	0.05	-1.45	0.07	NP
	122	0.01	-1.51	0.00	NP
Small Stimuli					
	210	-0.41	8.03	0.58	P
	211	-0.02	0.19	0.00	NP
	212	0.22	-4.19	0.29	P
	220	0.63	-9.04	0.93	P
	221	-0.23	2.68	0.32	P
Control					
	310	0.60	-10.7	0.87	P
	311	-0.02	0.17	0.01	NP
	312	-0.07	2.16	0.09	NP
	320	0.54	-9.45	0.69	P
	321	0.13	-2.61	0.14	P
	322	0.65	-8.88	0.89	P

Note. P = Partitioners; NP = Non-Partitioners

Together, these results suggest that when extra information is present in the display, fewer participants use the KP heuristic, even if the added information is far less useful than the context. Also, it is noteworthy that all the Ps in the control group showed positive slopes, which is consistent with the linear experts hypothesis [4, 5]. However, half of the Ps in the small stimuli group and the only P in the extra information group had negative slopes, which is consistent with the sine-like pattern of Fig. 3. The overestimation of moderate wind speeds is also present in the middle

[3] Precisely, previous research found between 13% [6] and 50% [5] of participants who were not partitioning their knowledge.

panel of Fig. 2 and further inspection of the middle panel suggests a partitioning of the stimuli in two quadratic functions with skewed vertices. Therefore, diminishing the stimulus' length does not prevent participants from using KP but entails a different, non-linear, type of partitioning, which is not consistent with POLE's predictions [5].

2.2.4 Independence of knowledge parcels

As Lewandowsky et al. [4] pointed out, participants who were uniquely trained on the left or right part of the function represent extreme cases of KP: they possess a single expert, associated with a single context. Therefore, if the knowledge of Ps in each context is truly independent, their responses should be similar to the left-only condition in the back-burning context and the right-only condition in the firefighting context. In the case of non-linear Ps, responses in the back-burning context were similar to responses from participants uniquely trained in this particular context (left-only condition: $r = 0.87$). However, the correlation between partitioners' responses in the firefighting context and those from the right-only condition was smaller ($r = 0.69$). This difference is significant according to Fisher's Z transform test ($Z = 2$, $p < .05$). Therefore, the back-burning parcel of knowledge seems more hermetic than the firefighting parcel. Also, results from Lewandowsky et al. suggested higher correlation coefficients [4].

In the case of linear Ps, responses from knowledge partitioners were similar to responses from participants trained in the left-only ($r = 0.81$) and right-only ($r = 0.83$) conditions (in the back-burning and firefighting contexts respectively). Also, the difference between correlation coefficients is not statistically significant ($Z = 0.25$, $p > .05$). Knowledge about the other half of the function acquired in another context did not affect the participants' responses, suggesting that knowledge was completely partitioned.

3 General discussion

In the Experiment, the usual settings used to assess the presence of KP [4, 5] were varied to check the robustness of this phenomenon. Precisely, two modifications were made: reducing the stimulus range, and adding potentially distracting information. First, results from the control group confirmed the adequacy of our reproduction of [4]. Second, it is well established that diminishing the span of the stimuli increases discrimination difficulty [7], hence making stimulus estimation more difficult. In the small stimuli group, participants, classified using our novel method, used KP in an expected proportion, but showed negatively-sloped best-fitting linear models (Table 1). This counter-intuitive result was first hinted by sine-like signed differences (Fig. 3) and the use of non-linear expert functions with skewed vertices (Fig. 2). Finally, adding less useful information to the display was sufficient to prevent most participants from using KP to achieve the task, even if the sufficient conditions for the use of this strategy were present. However, these participants, who did not use KP to simplify the function, were still able to learn it (as shown by an absence of group effect in the ANOVA).

3.1 Implications for current cognitive modeling

Our findings have numerous implications for cognitive modeling. In particular, results from the small stimuli group are challenging the POLE model [5]: when stimuli are more difficult to estimate, participants still partition their knowledge but non-linear experts are used. This can be explained by the added difficulty in the estimation of the function's vertex: if the range of applicability of an expert is fuzzy, the other experts must try to compensate. This strategy is adaptive because, by using more complex functions, the error resulting from an erroneous choice of expert is minimized. Hence, the results from the small stimuli group, while challenging to POLE's predictions, do not invalidate mixture-of-experts models in general [1, 2].

The results from the extra information group are more problematic to both POLE [5] and general mixture-of-experts models [1, 2], because they show that when potentially distracting information is present in the display, participants do not seem to be using the KP heuristic. Instead, participants are learning the quadratic function by simple associative learning. These findings might still be explained by the degenerate case of the mixture-of-experts, in which a single quadratic expert is used.

Together, these results confirms that KP [3-6], which is the empirical counterpart to mixture-of-experts models [1, 2], is a strategy used to achieve psychological tasks. However, this heuristic is less ubiquitous than Lewandowsky and his colleagues previously thought [5], and the constraint of using linear experts is too restrictive. Therefore, mixture-of-experts are adequate models of human cognition but further research is needed to detect the presence of experts (to distinguish simple associative learning from the degenerate case of using a single expert) as well as to determine the nature of the experts used to achieve particular tasks.

References

1. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive Mixtures of Local Experts. Neural Computation **3** (1991) 79-87
2. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, New York (1995)
3. Lewandowsky, S., Kirsner, K.: Knowledge Partitioning: Context-Dependent Use of Expertise. Memory & Cognition **28** (2000) 295-305
4. Lewandowsky, S., Kalish, M., Ngang, S. K.: Simplified Learning in Complex Situations: Knowledge Partitionning in Function Learning. Journal of Experimental Psychology: General **131** (2002) 163-193
5. Kalish, M.L., Lewandowsky, S., Kruschke, J.K.: Population of Linear Experts: Knowledge Partitioning and Function Learning. Psychological Review **111** (2004) 1072-1099
6. Yang, L.-X., Lewandowsky, S.: Context-Gated Knowledge Partitioning in Categorization. Journal of Experimental Psychology: Learning, Memory, and Cognition **29** (2003) 663-679
7. Goldstein, E.B.: Sensation & Perception. 5th edn. Brooks/Cole Publishing Company, Pacific Grove (1999)

Image Processing and Intelligent Multi-Media

Song Form Intelligence for Streaming Music across Wireless Bursty Networks

Jonathan Doherty, Kevin Curran, Paul Mc Kevitt

School of Computing and Intelligent Systems
Faculty of Engineering
University of Ulster, Magee Campus,
Derry/Londonderry, BT48 7JL, N. Ireland
{Doherty-J22, KJ.Curran, P.McKevitt}@ulster.ac.uk

Abstract. Preliminary research on the development of a system for streaming audio across a wireless network, whilst using Song Form Intelligence (SoFI) to correct bursty errors, is presented. Current problems identified with streaming audio across wireless networks are reviewed. Recent approaches on error concealment when bursty errors occur, and Music Information Retrieval are discussed. We propose an approach that uses an amalgamation of network approaches and music information retrieval techniques to solve streaming music problems. Initial findings indicate that this approach will benefit such problems.

1 Introduction

Streaming media across networks has been a focus for much research in the area of lossy/lossless file compression and network communication techniques. However, the rapid uptake of wireless communication has led to more recent problems being identified. Traffic on a wireless network can be categorised in the same way as cabled networks. File transfers cannot tolerate packet loss but can take an undefined length of time. 'Real-time' traffic can accept packet loss (within limitations) but must arrive at its destination within a given time frame.

Forward error correction (FEC) [1] which usually involves redundancy built into the packets, and automatic repeat request (ARQ) [1] are two main techniques currently implemented to overcome the problems encountered. However bandwidth restrictions limit FEC solutions and the 'real-time' constraints limit the effectiveness of ARQ. The increase in bandwidths across networks should help to alleviate the congestion problem. However, the development of audio compression including the more popular formats such as Microsoft's Windows Media Audio WMA and the MPEG group's mp3 compression schemes have peaked and yet end users want higher quality through the use of lossless compression formats on more unstable network topologies.

When receiving streaming media over a low bandwidth wireless connection, users can experience not only packet losses but also extended service interruptions. These dropouts can last for as long as 15 seconds. During this time no packets are received and, if not addressed, these dropped packets cause unacceptable interruptions in the audio stream. A long dropout of this kind may be overcome by ensuring that the buffer at the client is large enough. However, when using fixed bit rate technologies such as Windows Media Player or Real Audio a simple packet resend request is the only method of audio stream repair implemented.

1.1 Objectives of Song Form Intelligence (SoFI)

The principle behind the research presented here is to develop a streaming audio system called *SoFI* that uses pattern matching techniques. The core objectives of SoFI are:

- To match the current section of a song being received with previous sections.
- To identify incomplete sections and determine replacements based on previously received portions of the song.
- To use cognitive techniques to perform error concealment of the packet loss based on similarity analysis.

Satisfying these objectives requires investigation into areas including current approaches to packet loss, audio similarity analysis to satisfy the pattern matching constraint. Next, in section 2 we look at research related to network approaches to error concealment and research in the field of Music Information Retrieval that uses pattern matching techniques. In section 3 we present an overview of the architecture of SoFI. Finally, in section 4 our conclusions and future work are explained.

2 Related Work

Packet delay from network congestion has been partially alleviated using routing protocols and application protocols such as real-time transport protocol (RTP) that have been developed to assign a higher priority to time dependant data. However, it is also the case that some servers *automatically dump* packets that are time sensitive, so streaming applications have had to resort to 'masking' the packets by using HTTP port 80 so packets appear as normal web traffic.

The latest addition to network protocols specifically addressing 'real-time' communication include Voice over Internet Protocol (VoIP), a technology that allows telephone calls using a broadband Internet connection across a packet switched network instead of a regular (or analog) phone line.

2.1 Network Approaches to Error Concealment

Solutions to the inherited problems within streaming audio have included research into a number of varying techniques. The probability of packet loss across bursty

networks has been modelled where time delay is used to control the flow of packets and measure the difference between the current time and the time the packet arrives [2]. This technique can be used to predict network behaviour and adjust audio compression based on current network behaviour. Higher compression results in poorer quality audio but reduces network congestion through smaller packets. A variation of this theme has been used to create new protocols that allow scalable media streaming [3].

Randomising packet order to alleviate the large gaps associated with bursty losses was implemented, where the problem was reduced by re-ordering the packets before they are sent and reassembling the correct order at the receiver [4]. This reduced the bursty loss effect since packets lost were from different time segments. Although nothing is done to replace the missing packets, overall audio quality had improved through smaller gaps in the audio – albeit more frequent.

A number of techniques that use some form of redundancy where repetition is used to replace lost audio segments have been developed. Sending packets containing the same audio segments but with a lower bit-rate alongside the high bit-rate encoding increases the likelihood of packet arrival but at the loss of audio quality, as well as increasing the overall network bandwidth usage [1]**.** Another approach to using redundancy in the form of unequal error protection (UEP) was developed, where improvement is achieved with an acceptable amount of redundancy using advanced audio encoding (AAC) [5]. Segmentation of the audio into different classes such as drumbeats and onset segments allows priority to be applied to more important audio segments with ARQ applied to high priority segments and reconstruction techniques for the replacement of low priority segments based on the AAC received in previous segments.

One of the most recent methods of interpolation of low bit-rate coded voice is used where observation of high correlation of linear predictors within adjacent frames allowed descriptions to be inserted using linear spectral pairs (LSP), and then reconstruct lost LSPs using linear interpolation [6].

2.2 Music Information Retrieval

One of the core aspects of this research is to use pattern matching to identify similar segments within an audio file. Research into the field of music information retrieval (MIR) has gathered momentum over the past decade. With the increase of audio file sharing across heterogeneous networks, a need has arisen for more accurate search/retrieval of files. Research into the analysis of audio has led to the development of systems that can browse audio files in much the same way as search engines can browse web pages retrieving relevant data based on specific qualities [7], [8], [9].

Recent work in pattern matching within polyphonic music has shown that similarity within different sections of a piece of music can aid in both pattern matching for searching large datasets and pattern matching within a single audio file [14], [15], [17]. Results from analysis of an audio stream are stored in a similarity matrix used in [14] which can be seen in Fig 1. Using short time Fourier transform (a variation of discrete Fourier transform which allows for the influence of time as a

factor) to determine the spectral properties of the segmented audio. A chroma based spectrum analysis technique was used to identify the chorus or refrain of a song by identifying repeated sections of the audio waveform with the results also being stored in a similarity matrix [16].

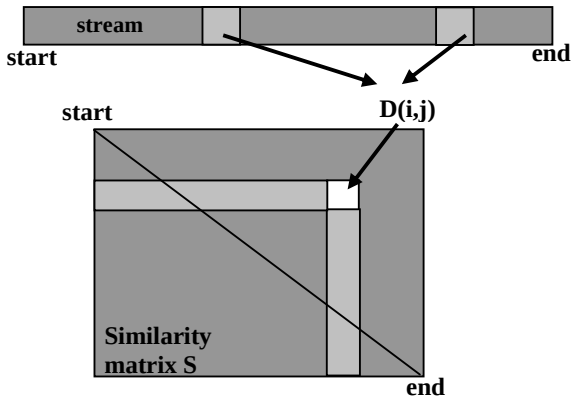


Fig. 1. Embedding an audio stream into a two dimensional similarity matrix [14]

2.3 Audio Complexity

Two inherent problems associated with MIR are the complexity of audio and the complexity of the query [10]. Music is a combination of pitch, tempo, timbre, and rhythm, making analysis more difficult than text. Structuring a query for music is made difficult owing to the varying representations and interpretations including natural transitions in music. Monophonic style queries usually perform better where simple note matching can be used whereas polyphonic audio files and queries simply compound the problem. Adding to the complexity of music structure and query structure is the method of analysis of audio.

The format of an audio file limits its type of use, different file formats exist to allow for better reproduction, compression and analysis. Hence it is also true that different digital audio formats lend to different methods of analysis. Musical Instrument Digital Interface (MIDI) files were created to distribute music playable on synthesisers of both the hardware and software variety among artists and equipment and because of its notational style allows analysis of pitch, duration and intensity [11]. An excellent tool for analysis of MIDI files is the MIDI Toolbox [12] which is based on symbolic musical data but signal processing methods are applied to cover such aspects of musical behaviour as geometric representations and short-term memory. Besides simple manipulation and filtering functions, the toolbox contains cognitively inspired analytic techniques that are suitable for context dependent musical analysis, a prerequisite for many music information retrieval applications.

However, reproduction of a MIDI file can vary greatly on different machines simply from differences between the composers and listeners equipment and it is

because of this it is not used for general audio playback. Pulse code modulation (PCM) is a common method of storing and transmitting uncompressed digital audio. Since it is a generic format, it can be read by most audio applications similar to the way a plain text file can be read by word-processing applications. PCM is used by Audio CDs and digital audio tapes (DATs). Support for WAV files was built into Windows 95 making it the de facto standard for sound on PCs. This format for storing sound in files in PCs was developed jointly by Microsoft and IBM.

One of the most common formats for audio compression is mp3, defined by the Moving Picture Experts Group (MPEG). The mp3 format uses perceptual audio coding and psychoacoustic compression to remove all the audio the ear cannot hear. It also adds a modified discrete cosine transform (MDCT) that implements a filter bank, increasing the frequency resolution 18 times higher than that of layer 2. The result in real terms is mp3 coding shrinks the original audio signal from a CD (PCM format) by a factor of 12 without sacrificing sound quality, i.e. from a bit rate of 1411.2 kbps of stereo music to 112-128 kbps. Because MP3 files are small, they can easily be transferred across the Internet. MPEG 7 [13] is a standardised description of various types of multimedia information. Where MPEG 4 defines the layout and structure of a file and codecs, MPEG 7 is a more abstract model that uses a language to define description schemes and descriptors – the Description Definition Language (DDL). Using a hierarchy of classification allows different granularity in the descriptions. All the descriptions encoded using MPEG 7 provide efficient searching and filtering of files.

3 Song Form Intelligence

Methods for error correction when packet loss occurs as discussed in Section 2.1 mainly try to minimise/prevent errors in the audio stream by masking missing or late packets with extra audio encoding or some method of interpolation to 'smooth' over the missing packets. Now we propose the use of pattern matching techniques within streaming audio to replace the lost segments.

The song header depicted in Fig. 2 describes a piece of music following a typical western tonal format (WTF), with a song form of intro (*I*), verse (*V*), chorus (*C*), verse (*V*), and chorus (*C*). It states that there is an introduction section of 10 seconds duration followed by a verse of 28 seconds, then a chorus of 32 seconds, then a verse of 28 seconds and finally repeats the chorus of 32 seconds.

I	10	*V*	28	*C*	32	*V*	28	*C*	32

Fig. 2. Song form structure

This research proposes a novel syntax audio error concealment buffering technology, made possible by the song form structure with the possibility of developing this in the field of music semantics for replacing unidentified portions of

the song structure. Modelling of music has given rise to a number of different research angles such as modelling the human mind's conscious perception of rhythm and its syntax and semantics [17], [18], [19].

One of the problems associated with streaming audio is the time factor. The time between when a packet is received, placed into the buffer in the correct order and then used for playback can be anything between 10 seconds to as little as microseconds, depending on delays from bandwidth and congestion. It is at this point that ARQ techniques fail as there is simply no time left for a new packet to arrive. However if the segment already exists in previous sections already received replacements can be used.

Analysis of an audio file using the MPEG 7 description scheme will allow frames to be tagged with the data stored in the header file of each packet prior to broadcasting. These descriptors will be based on the similarity between different sections of the song. As can be seen in Fig. 2 an almost exact match can be obtained where the chorus is repeated. Identification of lost segments in the second chorus can simply be replaced with segments already received in the first chorus. By identifying the beginning of the missing segment, and then the end of the missing segment, a replacement can be found by matching the preceding and following segments with the same-length section in the previous chorus. To ensure smooth replacement for exact matches, the audio packets will need to be of the same duration, have the same start time, and the same end time for each segment. This should ensure error concealment occurrences within chorus sections will be inaudible to the listener. The overall SoFI system architecture can be seen in Fig. 3 showing analysis of the audio performed on the server with the packet replacement process being performed on the client.

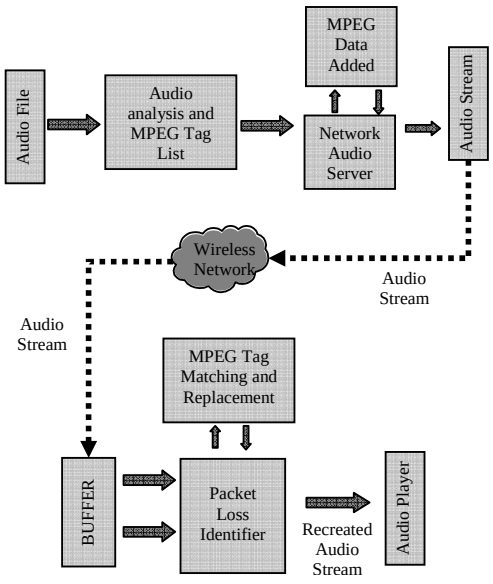


Fig. 3. System architecture

Not all songs have exactly repeating choruses, the underlying music and keys may be the same, but subtle differences in its pitch or even lyrics can have dramatic effects on matching segments. What appears to be the same to the human ear is very different when analysed by wavelength. With the use of 'best effort' matching, bursty losses can still be corrected with minimal perception to the listener.

Sections of the audio that contain lyrics that are different from any other section can still be repaired when bursty losses do occur. Pauses between words, phrases and sentences where only the music can be heard allow for repair in that repetition is inherent in WTF songs. Background instruments can follow the same repeated pattern throughout the entirety of the song. The following guitar chords are an extract of the chorus from R.E.M.s' "Everybody Hurts": *E minor / E minor / A / A / E minor / E minor / A / A / E minor / E minor / A / A.* Similarly the intro and verses contain a similar pattern: *D /G / D / G / D / G / D*. Pattern matching will require identification and grouping of these sections prior to streaming.

For 'best effort' matching of sections of the song with lyrics or unique sections of audio will be done using probability based on the already attached tags defining each of the sections and as near a possible match from previously received sections will be used to fill the missing segments. This will however reduce the overall quality of the audio signal, but based on the assumption that the typical length of bursty losses is no more than 1-2 seconds, it is acceptable for some degradation to occur. Studies of acceptable levels of audio quality have shown that listeners prefer to have some form of replacement rather than silence, by maintaining a rhythmic pattern using a percussive sound synthesiser to replace missing segments allows some continuity for the listener when dropouts do occur [20].

The minor increase in bandwidth from the inclusion of the MPEG 7 tags in the header sections of the packets can be justified based on the complexity of the analysis required. Calculations for real-time pattern matching will require a vast increase in processing demands on the system; to perform these at the same time as performing packet loss identification and matching lost sections, within the timescale of the buffer is not feasible.

4 Conclusion

In many ways the ideas presented in this paper are related to the field of FEC. However our point of departure and underlying methodology are different. Preliminary research indicates that by using similarity analysis and MPEG 7 we can identify and tag similar sections within an audio file and include the data in the audio stream when broadcast from the server. Pattern matching on the client side when receiving the audio allows error concealment through interpolation thereby placing the onus of error concealment on the client. Future research will address implementation and evaluation issues including packet replacement accuracy based on a comparative analysis of the initial song and the actual file received using both computer and human audible perception.

References

1. Perkins, C., Hodson, O., Hardman, V.: A Survey of Packet-loss Recovery Techniques for Streaming Audio. In IEEE Network Magazine, Vol. 12, Issue 5 (1998) 40-48
2. Lee, K.K., Chanson, S.T.: Packet Loss Probability for Bursty Wireless Real-time Traffic Through Delay Model. In IEEE Transactions on Vehicular Technology, Vol. 53, Issue 3 (2004) 929 – 938
3. Mahanti, A., Eager, D.L., Vernon, M.K., Sundaram-Stukel, D.J.: Scalable On-demand Media Sreaming with Packet Loss Recovery. In IEEE/ACM Transactions on Networking, Vol. 11, Issue 2 (2003) 195 – 209
4. Varadarajan, S., Ngo, H.Q., Srivastava, J.: Error Spreading: a Perception-driven Approach to Handling Error in Continuous Media Streaming. In IEEE/ACM Transactions on Networking, Vol. 10, Issue 1 (2002) 139 – 152
5. Wang, Y., Ahmaniemi, A., Isherwood, D., Huang, W.: Content-based UEP: A New Scheme for Packet Loss Recovery in Music Streaming. In Proc. of Eleventh ACM International Conference on Multimedia, Berkeley, CA, USA. (2003) 412 – 421
6. Wah, B., Lin, D.: LSP-based Multiple-description Coding for Real-time Low bit-rate Voice Over IP. In IEEE Transactions on Multimedia, Vol. 7, Issue 1 (2005) 167 – 178
7. Leman, M., Clarisse, L., De Baets, B., De Meyer, H., Lesaffre, M., Martens, G., Martens, J., and Van Steelant, D.: Tendencies, Perspectives, and Opportunities of Musical Audio-mining In Proc. of 3rd EAA European Congress on Acoustics, Seville, Spain (2002)
8. Gomez, E., Klapuri, A., Meudic, B.: Melody Description and Extraction in the Context of Music Content Processing. In Journal of New Music Research, Vol. 32, Issue 1 (2003)
9. Chai, W., Vercoe, B.: Structural Analysis of Musical Signals for Indexing and Thumbnailing. In Proc. of ACM/IEEE Joint Conference on Digital Libraries (2003) 27 – 34
10. Downie, J.S.: The Scientific Evaluation of Music Information Retrieval Systems: Foundations and Future. In Computer Music journal, Vol. 28, Issue 2 (2004) 12 – 23
11. Doraisamy, S., Rueger, S.: A Polyphonic Music Retrieval System Using N-Grams. Presented at the 5th International Conference on Music Information Retrieval, ISMIR 2004, Barcelona, Spain (2004) 204 – 209
12. Eerola, T., Toiviainen, P.: *MIDI Toolbox: MATLAB Tools for Music Research*. University of Jyväskylä, Kopijyvä, Jyväskylä, Finland. Available at http://www.jyu.fi/musica/miditoolbox (2004)
13. Martinez, J.M., Overview of MPEG-7 Description Tools. IEEE Multimedia, Vol. 9, Issue 3 (2002) 83 – 93
14. Foote, J., Cooper, M.: Media Segmentation using Self-similarity Decomposition. In Proc. of SPIE Storage and Retrieval for Multimedia Databases, Vol. 5021 (2003) 167 – 175
15. Meredith, D., Wiggins, G.A., Lemström, K.: Pattern Induction and Matching in Polyphonic Music and other Multi-dimensional Data. In the 5th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI'2001), Vol. X (2001) 61 – 66
16. Bartsch, M. A., Wakefield, G. H.: To Catch a Chorus: Using Chroma-based Representations for Audio Thumbnailing. In Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, NY (2001) 15 – 19
17. Dannenberg, R. B., Hu, N.: Pattern Discovery Techniques for Music Audio. In Proc. of ISMIR 2002 M. Fingerhut, Ed., Paris, IRCAM (2002)
18. Mc Kevitt, P., O'Nuallain, S., Mulvihill, C., (Eds.): Language, Vision and Music - Selected Papers from the 8th International Workshop on the Cognitive Science of Natural Language Processing, Galway, Ireland. Amsterdam, The Netherlands: John Benjamins Publishing Company (2002)
19. Wiggins, G. A.: Music, Syntax, and the Meaning of 'meaning'. In Proc. of First Symposium on Music and Computers (1998) 18 – 23

20. Wyse, L., Wang, Y., Zhu, X.: Application of a Content-based Percussive Sound Synthesizer to Packet Loss Recovery in Music Streaming. In Proc. of 11th ACM International Conference on Multimedia (2003) 335 – 338

Satellite Image Classification - A Contextual Evidence-based Approach

B.M. Al Momani, S.I. McClean and P.J. Morrow

School of Computing and Information Engineering
Faculty of Engineering, University of Ulster, Cromore Rd, Coleraine
Co. Londonderry, BT52 1SA, Northern Ireland
{al_momani-b, si.mcclean, pj.morrow}@ulster.ac.uk

Abstract. Remote sensing imaging techniques make use of data derived from high resolution satellite sensors. Image classification identifies and organises pixels of similar spatial distribution or similar statistical characteristics into the same spectral class (theme). Contextual data can be incorporated, or 'fused', with spectral data to improve the accuracy of classification algorithms. In this paper we use Dempster-Shafer's theory of evidence to achieve data fusion. Incorporating a knowledge base of evidence within the classification process represents a new direction for the development of reliable systems for image classification and the interpretation of remotely sensed data.

1 Introduction

Remote sensing is the observation and measurement of something outside the range of physical contact [1]. It is a process which captures light, in different wavelengths, reflected from objects on the earth's surface (e.g. soil, plants, buildings, etc) with cameras or sensor systems usually mounted on an orbiting satellite. The reflection process allows for the extraction of information about these objects from the data collected by satellite sensors. However, very high resolution satellite sensors produce many challenges in coping with huge amounts of detailed image data. The information gained from a satellite image can be vast and the quality of this information is largely dependent upon the understanding and correct interpretation of the data represented in the image. Image classification is therefore vital as useful information can only be gathered from correct classification of an image. The classification process is based on the assumption that pixels which have similar spatial distribution patterns, or statistical characteristics, belong to the same spectral class. The two main approaches used in image classification are 'supervised' and 'unsupervised' techniques. These two approaches have received considerable attention from the research community which has concentrated on a low-level pixel-based (within the image) approach, based on spectral pixel values only. In contrast, less work has been undertaken using external or contextual data (outside the image), such as Digital Terrain Models (DTM), OS maps or agriculture records. The major challenge is to build a framework and efficient methodologies which bring all types of data together (data fusion) either within the image or outside the image in order to

enhance the classification process. Section 2 of this paper will review traditional low-level classification techniques; section 3 outlines context-aware approaches using contextual information and introduces Dempster-Shafer's theory of evidence; and section 4 provides an example of our approach to data fusion illustrating how Dempster-Shafer theory can be used to improve the overall classification process.

2 Image classification

The main objective of image classification is to convert the numerical data recorded from satellite images into features in a landscape. In order to achieve the classification process, a method of extracting (analysing) multispectral image information, and applying some statistical decision rules, is required to give each pixel in an image a land cover identity (the pixel's labelling) . A thematic map is created upon completion of the classification process and this contains an informational representation of an image which shows the spatial distribution of a particular theme. Themes or classes can be, grass, water, sand, conifer, etc. Fig. 1 illustrates a simplified overview of the classification process.

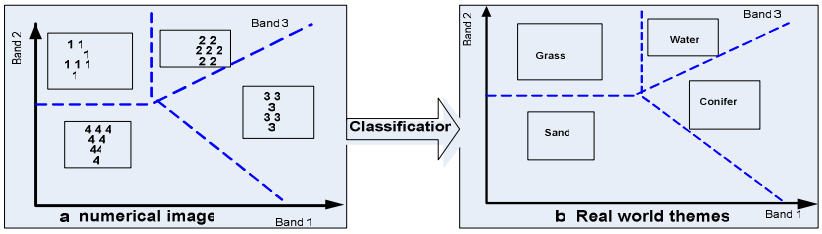


Fig. 1. A simplified overview of the image classification process. (a) Each class (termed spectral class) has a similar grey level in feature space. (b) Each class (termed an information class) is assigned by an expert after/before performing the classification process.

Classification Techniques of relevance to this paper fall into the following broad categories: supervised, unsupervised and semi-supervised (context-based).

Supervised classification. In supervised classification *a priori* knowledge about the area is required before the classification algorithm is performed. Supervised classifiers require the user to decide which classes exist in the image, then delineate samples of these classes, i.e. "training areas", and subsequently generate associated class signatures. Examples of this approach are minimum distance, maximum likelihood, Mahalanobis, Parallelepiped and table lookup classification [1]-[4].

Unsupervised classification. In contrast to the supervised approach, no training data are required in unsupervised classification techniques. In addition, they do not make assumptions regarding the underlying spatial distribution of the classes. Instead, they are based on assigning pixels to classes without prior knowledge of these classes, and then segmenting an image into unknown classes. Pixels are then examined by the algorithm and similar pixels (i.e. those which are 'spectrally homogeneous') are

grouped together based on their values to form clusters. Values within a given class should therefore be close together, whereas data in different classes should be well spaced apart. Since the generated classes are spectral classes with unknown identity initially, then it is the domain expert's role to compare the resulting classes with some type of reference data to assign them to a real land cover data [2], [3], [5]. The two most common examples of unsupervised algorithms are the Iterative Self-Organising DATA Analysis technique (ISODATA) [6] and the K-means algorithm [5], [7].

3. Context-based classification

3.1 Background

Context-based classification techniques have concentrated mainly on using contextual information *within* an image only and neglected other external contextual data. In addition, contextual and environmental information could be extracted by identifying the land cover/use characteristics which might be used as an external source for the classification process. In reality, adjacent pixels are related and for example, if a particular pixel is recognised to be grass then the neighbouring pixels will most likely be grass. Therefore, considering the labels of neighbouring pixels when seeking to assign an appropriate class to a pixel might be a good way to improve the classification process. This is termed *context-based classification* [8]. The degree to which the adjacent pixels are related is dependent on the sensor's spatial resolution and the land cover surface. For example, an agricultural area tends to have a stronger degree of pixel correlation than an urban area. Likewise, neighbouring pixels for images taken by the LandSat sensors tend to be less correlated than those taken by SPOT sensors [2]. Several techniques have been proposed in this area such as the use of neural networks [9], [10], [11], Fuzzy set theory [12], Markov Random Fields [13], [14], Bayesian inference [13]-[17], and Dempster-Shafer evidence theory [18], [19]. These techniques concentrate on low-level approaches which consider *within* image data. The major challenge is to utilize and handle all types of data available and to combine them in a way which enhances the classification process. The contextual data fusion technique considered in this paper is that of Dempster-Shafer and in particular the fusion of *external* contextual data with the results of a maximum likelihood classifier for *internal* image data.

3.2 Dempster-Shafer theory of evidence

Definitions. Evidence theory, of which Dempster-Shafer theory [20] is a major constituent, is a generalisation of traditional probability which allows us to better quantify uncertainty [21], [22]. It is described in Guan and Bell [23], [24], and previous work outlining how it may be used for knowledge discovery and for the combination of heterogeneously classified data may be found in [25] and [26]. The approach provides a means of representing data in the form of a mass function which quantifies our degree of belief in various propositions or sets of values. One of the

major advantages of evidence theory is that it provides a straightforward way of quantifying ignorance and is therefore a suitable framework for handling data which is subject to imprecision. The mass function assigns belief to sets which together form the frame of discernment Ω. The mass function m is defined on subsets of Ω (propositions) as follows:

$$m(\phi) = 0 \tag{1}$$

i.e. the mass function of the null proposition ϕ is always zero;

$$\sum_{A \subseteq \Omega} m(A) = 1 \tag{2}$$

i.e. the sum of the masses of all the propositions in the frame of discernment is one. The main difference between these definitions and conventional probability is that here the propositions may be overlapping. This Dempster-Shafer definition of mass functions may be used to provide a lower and upper bound for the probability assigned to a particular proposition. These bounds are called the belief and plausibility respectively: the belief in a proposition is the sum of masses of all propositions contained in it; the plausibility of a proposition is the sum of the masses of all propositions in which it is wholly or partly contained. The belief and plausibility functions are therefore defined by:

$$Bel\ (A) = \sum_{X \subseteq A} m(X) \quad \text{and} \quad Pls\ (A) = \sum_{X : X \cap A \neq \varnothing} m(X) \tag{3}$$

Alternatively, we may define mass functions in terms of the belief and plausibility. The belief and plausibility functions may thus be used to determine the amount of support for a proposition. They may then be used to induce rules based on the mass allocations for various propositions and may be regarded as providing pessimistic and optimistic measures of how strong a rule might be [27].

Example 3.1: *m({wheat}) = 0.9; m({oats}) = 0.05; m({wheat, oats}) =0.05.*

Here, we are 90% sure that the area refers to wheat, 5% sure that it is oats and 5% sure that it might be either. Hence *Bel({wheat}) = 0.9*; *Pls({wheat})=0.95*.

One of the main advantages of using Evidence Theory for our purposes is that Dempster's law of combination allows us to combine evidence, in the form of mass functions, from different sources. Let m_1 and m_2 be two mass functions on the frame of discernment Ω. Then, for any subset $H \subseteq \Omega$, the *orthogonal sum* $\oplus$ of two mass functions on H is defined as:

$$(m_1 \oplus m_2)(H) = \frac{\sum_{X \cap Y = H} m_1(X) * m_2(Y)}{1 - \sum_{X \cap Y = \phi} m_1(X) * m_2(Y)} \tag{4}$$

The orthogonal sum thus allows two mass functions to be combined into a third mass function, which pools pieces of evidence to support propositions of interest.

Using evidence theory for sensor data fusion. Evidence Theory has been used previously to combine *multisource* sensor data where the data are numerical and mass functions are assigned to various pixel labelling possibilities [2] thus we determine a set of mass functions for each pixel, separately for each source. This may be done using a variety of methods, the most obvious of which is the Bayesian approach using maximum likelihood. Here we determine probabilities for each pixel taking each possible class label. The mass functions are then simply the probabilities attached to each singleton set. So, for each source s_i i=1..s and each pixel p_j j=1..p we derive a probability π_{ijk} = Prob (pixel j has class label k | source i) for k=1, …, K, where

$$\sum_{k=1}^{K} \pi_{ijk} = 1 \text{ for } i = 1,...,s \text{ and } j = 1,...p. \quad (\mathbf{5})$$

Then, $m_{jk}^{(i)} = \pi_{ijk}$ is the mass function for source i and pixel j, associated with the singleton class label {k}. Using this approach we may fuse *numerical multi-source* data using Dempster's combination law. Evidence Theory has also previously been used for *multi-scale* fusion by Hégarat-Mascal et al in [19], where the different data sources collect data at different scales.

4. Knowledge-based approach

4.1 Rationale

Previous techniques which have been used to accommodate spectral and contextual data have some limitations. Besides being complex these techniques are limited to using numeric data only [2]. However, there are many other *non-numerical* contextual data that can be used to enhance the classification decision process which might be termed qualitative analysis, for example, soil maps, road networks etc. A suitable knowledge-based approach, if established for this purpose, could contain all these data and play a vital role in enhancing the classification process. In addition, the problems that need to be solved determine the type of knowledge representation that might be needed and in all cases, facts and rules are needed to construct the knowledge-base. In the remote sensing field, there are a huge variety of data that can be attached to the area of interest. The main idea of building the knowledge-based system is to handle mixed data types that can be attached to a specific satellite image under the classification process. These data can be analyzed separately and then combined together at the label level to get a composite label. Therefore, a type of prior information can be added to improve the data quality acquired for the classification algorithms. This prior information can be termed as *external (contextual)* information such as elevation, OS map, agriculture records, previous classifications and soil type. Fig 2 illustrates the process of fusing contextual knowledge data with spectral image data.

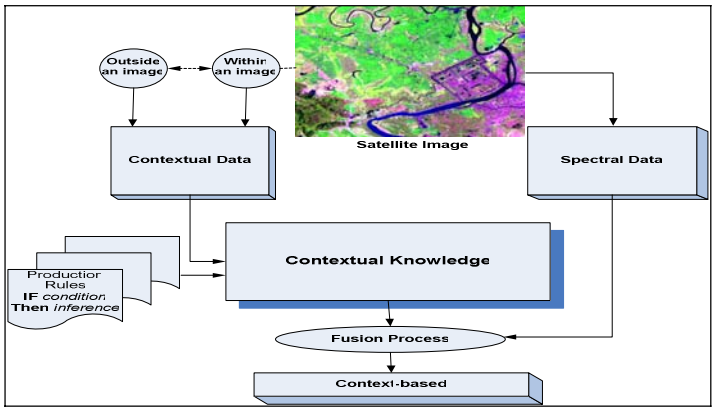


Fig. 2. The proposed approach- fusing the contextual knowledge data with spectral data

Taking maximum likelihood as an example, contextual information can be used as prior information which can be incorporated into the prior probability estimate for the classification decision. Therefore, a set of production rules should be generated for this purpose that can be derived from contextual data as a form of contextual knowledge. This data can be then fused with spectral data using fusion techniques (e.g. Dempster-Shafer).

4.2 Using evidence theory for knowledge-based contextual classification

Our intention is to improve classification of sensed data by combining it with contextual knowledge. Such knowledge may be in the form of deterministic (certain) rules or probabilistic (uncertain) rules. Thus for example, we may have a 'certain' rule such as:

Rule 1: **if** ($L < p_e < U$) **then** class $\neq$ {trees}, where p_e is a pixel's (terrain) elevation, L is a lower elevation value and U is an upper elevation value.

or an 'uncertain' rule such as:

> **if** ($L < p_e < U$) **and** (season ={winter}) **then** class $\neq$ {wheat, barley} with *confidence interval= (b, p).*

Where *b* and *p* are Dempster-Shafer belief and plausibility values, respectively. Here *b* and *p* may be specified by the expert or calculated from mass functions. They have considerable generality since they include:

(i) 'certain' rules where the confidence interval is (1, 1);
(ii) probabilistic rules where the confidence interval is (*p*, *p*);
(iii) rules that relate to sets, as specified in Dempster Shafer theory. This capability allows us to associate our rules with several classes as once, such as {wheat, barley}; and

(iv) we can associate our confidence intervals with linguistic summaries [26], e.g. (1,1) represents "always", (0.9, 1) might represent "typically".

If the rule 'fires', we represent the contextual knowledge as a confidence interval (b, p) which can then be converted into a mass function. This mass function is then combined with other mass functions, representing other pieces of contextual knowledge or other sensor data, using the combination law.

4.3 Experimental example

Figure 3 shows an example source image together with the classified image resulting from the use of the maximum likelihood algorithm. For the purposes of this example we assume that the 7 classes are labelled as wheat, sand, conifer, water, wetland, wood and dirt. The maximum likelihood classification was performed using the spectral pixel values only, calculated using the formula (taken from [2]):

$$f_i(\mathbf{x}) = -\ln|\Sigma_i| - (\mathbf{x}-\mathbf{m}_i)^t \Sigma_i^{-1} (\mathbf{x}-\mathbf{m}_i), \quad \textbf{(6)}$$

where $\mathbf{x}$ is the pixel to be classified, *ln* is the natural algorithm, $\sum_i$ is the variance-covariance matrix estimated from training pixels in class i and $\mathbf{m}_i$ is the mean spectrum of the class i.

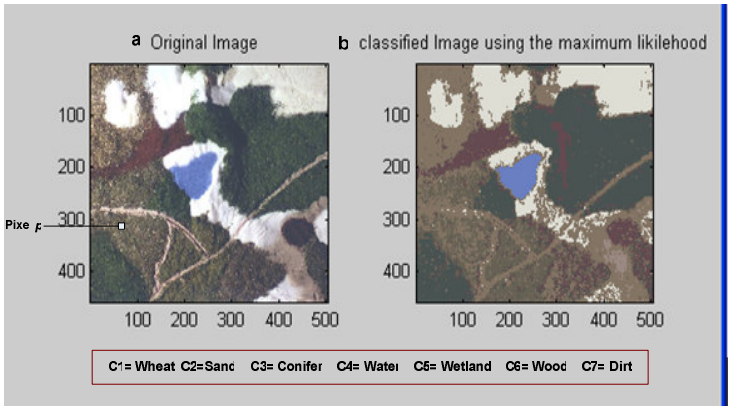


Fig. 3. Classifying a Satellite Image using the maximum likelihood classification using spectral data only

The pixel labelled p in the original image is a vector *(r, g, b)* which needs to be assigned to one of the 7 class labels. The probabilities obtained for each of the classes after applying the maximum likelihood algorithm are shown in Table 1 (MLC column). It can be seen from this table that pixel p would be assigned to c_6 (wood) since it has the highest probability (0.7175). The second largest probability is for c_1 (wheat), and although in this instance there would be little chance of an incorrect classification being made, in many cases misclassification does indeed occur.

Table 1. Results for pixel *p* for the maximum likelihood classifier (MLC) and the combined D-S approach.

Class Number	Class name	Mass	Probability (MLC)	$m_1 \oplus m_2$
c_1	Wheat	$m\{c_1\}$	0.1693	0.0101
c_2	Sand	$m\{c_2\}$	0.0521	0.0621
c_3	Conifer	$m\{c_3\}$	0.0056	0.0066
c_4	Water	$m\{c_4\}$	0.0354	0.0422
c_5	WetLand	$m\{c_5\}$	0.0019	0.0023
c_6	Wood	$m\{c_6\}$	0.7175	0.8550
c_7	Dirt	$m\{c_7\}$	0.0183	0.0219
Total			1	1

To reduce the possibility of a misclassification, we wish to combine rules associated with external data together with the available spectral data. Two sources of evidence are therefore used for classifying the image; the first one is the set of probabilities resulting from the maximum likelihood classifier and the second source of evidence is the contextual knowledge associated with 'elevation' data (i.e. for the area under consideration we would have a DTM and thus an elevation would be associated with every pixel in the image). We combine these sources using the Dempster-Shafer theory discussed above.

The probabilities resulting from the maximum likelihood algorithm can be expressed as masses: $m\{c_1\}= 0.1693$, $m\{c_2\}= 0.0521$ … $m\{c_7\}= 0.0183$. We now define an 'uncertain' rule relating the elevation of a pixel to the possibility of it being classed as wheat:

Rule: **if** $(L < p_e < U)$ **then** class $\neq$ {wheat} (i.e. $\neq c_1$), with confidence interval= [0.95, 1].

The mass functions for this second source of evidence can be expressed as:

$$m\{c_2,\ldots, c_7\} = 0.95 \text{ and } m\{c_1,\ldots, c_7\} = 0.05.$$

Assuming that the rule 'fires', we can now compute the orthogonal sum of two mass functions (equation 4) for each class (for pixel *p*) and thus combine our two sources of evidence:

$$(m_1 \oplus m_2)(\{c_1\}) = \frac{0.05 p_1}{1-0.95 p_1}, \qquad (m_1 \oplus m_2)(\{c_2\}) = \frac{p_2}{1-0.95 p_1}$$

$$(m_1 \oplus m_2)(\{c_3\}) = \frac{p_3}{1-0.95 p_1}, \cdots (m_1 \oplus m_2)(\{c_7\}) = \frac{p_7}{1-0.95 p_1}$$

where $p_1..p_7$ are the MLC probabilities in table 1 and the final column in the table is obtained by substituting these values into the above equations. By comparing the last

two columns of the table we can clearly see that in using the second source of evidence, within a Dempster-Shafer framework, we have improved the chances of a correct classification for the pixel under consideration. The probability of the pixel being classified as wood has increased from 0.7175 to 0.8550 while the probability for wheat has been reduced from 0.1693 to 0.0101. This therefore increases confidence when labelling the satellite image pixels and also enhances the classification process as a whole.

5. Conclusions and future work

The growth in non-numeric (symbolic) contextual data requires classification techniques that can combine both numeric and symbolic data within a single framework. Dempster-Shafer theory can deal with both types of data and as such is well suited to being used in the classification of remotely sensed imagery. This paper presents a novel approach to enhancing satellite image classification by including contextual data within the classification process. In particular we have demonstrated how Dempster-Shafer's theory of evidence can be used to combine two sources of evidence for the classification of pixel data. A knowledge base of production rules can be specified where the rules have different degrees of certainty, defined as confidence intervals and associated mass functions. Data is combined with the original spectral data by calculating the orthogonal sum of mass functions from each source of evidence. Further work is needed to investigate other types of contextual data available from different sources (e.g. OS maps, agricultural records, etc) including multiple images and to construct suitable production rules to incorporate this data into a knowledge based system. In addition, work is also required in defining appropriate accuracy assessment methods and measures to validate our approach including a thorough analysis of the algorithm's performance.

References

1. Paul M. Mather.: Computer Processing of Remotely-Sensed images analysis An Introduction, second edition, John Wiley & Sons (1999), ISBN 0-471-98550-3
2. John, A., Richards. Xiuping Jia.: Remote Sensing Digital Image Analysis an Introduction. Third Edition, Springer (1999). ISBN 3-540-64860
3. Thomas M. Lillesand, Ralph W. Kiefer.: Remote Sensing and Image interpretation, fourth edition, John Wily & Sons (2000), ISBN 0-471-25515-7
4. Xiuping Jia, John A Richards.: Efficient Maximum Likelihood Classification for Image Imaging Spectrometer Data Sets, IEEE Transactions on Geoscience and Remote Sensing, Vol. 3, No. 2, (1994) 274-281
5. T. Duda, M. Canty.: Unsupervised classification of satellite imagery: choosing a good algorithm. International Journals in remote sensing, Vol.23, No. 11, (2002) 2193-2212
6. M. K. Dhhodhi, J. A. Saghri, I. ahmed.: D-ISODATA : A Distributed Algorithms for Unsupervised Classification of Remotely Sensed Data on Network of Workstations. Journal of Parallel and Distributed Computing 59, (1999) 280-301

7. Paul J. Gibson, Clare H. Power.: Introductory Remote Sensing, Digital Image Processing Applications, Routledge, Taylor and francis Group (2000), ISBN 04-415-18961-6
8. Oliver, D., Patrice, L., IsabeeleVan Den, S.: Remotes Sensing Classification of Spectral, Spatial and Contextual Data using Multiple Classifier Systems. Image Anal Stereo1 20(Supp1 1)(2001) 584-589
9. H. Murai, S. Omatsu.: Remote Sensing image analysis using neural and knowledge-based processing. International Journals in Remote Sensing, Vol. 18, No. 4, (1997) 811-828
10. M. Egmont-Peterson, D. de Ridder, H. handles.: Image processing with neural networks – a review. Pattern Recognition 35 (2002) 2279-2301
11. Benediktsson, J.A., Swain, P.H., Ersoy, O.K.: Neural Network Approaches Versus Statistical Methods In Classification Of Multisource Remote Sensing Data. IEEE Transactions on Geoscience and Remote Sensing, Vol.28, No.4, (1990) 540 - 552
12. Fangju, W.: Fuzzy Supervised Classification of Remotely Sensing Images. IEEE Transactions on Geosciences and Remote Sensing. Vol. 28. No. 2. (1999) 194-201
13. Qiong J., David L. (2002).: Adaptive Bayesian Contextual Classification Based on Markov Random Fields. Geoscience and Remote Sensing, IEEE Transactions, Vol. 40. No. 11. (2002) 2454-2463
14. Warrender C.E, Augusteijn, M.F.: Fusion of Image Classification using Bayesian techniques with Markov random Fields. International Journals of Remote Sensing, Vol.20, No.10, (1999) 1987-2002
15. P. Cheeseman, J. Stutz.: Bayesian classification : Theory and results in Advances in Knowledge Discovery and Data Mining, U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds. Menlo Park, CA: AAAI Press, 1996
16. Bourman, C.A., Shapiro, M.: A Multiscale random Field Model for Bayesian Image segmentation. IEEE Trans. Image Processing, Vol.3, No.2, (1994) 162-177
17. Ben Gorte, Alfred Stein.: Bayesian Classification and Class Area Estimation of Satellite Images Using Stratification. IEEE Transactions on Geoscience and Remote Sensing, Vol. 36, No. 3, (1998) 803-812
18. Mascle, S., Bolch, I., Vidal-Madjar, D.: Unsupervised multisources remote sensing classification using Dempster-Shafer evidence theory. In Proc of SPIE- Synthetic Aperture Radar and passive Microwave Sensing, (1995) 200-211
19. S. Le H´egarat-Mascle, D. Richard, C. Ottl´e.: Multi-scale data fusion using Dempster-Shafer evidence theory. Integrated Computer-Aided Engineering 10 (2003) 9–22 IOS Press
20. I.S. Torsun.: Foundation of Intelligent Knowledge-Based Systems. Academic Press (1995). ISBN 0-12-696060-7
21. A.P.: Dempster, A Generalisation of Bayesian Inference, Journal of the Royal Statistical Society B, 30(1968) 205-247
22. G. Shafer.: Belief Functions and Parametric Models, Journal of the Royal Statistical Society B, 44(1982) 322-352
23. J.W. Guan, D.A. Bell.: Evidence Theory and its Applications (Volume 1, North- Holland, 1991)
24. J.W. Guan, D.A. Bell.: Evidence Theory and its Applications (Volume 2, North-Holland,1992)
25. S.S. Anand, B.W.Scotney, M.G. Tan, S.I. McClean, D.A. Bell, J.G. Hughes, I.C. Magill.: Designing a Kernel for Data Mining, IEEE Expert 12 (1997) 65-74
26. S.I. McClean and B.W. Scotney.: Using Evidence Theory for the Integration of Distributed Databases, International Journal of Intelligent Systems, 12 (1997) 763-776
27. R.R. Yager, K.J. Engemann, D.P. Filev.: On the Concept of Immediate Probabilities, International Journal of Intelligent Systems 10 (1995) 373-397

Multi-Class and Single-Class Classification Approaches to Vehicle Model Recognition from Images

Daniel T. Munroe and Michael G. Madden

Department of Information Technology
National University of Ireland, Galway
Galway, Ireland
{daniel.munroe, michael.madden}@nuigalway.ie

Abstract. This paper investigates the use of machine learning classification techniques applied to the task of recognising the make and model of vehicles. Although a number of vehicle classification systems already exist, most of them seek only to distinguish between vehicle categories, e.g. identifying whether a vehicle is a bus, truck or car. The system presented here demonstrates that a set of features extracted from the frontal view of a vehicle may be used to determine the vehicle type (make and model) with high accuracy. The performance of some standard multi-class classification algorithms is compared for this problem. A one-class k-Nearest Neighbour classification algorithm is also implemented and tested.

1 Introduction

The need for vehicle identification and classification technologies has become relevant in recent years as a result of increased security awareness for access control systems in parking lots, buildings and restricted areas. Vehicle recognition can also play an important role in the fields of road traffic monitoring and management. For example, in the automatic toll collecting systems on roads, vehicles have to be classified into categories in order to calculate the correct amount to charge.

Vehicle type recognition, as a process of identifying the correct make and model from a frontal image of a vehicle (car), represents a natural extension of conventional number-plate recognition systems. Number-plate recognition software could benefit from the system proposed in this paper, by providing a double-check to combat the problem of fake number plates.

The recognition process proposed in this paper is based on using specific feature extraction techniques from digital images. Different machine learning algorithms are tested on the dataset of 180 frontal view images of vehicles (30 images of each of five vehicle classes and 30 other miscellaneous vehicle classes), and experiments are carried out to assess their performance.

Two broad approaches to machine learning classification are considered: multi-class classifiers and single-class classifiers. As discussed below in Section 3, multi-class classification is the ‘standard’ approach used in machine learning, but the single-

class approach is more appropriate in some applications where standard assumptions about the distribution of examples do not apply.

After providing a brief overview of related research and the concept of single-class classification, the system is described in more detail. Then, the performance of various classification algorithms is analysed, and conclusions are drawn.

2 Related Research

Various approaches to vehicle classification and detection have been reported in the computer vision literature. Despite the large amount of literature in vehicle detection, there has been relatively little done in the field of vehicle classification. It is a relatively challenging problem due to the wide variety of vehicle shapes and sizes, making it difficult to categorise vehicles using simple parameters.

Most systems either detect (locate a vehicle against a background) or classify vehicles into broad categories such as cars, buses and trucks [3, 5, 7, 8, 14, 16]. Wei e*t al.* [14] use a 3-D parameterised model which corresponds to features of the vehicle's topological structure, classified using a neural network. They present results showing that 91% of the vehicles are correctly identified into six different categories. Lipton *et al.* [8] describe a vehicle tracking and classification system that can classify moving objects as vehicles or human beings, but its purpose is not to separate vehicles into different classes. Their system obtained over 86% correct classification on vehicles and 83% correct on humans.

Gupte *et al.* [5] present an algorithm for detection and classification of vehicles in image sequences of traffic scenes. The system classifies vehicles into two categories – cars and non-cars (e.g. buses, trucks, SUV's). In a 20-minute sequence of highway traffic, 90% of the vehicles were correctly detected and tracked, and of these correctly tracked vehicles, 70% were correctly classified. Kato *et al.* [7] propose the development of a driver assistance system using a vision-based preceding (vehicles travelling in the same direction as the subject vehicle) vehicle recognition method, which is capable of recognising a wide selection of vehicle types against road environment backgrounds. The classification method they used is the multiclustered modified quadratic discriminant function. The system classifies vehicles into three different categories and has a success rate of 97.7%.

Dubuisson Jolly *et al.* [3] use a deformable template algorithm consisting of finding a template that best characterises the vehicle into one of five categories. Their algorithm was tested on 405 image sequences and had a recognition rate of 91.9%. Similarly, Yoshida *et al.* [16] describe a local-feature based vehicle classification system, which classifies vehicles using a computer graphics model. They use a template matching technique and achieve a 54% accuracy rate, when classifying the images into five categories.

More strongly related work to ours, in terms of what is being achieved, is that of Petrovic *et al.* [11] who demonstrate that a relatively simple set of features extracted from frontal car images can be used to obtain high performance verification and recognition of vehicle types. Recognition is initiated through an algorithm that locates a region of interest (ROI) and using direct or statistical mapping feature extraction

methods, obtains a feature vector, which is classified using a nearest neighbour algorithm. They state that the system is capable of recognition rates of over 93% when tested on over 1000 images containing 77 different classes.

3 Single-Class Classification

All of the systems described in the previous section are based on multi-class classifiers. Multi-class (including two-class) classification is the standard approach used in machine learning, whereby a hypothesis is constructed that discriminates between a fixed set of classes. For example, a classifier may distinguish between images that either show a vehicle or do not, or distinguish between trucks, buses, vans and cars. However, multi-class approaches make two assumptions:

1. Closed set: all possible cases fall onto one of the classes
2. Good distribution: the training set is composed of cases that are statistically representative of each of the classes

While these assumptions do not appear onerous, they may or may not be reasonable in practice. For example, the closed-set assumption is valid when classifying images as having a vehicle present or not present in them, but may not be valid when classifying vehicles into categories (what about tractors, motorbikes and heavy machinery?) Conversely, when classifying vehicles into categories, the distribution assumption may be valid as it is straight-forward to acquire images that are representative of each category, but it might not be valid for the task of distinguishing vehicles from non-vehicles—should the counter-example images show just empty roads, or people, animals, birds, buildings, bicycles, trees and other subjects?

As machine learning researchers and practitioners in recent years have tackled problems where these assumptions are not valid, because for some classes there is either no data, insufficient data or ill-distributed data available, techniques for single-class classification have begun to receive some attention. Essentially, such techniques form a characteristic description of the target class, using this to discriminate it from any other classes (which are considered outlier classes). Clearly, this avoids the closed-set assumption, and also does not require the availability in the training data of statistically representative samples of classes other than the target class.

An key distinguishing feature of single-class classification algorithms is therefore that they only require positive examples in the training set (i.e. examples that represent the target class), as opposed to a statistically representative distribution of all possible classes, such as is needed for multi-class classification. Thus, the two assumptions listed above are not required.

The first algorithms for single-class classification were based on neural networks, such as those of Moya *et al.* [9, 10] and Japowicz *et al.* [6]. More recently, one-class versions of the support vector machine have been proposed, notably by Tax [13] and Scholkopf *et al.* [12]. Tax's approach is to find the smallest volume hypersphere (in feature space) that encloses most of the training data. Scholkopf *et al.* aim to find a binary function that takes the value +1 in a "small" region capturing most of the data,

and –1 elsewhere. They transform the data so that the origin represents outliers, and then find the maximum margin separating hyperplane between the data and the origin. Scholkopf *et al.* note that both methods are equivalent in some circumstances.

In this paper, we use a single-class classification technique based on the k-Nearest Neighbour (kNN) algorithm. The single-class kNN algorithm was chosen primarily because of its conceptual simplicity, although as will be discussed in Section 5.1, in our initial experiments comparing multi-class classification algorithms it was found that the multi-class kNN worked well, so it was felt that the single-class kNN using the same distance metric should also be appropriate for this problem. In this algorithm, a test object is classified as belonging to the target class when its local density is larger or equal to the local density of its nearest neighbour in the training set (target class) [13].

The single-class kNN classifier has a number of parameters that may be adjusted: the number of neighbours can be changed so that the average *k* distances to the first *k* neighbours is calculated; the threshold value of accepting outlier classes may be changed; also, the distance metric may be changed. Figure 1 shows an example of a target class consisting of Ford Focuses. The algorithm for detecting whether or not a test case A (e.g. Volkswagen Golf) is in the target class is shown immediately below.

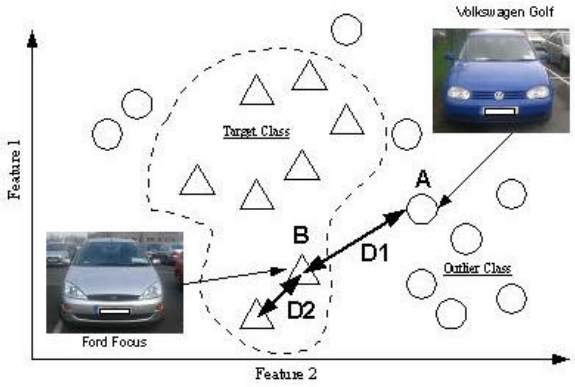


Fig. 1. One-class k-nearest neighbour classifier applied to vehicle recognition dataset

One-class *k*-nearest neighbour classification algorithm

To classify A as a member/not member of target class

1. Set a threshold value T (e.g. 1) and choose the number of *k* distances
2. If $k = 1$

   Find nearest neighbour for A in the target class, call this B and call the distance D1

   Else

   Find the average distances to the *k*-nearest neighbours for A in the target class and call this distance D1. Calculate the min distance and call this B

3. Find nearest neighbour for B in the target class and call this distance D2
4. If D1 / D2 > T
   Reject A as a target class
   Else
   Accept A as a target class

4 Vehicle Type Recognition

For this work, a dataset of frontal images of vehicles was compiled over a period of several weeks, and reflect a range of weather and lighting conditions. The dataset is made up of 180 images of vehicles — 30 images of each of five classes and 30 images of other miscellaneous vehicle types. The classes are: Opel Corsa, Ford Fiesta, Ford Focus, Volkswagen Polo and Volkswagen Golf. The 30 images of different vehicle types are collected particularly for testing and evaluating the one-class classifier. Naturally, care was taken to include only one version of each vehicle make/model, as for example the 1998 Golf would have to be considered as a different class from the 2004 Golf, since these two versions have quite different appearances. All images contain frontal views of a vehicle captured from slightly different distances and from a height of approximately 1 metre. The images have 1600 x 1200 colour pixels. A sample of each class of car is shown in Figure 2.

Fig. 2. Examples of the five different car types as they appear in the dataset

The system is implemented in Matlab using the Image Processing Toolbox. The image is converted to a grayscale image and automatically cropped to exclude the top half. The next step is to detect edges in the image. Edge detection highlights sharp changes in intensity, as differences in intensity can correspond to the boundaries of the features in the image. After experimenting with some alternative algorithms, the Canny edge detection [2] method was chosen because it succeeded in finding all the important features in the image. The Canny edge detector first smoothes the image using a Gaussian filter to eliminate noise before performing the edge detection. Dilation was then used to fill the gaps left in the detected edges of object outlines. Dilation is

an operation that “grows” or “thickens” objects in a binary image and is controlled by a shape referred to as a linear structuring element [4].

After having reduced the image to a series of objects, standard elements of the image such as the lights are identified automatically. A fixed-length numerical feature vector is then derived that includes some basic geometric properties of the extracted features, including the average shape signature of both lights in each vehicle. Shape signatures are 1-D functional representation of the boundaries of objects and are obtained by plotting the distance from an interior point (in this case, the centroid) in the object to the boundary as a function of angle.

Finally, as described in the next section, different machine learning classifiers are used to determine the vehicle make and model associated with each vector. The overall procedure is illustrated in Figure 3.

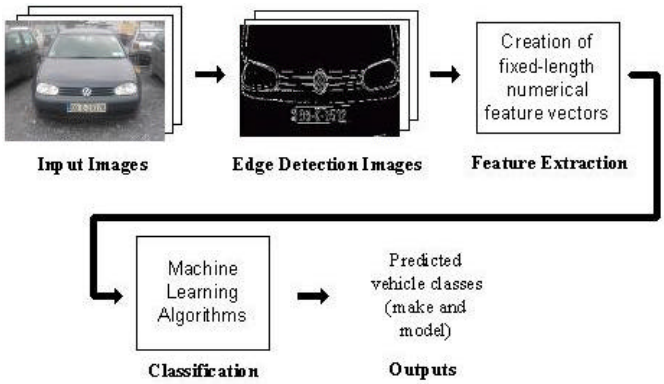


Fig. 3. Overall System for Vehicle Make/Model Identification

5 Experimental Results

Two sets of experiments have been performed. The first set of experiments is described below in Section 5.1. They involve comparing the performance of a range of standard multi-class classifiers on the dataset, since multi-class classifiers have been used in previous approaches to vehicle identification/classification. The dataset used in testing these algorithms consists of 150 samples containing 30 samples of each of the five different vehicle classes; the other 30 samples of miscellaneous vehicles were not used here. Previous approaches have used different forms of feature extraction, so it is interesting to consider how our approach to feature selection works with standard classifiers. The specific classification algorithms chosen are the C4.5 decision tree, the *k*-nearest neighbour classifier and a feed-forward neural network trained using backpropagation. The implementations of these in the WEKA machine learning package [15] were used. The default settings in WEKA for these algorithms were used.

The purpose of the second set of experiments is to evaluate the performance of a single-class classifier for this task. The dataset used in testing this algorithm consists of 180 samples containing 30 samples of each of the five different vehicle classes and

30 examples of unknown vehicle types. The single-class kNN algorithm, described above in Section 3, was implemented in Matlab and its performance evaluated as discussed in Section 5.2.

5.1 Multi-Class Classification Results

Figure 4 compares the learning curves of the three multi-class classification algorithms under consideration. A learning curve gives an indication of the amount of data required to achieve good performance with a classification algorithm. It is constructed by randomly sampling training sets from the overall dataset, at a range of percentages between 5% and 90% of the overall dataset. Each time, a classifier is constructed with the training data set and evaluated on the remainder of the data. This procedure is repeated 10 times for each training set size and the results averaged.

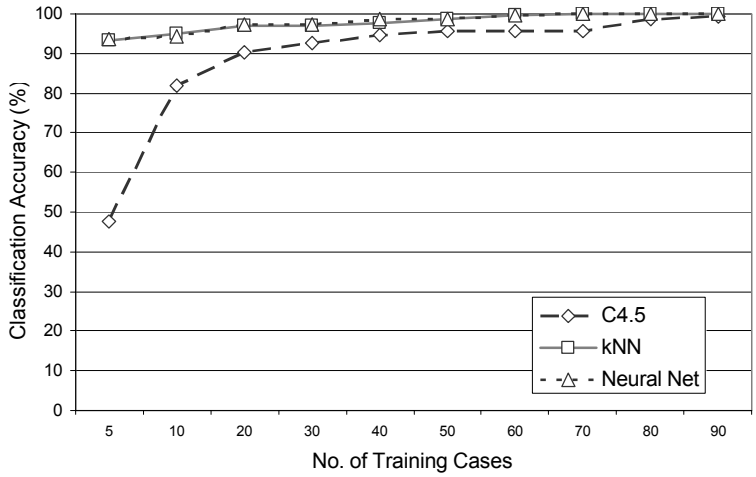


Fig. 4. Comparison of Learning Curves for the Multi-Class Classification Algorithms

The learning curves indicate that classification performances of the *k*-nearest neighbour and neural network algorithms are comparable with each other, and better than that of the decision tree algorithm, at least at lower training set sizes. The curves also show that 70% of the dataset is sufficient for 100% classification accuracy using kNN or the neural network.

The performance of each algorithm was also evaluated using 10 x 10-fold cross-validation with sorted runs [1]. Using this technique, the data is divided randomly into ten parts, each part is held out in turn and the learning scheme trained on the remaining nine-tenths. The procedure is repeated ten times and the average for the ten parts is calculated. The whole process is repeated for ten different runs and the average and standard deviation is calculated. Table 1 lists the accuracy (average ± standard deviation) on the training data of each of the three multi-class classification algorithms, computed using a 10 x 10-fold cross-validation.

Although the results for kNN are numerically higher than those of the other two algorithms, a paired t-test based on the 10 x 10-fold sorted cross-validation runs did not identify the difference as being statistically significant at the 5% level.

Table 1. Results of the 10 x 10-fold cross-validation

Algorithm	Accuracy (%)
C4.5	98.53 ± 3.34
*k*NN	99.99 ± 0.21
Neural Net	99.53 ± 1.47

5.2 One-class Classification Results

10 fold cross-validation is carried out to evaluate the performance of the one-class classifier algorithm. For each fold in this experiment, the target class contains 27 examples of numerical feature vectors representing a certain vehicle (e.g. Opel Corsa) and the test set contains 18 examples of numerical feature vectors of different images of vehicles (3 of the target class, 3 of each of the other 4 class types and 3 unknowns). The 3 unknowns are images of cars not in the dataset (e.g. Toyota Corolla). The process is repeated ten times for each target class and the average is calculated.

Table 2 presents results from separate experiments using one-class classifiers to identify each of the five target classes. The average results of these experiments is also listed. The performance of the kNN one-class classifier algorithm is naturally influenced by the learning parameters, as stated earlier. The k value may be changed so that the average k distances to the first k neighbours are calculated. The best-performing value was found experimentally to be $k = 1$. The threshold value of accepting outlier classes may also be changed; the best value found experimentally is $T = 5$. The one-class classifier correctly identified a high percentage of target and outlier classes, after some experimentation to optimise the learning parameters.

Table 2. Results of the one-class nearest neighbour classifier

Target Class	Accuracy
Opel Corsa	97.77%
Ford Fiesta	97.22%
Ford Focus	97.44%
VW Golf	97.22%
VW Polo	98.88%
Average:	97.70%

6. Conclusions

Vehicle recognition is an important technology for developing systems for road traffic monitoring and management and security issues. However, it is difficult task for computer systems to achieve because vehicles have a wide range of different appearances due to the variety of their shapes and colours.

This paper proposes a novel vehicle recognition process that identifies the vehicle make and model (e.g. Volkswagen Golf) from a frontal image. Extracted fixed-length numerical feature vectors are tested and classified using different machine learning techniques. Of the multi-class classifiers considered, the kNN and the neural network classifiers appear to be most effective for this task, with accuracy of over 99.5%.

A single-class kNN classifier was also evaluated, as single-class classifiers have the benefit of not making assumptions about having a closed set of classes or having a training data set that is fully representative of data that would be encountered in practice. This classifier was also shown to perform well, with an overall accuracy rate of about 97.70%.

Clearly, it is not reasonable to draw direct comparisons between the results of the multi-class and single-class classifiers presented here, as the experimental methodology and assumptions underlying are quite different. In particular, we note that multi-class results could be made arbitrarily bad by adding vehicle types to the test set that do not appear in the training set (since the multi-class classifier output cannot represent the concept 'none of the above'), whereas this should not be detrimental to the performance of the single-class classifier. Other approaches could be used to defend against this problem, for example using two-class classifiers and training them using a one-versus-all classification scheme. However, such an approach would not be theoretically well-motivated, as the negative examples would represent a diverse collection of classes, and would still not be statistically representative of the negative concept.

In the future, we propose to assess the performance of other forms of single-class classifier on this problem domain. We also intend to accumulate a library of vehicle images that do not fall into any of the classes considered here.

References

1. Bouckaert, R.R.: Choosing between two learning algorithms based on calibrated tests. International Conference on Machine Learning (ICML-2003), Washington DC, 2003
2. Canny, J.: A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, (6): 679-698, 1986
3. Dubuisson Jolly, MP., Lakshmanan, S., Jain, A.K.: Vehicle Segmentation and Classification using Deformable Templates. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 3, March 1996
4. Gonzalez, R.C., Woods, R.E., Eddins, S.L.: Digital Image Processing using Matlab. Prentice Hall, 2004

5. Gupte, S., Masoud, O., Martin, R.F.K. Papanikolopoulos, N.P.: Detection and Classification of Vehicles. IEEE Transactions on Intelligent Transportation Systems, Vol. 3, No. 1, March 2002
6. Japkowicz, N., Myers, C., Gluck, M.: A novelty detection approach to classi?cation. Proc. 14th International Joint Conference on Artificial Intelligence, Montreal, Canada, 1995
7. Kato, T., Ninomiya, Y., Masaki, I.: Preceding vehicle recognition based on learning from sample images. IEEE Transactions on Intelligent Transportation Systems, Vol. 3, No. 4, December 2002
8. Lipton, A.J., Fujiyoshi, H., Patil, R.S.: Moving target classification and tracking from real-time video. IEEE Workshop Applications of Computer Vision, 1998, pp. 8-14
9. Moya, M., Koch, M., Hostetler, L.: One-class classifier networks for target recognition applications. In Proceedings World Congress on Neural Networks, Portland, OR, International Neural Network Society, INNS, pp. 797–801, 1993
10. Moya, M.R. & Hush, D.R., 1996: "Network Constraints and Multi-Objective Optimisation for One-Class Classification." Neural Networks, Vol. 9, No. 3.
11. Petrovic, V., Cootes, T.: Analysis of features for rigid structure vehicle type recognition. BMVC2004, 2004
12. Scholkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural Computation, Vol. 13. (Also a Microsoft Research technical report, MSR-TR-99-87, 1999.) 2001
13. Tax, D.M.J.: One-class classification. PhD Thesis, Delft University of Technology. ISBN: 90-75691-05-x, 2001
14. Wei, W., Zhang, Q., Wang, M.: A method of vehicle classification using models and neural networks. IEEE Vehicular Technology Conference, IEEE, 2001
15. Weka Homepage: The Waikato Environment for Knowledge Analysis. Department of Computer Science, University of Waikato, NZ, http://www.cs.waikato.ac.nz/ml.
16. Yoshida, T., Mohottala, S., Kagesawa, M., Ikeuchi, K.: Vehicle Classification Systems with Local-Feature Based Algorithm using CG Model Images. IEICE Trans., Vol. E00-A, No. 12, December 2002

Natural Language Processing

The Role of Experience in the Interpretation of Noun-Noun Combinations

Phil Maguire and Arthur Cater

Department of Computer Science, University College Dublin
Belfield, Dublin 4, Ireland.
phil.maguire@ucd.ie, arthur.cater@ucd.ie

Abstract. Previous studies [2], [6] have shown that combinations whose modifiers are typically associated with the instantiated relation are interpreted reliably faster than those whose modifiers are less frequently associated with the instantiated relation. Gagné and Shoben attributed this effect to the influence exerted by prior experience of the modifier. They proposed that speakers maintain relation type distributions regarding the frequency with which modifier nouns have been used with each of 16 possible relation types. However, others [4], [5], [7] have claimed that the differences in response times observed in Gagné and Shoben's study may have arisen due to the nature of the concepts being combined. We contrasted these views by investigating whether these differences persisted when the constituent concepts of a combination were presented individually. We presented noun pairs without a modifier-head syntax; as a result interpretation could not be influenced by previous experience of how those nouns had been used as modifiers in the past. The results revealed that differences in response time remained using either method of presentation. This finding is problematic for the CARIN theory and, as a result, we consider other factors that might influence the difficulty of linking two arbitrary concepts.

1 Introduction

In everyday conversation, noun-noun compounds (also termed conceptual combinations) are frequently used in order to express new ideas and to encapsulate novel objects and experiences (e.g. beer headache, peasant dance). Compounding is a useful practice in that it greatly enhances the flexibility of language as well as increasing efficiency in communication. Although people have a well developed means of understanding these novel compounds, the comprehension process is often not trivial, requiring an understanding of the speaker's communicative goals and a representation of the situation at hand as well as detailed world knowledge. Accordingly, the study of conceptual combination is important because of the way in which it is intimately associated with the generativity and comprehension of natural language. In English, where compounding is particularly productive, combinations consist of a modifier followed by a head noun. Usually, the head noun denotes the main category while the modifier implies a relevant subcategory or a modification of this set's typical members. In this way, a *mountain flower* is interpreted as a type of flower, and more particularly as one which is located in mountains. In order to understand a combination like this, people

first have to deduce the most likely way in which *mountain* could be intended as a modification of the concept *flower*. Once a link between the two concepts has been definitively established, the combined concept can then be elaborated upon and further inferences can be made (e.g. *mountain flowers* are likely to be wild).

Gagné and Shoben demonstrated in [2] that the more frequently a relation is associated with the modifier noun of a combination, the easier it is to judge whether a combination involving that relation is sensical or not. The effect itself seems relatively intuitive. Consider the modifier *chocolate*: combinations in the form *chocolate X* can be most easily interpreted as *X* <*made of*> *chocolate* seeing as the most common instantiation of chocolate as a modifier involves the compositional <H *made of* M> relationship. In this way, even the combination *chocolate train* is more likely to be interpreted as a train made of chocolate than as a train containing chocolate, despite the latter perhaps being the more plausible. It also seems reasonable that combinations in which modifiers are used in an atypical fashion might prove more difficult to interpret: *chocolate magazine* prompts a momentary hesitation during which the most obvious instantiation of *chocolate* must be suppressed. Based on these principles, Gagné and Shoben attributed differences in response times observed during a sensicality judgement task to prior experience of the modifier [2]. Their Competition Among Relations in Nominals (CARIN) theory proposes that speakers maintain distributions recording how frequently (in relative terms) nouns have combined as modifiers using each of 16 possible relation types outlined therein.

Despite the apparent reasonableness of this theory, others have questioned whether prior experience is the central contributing factor towards the observed phenomenon [4], [5], [7]. Murphy argues in [5] that the storage of arbitrary relations for each noun, as proposed by CARIN, seems unconvincing mainly because the 16 relation types provided are ambiguous and nondescriptive. In effect, relations are extremely varied and detailed and as a result, classifying a relation under the CARIN taxonomy abstracts it to such a level that such information becomes useless. The focus of our study is thus on contrasting the CARIN theory with the alternative view that differences in ease of interpretation arise naturally and are not caused by prior experience of the modifier. The following experiment accomplishes this by examining the process of relation selection between noun concepts in the absence of a modifier-head syntax.

1.1 Overview of Rationale

The most unambiguous method by which to address the role of prior experience in conceptual combination would be to investigate cases in which no such experience is available. If it were practical, one might recruit speakers of a combination-free dialect as participants: speakers encountering combinations for the first time would be required to interpret them in the absence of any statistical knowledge, thereby providing an ideal condition for investigating CARIN's premises. Unfortunately, given the ubiquity of conceptual combination, virtually every language contains combinations of some form, thereby rendering this paradigm infeasible. Another approach would be to present the concepts as images, thus avoiding the activation of linguistic knowledge. Unfortunately, the use of images is unreliable and picture recognition is greatly influenced by canonicality and view specificity. Furthermore, some concepts cannot be represented pictorially in their prototypical form: a representation of fruit would in-

variably refer to one particular type of fruit. Similarly, conveying abstract concepts like *dilemma* or *justice* would be even more problematic.

In order to sidestep these problems while simultaneously suppressing the activation of statistical knowledge, an alternative method of presentation was adopted. This paradigm was based on the assumption that a noun concept cannot be interpreted as a modifier in the absence of a head on which it can act. Furthermore, statistical knowledge regarding relation type preference is irrelevant when the noun concept is not acting as a modifier. As a result, CARIN-style statistical knowledge can have no influence on the comprehension of an isolated noun concept. For example, in reading the noun *chocolate* on its own, only the material noun interpretation is activated. Its alternative interpretations such as <*made of chocolate*> or <*containing chocolate*> appear only in combination with an appropriate head. As a result, the distribution of *chocolate*'s relation type frequencies cannot influence the comprehension process in this case.

We therefore decided to present noun concept pairs as individual nouns. This ensured that the nouns would not be interpreted as constituents of a combination, and hence that statistical knowledge regarding the modifier's relation type preference would not be activated. By controlling the influence of past experience in this way, we were able to investigate the nature of relation selection when that process is based solely on the conceptual content of the constituent nouns.

2 Experiment

The experiment was designed to eliminate any influence of modifier history. Presenting nouns outside a combinational syntax obliged participants to relate the noun pairs by virtue of their semantic representation only. According to the CARIN theory, differences in response time should only be apparent in cases where statistical knowledge about modifier relation type frequencies is activated. As a result we expected that the differences in ease of interpretation found by Gagné and Shoben in [2] would disappear given this novel method of presentation. By investigating whether differences in ease of interpretation are mitigated when noun concept pairs are interpreted independently, we hoped to corroborate the claim that such differences arise due to statistical knowledge.

Two separate conditions were identified, both involving the same experimental materials. In the combined concept (CC) condition, participants were presented with noun pairs in the typical combinational format and were required to judge the sensicality of these combinations. In the independent concept (IC) condition, a different set of participants was presented with the same nouns in isolation and these participants were asked to decide whether a sensible linking relation could be found. According to CARIN, the interpretation of a novel noun-noun combination occurs when speakers identify a relation between the modifier and the head noun. Thus, as participants in the IC condition were required to search for a sensible relation linking the nouns, the participants in both conditions were effectively carrying out the same ultimate task, albeit in different ways. In the CC condition, participants could exploit statistical knowledge as CARIN supposes; in the IC condition, they could not.

2.1 Method

Participants Forty first-year undergraduate students from University College Dublin participated in the study for partial course credit. All were native English speakers.

Materials For the sake of comparison, we selected a subset of the materials used by Gagné and Shoben in [2] for our experiment. In their study, materials were divided into three categories, namely HH, HL and LH conditions. For these condition labels, the letters H and L refer to the frequency of the instantiated relation (High or Low), with the first letter denoting how frequently that relation is associated with the modifier and the second denoting how frequently it is associated with the head. Thus *mountain magazine* (a magazine <*about*> mountains) is considered to be an LH combination since the <*about*> relation is low-frequency for the modifier *mountain* but high-frequency for the head *magazine*.

We selected a sample of 10 materials from each of the HH, HL and LH categories for the purposes of the experiment. These materials were selected based on several criteria. Any of Gagné and Shoben's modifiers that were adjectives (e.g. *thermal*, *historical*) were excluded from consideration. We also ignored combinations in which the modifying noun was actually intended in a plural sense despite being denoted as singular (e.g. *servant language* is a language used by servants in general and not one servant in particular). In a re-analysis of the original CARIN study, Wisniewski and Murphy [7] suggested that the plausibility and familiarity of Gagné and Shoben's materials may not have been properly controlled, thus confounding response times for the various conditions. Indeed, many of Gagné and Shoben's materials are quite bizarre (e.g. *olive area*, *cooking hole*). In order to account for this inconsistency, two independent judges were asked to decide which of the eligible materials from the appendices of [2] were truly sensical and which ones were not. Any of the combinations judged as non-sensical by either of the judges was excluded. Following this, 10 materials were randomly selected from the remaining selection in each category.

Several measures were taken to ensure that noun pairs presented in the IC condition would not be interpreted as a combination. Firstly, the presentation of each noun in a pair was separated by a 1,000ms visual and phonological mask consisting of a blank screen containing the "+" symbol. This was intended to prevent participants' reading of the nouns coalescing into a combinational syntax within their phonological loop. As a further measure, we took advantage of the fact that modifiers in the English language are nearly always singular: having a plural in the modifier slot violates the (weak) constraints for combinational syntax. Thus when two nouns are presented together and the first is plural, the phrase can typically only be interpreted as two separate nouns and not as a combination (e.g., the two words *mountains* + *cloud* cannot be read as a single unit). We therefore pluralized a large portion of the filler modifiers in the IC condition (e.g. *dogs* + *vet*, *sandwiches* + *filling*). The presence of these pluralized fillers was intended to prevent participants from adopting the habit of viewing the two nouns as a combination, which might have supported their search for a linking relation.

The experimental materials used in both the CC and the IC conditions were identical. The method of presentation was different only in that the mask screen for the IC group contained the "+" symbol, whereas for the CC group it was completely blank. Aside from the instructions, the only other difference between the two conditions was the plurality of the filler modifiers. In the IC condition, we used 30 sensical combina-

tions with pluralized modifying nouns, 10 nonsensical combinations with singular modifying nouns and 10 nonsensical combinations with pluralized modifying nouns. In the CC condition, all of the corresponding fillers had singular modifiers.

Design A mixed 2 X 3 factorial design was used, with the two conditions of presentation as the between-participants measure (IC and CC), and Gagné and Shoben's [2] three conditions of relation type frequency as the within-participants measure (HH, HL and LH). The dependent measures were response time and accuracy rate, which were used to infer ease of interpretation. Participants were randomly assigned to the between-participants conditions, with a total of twenty in each. Each participant was presented with 80 stimuli, comprising the same set of 30 experimental stimuli for both between-subject conditions and the 50 filler items.

Procedure Participants sat in front of a computer screen and placed the index finger of their left hand on the F key of the computer keyboard and the index finger of their right hand on the J key. The participants in the CC condition were instructed that they would be presented with a series of concept combinations for which they had to make sensicality judgements and that the constituent nouns of the combinations would be displayed one after the other. In contrast, participants in the IC condition were instructed that they would be shown two concepts, and that they had to decide whether one concept could be combined with the other in a sensible manner. Both sets of participants were instructed to press J for sense and F for nonsense and emphasis was placed on the fact that they should only press F if the item was truly incomprehensible. In both conditions, each word was displayed by itself for one second in the centre of the screen, separated by the mask screen lasting for one second. After viewing the second word in the trial, participants had to make a sensicality judgement by pressing the appropriate key. The same materials were presented in both conditions and so the modifying noun was always presented first, although this was not made explicit.

Participants were initially given a short practice session where feedback was given regarding their judgements. The aim of this practice was to set a reliable threshold for sensicality and also to familiarize them with the nature of the task. Participants in the IC condition were shown pairs of concepts with pluralized modifying nouns (e.g. *tomatoes + sandwich*, *dogs + allergy*). After making a sensicality judgement participants were then informed whether the two nouns could be related and if so, the nature of the relation (e.g. "an allergy caused by dogs"). Similarly, in the CC condition, participants were shown concept pairs in a combinational format and after making a sensicality judgement, were shown how the combination was sensical or otherwise. Upon completing this practice session, participants were informed that they were beginning the main part of the experiment. The materials were then presented in a random order to each participant.

2.2 Results and Discussion

A total of 20.5% of trials were omitted from the analysis of the results, 15% in the CC condition and 26% in the IC condition. In 11.9% of CC trials and 23% of IC trials the incorrect response was given and hence these data were excluded. Responses were also eliminated if they were deemed unreasonably fast (0.1% of trials < 400 ms in the CC

condition) or slow (1.7% > 4000 ms in the CC condition, 2.5% > 8000ms in the IC condition). After this initial elimination process, any remaining response times which were more than three standard deviations outside each participant's mean were also excluded. This removed another 1.3% of trials in the CC condition and 0.5% in the IC condition. In the CC condition, the mean response time was 1,093; 1145 and 1,254 ms for the HH, HL and LH conditions respectively while in the IC condition, the mean response time was 1,853; 1,873 and 2,213 ms. Response time was analysed using a 2 X 3 repeated measures ANOVA, with participants as a random variable, presentation method as a between-participants variable and relation type frequency as a within-participants variable. The ANOVA revealed a main effect of relation frequency, $F(2, 76) = 7.89$, $p < .01$. There was also a significant main effect of presentation method: response times were significantly longer in the IC condition than in the CC condition, $F(1, 38) = 15.81$, $p < .01$. Contrary to our hypothesis, there was no significant interaction between method of presentation and relation type frequency, indicating that the influence of relation frequency was not affected by method of presentation, $F(2, 76) = 1.89$, $p = .16$. The mean accuracy rates for the HH, HL and LH in CC condition were .93, .87 and .76 respectively while in the IC condition the mean accuracy rates were .83, .76 and .64. A second repeated-measures ANOVA revealed a main effect of relation frequency, $F(2, 76) = 26.21$, $p < .01$, and of presentation method, $F(1, 38) = 21.17$, $p < .01$. Once again, there was no significant interaction between accuracy and method of presentation, $F(2, 76) = .20$, $p = .82$, indicating that the pattern of accuracy rates was similar for both methods of presentation.

These results show significant differences in response time and accuracy rates between the high and low frequency modifier conditions for both the CC and IC conditions. The absence of an interaction between method of presentation and modifier relation frequency contradicts our hypothesis that eliminating the influence of statistical knowledge would mitigate differences in response time. It also suggests that the factors influencing response time and accuracy rates are not related to past experience of the modifier. Given that relation selection in the IC condition could only be carried out based on the semantic representation of the constituent concepts, it appears that the variations in response time, equally evident in both conditions, can only be accounted for by naturally arising differences in the ease of combining an arbitrary pair of noun concepts.

2.3 Correlation analyses

In order to determine the contribution of modifier influence towards the overall variance in response time, we obtained a correlation between response time and relation type frequency. Each of the materials was assigned a value corresponding to the relative frequency with which the modifier was associated with the instantiated relation. We used the same frequencies as Gagné and Shoben [2], which they derived by pairing 91 heads with 91 modifiers. After rejecting uninterpretable combinations, they analysed the relative frequencies with which the modifiers combined using each of the possible relation types. For example, *mountain* is typically interpreted using the <*located*> relation (e.g. *mountain cloud*) and according to Gagné and Shoben's frequencies, the relative frequency of this relation for *mountain* is .82. Using these values, we obtained correlations between response time and relation frequency and between accu-

racy rate and relation frequency. Neither of the correlations was significant for either the CC (r = -.11, p = .58, r = .31, p = .09) or the IC conditions (r = -.24, p = .21, r = .28, p = .14), challenging the notion that the relation type frequency of the modifier is an important factor in the interpretation process.

We also examined how well plausibility and familiarity predicted response time relative to modifier relation frequency. For these correlations we used the ratings provided by a group of 30 participants in Wisniewski and Murphy's re-analysis [7] of Gagné and Shoben's study [2]. The correlations between mean response time and familiarity and plausibility were highly significant for the CC (r = -.61, r = .56 respectively) and the IC conditions (r = -.57, r = -.58). The correlations between accuracy rate and familiarity and plausibility were also highly significant for both the CC (r = .69, r = .66) and the IC conditions (r = .72, r = .73).

The fact that there was no significant correlation between modifier relation frequency and the dependent variables is a strong indicator that past experience of the modifier does not have a large influence on ease of interpretation. On the other hand, the high correlations involving plausibility and familiarity suggest that these factors account for a far greater portion of the variance in response time: stimuli referring to a more plausible and familiar concept were interpreted reliably faster and more accurately regardless of being presented as a combination or otherwise. These findings are consistent with those of Wisniewski and Murphy's in [7], which revealed that familiarity and plausibility are the strongest predictors of response time. Because these variables relate to the combined concept itself, they are thus independent of the method of presentation and this might explain how comparable differences were observed in both the CC and IC conditions.

In a further analysis, we investigated how well the ease of finding a relation between two separate noun concepts could predict the ease of interpreting the same concepts presented as a combination. In order to do this we correlated the two sets of dependent variable values from the IC and CC conditions, as both contained the same stimuli. The correlation between accuracy rates in the IC and CC conditions was highly significant, r = .78. The correlation between response times in both conditions was also significant, r = .44. Ignoring two pairs of stimuli for which the correct response was elicited less than 25% of the time in the IC condition, the correlation between both sets of response times increased to .52, which is considerable given that mean response times are quite variable. Placing it in perspective, the corresponding correlation between response times in our CC condition and those in Gagné and Shoben's study [2] was insignificant, p = .11, r = .30. These correlations indicate that noun pairs that were easy to interpret as a combination were also easy to relate when presented as individual noun concepts. Similarly, combinations that were often misjudged as nonsense when presented as a combination were also frequently misjudged when presented as two separate concepts. The high correlation between response times in both conditions suggests that the search for a linking relation accounts for a significant portion of the variance in the ease of interpreting a combination and that relation identification forms a fundamental part of the comprehension process.

At first blush, the strong correlation between familiarity and the dependent variables would seem to suggest that the frequency with which speakers are exposed to a certain combination directly influences how difficult that combination is to interpret. Certainly, combinations that are encountered very frequently can be stored as single en-

tries in the lexicon, thereby obviating the combination process. However, a pair of one-tailed z-tests revealed no significant differences between the familiarity correlation coefficients in the CC condition and those in the IC condition (z = -.23, p = .41, z = -.22, p = .41 for response time and accuracy correlations respectively). As materials in the IC condition were not presented as a combination, this suggests that familiarity of the combinational phrases per se was not responsible for the high correlations. As a result, it seems unlikely that participants were recalling previous encounters with the stimuli, which indeed were relatively novel.

Given this finding, the influence of familiarity must be due to factors other than memory retrieval. One possible explanation is that the familiarity ratings reflect the familiarity of the *referent concept* and not of the phrase itself: such familiarity would influence the ease of interpretation of a compound noun phrase whether it was presented as a combination or not. Another possible explanation is that familiarity covaries with plausibility: combinations which refer to more plausible concepts will happen to be encountered more frequently and will thus be rated as more familiar. This possibility is supported by a surprisingly high correlation of .94 between the plausibility and familiarity ratings for our materials. As a result, even though familiarity might appear to influence response time, this variable might only be a reflection of the referent's plausibility, this being the fundamental factor affecting the ease of interpretation.

2.4 Method of Interpretation

Given our results, one might propose that despite our efforts, participants in the IC condition were somehow interpreting the modifying nouns in a modifier sense. Several findings cast doubt on this possibility. Firstly, a main effect of presentation condition was observed and response times in the IC condition were significantly longer than those in the CC condition. Moreover, accuracy rates were significantly lower. If participants had been processing the word pairs as combinations then overall differences in response times or accuracy rates would not have been expected. One might argue that statistical knowledge about how a noun concept can be used as a modifier is still activated even when that noun is not being interpreted in a modifying role. In order to investigate this possibility we ran an analysis of the differences between response times to stimuli in the high modifier frequency conditions (HH and HL) and the sensical pluralized fillers. According to CARIN, combinations in the HH and HL conditions should benefit from statistical knowledge whereas pluralized modifiers should have no history because plural nouns like *tomatoes* are almost never used as modifiers. Despite this, we discovered that participants interpreted the pluralized modifiers significantly faster than the high modifier frequency stimuli, $t(19) = 2.89$, $p < .01$. This is the opposite to what would have been expected had participants been benefiting from the availability of statistical knowledge. It therefore suggests that the IC materials were evaluated based solely on the semantic representation of the individual noun concepts.

Despite the fact that the method of presentation did not affect the relative differences in response times between the relation type frequency conditions, it is worth noting that materials in the IC condition were interpreted reliably slower than those in the CC condition (an average of about 1800ms as opposed to 1200ms). This disparity indicates that the presence of a combinational syntax greatly enhances the fluency of interpretation. Although the effect may have been partially due to the unnaturalness of the task,

the longer response times in the IC condition suggest that the syntactical constraints imposed by having a designated modifier and a designated head are important for facilitating relation selection: knowing which noun is acting as a modifier greatly speeds up the interpretation process. Without the clue afforded by syntax, participants in the IC condition may have felt the need to select among a considerably greater number of possible relationships, in some of which the first of the two concepts filled a head slot and the second filled the modifier slot. Furthermore, the presence of a designated modifier may have streamlined the interpretation process as the head concept is typically evaluated in light of the modifier, thereby obviating the full activation of both concepts. In this way, the most relevant features regarding modification can be quickly identified while redundant information can be avoided, a process which may not have been possible in the IC condition. Considering as an example the stimulus *gas lamp*, participants in the CC condition evaluated the concept *lamp* in the context of it being related to *gas*, thus arriving directly at the referent concept. On the other hand, participants in the IC condition may have activated both concepts more fully before searching for an appropriate linking relation, thereby triggering a representation of the prototypical electrical lamp and then being forced to reconsider.

2.5 Implications for the CARIN theory

Various probabalistic models of human language comprehension have been proposed in the past, based on the idea that probabilistic information about words, phrases and other linguistic structure is represented in the minds of language users and plays a role in language comprehension. Indeed, experiments related to general statistical language models show that humans are in fact very good predictors of word usage (see [3]). This would seem to suggest that prior experience as well as something akin to frequency distributions could indeed be a factor in human language processing. Although the current findings challenge the CARIN theory, they do not deny such a possibility. In certain circumstances, speakers may well be aware of the more typical usages of a modifier and this might affect how a combination is likely to be interpreted and hence the level of context that is supplied. However, as an overall theory of conceptual combination, CARIN suffers from several limitations. The relation types suggested by the theory are somewhat arbitrary and ambiguous; many relations cannot be satisfactorily classified under CARIN's taxonomy while others can be placed into several categories. In addition, these relation categories sacrifice much of the detail and natural variation present, therefore rendering such labels uninformative (see [1]). However, CARIN's greatest inadequacy is that it fails to offer any explanation as to how the correct relation is eventually selected. Relation type frequencies alone can never suffice, as the interpretation process unavoidably requires the detailed consideration of both constituent concepts. For these reasons, it seems reasonable to accept that differences in ease of interpretation are predominantly dependent on the properties of the concepts being combined and indeed, this is the most reasonable explanation for our results.

3 Conclusion

Experiment has shown that the differences in response time observed by Gagné and Shoben in [2] are not eliminated by presenting stimuli without a modifier-head syntax. Because the experiment was designed so that participants would seek relations between concepts rather than words, any knowledge about the properties of words was rendered useless. As a result, our findings fail to support a central tenet of the CARIN theory, namely that ease of interpretation is significantly influenced by the combinational history of the modifier word. Correlations between modifier relation frequency and the dependent variables were not reliable. Conversely, correlations between plausibility and familiarity and the dependent variables proved highly significant, suggesting that these are far better predictors of ease of interpretation. The fact that the presentation of a combination as two individual noun concepts had no reliable influence on the relative differences in response times suggests that any factors that do influence interpretation are not linked to the combinational format itself but rather to the properties of the constituent nouns being combined. As a result, greater emphasis should be placed on understanding how the properties of multiple concepts are reconciled rather than focusing on how those words have been used in combinations in the past.

Importantly, this study has demonstrated that concept pairs can be related without a modifier-head syntax. The results revealed that the time taken to relate these concepts in isolation was a strong predictor of the time taken to interpret the corresponding combination. This highlights the important role of relation selection in the interpretation process and indicates that much of the variance in ease of interpretation can be accounted for by the complexity of identifying an appropriate relation. For this reason, future research should seek to explain both the process by which a linking relation is determined and the way in which a combinational syntax facilitates this process.

References

1. Devereux, B. & Costello, F.: Assessing the taxonomic approach to noun-noun compounds. Proceedings of the European Cognitive Science Conference (2003)
2. Gagné, C. L. & Shoben, E. J.: Influence of thematic relations on the comprehension of modifier-noun combinations. Journal of Experimental Psychology: Learning, Memory and Cognition, 23 (1997) 71-87
3. Jurafsky, D.: Probabilistic Modeling in Psycholinguistics: Linguistic Comprehension and Production. In Bod, R., Hay, J. & Jannedy, S. (eds.) Probabilistic Linguistics. Cambridge, MA: MIT Press (2003)
4. Maguire, P. & Cater, A.: Interpreting noun-noun compounds with rare modifiers. Proceedings of the Seventh Conference of the German Cognitive Science Society (2005)
5. Murphy, G. L.: The Big Book of Concepts. Cambridge, MA: MIT Press (2002) 463-464
6. Storms, G. & Wisniewski, E. J.: Does the order of head noun and modifier explain response times in conceptual combination? Memory and Cognition (in press)
7. Wisniewski, E. J., & Murphy, G. L.: Frequency of relation type as a determinant of conceptual combination: a reanalysis. Journal of Experimental Psychology: Learning, Memory and Cognition, 31 (2005) 169-74

Identifying Semantic Equivalence for Multi–Document Summarisation

Eamonn Newman, Nicola Stokes, John Dunnion, Joe Carthy

Intelligent Information Retrieval Group (IIRG), Department of Computer Science
University College Dublin
Ireland
{Eamonn.Newman, Nicola.Stokes, John.Dunnion, Joe.Carthy}@ucd.ie

Abstract. We describe Semantic Equivalence and Textual Entailment Recognition, and outline a system which uses a number of lexical, syntactic and semantic features to classify pairs of sentences as "semantically equivalent". We describe an experiment to show how syntactic and semantic features improve the performance of an earlier system, which used only lexical features. We also outline some areas for future work.

1 Recognising Semantic Equivalence and Textual Entailment

Our current research focuses on the problem of Recognising Semantic Equivalence and Textual Entailment (RSETE). Two sentences are semantically equivalent if they attempt to convey the same information, or if one is entailed by the other, i.e., the information conveyed by one is covered by the information conveyed by the other.

An RSETE system could be applied in a number of Natural Language Processing applications:

Summarisation By identifying those sentences which contain the same information, we can eliminate redundancies. This means that the summary generator will reduce the amount of repetition in the resultant summary.

Question–Answering By reformulating the question as a statement, we can find sentences which match the sentence in the corpus, using RTESE.

Machine–Translation (Evaluation) RTESE could be used to improve the automated evaluation of MT texts. To date, these are generally based on n-gram models. RTESE would allow for translations which have the same meaning but use different words and composition to be recognised.

Query Expansion Having identified semantically similar concepts, we can use them to add new terms to queries in IR systems to improve retrieval precision and recall.

Information Extraction By locating texts with similar semantic content, we can extract information with less reliance on pattern recognition.

It can be seen that RSETE is a very interesting problem area, and any solution in the area will have a wide range of applications.

In Section 2, we provide a thorough description of our system, and the features used to detect Textual Entailment. Section 3 outlines the corpora we used for training and testing our classifiers. We also describe the nature of the experiments conducted. The results of these experiments are presented in Section 3.4. Finally, in Section 4, we give some examples of areas which warrant further research and development.

2 System Description

Our system uses a decision tree classifier whose features include lexical, semantic and grammatical attributes of nouns, verbs and adjectives to identify an entailment relationship between pairs of sentences (comprising of a *text* and an *hypothesis*). We generated our classifier from training sets using the C5.0 machine learning algorithm[10].

The features used are calculated using the WordNet taxonomy [8], the VerbOcean semantic network [1] (developed at ISI) and a Latent Semantic Indexing [4] technique. Other features are based on the ROUGE(Recall–Oriented Understudy for Gisting Evaluation) [6] n-gram overlap metrics and cosine similarity between the text and hypothesis.

Our most sophisticated linguistic feature finds the longest common subsequence in the sentence–pair, and then detects contradictions in the pair by examining verb semantics for the presence of synonymy, near-synonymy, negation or antonymy in the subsequence.

In addition to these measures, there is also a **task** feature which identifies the task definition from which the sentence pair was derived. This allows the system to build separate classifiers for each task which we hoped would capture the different aspects of entailment specific to each task.

We investigated the usefulness of a number of distinct features during the development of our decision tree approach to textual entailment. These features were developed using the training part of the corpus made available for the PASCAL Recognising Textual Entailment Workshop [2][1]. We describe the corpus in Section 3. Not all of these features were contributing factors in our final classification systems, but we list all of them here for the sake of completeness because some features are combinations of other atomic features. Table 1 gives a list of the features we used, and their C5.0 data types.

[1] The corpus may be downloaded from:
`http://www.pascal-network.org/Challenges/RTE/Datasets`

entails	boolean, unknown
<rouge>	continuous
<wordnet>	continuous
LSI	continuous
cosine	continuous
<verbOcean>	continuous
negation_t	continuous
negation_h	continuous
negdiff	continuous
<lcs>	boolean
lcs+not	boolean

Table 1. Features used by decision–tree classifier. <*name*>indicates a tuple of related features.

2.1 Sentence–pair Features

The first of our equivalence features are derived using the **ROUGE metrics**, which were used as a means of evaluating summary quality against a set of human–generated summaries in the 2004 Document Understanding Conference workshop [7]. The metrics provide a measure of word overlap (i.e. unigram, bigram, trigram and 4-gram), and a weighted and unweighted longest common subsequence measure.

The next semantic equivalence feature is calculated using the **cosine similarity measure**, which calculates the distance (or cosine of the angle) between the text/hypothesis pair in an n-dimensional vector space.

Using a **Latent Semantic Indexing** matrix constructed using the DUC 2004 corpus, we attempted to identify words in entailment pairs which have high cooccurrence statistics. This is an enhancement of the similarity measure given by the WordNet features, as it matches not only synonymy in the plaintext, but also uses data from other corpora to identify other latent relationships.

WordNet was used to identify entailment between sentence pairs where corresponding synonyms are used. Words from the same synset (set of one or more synonyms, as defined by WordNet) were considered to indicate a greater likelihood of entailment. We believe that the accuracy of this feature could be greatly improved by disambiguating the sentence pair before calculating synset overlap. More specifically, in some instances multiple senses of a single term could be matched with terms in the corresponding entailment pair, resulting in sentences appearing more semantically similar than they actually are.

VerbOcean is a lexical resource that provides fine–grained semantic relationships between verbs. These related verb–pairs were gleaned from the web using lexico–syntactic patterns that captured 5 distinct verb relationships:

– similar–to (e.g., *escape*, *flee*)

– strength (e.g., *kill* is stronger than *wound*)
– antonymy (e.g., *win*, *lose*)
– enablement (e.g., *fight*, *win*)
– happens–before (*marry* happens before *divorce*)

VerbOcean also lists relationship strengths between verb pairs. In our experiments we only use the antonym and similar–to relationships for verb semantics analysis.

We also identify **adverbial negation** in the sentences. Adverbial negation occurs where the presence of a word (e.g., "nor", "not") modifies the meaning of the verb in the sentence. We generate three features from this information:

– **negation_t** counts the number of occurences of adverbial negation in the text
– **negation_h** counts the number of occurences of adverbial negation in the hypothesis.
– **negdiff** is the difference between negation_t and negation_h.

Examination of the development set suggested that for a significant proportion of sentence pairs, the **longest common subsequence** [2] is largely similar to the hypothesis element, i.e. most of the hypothesis is contained in the text element. For this feature, we only examined verb semantics in the longest common subsequence of the two sentences rather than in the full sentences. An example is shown in Figure 1. There are three variations of this feature: lcs, lcs_pos and lcs_neg.

– **lcs** This feature holds one of three values $\{-1, 0, 1\}$, which correspond to the presence of an antonym, no relationship, or a synonym relationship between the longest common subsequence of the text and the hypothesis sentence respectively.
– **lcs_pos** and **lcs_neg** are simpler features which indicate the presence of a synonym relationship, or antonym relationship respectively.

lcs+not is another feature based on the longest common subsequence. It combines the above lcs features and also looks for the presence of words like "not", which reverse the meaning of the sentence. Thus, for example, if an antonym and "not" occur in a sentence then this is considered to be a positive indication of entailment. Even though lcs+not is a combination of our lcs features we still retain the simpler features as it has been shown that they improve entailment accuracy.

[2] The Longest Common Subsequence of a sentence pair is the longest (not necessarily contiguous) sequence of words which is common to both text and hypothesis.

id=1954; task=PP; judgement=FALSE
Text: *France on Saturday* flew *a planeload of United Nations aid into eastern Chad* where French soldiers prepared to deploy from their base in Abeche towards the border with Sudan's Darfur region.
Hypothesis:*France on Saturday* crashed *a planeload of United Nations aid into eastern Chad*

Fig. 1. Longest Common Subsequence. Italics denote the longest common subsequence.

3 Evaluation

3.1 Corpora

We use three corpora for our experiments. The **RTE** corpus was developed for the Pascal RTE Challenge [2] (See Figs 1–5 for examples of the sentence–pairs in the corpus). The corpus consists of three parts: two development sets which were released for training purposes during the system development stage, and one large test set used for evaluation of the participating systems. In each set positive and negative examples were divided into a number of different NLP tasks where textual entailment is used. These tasks include:

- CD: Comparable Documents
- IE: Information Extraction
- IR: Information Retrieval
- MT: Machine Translation
- QA: Question Answering
- RC: Reading Comprehension
- PP: Paraphrasing

The two development sets contained 287 and 280 sentence–pairs, respectively. The evaluation set contained 800 sentence–pairs. Each sentence–pair consists of a text and a hypothesis. The sentences come from datasets and corpora pertaining to the different NLP tasks. Some examples are given in Section 4

The second corpus we use was developed by Microsoft Research [3]. It consists of training and test sets which contain 4076 and 1725 pairs, respectively. The sentences were taken from online news articles which had been clustered by topic.

Manual investigation of this corpus showed a bias of approximately 2:1 in favour of positive classification. We have found that any bias in the training data tends to be reflected in the classifications produced by our system, e.g. the more negative examples in the training data, the more likely it is that the decision tree will return a negative classification. To investigate this further, we built a third corpus by removing positive instances from the MSR corpus until there were an equal number of positive and negative instances. Our modified corpus (MSR-5050) contains 2646 pairs.

3.2 Evaluation Metrics

We use a simple measure of accuracy to rate system performance. We examine the output classification of our systems, and determine the number of True and False classifications and how many of each were correct. We then define *accuracy* to be the number of correct classifications as a percentage of the total number of instances.

3.3 Experimental Methodology

We wished to show that our current system, which focusses on a number of different aspects of sentence–pairs, is more effective than our previous system which used only the cosine and LSI features [9]. Therefore, we have two classifiers which model these systems. Both classifiers are trained and tested on each of the three corpora described above.

3.4 Results

Examining the accuracy results for each corpus (see Table 2), we see that there is an improvement in performance on each corpus when the additional syntactic and lexical features are used.

Corpus	Original System	New System
RTE	52.25%	74.00%
MSR	67.71%	68.75%
MSR-5050	59.71%	63.94%

Table 2. Classifier Accuracy on each corpus, using Original and New features

On the RTE corpus, we achieved an improvement in performance of almost 22% over our baseline entailment system. In contrast, there are relatively modest increases in performance for the MSR and MSR-5050 corpora. This suggests that these corpora contain more challenging instances of entailment than the RTE dataset. We hope that an in–depth analysis will reveal the differences between the entailment corpora.

We also note that the MSR-5050 corpus showed a larger rise in performance (4%) than the 2:1–biased MSR corpus (1%). Since MSR-5050 is a subset of the latter, this supports our assertion (See end of Section 3.1) that the bias of the training set is an important factor when using a decision tree classifier for this task. Since the features we currently use were developed using a balanced dataset as reference, this is reflected in the different rates of improvement.

4 Illustrations of System Performance

In this section we discuss, with examples, some common system errors made by our decision tree classifier.

4.1 Compositional Paraphrases and Syntactic Paraphrases

Recognition of textual entailment is difficult in the case where sentence structure has been changed (thus nullifying our lcs metrics), or where synonyms are extensively used (where we must rely on WordNet and VerbOcean to identify if a relationship exists). It is clear from our system description in Section 2 that the majority of our features deal with the identification of word–level, atomic paraphrase units (e.g., child = kid; eat = devour). Consequently, there are a number of examples where phrasal and compositional paraphrasing has resulted in misclassifications by our system. Some examples of this are shown in Figure 2.

id=1560; task=QA; judgement=TRUE
Text: The technological triumph known as GPS - the Global Positioning System of satellite-based navigation - was incubated in the mind of Ivan Getting.
Hypothesis: Ivan Getting invented the GPS.

id=858; task=CD; judgement=TRUE
Text: Each hour spent in a car was associated with a 6 percent increase in the likelihood of obesity and each half-mile walked per day reduced those odds by nearly 5 percent, the researchers found.
Hypothesis: The more driving you do means you're going to weigh more – the more walking means you're going to weigh less.

Fig. 2. Compositional Paraphrases (misclassified by our system).

Another important type of paraphrase, not addressed explicitly by our system, is the syntactic paraphrase (e.g., "I ate the cake" or "the cake was eaten by me"). However, although we didn't include a parse tree analysis in our approach, it appears that the ROUGE metrics (and to some extent the cosine metric) were an adequate means of detecting syntactic paraphrases. The position of the ROUGE features in high-level nodes in the decision tree confirms that n-gram overlap is an important aspect of textual entailment, but obviously not enough on its own.

We also observed that in some cases syntactic paraphrases prevented the detection of longest common subsequences, and reduced the effectiveness of features that relied on this syntactic analysis. Consequently, parse tree analysis and subsequent normalisation of sentence structure could be an effective solution to this problem.

4.2 Modifying Pre–Texts and Post–Texts

Our LCS–based features focus, by definition, on the parts of the text and hypothesis which are most common to both.

Overall, our LCS–based features were critical to the classification decision; however, we did find instances where sentence pairs were misclassified by over–simplification of the textual entailment task. For example, pair 2028 in Figure 3 shows how the true meaning of the text sentence can extend beyond the longest common subsequence. In addition, pair 1964 shows how coverage limitations in the VerbOcean resource resulted in this example being misclassified as negative, because an antonym relationship between "agree" and "oppose" was not listed.

id=2028; task=QA; judgement=FALSE
Text: *Besancon is the capital of France*'s watch and clock-making industry and of high precision engineering.
Hypothesis: *Besancon is the capital of France*.

id=1964; task=PP; judgement=FALSE
Text: Under the avalanche of Italian outrage *London Underground* has apologised and agreed *to withdraw the poster.*
Hypothesis: *London Underground* opposed *to withdraw the poster*.

Fig. 3. LCS features

4.3 Numerical Strings

During our manual examination of the results we also noticed another crucial analysis component missing from our system: numerical string evaluation. An example is shown in Figure 4. Future development will focus on a normalisation method for evaluating numeric values in the entailment pair.

id=828; task=CD
Text: Jennifer Hawkins is the 21-year-old beauty queen from Australia.
Hypothesis: Jennifer Hawkins is Australia's 20-year-old beauty queen.

id=868; task=CD
Text: Several other people, including a woman and two children, suffered injuries in the incident.
Hypothesis: Several people were slightly wounded, including a woman and three children

Fig. 4. Numerical examples (misclassified by our system)

4.4 The effectiveness of lcs+not feature

The example in Figure 5 illustrates the effectiveness of our lcs+not feature overriding the generally more dominant lcs score. As we can see, there is a strong overlap between the text and hypothesis. However the presence of the antonym pair in the middle of the subsequences acts to negate any entailment, and our system returns a result of "false" in these cases.

id=1432; task=PP
Text: *The report* faults *intelligence agencies* for a "lack of imagination" in not anticipating that al Qaeda could attack the United States using hijacked aircraft.
Hypothesis: *The report* backs *intelligence agencies.*

Fig. 5. LCS+not feature in action

Unfortunately, there are a number of similar cases to this in the corpus, which we classified incorrectly. This may be due to the coverage limitations of the knowledge repositories we are using (i.e. WordNet, VerbOcean) to recognise all correct synonym and antonym verb relations.

5 Conclusion

In this paper, we showed how Textual Entailment Recognition can contribute to various areas of research in Natural Language Processing. We showed that the syntactic and semantic features provide an improvement in system performance, compared to the purely lexical features. This was shown to be especially true for sentence–pairs derived from such tasks as Comparable Documents (CD) and Paraphrase Acquisition (PP). Future research will focus on deriving new features which will improve performance further. Some of these will attempt to overcome some of the problems described earlier.

We also plan to evaluate our system by judging its performance in two challenging NLP applications: Question–Answering and Multi–document Summarisation. Our aim will be to show that the identification of semantically–equivalent sentences using our technique can improve the overall performance of our system on these tasks.

References

1. Chklovski, T., Pantel, P.: *VerbOcean: Mining the Web for Fine–Grained Semantic Verb Relations*. Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP-04), 2004.

2. Dagan, I., Glickman, O., Magnini, B.(eds): Proceedings of the PASCAL Recognising Textual Entailment Challenge Workshop. April 11-13 2005, Southampton, UK.
3. Dolan, W., et al.: *Microsoft Research Paraphrase Corpus.* April 2005. At `http://research.microsoft.com/~billdol/`.
4. Deerwester, S., et al.: *Indexing by Latent Semantic Analysis.* Journal of the American Society for Information Science, 1990.
5. Landauer, T.K., Foltz, P.W., Latham, D: *Introduction to Latent Semantic Analysis.* Discourse Processes, 1998.
6. Lin, C.-Y., Hovy, E.: *Automatic Evaluation of Summaries using n-gram co–occurence statistics.* Proc. Document Understanding Conference (DUC), National Institute of Standards and Technology, 2004.
7. Document Understanding Conference (DUC), National Institute of Standards and Technology, USA. At `http://duc.nist.gov`.
8. Miller, G.A., et al.: *WordNet: Lexical Database for the English Language.* Cognitive Science Laboratory, Princeton University. At `http://www.cogsci.princeton.edu/~wn`.
9. Newman, E., et al.: *Comparing Redundancy Removal Techniques for Multi–Document Summarisation.* Proc. STAIRS 2004, Valencia, Spain.
10. Quinlan, J.R.: *C5.0 Machine Learning Algorithm.* At `http://www.rulequest.com`

HybridTrim: A Hybrid Approach To News Headline Generation

Ruichao Wang, Nicola Stokes, William Doran, John Dunnion, Joe Carthy

Intelligent Information Retrieval Group, Department of Computer Science,
University College Dublin, Belfield, Dublin 4, Ireland
{rachel, Nicola.Stokes, William.Doran, John.Dunnion, Joe.Carthy}@ucd.ie
http://iirg.ucd.ie

Abstract. In this paper, we present the HybridTrim system which uses a machine learning technique to combine linguistic, statistical and positional information to identify topic labels for headlines in a text. We compare our HybridTrim system with the Topiary system which, in contrast, uses a statistical learning approach to finding topic descriptors for headlines. The Topiary system, developed at the University of Maryland with BBN, was the top performing headline generation system at DUC 2004. Topiary-style headlines consist of a number of general topic labels followed by a compressed version of the lead sentence of a news story. The Topiary system uses a statistical learning approach to finding topic labels. We also present a novel approach called HedgeTrim-TF as a baseline system for improving the HedgeTrimmer algorithm, which is part of the Topiary system. The performance of these systems is evaluated using the ROUGE evaluation suite on the DUC 2004 news stories collection.

1 Introduction

A headline is a very short summary (usually less than 10 words) describing the essential message of a piece of text. Like other types of summary, news story headlines are used to help a reader to quickly identify information that is of interest to them in a presentation format such as a newspaper or a website. Although newspaper articles are always accompanied by headlines, there are other types of news text sources, such as transcripts of radio and television broadcasts, where this type of summary information is missing. In this paper we present an approach to headline generation for a single document. This headline generation task was added to the annual summarisation evaluation in the Document Understanding Conference (DUC) 2003 [1]. It was also included in the DUC 2004 evaluation plan where summary quality was automatically judged using a set of word overlap metrics called ROUGE [2].

Eighteen research groups participated in the headline generation task at DUC 2004, Task 1: very short summary generation. The headline system which performed the best was the Topiary system. It generated headlines by combining a set of topic descriptors with a compressed version of the lead sentence, e.g. **SLOBODAN_MILOSEVIC KOSOVO SERB TRIBUNAL: Yugoslavia**

must cooperate. As can be seen these topic descriptors provide the reader with a general event description while the lead compressed sentence provides a more focussed summary of the news story. These topic descriptors were automatically identified using a statistical approach called Unsupervised Topic Discovery (UTD) [3]. The disadvantage of this technique is that meaningful topic descriptors will only be identified if this technique is trained on the corpus containing the news stories that are to be summarised. In addition, the corpus must contain clusters of related news stories to ensure that reliable cooccurrence statistics are generated.

The aim of this paper is to improve Zajic, Dorr and Schwartz's Topiary-style parse-and-trim approach to headline summarisation [3]. In this paper we compare the UTD method with an alternative topic label identifier that can be trained on an auxiliary news corpus, and observe the effect of these labels on summary quality when combined with compressed lead sentences. Our topic labelling technique works by combining linguistic and statistical information about terms using the C5.0 [4] machine learning algorithm to predict which words in the source text should be included in the resultant gist with the compressed lead sentence. In this paper, we compare the performance of this system, HybridTrim, with the Topiary system and a number of other baseline gisting systems on a collection of news documents from the DUC 2004 corpus [1].

We describe the HybridTrim system, Topiary system and our new baseline HedgeTrim-TF system in more detail in Sections 2, 3 and 4, respectively. The performance of the HybridTrim and Topiary systems is compared with a baseline system called TFTrim, which is a term frequency-based approach. The performance of the HedgeTrim-TF system is compared with the Trim system [5]. These systems were evaluated using the ROUGE evaluation metrics on the DUC 2004 collection, and a manual evaluation performed by four human evaluators. The results of these experiments and our overall conclusions are discussed in Sections 5 and 6, respectively.

2 HybridTrim System

The HybridTrim system uses our implementation of the Hedge Trimmer algorithm [5] and the C5.0 machine learning algorithm to create a decision tree capable of predicting which words in the source text should be included in the resultant gist.

To identify pertinent topic labels the algorithm follows a two-step process: the first step involves creating an intermediate representation of a source text, and the second involves transforming this representation into a summary text. The intermediate representation we have chosen is a set of features that we believe are good indicators of possible 'summary words'. We focus our efforts on the content words of a document, i.e. the nouns, verbs and adjectives that occur within the document. For each occurrence of a term in a document, we calculate several features: the *tf*, or term frequency of the word in the document; the *idf*, or inverse document frequency of the term taken from an auxiliary corpus [6]; and

the relative position of a word with respect to the start of the document in terms of word distance. We also include binary features indicating whether a word is a noun, verb or adjective and whether it occurs in a noun or proper noun phrase. The final feature is a lexical cohesion score calculated with the aid of a linguistic technique called *lexical chaining*. Lexical chaining is a method of clustering words in a document that are semantically similar with the aid of a thesaurus, in our case WordNet. Our chaining method identifies the following word relationships (in order of strength): repetition, synonymy, specialisation/generalisation, and part/whole relationships. Once all lexical chains have been created for a text then a score is assigned to each chained word based on the strength of the chain in which it occurs. More specifically, as shown in Equation 1, the chain strength score is the sum of the strength score assigned to each word pair in the chain.

$$Score(Chain) = \sum((reps_i + reps_j) \times rel(i,j)) \quad (1)$$

where reps_i is the frequency of word i in the text, and *rel(i,j)* is a score assigned based on the strength of the relationship between word i and j. More information on the chaining process and cohesion score can be found in [7], [8].

Using the DUC 2003 corpus as the training data for our classifier, we then assigned each word a set of values for each of these features, which are then used with a set of gold standard human-generated summaries to train a decision tree summarisation model using C5.0. The DUC 2003 evaluation provides four human summaries for each document, where words in the source text that occur in these model summaries are considered to be positive training examples, while document words that do not occur in these summaries are considered to be negative examples. Further use is made of these four summaries, where the model is trained to classify a word based on its summarisation potential. More specifically, the appropriateness of a word as a summary term is determined based on the class assigned to it by the decision tree. These classes are ordered from strongest to weakest as follows: 'occurs in 4 summaries', 'occurs in 3 summaries', 'occurs in 2 summaries', 'occurs in 1 summary', 'occurs in none of the summaries'. If the classifier predicts that a word will occur in all four of the human-generated summaries, then it is considered to be a more appropriate summary word than a word predicted to occur in only three of the model summaries, and so on. This resulted in a total of 103267 training cases, where 5762 instances occurred in one summary, 1791 in two, 1111 in three, 726 in four, and where finally 93877 instances were negative. A decision tree classifier was then produced by C5.0 based on this training data. To gauge the accuracy of our decision tree topic label classifier, we used a training/test data split of 90%/10%, and found that on this test set the classifier had a precision (true positives divided by true positives and false positives) of 63% and recall (true positives divided by true positives and false negatives) of 20%.

3 Topiary System

In this section, we describe the Topiary system developed at the University of Maryland with BBN Technologies. As already stated, this system was the top performing headline generation system at DUC 2004. A Topiary-style headline consists of a set of topic labels followed by a compressed version of the lead sentence. Hence, the Topiary system views headline generation as a two-step process: first, create a compressed version of the lead sentence of the source text, and second, find a set of topic descriptors that adequately describe the general topic of the news story. We will now look at each of these steps in more detail.

In [5] Dorr, Zajic and Schwartz stated that when human subjects were asked to write titles by selecting words in order of occurrence in the source text, 86.8% of these headline words occurred in the first sentence of the news story. Based on this result Dorr, Zajic and Schwartz, concluded that compressing the lead sentence was sufficient when generating titles for news stories. Consequently, their DUC 2003 system HedgeTrimmer used linguistically-motivated heuristics to remove constituents that could be eliminated from a parse tree representation of the lead sentence without affecting the factual correctness or grammaticality of the sentence. These linguistically-motivated trimming rules [3], [5] iteratively remove constituents until a desired sentence compression rate is reached.

The compression algorithm begins by removing determiners, time expressions and other low content words. More drastic compression rules are then applied to remove larger constituents of the parse tree until the required headline length is achieved. For the DUC 2004 headline generation task systems were required to produce headlines no longer than 75 bytes, i.e. about 10 words. The following worked example helps to illustrate the sentence compression process.[1]

Lead Sentence: The U.S. space shuttle Discovery returned home this morning after astronauts successfully ended their 10-day Hubble Space telescope service mission.

Parse: (S (NP (NP The U.S. space shuttle) Discovery) (VP returned (NP home) (NP this morning)) (SBAR after (S (NP astronauts) (VP (ADVP successfully) ended (NP their 10-day Hubble Space telescope service mission)))))

1. Choose leftmost S of parse tree and remove all determiners, time expressions and low content units such as quantifiers (e.g. each, many, some), possessive pronouns (e.g. their, ours, hers) and deictics (e.g. this, these, those):

   Before: (S (NP (NP **The** U.S. space shuttle) Discovery) (VP returned (NP home) (NP **this morning**)) (SBAR after (S (NP astronauts) (VP (ADVP successfully) ended (NP **their** 10-day Hubble Space telescope service mission)))))

[1] The part of speech tags in the example are explained as follows: **S** represents a simple declarative clause; **SBAR** represents a clause introduced by a (possibly empty) subordinating conjunction; **NP** is a noun phrase; **VP** is a verb phrase; **ADVP** is an adverbial phrase.

After: (S (NP (NP U.S. space shuttle) Discovery) (VP returned (NP home)) (SBAR after (S (NP astronauts) (VP (ADVP successfully) ended (NP 10-day Hubble Space telescope service mission)))))

2. The next step iteratively removes constituents until the desired length is reached. In this instance the algorithm will remove the trailing SBAR.

   Before: (S (NP (NP U.S. space shuttle) Discovery) (VP returned (NP home)) *(SBAR after (S (NP astronauts)(VP (ADVP successfully) ended (NP 10-day Hubble Space telescope service mission))))*)

   After: U.S. space shuttle Discovery returned home.

Like the 'trailing SBAR' rule, the other iterative rules identify and remove non-essential relative clauses and subordinate clauses from the lead sentence. A more detailed description of these rules can be found in [3], [5]. In this example, we can see that after compression the lead sentence reads more like a headline. The readability of the sentence in this case could be further improved by replacing the past tense verb 'returned' with its present tense form; however, this refinement is not currently implemented by the Topiary system or by our implementation of this compression algorithm.

As stated earlier, a list of relevant topic words is also concatenated with this compressed sentence resulting in the final headline. The topic labels are generated by the UTD (Unsupervised Topic Discovery) algorithm [3]. This unsupervised information extraction algorithm creates a short list of useful topic labels by identifying commonly occurring words and phrases in the DUC corpus. So for each document in the corpus it identifies an initial set of important topic names for the document using a modified version of the *tf.idf* metric. Topic models are then created from these topic names using the OnTopic^{TM} software package. The list of topic labels associated with the topic models closest in content to the source document is then added to the beginning of the compressed lead sentence produced in the previous step, resulting in a Topiary-style summary.

One of the problems with this approach is that it will only produce meaningful topic models and labels if they are generated from a corpus containing additional on-topic documents on the news story being summarised.

4 HedgeTrim-TF System

The HedgeTrim-TF approach is based on the HedgeTrimmer algorithm, also developed at the University of Maryland with BBN Technologies [5]. It uses simple term frequencies to measure terms significance in the document, then the term frequencies are used as a guide to remove the constituents of the parsed lead sentences. Our previous experiments showed that a TF technique that uses source document term frequency statistics to identify salient topic labels can outperform both a knowledge-based NLP approach (using WordNet), and a statistical-based approach (UTD) requiring additional word frequency and cooccurrence information from the entire DUC 2004 corpus [10]. In this experiment, we combine the

TF technique with a linguistically-motivated heuristic approach since we are concentrating on sentence extraction rather than word extraction. There are two major steps to this process. Firstly, the system creates a word list ordered by frequency. A stopwords list is used to exclude low content words and the top 10 most frequent words are kept. Secondly, the system creates a compressed version of the lead sentence of the source text. If one of these words occurs in the parsed lead sentence when the system applies the HedgeTrimmer algorithm, then we consider keeping the constituent of the sentence rather than removing it. The operation of this technique is illustrated in the following example.

Lead Sentence: The United States and Russia are ratcheting up the pressure on President Slobodan Milosevic, warning that NATO airstrikes are inevitable unless he takes decisive action soon to end the humanitarian crisis in the southern Serbian province.

Parse: (S (NP The United States and Russia) (VP are (VP ratcheting up (NP the pressure on President Slobodan Milosevic, (VP warning (SBAR that NATO airstrikes are inevitable unless he takes decisive action soon to end the humanitarian crisis the southern Serbian province))))))

Top 10 TF words: Situation Kosovo Milosevic Airstrikes Nato President Yugoslavia Defense Takes Holbrooke

HedgeTrimmer: United States and Russia ratcheting up pressure.

HedgeTrim-TF: United States and Russia ratcheting up pressure on President Slobodan Milosevic.

A more detailed description of these rules can be found in [3], [5]. In this example, we can see that the headline produced by the HedgeTrim-TF system gives a little bit more information. However, one of the problems with this approach is that sometimes the compressed sentence is longer than the threshold ($\leq$ 75 bytes).

5 Evaluation and Results

In this section we present the results of our headline generation experiments on the DUC 2004 corpus.[2] We use the ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metrics to evaluate the quality of our automatically generated headlines. In DUC 2004 Task 1, participants were asked to generate very short ($\leq$75 bytes) single-document summaries for documents on TDT-defined events.

The DUC 2004 corpus consists of 500 Associated Press and *New York Times* newswire documents. The headline-style summaries created by each system were evaluated against a set of human-generated (or model) summaries using the ROUGE metrics. The format of the evaluation was based on six scoring metrics: ROUGE-1, ROUGE-2, ROUGE-3, ROUGE-4, ROUGE-LCS and ROUGE-W.

[2] Details of our official DUC 2004 headline generation system can be found in [9]. This system returned a list of keywords rather than 'a sentence + keywords' as a headline. It used a decision tree classifier to identify appropriate summary terms in the news story based on a number of linguistic and statistical word features.

The first four metrics are based on the average n-gram match between a set of model summaries and the system-generated summary for each document in the corpus. ROUGE-LCS calculates the longest common sub-string between the system summaries and the models, and ROUGE-W is a weighted version of the LCS measure. Thus for all ROUGE metrics, the higher the ROUGE value the better the performance of the summarisation system, since high ROUGE scores indicate greater overlap between the system summaries and their respective models. Lin and Hovy [2] have shown that these metrics correlated well with human judgements of summary quality, and the summarisation community is now accepting these metrics as a credible and less time-consuming alternative to manual summary evaluation. In the official DUC 2004 evaluation all summary words were stemmed before the ROUGE metrics were calculated; however, stopwords were not removed. No manual evaluation of headlines was performed.

5.1 ROUGE Evaluation Results

Table 1 shows the results of our headline generation experiments on the DUC 2004 collection. Seven systems in total took part in this evaluation, three Topiary-style headline generation systems and four baselines: the goal of our experiments was to evaluate linguistically-motivated heuristic approaches to title generation, and establish which of our alternative techniques for padding Topiary-style headlines with topic labels works best.

Since the DUC 2004 evaluation, Lin [11] has concluded that certain ROUGE metrics correlate better with human judgements than others, depending on the summarisation task being evaluated, i.e. single document, headline, or multi-document summarisation. In the case of headline generation, Lin found that ROUGE-1, ROUGE-L and ROUGE-W scores worked best and hence only these scores are included in Table 1.

Table 1. ROUGE scores for headline generation systems

	Systems	ROUGE-1	ROUGE-L	ROUGE-W
Topiary-style Systems	TFTrim	0.2793	0.2134	0.1260
	HybridTrim	0.2736	0.2138	0.1267
	Topiary	0.2491	0.1995	0.1189
Baseline Systems (Keywords Extraction)	TF	0.2443	0.1707	0.0981
	Hybrid	0.2190	0.1760	0.1020
	UTD	0.1591	0.1304	0.0780
Baseline Systems (Sentence Extraction)	HedgeTrim-TF	0.2250	0.1956	0.1163
	Trim	0.2006	0.1825	0.1010

As the results show, the best performing topic labelling techniques are the TF and Hybrid systems. The TF system is a baseline system that chooses high

frequency content words as topic descriptors. The Hybrid system is our decision tree classifier described in section 2. Both these systems outperform the UTD method. The HedgeTrim-TF system is our new baseline system that uses high frequency content words as a guide to applying the HedgeTrimmer algorithm to a lead sentence, as descibed in section 4, and it displays improved performance over the Trim (HedgeTrimmer algorithm) system.

The top three performing systems in the table combine topic labels with a compressed version of the lead sentence. Comparing these results to the Trim system and HedgeTrim-TF system (that both return the reduced lead sentence only), it is clear that the addition of topic descriptors greatly improves summary quality. The performance of the baseline TFTrim system and the HybridTrim system are very similar for all three Rouge metrics; however, both systems outperform the Topiary headline generator.

6 Conclusions and Future work

In this paper, we have compared the performance of three Topiary-style headline generation systems that use three distinct techniques for 'padding out' compressed lead sentences in the automatic generation of news story headlines. The results of our experiments using the ROUGE evaluation suite indicate that topic descriptors identified by simple term frequency counts in the source document outperform either keywords identified by a statistical/linguistic approach to topic label identification, or statistically-derived topic labels from the DUC 2004 corpus using the UTD algorithm.

In future work, we intend to proceed by improving the sentence compression procedure described in this paper. We are currently working on the use of term frequency information as a means of improving the performance of the Hedge Trimmer algorithm by limiting the elimination of important constituents of the parse tree during sentence compression.

References

1. Document Understanding Conference (DUC). At: http://duc.nist.gov/. (Accessed May 2005).
2. Lin C-Y. and Hovy E.: Automatic Evaluation of Summaries using n-gram Co-occurrence Statistics. In: Proceedings of HLT/NACCL (2003)
3. Zajic D., Dorr B. and Schwartz R.: BBN/UMD at DUC-2004: Topiary. In: Proceedings of the Document Understanding Conference (DUC) (2004)
4. Quinlan R., C5.0: An Informal Tutorial. At: http://www.rulequest.com/see5-unix.html (1998), (Accessed March 2005)
5. Dorr B., Zajic D., Schwartz R.: Hedge Trimmer: A Parse-and-Trim Approach to Headline Generation. In: Proceedings of the Document Understanding Conference (DUC 2003)
6. TDT Pilot Study Corpus, http://www.nist.gov/speech/tests/tdt. (Accessed May 2005)

7. Doran W., Stokes N., Dunnion J., Carthy J.: Assessing the Impact of Lexical Chain Scoring Methods and Sentence Extraction Schemes on Summarization. In: Alexander Gelbukh (ed.): Proceedings of the Conference on Intelligent Text Processing and Computational Linguistics (CICLing), Seoul (2004)
8. Stokes N.: Applications of Lexical Cohesion Analysis in the Topic Detection and Tracking domain. Ph.D. Thesis. Dept. of Computer Science, University College Dublin (2004)
9. Doran W. P., Stokes N., Newman E., Dunnion J., Carthy J., Toolan F.: News Story Gisting at University College Dublin. In: Proceedings of the Document Understanding Conference (DUC 2004)
10. Wang R., Stokes N., Doran W., Newman E., Carthy J., Dunnion J.: Comparing Topiary-style approaches to Headline Generation. In: Proceedings of the 27th European Conference on Information Retrieval, Spain (2005)
11. Lin C-Y.: ROUGE: A Package for Automatic Evaluation of Summaries. In: Proceedings of the ACL workshop, Text Summarization Branches Out, Barcelona, Spain (2004) 56-60

Information Retrieval

Evolving Co-occurrence Based Query Expansion Schemes in Information Retrieval Using Genetic Programming

Ronan Cummins and Colm O'Riordan

Dept. of Information Technology,
National University of Ireland,
Galway, Ireland.
ronan.cummin@nuigalway.ie,colmor@it.nuigalway.ie

Abstract. Global query expansion techniques have long been proposed as a solution to overcome the problem of term mismatch between a query and its relevant documents. This paper describes a method which automatically tackles the problems of how to find the best terms for the expansion of a particular query and secondly, how to weight these terms for use with the original query. Genetic Programming is used to evolve schemes for term selection using global (collection-wide) co-occurrence measures. The schemes evolved are also used to weight the term in the expanded query as they are a measure of the term's importance in relation to the query. As a result, the genetic program has to learn a suitable scheme for identifying the best correlates for the query concept and also a scheme that correctly weights these in relation to each other. These schemes are tested on standard test collections and show a significant increase in performance on the training data but only modest improvement on the collections that are not included in training.

1 Introduction

Information Retrieval (IR) is concerned with the automatic retrieval of all relevant documents given a user need (query). However, vocabulary differences between the user and the supplier of information have often lead to a difficulty in retrieving many documents. Query expansion techniques have long been proposed as a means of overcoming term mismatch between the user's vocabulary and the vocabulary of the documents in the collection. Query expansion techniques typically add a number of non-query terms to the original query based on some heuristics in order to improve the performance of the original query. Typically, there are two types of query expansion methods; Local (pseudo-relevance or blind feedback) and Global (automatic thesaurus construction) query expansion techniques. This paper is concerned with the latter. In automatic thesaurus construction, terms are added to the original query based on their co-occurrence frequencies with query terms throughout the entire collection.

Recently there have been more and more attempts applying machine learning techniques to the domain of IR. Genetic Programming (GP) has been adopted

by some researchers as it has certain advantages over other machine learning techniques. In particular, GP outputs a symbolic representation of a solution which can be used in further analysis. As a result, GP solutions are often quite general and are particularly suited to such problems. Developed in the early 1990's, the GP area [1] has grown and helped to solve problems in a variety of domains. GP is inspired by Darwinian theory of natural selection, where individuals that have a higher fitness value will survive and produce offspring. GP can be viewed as an artificial way of selective breeding.

This paper presents a Genetic Programming framework that artificially breeds query expansion selection schemes for use in a standard vector space framework. The next section introduces some background material in both query expansion and GP. Section Three describes the system and experimental design. Results and analysis are discussed in detail in section Four. Finally, our conclusions are presented in section Five.

2 Background

2.1 Global Query Expansion approaches

Global query expansion approaches analyse the entire document collection and use co-occurrence relationships between terms to build a matrix of term-term relationships. Usually, term-term matrices of this type contain weights which are a measure of how synonymous one term is with another. These matrices are large and computationally expensive to compute. The matrices are used to cluster terms based on their co-occurrence data in the hope that terms that are closer together in this term-space are synonymous. Conceptually, the role of documents and terms are interchanged in the retrieval model. In essence, documents become the features of the term. Thus, two terms that appear in the same document are indexed by a similar feature and are deemed to have some type of synonymous relationship. Many formulas have been proposed to measure the association between two terms using co-occurrence data. The similarity between two terms t_i and t_j can be determined by evaluating the difference between the two-vectors $\overrightarrow{t_i} = (d_{i1}, d_{i2}, ..., d_{in})$ and $\overrightarrow{t_j} = (d_{j1}, d_{j2}, ..., d_{jn})$ in the document vector space. A simple binary weighting on these document weights would lead to the following cosine formulation of similarity between two terms:

$$cos(t_i, t_j) = \frac{df(t_i, t_j)}{\sqrt{df(t_i)df(t_j)}} \tag{1}$$

where $df(t_i, t_j)$ is the number of documents in which both t_i and t_j occur and $df(t_i)$ is the number of documents in which t_i occurs. There are many variations of such formulas which aim to accurately find the best synonyms for a term. Many approaches have attempted to add the best synonyms for each individual term in the query to the original query. Many of these approaches have seen relatively little or no improvement in the retrieval of relevant information over the original query [2, 3].

However, independently analysing each query term ignores the concept of the query. Thus, terms that are selected for expansion based on this type of method typically have no context related to them (i.e. a term maybe closely related to one of the query terms but may not be related to the concept of the entire query). A concept based approach to query expansion has previously been attempted by promoting terms similar to the entire query by summing the individual associations for each term in the query [2]. Thus, terms that would be chosen for expansion would be similar to many terms in the query and thus have a concept associated with them. In this way the problem of term independence of query terms is somewhat overcome. This approach was somewhat successful on certain collections although the baseline weighting scheme used to weight the original terms in the query was poor as shown by the results on the NPL collection [2, 4]. Other attempts at creating collection dependent automatic thesauri by limiting the co-occurrence of terms to certain parts of text (e.g. paragraphs, sentences and phrases) have shown to be somewhat effective [5, 4].

Once terms have been chosen to be added to the original query by some expansion algorithm, the weight of the terms to be added must be determined. Term selection and re-weighting are the two main challenges that face global thesaurus techniques.

2.2 Standard term-weighting approaches

The BM25 weighting scheme, developed by Robertson et al. ([6]), is a weighting scheme based on the probabilistic model. The weight assigned to a term in the BM25 scheme is a product of Okapi-*tf* and *idf*. Okapi-*tf* is calculated as follows:

$$Okapi\text{-}tf = \frac{rtf}{rtf + k_1((1-b) + b\frac{dl}{dl_{avg}})} \tag{2}$$

where rtf is the raw term frequency and dl and dl_{avg} are the length and average length of the documents respectively. k_1 and b are tuning parameters. The idf of a term as determined in the BM25 formula is as follows:

$$idf_t = log(\frac{N - df_t + 0.5}{df_t + 0.5}) \tag{3}$$

where N is the number of documents in the document set and df_t is the document frequency of term t. The score for the document d can be calculated as followed:

$$BM25(Q, d) = \sum_{t \in Q \cap d} (Okapi\text{-}tf \times idf_t \times qrtf_t) \tag{4}$$

where $qrtf_t$ is the raw term frequency of t in the query Q. Thus, $BM25(Q, d)$ is a measure of the similarity between the document d and the query Q.

2.3 Genetic Programming

GP is a heuristic stochastic searching method that is efficient for navigating large, complex search spaces. The advantage of this evolutionary approach is

that it can help to solve problems in which the roles of variables are not correctly understood. GP is often used to automatically derive functions whose variables combine and react in complex ways.

Initially, a random population of solutions is created. The solutions are modelled as tree-like structures with operators as internal nodes (functions) and operands as leaf nodes (terminals). These nodes are often referred to as genes and their values as alleles. Each solution is rated based on how it performs in its environment. This is achieved using a fitness function. Once this is done, reproduction can occur. Solutions with a higher fitness will produce more offspring. Goldberg uses the roulette wheel example where each solution is represented by a segment on a roulette wheel proportionately equal to the fitness of the solution [7]. Reproduction (recombination) can occur in variety of ways. The most common form is sexual reproduction where two different individuals (parents) are selected and two separate children are created by combining the genotypes of both parents. The coded version of a solution is called its genotype, as it can be thought of as the genome of the individual, while the solution in its environment is called its phenotype. The fitness is evaluated on the phenotype of a candidate solution while reproduction and crossover is performed on the genotype. Once the recombination process is complete each individual's fitness in the new generation is evaluated and the selection process starts again. The algorithm usually ends when a certain number of generations have been completed, when convergence of the population has been detected or when an individual is found with an acceptable fitness.

3 Design and Experimental Setup

3.1 Term-Selection

The GP approach adopted evolves the scheme used to select and weight terms for use in the expanded query in order to improve the retrieval performance of the system. For each query expansion scheme, each term in the corpus is rated based on how close it is to the query concept. The following formula shows the similarity (or Term Selection Value) between the term t and the entire query Q:

$$TSV(Q,t) = \sum_{q \in Q} (correlation_{qt} \times qrtf_q) \tag{5}$$

where Q is the query, q is a query term, t is a non-query term in the corpus, $qrtf_q$ is the raw term-frequency of term q in the query and $correlation_{qt}$ is the query expansion scheme to be evolved. As a result, $correlation_{qt}$ is a measure of the degree to which term t and the query term q are related by co-occurrence measures. By extension, $TSV(Q,t)$ represents the similarity between the entire query Q and a non-query term t. For each query Q, a number of top terms are chosen and added to the query vector. The number of terms added to the query can easily be increased without any change to the formula as terms further down the ranked list should have less significance in the expanded query as they are weighted as a function of their $TSV(Q,t)$ value.

3.2 Term Re-Weighting

We assume that the weight of an expanded term is a function of $TSV(Q,t)$ (i.e. the similarity of that term to the query). It is also logical to assume that the weight of the expansion term is also related to the weighting scheme applied to the original query terms (i.e. a *tf-idf* type scheme). Thus, the following formula is how our system scores the complete expanded query (EQ) in relation to a document d:

$$sim(EQ,d) = BM25(Q,d) + \sum_{t \in E} TSV(Q,t) \times Okapi\text{-}tf \times idf_t \qquad (6)$$

where EQ is the expanded query, Q is the original query, E is the set of expansion terms. Thus, a weighting of 1 for $TSV(Q,t)$ would indicate that the expansion term t is as important as if it had occurred in the original query. In this way the GP can also learn the correct weighting for expansion terms.

3.3 Document collections and preprocessing

The document collections used in this research are the Medline, CISI, Cranfield, NPL, LISA and OHSUMED collections[1]. Only the first 30 queries for the CISI and Cranfield collections are used as efficiency is of prime concern. The largest collection (OHSU88) is a subset of document of the full OHSUMED collection. It consists of half of the documents from the 1988 collection. All documents and queries are pre-processed by removing standard stop-words and stemmed using Porter's stemming algorithm [8]. The weighting scheme applied to the query terms is a relative term frequency weighting scheme. All queries with no relevant documents are ignored by the system.

Global query expansion techniques are computationally intensive. We reduce the number of terms in the collection by using a feature selection technique which reduces the number of the terms in each corpus to roughly 25%. We eliminate all dilute terms (i.e. terms whose document frequency equals its collection frequency). This has been shown to be a characteristic of evolved weighting schemes on both small and large collections [9]. Typically, this eliminates terms of a low frequency and variations have been used in other feature extraction techniques like document frequency thresholding. Table 1 shows the characteristics of the document collections after preprocessing and feature selection.

3.4 Terminal and Function set

To determine the terminal and function set, it is neccessary to consider the characteristics of the documents in which the query terms and possible expansion terms co-occur in the entire corpus. It is also important to consider the characteristics of each query term and each possible expansion term independently in the entire corpus. Table 2 shows the terminal set chosen. This set is divided into

[1] http://www.dcs.gla.ac.uk/idom/ir_resources/test_collections/

Table 1. Characteristics of document collections

Collection	Docs	Terms	Reduced Terms	Avg Len	Qrys	Avg len
Medline	1,033	10,975	3,614	56.8	30	11
Cranfield	1,400	9,014	2,518	59.6	30	8.3
CISI	1,460	8,342	2,110	47.8	30	7.56
LISA	6,004	16,168	4,411	36.3	35	6.78
NPL	11,429	7,759	2,468	18.78	93	6.78
OSHU88	35,412	113,145	31,496	48.03	61	5.05

two parts in order to draw attention to the source of information for the chosen terminals. The top half of the table shows collection-wide statistics for both the query term q and the possible expansion term t independently of each other. The bottom part of the terminal set shows measures of the set of documents in which both the query term q and possible expansion term t co-occur. We will define the set of documents in which both t and q occur as C_{qt}.

Table 2. Terminal Set

Terminal	Description
1	*the constant* 1
0.5	*the constant* 0.5
cf_q	frequency of a query term (q) in the collection
cf_t	frequency of a non-query term (t) in the collection
df_q	no. of documents a query term (q) appears in
df_t	no. of documents a non-query term (t) appears in
N	no. of documents in a collection
S	no. of words in the collection
$\|Q\|$	no. of terms in the query Q
bin_{qt}	no. of documents in C_{qt}
$prod_{qt}$	sum of the product of the term-frequencies in C_{qt}
min_{qt}	sum of the minimum of the term-frequencies in C_{qt}
sum_{qt}	sum of the sum of the term-frequencies in C_{qt}
cof_q	sum of the term-frequencies for q in C_{qt}
cof_t	sum of the term-frequencies for t in C_{qt}
W_{qt}	total no. of words in C_{qt}

For the set of documents in which two terms co-occur, we combine the within-document (local) measures for those terms in an intiutive manner. For example the $prod_{qt}$ measure is often used to measure the correlation between two terms and is calculated as follows:

$$prod_{qt} = \sum_{d \in N} (tf_{dq} \times tf_{dt}) \tag{7}$$

where tf_{dt} is the term-frequency of t in document d. The bin_{qt} measure is calculated similary assuming a binary weighting on the within-document term-frequencies. The min_{qt} measure is the sum of the intersection (or minimum) of the term-frequencies.

Table 3. Function Set

Function	*Description*
+, ×, /, -	standard arithmetic functions
log	the natural log
$\sqrt{}$	square-root function
sq	square

3.5 Fitness Function

The mean average precision (MAP), used as the fitness function, is calculated for each scheme by comparing the ranked list returned by the system for each query expansion scheme against the human determined relevant documents for each query. Mean average precision is calculated over all points of recall and is frequently used as a performance measure in IR systems as it provides a measure of both the accuracy and recall of the retrieval system.

3.6 GP Parameters

All experiments are run for 70 generations with an initial population of 2000. Populations of less than 500 for this problem converge prematurely as the terminal set is quite large. Experimental analysis shows us that the population converges before 70 generations when using the largest terminal and function set. The solutions are trained on an entire collection and query set. They are then tested for generality on the collections that were not included in training. Trees are limited to a depth of 10. The aim is to discover general natural language characteristics for query expansion that will aid retrieval performance. We evolve these term-selection schemes by adding the top 8 terms to the original query.

4 Results and Analysis

4.1 Evolved Term Selection Schemes

We evolved solutions on the three smaller collections. The best solution for each collection was chosen for evaluation on previously unseen data. These solution will be refered to as the Medline, CISI and Cranfield solutions for the remainer of this paper. The CISI solution (8) and Cranfield solution (9) evolved are shown as an example of the solutions found.

$$correlation_{qt} = \frac{min_{qt}}{(rdf_t \times log(cf_t)) + \sqrt{sum_{qt} \times cf_q}} \tag{8}$$

$$correlation_{qt} = \frac{scf_q + bin_{qt}}{(((((bin_{qt}/W_{qt}) \times (0.5 + rdf_t)) \times N) + scf_q)/(log(prod_{qt}))) + W_{qt}} \tag{9}$$

Table 4 shows the MAP for the original query and the expanded queries on all the collections included in this research. From an evaluation perspective the most important collections are those which are not included in training.

Table 4. MAP for expanded queries using best evolved solutions

				Solution		
Collection	Docs	Qrys	BM25	CISI	Medline	Cranfield
CISI	1,460	30	19.51%	23.37%	20.18%	20.40%
Medline	1,033	30	53.51%	55.63%	65.37%	56.50%
Cranfield	1,400	30	38.43%	36.60%	39.73%	41.61%
LISA	6,004	35	35.01%	36.43%	35.37%	34.25%
NPL	11,429	93	28.75%	29.01%	28.74%	28.77%
OHSU88	35,412	63	23.25%	24.33%	21.78%	23.43%

Firstly we can see that there is a significant increase in MAP on the Medline collection when using the solution specific to that collection. This confirms previous concept-based approaches [2] which also show a similar increase on this collection. However, this Medline solution seems to be specific to the collections as there is no substantial improvement on other collections. The CISI and Cranfield solutions do not achieve as high an average precision on the Medline collection as the Medline solution does on its own training data. The solution found on the CISI collection is the only solution that increases average precision on all three larger collections.

4.2 Increasing Terms added to Queries

To determine whether adding only 8 terms to each query is sufficient to learn a general term clustering algorithm, we calculated the MAP for the evolved

formulas for expanded queries of various lengths. This is investigated as previous research recommends expanding the query by up to 100 terms [2]. This amount of expansion is computationally very expensive and rather unrealistic in a real IR setting. Table 5 shows the MAP of the evolved formulas tested on their training data adding a varying number of terms. We see that, in general, adding more terms does not significantly increase or decrease the MAP of the queries.

Table 5. MAP for different length expanded queries

				Terms Added to each query						
Collection	Docs	Qrys	BM25	Top 8	Top 16	Top 24	Top 32	Top 40	Top 48	Top 96
CISI	1,460	30	19.51%	23.37%	23.68%	23.77%	23.35%	23.43%	23.41%	22.40%
Medline	1,033	30	53.51%	65.37%	65.62%	66.36%	66.46%	66.38%	66.32%	66.30%
Cranfield	1,400	30	38.43%	41.61%	41.56%	41.17%	40.79%	40.62%	40.96%	38.23%

Table 5 indicates that the schemes learned for the respective collections correctly weight the quality of each expanded term. As an example, we take two queries from the Medline collection and look at the weights assigned to the top 8 most similar terms according to the Medline solution. The 21^{st} Medline query which is stemmed to the following stems:

```
Medline Query 21: {languag develop infanc pre-school ag}
```

and has its 8 most similar terms, according to the Medline solution, shown in Table 6. Similarly, the 23^{rd} query which is preprocessed to the following:

```
Medline Query 23: {infantil autism}
```

is also shown in Table 6.

Table 6. Scores for Expansion terms for two sample queries

Query 21		Query 23	
Terms	TSV Score	Terms	TSV Score
deaf	0.891525	autist	2.12915
children	0.659627	mental	1.17509
learn	0.645482	child	0.733794
speech	0.497686	children	0.606834
word	0.356095	schizophrenia	0.569791
impair	0.323454	contact	0.407135
spoken	0.314593	symptom	0.324127
teach	0.302923	situat	0.27934

From these tables we can see that the evolved schemes can promote terms which seem to be related to the query concept and provides a weighting which is related to the quality of the expansion term. It can also promote different forms of query terms that Porter's stemming algorithm has failed to conflate. However, although solutions can be evolved that correctly find good expansion terms for a query, these solutions seem to be domain specific.

5 Conclusion

The results of this approach seem to confirm many previous approaches in that global co-occurrence data is unlikely to bring about a substantial general increase in the performance of IR systems [3]. However, we have learned domain specific formulas for finding good expansion terms. Importantly, the approach adopted also learns a mechanism for weighting these terms in relation to the original query without having to develop the weighting of such expansion terms analytically.

References

1. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge, MA, USA (1992)
2. Qiu, Y., Frei, H.P.: Concept-based query expansion. In: Proceedings of SIGIR-93, 16th ACM International Conference on Research and Development in Information Retrieval, Pittsburgh, US (1993) 160–169
3. Peat, H.J., Willett, P.: The limitations of term co-occurrence data for query expansion in document retrieval systems. JASIS **42** (1991) 378–383
4. Jing, Y., Croft, W.B.: An association thesaurus for information retrieval. Technical report, University of Massachusetts, Amherst, MA, USA, Amherst, MA, USA (1994)
5. Xu, J., Croft, W.B.: Query expansion using local and global document analysis. In: SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (1996) 4–11
6. Robertson, S.E., Walker, S., Hancock-Beaulieu, M., Gull, A., Lau, M.: Okapi at TREC-3. In: In D. K. Harman, editor, The Third Text REtrieval Conference (TREC-3) NIST. (1995)
7. Goldberg, D.E.: Genetic Algorithms in Search, Optimisation and Machine learning. Addison-Wesley (1989)
8. Porter, M.: An algorithm for suffix stripping. Program **14** (1980) 130–137
9. Cummins, R., O'Riordan, C.: Determining general term weighting schemes for the vector space model of information retrieval using genetic programming. In: 15th Artificial Intelligence and Cognitive Science Conference (AICS 2004). (2004)

Probability-Based Fusion of Information Retrieval Result Sets

D. Lillis, F. Toolan, A. Mur, L. Peng, R. Collier, and J. Dunnion

Department of Computer Science
University College Dublin
Ireland
{david.lillis,fergus.toolan,mur.angel,
peng.liu,rem.collier,john.dunnion}@ucd.ie

Abstract. Information Retrieval (IR) forms the basis of many information management tasks. Information management itself has become an extremely important area as the amount of electronically available information increases dramatically. There are numerous methods of performing the IR task both by utilising different techniques and through using different representations of the information available to us. It has been shown that some algorithms outperform others on certain tasks. Very little progress has been made in fusing various techniques to improve the overall retrieval performance of a system. This paper introduces a Probability-Based Fusion technique *probFuse* which shows initial promise in addressing this question. It also compares *probFuse* with the common CombMNZ data fusion technique.

1 Introduction

Numerous Information Retrieval models have been proposed to solve the problem of identifying documents in a collection that are relevant to given queries. In recent years, much research has been conducted into what has become known as *data fusion* or *collection fusion* [1]. Data fusion involves the combination of results from different sources, using any information that is available, in order to obtain results which are superior to those of any of the individual sources.

In order to achieve this, a number of solutions have been proposed to achieve high-performance data fusion. Some of these rely on the relevance scores provided by the individual retrieval sources, some make use of the ranking of the individual result sets alone and others introduce weighting to create a bias to favour some sources over others. In many cases, such research has been in the context of metasearch engines [1], which involve the fusion of result sets produced by distinct, autonomous IR systems.

This paper is organised as follows: in section 2, we provide a brief overview of some of the approaches that have been taken by others in solving the data fusion problem in the past. Section 3 details the problem in question. In section 4 we

introduce the *probFuse* algorithm, a probability-based approach to data fusion. Section 5 describes the results of running *probFuse* on a number of collections, along with a comparison with the popular CombMNZ fusion technique. Finally, section 6 deals with conclusions and future work.

2 Prior Work

An early, simple method of merging distinct result sets is to interleave the results in round-robin fashion [1], whereby the first-ranked documents are placed at the beginning of the merged set, followed by the second-ranked documents and so on. The effectiveness of this method is largely dependent on the rather naive assumption that each server returns results of equal quality and an empirical study [2] demonstrates a 40% degradation in effectiveness when compared to the performance of a single centralised collection.

A number of later approaches rely on the relevance scores assigned by each retrieval technique to each document in order to rank those documents appropriately [3] [4]. The relevance scores returned by each IR model are not necessarily comparable in their raw form, since each will typically return scores in different ranges. In order to compare these scores in a meaningful way, it is necessary to normalise them, so that they lie within a common range.

A number of fusion techniques based on normalised scores were proposed by Shaw and Fox [5]. These included CombSUM, in which the ranking score for each document is the sum of the normalised scores returned by the individual techniques, and its variant CombMNZ, which introduces a bias in favour of documents which are judged relevant by a higher number of individual techniques. CombMNZ has become the standard data fusion technique [6] [7], as it has been shown to outperform the other techniques they proposed. In particular, Lee [8] was able to achieve significant improvements by using CombMNZ.

A Linear Combination model has been used in a number of studies [9] [10]. Here, each individual source is assigned a weight, based on past performance. Each document's ranking score is then calculated based both on this weight and the estimation of relevance it receives from each source. Vogt and Cottrell made use of training methods to find optimal values for these weights.

Another training-based technique is proposed by Voorhees et. al. in [2]. For each query, they assigned a weight to each seperate collection based on the prior performance of clusters of similar queries. This allowed them to select more documents from the result set returned by the collection with the highest weighting.

Montague and Aslam have developed the *Borda* [7] and *Condorcet* [11] voting-based fusion techniques. They make use of two algorithms that were developed in the 18th century to address shortcomings in the straight vote system for elections in where there were more than two candidates. Applying these algorithms

to fusion they were able to achieve improved results using the document rankings alone, ignoring estimations of relevance returned by the underlying sources. They also produced a weighted variation of each technique, which, like other weighted techniques, uses training data on past performance to calculate the appropriate weights.

Beitzel et. al. [6] argue that the task of fusing result sets from different techniques within the same system is different to the meta search task. They claim that CombMNZ's effectiveness is largely attributable to differences between the autonomous IR systems, such as different stopword lists, different stemming algorithms and relevance feedback. In addition, they argued that Lee's improvements were likely to have arisen because of an increase in overall recall, given that his approach was specifically designed to retrieve documents of different types. Therefore, they claim that CombMNZ's use for fusing result sets produced by the same IR system is limited.

3 Problem Description

The characteristics of fusion are outlined by Vogt and Cottrell [9]. If the individual sources are retrieving different documents, this is likely to increase recall (the fraction of total relevant documents that have been retrieved). They describe this as the "Skimming Effect", as a fusion technique would "skim" the top-ranked documents from each result set, since the highest density of relevant documents is most likely to appear there. They also describe the "Chorus Effect", in which several retrieval sources are in agreement that a document is relevant. In situations where this agreement is correct, fusion techniques which attach a greater significance to documents which are common to multiple sources will perform well. This has been shown to have a significant effect by the research involving the CombMNZ algorithm.

They also identify a "Dark Horse Effect", in which one retrieval approach returns results of a much different quality than the others. This may either be the returning of unusually accurate or inaccurate relevance judgments. Vogt and Cottrell note that the Chorus and Dark Horse effects are somewhat contradictory in nature, with the former encouraging fusion techniques to take as many sources into account when fusing and the latter suggesting that a single technique may provide the best performance.

If we have a system in which we use multiple IR models, it is likely that different models will perform better on different queries. In addition, it is unlikely to be possible to identify which technique will produce the best performance on any specific query. For these reasons, it is desirable to be able to combine the results returned by each model in order to achieve results that are superior to any of the individual techniques. An acceptable minimum performance level would be to match the best performing technique for each query. When evaluating our *probFuse* algorithm in section 5, we use the maximum precision achieved by any

single technique at each point of recall as the benchmark to be improved upon. An ability to improve upon this benchmark supports the case in favour of fusion, rather than merely creating an algorithm to attempt to select the best individual technique for a given query.

4 Probability-Based Fusion

In this section, we describe *probFuse*, a probability-based approach to fusing results from different Information Retrieval models within the same system. Using this approach, each document contained in any of the individual result sets to be fused is assigned a score, based on its probability of relevance, which is used in ranking the documents in the final, fused result set.

In order to calculate this probability, each result set is divided into x segments. Using a training set comprising $t\%$ of the queries available, the probability of relevance for each segment must be calculated.

In a training set of Q queries, $P(d_k|m)$, the probability that a document d returned in segment k is relevant, given that it has been returned by retrieval model m, is given by:

$$P(d_k|m) = \frac{\sum_{q=1}^{Q} \frac{|R_{k,q}|}{|k|}}{Q} \tag{1}$$

where $|R_{k,q}|$ is the number of documents in segment k that are relevant to query q, and $|k|$ is the total number of documents in segment k.

This probabilty should be calculated for each segment in each retrieval model.

The ranking score S_d for each document d is given by

$$S_d = \sum_{m=1}^{M} \frac{P(d_k|m)}{k} \tag{2}$$

where M is the number of retrieval models being used, $P(d_k, m)$ is the probability of relevance for a document d_k that has been returned in segment k in retrieval model m, and k is the segment that d appears in (1 for the first segment, 2 for the second, etc.). For any technique that does not return document d in its result set at all, $P(d_k|m)$ is considered to be zero, in order to ensure that documents do not receive any boost to their ranking scores from techniques which do not return them as being judged relevant.

Using the segment a document is returned in, rather than the specific rank, recognises that different queries will likely result in result sets of varying lengths, depending on how common the terms in the query are. For example, a document ranked 10th in a 20-document result set is less likely to be relevant than the 10th in a 200-document result set.

This approach strives to balance the three effects identified by Vogt and Cottrell. Firstly, by considering the probability of relevance, we make use of the Dark Horse effect, by attaching a greater importance to techniques which are more likely to return relevant documents in particular segments. By using the sum of the scores from each individual technique, rather than the maximum, we make make use of the Chorus effect. Finally, the division by k attaches a greater weight to documents returned near the beginning of the result set, where retrieval techniques will typically have their highest density of relevant documents (Skimming Effect).

5 Experiment and Evaluation

In this section, we describe a number of experiments which were run in order to test the effectiveness of the *probFuse* algorithm. Firstly, we use various training set sizes and x values (the number of segments each result set should be divided into) in order to find optimal values for each. Once these have been identified, we compare the results with that of Shaw and Fox's CombMNZ algorithm.

The experiments were run over four document collections: Cranfield, LISA, NPL and Med. The characteristics of each collection are outlined in Table 1. Initially, the queries for each collection were arranged in a random order. Once this was done, this order was maintained for each experimental run, in order to eliminate inconsistencies of results due to a change in the ordering of the queries. We then obtained the result sets to be fused using three Information Retrieval models: the Vector Space Model [12], the Extended Boolean Model [13] and the Fuzzy Set Model [14]. We then ran *probFuse* on each, using various training set sizes and x values.

Collection	Documents	Queries
Cranfield	1,400	225
LISA	5,872	35
Med	1,033	30
NPL	11,429	93

Table 1. Characteristics of Document Collections Used

The training set sizes used ranged from 10% to 90% inclusive, in intervals of 10 percentage points. For each of those training set sizes, we ran *probFuse* with x values of 2, 4, 6, 8, 10, 20, 30, 40 and 50.

In order to evaluate the performance of our experiments, we firstly calculated the interpolated precision at the 11 standard recall levels [14] (0% to 100% inclusive, at intervals of 10 percentage points) for the result set returned for each document collection by each individual retrieval model and also for the

fused result set. Once this is done, $\overline{\Delta P_c}$, the mean difference in precision for collection c is given by

$$\overline{\Delta P_c} = \frac{\sum_{r=1}^{R} P_{f,r} - MAX(P_{c,r})}{R} \tag{3}$$

where R is the number of standard recall levels, $P_{f,r}$ is the precision of the fused result set at recall level r and $MAX(P_{c,r})$ is the maximum precision obtained by any single retrieval model on collection c at recall level r. The single value used in Figures 1 and 2 is the average $\overline{\Delta P_c}$ across all four collections.

Figure 1 shows the change in average precision for the various values of x and t with each line representing a particular training set size. The poorest-performing training set sizes are 10% and 90%, demonstrating that training set sizes that are either very large or very small will lead to poor performance. Using a training set size of 50% results in the best performance for all but one value of x.

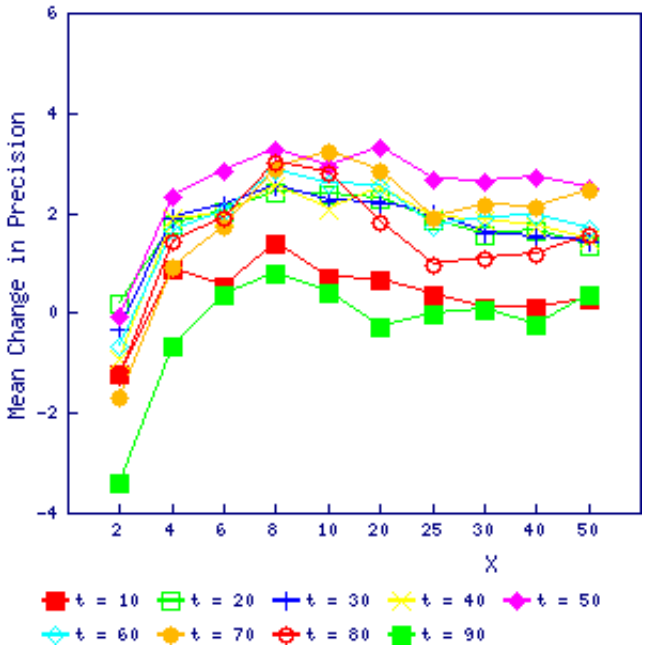


Fig. 1. Mean difference in precision for different training set sizes

In Figure 2, each line represents the change in average precision for a particular value of x. The worst-performing x value is 2. At this value, probability of relevance is assigned to a document based on whether it appears in the first

half or the second half of a result set. Increasing values for x produce superior results, to a point, with x values of 10 and 20 showing the highest mean precision increase.

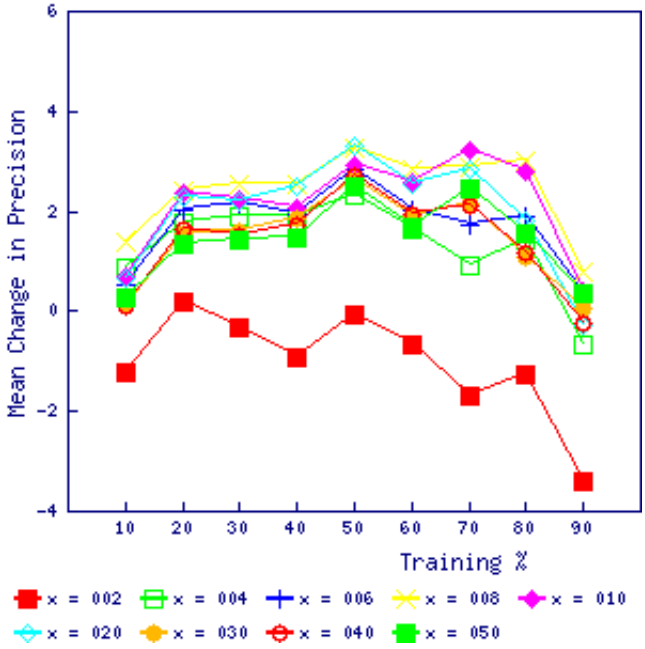


Fig. 2. Mean difference in precision for different values of x

From these two graphs, we can see that the best performance is achieved using a training set size of 50% and dividing each result set into 20 segments.

Having identified the best perfoming combination of x and t values, we then performed a comparison of those results and the CombMNZ algorithm. The CombMNZ algorithm is based on the relevance scores assigned to each document by each retrieval model. However, the raw scores returned by each model are not necessarily directly comparable, so it is necessary to normalise them. Lee's implementation of CombMNZ normalised scores using

$$normalised\_sim = \frac{unnormalised\_sim - min\_sim}{max\_sim - min\_sim} \tag{4}$$

where max_sim and min_sim are the maximum and minimum score, respectively, that are actually seen in the retrieval result. Once the scores have been normalised, the $CombMNZ_d$, the CombMNZ ranking score for any document d is given by

$$CombMNZ_d = \sum_{s=1}^{S} N_{s,d} * |N_d > 0| \quad (5)$$

where S is the number of result sets to be fused, $N_{s,d}$ is the normalised score of document d in result set s and $|N_d > 0|$ is the number of non-zero normalised scores given to d by any result set.

	probFuse	CombMNZ
Cranfield	+1.92**	-1.48*
LISA	+3.09**	+2.24
Med	+3.48	+3.07
NPL	+4.80**	+4.13**
Max	+4.80	+4.13
Min	+1.92	-1.48
Avg	+3.32	+1.99

Table 2. Comparison of the mean difference in precision achieved by the *probMerge* and CombMNZ algorithms for each collection. Entries with a "*" are significant for a significance level of 5%. Entries with a "**" are significant for a significance level of 1%, as calculated by the Wilcoxon test

Table 2 shows a comparison in the mean difference in precision for *probFuse* and CombMNZ, where *probFuse* uses a training set of 50% and an x value of 20. As the first half of the collection is being used solely as training data by *probFuse*, we have ignored it for the purposes of CombMNZ, so that we are comparing the two algorithms' performance over the same queries. The table shows the mean difference in precision both for each collection individually and as an overall average. The table shows us that *probFuse* outperforms CombMNZ on each collection, and that the use of CombMNZ actually causes a significant reduction in performance when applied to the Cranfield collection. For all collections except Med, *probFuse* shows highly significant improvements over the maximum precision values of the individual techniques. In contrast, CombMNZ only achieves significant improvements for the NPL collection.

Figure 3 illustrates the performance of *probFuse* and CombMNZ on the Cranfield collection. It shows the interpolated precision at the standard recall levels for each individual technique, as well as for each fusion technique.

6 Conclusions and Future Work

In this paper, we have proposed a new data fusion technique, *probFuse*. Using this algorithm, documents are ranked based on their probability of relevance. In experiments on small collections, *probFuse* shows initial promise, outperforming

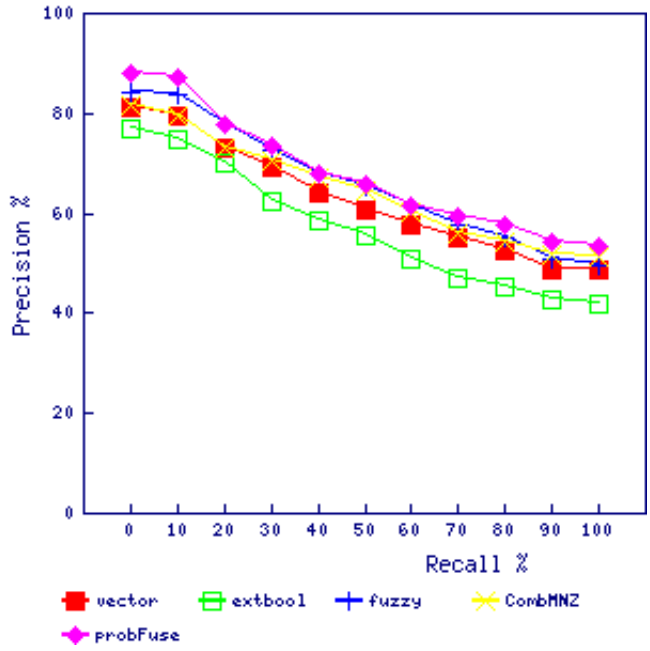


Fig. 3. Interpolated Precision graph for the Cranfield Collection

the best performance of any of the individual retrieval models that we used, namely the Vector Space Model, the Fuzzy Set Model and the Extended Boolean Model. It also was shown to produce superior results to the popular CombMNZ algorithm.

While *probFuse* shows promise on these small collections, it remains to be seen whether the increase in retrieval effectiveness achieved on small collections can be replicated on larger document collections, such as data from the Text REtrieval Conferences (TREC), which is widely used to evaluate fusion techniques.

References

1. Voorhees, E.M., Gupta, N.K., Johnson-Laird, B.: The collection fusion problem. In: Proceedings of the Third Text REtrieval Conference. (1994)
2. Voorhees, E.M., Gupta, N.K., Johnson-Laird, B.: Learning collection fusion strategies. In: SIGIR '95: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (1995) 172–179
3. Selberg, E., Etzioni, O.: The MetaCrawler architecture for resource aggregation on the Web. IEEE Expert (1997) 11–14

4. Howe, A.E., Dreilinger, D.: SAVVYSEARCH: A metasearch engine that learns which search engines to query. AI Magazine **18** (1997) 19–25
5. Shaw, J.A., Fox, E.A.: Combination of multiple searches. In: Proceedings of the 2nd Text REtrieval Conference. (1994) 243–252
6. Beitzel, S.M., Jensen, E.C., Chowdhury, A., Grossman, D., Frieder, O., Goharian, N.: Fusion of effective retrieval strategies in the same information retrieval system. J. Am. Soc. Inf. Sci. Technol. **55** (2004) 859–868
7. Montague, M., Aslam, J.A.: Condorcet fusion for improved retrieval. In: CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management, New York, NY, USA, ACM Press (2002) 538–548
8. Lee, J.H.: Analyses of multiple evidence combination. In: SIGIR '97: Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (1997) 267–276
9. Vogt, C.C., Cottrell, G.W.: Fusion via a linear combination of scores. Inf. Retr. **1** (1999) 151–173
10. Bartell, B.T., Cottrell, G.W., Belew, R.K.: Automatic combination of multiple ranked retrieval systems. In: SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, Springer-Verlag New York, Inc. (1994) 173–181
11. Aslam, J.A., Montague, M.: Models for metasearch. In: SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (2001) 276–284
12. Salton, G., Lesk, M.E.: Computer evaluation of indexing and text processing. J. ACM **15** (1968) 8–36
13. Salton, G., Fox, E.A., Wu, H.: Extended boolean information retrieval. Commun. ACM **26** (1983) 1022–1036
14. Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1999)

Natural Language and Intelligent Multi-Media

Integrating Visual and Linguistic Salience for Reference Resolution

John Kelleher

German Research Center for
Artificial Intelligence (DFKI)
Saarbrücken, Germany
kelleher@dfki.de

Abstract. This paper presents a reference resolution framework for visually situated discourse. The approach taken positions salience, both visual and linguistic, as the fundamental factor in reference resolution. Essentially, resolving a linguistic reference involves modifying the spread of salience across the set of objects in the context to reflect the preferences encoded in the reference and then selecting the most salient element in the resulting context.

1 Introduction

The goal of the CoSy[1] project is to develop an architecture for an embodied cognitive agent that can interact through natural language. The LIVE framework [1] has been adopted as a testbed and prototyping tool for parts of the developing architecture. The LIVE framework provides a virtual reality simulation with a natural language interface. The work presented in this paper investigated how the agent should resolve visually situated references. For example, using Figure 1 as the visual context how should the system interpret *the red house* in (1) and *the tree to the left of it* in (2). Moreover, how do these interpretation processes relate to each other. The reference *it* presupposes a linguistic context and cannot be resolved without knowledge of it, while *the red house* and *the tree to the left of it* both denote objects that have not previously been referred to.

1. make the red house green
2. make the tree to the left of it bigger

Overview §2 reviews previous work on visual and linguistic salience. §3 presents the data structures and the algorithms used by the frameowork. §4 illustrates the functioning of the framework with a worked example. The paper finishes with conclusions.

[1] EU FP6 IST Cognitive Systems Integrated project Cognitive Systems for Cognitive Assistants (CoSy) FP6-004250-IP.

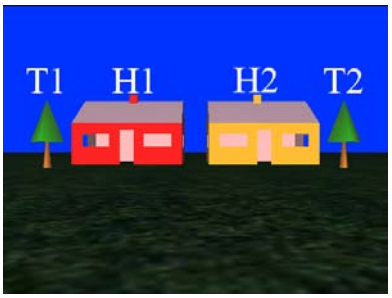


Fig. 1. Visual context. H1 = red, H2 = green.

Fig. 2. Resulting visual context.

2 Related Work

2.1 Modelling Visual Salience

Studies of visual attention, for example [2–5], indicate that both bottom-up and top-down processes influence visual attention. Bottom-up processing guides visual attention based on image-based low-level cues and gaze. Top-down processing is driven by factors such as agent intention. Indeed, the results of several eye-tracking experiments, for example [6, 3, 7], indicate that language comprehension is one of the top-down processes influencing visual attention.

Most previous computational models of visual attention focus on bottom-up processing, see [8] and [9] for recent reviews. In most of these models of attention several feature maps (such as colour, intensity etc.) are computed in parallel across the visual field and these are then combined into a single saliency map. Then a selection process deploys attention to locations in decreasing order of salience. In [10] a simple bottom-up model of visual attention that worked in simulated 3D environments was presented. This paper adapts this model and uses it to model basic visual salience.

2.2 Linguistic Salience

There is a substantial research literature on the topic of linguistic salience. Some of this work has formulated linguistic salience in terms of **hierarchical recency** [11–13]. These hierarchical models of discourse structure maintain a tree structure representation of the discourse. The representation of a previous utterance is hierarchically recent to the current utterance if it is adjacent to the current utterance within this tree structure. However, **linear recency** (i.e. recency of mention) is the fundamental factor influencing linguistic salience and, several framework have been proposed that model linguistic salience in terms of linear recency [14–18]. To a first approximation the basic idea underlying these methods is similar: among the available potential referents, the more recently a referent has been mentioned the more salient it is in the linguistic discourse.

3 Approach

The approach to reference resolution proposed in this paper involves modifying the spread of salience across the set of objects in the context and then selecting the most salient element in the resulting context. This approach treates the semantics of a linguistic reference as a set of instructions that specifies how the spread of salience across the set of objects within the context should be modified before the selection of the referent. These instructions take the form of selectional preferences encoded within the reference. For example, the definite noun phrase *the red house* informs the hearer that they should increase the salience of all the red objects and all the objects of type house in their model of the joint attention context.

The framework distinguishes three levels of salience for each object:

1. basic visual salience and linguistic salience,
2. modality specific reference relative visual and linguistic salience,
3. cross-modal reference relative salience.

Level 1 models the prominence of objects due to bottom-up visual cues and previous discourse. Level 2 models the prominence of an object relative to the task of interpreting a particular reference. These reference based saliency scores are calculated for each object by combining its basic level salience in each modality with a rating of how well the object fulfills the selectional preferences encoded in the reference. Level 3 represents the objects overall salience with respect to the reference. This is computed using a weighted combination of the object's level 2 salience scores. The weights used in this combination reflect the biasing associated with different forms of reference towards a particular information source. The flow of information during reference resolution is from level 1 to level 3. Algorithm 1 lists the basic steps in reference resolution. In the following sections we describe the data structures and algorithms used by the framework to maintain a model of the evolving context and the algorithms used during reference resolution.

Algorithm 1 Reference Resolution Algorithm

For each object in the context compute its modality specific reference relative saliences.
For each object in the context compute its overall reference relative salience.
Select the object with the highest overall salience as the referent.

3.1 Data Structures

The data structure used by the framework is called an **equivalence class**. An equivalence class stores the saliency information for one object in the context model. Figure 3 illustrates the internal structure of an equivalence class. The equivalence class `id` is a string identifier that is created by the vision processing

when it first detects an object. The other components of the data structure reflect the 3 levels of salience distinguished by the framework. The visual and linguistic salience components provide variables to store the object's modality specific level 1 (basic) and level 2 (reference-relative) salience scores. The integrated salience component stores the object's level 3 cross-modal reference relative salience.

New equivalence classes are added to the context model as a result of visual processing. Each time an object is detected in the visual scene the context model is queried for the equivalence class representing the object. If there is no equivalence class for the object in the context model a new equivalence class is created and is assigned the id used by the vision processing. The basic visual salience component is initialised to the value created by vision processing when the object was detected in the scene and is updated after each scene is rendered. All the other components are initialised to 0 and are updated after each reference has been processed. Equivalence classes are removed from the context model when both their basic visual and linguistic saliences fall below a threshold (.0001). In the following sections we describe the algorithms that provide and use the information stored in these structures.

1. id = String value
2. visual salience
   (a) basic = $[0 \dots 1]$
   (b) reference-relative = $[0 \dots 1]$
3. lingustic salience
   (a) basic = $[0 \dots 1]$
   (b) reference-relative = $[0 \dots 1]$
4. integrated salience = $[0 \dots 1]$

Fig. 3. An Equivalence Class

3.2 Computing Basic Visual Salience

The basic visual saliency algorithm uses a **false-colouring** technique. Each object in the simulation is assigned a unique colour or **vision-id**. This colour differs from the normal colours used to render the object in the world; hence the term false colouring. Each frame is rendered twice: firstly, using the objects' normal colours, textures and shading, and secondly, using the vision-ids. The first rendering is on screen (i.e. the user sees it), the second rendering may be off screen. After each frame is rendered, a bitmap image of the false colour rendering is created. The bitmap is then scanned and a list of the colours in the image is created. Using this list the system can recognise which objects are visible and which are not. Moreover, the system can identify, at the pixel level, the area covered by each object in the scene. This pixel information is used to compute the basic visual saliency of each object.

Mimicking the spread of visual acuity across the retina, the algorithm weights each pixel in the image based on its distance from the point of visual focus. The weighting is computed using Equation 1. In this equation, D equals the distance between the pixel being weighted and the point of focus, M equals the maximum distance between the point of focus and any point on the border of the image. The point of focus can be determined using eye tracking technology to compute the user's gaze at each scene rendering. However, if eye tracking is not being used the point of focus defaults to the center of the image or to the center of silhouette of the last object referred to. Algorithm 2 lists the procedure used to compute basic visual saliency and to update the equivalence classes.

$$Weighting = 1 - \left(\frac{D}{M+1}\right) \tag{1}$$

Algorithm 2 The basic visual salience algorithm.

for each object O_i in the scene **do**
  $AW(O_i)$ = average weighting of the pixels covered by O_i
  $Total_{AW} = Total_{AW} + AW(O_i)$
end for
for each equivalence class EC_i in the context model **do**
  if EC_i is the equivalence class representing O_i **then**
    $EC_i.basic\_visual = (EC_i.basic\_visual/2) + (AW(O_i)/Total_{AW})$
  else
    $EC_i.basic\_visual = EC_i.basic\_visual/2$
  end if
  $Total_{bvs} = Total_{bvs} + EC_i.basic\_visual\_salience$
end for
for each equivalence class EC_i in the context model **do**
  $EC_i.basic\_visual = EC_i.basic\_visual/Total_{bvs}$
end for

3.3 Computing Basic Linguistic Salience

In §2.2, we noted the distinction between hierarchical and linear recency within discourse and introduced the basic idea underlying the models of linear recency. In this section we adapt the linear recency salience algorithm presented in [18]. The algorithm is based on the ranking of the so-called **forward looking centers** (C_f) of an utterance. The set of forward looking centers of an utterance contains the objects referred to in that utterance. This set is partially ordered to reflect the relative prominence of the referring expressions within the utterance. Grammatical roles are a major factor here, so that `subject` > `object` > `other`. The central component of the algorithm is a function sf that maps the objects in a domain D to the set $\{0, \ldots, 1\}$, with the intuition that 0 represents complete non-salience and 1 maximal salience. Algorithm 3 defines the salience function sf. The algorithm assumes that in the initial situation s_0 all the objects in the domain are equally (not) salient: $sf(s_0, d) = 0$ for all $d \in D$.

It is not claimed that the function sf is the best way to assign linguistic salience. However, it does provide an operational and, for this discussion, reasonable model of linguistic salience. Algorithm 4 defines the procedure used to update the basic linguistic saliences after each utterance has been processed.

Algorithm 3 Linguistic Salience Weight Assignment

Let U_i be a sentence uttered in state s_i, in which reference is made to $\{d_i, \ldots, d_n\} \subseteq D$. Let $C_f(U_i)$ (the f orward looking center of U_i) be a partial order defined over $\{d_i, \ldots, d_n\} \subseteq D$. Then the salience weight of objects in $s_{\ i+1}$ is determined as follows:

$$sf(s_{i+1}, d) = \begin{cases} (sf(s_i, d)/2) + 1 & \text{if } d = subject(U_i) \\ (sf(s_i, d)/2) + .5 & \text{if } d = object(U_i) \\ (sf(s_i, d)/2) + .25 & \text{if } d = other(U_i) \\ sf(s_i, d)/2) & \text{if } d \notin C_f(U_i) \end{cases}$$

Algorithm 4 The basic linguistic salience algorithm

for each equivalence class EC_i in the context model **do**
  $EC_i.basic\_linguistic = sf(s_j, EC_i)$
  $Total_{DS} = Total_{DS} + EC_i.basic\_linguistic$
end for
for each equivalence class EC_i in the context model **do**
  $EC_i.basic\_linguistic = EC_i.basic\_linguistic/Total_{DS}$
end for

3.4 Computing Reference Relative Saliences

The reference relative saliences for each object are computed by integrating the object's basic visual and linguistic saliences with a rating of how well the object fulfils the selectional preferences encoded in the description. This rating is called an **f-score**. Two f-scores are computed for each object for each reference. A visual and a linguistic f-score. These f-scores are computed by integrating the ratings of how well the object fulfils each of the selectional preferences included in the referring expression. Currently, the system can rate objects relative to their type, colour, size[2] and location[3]. Table 1 lists the ratings ascribed to an object for each type of selectional preference.

Table 1. Selectional Preferences Scores

	TYPE	COLOR	SIZE	LOCATION
Fulfills	1	1	[1 . . . 0]	[1 . . . 0]
Not Fulfils	0	0		

[2] An objects size rating is based on the number of pixels it covers relative to the other objects in the scene.
[3] An objects location rating is computed using the AVS model described in [19]

The motivation for computing two f-scores is the observation that the processing of a linguistic reference influences the visual and linguistic salience of the objects that partially fulfil the description in different ways. For example, when searching a scene containing different colour shapes for *the yellow box on the right* the visual prominence of all the yellow boxes in the scene increases even if they are not located in the correct region. However, in general, an object will not be considered as possible referent for a given referring expression unless it fulfils all of the description provided in the expression (i.e. its linguistic salience relative to the referring expression will be 0) even if it is the most prominent object within the prior discourse. The framework reflects this difference by integrating the individual rating scores an object achieves differently in the computation of the visual and linguistic f-scores. An object's visual f-score is initialised to 0 and its ratings are integrated using addition. An object's linguistic f-score is initialised to 1 and its ratings are integrated using multiplication.

Once an object's f-scores have been computed its reference relative visual and linguistic saliences are computed by integrating its f-scores with its basic visual and linguistic salience. This same operator used in the computation of the f-score in a particular modality is used to integration the f-score with the object's basic salience in that modality. As a result, an object's reference relative visual salience will be greater than 0 if it fulfils any of the selectional preferences in the description, and its linguistic reference relative salience will equal 0 if it does not fulfil all of the selectional preferences in the description. Algorithm 5 lists the algorithm for computing the reference relative saliences.

Algorithm 5 Computing the reference relative saliences.

for each equivalence class EC_i in the context model **do**
  $f\_score_{linguistic} = 1$
  $f\_score_{vision} = 0$
  for each selectional preference sp_j in the description **do**
    $f\_score_{linguistic} = f\_score_{linguistic} * rating(EC_i, sp_j)$
    $f\_score_{vision} = f\_score_{vision} + rating(EC_i, sp_j)$
  end for
  $EC_i.reference\_language = EC_i.basic\_language * f\_score_{linguistic}$
  $Total_{rls} = Total_{rls} + EC_i.reference\_language$
  $EC_i.reference\_visual = EC_i.basic\_visual * f\_score_{vision}$
  $Total_{rvs} = Total_{rvs} + EC_i.reference\_visual$
end for
for each EC_i in the context model **do**
  $EC_i.reference\_langauge = EC_i.reference\_language/Total_{rls}$
  $EC_i.reference\_visual = EC_i.reference\_visual/Total_{rvs}$
end for

3.5 Computing the Overall Reference Based Salience

The final step before the selection of the referent is the integration of each object's reference based saliences. This is done using a weighted combination. The weightings are dependent on the form of referring expression (e.g. definite

descriptions versus pronominal references) being resolved. The motivation for the weighting is that often the form of referring expression provides an indication of the type of contextual object the reference denotes. For example, in general, a pronoun is not used to refer to an object in the visual scene, no matter how prominent, if the object has not been previously mentioned in the discourse. By contrast, a definite description can be used to refer to both an object from the visual scene and to previously mentioned objects. Currently, the system uses predefined weights for this integration. When resolving a definite description visual and linguistic salience are integrated evenly. When resolving a pronominal reference linguistic salience is preferenced. Algorithm 6 defines the procedure used to integrate the visual and linguistic saliences, and to select the reference.

Algorithm 6 Integrating the reference based saliences and selecting the references.

for each equivalence class EC_i in the context model **do**
  Let index = 0, max = 0
  if reference = definite description **then**
    $EC_i.integrated = (EC_i.reference\_visual * 0.5) + (EC_i.reference\_language * 0.5)$
  else if reference = pronominal reference **then**
    $EC_i.integrated = (EC_i.reference\_visual * 0.1) + (EC_i.reference\_language * 0.9)$
  end if
  if $EC_i.integrated > max$ **then**
    index = i
    $max = EC_i.integrated$
  end if
end for
textbfreturn EC_{index}

4 Worked Example

We illustrate the functioning of the framework using a worked example, based on the system interaction presented in §1. Table 2 lists the saliences scores computed during the different stages of this interaction. Rows 1 and 2 of the table present the basic visual and linguistic salience scores of the objects in Figure 1 before any commands are input. Rows 3 to 7 presents the f-scores and reference and integrated saliences computed for the objects when the system processed *the red house*. The asterix in line 7, H1's column, indicates the highest integrated salience at the end of the resolution process. As a result of obtaining the maximum salience H1 is selected as the referent. Rows 8 and 9 list the basic salience scores for the objects after the basic linguistic salience has been updated and the point of visual focus has been located at the center of H1's silhouette. The movement of the visual focus away from the center of the image is reflected in the increases in the basic visual salience of H1 (0.3271 → 0.3581) and T1 (0.1728 → 0.2938). Rows 10 to 14 list the f-scores and reference and integrated saliences computed

for the objects when the system processed *it*. The biasing towards linguistic salience is apparent in the dominance of H1's integrated salience. Rows 15 to 19 list the f-scores and reference and integrated saliences computed for the objects when the systems processed *the tree to the left of it*. The difference between the visual and linguistic f-scores of T1 and T2 is due to the locational description: T1 was judged by the system to fulfil the locational description with a rating of 0.9396, while T2 was judged not to fulfil the description and was ascribed a rating of 0.0000 for this selectional preference. As a result, T1 achieved the highest salience (0.5669) and was selected as the referent. Figure 1 illustrates the visual context at the end of the interaction.

Table 2. Salience scores computed during the example interaction.

		H1	H2	T1	T2
Initial Context					
1	Basic visual salience	0.3271	0.3272	0.1728	0.1728
2	Basic linguistic salience	0.0000	0.0000	0.0000	0.0000
the red house					
3	Visual f-score	2.0000	1.0000	0.0000	0.0000
4	Linguistic f-score	1.0000	0.0000	0.0000	0.0000
5	Reference visual salience	0.5818	0.3318	0.0432	0.0432
6	Reference linguistic salience	0.0000	0.0000	0.0000	0.0000
7	Integrated salience	0.5818*	0.3318	0.0432	0.0432
8	Basic visual salience	0.3581	0.2273	0.2938	0.1208
9	Basic linguistic salience	1.0000	0.0000	0.0000	0.0000
it					
10	Visual f-score	0.0000	0.0000	0.0000	0.0000
11	Linguistic f-score	1.0000	1.0000	1.0000	1.0000
12	Reference visual salience	0.3581	0.2273	0.2938	0.1208
13	Reference linguistic salience	1.0000	0.0000	0.0000	0.0000
14	Integrated salience	0.9358*	0.0227	0.0293	0.0120
the tree to the left of it					
15	Visual f-score	0.0000	0.0000	1.9396	1
16	Linguistic f-score	0.0000	0.0000	0.9396	0
17	Reference visual salience	0.0909	0.0577	0.5669	0.2845
18	Reference linguistic salience	0.0000	0.0000	0.0000	0.0000
19	Integrated salience	0.0909	0.0577	0.5669*	0.2845

5 Conclusions

This paper presented a salience based reference resolution framework for visually situated discourse and illustrated how the framework resolves references both to objects in the visual context that have not been mentioned previously and to objects in the linguistic context. The framework distinguishes between an object's basic salience and its salience relative to the context provided by a particular referring expression. A key element of the framework is the weighted integration of saliences from different modalities. The weightings used reflect the contextual preferences associated with different forms of referring expressions.

References

1. Kelleher, J., Costello, F., van Genabith, J.: Dynamically updating and interrelating representations of visual and linguistic discourse. Artificial Intelligence Journal (2005 (forthcoming))
2. Enns, J., R.A., R.: Influence of scene-based properties on visual search. Science **247** (1990) 721–723
3. Spivey-Knowlton, M., Tanenhaus, M., Eberhard, K., Sedivy, J.: Integration of visuospatial and linguistic information: Language comprehension in real time and real space. In Olivier, P., Gapp, K., eds.: Representation and Processing of Spatial Expressions. Lawrence Erlbaum Associates (1998) 201–214
4. Hopfinger, J., Buonocore, M., Mangun, G.: The neural mechanisms of top-down attentional control. Nature Neuroscience **3** (2000) 284–291
5. Chum, M., Wolfe, J.: Visual attention. In Goldstein, E.B., ed.: Blackwell Handbook of Perception. Handbooks of Experimental Psychology. Blackwell (2001) 272–310
6. Yarbus, A.: Eye Movements and Vision. New York: Plenium Press (1967)
7. Tanenhaus, M., Spivey-Knowlton, M., Eberhard, K., Spivey, J.: Integration of visual and linguistic information in spoken language comprehension. Science **268** (1995) 1632–1634
8. Koch, C., Itti, L.: Computational modelling of visual attention. Nature Reviews Neuroscience **2** (2001) 194–203
9. Heinke, D., Humphreys, G.: Computational models of visual selective attention: A review. In Houghton, G., ed.: Connectionist Models in Psychology. Psychology Press (2004)
10. Kelleher, J., van Genabith, J.: Visual salience and reference resolution in simulated 3d environments. AI Review **21** (2004) 253–267
11. Hobbs, J.: On the coherence and structure of discourse. Technical Report CSLI-85-37, Center for the Study of Language and Information (1985)
12. Grosz, B., Sidner, C.: Attention, intentions, and the structure of discourse. Computational Linguistics **12** (1986) 175–204
13. Mann, W., Thompson, S.: Rhetorical structure theory: Description and construction of text structures. In Kempen, G., ed.: Natural Language Generation: New Results in Artificial Intelligence, Psychology and Linguistics. Nijhoff., Dordrecht (1987) 83–96
14. Alshawi, H.: Memory and Context for Language Interpretation. Cambridge University Press, Cambridge, UK (1987)
15. Hajicová, E.: Issues of Sentence Structure and Discourse Patterns. Volume 2 of Theoretical and Computational Linguistics. Charles University Press (1993)
16. Lappin, S., Leass, H.: An algorithm for pronominal anaphora resolution. Computational Linguistics **20** (1994) 535–561
17. Grosz, B., Joshi, A., Weinstein, W.: Centering: A framework for modelling local coherence of discourse. Computational Linguistics **21** (1995) 203–255
18. Krahmer, E., Theune, M.: Efficient context-sensitive generation of referring expressions. In van Deemter, K., Kibble, R., eds.: Information Sharing: Reference and Presupposition in Language Generation and Interpretation. CLSI Publications, Standford (2002)
19. Regier, T., Carlson, L.: Grounding spatial language in perception: An empirical and computational investigation. Journal of Experimental Psychology: General **130** (2001) 273–298

Presenting Temporal Relations of Virtual Human Actions by Multiple Animation Channels

Minhua Ma and Paul Mc Kevitt

School of Computing & Intelligent Systems
Faculty of Engineering
University of Ulster, Magee
Derry/Londonderry, BT48 7JL, Northern Ireland
{m.ma, p.mckevitt}@ulster.ac.uk

Abstract. This paper proposes a possible approach to present various temporal relations of virtual human actions. It combines precreated and dynamically generated (procedural) animation facilities into a unified mechanism, and focusses on controlling simultaneous animations by multiple animation channels. We present our work on language visualisation (animation) in our intelligent multimodal storytelling system, CONFUCIUS, and describe how the proposed approach is employed in CONFUCIUS' animation engine. This approach allows the intelligent storytelling to take advantage of procedural animation effects in the same manner as regular animations, adding an additional level of flexibility and control when animating virtual characters.

1 Introduction

Simulating human motion and behavior by computer is an active and challenging area. Existing virtual human animations are either controlled by precreated animations [2] (e.g. hand-animated using authoring tools like 3D Studio Max, Maya, and Poser, or motion captured data), or dynamically generated by animation techniques [8] such as inverse kinematics (IK). However, there is a lack of consideration for presenting temporal relations between multiple animation sequences and integrating different human animation sequences to present simultaneous motions. Here, we propose an approach to present various temporal relations of virtual human actions (especially overlapped interval relations) using multiple animation channels. We also present our work on language visualisation (animation) in the intelligent multimodal storytelling system, CONFUCIUS, and show how this approach is employed in CONFUCIUS' animation engine and achieves more flexibility and control on virtual characters' animation.

First, in section 2 we introduce the intelligent multimedia storytelling system, CONFUCIUS and review various techniques of humanoid animation. Next in section 3, the thirteen interval temporal relations, in particular, overlapped relations which indicate simultaneous motions, are introduced and sense of iteration and its presentation are discussed. Then we propose the motion integration approach of using multi-

ple animation channels in section 4. Finally, section 5 compares our work with related work on humanoid animation, and summarizes with a discussion of possible future work on integrating other animation generation techniques such as IK and machine learning.

2 Background

We are developing an intelligent multimedia storytelling interpretation and presentation system called CONFUCIUS. It automatically generates 3D animation and speech from natural language input as shown in Fig. 1. The dashed part in the figure is the knowledge base including language knowledge (lexicons and a syntax parser) which is used in the Natural Language Processing (NLP) module, and visual knowledge such as 3D models of characters, props, and animations of actions, which is used in animation engine. The surface transformer takes natural language sentences as input and manipulates surface text. The NLP module uses language knowledge to parse sentences and analyse their semantics. The media allocator then generates an XML-based specification of the desired multimodal presentation and assigns content to three different media: animation, characters' speech, and narration, e.g. it sends the parts bracketed in quotation marks near a communication verb to the text-to-speech engine. The animation engine accepts semantic representations and uses visual knowledge to generate 3D animations. The animation engine and Text-to-Speech (TTS) operate in parallel. Their outputs are combined in the synchronising module, which outputs a holistic 3D virtual world including animation and speech in VRML. Finally, the narration integration module integrates the VRML file with the presentation agent, Merlin the Narrator, to complete a multimedia story presentation. Currently, CONFUCIUS is able to visualise single sentences which contain action verbs with *visual valency* of up to three, e.g. "John left the gym", "Nancy gave John a loaf of bread" [6].

2.1 Previous Work on Humanoid Animation

Representing humanoid kinematics is a main task in the animation generation model. The kinematic animation techniques vary from a simple application of precreated animation frame data (keyframes, either hand-animated or motion-captured), to a complex on-the-fly inverse kinematics (IK). IK is a system in which the movement of the children is passed back up the chain to the parent in the hierarchical skeleton tree. Given a desired position and orientation for a final link in a hierarchy chain, IK establishes the transformations required for the rest of the chain. Animation is performed by affecting the ends of the chain, e.g. in biped walking animation, by moving the foot and the shin, knees and thighs rotate in response. A good overview of IK techniques can be found in [4]. IK models the flexibility and possible rotations of joints and limbs in 3D creatures. IK adds flexibility which avoids canned motion sequences seen in keyframing animations, and hence enables having an infinitely expandable variety of possible animations available to a virtual character. The character control

file is also reduced to a physical description of the character and a set of behavioural modifiers that are used to alter the flavour of the animations [3].

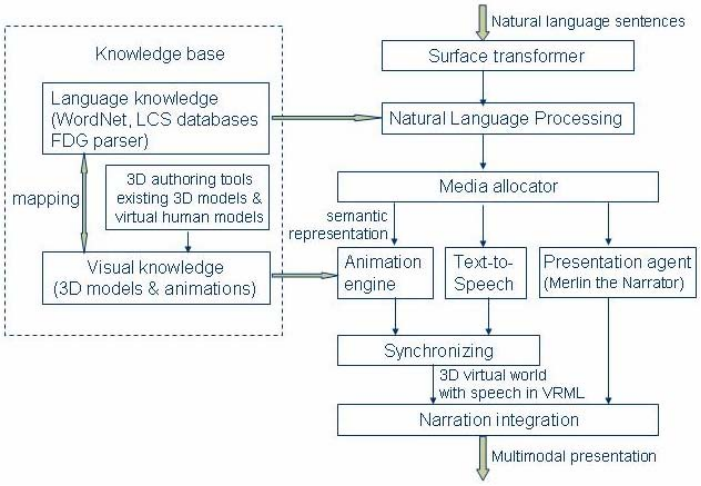


Fig. 1. Architecture of CONFUCIUS

In keyframing animation, animators have to specify explicit definition of the key values of the character's joints at specific time instants, namely "key frames", to generating a motion. Then the key values are interpolated so that in-between frames are generated. CONFUCIUS' animation uses the keyframing technique for virtual human motions. The traditional approach of animating characters (agents/avatars) provides a set of animations from which the user/system can select. In most current graphical chat rooms the user can control his avatar behavior by selecting an animation sequence from a list of available motions. The avatar can only play one animation at a time, i.e. only apply one precreated animation for the entire duration of the animation sequence.

IMPROV [7] uses procedural animation combined with behavioural scripting for creating flexible characters for virtual theatre. IMPROV divides the actions of avatars into a set of groups. The action, in this case, is defined as a single atomic or repetitive activity that does not require explicit higher-level awareness or conscious decisions. Actions within a group are mutually exclusive of one another; activating one causes the action currently active to end. Actions in different groups can operate simultaneously, so activities of certain parts of the body can be layered over those involving others. Using this structure, the basic set of actions can be combined to create dozens of composite animations while minimising the risk of inadvertently creating a behaviour that is either unbelievable or not lifelike. The solution serves as the mechanism for user-controlled avatars by enabling multiple levels of abstraction for the possible actions.

3 Temporal Relations between Simultaneous Animations

Table 1 lists the thirteen Allen temporal relations [1] that are used in visual semantic representation of verbs in CONFUCIUS' language visualisation [8]. Simultaneous animations playing on multiple channels of a virtual human are closely related to the overlapped temporal relations, i.e. the relations 5-13, in the table.

Table 1. Allen's thirteen interval relations. ("e" denotes "end point", "s" denotes "start point".)

Temporal relations		Example	Endpoints	Example sentences
1. precede 2. inverse precede	x p y y p^{-1} x	xxxx yyy	$x_e < y_s$	John left before Mary arrived.
3. meet 4. inverse meet	x m y y m^{-1} x	xxxx yyyy	$x_e = y_s$	All passengers died when the plane crashed into the mountain.
5. overlap 6. inverse overlap	x o y y o^{-1} x	xxxxx yyyyy	$x_s < y_s < x_e$ $\cap\ x_e < y_e$	Mary got up. She felt very ill.
7. during 8. include	x d y y d^{-1} x	xxxx yyyyyyyy	$x_s > y_s \cap$ $x_e < y_e$	John arrived in Boston last Thurs-day.
9. start 10. inverse start	x s y y s^{-1} x	xxxx yyyyyyyy	$x_s = y_s \cap$ $x_e < y_e$	John has lived in Boston since 2000.
11. finish 12. inverse finish	x f y y f^{-1} x	xxx yyyyyy	$x_e = y_e \cap$ $x_s > y_s$	John stayed in Boston till 2000.
13. equal	x ≡ y y ≡ x	xxxxx yyyyy	$x_s = y_s \cap$ $x_e = y_e$	John drove to London. During his drive he listened Classic FM.

3.1 Sense of Iteration and Animation Loops

Iteration is another temporal factor affecting animated characters. Sense of iteration is not encoded in English syntax through it may be added by some prepositional phrases like "for hours", "until midnight", or temporal quantifier such as *twice*, *three times*, *every*, and so forth. Consider, for example, the difference between the two sentences below.

```
John taught two hours every Monday. (iteration)
John taught two hours on Monday.
```

Jackendoff [5] ascribes the sense of iteration to *temporal boundedness*. Table 2 shows some examples of temporal boundedness of events. Temporal bounded events (e.g. Table 2: 2, 3) are also called *punctual events* or *achievement* events (distinct from *accomplishment* events [11]). The prepositional phrases "for hours" and "until midnight" can follow temporally unbounded processes, and place either a measure or a boundary on them. "John slept", for instance, expresses an unbounded event, so it can be felicitously prefixed with these prepositional phrases. But "John waked" expresses a temporally bounded event, so it cannot be further measured or bounded by these prepositional phrases.

Table 2. Temporal boundedness of events

Examples	Temporal Boundedness	Prefixing "for hours" or "until midnight"
1) John slept.	Unbounded	acceptable
2) John waked.	Bounded	not acceptable
3) John entered the house.	Bounded	not acceptable
4) John walked toward the house.	unbounded	acceptable
5) The light flashed.	bounded (repeatable)	acceptable, add the sense of repetition
6) Somebody hammered the door.	bounded (repeatable)	acceptable, enhance the sense of repetition

Some verbs have the sense of repetition included/hinted in their lexical semantics, e.g. Table 2: 5 and 6. Prefixing "for hours" or "until midnight" will add/enhance the sense of repetition to them. However, there is a nuance between 5 and 6. Without those prepositional phrases, "the light flashed" means it flashed once, whereas "Somebody hammered the door" suggests (s)he hammered the door repeatedly. Therefore, "for hours" *adds* the sense of repetition in 5, and *enhances* it in 6. Example 2 and 3 are bounded but unrepeatable, so they cannot give grammatical productions when prefixing "for hours" or "until midnight".

Jackendoff [5] thinks that the operator, which maps a conceptual constituent that encodes a single event into a conceptual constituent that encodes a repeated sequence of individual events of the same type, has the same semantic value as the plural marker, which maps a conceptual constituent that encodes an individual thing into a conceptual constituent that encodes a collection of things of the same type, to wit, the bounded/unbounded distinction in events is strongly parallel to the count/mass distinction in NPs. The criterion for the boundedness and countableness distinction has to do with the description of parts of an entity. For instance, a part of "an apple" (count) cannot itself be described as "an apple", but any part of a body of "water" (mass) can itself be described as "water"; a part of the event "John entered the house" (bounded) cannot itself be described as "John entered the house", but any part of "John walked toward the house" (unbounded) can be described as "John walked toward the house". Therefore, a static graphic scene can only represent unbounded events such as "John walked toward the house" properly, by selecting a representative part of the event; while bounded events are better presented by animation.

Distinction for sense of iteration is very important for visualise events in CONFUCIUS' storytelling since the animation generator needs to know whether it's necessary to repeat an action loop, and whether it's necessary to animate the complete process of an event (a bounded event) or just a part of it (an unbounded event).

CONFUCIUS' semantic representation has a facility to represent repeatable periods of subactivities. Square brackets and a subscript R are used to indicate the repetition constructs in the examples of Fig. 2, which can also be nicely captured by Kleene

iteration in finite state descriptions for temporal semantics. The activities bracketed by $[\ ]_R$ are repeatable. Besides periodical repetition of subactivities, it can represent morphological prefix "re-" as well, as the “recalculate” example in Fig. 2, substituting the number of iterations (which is 2 in this case) for R. This facility of representing iteration may be used for post-lexical level repetition, such as events marked by “again”, “continues to”, or “a second time”.

walk():- [step()]$_R$. hammer(…):- [hit(…,hammer)]$_R$. recalculate():- [calculate()]$_2$.

Fig. 2. Verbs defined by repeatable subactivities

Animation loops are used to present action repetition. This facility indicates whether the played animation should loop. If not specified, the animation will loop, i.e. looping is enabled as default, which is controlled by a time sensor in the VRML file.

4 Animating Virtual Characters

CONFUCIUS uses the H-anim [2] standard for character modelling and animation. H-anim is a VRML97 representation for humanoids. It defines standard human *Joints* articulation (e.g. knee and ankle), *Segments* dimensions (e.g. thigh, calf, and foot), and *Sites* (e.g. hand_tip, foot_tip) for “end effector” and attachment points for clothing. An H-Anim file contains a joint-segment hierarchy as shown in Fig. 3. Each joint node may contain other joint nodes and a segment node that describes the body part associated with the joint. Each segment is a normal VRML transform node describing the body part's geometry and texture. H-Anim humanoids can be animated using keyframing, inverse kinematics (IK), and other animation techniques.

Since our task of language animation in CONFUCIUS focuses on off-line generation, and real-time interaction is never our concern, we adopt the H-anim standard to model the virtual characters in our storytelling. H-anim provides four Levels of Articulation (LOA) for applications which require different *levels of detail*. Some applications such as medical simulation and design evaluation require high fidelity to anthropogeometry and human capabilities, whereas games, training and visualised living communities are more concerned with real-time performance. Storytelling is not usually concerned with accurate simulation of humans. We use the Level 2 of Articulation (LOA2) of H-anim in character modelling for CONFUCIUS. This level ensures enough joints for human movements in storytelling, e.g. it includes enough hand joints for grasp postures. Fig. 3 illustrates the joints of LOA2.

Based on Babski’s [2] animation prototype, we design our virtual characters’ animation which is capable to be applied to various human models with different LOAs. Needed ROUTEs are generated dynamically based on the joint list of the H-anim body and the joint list of the animation. Fig. 4 shows an example external prototype inserted at the end of a virtual character’s H-anim file by the animation engine. It uses keyframe information in the external VRML file “..\animation\walk.wrl” to make the virtual character walk.

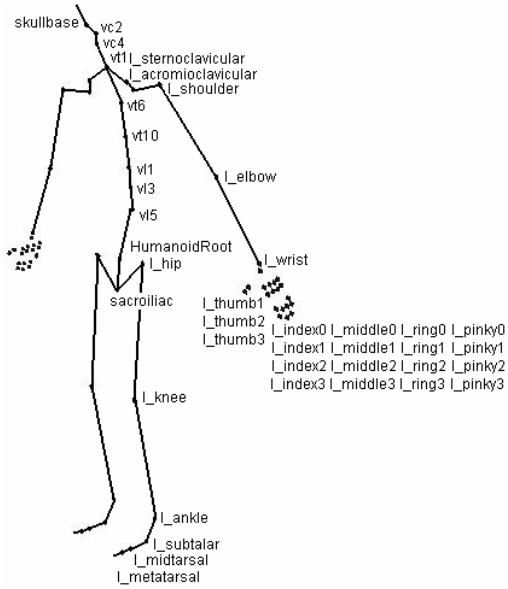


Fig. 3. H-Anim joint hierarchy

```
# -------- inserted by CONFUCIOUS animation engine ---
EXTERNPROTO Behaviour [
      eventIn SFTime LaunchAnim
      exposedField SFTime set_startTime
      exposedField SFTime set_stopTime
      field MFNode HumansList
]"..\animation\walk.wrl"
DEF behv Behaviour {
      HumansList [
            USE humanoid
      ]
}
ROUTE hanim_BodyTouch.touchTime TO behv.LaunchAnim
```

Fig. 4. External prototype of H-anim animation

The animation file defines keyframes of all `OrientationInterpolator` and `PositionInterpolator` involved in the movement. The Script node dynamically add ROUTEs according to the list specified in `InvolvedJointNameList` and `InvolvedJointPtrList` in the animation file. The matching between the animation and the body is performed by using the joints list in the Humanoid prototype. Therefore, `InvolvedJointNameList` must have a one-to-one matching to the humanoid `joints` list defined in the virtual character's geometry file. If the animation is applied to a lower LOA character, e.g. LOA1, and a joint is not implemented, the corresponding field should be a dummy Transform/Joint node.

4.1 Simultaneous Animations and Multiple Animation Channels

Performing simultaneous animations is not a problem for the lower level procedural human animation modeling languages, e.g. VHML [13] (an XML-based human animation language) and STEP [5] (a Prolog-like human animation script language), since they provide a facility to specify both sequential and parallel temporal relations. Fig. 5 shows how VHML and STEP represent the parallel temporal relation. However, simultaneous animations cause the Dining Philosopher's problem for higher level animation using pre-defined animation data, i.e. multiple animations may request to access same body parts at the same time. In order to solve this problem, we introduce the approach of multiple animation channels to control simultaneous animations.

```
<left-calf-flex amount=”medium”>
<right-calf-flex amount=”medium”>
 <left-arm-front amount=“medium">
 <right-arm-front amount=“medium">
Standing on my knees I beg you pardon
 </right-arm-front></left-arm-front>
</right-calf-flex></left-calf-flex>
```

A. A VHML example

```
script(walk_forward_step(Agent),ActionList):-
  ActionList=[parallel(
                [script_action(walk_pose(Agent),
                      move(Agent,front,fast))])].
```

B. A STEP example

Fig. 5. Representing parallel temporal relation

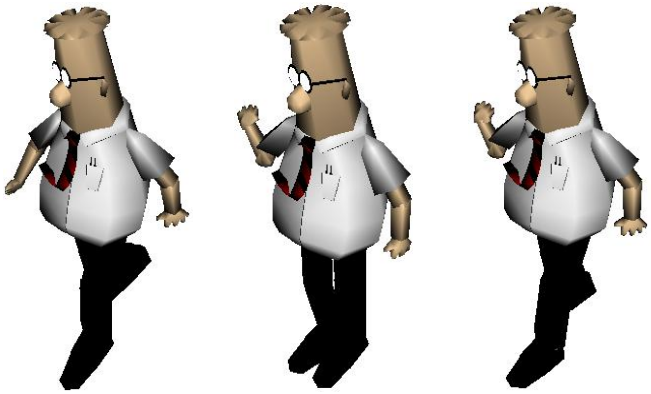
A character that plays only one animation at a time has only a single channel, while a character with upper and lower body channels will have two animations playing at the same time. Multiple animation channels allow characters to run multiple animations at the same time (such as walking with the lower body while waving with the upper body). Multiple animation channels often need to disable one channel when a specific animation is playing on another channel to avoid conflicts with another animation.

We use an animation table as shown in Table 3 to implement multiple animation channels. Every pre-defined animation must register in the animation table and specify which joints are used for the animation. In Table 3, each row represents one animation, and each column represents one joint. 0 indicates that the joint is not used for the animation; 1 indicates that it is used and can be disable when playing simultaneous animations; and 2 means that the joint is used and cannot be disabled. When simultaneous animations are requested, the animation engine checks the animation table and finds if the involved joints of these animations conflict, i.e. if there is any joint whose values for both animations are 2, these animations conflict and they cannot be played at the same time. If two animations do not conflict (for example, “run” and “throw”), the animation engine merges their keyframes information, i.e. interpolaters, and creates a new animation file which will be applied to the virtual character.

Table 3. THE ANIMATION TABLE

Involved joints /Animations	sacroiliac	l_hip	r_hip	…	r_shoulder
walk	2	2	2	…	1
jump	2	2	2	…	1
wave	0	0	0	…	2
run	2	2	2	…	1
scratch head	0	0	0	…	2
sit	2	2	2	…	1
…	…	…	…	…	…

Fig. 6 shows an example of integrating the two animations “walk” and “wave”. The motion of waving only uses three rotation interpolators: r_shoulder, r_elbow, r_wrist. The animation engine looks up the animation table and finds that the walking animation also uses these three joints and their values are all 1, which means the right arm movements of walking can be disabled and overwritten by the movements of waving. The animation engine then replaces the keyframes of these three joints in the walking animation file with those in the waving file and generates an integrated motion.



Walking Waving Walking and waving

Fig. 6. An example of motion integration. The first figure is a snapshot of walking animation, the second is waving animation, and the third animation is integrated from walking and waving, using the multiple animation channels approach.

5 Conclusion

Many intelligent multimedia systems are currently based on virtual human animation, exploring a variety of applications in different domains such as medical, training, interface agents, and virtual reality. The animation in these systems are either controlled by precreated animations [3] (e.g. hand-animated keyframes or motion cap-

tured data), or dynamically generated by animation techniques such as IK. However, few of these systems take temporal relations between multiple animation sequences into consideration, and integrate different human animation sequences to present simultaneous motions.

We have investigated various temporal relations between human motions performed by one animated character, and employed multiple animation channels to integrate non-conflict simultaneous motions. Compared with IMPROV's grouping method, our approach provides a finer integration of simultaneous animations, and hence achieves more flexibility and control on virtual character animation. We believe this technique has the potential to have an impact on various areas such as computer games, movie/animation production, and intelligent agents. Future research may address incorporating more complex animation generation techniques and algorithms such as machine learning.

References

1. Allen, J. F.: Maintaining knowledge about temporal intervals. Communications of the ACM, 26(11) (1983) 832-843.
2. Babski, C.: *Humanoids on the Web*. Ph.D. thesis, Computer Graphics Lab (LIG), at the Swiss Federal Institute of Technology (EPFL) (2000)
3. Badler N.I., Philips C.B., Webber B.L.: *Simulating Humans*, Oxford University Press (1993)
4. Goldberg A.: Avatars and Agents, or Life Among the Indigenous Peoples of Cyberspace. In: Dodsworth C (ed.) *Digital Illusion: Entertaining the Future with High Technology*. ACM Press, New York (1998) 161-180.
5. Huang, Z., Eliens, A. and Visser, C.: STEP: A Scripting Lan-guage for Embodied Agents, Proceedings of the Workshop on Lifelike Animated Agents, Tokyo (2002) 46-51
6. Jackendoff, R.: *Semantic Structures*. Current studies in linguistics series, Cambridge, MA: MIT Press (1990)
7. Lander, J.: *Inverse Kinematics for Real-Time Games* Handout. http://www.darwin3d.com/confpage.htm Game Developer Conference (1999)
8. Ma, M. and Mc Kevitt, P.: Interval relations in visual semantics of verbs. Special Issue on Research in Artificial Intelligence and Cognitive Science, Artificial Intelligence Review (21) (2004) 293-316, Dordrecht, The Netherlands: Kluwer-Academic Publishers.
9. Ma, M. and Mc Kevitt, P.: Visual semantics and ontology of eventive verbs. Natural Language Processing - IJCNLP-04, First International Joint Conference, Hainan Island, China, March 22-24, 2004, Keh-Yih Su, Jun-Ichi Tsujii, Jong-Hyeok Lee and Oi Yee Kwong (Eds.) (2004) 187-196, *Lecture Notes in Artificial Intelligence* (LNAI) series, LNCS 3248. Berlin, Germany: Springer Verlag.
10. Perlin, K., Goldberg, A.: Improv: A System for Scripting Interactive Actors in Virtual Worlds, In Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques SIGGRAPH '96, ACM Press, New York (1996) 205-216
11. Szarowicz, A. and Francik, J.: Human Motion for Virtual People, International Conference on Computer Games: Artificial Intelligence, Design and Education, CGAIDE (2004) Reading, UK.
12. Vendler, Z.: *Linguistics and Philosophy*. Ithaca, NY: Cornell University Press (1967)
13. VHML, Virtual Human Modelling Language, http://www.vhml.org

Coherence and Trust

Modeling Trust in Collaborative Web Search

Peter Briggs and Barry Smyth*

Smart Media Institute,
Department of Computer Science, University College Dublin (UCD), Ireland.
{Peter.Briggs, Barry.Smyth}@ucd.ie

Abstract. Recommender systems combine ideas from information retrieval, user modeling and artificial intelligence to focus on the provision of more intelligent and proactive information services. As such, recommender systems play an important role when it comes to assisting the user during both routine and specialised information retrieval tasks. Like any good assistant it is important for users to *trust* in the ability of a recommender system to respond with timely and relevant suggestions. In this paper we show how explicit models of trust can help a recommender system operating in the domain of Web search to deliver more relevant suggestions.

1 Introduction

Recommender systems are a modern response to the difficulties that end-users routinely experience when attempting to access specific information items. By combining techniques from information retrieval, user modeling, artificial intelligence and user interface design, they provide users with a more intelligent and personalized facility for information retrieval. Their value has been demonstrated in a wide range of application domains, from the recommendation of products in e-commerce applications [14] to the recommendation of Web links [6] to assist browsers.

Collaborative recommendation techniques represent a common recommendation strategy [12]. For example, collaborative filtering recommenders record and reuse rating-based profiles for individual users. Each profile stores a set of information items that a user has accessed (or purchased, used, etc.) and some form of rating for each item that reflects their interest. When it comes to making a recommendation for some *target* user, a collaborative filtering system will locate a set of *recommendation partners* whose profiles are similar to, or correlate closely with, the target user profile. The highly rated items from these partners' profiles then serve as recommendation candidates for the target user.

Recently, researchers have begun to question the *similarity assumption* that has traditionally guided recommendation partner selection in collaborative filtering research, arguing that other factors may serve to deliver more reliable

* The support of the informatics initiative of Enterprise Ireland is gratefully acknowledged.

partners as a source of recommendation knowledge. For example, the idea that different users may be more or less reliable or trustworthy has motivated the development of explicit models of trust to guide partner selection in a way that improves overall recommendation quality. This type of approach may also have important implications when it comes to improving the robustness of collaborative filtering techniques in the face of malicious users [11].

In our own work, we are primarily interested in Web search and we have previously described how a form of collaborative filtering can be used to personalise the results of traditional search engines for the needs of communities of users [15]. However, this *collaborative Web search* (CWS) technique does not maintain individual user profiles, but rather reuses the search patterns (queries and selected results) of past searches by community members (without regard to which member performed which search in the past) in order to promote results that may be relevant for new queries. The question has often been asked as to whether a more explicit form of user modeling may enhance the ability of CWS to deliver more informed recommendations. We examine this question in this paper by adapting the standard model of CWS to allow for the maintenance and reuse of individual users' search profiles. In addition, we describe how this adaptation facilitates the development of an explicit model of user trust that can be used to improve the quality of search result recommendations.

The remainder of this paper is organised as follows. The next section discusses recent work on the issue of trust in recommender systems and reviews the basic CWS model that serves as a starting point for the core contribution of this paper. In Section 3, we describe how CWS can be adapted to support the reuse of individual user search profiles, and discuss an explicit model of trust for this revised version of CWS, showing how user trust can be modeled at the user level or at the level of individual user searches. This section also describes how this model can be used during search result recommendation to prioritise results that are suggested by more trustworthy users. The results of a live-user trial are presented in Section 4 to demonstrate the potential value of these models of trust. Finally, before concluding, in Section 5 we discuss some of the implications of this revised approach to CWS.

2 Background

Trust is an instrument that has been inherent in the workings of human society since the very beginning, and yet its definition varies greatly depending on the context in which it is used (social sciences, economics, AI etc.). In this section we will focus in on a particular definition of trust as it relates to recommender systems, outlining previous work on trust modeling and the aspects that are applicable to recommender systems, and finally provide an brief overview of CWS itself.

2.1 Trust in Recommender Systems

Before investigating the potential role of trust in CWS, we first need to understand the concept that we are modeling. In the domain of CWS, we will be thinking of trust in the sense described by Mayer et al [8] as "*the willingness of a party to be vulnerable to the actions of another party based on the expectations that the other will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other party*". In a CWS system, a user has a certain level of expectation about the quality of results that will be recommended to them via other users of their search community, even though they have no direct control over those users.

Trust as a concept has a number of properties which, although largely dependant on its definition, are interesting to note nonetheless. For example, when it comes to trust between the users of a particular system, there can be different scopes of trust within that system [7]. Trust can be assessed between individual pairs of users within the system (local trust), or there might simply be one trust rating per user that acts as a barometer for how each user is viewed by the system as a whole (global trust). Carter et al. [3] suggest that the level of trust between individuals can arise as a result of past experiences between the parties, or also as a by-product of a friendship between them. In this sense, trust can have both objective and subjective aspects to it. Within systems in which trust is modeled, it is often possible for trust to be propagated from one user to another [4]. For example, if user A is trusted by user B, and user B is trusted by user C, then we might reason that user C can also trust user A (possibly with some discounting factor). This allows inferences to be made regarding the degree of trust likely to be held between users for which no direct trust ratings are available. In [5], Gyöngyi et al. apply this property to the identification of spam on the Web. Given a set of seed pages, each evaluated by a human expert as being either "good" or "bad", they make use of the concept of trust propagation and the link structure of the Web to estimate the trustworthiness of other connected Web pages. Metaxas et al. [9] use a similar approach to examine the back-propagation of *distrust* from seed pages which are known to be spam.

So where do systems that employ trust modeling get their user trust values? The most reliable source is obviously the users themselves and so some systems allow their users to explicitly rate other users; for example, the Epinions.com site used by Richardson et al [13] in their experiments. In Yahoo!'s My Web 2.0 (Beta) social search engine [16], users must explicitly invite others to join their community. This can be seen as a vote of confidence in those invited, and so a certain level of trust is then assumed to be present within the community as a whole. The other option is for the system itself to infer the trustworthiness of its users by observing their actions within some set of scenarios [10]. While perhaps not as accurate as having ratings provided by the users themselves, it is an option available to systems that maintain anonymity between their users.

Existing models of trust are heavily dependant on the definition of trust being used, and also on the requirements and limitations of their target domain. In the area of multi-agent systems, for example, trust has been modeled as a multi-

faceted concept that incorporates the sub-concepts of self-esteem, reputation and familiarity [3]. Whereas in the work of Massa et al. [7] the distinction between trust and reputation is emphasised, in our work we treat them as equivalent. Due to the nature of our CWS system, our model of trust is based solely on each user's past actions, without any input from other more subjective factors. The work of O'Donovan et al. [10] concentrates specifically on the modeling of implicit trust within collaborative recommender systems and so is of particular relevance to us. They describe techniques that can be used to estimate trust at both a user profile-level and an item-level, and in this work we focus on adapting these models for use in a CWS environment.

2.2 A Review of Collaborative Web Search

In our investigation of trust and how it relates to the domain of CWS, we use the I-SPY system [15] as the platform for our experiments. I-SPY implements collaborative search by taking advantage of the search histories of a community of like-minded users. I-SPY does not itself comprise a new search engine per se. Instead, it operates in the mode of a meta-search engine, combining the results of underlying search engines and promoting results that have been previously selected (see Fig. 1).

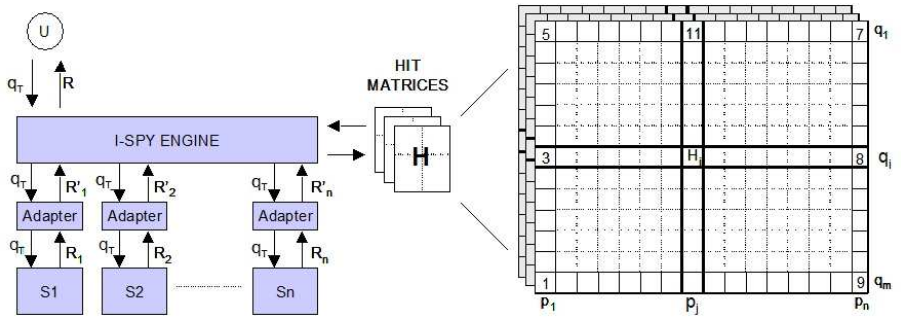


Fig. 1. The I-SPY System Architecture

Very briefly, when a user enters a new query q_T, I-SPY combines the result-lists produced by its underlying search engines, $S_1, ..., S_n$. In addition, I-SPY uses query-term overlap analysis to locate a set of similar queries, $q_1, ..., q_k$, from a *hit-matrix* that captures the search history of a given community of users. The hit-matrix records a relevance value for each query-result pair selected by the community. In I-SPY's default mode of operation, these relevance values are based on the number of times, H_{ij}, a given result has been selected by a user for a given query. I-SPY extracts the results that have been selected in the past for its k similar queries, ranks these results according to a weighted sum of their relevance values [1], and promotes them to the top of the final result-list.

In CWS, where the basis for result recommendations is provided solely by the users of the system, it is clear that the trustworthiness of those users becomes an important issue. For example, if certain users within a community are particularly unreliable at selecting good-quality search results, then their poor selections may adversely affect the quality of I-SPY's result recommendation for the entire community [2]. Likewise, malicious users might seek to purposely introduce noise into the system to promote certain results. Such vulnerabilities affect collaborative recommender systems in general [11] to varying degrees, and it is hoped that by utilising information on the trustworthiness of users, the impact of these issues can be minimised.

3 A Model of Trust for Collaborative Web Search

In this paper our key intuition is that different community members are likely to be more or less trustworthy when it comes to the ability to select (and therefore recommend) genuinely useful results in response to their queries. Accordingly, we advocate the use of an explicit model of trust in order to mediate between competing result recommendations from different community members.

3.1 Personalising Collaborative Web Search

In the standard model of CWS (see Section 2.2), the community hit-matrix stores the only record of search behaviour for the community. It does not record the identity of the user who submitted a particular query or selected a result, and as such it is not possible to analyse separately the behaviour of individual users. Of course, developing a model of user trust means that we do need to analyse the behaviour of individuals. Thus we propose an important architectural change to the standard CWS model which sees each community member associated with their own personal hit-matrix.

It is important to note that this adaptation does not change the way in which CWS reuses the search behaviours of community members in order to promote more relevant results. For example, the selection patterns for queries that are related to the target query can be extracted from the individualised hit-matrices and ranked according to the standard relevance weighting function [1].

3.2 Modeling Trust

Modeling the trustworthiness of users requires some mechanism for evaluating the quality of individual user behaviour. In other words, in order to begin to understand how reliable or trustworthy some user u is when it comes to their search activity within a community C we must be able to evaluate the reliability of their individual result selections, $r_1, ..., r_n$ for some query q relative to the needs of other community members. For example, are other community members also likely to find $r_1, ..., r_n$ relevant (see Equation 1)?

$$\mathit{Relevant}?(C, q, r) \text{ iff } r \text{ is relevant to } C \text{ for query } q \tag{1}$$

Of course, this notion of whether a result is likely to be relevant is rather loosely defined. However in what follows we will demonstrate that even relatively straightforward heuristic implementations of this can be used to model different types of trust to good effect. In the following subsections we will describe two separate models of trust, one coarse-grained model operating at the level of the user as a whole, and one finer-grained model operating at the level of a user's individual searches.

User-Level Trust Perhaps the simplest way to model the trustworthiness of a user is to consider the proportion of their selections ($Selections(C, q, u)$) that are relevant across all of the searches that they have conducted so far; see Equation 2. For example, consider an active community user who has engaged in a number of searches and selected let us say 100 different results during the course of these searches. If 70 of these selections are deemed to be relevant then this user will have a trust rating of 0.7.

$$Trust^u(C, u) = \frac{|\{r_i\} \subseteq Selections(C, q, u) \forall q : Relevant?(C, q, r_i)|}{|Selections(C, q, u) \forall q|} \quad (2)$$

The dependence of this model on a reliable measure of relevance aside, its success is likely to be limited in the case where different users are more or less reliable when it comes to searching for different types of information within the community context. For example, consider a developer within a software company and let us assume that the developers form a single community. This developer's core experience might relate to back-end Java development, and they may prove to be very reliable when it comes to searching for this type of information. However, as part of their current assignment they may be learning about Java Servlets and their searches may be far less reliable within this topic. Their wealth of experience in back-end Java, and the number of searches they have conducted in this area, is likely to obscure their inexperience in Java Servlets, and so their user-level trust score is likely to be high - which is not appropriate if they are called upon by CWS to 'help' in response to a Java Servlet query.

Query-Level Trust A query-level model of trust makes it possible to distinguish between a user's different competencies within a given community by separately measuring their trustworthiness across individual queries; see Equation 3. For example, one might expect that a user may be better at selecting relevant results when they are submitting queries about a topic with which they are familiar.

$$Trust^q(C, q, u) = \frac{|\{r_i\} \subseteq Selections(C, q, u) : Relevant?(C, q, r_i)|}{|Selections(C, q, u)|} \quad (3)$$

Thus, returning to our Java developer example above, this user will receive a high trust score for queries related to back-end Java development, but a lower trust score for their Java Servlet related queries.

3.3 Trust-Based Filtering

Once a model of trust has been defined, it is possible to use this model to manipulate the recommendations made by CWS to prioritise those that come from more trustworthy users. Ordinarily when a target user u_T submits some target query q_T, CWS will select a set of similar queries $q_1, ..., q_n$ (from across the community of users) and rank their associated result selections using the standard weighted relevance technique [1].

Here we propose a straightforward trust-based filter to eliminate the contributions from users with a trust score below some predefined threshold prior to recommendation. For example, using the user-level trust model, only the selections associated with similar queries that come from users with overall trust scores above the threshold are considered. The query-level model allows this filter to be refined to the level of an individual query so that instead of eliminating all queries for a given user, only those queries with low trust scores are eliminated, thus allowing the user to contribute search selections in niche areas where they have proven trustworthy.

4 Evaluation

In this paper we have proposed that by explicitly modeling the trustworthiness of users, we are in a better position to evaluate their individual contributions during CWS. For our evaluation, we compare the search performance of this trust-based approach (using user-level and query-level filters) to the standard version of CWS using live-user search data.

4.1 Methodology

To examine the impact of trust-based filtering in a CWS environment, we reuse the search-logs generated by a recent live-user trial of the I-SPY CWS implementation, in which 92 computer science students used I-SPY to answer 25 computer science related questions.

Trial Data The user trial was split into two sessions of equal length. Very briefly, the first session acted as a training phase in which user queries and result selections were used to train the I-SPY hit-matrix, but no result recommendations were made by I-SPY. The second session constituted a test phase in which users were recommended results based on relevance values derived from the previously populated hit-matrix. The results indicated a clear benefit to CWS, with test users enjoying significant performance gains when compared to the first-phase users. For the current evaluation we can 're-run' the trial by calculating trust ratings from the search histories of the training users and incorporating our trust-based filter into I-SPY when it comes to generating results for the test users. We can then compare the standard I-SPY results to the trust-filtered results with reference to a known set of 'correct' results for each test question.

In the experiments that follow we evaluate a number of different trust thresholds (0, 0.25, 0.5, 0.75, 1) during filtering in order to understand the sensitivity of the different techniques (user-level and query-level) to these thresholds.

Estimating Relevance As mentioned in the previous section, our trust-model relies on a way of recognising when a particular result selection is relevant for a given user and query. In this evaluation we use two different approaches to estimating relevance. First we use an *ideal* relevance measure, which is based on a manual inspection of result selections. This measure is ideal in the sense that it is perfectly accurate, but such measures tend not to be available in practice. Thus, we also explore a more practical alternative, which judges a result selection to be relevant only if that result has also been selected in the past by at least one other user in the community and, in the case of the query-level model, in response to the target query. This *heuristic* approach is far less reliable, however, and the key question is whether or not it can deliver improved search performance.

4.2 Results

We are primarily concerned with two important performance metrics. First and foremost we are interested in the impact that the trust-based filtering will have on the *precision* of the search results, compared to the standard version of I-SPY. Second, we are interested in *coverage*, specifically whether increasingly strict trust thresholds will lead to a decline in the number of search sessions that CWS will be able to influence through its result promotions. If the trust threshold is very high then there may be no users who are trustworthy enough to make result recommendations. The key question is the extent to which coverage declines for the different trust models.

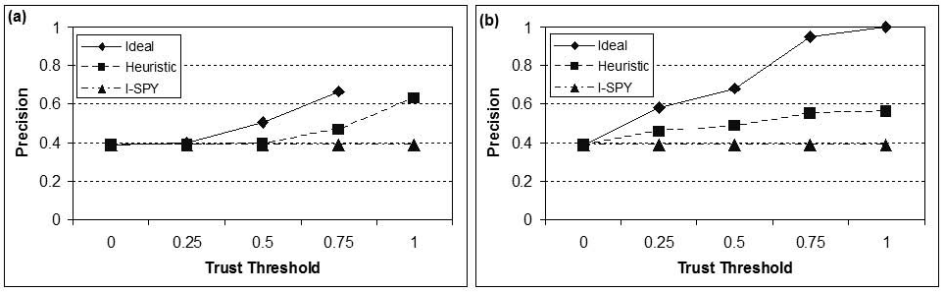


Fig. 2. (a) Precision Using User-Level Trust, (b) Precision Using Query-Level Trust

Precision Fig. 2 (a) and (b) show the mean precision of recommendations made by I-SPY using user- and query-level trust filters respectively, and the precision

of the standard I-SPY system (unaffected by trust thresholds) is also shown for comparison. In Fig. 2 (a), we can see that the user-level filter using the ideal relevance measure causes a significant increase in precision for trust thresholds > 0.25. No precision is recorded at a threshold of 1, as there are no users with an ideal trust rating of 1, and so all potential result recommendations are filtered out. The more practical heuristic filter also displays a significant improvement in precision over I-SPY, although only when the threshold used to filter user trust values is > 0.5.

Fig. 2 (b) shows a corresponding graph for the query-level filters. As expected, the increase in precision due to the finer-grained ideal query-level filter is far more pronounced than that of its user-level counterpart. At a threshold of 0.5, it has already surpassed the best precision of the ideal user-level filter, and it achieves perfect precision at a threshold of 1. In comparison, the heuristic query-level filter shows a more gradual improvement, but nevertheless reaches a precision of 0.56 at a threshold of 1 - an improvement of 44% on the precision of the standard I-SPY system.

Coverage In Fig. 3 (a) and (b), we can see the coverage provided by the trust-based versions of I-SPY relative to the coverage of the standard I-SPY system; that is, the percentage of sessions where I-SPY was still able to make a recommendation after the removal of less reliable results via the filtering process. Fig. 3 (a) shows that the user-level trust filters are potentially very susceptible to coverage decline, especially at high trust thresholds (> 0.5). For example, we see that the 63% precision enjoyed by the heuristic user-level filter comes at a significant coverage cost, with coverage falling to 52%.

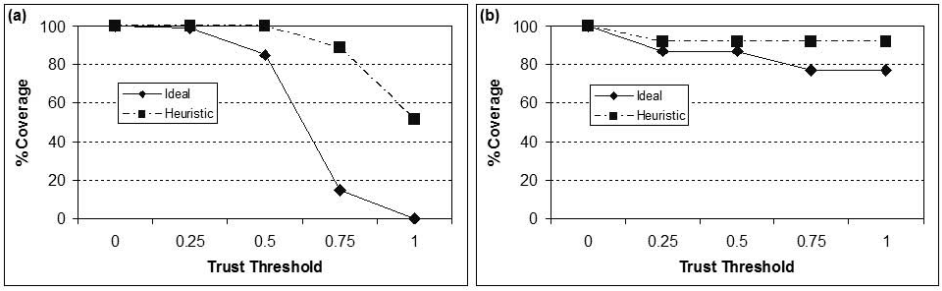


Fig. 3. (a) Coverage Using User-Level Trust, (b) Coverage Using Query-Level Trust

In contrast, the finer-grained query-level filters (Fig. 3 (b)) appear far more robust when it comes to coverage. Indeed, the heuristic measure maintains a >90% coverage level across all trust thresholds, and thus its significant precision gains (over the standard I-SPY) come at a minimal cost.

5 Conclusions

We have argued that an explicit model of trust can improve the recommendation quality of CWS. Two models with different levels of granularity have been proposed. In each case, we have found trust-based filtering capable of improving recommendation precision by up to 62% for practical trust measures. However, in the case of the coarse-grained user-level trust model, this comes at a significant potential coverage cost, whereas the finer grained query-level model is much more robust in terms of coverage.

References

1. E. Balfe and B. Smyth. Case-Based Collaborative Web Search. In *Proceedings of the 7th European Conference on Cased Based Reasoning*, pages 489–503, 2004.
2. O. Boydell, B. Smyth, C. Gurrin, and A. F. Smeaton. A Study of Selection Noise in Collaborative Web Search. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence, IJCAI-05*. Edinburgh, Scotland, (In Press).
3. J. Carter and A. A. Ghorbani. Towards a Formalization of Trust.
4. R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of Trust and Distrust, 2004.
5. Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating Web Spam with TrustRank. In *Proceedings of the 30th International Conference on Very Large Databases*, pages 576–587. Morgan Kaufmann, 2004.
6. T. Joachims, D. Freitag, and T. M. Mitchell. Web Watcher: A Tour Guide for the World Wide Web. In *IJCAI (1)*, pages 770–777, 1997.
7. P. Massa and P. Avesani. Trust-Aware Collaborative Filtering for Recommender Systems. In *CoopIS/DOA/ODBASE (1)*, pages 492–508, 2004.
8. R. C. Mayer, J. H. Davis, and F.D. Schoorman. An Integrative Model of Organizational Trust. *Academy of Management Review*, 20(3):709–734, 1995.
9. P. T. Metaxas and J. DeStefano. Web Spam, Propaganda and Trust. In *First International Workshop on Adversarial Information Retrieval on the Web*, 2005.
10. J. O'Donovan and B. Smyth. Trust in Recommender Systems. In *IUI '05: Proceedings of the 10th International Conference on Intelligent User Interfaces*, pages 167–174, New York, NY, USA, 2005. ACM Press.
11. M. P. O'Mahony, N. J. Hurley, N. Kushmerick, and G. C. M. Silvestre. Collaborative Recommendation: A Robustness Analysis. *ACM Transactions on Internet Technology - Special Issue on Machine Learning for the Internet*, 4(4):344–377, Nov 2004.
12. P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of the 1994 Conference on Computer Supported Collaborative Work*, pages 175–186, 1994.
13. M. Richardson, R. Agrawal, and P. Domingos. Trust Management for the Semantic Web. In *International Semantic Web Conference*, pages 351–368, 2003.
14. J. Ben Schafer, Joseph A. Konstan, and John Riedl. E-Commerce Recommendation Applications. *Data Mining and Knowledge Discovery*, 5(1/2):115–153, 2001.
15. B. Smyth, E. Balfe, J. Freyne, P. Briggs, M. Coyle, and O. Boydell. Exploiting Query Repetition and Regularity in an Adaptive Community-based Web Search Engine. *User Modeling and User-Adapted Interaction: The Journal of Personalization Research*, 14(5):383–423, 2004.
16. Yahoo! My Web 2.0. `http://myweb2.search.yahoo.com`, Accessed: 29/07/2005.

Coherence Measures and their Relation to Inconsistent Knowledge Bases

David H. Glass

School of Computing and Mathematics, Faculty of Engineering,
University of Ulster, Newtownabbey, BT37 0QB, UK
dh.glass@ulster.ac.uk

Abstract. This paper describes recent work on probabilistic measures of coherence and then relates it to inconsistency in knowledge bases. Two measures, analagous to two probabilistic measures, are defined to quantify the relationship between two conflicting knowledge bases.

1 Introduction

In situations where information is available from different sources it can be useful to know how coherent the information is. For example, we might be more inclined to accept the testimony of a witness if her story coheres with that of other witnesses or with information obtained in other ways. But what exactly is coherence? One point to note is that it is not the same as consistency since coherence seems to involve some degree of agreement or support among the pieces of information in question. While this much can be granted, no generally accepted definition of coherence has been forthcoming. Nevertheless, in recent years considerable attention has been given to characterizing coherence in probabilistic terms. Probabilistic accounts of coherence and its relation to theory choice in science, reliability, confirmation and testimony have been explored in some detail by Bovens and Hartmann [3].

This paper provides a brief description of probabilistic measures of coherence by considering two particular measures and some significant differences between them. Coherence measures are then compared with similarity measures for fuzzy sets before going on to investigate how coherence might be related to inconsistency in knowledge bases. While considerable attention has been given to dealing with inconsistency (see for example [8, 1, 10]), in recent years there have been papers proposing measures of information and conflict within a single knowledge base [11, 13]. The approach in this paper is to consider two knowledge bases which are individually consistent, although their union may be inconsistent. After proposing necessary conditions for a coherence measure analogous to the probabilistic case, two measures are proposed which take account of both the agreement and conflict between the knowledge bases.

2 Probabilistic Measures of Coherence

Although there is no agreed definition of coherence for a set of beliefs, there is some agreement about some of the features such a measure should possess. Here we focus on the case of two beliefs since there is no agreement even in this case and since this problem will be the most relevant for relating coherence to other areas in the remaining sections of this paper. The following three points are widely (although not universally) accepted as necessary conditions for the coherence of two beliefs A and B, denoted $C(A, B)$.

Necessary Conditions for a Coherence Measure

1. $C(A, B) = C(B, A)$,
2. $C(A, B)$ is maximal if A and B are logically equivalent, and
3. $C(A, B)$ is minimal if A and B are logically inconsistent.

A simple measure for the coherence of two beliefs satisfying these criteria whenever $P(A \vee B) \neq 0$ has been discussed by Olsson [14] and Glass [9] and is defined as follows.

Definition 1 *The coherence measure, C_1, for two beliefs A and B is defined as*

$$C_1(A, B) = \frac{P(A \wedge B)}{P(A \vee B)}, \tag{1}$$

provided $P(A \vee B) \neq 0$.

In addition to satisfying the three criteria noted above C_1 also possesses the following properties.

Proposition 1 *For probability distributions P and P',*

(a) if $P(A|B) > P'(A|B)$ and $P(B|A) > P'(B|A)$, then A and B are more coherent on distribution P than on distribution P' (see [4]),
(b) if $P(A|B) = P'(A|B)$ and $P(B|A) = P'(B|A)$, then A and B are equally C_1-coherent on distribution P and distribution P';

Proof 1 *Trivial if $P'(A|B) = 0$. Otherwise note that C_1 can be written as*

$$C_1(A, B) = \Big[\frac{1}{P(A|B)} + \frac{1}{P(B|A)} - 1\Big]^{-1}, \tag{2}$$

from which the proof follows immediately.

Proposition 1 captures the idea that it is the conditional probability of each belief given the other that is important. The important factor for the C_1 measure is the degree of overlap between the beliefs rather than how probable they are in the first place, i.e. if the relevant conditional probabilities are the same

for the distributions P and P' the coherence according to C_1 will be the same irrespective of the marginal probabilities of A and B. Although, proposition 1 seems quite plausible, it is not generally considered to be a necessary requirement for a coherence measure. Presently, we shall consider another measure of coherence proposed in the literature which does not satisfy proposition 1. First, we consider another property of C_1.

Proposition 2 *For any belief C, $C_1(A \vee C, B \vee C) \geq C_1(A, B)$.*[1]

Proposition 2 is analogous to a property shared by many measures of the similarity between two fuzzy sets and will be discussed further in section 3.

Relating coherence to the notion of support Fitelson [7] defines a measure of coherence in terms of the following measure of support.

Definition 2 *A measure of support, F, which B gives to A can be defined as*

$$F(A, B) = \frac{P(B|A) - P(B|\neg A)}{P(B|A) + P(B|\neg A)},$$

if A is contingent and B is not a necessary falsehood. $F(A, B)$ equals 1 if A and B are necessary truths, 0 if A is a necessary truth and B is contingent, and -1 if B is a necessary falsehood.

This measure of support is then used to define the following coherence measure.

Definition 3 *[7] The coherence measure, C_2, of two beliefs A and B is defined as*

$$C_2(A, B) = \frac{1}{2}\{F(A, B) + F(B, A)\}. \tag{3}$$

The C_2 measure satisfies the three requirements noted earlier for a coherence measure and thus has much in common with C_1 even though C_2 is defined on the interval $[-1, 1]$ rather than $[0, 1]$. Nevertheless, there are also some very significant differences between C_1 and C_2. In particular, Fitelson's measure, C_2, is constructed to be sensitive to probabilistic dependence so that $C_2(A, B) > 0$ if A and B have a positive probabilistic dependence on each other (i.e. $P(A|B) > P(A|\neg B)$), $C_2(A, B) < 0$ if A and B have a negative probabilistic dependence, and $C_2(A, B) = 0$ if A and B are probabilistically independent. Thus, zero provides a *neutral* point distinguishing positive and negative dependence. A consequence of taking account of probabilistic dependence in this way is that neither proposition 1 nor proposition 2 hold for the C_2 measure. The following proposition emphasizes this difference between the two measures by pointing out the insensitivity of C_1 to probabilistic dependence.

Proposition 3 $\forall \varepsilon \in [0, 1]$

[1] Most proofs have been omitted due to lack of space.

(a) it is possible to define a probabilistic distribution such that A and B have a negative probabilistic dependence and $C_1(A, B) > 1 - \varepsilon$,
(b) it is possible to define a probabilistic distribution such that A and B have a positive probabilistic dependence and $C_1(A, B) < \varepsilon$.

Clearly, the C_2 measure does not satisfy proposition 3. Thus, propositions 1, 2 and 3 provide ways of distinguishing C_1 and C_2 and, more generally, ways of distinguishing very different conceptions of coherence.

3 Similarity of Fuzzy Sets

Before investigating inconsistency in knowledge bases it is worth briefly pointing out strong parallels between probabilistic measures of coherence as discussed in section 2 and similarity measures for fuzzy sets. Many similarity measures have been proposed and their properties investigated and compared (see for example [5, 17] and references therein), but only one will be considered here to illustrate the parallel.

Suppose that two fuzzy sets A and B are represented by the vectors $(a_1, a_2, ...a_n)$ and $(b_1, b_2, ...b_n)$ respectively. The following points provide suitable necessary conditions for a measure S to be a similarity measure.

Necessary Conditions for a Similarity Measure

1. $S(A, B) = S(B, A)$,
2. $S(A, B)$ is maximal (usually 1) if A and B are identical crisp sets, and
3. $S(A, B)$ is minimal (usually 0) if A and B are crisp and A is the complement of B.

In fact, many similarity measures satisfy necessary as well as sufficient conditions for maximality and minimality. For example, as the comparison of Chen *et al* [5] illustrates, a lot of measures satisfy the following versions of conditions 2 and 3,

$2'$. $S(A, B)$ is maximal (usually 1) iff A and B are identical, and
$3'$. $S(A, B)$ is minimal (usually 0) iff $A \cap B = 0$.

A commonly used similarity measure that satisfies the stronger requirements is given by

$$S_1(A, B) = \frac{|A \cap B|}{|A \cup B|}, \tag{4}$$

provided $|A \cup B| \neq 0$. Clearly, S_1 corresponds with the C_1 coherence measure and shares some of its properties. For example, the following proposition corresponds with proposition 2.

Proposition 4 *For any fuzzy set C, $S_1(A \cup C, B \cup C) \geq S_1(A, B)$.*

Proof 2 *See the proof of proposition 2.1 in [17].*

It should be noted that Sancho-Royo and Verdegay [16] have discussed the notion of coherence between two fuzzy sets. Their account of coherence, however, is very different from that proposed here since they require that $C(A, B^C) = 1 - C(A, B)$, which is not satisfied by the measures in this paper.

4 Coherence of two Knowledge Bases

In addition to determining the degree of inconsistency of an information source, it is also important to determine the degree to which two sources agree or disagree. Hence, rather than focussing on individual knowledge bases to determine their properties and then using this information to order them in some way, the approach adopted here is to compare two knowledge bases directly. Analogous to the probabilistic case discussed in section 2, the idea is to define the degree of coherence between two knowledge bases. The knowledge bases are here assumed to be individually consistent, although their union may be either consistent or inconsistent. Before specifying conditions for coherence in this case, the notion of *complete inconsistency* is defined.

Definition 4 *Consider two classical knowledge bases K_1 and K_2, each of which is individually consistent. K_1 and K_2 are defined to be completely inconsistent with each other if and only if (i) K_1 has exactly one model, X, and K_2 has exactly one model, Y, (ii) X and Y contain the same number of literals, and (iii) $\forall \alpha$ in X, $\neg\alpha$ is in Y.*

Analogous to the necessary conditions for a probabilistic measure of coherence, the following conditions can now be proposed as necessary conditions for a satisfactory coherence measure for two individually consistent knowledge bases K_1 and K_2, denoted $C(K_1, K_2)$.

Necessary Conditions for a Coherence Measure

1. $C(K_1, K_2) = C(K_2, K_1)$,
2. $C(K_1, K_2)$ is maximal if K_1 and K_2 are logically equivalent, and
3. $C(K_1, K_2)$ is minimal if K_1 and K_2 are completely inconsistent.

In order to construct a measure satisfying these requirements, we first define the coherence between two classical models, X and Y

Definition 5 *The coherence of two models, X and Y is defined as*

$$C(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}. \tag{5}$$

This definition forms the basis for a coherence measure between two knowledge bases, K_1 and K_2. Let $M_1 = M(K_1, K_1 \cup K_2)$ be the set of classical models of K_1 that are in the set of interpretations arising from the atoms contained in $K_1 \cup K_2$. Similarly, let $M_2 = M(K_2, K_1 \cup K_2)$. The C_{KB1} coherence measure for two knowledge bases is now defined.

Definition 6 *Let K_1 and K_2 be two knowledge bases, each of which is individually consistent, and let $M_1 = M(K_1, K_1 \cup K_2)$ be the set of models of K_1 and $M_2 = M(K_2, K_1 \cup K_2)$ be the set of models of K_2. The coherence measure, C_{KB1}, for the coherence of K_1 and K_2 is defined as*

$$C_{KB1}(K_1, K_2) = \Big\{ \sum_{X \in M_1} Max_{Y \in M_2} C(X, Y) + \sum_{Y \in M_2} Max_{X \in M_1} C(X, Y) \Big\} / \; (|M_1| + |M_2|). \tag{6}$$

Example 1 $C_{KB1}(\{p\}, \{p, q\}) = 7/9$, $C_{KB1}(\{p\}, \{p \vee q\}) = 13/15$, $C_{KB1}(\{p\}, \{q\}) = 2/3$, $C_{KB1}(\{\neg p\}, \{p, q\}) = 2/9$, $C_{KB1}(\{p, q\}, \{\neg p, \neg p \rightarrow \neg q\}) = 0$ *and* $C_{KB1}(\{p \rightarrow q, r\}, \{q \rightarrow \neg r, p \vee (\neg p \wedge \neg r)\}) = 37/80$.

In addition to satisfying the three conditions proposed for a coherence measure of two knowledge bases, the C_{KB1} measure also possesses the following properties which are analogous to the properties of the probabilistic measure of coherence C_1.

Proposition 5 $\forall \varepsilon \in [0, 1]$

(a) it is possible to construct conflicting knowledge bases K_1 and K_2 such that $C_{KB1}(K_1, K_2) > 1 - \varepsilon$,
(b) it is possible to construct consistent knowledge bases K_1 and K_2 such that $C_{KB1}(K_1, K_2) < \varepsilon$.

Proposition 5 shows that the relationship of the C_{KB1} measure to consistency/inconsistency is similar to the relationship between the C_1 probabilistic measure and probabilistic dependence since consistency (inconsistency) does not guarantee a high(low) coherence. The C_{KB1} measure can be seen as measuring the degree of agreement between two knowledge bases and so coherence can be high provided the agreement outweighs the disagreement. The analogy with the probabilistic measure suggests that it should be possible to construct a coherence measure for the relationship between two knowledge bases that is sensitive to consistency/inconsistency analogous to the way in which the C_2 measure was sensitive to probabilistic dependence. The idea is that there will be a *neutral* point (given the value zero) in the coherence measure where the two knowledge

bases have no agreement or disagreement with each other. Values of coherence above this neutral point will be in agreement to a greater or lesser extent while values below it will be inconsistent to a greater or lesser extent.

Suppose that such a coherence measure is to be defined on the interval [-1,1] with zero being the neutral point as in the case of the C_2 measure in section 2. This would mean that two knowledge bases which are inconsistent with each other would have a coherence value in the interval [-1,0). In effect this would be a measure of the degree of inconsistency between the bases, with a value of -1 for completely inconsistent bases and values close to 0 for cases where there is substantially more agreement than disagreement. One approach would be to adopt a measure for the degree of conflict within a single case and apply it to the union of our knowledge bases $K_1 \cup K_2$. One such measure is proposed by Hunter [11] who defines a degree of conflict, which he calls a coherence measure, within the framework of quasi-classical logic. A brief summary of this approach is given below since some of the main features can be adapted for the present problem.

Quasi-classical (QC) logic was proposed by Besnard and Hunter [2] as a paraconsistent logic that permits derivation of non-trivializable classical inferences. Semantics for the language depends on their notion of a QC model.

Definition 7 *Let $\mathcal{A}$ be a set of atoms. Let $\mathcal{O}$ be the set of objects defined as,*

$$\mathcal{O} = \{+\alpha | \alpha \in \mathcal{A}\} \cup \{-\alpha | \alpha \in \mathcal{A}\}.$$

Any $X \subseteq \mathcal{O}$ is called a QC model. In general X can contain both $+\alpha$ and $-\alpha$.

For any atom $\alpha \in \mathcal{A}$, $+\alpha \in X$ means there is a reason for α in X and a reason against $\neg\alpha$, while $-\alpha \in X$ means there is a reason against α and a reason for $\neg\alpha$. Based on a satisfiability relation called strong satisfaction, Besnard and Hunter are able to define X as a (strong) QC model of a knowledge base K if every formula in K is strongly satisfiable in X. $QC(K)$ denotes the set of QC models of K and $MQC(K)$ denotes the set of minimal QC (MQC) models of K, i.e. QC models in $QC(K)$ that do not have a proper subset in $QC(K)$.

Hunter's coherence measure for inconsistency [11] depends on the following definition.

Definition 8 *Let X be a QC model.*

$$\text{Conflictbase}(X) = \{\alpha | +\alpha \in X and -\alpha \in X\}$$
$$\text{Opinionbase}(X) = \{\alpha | +\alpha \in X or -\alpha \in X\}.$$

Definition 9 *[11] Let X be a QC model. The coherence function for X is given by*

$$\text{Coherence}(X) = 1 - \frac{|\text{Conflictbase}(X)|}{|\text{Opinionbase}(X)|}$$

when X is non-empty, and Coherence$(\emptyset) = 1$.

The definition of coherence can then be extended to a knowledge base K by taking the maximum value of coherence over all minimum QC models of K.

Hunter's definition of coherence is related to the notion of coherence in this paper (particularly the C_1 measure in section 2) since it describes the ratio of atoms for which there is no conflicting information to the total number of atoms for which there is information. Furthermore, suppose that for two individually consistent knowledge bases K_1 and K_2, Coherence($K_1 \cup K_2$) is used to represent their coherence. This would mean that the coherence would be minimal in the case where the two knowledge bases are completely inconsistent in line with one of the criterion proposed for a coherence measure.

One difficulty, however, with applying the coherence measure in definition 9 is illustrated by noting that Coherence($\{\neg p\} \cup \{p, q\}$) = Coherence($\{\neg p, q\} \cup \{p, q\}$) = 1/2. Such an application to the union of two sets does not take into account the agreement in the second case, which would reduce the degree of conflict. This approach can be adapted, however, by taking the QC models of the individual knowledge bases and using these to calculate the coherence measure.

Definition 10 *Given a QC model X, X^- is defined by*

$$+\alpha \in X^- \; iff \; -\alpha \in X,$$
$$-\alpha \in X^- \; iff \; -\alpha \in X.$$

Definition 11 *Let K_1 and K_2 be two knowledge bases, each of which is individually inconsistent, and let $M_1 = MQC(K_1)$ be the set of minimum QC models of K_1 and M_2 be the set of minimum models of K_2. The coherence measure, C_{KB2}, for the coherence of K_1 and K_2 is defined as*

$$C_{KB2}(K_1, K_2) = \Big\{ \sum_{X \in M_1} Max_{Y \in M_2} C(X, Y) + \sum_{Y \in M_2} Max_{X \in M_1} C(X, Y) \Big\} / \; (|M_1| + |M_2|),$$

if K_1 and K_2 are consistent and as

$$C_{KB2}(K_1, K_2) = -1 \times \Big\{ \sum_{X \in M_1} Max_{Y \in M_2} C(X, Y^-) + \sum_{Y \in M_2} Max_{X \in M_1} C(X, Y^-) \Big\} / \; (|M_1| + |M_2|)$$

otherwise.

Example 2 $C_{KB2}(\{p\},\{p,q\}) = 1/2$, $C_{KB2}(\{p\},\{p \vee q\}) = 2/3$, $C_{KB2}(\{p\},\{q\}) = 0$, $C_{KB2}(\{\neg p\},\{p,q\}) = -1/2$, $C_{KB1}(\{p,q\},\{\neg p, \neg p \rightarrow \neg q\}) = -1$ *and* $C_{KB1}(\{p \rightarrow q, r\},\{q \rightarrow \neg r, p \vee (\neg p \wedge \neg r)\}) = -2/3$.

As with the C_2 measure, C_{KB2} is defined on the interval $[-1,1]$ with consistent knowledge bases having a value of coherence in the interval $[0,1]$ and inconsistent bases in $[-1,0)$. In addition to satisfying the three criteria specified earlier for a coherence measure between two knowledge bases, C_{KB2} has the property that $C_{KB2}(K_1,K_2) = 0$ if K_1 and K_2 have no atoms in common, eg $C_{KB2}(\{p\},\{q,r\}) = 0$. Thus, by way of analogy with the C_2 probabilistic measure of coherence, consistency (inconsistency) between knowledge bases in C_{KB2} corresponds with positive (negative) probabilistic dependence between beliefs in C_2 and two knowledge bases having no atoms in common corresponds with two beliefs being probabilistically independent. The analogy with the probabilistic measures can also be emphasized by noting that C_{KB2} does not satisfy proposition 5.

Although the C_{KB1} and C_{KB2} measures have a number of features in common, they also differ in significant ways. In some respects C_{KB2} seems more intuitive, eg $C_{KB2}(\{p\},\{p,q,r\}) = 1/3$ whereas $C_{KB1}(\{p\},\{p,q,r\}) = 16/25$. By contrast, however, C_{KB2} is always negative for two inconsistent knowledge bases even if there is a lot more agreement than disagreement between them. Perhaps no overall judgement can be made as to which measure is better - perhaps they are simply explicating differing intuitions regarding coherence and may be useful for different purposes.

Nevertheless, both measures overcome a problem raised by Qi *et al* [15] concerning a measure of the degree of conflict between two knowledge bases proposed by Hunter [12]. They illustrate the problem by pointing out that the measure yields the conflict between $\{p,q,r\}$ and $\{\neg p,q,r\}$ to be the same as that between $(\{p,q,r\}$ and $\{\neg p\}$ (a value of $1/3$), whereas one would expect the agreement on literals q and r in the first case would result in a lower degree of conflict. Both the C_{KB1} and C_{KB2} measures deal with this example in an appropriate way since $C_{KB1}(\{p,q,r\},\{\neg p,q,r\}) = 1/2 > 7/25 = C_{KB1}(\{p,q,r\},\{\neg p\})$ and $C_{KB2}(\{p,q,r\},\{\neg p,q,r\}) = -1/5 > -1/3 = C_{KB2}(\{p,q,r\},\{\neg p\})$ and so the first pair of knowledge bases are more coherent than the second pair according to C_{KB1} and C_{KB2}.

5 Conclusions

In this paper we have discussed differences between two probabilistic measures of coherence and explored parallels between this work, similarity measures for fuzzy sets and the inconsistency between two knowledge bases. By drawing on the work in the probabilistic case two new measures have been proposed to quantify the relationship between two knowledge bases which may be inconsistent with each other even though they are individually consistent. It is hoped that these comparisons will give rise to fruitful new directions in these areas. For example, it

should be relatively straightforward to extend the last area of work to prioritized knowledge bases within the framework of possibilistic logic analogous to the way in which Dubois *et al* [6] have extended Hunter's work to treat a single inconsistent prioritised knowledge base.

References

1. S. Benferhat, D. Dubois and H. Prade. Some syntactic approaches to the handling of inconsistent knowledge bases: a comparative study, part 1: the flat case. *Studia Logica*, 58:17–45, 1997.
2. P. Besnard and A. Hunter. Quasi-classical Logic: non-trivializable classical reasoning from inconsistent information. In *Symbolic and Quantitative Approaches to Uncertainty, LNCS*, 946:44-51, 1995.
3. L. Bovens and S. Hartmann. *Bayesian Epistemology*. Oxford University Press, Oxford, 2003.
4. L. Bovens and E. Olsson. Coherence, reliability and Bayesian networks. *Mind*, 109:685–719, 2000.
5. S. Chen, M. Yeh, P. Hsiao. A comparison of similarity measures of fuzzy values. *Fuzzy Sets and Systems*, 72:79–89, 1995.
6. D. Dubois, S. Konieczny and H. Prade. Quasi-possibilistic logic and its measures of information and conflict. *Fundamenta Informaticae*, 57:101–125, 2003.
7. B. Fitelson. A probabilistic theory of coherence. *Analysis*, 63:194–199, 2003.
8. P. Gärdenfors. *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. MIT Press, Cambridge, Massachusetts, 1988.
9. D.H. Glass. Coherence, explanation and Bayesian networks. In *Proceedings of the 13th Irish Conference on AI and Cognitive Science, LNAI*, 2464:177–182, 2002.
10. A. Hunter. Reasoning with inconsistency using quasi-classical logic. *Journal of Logic and Computation*, 10:677–703, 2000.
11. A. Hunter. Measuring inconsistency in knowledge via quasi-classical methods. In *Proceedings of the 18th American National Conference on AI (AAAI'02)*, pages 68–73, 2002.
12. A. Hunter. Making argumentation more believable. In *Proceedings of the 19th American National Conference on AI (AAAI'04)*, pages 269–274, 2004.
13. S. Konieczny, J. Lang, P. Marquis. Quantifying information and contradiction in propositional logic through test actions. In *Proceedings of the 18th International Joint Conference on AI (IJCAI'03)*, pages 106–111, 2003.
14. E. Olsson. What is the problem of coherence and truth? *Journal of Philosophy*, 99:246–272, 2002.
15. G. Qi, W. Liu and D. A. Bell. Measuring conflict and agreement between two prioritized belief bases. In *Proceedings of the 18th International Joint Conference on AI (IJCAI'03)*, to appear, 2005.
16. A. Sancho-Royo and J. L. Verdegay. Coherence measures on finite fuzzy sets. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 8:641–663, 2000.
17. X. Wang, B. De Baets and E. Kerre. A comparative study of similarity measures. *Fuzzy Sets and Systems*, 73:259–268, 1995.

Constraint Satisfaction

Robust constraint solving using multiple heuristics

Alfio Vidotto[1], Kenneth N. Brown[1], J. Christopher Beck[2]

[1]Cork Constraint Computation Centre,
Dept of Computer Science, UCC, Cork, Ireland
av1@student.cs.ucc.ie, k.brown@cs.ucc.ie

[2]Toronto Intelligent Decision Engineering Laboratory,
Dept of Mechanical and Industrial Engineering, University of Toronto, Canada
jcb@mie.utoronto.ca

Abstract. Constraint Programming is a proven successful technique, but it requires skill in modeling problems, and knowledge on how algorithms interact with models. What can be a good algorithm for one problem class can be very poor for another; even within the same class performance can vary wildly from one instance to another. CP could be easier to use if we could design robust algorithms that perform well across a range of problems, models and instances. In this paper we look specifically at variable and value ordering heuristics for backtracking search and propose a multi-heuristic algorithm based on time-slicing, and we demonstrate its performance on two different problem classes, showing it is more robust than the standard heuristics.

1 Introduction

Constraint Satisfaction is a proven AI technique, with many successful and profitable applications. However, representing and solving problems in terms of constraints can be difficult to do effectively. A single problem can be modeled in many different ways, either in terms of representation or in terms of the solving process. Different approaches can outperform each other over different problem classes or even for different instances within the same class. It is possible that even the best combination of model and search on average is still too slow across a range of problems, taking orders of magnitude more time on some problems than combinations that are usually poorer. This fact complicates the use of constraints, and makes it very difficult for novice users to produce effective solutions. The modeling and solving process would be easier if we could develop robust algorithms, which perform acceptably across a range of problems.

In this paper, we present one method of developing a robust algorithm. We combine a single model and a single basic search algorithm with a set of variable and value ordering heuristics, in a style similar to iterative deepening from standard AI search. The aim is to exploit the variance among the orderings to get a more robust procedure, which may be slower on some problems, but avoids the significant deterioration on others. During the search, we allocate steadily increasing time slices to each ordering, restarting the search at each point. We demonstrate its performance on two different problem classes, showing that it is robust across problem instances, and is competitive with standard orderings used for those problems.

2 Background

A Constraint Satisfaction Problem (CSP) is defined by a set of decision variables, $\{X_1, X_2, ..., X_n\}$, with corresponding domains of values $\{D_1, D_2, ..., D_n\}$, and a set of constraints, $\{C_1, C_2, ..., C_m\}$. Each constraint is defined by a scope, i.e. a subset of variables, and a relation which defines the allowed tuples of values for the scope. A state is an assignment of values to some or all of the variables, $\{X_i = v_i, X_j = v_j, ...\}$. A solution to a CSP is a complete and consistent assignment, i.e. an assignment of values to all of the variables, $\{X_1 = v_1, X_2 = v_2, ..., X_n = v_n\}$, that satisfies all the constraints.

The standard process for generating solutions to a CSP is based on backtracking search. This proceeds by selecting a variable and then choosing a value to assign to it. After each assignment, it propagates the constraints by removing inconsistent values from the domains of future variables. If none of the future domains are empty then search continues by selecting another variable; otherwise it backtracks, selects another value from the domain of the current variable and continues; if no other values are possible, it backtracks to the previous variable. The order in which variables and values are tried has to be specified as part of the search algorithm, and has a significant effect on the size of the search tree.

The standard ordering heuristic is based on the so called "fail-first" principle, stating that we should choose the variable with the tightest constraints. This is normally implemented by choosing the variable with the smallest remaining domain, or the smallest ratio of domain size to degree (representing the CSP as a graph, with variables as nodes and constraints as edges). Strategies aiming to "succeed first" have also been investigated, e.g. in [4] where different variable heuristics showed different search efforts, depending on their level of "promise". Even the choice of a value ordering heuristic represents an important aspect in setting up a good search algorithm. Among the most effective for many CSPs is the min-conflicts value heuristic [5], which chooses the value that rules out the fewest choices for the neighboring variables in the constraint graph. The reason why ordering heuristics matter is because if the search makes a bad choice at the top of the search tree, it can waste a lot of effort exploring sub-trees that have no solution. In [8], the behavior of standard variable ordering heuristics over insoluble sub-trees is compared to optimal refutations, with the advice that some knowledge on how refutations distribute may be relevant to improve the search.

For a single instance of a CSP, a single run with a single ordering heuristic can get trapped in the wrong area of the tree, even if the heuristic is the best on average. For this reason, the randomized restart strategy has been proposed - for a single heuristic, if no result has been found up to a given time limit, the search is started again. Tie breaking and, typically, value ordering are done randomly, and so each restart explores a different path. This approach has been shown to work well on certain problems, including quasi-group with holes [7]. Algorithm portfolios [6] is another randomized restart search method, which interleaves a number of randomized algorithms.

3 Multi-heuristic and time-slicing

As discussed above, for many problem classes no single ordering heuristic performs well across all problem instances. In some initial experiments on a scheduling problem, we had noticed that some instances caused a 1000-fold increase in running time in comparison to others. Further, the hard instances appeared to be different for each ordering. Therefore, we have developed an approach which tries each ordering in turn for a limited time, restarting the search after each one, and gradually increasing the time limit if no result was found. This is similar to the way iterative deepening explores each branch to a certain depth, and then increases the depth limit, and is similar to randomized restarts, except we use different ordering heuristics.
The pseudocode for the multi-heuristic (MH) algorithm is:

```
while solve(heuristic(i),limit) == false
    limit = increase(i,limit)
    if i == n then i = 1
    else i = i + 1
```

Solve(.,.) simply takes heuristic *i* (composed of a variable ordering and a value ordering), and applies standard search up to a time *limit*. If it finds a solution, or proves there is no solution, it returns *true*; otherwise it hits the time limit and returns *false*. *Increase(.,.)* is the time limit function. We have considered two versions: (linear) *increase(i,limit)=limit+*δ and (magnitude) *increase(i,limit)=limit*10 if i=n; limit otherwise*.

Note that MH is complete: the CSP backtracking search space is finite, each ordering heuristic is systematic, and *limit* increases indefinitely, so eventually one of the heuristics will be given enough time to complete the search. Secondly, if any one of the heuristics is deterministic, then MH has a guaranteed upper bound on the ratio of the time it takes compared to that heuristic.

4 Experiments

We want to test the performance of the time-sliced multi-heuristic approach. Specifically,

(i) is it more robust than the standard default ordering heuristic, i.e. does it report a result within acceptable time limits in more cases across a range of problems?

(ii) does it avoid a significant increase in run time, i.e. is the overhead of restarting the search, and repeating some search paths, significant?

(iii) how does it compare to the randomized restart method, i.e. is its performance due to the restart mechanism, or to the multiple heuristics?

To answer these questions, we have tested the approach on two problem classes: scheduling tasks with fixed start and end points, and quasi-groups with holes (QWH).

All implementations are coded in C++ using Ilog Solver 6.0, and run on a Pentium 2.6 GHz processor under Linux. In each case we compare our multi-heuristic approach against the recommended heuristics. For (i) and (ii), we compare MH against the smallest remaining domain (*msd*) variable ordering heuristic (with lexicographic tie breaking). For (iii) we compare against the same variable ordering heuristic but with random tie breaks, and random value ordering.

4.1 Scheduling

The problem - We considered one class of scheduling problems, where tasks have fixed start and end times, but can be allocated to a number of different resources. We assume that resources come in categories, and that categories are ranked. Each task has a rank, and must be allocated to a resource of that rank or higher. Each resource can process one task at a time, and each task must be processed without interruption on a single resource. Given a set of categorized resources and ranked tasks, with fixed start and end times, the problem is to determine whether or not the tasks can be scheduled. This problem is known to be NP-complete [2]. In our model, we represent the tasks as variables, and the resources as the values to be assigned, and the constraints ensure tasks do not overlap.

Example - In Fig. 1 we represent: four tasks with rank, and fixed start and end times (*left*); and a possible solution (*right*).

Task	Rank	Start	End
T1	3	0	2
T2	2	0	2
T3	3	1	3
T4	1	2	4

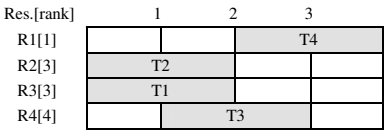


Fig. 1. Scheduling tasks with fixed start and end times over ranked resources

Variable orderings - We utilized the list of variable (task) orderings represented in Table 1. H1 and H2 are two standard versions of min-size domain. H3 to H10 are static orderings created from sorting the set of tasks by start time and minimum resource class. H11 involves a measure of time contention [3] among tasks, i.e. it sorts by counting, for each task, the number of other tasks which overlaps in time. Thus, in the example of Fig. 1, task T3 would count 3 (overlapping with T1, T2, T4), T1 and T2 would count 2, and T4 would count 1, so T3 would be tried first.

Value orderings - We utilized the list of value (resource) orderings represented in Table 2, including three static orders: two choose among resources with the smallest or highest (suitable) class first; one consisting of a random resource order.

Table 1. List of variable ordering heuristics

Heuristic id	Ordering	Tie breaking
H1	min size domain	random
H2	min size domain	lexicographic
H3	increasing start time	increasing min-resource class
H4	increasing start time	decreasing min-resource class
H5	decreasing start time	increasing min-resource class
H6	decreasing start time	decreasing min-resource class
H7	increasing min-resource class	increasing start time
H8	increasing min-resource class	decreasing start time
H9	decreasing min-resource class	increasing start time
H10	decreasing min-resource class	decreasing start time
H11	most overlapping in time	lexicographical

Table 2. List of value ordering heuristics

Heuristic id	Ordering	Tie breaking
W1	min class resource first	lexicographic
W2	max class resource first	lexicographic
W3	arbitrary fixed order	

Multi heuristic approach - We combined both lists of variable and value heuristics together, implementing four different MH versions: MH(11x3), MH(11x1), MH(1x3), and MH(1x1), all with H1 and W1 as first variable and value heuristics.

Test setting - We consider one set of test problems, $\langle$ 100, 10, N $\rangle$, with 100 resources in 10 classes. We varied the number of tasks, N, from 130 to 200 (in single steps), and for each one we generated 500 random problems, choosing start times in [0..40], durations in [17..25] and ranks in [1..10], all uniformly at random. For each instance, we impose a maximum time of 41 seconds, which allows time slices of 0.01, 0.1, and 1 second for 33 possible heuristics, including the overhead on initializing the problem.

4.2 Quasi-group with holes

The problem - A quasi-group of order N is a Latin Square of N by N cells. The solution of a Latin Square requires an allocation to each cell of a number from 1 to N, so that all the elements appearing on each row are different and all the elements appearing on each column are also different. A quasi-group with holes (QWH) is a solved Latin Square from which some allocations are deleted. The problem is to find an allocation which completes the Latin Square. In our model, the variables are the empty square cells and the values are the elements to be assigned. In our model, we represent the empty cells as variables, and the numbers as the values to be assigned. We use the Ilog global constraint IloAllDiff to ensure each row and column has allocations that are all different.

Example - In Fig. 2 we represent: a problem instance of QWH(N=4) with H=13 holes (*left*); the remaining domains (*centre*); and a possible solution (*right*).

1		2	
	2		

1	3,4	2	3,4
3,4	2	1,3,4	1,3,4
2,3,4	1,3,4	1,3,4	1,2,3,4
2,3,4	1,3,4	1,3,4	1,2,3,4

1	3	2	4
3	2	4	1
2	4	1	3
4	1	3	2

Fig. 2. Quasi group with holes: an instance, remaining domains, and a solution

Variable orderings - We utilized the list of variable (cell) orderings represented in Table 3. H1 and H2 are two standard versions of min-size domain. H3 to H10 are static orderings created from sorting the square cells by column and row.

Table 3. List of variable ordering heuristics

Heuristic id	**Ordering**	**Tie breaking**
H1	min size domain	random
H2	min size domain	lexicographic
H3	increasing column	increasing row
H4	increasing column	decreasing row
H5	decreasing column	increasing row
H6	decreasing column	decreasing row
H7	increasing row	increasing column
H8	increasing row	decreasing column
H9	decreasing row	increasing column
H10	decreasing row	decreasing column

Value orderings - We utilized the list of value (number) orderings represented in Table 4, which includes three orders, two static and one dynamic. W1 (W2) simply chooses smallest (biggest) numbers first. W3 involves a measure of conflict among numbers: if variable X is chosen, W3 looks the number frequency in the domains of all the unassigned variables in the same row and column as X. Knowing that all numbers must appear once in the column and once in the row W3 choose the number that appears least in domains of the other unassigned variables in the row and column. Thus, in the example of Fig. 2 above, assuming the bottom-right cell (variable) is chosen first, number 1 would count 4, number 2 would count 2, and numbers 3 and 4 would count 6, so W3 would choose number 2 first.

Table 4. List of value ordering heuristics

Heuristic id	Ordering	Tie breaking
W1	min number first	lexicographic
W2	max number first	lexicographic
W3	least (x, y)-conflicted number	lexicographic

Multi heuristic approach - Similarly to the way we proceeded for the scheduling problem, we combined both lists of variable and value heuristics together, implementing MH(10x3), with H1 and W1 as first variable and value heuristics.

Test setting - Experiments regarded balanced QWH problems of order N=20. We used the Gomes generator [1] and generated 10 balanced instances for problems with H holes, and did it for a series of different H around the difficulty peak. On each instance, each algorithm had a limited time length t-max of 200 seconds to solve, after that we considered the run as failed.

5 Results

5.1 Comparing to the standard recommended heuristic (*msd*)

Scheduling - In Fig. 3 we show the number of times msd and MH(11x3) hit the time limit, and the mean run time of all 4 versions. MH in its full 11x3 version consistently outperforms and improves msd (i.e. 1x1). It is more robust - it hits the time limit on fewer occasions. It also has a lower mean run time across the range. We can also observe that, as we start introducing more than one value heuristic, i.e. 1x3, we obtain a first clear improvement. There is benefit also by including more than one variable heuristic, i.e. 11x1. Note that passing from 1x1 to 11x1 the majority of the variable heuristics we add to the dynamic msd are all static. Things get better again when combining all the variable heuristics with all the value heuristics, i.e. 11x3. On other tests we also saw that excluding msd from the full version 11x3 has only a small effect on performance. Note that the line on the graph from top left to bottom right shows solubility, and relates to the right hand axis - e.g. almost 50% of size 150 problems have a solution. The hardness peak is where most problems have no solution.

failure frequency [%]		
Size [N]	*msd*	MH magnitude
130	10	4
140	22	12
150	62	16
160	58	32
170	82	40
180	28	22
190	2	0
200	0	0

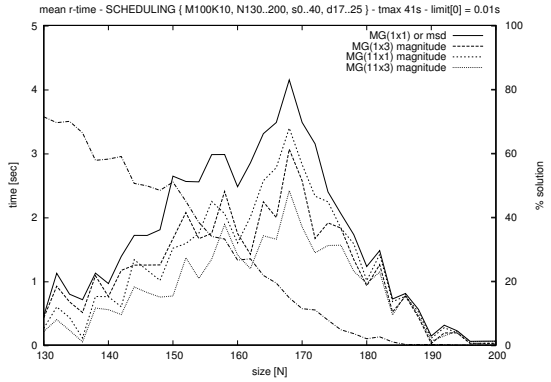


Fig. 3. Scheduling(MH vs. *msd*): left, frequency of failure to solve within *tmax*; right, mean r-time

QWH - In Fig 4, we again show robustness and run time, this time for balanced QWH(20). MH (10x3) again consistently outperforms min-size domain both in terms of robustness and run time. The graphs show two versions of MH, one with linear time-limit increase, and one with the order of magnitude increase every n restarts. All problems have solutions.

failure frequency [%]		
Size [N]	*msd*	MH magnitude
150	0	0
170	70	20
190	100	50
210	60	20
230	70	0
250	60	0
270	30	0
290	20	0

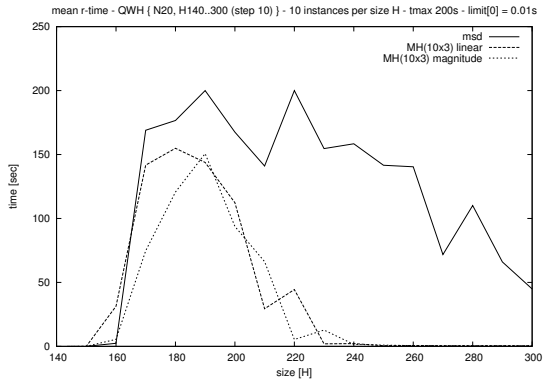


Fig. 4. QWH(MH vs. *msd*): left, frequency of failure to solve within *tmax*; right, mean r-time

5.2 Comparing to randomized-restarts on min domain

Randomized restarts (RR) is regarded to be the best method for QWH. We have compared MH with RR on both QWH and scheduling. RR is generally used with time limits that increase each restart, so we have implemented MH with the same time policy, and RR with an order of magnitude time increased every n restarts, for comparison.

QWH (Fig. 5) - RR is better than MH almost everywhere, regardless of which time slicing mechanism we use. MH performed slightly better with time slices increased by a magnitude every loop of restarts, for which version we report the statistic on the frequency of failure.

failure frequency [%]		
Size [N]	RR magnitude	MH magnitude
150	0	0
160	0	0
170	30	20
180	40	50
190	20	50
200	10	30
210	0	20
220	10	0

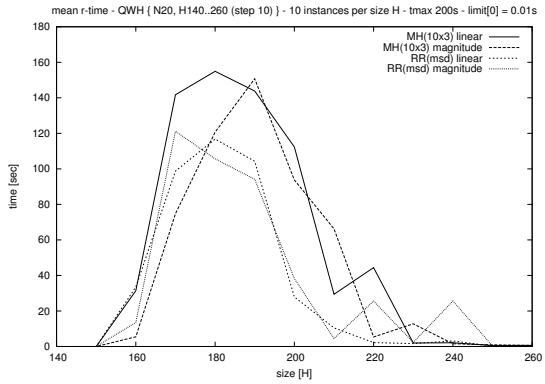


Fig. 5. QWH(MH vs. RR): left, frequency of failure to solve within *tmax*; right, mean r-time

Scheduling (Fig. 6) - MH clearly improves on RR at the peak of difficulty, which is located in the region where approximately 90% of instances have no solution. The gap is present for both slicing versions, i.e. increasing linearly every single restart (MH-Linear vs. RR-Linear), and increasing by an order of magnitude every loop (MH-Magnitude vs. RR-Magnitude). There is actually only a slight difference between the two slicing versions, with the ”magnitude” mechanism better on average.

failure frequency [%]		
Size [N]	RR magnitude	MH magnitude
130	2	4
140	14	12
150	18	16
160	42	32
170	62	40
180	32	22
190	0	0
200	0	0

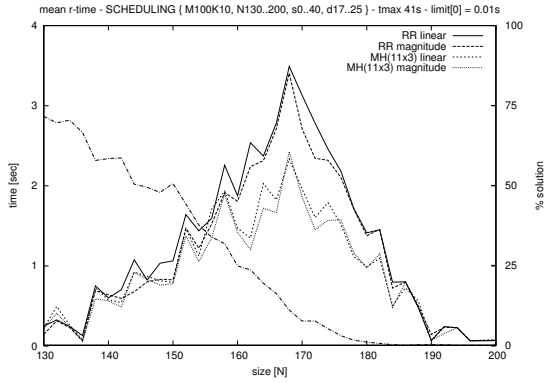


Fig. 6. Scheduling(MH vs. RR): left, frequency of failure to solve within *tmax*; right, mean r-time

6 Conclusions and future work

We have developed a multi-heuristic approach for constraint solving, designed to improve search robustness. We have tested it on two problem classes, and shown that it is more robust than the standard recommended heuristic, without degrading the run time - in fact, on average it improves the run time. We have also compared to randomized restarts, the leading method for one of our problem classes (QWH) and which uses a similar restart policy. We have shown that the multi heuristic approach is poorer in run time and robustness on QWH, but better on our scheduling problem class. Note that the different heuristics we use and the different time limits have not been tuned - they were generated by inspection of the problem characteristics, and better performance should be achievable. For the immediate future, we intend to investigate whether MH does perform better on insoluble problems (as indicated by the scheduling results).

We can conclude that the multi heuristic method offers a robust and competitive approach to constraint solving, and merits further investigation, since it offers one possible solution to the goal of making CP easier to use.

Acknowledgements

This work is funded by Enterprise Ireland (SC/2003/81), with additional support from Science Foundation Ireland (00/PI.1/C075), and ILOG, SA.

References

1. D. Achlioptas, C. P. Gomes, H. Kautz, and B. Selman. Generating satisfiable problem instances. In *Proceeding of the Seventeenth National Conference on Artificial Intellicence (AAAI-00)*, pages 256–261, New Providence, RI: AAAI Press, 2000.
2. E. M. Arkin and E. B. Silverberg. Scheduling jobs with fixed start and end times. *Discrete Applied Mathematics*, 18:1–8, 1987.
3. J. C. Beck and M. S. Fox. Dynamic problem structure analysis as a basis for constraint-directed scheduling heuristics. *Artificial Intelligence*, 117(1):31–38, 2000.
4. P.; Beck, J.C.; Prosser and R.J. Wallace. Variable ordering heuristics show promise. In *Proceedings of the Tenth International Conference on Principles and Practice of Constraint Programming, CP'04*, pages 711–715, Montreal, Canada, 2004.
5. D. Frost and R. Dechter. Look-ahead value ordering for constraint satisfaction problems. In *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI'95*, pages 572–578, Montreal, Canada, 1995.
6. C. P. Gomes and B. Selman. Algorithm portfolios. *Artificial Intelligence*, 126(1-2):43–62, 2001.
7. C. P. Gomes and D. B. Shmoys. Approximations and randomization to boost csp techniques. *Annals of Operation Research*, 130:117–141, 2004.
8. T. Hulubei and B. O'Sullivan. Optimal refutations for constraint satisfaction problems. In *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI'05*, pages 163–168, Edimburgh, Scotland, 2005.

CSP Heuristics Categorised with Factor Analysis

Richard J. Wallace

Cork Constraint Computation Centre and Department of Computer Science
University College Cork, Cork, Ireland
email: r.wallace@4c.ucc.ie

Abstract. Factor analysis is a statistical technique for reducing the number of measures responsible for a matrix of correlations to a smaller number of factors that may reflect underlying causes. In this study factor analysis was used to determine if variation in search efficiency due to different heuristics could be analyzed to reveal basic sources of variation. It was found that the variation can be ascribed to two major factors related to contention (increasing likelihood of failure) and greater propagation (immediate failure). Heuristics can be classified in terms of whether they tend to support one or the other strategy, or whether they balance the two, as reflected in the pattern of loadings on the two major factors. Improvements in efficiency can be obtained by a heuristic combination only if the combination includes heuristics that are highly correlated with each factor; therefore, two such heuristics are sufficient. This work represents a step toward understanding the action of heuristics as well as suggesting limits to heuristic performance.

1 Introduction

Variable ordering heuristics are an effective means of reducing search effort. Numerous heuristics have be devised, and many others are conceivable. Currently, there is no effective way to classify them other than in terms of features of the problem that they discriminate, such as domain size or number of constraints associated with a variable, or their overall effect on efficiency.

Outstanding open questions in this domain include,

1. To what degree are heuristics doing different things?,
2. How many factors serve to differentiate heuristics? In other words, how many distinguishable heuristic actions or strategies are there?

Related to the latter is the question, are better heuristics better because they're doing something new, or are they just better with respect to some fundamental factor like responsiveness to conflict?

At present, we have not really begun to address such questions let alone answer them cogently. In particular, there has been no attempt to analyze variability in performance in order to link it to differences in heuristics, except to say that heuristic X is more efficient than heuristic Y (on problem P).

However, statistical methods exist that are designed to analyze variation across a set of measures by attempting to associate the total variance with a small number of

common factors (ideally a much smaller number than the number of original measures). These techniques, known collectively as factor analysis, may be well-suited for the present task.

This approach is based on inter-problem variation. As will be shown, heuristics can be distinguished by the pattern of variation in search effort across problems. If the action of two heuristics is due to a common strategy, then the pattern of variation should be similar.

The next section gives a brief, general description of the basic technique and discusses its use in this domain. Section 3 outlines the experiment methodology. Section 4 describes some representative results of this analysis, and clarifies some of 'anomalies' that can occur in the factor loadings. Section 5 shows the generality of the basic factor pattern across different types of problems and algorithms. Section 6 discusses hypotheses regarding the factors obtained from this analysis. Section 7 considers other problem classes. Section 8 gives conclusions.

2 Background: Factor Analysis

"Factor analysis" refers to a collection of methods for reducing a set of variables to a smaller set that is equivalent in the sense that the set of measurements associated with the original variables can be derived from linear combinations of the new variables. This allows the investigator to reinterpret the original measurements in terms of a smaller set of possibly more fundamental variables. For details on these procedures see [1] [2].

The basic model can be described in this form (taken from [1]),

$$z_j = a_{j1}F_1 + a_{j2}F_2 + \ldots + a_{jm}F_m + d_jU_j \quad (j = 1, 2, \ldots, n),$$

for the jth measure, where the F_i are *common factors*, i.e. factors common to all measures and U_j a *unique factor* associated with measure j. Usually $m << n$. The coefficients a_{jk} are called "loadings". The square of the coefficient of U_j is called the "uniqueness" because this is the portion of the variance unique to measure j.

The process begins with a matrix of correlations, derived from samples of n cases for each type of measurement. In the present case, n is the number of problems on which search is measured and correlations are between sets of measures based on different heuristics. The factor extraction process yields a set of *uncorrelated* factors that account for a maximal amount of the variance in the original matrix. In this case, the a_{jk} above are equal to the correlation coefficients between z_j and F_k [1].

There are many methods of factor extraction. Here, the maximum-likelihood method was used, which starts from a hypothesis of m common factors and determines maximum likelihood estimates using the original correlation matrix [2] [1]. Once obtained, the set of factors, which forms a basis of an m-space, can be rotated according to various criteria. Here varimax rotation was used; this method tries to eliminate negative loadings while producing maximal loadings on the smallest set of measures possible.

In interpreting patterns of differences, one cannot assume that causal factors behave additively, only that patterns of variation can be derived from additive combinations. Although factor analysis can thereby identify independent sources of variation, their interpretation requires further investigation.

3 Experimental Methods

3.1 Factor analysis

System R was used for these analyses, downloaded from http://www.r-project.org. The factanal() function was used for the factor analysis. This program uses the maximum likelihood method for finding factors [2].

Since the number of significant factors was not known beforehand, different numbers of factors were tested, first, to determine at what point factor extraction ceased to account for any significant part of the variance, second, to determine which of these factors gave strong, reliable results.

In comparisons between sample sizes of 100 and 500 problems, the proportion of variance accounted for by successive factors and the factor loadings were similar. Hence, the results described in this paper are all based on sets of 100 problems. In each analysis, the input was a set of measures of search effort for each heuristic on each problem.

3.2 Heuristics, problems, and procedure

Heuristics used in basic tests included well-known heuristics based on simple CSP parameters, heuristics chosen for their analytic properties with respect to features of search (the FFx series [3] and the promise variable ordering heuristic [4]), and a few other heuristics gathered from the literature.

The basic analyses used a set of twelve heuristics (abbreviations in parentheses are those used in the following tables):

- Minimum domain size (dm). Choose a variable with the smallest current domain size
- Minimum domain over static degree (d/dg). Choose a variable for which this quotient is minimal.
- Minimum domain over forward degree (d/fd). Ditto.
- Maximum forward degree (fd). Choose a variable with the largest number of neighbors (adjacent nodes) in the set of uninstantiated variables.
- Maximum backward degree (bkd). Choose the variable with largest number of neighbors in the set of instantiated variables.
- Maximum product of static degree and forward degree (dg*fd).
- Maximum (future) edgesum (edgsm). Choose an edge between future (uninstantiated) variables for which the sum of the degrees of the two adjacent variables is maximal, then choose the variable in this pair with the largest forward degree.
- FF2 (ff2) See [3] for descriptions of the FFx heuristics.
- FF3 (ff3)
- FF4 (ff4)
- Maximum promise (prom). Choose the variable with the largest summed promise scores across its domain. (Promise for a value is the product ($\prod$) of the supporting values taken across all domains of neighboring future variables.)
- Maximum static degree (stdeg).

In all cases, ties were broken according to the lexical order of the variable labels. Values were chosen according to their lexical order.

Initial tests were done with homogeneous random CSPs. Problems were generated according to a probability-of-inclusion model for possible constraints, domain elements and constraint tuples (cf. [5]). In all cases graphs were fully connected. Densities given are graph densities. Unless otherwise noted, problems were in a critical complexity region, and all problems had solutions.

Later tests were based on other problem classes: geometric problems, which are random problems with small-world characteristics, and quasigroups-with-holes. Geometric problems are generated by choosing n points with random coordinates within the unit square to represent the n variables, and adding edges between all pairs of variables whose points are separated by a distance less than some threshold.

The algorithms used in these experiments were MAC-3 and forward checking, coded in Lisp. The main tests were based on (i) nodes visited during search, (ii) constraint checks. Since both measures produced similar patterns of factor loadings, results in this paper are restricted to search nodes.

4 Factor Patterns for CSP Heuristics

4.1 Basic results

The most important result of the initial analyses was that every test yielded two main factors which produced a similar pattern of loadings for the heuristics and which together accounted for most of the variance ($\geq 70\%$) (Table 1). In most tests, the bulk of the remaining variance was associated with high uniquenesses for minimum domain size and maximum backward degree. Overall, there was no simple relation between average performance and loading on a particular factor (cf. Table 1). However, the differences between min domain and backward degree and the other heuristics do have a definite relation to the pattern of loadings, as indicated by results in the next section.

4.2 Resolution of anomalous loadings for specific heuristics

On the basis of their selection strategies, it is not clear why minimum domain size and maximum backward degree show distinct patterns of variation, reflected in high unique factor loadings. One explanation is that these heuristics, unlike the others, 'start out blind', since at the top level(s) of search the features they use do not distinguish among the variables. For domain size, this is true because these problems had equal domains initially, and few or no values were filtered before the first assignment. For backward degree, at the top of the search tree all variables have a backward degree of zero, and it may be necessary to instantiate several variables before this heuristic can make sensible choices. Thus, the pattern of variation associated with these heuristics may be related to lexical choices at the top of the search tree.

To evaluate this hypothesis, tests were carried out in which the first k variable choices were made according to their lexical order, and the remaining choices were made using one of the 12 heuristics. Under this condition, all tests begin in the same

Table 1. Factor Analysis for CSP heuristics

heuristic	<30,8,0.31,0.34>				<50,10,0.18,0.37>			
	nodes	factor 1	factor 2	unique	nodes	factor 1	factor 2	unique
dom	261		0.310	**0.904**	11334	0.146	0.281	**0.900**
d/dg	143	**0.695**	**0.638**	0.018	2076	**0.913**	0.394	0.011
d/fd	130	**0.726**	**0.599**	0.114	1621	**0.909**	0.404	0.010
fd	164	**0.940**	0.300	0.027	2625	0.443	**0.873**	0.042
bkd	481	0.154	0.316	**0.876**	27391	0.107	0.224	**0.938**
dg*fd	151	**0.937**	0.322	0.018	2418	0.436	**0.897**	0.005
edgsm	160	**0.925**	0.286	0.062	2840			
ff2	163	0.488	**0.846**	0.046	3148	**0.801**	0.364	0.225
ff3	154	0.475	**0.847**	0.057	2579	**0.635**	0.448	0.396
ff4	122	0.519	**0.798**	0.095	1562	**0.734**	0.445	0.263
prom	232	**0.823**	0.212	0.278	7777	0.380	**0.702**	0.363
stdeg	147	**0.923**	0.315	0.050	2000	0.486	**0.835**	0.067

Notes. Numbers under "nodes" are averages. Other columns show loadings on two factors and unique factor loadings ("unique") for each heuristic. Unless otherwise noted, in these tables blank cells indicate factor loadings $< |0.1|$. Factors in descending order by amount of variance accounted for. Not all 12 heuristics could be analyzed together in second set of problems, so edgesum was removed from the analysis. In these tables, loadings ≥ 0.6 are in boldface.

way, so differences in the pattern of variation cannot be due to a distinct pattern of choices at the beginning of search.

The results showed that this manipulation does eliminate the high unique factor loadings, first for domain size and then for backward degree as well (Table 2). Moreover, when this occurred both heuristics showed high correlations (loadings) with one of the main factors. When the first three choices were lexical, the proportion of variance accounted for by the first two factors was 0.96. This indicates that for these problems and heuristics, the pattern of variation in search efficiency can be ascribed to a very small number of distinct factors - in the causal sense.

These results were corroborated by (i) tests in which results for lexical ordering were added to the original data sets, (ii) tests of problems with varying domain size. In the first case, the pattern of loading on the first two factors was essentially unchanged, while a distinct third factor emerged with high loadings (0.84-0.85) for min domain, max backward degree, and lexical ordering. This supports the hypothesis that the patterns of variation associated with the first two heuristics are related to lexical choices early in search. In the second case, there was a higher loading (0.55) for min domain on the same factor as in tests with initial lexical choices, although there was still a moderately high unique factor loading ($\hat{d_j}$ = 0.58).

Table 2. Factor Analysis for Heuristics after k Lexical Choices

heuristic	$k = 1$				$k = 3$			
	nodes	factor 1	factor 2	unique	nodes	factor 1	factor 2	unique
dom	11378	**0.787**	0.301	0.290	19587	**0.804**	0.565	0.034
d/dg	2738	**0.956**	0.271	0.013	7712	**0.752**	0.652	0.010
d/fd	2192	**0.952**	0.272	0.019	6473	**0.744**	0.660	0.011
fd	3762	0.575	0.571	0.343	9551	0.602	**0.796**	0.005
bkd	27391	0.526	0.312	**0.626**	37536	**0.708**	0.488	0.261
dg*fd	5456	0.315	**0.946**	0.005	8567	0.626	**0.775**	0.008
edgsm	6377	0.295	**0.939**	0.032	9462	0.617	**0.783**	0.007
ff2	5499	**0.824**	0.311	0.224	11583	**0.802**	0.587	0.012
ff3	4559	**0.831**	0.294	0.223	10402	**0.794**	0.602	0.008
ff4	2851	**0.798**	0.352	0.239	6435	**0.789**	0.607	0.009
prom	9564	0.581	0.483	0.430	21344	0.588	**0.763**	0.073
stdeg	4839	0.311	**0.937**	0.024	7980	0.648	**0.752**	0.015

Notes. <50,10,0.18,0.37> problems. To better distinguish the highest load-, ings in the columns under k=3 only those 0.7 or greater are in boldface.

5 Generality of Factor Patterns

In tests with easy problems (<50,10,0.184,0.32>), there were fewer substantial loadings on the major factors, and for most heuristics the unique factor loading was very high. However, if instead of one run per problem, each problem was tested 100 times with values chosen randomly, the simple factor structure found earlier was observed.

These results can be explained as follows. For problems with many solutions, it is more likely that differences in variable selection will lead to differences in value selection and thus to different solutions. As a result, each heuristic will show a more distinct pattern of variation across problems than it does with harder problems. In this case, if there are basic factors affecting heuristic performance, it should be possible to average out peculiarities due to differences in value selection by testing problems repeatedly and choosing values at random.

The original pattern of factor loadings was also found for insoluble problems and for a set of problems with singleton solution sets (obtained by generating problems in the usual manner and collecting only those with a single solution). In both tests, differences in the pattern of variation across problems cannot be due to differences in the solution found.

Tests with forward checking give the same pattern of factor loadings as with MAC, despite the fact that max forward degree and other heuristics that load highly on the same factor perform poorly when used with this algorithm. Table 3 shows results from a composite test that included both MAC and forward checking (only some heuristics are shown). Since the loadings are very similar with each algorithm, there are still only two major factors. In this case min domain and max backward degree are each associated with a separate factor in which they are the only heuristics with high loadings; hence, they must also perform similarly when associated with either algorithm.

6 Interpreting the Factors

To assess the significance of the two major factors, we first consider the heuristics most closely associated with each of them. Since the FF (fail-first) heuristics always have high loadings on the same factor, and these pseudo-heuristics are designed to select for failure, we will tentatively label this factor as a "contention" or "fail soon" factor. In this connection, the high loading of backward degree on this factor once the effect of random selection has been removed is significant, since this heuristic does not consider future variables at all.

High loadings on the other major factor always seem to involve future variables, either directly adjacent (max forward degree) or one edge away (edgesum). It is also significant that the other diagnostic heuristic, max summed promise, which is based entirely on look-ahead assessment, loads highly on this factor.

Increasing the number of future variables adjacent to the current variable could have either of two effects. It could lead to more likely failure through greater propagation (because more variables are adjacent to the variable just assigned a value). It might also enhance the promise of the heuristic (in terms of the "promise policy" [6]), since there are now more small domains among future variables and, therefore, less chance for choosing invalid assignments. However, the latter hypothesis cannot account for the presence of this factor when problems are insoluble. Hence, we tentatively label this factor as a "propagation" or "fail now" factor.

Most heuristics have at least moderate loadings on both factors. This is not surprising, since most heuristics would be expected to affect both the level of contention and the amount of propagation. Here, it is critical to note that factor analysis guarantees that the factors are uncorrelated. This gives us reason to think that there really are two separate *causal* factors.

6.1 Evidence from heuristic synergies

A separate line of evidence comes from the analysis of heuristic combinations, especially in the form of weighted sums of ranked selections by different heuristics. It has been shown that weighted sums of ranked selections by different heuristics can sometimes outperform the individual heuristics in isolation.

Sample results are shown in Table 4, for five heuristics and for heuristic combinations. The latter were obtained by rating each choice with respect to each heuristic on a descending scale from 10 to 1 and then adding these ratings, weighted according to the heuristic. Here, heuristics were given equal weights.

These results show that some combinations do better than any heuristic by itself, while other combinations yield no improvements. The most significant finding is that improvement can be roughly predicted by the factor loadings of the heuristics (cf. Table 1). In fact, all instances of heuristic 'synergy' occurred when heuristics in a combination loaded highly on different major factors. Moreover, the best results for combinations of two heuristics were as good as the best results for combinations of more than two. This supports the conclusion that there are only two major factors.

Table 3. Combined Analysis for MAC and FC

heuristic	nodes	factor 1	factor 2	factor 3	factor 4
		MAC			
dom	11,334	0.126	0.114	**0.886**	0.188
d/dg	2076	**0.864**	0.398	0.138	
fd	2625	0.451	**0.848**	0.167	
bkd	27,391		0.114	0.297	**0.851**
ff3	2579	**0.718**	0.383	0.121	
ff4	1562	**0.784**	0.404		0.174
stdeg	2000	0.493	**0.819**	0.159	
		forward checking			
dom	212,389		0.136	**0.956**	0.246
d/dg	32,368	**0.896**	0.322		
fd	38,568,409	0.373	**0.680**		
bkd	7,101,104		0.128	0.150	**0.943**
ff3	151,893	**0.744**	0.388		
ff4	43,416	**0.786**	0.379		0.186
stdeg	2,450,958	0.384	**0.761**		0.105

Notes. <50,10,0.184,0.37> problems, selected heuristics.

Table 4. Results for Heuristic Combinations

heur/combinat	nodes	combinat	nodes	combinat	nodes
dom	11,334	dom+d/dg	2327	dom+fd+bkwd	1430
d/dg	2076	dom+fd	1317	dom+fd+stdeg	1374
fd	2625	dom+bkwd	12,521	dom+bkwd+stdeg	1822
bkd	27,391	dom+stdeg	1427	fd+bkwd+stdeg	1991
stdeg	2000	fd+stdeg	2344	dom+d/dg+fd+stdeg	1374
		bkwd+stdeg	1876	dom+d/dg+bkwd+stdeg	1834
				all five heuristics	1470

Note. <50,10,0.184,0.37> problems.

6.2 Analysis of fail-first measures

Data on five measures related to failure during search were collected for the <50,10, 0.184, 0.37> problems. These data are based on all-solutions runs, to avoid confounding effects of promise factors (cf. [5]). The measures (means or counts per problem) were mistake-tree size (mistake-trees are insoluble subtrees rooted at the first bad assignment or "mistake", which gives the most adequate direct measure of fail-firstness [cf. [5]]), number of mistakes (initial bad assignments, which must eventually be retracted; this is the number of mistake-trees), faildepth (depth at which an assignment 'fails', i.e. leads to a domain wipeout), fail-length (difference between the level of a mistake and the level of subsequent failure), and number of failures.

When factor analysis was applied to these measures, using the set of 12 heuristics and the MAC algorithm, the same factor patterns were found as for number of nodes, with the exception of the mistakes measure. In other words, all measures related to fail-firstness (the tendency to fail as soon as possible *given that one is in an insoluble subtree*, [5]) gave the same factors and pattern of loadings as the original analysis based on search efficiency.

For mistakes, most heuristics loaded about equally on both factors. Moreover, in a one-factor analysis based on this measure, the single factor accounted for 94% of the variance. Since this measure reflects the promise of a heuristic [6], this suggests that for these problems all heuristics behaved similarly with respect to promise.

7 Results for Other Problem Classes

When factor analysis was used with geometric problems, results were similar to those with homogeneous random problems. In particular, the pattern of loadings suggested that contention and propagation effects were also the most important factors affecting patterns of variation in search. However, there were three important differences: (1) the domain/degree heuristics loaded highly on a separate (third) factor, (2) the promise heuristic did not load highly on a major factor and showed a high unique factor loading, (3) even in tests based on single heuristics, min domain loaded highly on the 'contention' factor.

These differences must have to do with the graph topology of these problems. (While these problems were denser than the random problems discussed earlier, tests with random problems of greater density gave results similar to the latter.) Presumably, with the min domain heuristic, search will tend to stay within a cluster of variables, while the forward-degree heuristics are likely to move search to other clusters where the number of uninstantiated neighbors is greater. The domain/degree heuristics should behave in an intermediate fashion. This is borne out by the patterns of correlation: while the FF heuristics and the forward-degree heuristics were highly correlated among themselves, domain/degree heuristics were almost equally well-correlated with both of these groups. It is therefore possible that the better balance between contention and propagation that would be expected in the domain/degree heuristics results in a distinct pattern of variation in search efficiency for these problems.

Evaluating tests with quasigroups was not straightforward. This was partly because some heuristics could not be used, since they either did not discriminate among variables (max static degree) or they were equivalent to other heuristics (e.g. domain/static degree, degree * forward-degree, backward degree, and less obviously, the edgesum heuristic). In addition, in cases involving forward-degree heuristics, the heuristic/anti-heuristic roles were reversed, so that min forward degree and min promise became the heuristics; but these heuristics cannot be considered to enhance propagation.

A three-factor analysis gave the most reasonable results, with min domain and min domain/forward degree loading highly on the first factor, ff2 and ff3 on the second and ff4 on the third, while min forward degree and min promise had high unique factor loadings. Since min forward degree's heuristic effects were apparently based on search remaining within a partly instantiated clique, this may represent a contention strategy

distinct from that of min domain, which should not be clique-bound especially at the beginning of search. This suggests that for highly structured problems, factor analysis may produce further (significant) factors when some heuristics make selections in relation to the structural features.

8 Conclusions

The usefulness of factor analysis is that it gives us hints about where to look for meaningful causal relations. It has the added strength of 'bounding our quest' by giving us some idea of the number of significant variables involved in a domain of study such as the effect of heuristic selection on search efficiency.

In the present work this technique has allowed us to delineate two basic factors that account for much, in some cases nearly all, of the variation in search efficiency for a set of variable ordering heuristics. Further experimentation has allowed us to associate these factors with two causal factors related to greater propagation versus a buildup of contention. This leads to a new basis for classifying variable ordering heuristics. Finally, evidence has been adduced to support the hypothesis that heuristic selection can lead to improved performance when both factors are taken into account.

In this domain, factor analysis not only gives evidence (when properly interpreted) of differences in heuristic strategies, but it also shows that problems are differentially affected by each strategy; otherwise, there would not be discernible differences in the pattern of variation under heuristics that emphasize one or the other strategy.

In this analysis, the absolute magnitude of search efficiency is not reflected in the factors or the patterns of loadings. This occurs because all distributions are standardized before the analysis. This is both a strength and a limitation. It allows the analysis to delineate similar patterns of variation despite great differences in overall efficiency (as in the MAC/FC analysis). At the same time, further analysis is needed to account for differences in absolute magnitude, as in the case of FC and propagation heuristics.

Acknowledgment. This work was supported by Science Foundation Ireland under Grant 00/PI.1/C075. R. Heffernan assisted during the early stages of this project.

References

1. Harman, H.H.: Modern Factor Analysis. 2nd edn. University of Chicago, Chicago and London (1967)
2. Lawley, D.N., Maxwell, A.E.: Factor Analysis as a Statistical Method. 2nd edn. Butterworths, London (1971)
3. Smith, B.M., Grant, S.A.: Trying harder to fail first. In: Proc. Thirteenth European Conference on Artificial Intelligence-ECAI'98, John Wiley & Sons (1998) 249–253
4. Geelen, P.A.: Dual viewpoint heuristics for binary constraint satisfaction problems. In: Proc. Tenth European Conference on Artificial Intelligence-ECAI'92. (1992) 31–35
5. Beck, J.C., Prosser, P., Wallace, R.J.: Trying again to fail-first. In: Recent Advances in Constraints. Papers from the 2004 ERCIM/CologNet Workshop-CSCLP 2004. LNAI No. 3419, Berlin, Springer (2005) 41–55
6. Beck, J.C., Prosser, P., Wallace, R.J.: Variable ordering heuristics show promise. In: Principles and Practice of Constraint Programming-CP'04. LNCS No. 3258. (2004) 711–715

Evolutionary Computing and Neural Networks

Applying Cultural Learning to Sequential Decision Task Problems

Dara Curran and Colm O'Riordan

Department of Information Technology,
NUI,Galway

Abstract. Cultural learning is the process of information transfer between individuals in a population through non-genetic means. Typically this is achieved through communication or the creation of artifacts available to all members of a population. This paper explores the effect of a cultural learning approach to the development of solutions for three test-case sequential decision tasks: connect-four, tic-tac-toe and blackjack. A teacher pupil scenario is used where highly fit individuals are selected as teachers to instruct the next generation. Experiments are conducted with populations employing population learning alone and populations combining population and cultural learning. Cultural learning has been simulated by combining genetic algorithms and neural networks using a teacher/pupil scenario.

1 Introduction

A number of learning models may be readily observed from nature and have been the focus of much study in artificial intelligence research. Population learning (i.e. learning which occurs at a population level through genetic material) is typically simulated using genetic algorithms. Life-time learning (i.e. learning which takes place during an organisms's life time through reactions with its environment) can be simulated in a variety of ways, typically employing neural networks or reinforcement learning models.

A relatively new field of study in artificial intelligence is synthetic ethology. The field is based on the premise that language and culture are too complex to be readily analysed in nature and that insight can be gained by simulating its emergence in populations of artificial organisms. While many studies have shown that lexical, syntactical and grammatical structures may spontaneously emerge from populations of artificial organisms, few discuss the impact such structures have on the relative fitness of individuals and of the entire population.

The focus of this paper is to investigate the effect of cultural learning on a population of artificial organisms attempting to find solutions to three distinct sequential decision problems. The remainder of this paper is arranged as follows. Section 2 introduces background research, including descriptions of diversity measures and cultural learning techniques that have been employed for this study. Section 3 describes the experimental setup. Section 4 presents the a description of each experiment as well as their results and Section 5 provides a conclusion.

2 Background research

2.1 Cultural Learning

Culture can be succinctly described as a process of information transfer within a population that occurs without the use of genetic material taking many forms such as language, signals or artifactual materials. Such information exchanges occur during the lifetime of individuals in a population and can greatly enhance the behaviour of such species.

An approach known as synthetic ethology [1, 2] argues that the study of language is too difficult to perform in real world situations and that more meaningful results could be produced by modelling organisms and their environment in an artificial manner. Artificial intelligence systems can create tightly controlled environments where the behaviour of artificial organisms can be readily observed and modified. Using genetic algorithms, the evolutionary approach inspired by Darwinian evolution, and the computing capacity of neural networks, artificial intelligence researchers have been able to perform a variety of experiments.

Cultural learning is the process employed by a population to transfer acquired knowledge to the next generation. A number of implementations of cultural learning have been developed including fixed lexicons [3, 4], indexed memory [5], cultural artifacts [6, 7] and signal-situation tables [1] and teacher pupil scenarios[8, 4]. The teacher/pupil approach involves the stochastic selection of highly fit agents from the population to act as teachers for the next generation of agents, labeled pupils. The number of teachers selected as a proportion of the population using a parameter termed teacher ratio. Thus, if the teacher ratio is set to 0.1, ten percent of the population's best individuals will become teachers. Pupils learn from teachers through a teaching cycle, where pupils observe the teacher's verbal output and attempt to mimic it using their own verbal apparatus. As a result of these interactions, a lexicon of symbols evolves to describe situations within the population's environment.

2.2 Sequential Decision Tasks

Sequential decision tasks are a complex class of problem that require agents to make iterative decisions at many steps throughout the task. Each decision has a direct effect on the agent's environment and in turn affects its subsequent decisions. Turn-based games are examples of sequential decision tasks that are often chosen as test beds for artificial intelligence research and many implementations exist for ready comparison and analysis[9–11].

3 Experiments

The aim of the following experiments is to investigate the effect of cultural learning on the performance of populations attempting to solve a number of sequential decision tasks. The experiments compare two populations: one which

is allowed to evolve through population learning (by a genetic algorithm) and the other which is allowed to evolve using both population and cultural learning. The experiments are carried out using an artificial life simulator developed by Curran and O'Riordan [12] capable of simulating population and cultural learning. The simulator employs an encoding scheme inspired by marker based encoding [9] and uses 2-point crossover and weight mutation.

Cultural learning is implemented based on a scheme developed by Hutchins and Hazlehurst [6] and further explored by Denaro [8] where the last hidden layer (or in Denaro's case, the output layer) of a neural network functions as a verbal input/output layer. At the end of each generation, a percentage of the best individuals in the population is selected to instruct the next. Pupil networks observe teacher networks as they interact with their environment and at each stimuli, teacher networks produce an utterance through their verbal I/O layer. The pupil responds to the utterance with its own utterance, which is then corrected to approximate that of the teacher's. After the required number of these interactions (teaching cycles) have been completed, the teachers are removed from the population and the pupils continue to interact with their environment.

In previous work by Denaro *et al.*[8], it was suggested that the addition of noise to a teacher's verbal output could enhance a population's ability to retain culturally acquired information. In our simulations, noise in the range [-0.5, 0.5] is added to the teacher's output when instructing a pupil. Like the mutation and crossover operators, noise is added with a (usually low) probability that can be set as one of the simulator parameters.

The sequential decision tasks chosen for this set of experiments are three games roughly grouped in perceived order of difficulty, beginning with tic-tac-toe following with the game of blackjack and concluding with the game of connect-four. The results presented are averaged from 20 independent runs.

3.1 The Game of Tic Tac Toe

Tic-tac-toe, or three in a row, is a very simple two player game played on a 3x3 board. Each player is assigned either the X or O symbol and takes turns placing one symbol onto the board at a time. Each player attempts to place three of his/her pieces in a horizontal, vertical or diagonal line of three.

In order to evolve good players, agents in the population compete against a perfect player rather than compete against each other. It was felt that populations of agents competing against each other would be likely to converge only to local maxima due to the lack of competitive pressure. To avoid over-fitting, the perfect player's first move is randomised to provide game diversity. The perfect player is a minimax implementation and merely provides an incentive for the population to improve. It does not provide a neural network implementation of a perfect tic-tac-toe player.

Each agent's neural network structure contains 18 input nodes, 2 for each board position where 01 is X, 10 is O and 11 is an empty square. Nine output nodes corresponding to each board position are used to indicate the agent's

desired move. The node with the strongest response corresponding to a valid move is taken as the agent's choice.

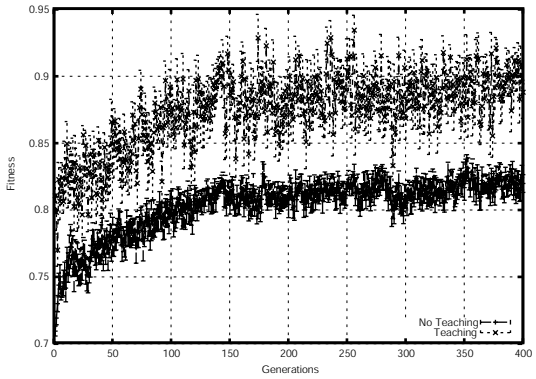


Fig. 1. *Tic-Tac-Toe Population Fitness*

Since the agents play against a perfect player, fitness is assigned according to how long each agent is capable of avoiding a loss situation. An agent's fitness is therefore correlated with the length of a game, giving the highest fitness levels to agents capable of achieving a draw with the perfect player. The fitness function produces values in the range [0,32], where 32 is the maximum fitness (the situation where the agent draws all four games).

Populations of 100 agents were generated for these experiments and allowed to evolve for 250 generations. Crossover was set at 0.6 and mutation at 0.02. The cultural learning settings of teacher ratio and teaching cycles were set at 0.1 and 5 respectively. Cultural mutation (equivalent to noise introduction described in Denaro *et al.*[8]) was set at 0.05. These parameters were determined empirically to provide the best performance.

Experimental Results Two experiments were undertaken: one using only population learning to evolve players, and the other using population and cultural learning. Figure 1 and Table 1 show the average fitness values for the two evolving populations. While both types of learning begin at similar levels of fitness, there is strong evidence (p value < 0.0001, 95% C.I.) to suggest that agents employing cultural evolution are performing better as the experiment progresses.

Discussion It is interesting to compare these results with those obtained by Angeline and Pollack [13] who used a competitive fitness function to evolve populations of neural network tic-tac-toe players. The population of evolving players was pitted against a number of 'expert' player strategies, including a perfect player. If we examine their results in terms of a draws/losses ratio, we

Table 1. Tic-Tac-Toe Average Fitness

Population	Avg. Fitness	Max Fitness	Min Fitness	S. D.
Pop. Learning	0.8029272	0.8308780	0.7086666	0.0004040
Cultural Learning	0.8732511	0.9337738	0.7692220	0.0007648

find that their best evolved players (playing against a perfect player) obtain a ratio of 0.2405. By contrast, the cultural learning approach presented in this paper obtains an average of 0.72 with highs of 0.94 and lows of 0.625.

3.2 The Game of BlackJack

Blackjack, or twenty-one, begins with the dealer dealing two cards face-up to each player and two to his/herself, with one card visible (the *up-card*) and the other face down. Cards are valued by their face value (10 for all picture cards) except for the ace which can be counted either as 11 or 1. The object of the game is to obtain a higher score (the sum of all card values) than that of the dealer's without exceeding 21. Each player can *draw* additional cards until they either *stand* or exceed 21 and go *bust*. Once all players have obtained their cards, the dealer turns over the hidden card and draws or stands as appropriate. Should the dealer's hand bust, all players win.

The dealer is at considerable advantage because he/she only enters the game once all players have fully completed their play. Thus, it is probable that some players will have bust even before the dealer reveals the hidden card. In addition, the fact that only one of the dealer's cards is visible means that players must make judgements based on incomplete information. As a rule, the dealer follows a fixed strategy, typically standing on a score of 17 or more and drawing otherwise.

All aspects related to betting such as doubling down and splitting have been removed from this implementation and only one deck is used in each game. This is in order to facilitate comparison with previous work which employs a similar approach.

Several attempts have been made to develop high performing blackjack strategies with populations of neural networks using reinforcement learning techniques[14, 15]. The nature of the game means that there is no perfect set of neural network outputs from which to perform back-propagation. It is for this reason that we wish to show that the introduction of cultural learning can generate superior strategies than reinforcement learning methods.

In this implementation, each agent's neural network is given information about the card value currently held, as well as a flag indicating the presence of an ace. In addition, each neural network is given the value of the dealer's upcard. Each experiment allows 100 agents to evolve over 500 generations. At each generation, agents play 100 games against a dealer strategy and an agent's fitness is determined by the percentage of wins obtained scaled to [0.0,1.0]. Crossover was set at 0.6 and mutation at 0.02. The cultural learning settings of teacher

ratio and teaching cycles were set at 0.1 and 5 respectively. Cultural mutation was also added with probability 0.05.

Experimental Results Figure 2 and Table 2 show that the addition of cultural learning allows the population to perform substantially better than population learning alone achieving highs of over 0.45 (45% wins) versus 0.44 for population learning. The evolved strategy outlined below was extracted from the population by examining the neural network response to all possible card values.

Table 2. Blackjack Average Fitness

Population	Avg. Fitness	Max Fitness	Min Fitness	S. D.
Pop. Learning	0.4726593	0.4898939	0.3129627	0.0002623
Cultural Learning	0.4941667	0.5128222	0.3311286	0.0004356

There is strong evidence ($p < 0.05$, 95% C.I.) to suggest that cultural learning agents are statistically different than population learning agents. The strategy is tested in the next section to ascertain its performance with respect to the bench-marked strategies.

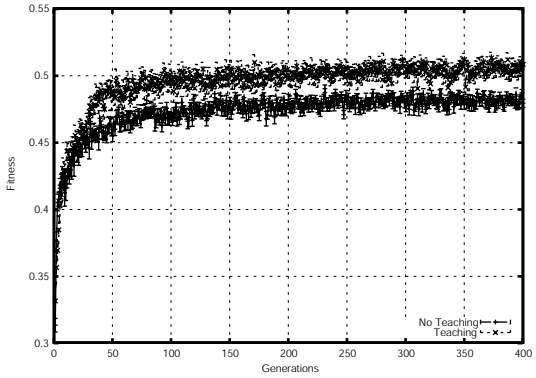


Fig. 2. *Blackjack Population Fitness*

Strategy Testing In order to assess the performance of the evolved strategy, a set of bench-marks are obtained for comparison purposes. This is achieved using a blackjack simulator consisting of a dealer, who employs the traditional dealer strategy of standing on 17 or greater, and a single player whose strategy can be set at the beginning of the simulation. The evolved strategy was compared to

a number of strategies, including an evolved strategy developed by Uribe and Sanchez[15] and 1000 runs of 1000 games were performed for each strategy to produce statistically significant results.

Table 3. Blackjack Benchmarking

Strategy	Average Fitness	Standard Deviation
Hoyle	0.4370	0.01587
Evolved Strategy	**0.4367**	**0.01582**
Dealer	0.4152	0.01576
Sanchez et al	0.3843	0.01505
Always Stand	0.3800	0.01531
Random	0.3067	0.01511

The results of the simulation (see Table 3) show that the evolved strategy does not quite reach the level of Hoyle's[16] strategy but is very close. On examination of the standard deviations, it is clear that the top two strategies are very similar, suggesting that the population has evolved an optimum strategy given the information available. It is likely that in order to out-perform Hoyle's strategy it is necessary to employ card-counting strategies.

Discussion The results presented show that cultural learning provides a modest improvement on population learning, provided that sufficient environmental information is present. It is clear that the addition of dealer information to the population significantly improves the performance of both population learning and cultural learning.

While these improvements are small, it is worth remembering that the game of blackjack is inherently very noisy and odds are very much stacked in favour of the dealer. Consequently, any statistically significant improvement such as we have shown, represents an achievement on the part of the evolutionary process. Through the bench-marking process we have shown that the evolved strategy is equivalent to the best human strategy which does not incorporate card-counting.

3.3 The Game of Connect Four

The game of connect-four is a two-player game played on a vertical board of 7x6 positions into which pieces are slotted in one of seven available slots. Each player is given a number of coloured pieces (one colour per player) and must attempt to create horizontal, vertical or diagonal piece-lines of length four. Players place one piece per turn into one of the seven slots. The piece then falls onto a free position in the chosen column, creating piles, or towers, of pieces. If a column is full, the player must select an available slot.

While the game appears simple, a certain amount of tactical knowledge is required to play proficiently. The most obvious approach is to scan the board

for existing lines of three and either finish them to create four-in-a-line, or if the line is the opponent's, block it. However, as is the case in many games, the best approaches focus on forcing the opponent to contribute to the player's victory, requiring more complex strategies.

Some research has been undertaken in the evolution of connect-four players employing a library of existing games to train the neural networks by back-propagation [17] as well as reinforcement learning methods [11].

In order for a population of neural networks to play games of connect-four, a method must be developed to encode both the board's current position and decode the network's output into a valid move. Following a number of empirical trials examining a number of techniques, the best approach, dubbed Multiple Board Selection, was chosen for this set of experiments.

Multiple Board Selection takes the current board position and produces a number of board positions resulting from each possible move. The neural network is presented with each board position and responds with an estimation of the board position's worth, expressed through its single output node. The move producing the best board position is taken to be the agent's preferred choice and is chosen as the agent's next move.

Experimental Results A population of 20 agents were allowed to evolve for 100 generations. At each generation, agents play in a tournament against all other players. In addition, each agent plays a minimax player with three levels of difficulty. In total, each agent plays 22 games of connect-four in its lifetime. Agents are assigned fitness according to each game's result: 3 points for both a win and a draw and 0 points for a loss. This gives a fitness range of [0,66]. Teachers play a full tournament while pupils observe, and at each move, the teacher corrects the pupil's output through back-propagation.

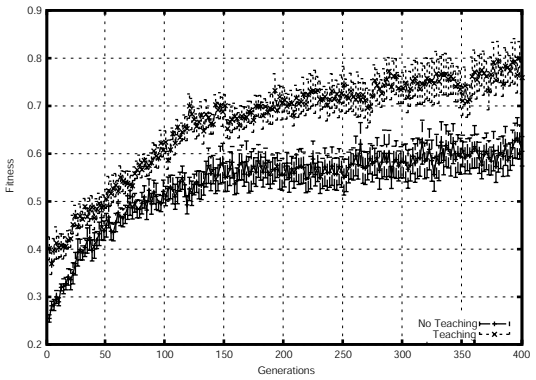


Fig. 3. *Connect-Four Population Fitness*

Table 4. Connect-Four Average Fitness

Population	Avg. Fitness	Max Fitness	Min Fitness	S. D.
Pop. Learning	0.5369646	0.6320001	0.2542307	0.0059148
Cultural Learning	0.6648816	0.7993531	0.3689372	0.0109289

Crossover was set at 0.6 and mutation at 0.02. The cultural learning settings of teacher ratio and teaching cycles were set at 0.1 and 5 respectively. Cultural mutation was also added with probability 0.05. The results in Fig. 3 and Table 4 show that the addition of cultural learning provides the best performance and that the fitness levels show an upward trend at the end of the experiment, suggesting that the population is capable of further improvement.

Discussion There is strong evidence (p value < 0.05, 95% C.I.) to suggest that the increase in performance brought by the addition of cultural learning to the population is statistically significant. It is therefore possible to conclude from this set of results that cultural learning improves the performance of agents in the connect-four environment.

4 Conclusion

This paper presents a set of experiments which highlight the usefulness of cultural learning in sequential decision problems. The results indicate that cultural learning provides improved performance over population learning in each experiment. We have shown that unlike traditional life-time learning techniques of neural network optimisation, cultural learning does not require explicit solution information.

Cultural learning gives populations the opportunity to sample acquired information within the population itself. This allows weaker members of the population to gain access to environmental information which would otherwise be impossible to attain without incurring possible fitness losses. In addition, experiments such as these provide a possible explanation of complex behaviour in nature: since no perfect solution is possible for a given environmental situation, organisms are not capable of receiving direct error feedback in the manner of synthesised life-time learning simulations. Instead, they must either rely on purely genetic information, or develop a mechanism for imparting useful knowledge to the next generation.

Future work will concentrate on more complex sequential decision problems and examine the effect of dynamic environments on populations employing cultural learning.

References

1. MacLennan, B., Burghardt, G.: Synthetic ethology and the evolution of cooperative communication. In: Adaptive Behavior 2(2). (1993) 161–188
2. Steels, L.: The synthetic modeling of language origins. In: Evolution of Communication. (1997) 1–34
3. Yanco, H., Stein, L.: An adaptive communication protocol for cooperating mobile robots. In: From Animals to Animats 2. Proceedings of the second International Conference on Simulation of Adaptive Behavior, MIT Press, Cambridge Ma. (1993) 478–485
4. Cangelosi, A., Parisi, D.: The emergence of a language in an evolving population of neural networks. Technical Report NSAL–96004, National Research Council, Rome (1996)
5. Spector, L.: Genetic programming and AI planning systems. In: Proceedings of Twelfth National Conference on Artificial Intelligence, Seattle, Washington, USA, AAAI Press/MIT Press (1994) 1329–1334
6. Hutchins, E., Hazlehurst, B.: Learning in the cultural process. In: Artificial Life II, ed. C. Langton et al., MIT Press (1991) 689–706
7. Cangelosi, A.: Evolution of communication using combination of grounded symbols in populations of neural networks. In: Proceedings of IJCNN99 International Joint Conference on Neural Networks (vol. 6), Washington, DC, IEEE Press (1999) 4365–4368
8. Denaro, D., Parisi, D.: Cultural evolution in a population of neural networks. In: M.Marinaro and R.Tagliaferri (eds), Neural Nets Wirn-96.New York: Springer. (1996) 100–111
9. Moriarty, D., Miikkulainen, R.: Discovering complex othello strategies through evolutionary neural networks. Connection Science **7** (1995) 195–209
10. Fogel, D.B.: Evolving a checkers player without relying on human experience. Intelligence **11** (2000) 20–27
11. Sommerlund, P.: Artificial neural networks applied to strategic games. Unpublished Manuscript, URL: http://www.diku.dk/student/peso/960513.ps.Z. (Last accessed 20/05/2005) (1996)
12. Curran, D., O'Riordan, C.: On the design of an artificial life simulator. In Palade, V., Howlett, R.J., Jain, L.C., eds.: Proceedings of the Seventh International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES 2003), University of Oxford, United Kingdom (2003)
13. Angeline, P.J., Pollack, J.: Competitive environments evolve better solutions for complex tasks. In Forrest, S., ed.: Proceedings of the Fifth International Conference on Genetic Algorithms, San Francisco, CA, Morgan Kaufmann (1993) 264–270
14. Olson, D.K.: Learning to Play Games from Experience: An Application of Artificial Neural Networks and Temporal Difference Learning. Pacific Lutheran University (1993)
15. Perez-Uribe, A., Sanchez, E.: Blackjack as a test bed for learning strategies in neural networks. In: International Joint Conference on Neural Networks, IJCNN'98. (1998) 2022–2027
16. Kendall, G., Smith, C.: The evolution of blackjack strategies. In Sarker, R., Reynolds, R., Abbass, H., Tan, K.C., McKay, B., Essam, D., Gedeon, T., eds.: Proceedings of the 2003 Congress on Evolutionary Computation CEC2003, Canberra, IEEE Press (2003) 2474–2481
17. Schneider, M.O., J, L.G.R. In: Proceedings of the VIIth Brazilian Symposium on Neural Networks (SBRN'02), IEEE Computer Press (2002) 236–241

Autonomous dynamics in a dense associative network for thought processes

Claudius Gros

Institute for Theoretical Physics, Frankfurt University, Germany.

Abstract. A dynamical system model with continuous-time dynamics for a dense and homogeneous associative network (dHAN) for associative thought processes is presented and studied. Activity centers representing cooperative secondary neural states of primary sensory processing areas of a cognitive system are dynamically connected to form overlapping associative memory states.
For this network we propose an autonomous dynamics. Only one memory state is active at any given time, forming a transient attractor state. It activates spontaneously a closely associated different memory state, giving rise to a time-series of associatively connected memory states, representing an associative thought process.
External activation of individual activity-centers by preprocessed sensory inputs will lead to a deviation of the thought process, activating memory-states associated both with the actual state-of-mind (the stable memory state) and the sensory input.
The dHAN is capable to recognize external patterns in a noisy background, to focus attention autonomously and allows for hierarchical memory states with an internal structure. Possible relevance to the dynamics of human thought processes is discussed.

1 Introduction

Substantial progress has been made in the last decades in the understanding of the primary sensory processing in biological cognitive systems. It is well known, that the optical input from eyes is not analyzed and memorized bit-wise. Recognition and enhancement of features and contrasts, like edges, shapes, relative distances, movements and so on, are important jobs of the primary and the secondary visual cortex and other areas devoted to the preprocessing of external sensory inputs [1].

We do not yet have firm data on the neurobiological nature of higher-level long-term memories (memories of faces, experiences, grammar representation, abstract knowledge, etc.). We assume here, that a higher-level memory state connects (associates) two or more separate activity centers. As an activity center we may regard in this context specialized output-neurons, (or assemblies of output neurons) of the primary or of the secondary visual cortex, or of any other

part of the brain involved in some specific tasks as the processing of external or internal information (position of the limbs, pain, hunger, desires, etc.).

We consider here thought processes in cognitive systems which do not involve the activation of the short-term, the working or the episodic memory. This kind of thought process might be considered as "associative thinking". We presume that it would also occur in the isolated brain. It must therefore involve the information stored in the long-term memory, the (long-term) memories.

We take hence the view, that this kind of associative thought process is characterized by the spontaneous and autonomous sequential activation of memory states. This process should be self-regulated without any controlling central unit. Furthermore, in order to make semantic sense, each activated memory state should be closely associated to its predecessor.

It would be possible to fulfill these postulates by constructing networks with two distinct components: the memory states themselves and the associative links (also called 'conjunction units' [2]) in between them. Here we take the stance, that the associative thought process occurs in a homogeneous network: memory states and conjunction units are functionally identical, there is only one kind of constituent building blocks. Depending on the initial condition, an associative link between two or more activity centers could be either a stationary memory state by itself or it could serve to form an association in between two sequential memory states in the course of a thought process.

We propose a dynamical model capable of simulating the above defined kind of thought processes. We do not claim that actual thought processes in biological cognitive systems (in our brain for instance) will be described accurately by this model. However, thought processes of the kind proposed here seem to be mandatory if a homogeneous associative network without external regulative units wants to acquire information-processing capabilities. In such kind of networks the information-processing must be self-organized by an autonomous dynamical process.

We note that these self-organized association processes work only within dense associative networks, where essentially all activity centers are connected to some others, forming what one calls in network-theory a 'Giant Strongly Connected-Component' [3]. In a sparse network there would be many unconnected subclusters incapable to communicate autonomously. We do therefore consider dense and homogeneous associative networks (dHAN) here, and one might argue that the human brain, with its about 10^4 connections per neuron, may fall into this category.

2 The model - Short-Term Relaxational Dynamics

The model we propose here consists of N activity centers (AC). Each AC is characterized by the individual activity $x_i \in [0,1]$ $(i = 1, \ldots, N)$ and by the level of the activity reservoir $\varphi_i \in [0,1]$, which we will specify later on. We consider here a continuous-time (t) evolution, $x_i = x_i(t)$ and $\varphi_i = \varphi_i(t)$. The

dynamics for the AC is governed by

$$\dot{x}_i = (1 - x_i)\Theta(r_i)r_i \; + \; x_i[1 - \Theta(r_i)]r_i \tag{1}$$

$$r_i = b_i \; + \; g(\varphi_i)\sum_{j=1}^{N} w_{i,j}x_j \; + \; \sum_{j=1}^{N} z_{i,j}x_j f(\varphi_j) \; , \tag{2}$$

where the r_i are growth rates with a respective bias b_i, akin to those in a reaction-network [4]. The role of the bias will be discussed further below, we consider $b_i \equiv 0$, if not stated otherwise. The unit for the time is arbitrary in principle and could be tuned, as well as most of the parameters entering Eqs. (1) and (2), in order to reproduce neurobiologically observed time-scales. For convenience one could take a millisecond for the time-unit, in order-of-magnitude.

The function $\Theta(r)$ occurring in Eq. (1) is the step function: $\Theta(r) = 1, 0$ for $r > 0$ and $r < 0$ respectively. The dynamics, Eq. (1), respects the normalization $x_i \in [0, 1]$, due to the prefactors $(1 - x_i)$ and x_i for the growth and depletion processes. The interactions in between the activity centers are

$$0 \leq w_{i,j} \leq w, \quad z_{i,j} = 0, \qquad \text{or} \qquad w_{i,j} = 0, \quad z_{i,j} \leq -|z| \; .$$

The breakdown of the link-matrix in an excitatory part $\sim w_{i,j}$ and inhibitory part $\sim z_{i,j}$ can be considered, as a reflection of the biological observation, that excitatory and inhibitory signals are due to neurons and interneurons respectively. We do no consider auto-associations: $w_{i,i} = z_{i,i} \equiv 0$.

A long series of experimental results from applied psychology has demonstrated that human binary associations are stored in many instances symmetrically, with some notable exceptions (see [5] and references therein). The model we propose here works fine in both cases, we consider here for the time being the recurrent case with $w_{i,j} = w_{j,i}$ and $z_{i,j} = z_{j,i}$. This initial symmetry will be broken spontaneously by the long-time dynamics, as discussed in Sect. 3.

The functions $f(\varphi)$ and $g(\varphi)$ govern the interaction in between the activity-levels x_i and the reservoir-levels φ_i. We will specify them in Sect. 3, initially we consider $f(\varphi) = g(\varphi) \equiv 1$. The network is then analogous to a neural-network with a continuous activity state $x_i \in [0, 1]$ for any neuron. After a short time period this network will relax in a stationary state dependent on the initial condition and on the link-matrices.

Contrary to most neural networks, where a finite fraction of all neurons might be active simultaneously, we will consider here link-matrices $w_{i,j}$ and $z_{i,j}$ such that only a finite number, typically between 2 and 7, of activity centers are active in a stationary state. We will identify these stationary states (the winning coalition) with memory states and consider in a second step the time evolution of these memory states through the coupling of the reservoir-levels φ_i via the reservoir functions $f(\varphi)$ and $g(\varphi)$.

w	z	x_c	Γ_φ^+	Γ_φ^-	$\varphi_c^{(f)}$	$\varphi_c^{(g)}$	Γ_φ
0.15	-1.0	0.85	0.004	0.009	0.15	0.7	0.05

Table 1. The set of model-parameters used. Other values are possible, e.g. for the modeling of biologically observed time-scales. w/z denote the non-zero matrix elements of the link-matrices $w_{i,j}/z_{i,j}$ entering Eq. (1). The filling/depletion rates for the reservoir $\Gamma_\varphi^\pm$ and x_c enter Eq. (5), the critical reservoir-levels for inhibition and activation, $\varphi_c^{(f/g)}$, enter Eq. (6) such as the width Γ_φ for the reservoir function. In the actual simulations a 5%-noise was added to the non-zero $w_{i,j}$ matrix elements.

2.1 Stability Condition and Hierarchical Memory States

The stabilization of memory states made up of clusters with $Z = 2, 3, \ldots$ activity centers is provided by an inhibitory background of links:

$$z_{i,j} \equiv z < 0, \qquad \forall (w_{i,j} = 0, i \neq j) \ , \tag{3}$$

for the uniform case. In Fig. 1 we illustrate a 12-center network. Illustrated in Fig. 1 by the black lines are the excitatory links, i.e. the non-zero matrix-elements of $w_{i,j}$. All pairs (i, j) of activity-centers not connected by a line in Fig. 1 have $z_{i,j} = -|z|$. If $|z|$ is big enough, then only those clusters of activity centers are dynamically stable in which all participating centers are mutually connected. Examples of stable memory states are highlighted.

This kind of encoding of the link-matrices is called a 'winners-take-all' situation, since fully interconnected clusters will stimulate each other via positive intra-cluster $w_{i,j}$. There will be at least one negative $z_{i,j}$-link amongst an active center of the winning memory state to every out-of-cluster AC, suppressing in this way the activity of all out-of-cluster activity centers. One can regard the associative network presented here also in terms of statistical mechanics of magnets. The memory-states can then be considered as ferromagnetically coupled clusters in a background of long-range antiferromagnetic couplings.

No AC (i) outside a Z-center memory state (MS) may be linked to all activity centers (j) making up this memory state. Otherwise (i) would be part of this MS. There may therefore be maximally $Z - 1$ positive connections between (i) and the MS. In addition, if the memory state is to be dynamically stable, the links between (i) and the MS may not be too strong:

$$|z| \ > \ \sum_{j \in \mathrm{MS}} w_{i,j}, \qquad |z| \ > \ (Z-1)\, w \ , \tag{4}$$

where the second equation holds for the uniform case, $w_{i,j} \equiv w > 0$ for the non-zero excitatory matrix elements. No 'spurious memory state' will occur if Eq. (4) is fulfilled.

In the above discussion we have used for simplicity mostly the uniform case $w_{i,j} \equiv w$ for all non-zero excitatory links. In this case all features making up a memory-state are bound together with the same strength. Memory states

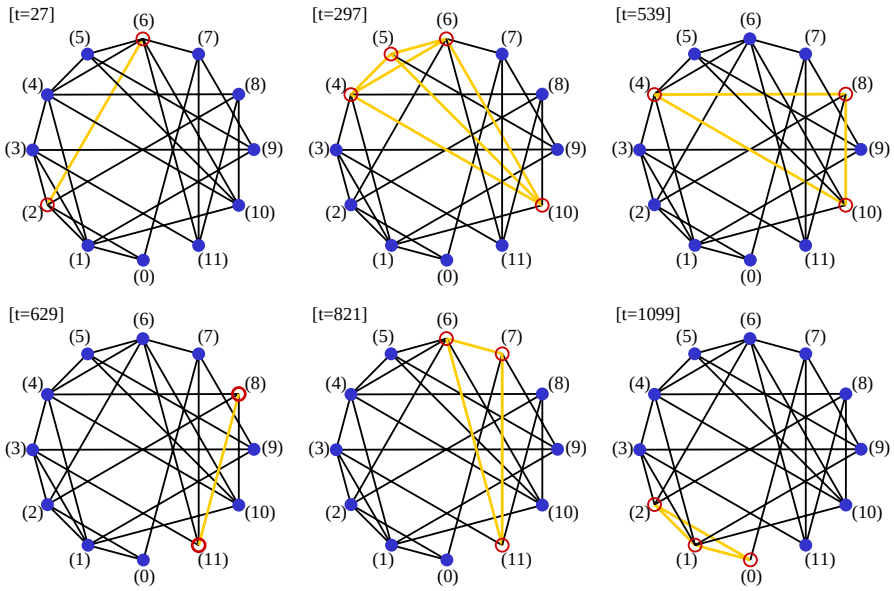


Fig. 1. The thought process $(2,6) \to (4,5,6,10) \to (4,8,10) \to (8,11) \to (6,7,11) \to (0,1,2)$ of a 12-site network with 7 2-center, 7 3-center and one 4-center memory state. The arrangement of the activity centers (filled blue circles) is arbitrary, here we have chosen a circular arrangement for a good overview. The active memory-state (open red circles connected by yellow lines) are highlighted. The non-zero excitatory links $w_{i,j} > 0$ differ from the uniform level w randomly by at most 5%. Compare Fig. 2 for the time-evolution of the variables.

corresponding to biological relevant objects will however exhibit in general a hierarchical structure [6, 7].

A memory state corresponding to 'boy' may correspond, to give an example, to a grouping of ACs corresponding to (face), (shirt), (pants), (legs), (red), (green) and so on. All these ACs would need positive links $w_{i,j} > 0$. But if red is the color of the shirt, then the link (red)-(shirt) will be much stronger than the link (red)-(legs) or (red)-(pants). The structure of the memory-states defined here is therefore flexible enough to allow for a (internal) hierarchical object representation.

No confusion regarding the colors of the shirt and of the pants arises in above example when variable link-strengths $w_{i,j}$ are used. Note however, that this is possible only because small and negative links $z_{i,j}$ are not allowed in our model, a key difference to most classical models of neural networks [8]. If weak inhibitory links would be present, the boundary of memory states could not be defined precisely, as the stability condition Eq. (4) would break down.

3 The model - Autonomous Associative Dynamics

The dynamics of most traditional neural networks is predominantly input-driven: Without external stimuli (no task to be performed) most proposed networks so far are quiet without an autonomous activity [8]. The aim of the present work is to propose a possible scenario for a continuously and spontaneously active associative network. Without external stimuli the network would be constantly active, it would be 'thinking' on its own. The network would react to external inputs and perform (or not perform, depending on the actual 'state of mind') then a trained task. We will now propose a possible mechanism allowing for autonomous and continuously ongoing thought processes.

The activity-centers will generally be made up of (biological or cybernetical) subunits. We assume here that a continuous and high activity-level depletes the resources of these subunits. Lumping all subunits together we define with $\varphi_i = \varphi_i(t) \in [0,1]$ the level of these resources (the reservoir). For the time-evolution $\dot{\varphi} = \frac{d}{dt}\varphi(t)$ of the reservoir-levels we propose

$$\dot{\varphi}_i \;=\; \Gamma_\varphi^+ \,\Theta(x_c - x_i)(1 - \varphi_i) \;-\; \Gamma_\varphi^- \,\Theta(x_i - x_c)\,\varphi_i \;. \tag{5}$$

For $x > x_c$ (high activity level) the reservoir-level φ_i decreases with the rate Γ_φ^-. For $x < x_c$ (low activity level) the reservoir-level φ_i increases with the rate Γ_φ^+. The factors $(1 - \varphi_i)$ and φ_i in Eq. (5) enforce the normalization $\varphi_i \in [0,1]$.

The dynamics induced by Eq. (1) leads to a relaxation towards the next stable memory state within a short time-scale of $\Gamma_r \approx 1/w \approx 1/|z|$, in order-of-magnitude. Choosing the rates $\Gamma_\varphi^\pm$ for the reservoir dynamics to be substantially smaller than the relaxation-rates Γ_r we obtain a separation of time-scales for the stabilization of memory state and for the depletion/filling of the activity reservoirs $\varphi_i(t)$.

A low reservoir level will have two effects: The ability to suppress another activity center via an inhibitory link $z_{i,j}$, which will be reduced by $f(\varphi)$ and the activation by other centers via an excitatory link $w_{i,j}$, which will be reduced by $g(\varphi)$, see Eq. (2). These functions may be chosen as smoothed step functions,

$$f(\varphi) \;=\; S_f(\varphi - \varphi_c^{(f)}), \qquad\qquad g(\varphi) \;=\; S_g(\varphi - \varphi_c^{(g)}) \;, \tag{6}$$

where $S_{f,g}(\Delta\varphi)$ have a sigmoidal form with a suitable width Γ_φ and the normalization conditions $S_{f,g}(0 - \varphi_c^{(f,g)}) = 0$ and $S_{f,g}(1 - \varphi_c^{(f,g)}) = 1$. For the simulations presented here, we have used a width of $\Gamma_\varphi = 0.05$.

If the network was initially recurrent with $w_{i,j} = w_{j,i}$, the time-dependence of $\varphi_i(t)$ has the consequence that this symmetry will be dynamically broken, see Eq. (2).

In Fig. 1 we present, for illustrational purposes, an autonomous thought process within a 12-center network with 15 stable memory states. The time-evolution of the corresponding activities $x_i(t)$ and the growth-rates $r_i(t)$ are shown in Fig. 2. Note that the growth-rates are positive only at and shortly after a transition to a new memory state. Competing coalitions of activity centers are suppressed due to the inhibitory background.

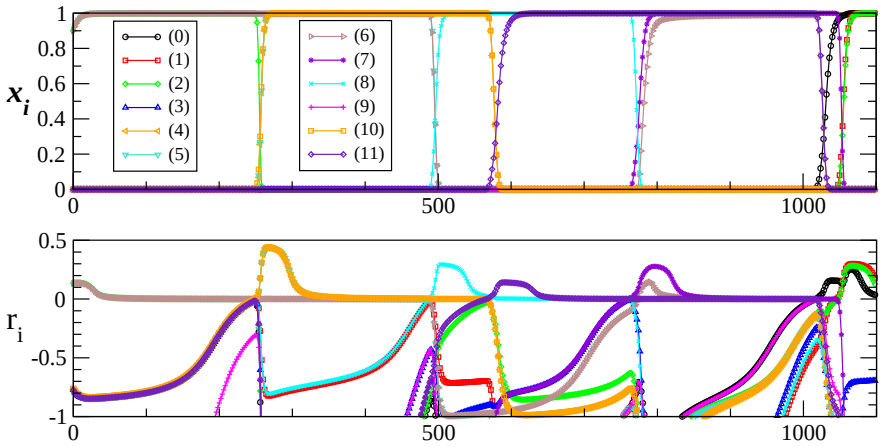


Fig. 2. The activity $x_i(t)$ (top) and the growth-rates $r_i(t)$ (bottom) for the thought process $(2, 6) \rightarrow (4, 5, 6, 10) \rightarrow (4, 8, 10) \rightarrow (8, 11) \rightarrow (6, 7, 11) \rightarrow (0, 1, 2)$ illustrated in Fig. 1.

3.1 Recognition as an Emergent Cognitive Capability

The short-term dynamics of the associative network presented here is a winners-take-all dynamics. One could imagine, in principle, the network to be dormant in its default state, reacting only to external stimuli. This 'operating modus' would, however, not allow the network to perform one of its basic duties: the recognition of patterns in external stimuli. The network should be able to distinguish a random external stimulus from a stimulus corresponding to a stored memory state.

Any external stimulus impressed on a dormant network leads to the activation of a single memory state, the one having the largest overlap with the input-pattern. This response would occur both for an input containing a pattern close to a previously stored memory state as for any random (nonsensical) input. In the later case the relaxation to a (arbitrary) memory state would just take somewhat longer. This behavior is a direct consequence of the competitive dynamics of Eq. (1), the dormant state is unstable. Such a behavior is however highly undesirable. Only sensory input corresponding closely to a stored internal representation (or an input with a high saliency) should be able to influence the state of the network.

The situation changes if the network is in its normal operating mode with a continuous autonomous activity. An external stimulus then competes with the ongoing thought process and only when it matches closely a stored memory state, will it be able to activate the corresponding memory state, suppressing the previously ongoing thought process. When this occurs, we may speak of the network having 'recognized' the object encoded in the input pattern. Recognition therefore occurs as an emergent cognitive capability within the dHAN.

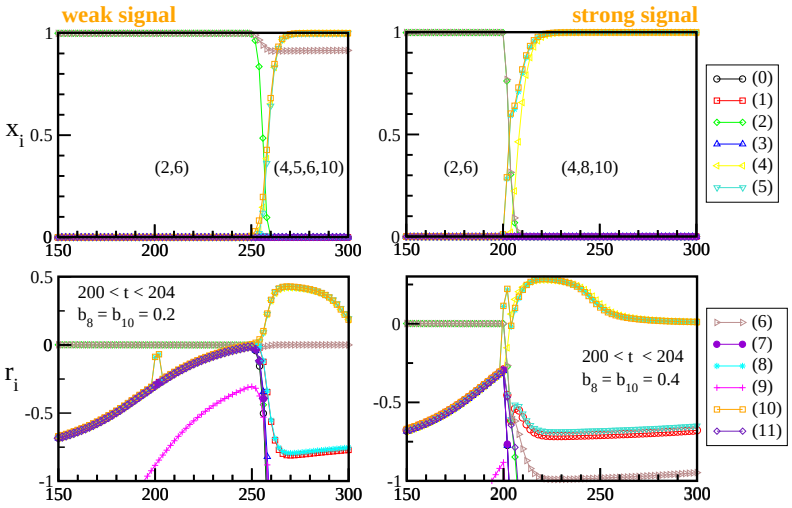


Fig. 3. Recognition as discrimination between weak and strong signals. A short external stimulus acts on sites (8) and (10) while the memory-state (2,6) of the thought-process illustrated in Fig. 1 is active. Top/bottom row: activities/growth-rates.
Left: A weak stimulus ($b_8(t) = b_{10}(t) = 0.2$ for $t \in [200, 204]$) does not manage to deviate the autonomous thought-process, the next activated memory-state is (4,5,6,10).
Right:A strong stimulus ($b_8(t) = b_{10}(t) = 0.4$ for $t \in [200, 204]$) is recognized as the state (4,8,10) and completed.

This situation shows up in the simulation presented in Fig. 3, where an external signal acts for a short time-interval on the autonomous thought-process illustrated in Fig. 1. The external signal acts by changing the bias $b_i(t)$, entering Eq. (2), with varying intensity during a finite time interval.

3.2 Dynamical Attention-Fields and Inspiration

In Fig. 4 we present a non-trivial processing of an external signal by the autonomous process shown in Fig. 1. The dHAN is in the transient attractor state (8,11). The subsequent memory-state in the absence of an external signal would be (6,7,11), see Fig. 1 and left column of Fig. 4. The response of the system is dominated by the external signal when this signal is strong, see right column of Fig. 4. Two stored memory states (1,3,4) and (1,3,9) are compatible with the external stimulus (1,3). Due to slight (random) variations in the magnitude of the link-matrices, the dHAN completes the external signal to (1,3,9), a recognition process.

It is interesting to analyze what happens in the case of a weak signal. In the process illustrated by the middle column of Fig. 4, the signal was strong enough to deviate the ongoing self-sustained thought process, but too weak to

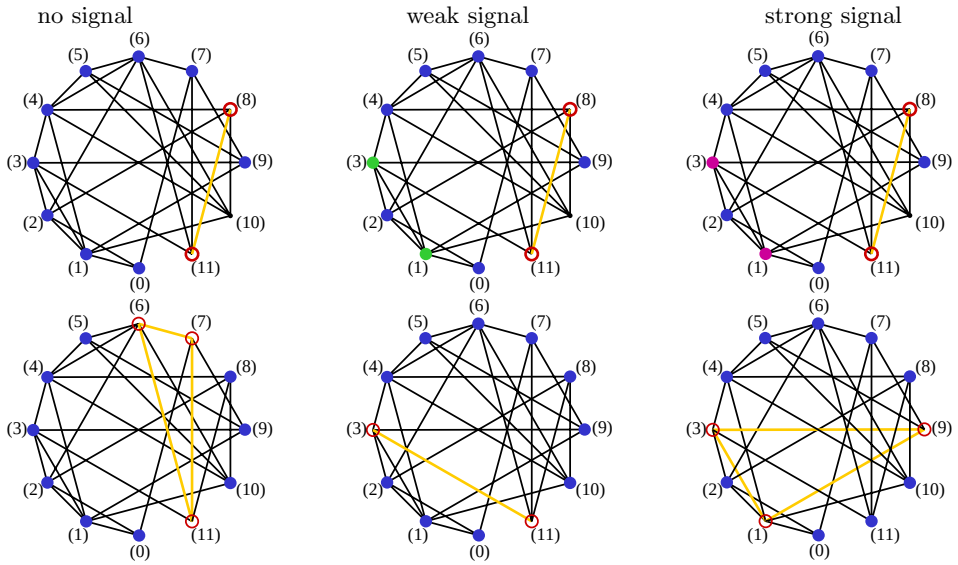


Fig. 4. External stimuli acting on sites (1) and (3) while the state (8,11) of the thought-process illustrated in Fig. 1 is active. Top/bottom row: Initial/final states.
Left: Autonomous process in the absence of an external stimulus.
Middle: A weak stimulus ($b_1(t) = b_3(t) = 0.8$ for $t \in [700, 704]$), inspiring the state (3,11).
Right: A stimulus strong enough ($b_1(t) = b_3(t) = 1.2$ for $t \in [700, 704]$) for recognition and completion of the pattern (1,3,9).

result in recognition of the external pattern. In this situation the external signal is perceived as a kind of inspiration and the memory-state (3,11) is activated.

The cause for this inspiration-process is the extension of the dynamical attention field. The resulting attention-focussing leads to a partial perception of the input. Every active memory state $MS = \{i_1,\ i_2, \ldots, i_Z\}$ comes with an attention field AF

$$AF = \{\text{sites } j \text{ such that } \ w_{j,i_l} > 0,\ i_l \in MS\}\ .$$

Signals acting on sites in AF induce a larger response, see Eqs. (1) and (2), than signals acting on sites outside the attention field. We note, that the activity center (3) is part of the attention field of (8,11), but (1) is not. This difference leads to the partial recognition of the external signal acting simultaneously on (1,3), see Fig. 4.

There is an intrinsic and deep-rooted relation between consciousness and attention [9]. One usually recognizes an object consciously when paying attention to it or if the object has salient features [10]. Both aspects emerge naturally from the autonomous associative dynamics proposed here, at least to a certain extend, as shown by the results of the simulations presented in Figs. 3 and 4. As we can

see, the dHAN is able to generate attention dynamically and autonomously due to its self-sustained dynamics.

4 Conclusions

We have discussed the principles underlying the notion of autonomous associative thinking as they would occur when short-term, working and episodic memories would be turned off in the brain. Only with the collaboration of the later three types of memories is the human brain capable of logical reasoning, the domain of most studies in artificial intelligence [11] and computational intelligence [12]. We have shown that a generalized winners-take-all implementation of a dense, homogeneous network (dHAN) with overlapping memory states is capable of a self-sustained associative thought process. This autonomous process is made-up by a time-series of transient attractors, the memory states. It is capable of recognition and might constitute the core of a more complete autonomous operating cognitive system.

References

1. Hubel, D., Wiesel, T.: Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) in the cat. J. Neurophysiol. **28** (1965) 229-289
2. Singer, W.: Neural Synchrony: A Versatile Code for the Definition of Relations? Neuron **24**, 49-65 (1999).
3. Dorogovtsev, S.N., Mendes, J.F.F.: Evolution of Networks. Oxford University Press (2003)
4. Fukai, T., Tanaka, S.: A simple neural network exhibiting selective activation of neuronal ensembles: from winner-take-all to winners-share-all. Neural Comp. **9** (1997) 77-97
5. Rizzuto, D.S., Kahana, M.J.: An Autoassociative Neural Network of Paired-Associate Learning. Neural Computation **13**, 2075-2092 (2001).
6. Riesenhuber, M., Poggio, T.: Are cortical models really bound by the "Binding Problem"? Neuron **24** (1999) 87-93
7. Mel, B., Fiser, J.: Minimizing Binding Errors Using Learned Conjunctive Features. Neural Comp. **12** (2000) 731-762
8. von der Marlsburg, C.: The what and why of binding: The Modeler's perspective. Neuron **24**, 95-1004 (1999).
9. Koch, C.: The Quest for Consciousness - A Neurobiological Approach. Robert and Company (2004).
10. Reynolds, J.H., Desimone, R.: The role of neural mechanisms of attention to solve the binding problem. Neuron **24** (1999) 19-29
11. Russel, S. and Norvig, P.: Artificial Intelligence: A modern approach. Prentice Hall (2003).
12. Konar, A.: Computational Intelligence. Springer (2004).

The Harmonic Topographic Map

Marian Peña and Colin Fyfe

Applied Computational Intelligence Research Unit,
The University of Paisley,
Scotland.
email:marian.pena,colin.fyfe@paisley.ac.uk

Abstract. We review a new form of self-organizing map which is based on a nonlinear projection of latent points into data space, identical to that performed in the Generative Topographic Mapping (GTM) [1]. We introduce a new mapping based on harmonic averages and show that it too creates a topographic mapping of the data. We illustrate these mappings on real and artificial data sets.

1 Introduction

Recently [2], one of us introduced a new topology preserving mapping which we called the Topographic Products of Experts (ToPoE). Based on a generative model of the experts, we showed how a topology preserving mapping could be created from a product of experts in a manner very similar to that used by Bishop *et al* [1] to convert a mixture of experts to the Generative Topographic Mapping (GTM).

A topographic mapping of a data set is a mapping which retains some property of the data set in an ordered manner. For example, in the visual cortex, we have neurons which have optimal response to different orientation of bars. Crucially, however, as we traverse part of the cortex, the optimal orientation changes smoothly and gradually: nearby neurons respond optimally to similar orientations. Topographic mappings are rather ubiquitous in the cortex, appearing for example in the visual, auditory, somatosensory and motor cortex. In this paper, we discuss a new method of finding topographic mappings.

In this paper, we first discuss the underlying model for the mapping and then review the K-Harmonic Means algorithm and show how it may be adapted so that it creates a topology preserving mapping which we call the Harmonic Topographic Map, HaToM.

2 The Model

We begin with a set of experts[1] who reside in some latent space and take responsibility for generating the data set. With a mixture of experts [4, 5], the

[1] Where an expert is a local model of part of the data

experts divide up the data space between them, each taking responsibility for a part of the data space. This division of labour enables each expert to concentrate on a specific part of the data set and ignore those regions of the space for which it has no responsibility. The probability associated with any data point is the sum of the probabilities awarded to it by the experts. There are efficient algorithms, notably the Expectation-Maximization algorithm, for finding the parameters associated with mixtures of experts. Bishop *et al* [1] constrained the experts' positions in latent space and showed that the resulting mapping (the GTM) also had topology preserving properties.

Each expert is deemed to generate a maximal response at some center, $\mathbf{m}_k$ in data space. We envisage that the underlying structure of the experts can be represented by K latent points, $t_1, t_2, \cdots, t_K$. To allow local and non-linear modelling, we map those latent points through a set of M basis functions, $f_1(), f_2(), \cdots, f_M()$. This gives us a matrix Φ where $\phi_{kj} = f_j(t_k)$. Thus each row of Φ is the response of the basis functions to one latent point, or alternatively we may state that each column of Φ is the response of one of the basis functions to the set of latent points. One of the functions, $f_j()$, acts as a bias term and is set to one for every input. Typically the other Φ are gaussian centered in the latent space. The output of these functions are then mapped by a set of weights, W, into data space. W is $M \times D$, where D is the dimensionality of the data space, and is the sole parameter which we change during training. We will use $\mathbf{w}_i$ to represent the i^{th} column of W and Φ_j to represent the row vector of the mapping of the j^{th} latent point. Thus each basis point is mapped to a point in data space, $\mathbf{m}_j = (\Phi_j W)^T$.

One of us [2] has previously developed this model as a topological product of experts (ToPoE) and updated the weights using gradient descent on $||\sum_i \mathbf{x} - \mathbf{m}_i||^2$ to get the learning rule

$$\Delta_n w_{md} = \sum_{k=1}^{K} \eta \phi_{km} (x_d^{(n)} - m_d^{(k)}) r_{kn} \tag{1}$$

where we have used Δ_n to signify the change due to the presentation of the n^{th} data point, $\mathbf{x}_n$, so that we are summing the changes due to each latent point's response to the data points. r_{kn} is the responsibility which the k^{th} latent point takes for the n^{th} data point.

One feature of ToPoE which is less than satisfactory is that when no latent point accepts responsibility for a data point, all latent points are given equal responsibility for that data point. This ensures that every data point is covered by the projections of the latent points but, while this is a sensible thing to do at the start of training, it seems unconvincing when it is performed in the middle of training. We therefore seek a mapping which does not have this feature. This has led us to investigate alternative criteria such as used in the HaToM.

3 Harmonic Averages

Harmonic Means or Harmonic Averages are defined for spaces of derivatives. For example, if you travel $\frac{1}{2}$ of a journey at 10 km/hour and the other $\frac{1}{2}$ at 20 km/hour, your total time taken is $\frac{d}{10} + \frac{d}{20}$ and so the average speed is $\frac{2d}{\frac{d}{10}+\frac{d}{20}} = \frac{2}{\frac{1}{10}+\frac{1}{20}}$. In general, the Harmonic Average is defined as

$$HA(\{a_i, i = 1, \cdots, K\}) = \frac{K}{\sum_{k=1}^{K} \frac{1}{a_k}} \tag{2}$$

3.1 Harmonic K-Means

The harmonic means have recently [8, 7] been used to robustify the K-means algorithm. The k-Means algorithm [3] is a well-known clustering algorithm in which N data points are allocated to K means which are positioned in data space. The algorithm is known to be dependent on its initialization: a poor set of initial positions for the means will cause convergence to a poor final clustering. [8, 7] have developed an algorithm based on the Harmonic Average which converges to a better solution than the standard algorithm.

The algorithm calculates the Euclidean distance between the i^{th} data point and the k^{th} centre as $d(\mathbf{x}_i, \mathbf{m}_k)$. Then the performance function using Harmonic averages seeks to minimize

$$Perf_{HA} = \sum_{i=1}^{N} \frac{K}{\sum_{k=1}^{K} \frac{1}{d(\mathbf{x}_i, \mathbf{m}_k)^2}} \tag{3}$$

Then we wish to move the centres using gradient descent on this performance function

$$\frac{\partial Perf_{HA}}{\partial \mathbf{m}_k} = -K \sum_{i=1}^{N} \frac{4(\mathbf{x}_i - \mathbf{m}_k)}{d(\mathbf{x}_i, \mathbf{m}_k)^4 \{\sum_{l=1}^{K} \frac{1}{d(\mathbf{x}_i, \mathbf{m}_l)^2}\}^2} \tag{4}$$

Setting this equal to 0 and "solving" for the $\mathbf{m}_k$'s, we get a recursive formula

$$\mathbf{m}_k = \frac{\sum_{i=1}^{N} \frac{1}{d_{i,k}^4 (\sum_{l=1}^{K} \frac{1}{d_{i,l}^2})^2} \mathbf{x}_i}{\sum_{i=1}^{N} \frac{1}{d_{i,k}^4 (\sum_{l=1}^{K} \frac{1}{d_{i,l}^2})^2}} \tag{5}$$

where we have used $d_{i,k}$ for $d(\mathbf{x}_i, \mathbf{m}_k)$ to simplify the notation. There are some practical issues to deal with in the implementation details of which are given in [8, 7].

[8] have extensive simulations showing that this algorithm converges to a better solution (less prone to finding a local minimum because of poor initialisation) than both standard K-means or a mixture of experts trained using the EM algorithm.

3.2 The Harmonic Topographic Map

The (5) formula is now used with the latent variable model. Since

$$\frac{\partial Perf_{HA}}{\partial W} = \frac{\partial Perf_{HA}}{\partial \mathbf{m}_k}\frac{\partial \mathbf{m}_k}{\partial \mathbf{W}} = \frac{\partial Perf_{HA}}{\partial \mathbf{m}_k}\Phi_k \tag{6}$$

With this learning rule on the latent model, we get a mapping which has elements of topology preservation but which often exhibits twists, such as are well-known in Kohonen's SOM [6]. We therefore opt to begin with a small value of K (for one dimensional latent spaces, we tend to use K=2, for two dimensional latent spaces and a square grid, we use K=2*2) and grow the mapping. We do not randomise W each time we augment K. The current value of W is approximately correct and so we need only continue training from this current value. Also for this paper we have implemented a pseudo-inverse method for the calculation of W from the positions of the centres, rather than (6). Then the algorithm is

1. Initialise K to 2. Initialise the W weights randomly and the centres of the M basis functions uniformly across latent space.
2. Initialise the K latent points in latent space.
3. Calculate the projection of the latent points to data space. This gives the K centres, $\mathbf{m}_k$.
   (a) count=0
   (b) For every data point, $\mathbf{x}_i$, calculate $d_{i,k} = ||\mathbf{x}_i - \mathbf{m}_k||$.
   (c) Recalculate means using (5).
   (d) If count<MAXCOUNT, count= count +1 and return to 3b
4. Recalculate W using $(\Phi^T\Phi + \delta I)^{-1}\Phi^T\mathbf{m}$ where $\mathbf{m}$ is the matrix containing the centres, I is identity matrix and δ is a small constant, necessary because initially $K < M$ and so the matrix $\Phi^T\Phi$ is singular.
5. If $K < K_{max}$, $K = K + 1$ and return to 2.

In the simulations below, MAXCOUNT was set at 20.

3.3 Simulations

Artificial Data We show in Figure 1 the result of a simulation in which we have 20 latent points deemed to be equally spaced in a one dimensional latent space, passed through 5 Gaussian basis functions and then mapped to the data space by the linear mapping W which is the only parameter we adjust. We generated 60 two dimensional data points, (x_1, x_2), from the function $x_2 = x_1 + 1.25\sin(x_1) + \mu$ where μ is noise from a uniform distribution in [0,1] and x_1 is drawn from a uniform distribution in $[0,2\pi]$. We use 10000 iterations of the learning rule (randomly sampling with replacement from the data set) We see that for a small number of latent points the mapping from latent space to data space preserves the 1 dimensional nature of the data. However the last diagram in that figure shows the mapping of 20 latent points to data space. We see that the algorithm is so eager to spread these projections about in data space that the mapping moves

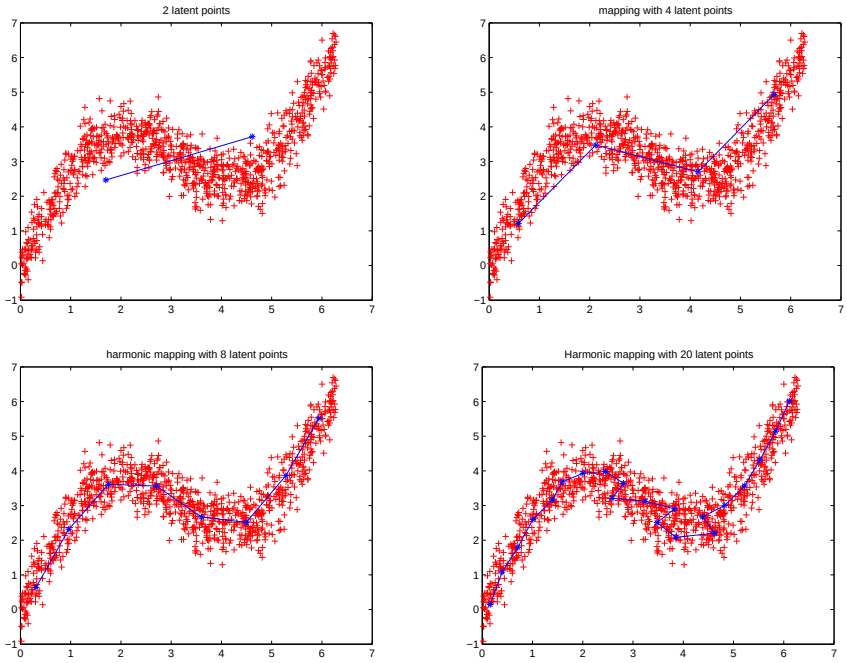


Fig. 1. The harmonic topology preserving mappings with 2, 4, 8 and 20 latent points.

across the data set rather than just along the manifold. This begins to happen with about 16 latent points and becomes more pronounced as more latent points are added.

With the standard (for illustrative purposes) data set of data drawn from a uniform distribution in [-1,1]×[-1,1], we get the results from a grid of latent points in a two dimensional latent space shown in Figure 2. We see that the mapping loses its shape fairly quickly. We consider this as evidence of an over-responsiveness to the data so that the structure of the latent space is very strongly deformed in its projection in data space.

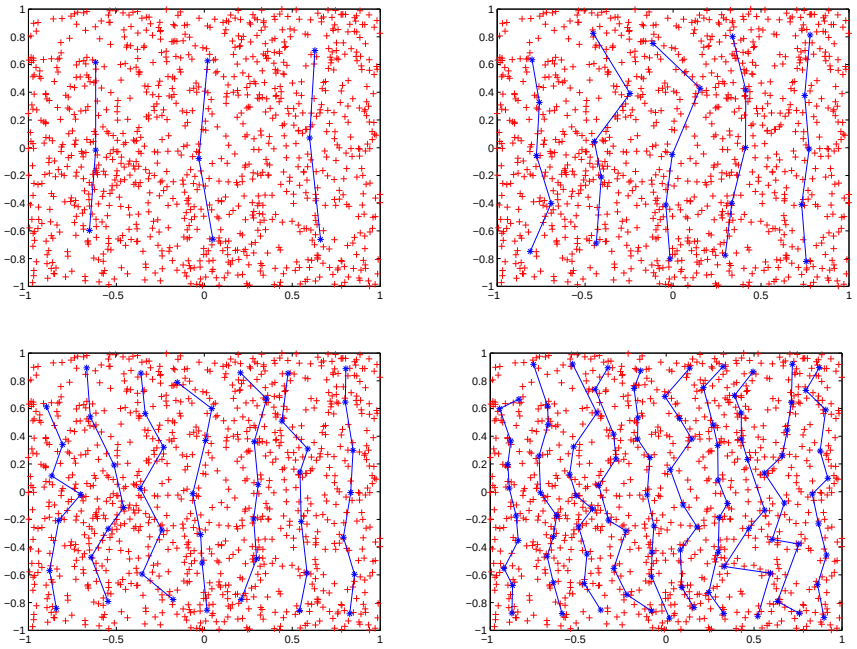


Fig. 2. The harmonic topology preserving mappings with grids of size 3*3, 5*5, 7*7 and 10*10.

The Algae data set As a visualization technique the HaToM has one advantage over the standard SOM: the projections of the data onto the grid need not be solely to the grid nodes. We illustrate the method as a visualisation technique on a set of 118 samples from a scientific study of various forms of algae some of which have been manually identified. Each sample is recorded as an 18 dimensional vector representing the magnitudes of various pigments. 72 samples have been identified as belonging to a specific class of algae which are labelled from

1 to 9. 46 samples have yet to be classified and these are labelled 0. We show in Figure 3 (top) the projections of the classified algae onto a two dimensional manifold from a 10*10 harmonic topographic map. All of the classes are easily identified, and we can see how the order of the classes in the projected map is the same as the order of the algae labels, which seems to imply a logical order in the labeling according to the properties of the algae. The bottom diagram in that figure shows the projection of the whole data set. From this, we conjecture that

- there are other classes in the data set which have not yet been identified.
- some of the unclassified samples belong to classes already identified.
- some may be simply outliers.

These are, however, speculations on our part and must be validated by a scientist with biological expertise.

If we compare this map with the equivalent map from ToPoE (Figure 4)or the GTM, we see that this map is far more spread out than the other maps; the data points' projections into this space are more diffuse and so more of the space is being used for discriminating the data. Also the unclassified points' projections show an interesting structure composed of a central cluster and two extruding arms, the meaning of which would have to be the subject of a biologist's investigation.

The GTM makes a very accurate classification: we see that the responsibilities for data points are very confidently assigned in that individual classes tend to be allocated to a single latent point. This, however works against the GTM in that, even with zooming in to the map, one cannot sometimes disambiguate the two different classes such as at the points (1,-1) and (1,1). This was not alleviated by using regularisation in the GTM though we should point out that we have a very powerful model for a rather small data set.

3.4 Generalised Harmony Learning

[7] generalises the model in Section 3.2 using

$$Perf_{HA} = \sum_{i=1}^{N} \frac{K}{\sum_{k=1}^{K} \frac{1}{d(\mathbf{x}_i, \mathbf{m}_k)^p}} \tag{7}$$

Then we wish to move the centres using gradient descent on this performance function

$$\frac{\partial Perf_{HA}}{\partial \mathbf{m}_k} = -K \sum_{i=1}^{N} \frac{2p(\mathbf{x}_i - \mathbf{m}_k)}{d(\mathbf{x}_i, \mathbf{m}_k)^{p+2} \{\sum_{l=1}^{K} \frac{1}{d(\mathbf{x}_i, \mathbf{m}_l)^p}\}^2} \tag{8}$$

Solving now for $\mathbf{m}_k$, we get

$$\mathbf{m}_k = \frac{\sum_{i=1}^{N} \frac{1}{d_{i,k}^{p+2} (\sum_{l=1}^{K} \frac{1}{d_{i,l}^{p}})^2} \mathbf{x}_i}{\sum_{i=1}^{N} \frac{1}{d_{i,k}^{p+2} (\sum_{l=1}^{K} \frac{1}{d_{i,l}^{p}})^2}} \tag{9}$$

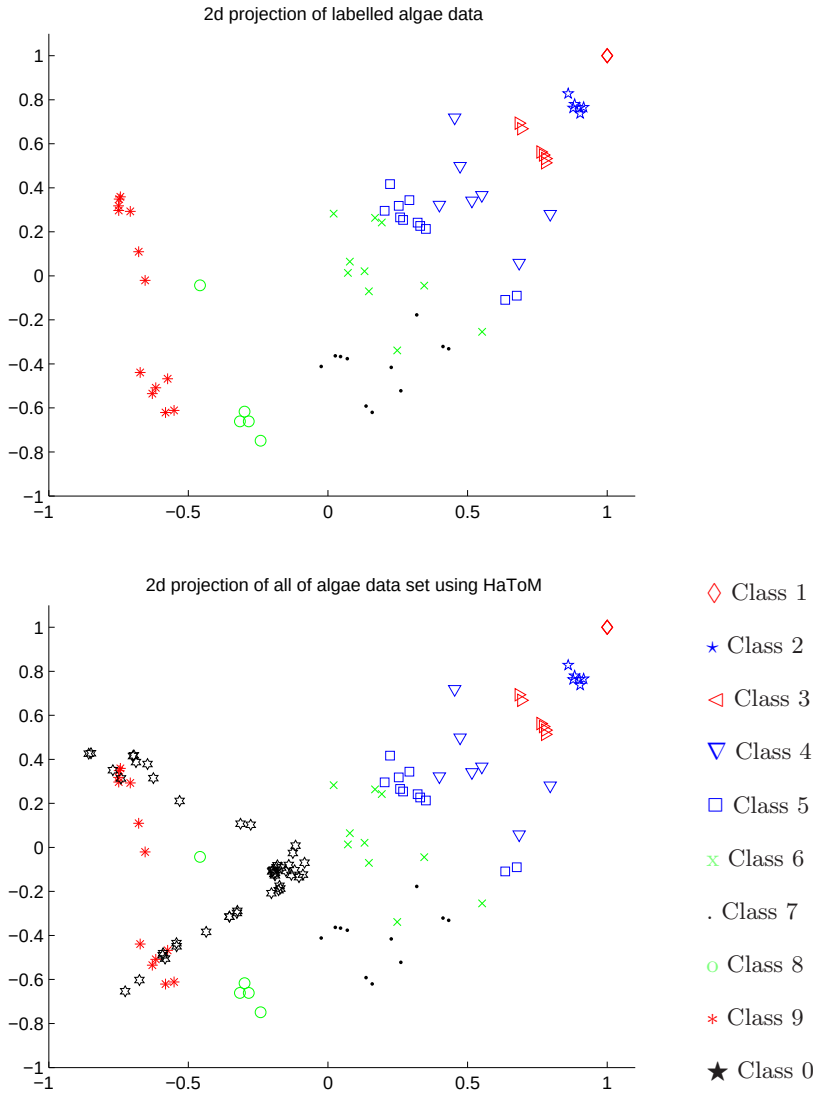


Fig. 3. Top: the projection of the 9 labelled classes on a harmonic mapping with a 2 dimensional set of 10*10 latent points. Bottom: the projection of the whole data set.

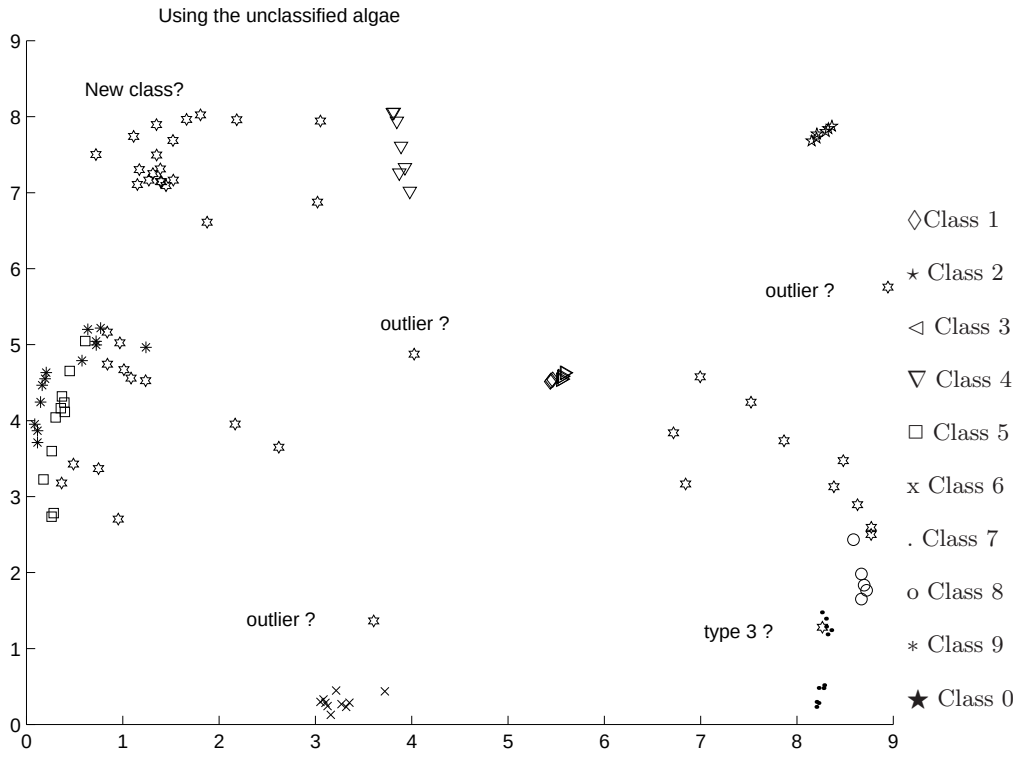


Fig. 4. The projection of the whole data set by the ToPoE.

with which we can readily replace (5) in the above algorithm.

[7] shows how this algorithm with $p > 2$ acts like boosting for supervised learning: data points which are not well represented by the K-Harmonic Means are given greater priority in the recalculation of the positions of the means. Since the current data sets are already well covered by the HaToM, we are currently seeking especially difficult data sets to investigate this effect. This will be the subject of a future paper.

4 Conclusion

We have discussed a model which uses latent points which have some structure in an underlying latent space. We have investigated projecting these latent points into data space by mapping them through a nonlinear basis and then taking linear combinations of this to map a data set. We have trained the weights of this mapping by adapting the Harmonic K-Means algorithm which is extensively shown in [8] to converge to better solutions than K-means or the mixture of experts. In practice, we may show that HaToM is more data driven than ToPoE.

The fact of being more data driven is not necessarily a good thing. If we wish to emphasise the low dimensionality of a data set, then allowing the mapping to spread may reduce insight into a low dimensional manifold. On the other hand, we have shown with the algae data set that more insight into a data set can be achieved through diverse mappings.

References

1. C. M. Bishop, M. Svensen, and C. K. I. Williams. Gtm: The generative topographic mapping. *Neural Computation*, 1997.
2. C. Fyfe. The topographic product of experts. In *International Conference on Artificial Neural Networks, ICANN2005*, 2005.
3. T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.
4. R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991.
5. M.I. Jordan and R.A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6:181–214, 1994.
6. Tuevo Kohonen. *Self-Organising Maps*. Springer, 1995.
7. B. Zhang. Generalized k-harmonic means – boosting in unsupervised learning. Technical report, HP Laboratories, Palo Alto, October 2000.
8. B. Zhang, M. Hsu, and U. Dayal. K-harmonic means - a data clustering algorithm. Technical report, HP Laboratories, Palo Alto, October 1999.

Machine Learning

An Assessment of Case Base Reasoning for Short Text Message Classification

Matt Healy[1], Sarah Jane Delany[1], and Anton Zamolotskikh[2]

[1] Dublin Institute of Technology,
Kevin Street, Dublin 8, Ireland
matt.healy@student.dit.ie, sarahjane.delany@comp.dit.ie
[2] University of Dublin, Trinity College,
Dublin 2, Ireland
zamolota@cs.tcd.ie

Abstract. Message classification is a text classification task that has provoked much interest in machine learning. One aspect of message classification that presents a particular challenge is the classification of short text messages. This paper presents an assessment of applying a case-based reasoning approach that was developed for long text messages (specifically spam filtering) to short text messages. The evaluation involves determining the most appropriate feature types and feature representation for short text messages and then comparing the performance of the case-based classifier with both a Naïve Bayes classifier and a Support Vector Machine. Our evaluation shows that short text messages require different features and even different classifiers than long text messages. A machine learner which is to classify text messages will require some level of configuration in these aspects.

1 Introduction

Message classification is a text classification task that has provoked much interest in machine learning [1–4]. One aspect of message classification that presents a particular challenge is the classification of short text messages where the *signature* of the concept being learned is weak. This is an important issue as there are a number of message classification application areas where short text is inherent. The classic example is the Short Message Service (SMS) where a text message has a limit of 160 characters. Other application areas include classification of text entered into comment boxes, online or otherwise.

This paper presents an assessment of using case-based reasoning for the classification of short text messages. Our earlier work on message classification has been in the area of spam filtering [3, 5]. We propose in this paper, to evaluate and extend this case-based approach to the classification of short text messages. We evaluate the case-based classifier on a number of datasets of two types, SMS messages with classification of spam and nonspam and customer comments with classifications of satisfactory and unsatisfactory from guests of a large hotel chain. We assess how the configuration of the case-based classifier (with respect

to the feature representation, feature selection and case selection policies) differs from that used for email (long) text messages.

This paper begins with an overview of existing research into short text message classification in Section 2. Section 3 then describes the case-based approach that we use. Section 4 discusses the evaluation of the case-based approach on short text messages, describing the datasets used, the evaluation methodology and the most appropriate configuration of the classifier for short text. In Section 5 we compare the performance of the case-based classifier to two other machine learning approaches, Naïve Bayes (NB) and Support Vector Machines (SVM) both of which are popular in text classification. The paper concludes in Section 6 with directions for future work.

2 Review of Existing Short Text Classification Approaches

The machine learning approach to text classification has been studied and analysed for many years [6–9] but there has been little previous work in the short text classification domain. The techniques used for text classification work well for datasets with large documents such as scientific papers but suffer when the documents in the training corpus are short. The performance loss can be attributed to the weak signature of the concept being modeled due to the short length of the text.

Previous research into short text classification has focused on including additional information with the training data to aid the classification process. Zelikovitz's [10] approach to short text classification uses Latent Semantic Indexing (LSI). LSI is an unsupervised learner that creates a reduced vector space through singular value decomposition (SVD). Zelikovitz combines the training data with the unlabeled test examples when creating the reduced vector space. She concludes that this expanded feature space includes semantic associations that help to classify short text documents. Zelikovitz has also used unlabeled background information that is related in some way to the training data [11, 12]. For example if the classification task is to classify titles of scientific papers, the unlabeled background information used could be the abstracts of the papers in the training dataset.

3 A Case-based Approach to Text Classification

In this section we describe Email Classification Using Examples (ECUE) the case-based approach that we used to classify email into spam and legitimate email [3] and outline the design decisions that were made for long text messages.

The ECUE system extracts three types of features, these are word features (i.e. a sequence of characters separated by white space), single characters features (i.e. letters) and statistical features (e.g. the proportion of uppercase). This combination of features gave the best generalisation accuracy for email classification.

There are two possible feature representations for text features; binary (i.e. true or false, indicating that a particular feature simply exists in the text or not) and numeric (i.e. a number representing the frequency of a particular feature in the text). The ECUE system uses a binary feature representation as it was found to produce the best generalisation accuracy for spam filtering [3].

ECUE represents a case e_i in the case-base as a vector of features values, $e_i = (f_1, f_2 ... f_n, s)$, where f is a feature and s is the class. Binary feature representation for word features uses the existence rule i.e. if the feature exists in the case f_i =1 otherwise $f_i = 0$. For statistical and single character features we use the Information Gain (IG) [13] value, as calculated during the feature selection process, to determine if f_i is set to 1 or 0. This is determined by comparing the normalised frequency of the feature with the threshold value which returns the highest IG. If the normalised frequency is greater than the threshold value $f_i = 1$ otherwise $f_i = 0$. For numeric feature representation we simply use the normalised frequency of a feature.

ECUE uses a k-nearest neighbour classifier which returns the k-nearest neighbours (k-NN) that are most similar to the target case. A False Positive (FP) (a legitimate message classified incorrectly as spam) is significant for spam filtering as a legitimate email being misclassified as spam is unacceptable in most situations. To reduce the rate of FPs, ECUE uses the k-NN algorithm with unanimous voting to bias the classifier away from FPs. In unanimous voting all k-nearest neighbours have to be classified as spam before the test email (case) is classified as spam otherwise it is classified as a legitimate email. As we are looking at general message classification of short text messages, in this evaluation we use the k-NN algorithm with weighted distance voting [14].

The ECUE system uses a case-base editing technique called Competence Base Editing (CBE) [15] to manage the size of the case-base by removing noisy and redundant cases. CBE initially builds a competence model of the case-base identifying the usefulness of each case (represented by the cases that it contributes to classifying correctly) and also the damage that each case causes (represented by the cases that it causes to be misclassified). These properties are used to identify the cases to be removed. It was found that editing a case-base using CBE yields the best generalisation accuracy in the spam filtering domain [15].

4 Evaluation of CBR for Short Text Message Classification

In this evaluation there are two objectives. The first objective is to determine the most appropriate case representation for short text message classification. The second objective is to determine if the CBE editing technique is effective for short text messages.

4.1 Datasets Used

This assessment used two types of datasets that contain short text messages, a corpus of customer comments and a corpus of Short Message Service (SMS) messages.

The customer comments corpus consists of over 5000 comments from guests visiting hotels that are part of a large hotel chain. The comments are graded in the range of 1 to 3 where 1 indicates high satisfaction with the service provided and 3 indicates low satisfaction. We grouped all comments with satisfaction level 1 into the class *satisfactory* and comments with satisfaction levels of 2 and 3 into the class *unsatisfactory*. The average message length for satisfactory comments is 54 characters with an average feature frequency of 8, while the average message length for unsatisfactory comments is 74 characters with an average feature frequancy of 12. The corpus was divided into four datasets each consisting of 500 satisfactory and 500 non satisfactory comments. The comments themselves range from a few words e.g. "V good" to a detailed description of what a guest found good or bad e.g. "Enjoyed our stay in our family room immensely. Can't wait to come back." The customer comments present a particular challenge for a text classifier as the difference between satisfactory and non satisfactory comments can be slight, for example "The room was good" and "The room was not good".

The SMS corpus consists of two datasets with 100 legitimate and 100 spam messages in both. The average message length for the legitimate SMS is 95 characters with an average feature frequency of 18, while the average message length for the spam SMS is 133 characters with an average feature frequancy of 22. The legitimate SMS messages consist of personal and business text messages and the spam SMS messages contain promotional SMS messages and unwanted text alerts. While legitimate SMS messages are normally from personal corespondents and are normally short messages such as "Where are you?", spam SMS are normally from companies who are trying to offer some service or product such as "1000 Downloads 2 choose. Txt Sir to 80082 EUR3".

4.2 Evaluation Metrics

The main evaluation metric that will be reported in this evaluation is the percentage error, i.e. the percentage of test instances incorrectly classified by the classifier. For the SMS message datasets the rate of FPs will also be reported, as similar to the situation with spam filtering, the rate of FP is significant in the SMS domain, although we haven't biased the classifiers away from FPs as was done in ECUE.

4.3 Evaluation Methods

The evaluation method used for each dataset was a 10 fold cross-validation, dividing the dataset into 10 stratified divisions or folds. In this method each fold in turn is used as a test dataset while the rest of the nine folds are considered to be the training dataset. A case-base was built from each training dataset using

the top 500 features ranked using Information Gain [13]. For each fold and test set combination the performance measures were calculated for different case-base configurations (e.g. with different feature types and feature representation).

Confidence levels were calculated using McNemar's test [16] to determine whether significant differences exists between any two case-base configurations. McNemar's test has some advantages over other performance measures (e.g. paired t-test), it has a lower Type I error (the probability of incorrectly detecting a difference when no difference exists) and has a better ability to detect a deference where one exists [16].

4.4 Evaluation to Determine Case Representation

The objective of this evaluation was to determine (i) the combination of either word, statistical and/or single character features and (ii) the feature representation (binary or numeric) that gives the best generalisation accuracy.

We performed a number of experiments varying k from $k = 1$ to $k = 9$ and varying the type of features used. The combination of the types of feature we evaluated were word features only, word and statistical features, word and letter features and word, statistical and letter features. Our results indicated that for both types of dataset the types of features that perform best are word and statistical features with no letter features. This combination of feature types is different from the combination that ECUE uses for email messages. ECUE uses all three types of features to filter spam emails. It is not surprising that letter features are not predictive for the datasets we have used here. Email spammers use obfuscation to confuse email filters by including punctuation in the middle of words or by replacing certain letters such as 'i' with 1's or l's e.g. V.1:a.g.r:a. Spammers also tend to use a lot of uppercase characters e.g. I.N.V.E.S.T.M.E.N.T. This would explain why letters are so predictive. The text message and customer comments datasets use normal structured English where letters would not necessarily be as predictive. Also SMS spam is in its infancy and SMS spammers have not had to obfuscate their text messages to bypass filters yet.

Our experiments also found that a k-NN classifier with k=7 gave the best performance for the customer comments datasets and a k-NN classifier with k=3 gave the best performance for SMS messages datasets. This indicated that the signature of SMS spam is stronger and more easily differentiated from legitimate SMS messages, even with short text, than that of satisfactory and unsatisfactory customers comments. This is to be expected as customer comment messages with different classifications can differ in as little as just one single word.

Our next objective was to determine the best feature representation, either binary or numeric. We ran our experiments using the combination of feature types which gave the best generalisation accuracy (i.e. word and statistical features) and compared the results for a binary and numeric feature representation. Fig 1 shows the percentage error for binary and numeric feature representation for each of the four customer comments and the overall result across all four datasets.

The results for the customer comments dataset shows that a numeric representation of features gives better performance and reduces error across all four datasets. The difference between numeric and binary representation for dataset 1, 2 and 3 are significant at the 99.9% confidence level whereas the difference for dataset 4 and the overall result are significant at the 90% confidence level.

Fig 2 shows the percentage error and FP rate for each of the SMS datasets and the overall result across both datasets. The graphs in Fig 2 show that a binary representation of features gives better performance and reduces the error and the FP rate across all datasets. The difference in the percentage error between numeric and binary representation is only significant for dataset 1 and the overall result at the 95% confidence level whereas the difference in the FP rate is only significant at the 95% confidence level for dataset 1.

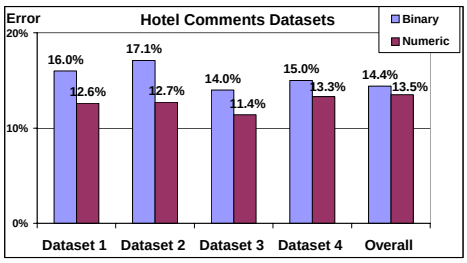


Fig. 1. Percentage error results of comparing binary and numeric feature representation for the customer comments datasets

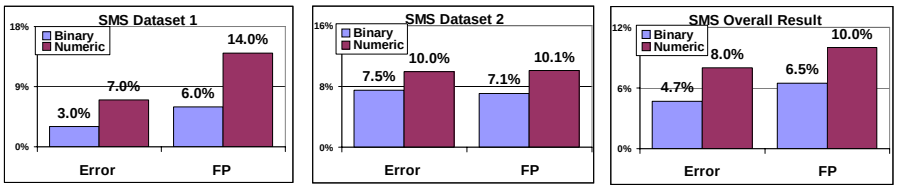


Fig. 2. Percentage error & FPs results of comparing binary and numeric feature representation for the SMS datasets

4.5 Evaluation of Case-Base Editing Technique

The objective of this evaluation was to determine whether applying the case-base editing technique (CBE) would improve the generalisation accuracy of a case-base of short text messages. Fig 3 and 4 show the results for the customer comments datasets and the SMS datasets respectively. The reduced size of each dataset is included in the figures as a percentage of the original size.

The results show that the percentage error is higher for the edited case-base compared to the unedited case-base in the customer comments datasets. The results show little difference in the percentage error or FPs for the SMS datasets. The differences for the customer comments datasets are all significant at the 95% level or higher except for dataset 2 where the difference is not significant. None of the differences on the SMS datasets are significant in any case possibly due to the limited size of the datasets.

This indicates that the editing technique CBE is not appropriate for short text message classification. One of the objectives of CBE was to conservatively reduce the size of the case-base [15]. ECUE reduces an email case-base by approximately 30%, but applying CBE to the customer comments and SMS datasets results in an overall average reduction of 56% and 65% respectively. This suggests that the sparsity of the cases due to the short text content of the messages is not appropriate for the editing technique resulting in too many cases being removed.

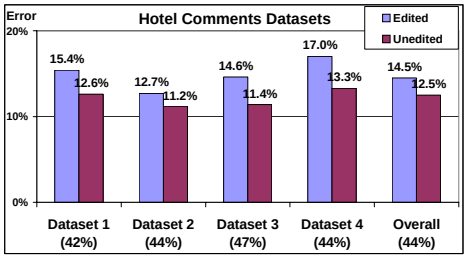


Fig. 3. Percentage error rsults of applying CBE to the customer comments datasets

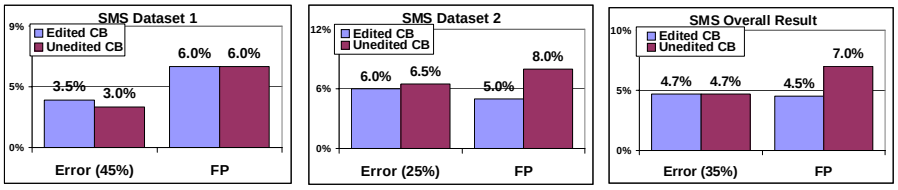


Fig. 4. Percentage error & FPs results of applying CBE to the SMS datasets

5 Comparison with Naïve Bayes and Support Vector Machines

Naïve Bayes (NB) [14] and Support Vector Machines (SVMs) [17] are both popular classifiers used in text categorisation. Our comparisons of the ECUE system

with NB for spam filtering concluded that neither classifier outperformed the other consistently but that the k-NN classifier was in fact better at handling the concept drift in the email [5]. For the purposes of general message classification, it is important to compare the performance of the k-NN classifier on short text messages with that of NB and an SVM.

We evaluated our case-based approach using a NB classifier and an SVM on each of the six datasets. The NB implementation used is given in Equation 1 below, where each email is labelled as one of a set of classifications $c_i \in C$ and is described by a set of attributes $\{a_1, a_2, \ldots a_n\}$ where a_i indicates the presence of that attribute in the document. This implementation incorporates a Laplace correction in the conditional probabilities to prevent zero probabilities dominating [3].

$$c_{NB} = \ argmax_{c_i \in C} P(c_i) \prod_j P(a_j|c_i) \tag{1}$$

The SVM implementation used is a 2-norm soft-margin SVM as described in [17] with a normalised dot product kernel function. The results are displayed in Fig 5 and 6 respectively.

It is evident from Fig 5 that NB and the SVM consistently outperform the k-NN classifier for both datasets with a lower percentage error in all cases. The differences in percentage error between NB and k-NN are significant at the 99.9% level in all cases, whereas the differences in percentage error between the SVM and k-NN are significant at the 99% level or higher in all cases except dataset 3 where there is no significant difference. This is contrary to what was found for email messages. It was found that the case-based approach performed better than NB in the email domain. Email, both spam and legitimate, is a diverse concept; spam offering cheap prescription drugs has little in common with spam covering investment opportunities and personal email messages will be quite different from business email. This suggests that the lack of diversity in the customer comments datasets is not appropriate for a local learner like k-NN but is more appropriate for a classifier that uses a globalconcept like NB or SVM.

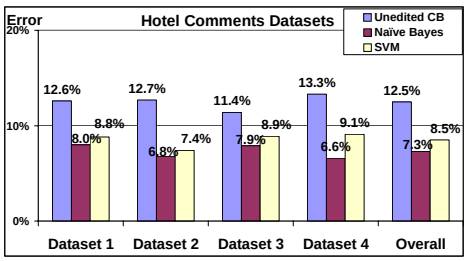


Fig. 5. Percentage error results of comparing the k-NN classifier with Naïve Bayes and SVM on the customer comments datasets

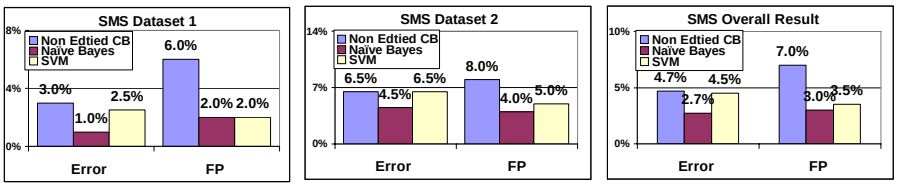


Fig. 6. Percentage error & PFs results of comparing the k-NN classifier with Naïve Bayes and SVM on the SMS datasets

The SMS figures reported in Fig 6 show similar results in that both NB and SVM outperform k-NN in all cases with equal or lower overall error rates and lower FP rates. In spite of some of the differences being large, (e.g. on dataset 1 the k-NN FP rate of 6% compared with a NB or SVM FP rate of 2%), none of the differences are significant using McNemar's test. This is most probably due to the small numbers of messages in the SMS datasets under consideration. It will be important to source a significant number of SMS text messages, both spam and legitimate, to confirm these results.

6 Conclusions and Future Work

In this paper we have identified that the most appropriate types of feature to use for short text message classification are words and statistical features. Unlike longer email message classification, letter features do not improve performance. We have also shown that the feature representation used is dependent on the domain and types of data that are being classified. Email and text messages require a binary representation but the classification of the customer comment text messages requires a numeric representation which includes feature frequency information.

The results of our evaluations presented here have shown that short text messages require different feature representation, different feature types and even different classifiers than longer email messages to achieve best performance. This suggests that any machine learning system which is to classify text messages needs to be configurable in all these respects. The configuration could be automatically performed using the data on which the system will be trained.

Our future work in this area is to extend such a configurable system to cater for multiple classifications to facilitate such applications as message routing or general email filtering.

References

1. Busemann, S., Schmeier, S., Arens, R.G.: Message classification in the call center. In: Procs of the 6th conference on Applied Natural Language Processing, Morgan Kaufmann (2000)

2. Neumann, G., Schmeier, S.: Combining shallow text processing and machine learning in real world applications. In: Proc of the 16th International Joint Conference on Artificial Intelligence (IJCAI '99). Workshop on Machine Learning for Information Filtering, Stockholm, Sweden (1999)
3. Delany, S., Cunningham, P., Coyle, L.: An assessment of case-based reasoning for spam filtering. In McGinty, L., Crean, B., eds.: Procs. of 15th Irish Conference on Artificial Intelligence and Cognitive Science. (2004) 9–18
4. Androutsopoulos, I., Koutsias, J., Chandrinos, K.V., Spyropoulos, C.D.: An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages. In: SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, ACM Press (2000) 160–167
5. Delany, S.J., Cunningham, P., Tsymbal, A., Coyle, L.: A case-based technique for tracking concept drift in spam filtering. In Macintosh, A., Ellis, R., Allen, T., eds.: Applications and Innovations in Intelligent Systems XII, Procs. of AI 2004, Springer (2004) 3–16
6. Sebastiani, F.: Machine learning in automated text categorization. In: CM Computing Surveys. (2002) 1–47
7. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: Proc of the 10th ECML, Springer (1999)
8. Cohen, W., Singer, Y.: Context-sensitive learning methods for text categorization. In: Proc of SIGIR-96. (1996)
9. Lewis, D.: Feature selection and feature extraction for text categorization. In: Proceedings of Speech and Natural Language Workshop. (1992) 212–217
10. Zelikovitz, S.: Transductive LSI for short text classification problems. In: Proceedings of the 17th International FLAIRS Conference. (2004)
11. Zelikovitz, S., Hirsh, H.: Improving short-text classification using unlabeled background knowledge to assess document similarity. In: Proceedings of the Seventeenth International Conference on Machine Learning (ICML). (2000)
12. Zelikovitz, S., Hirsh, H.: Using LSI for text classification in the presence of background text. In: Proceedings for the Conference on Information and Knowledge Management (CIKM). (2001)
13. Quinlan, J.R.: C4.5 Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Mateo, CA. (1997)
14. Mitchell, T.: Machine Learning. McGraw Hill, New York (1997)
15. Delany, S.J., Cunningham, P.: An analysis of case-based editing in a spam filtering system. In Funk, P., P.González-Calero, eds.: 7th European Conference on Case-Based Reasoning (ECCBR 2004). Volume 3155 of LNAI., Springer (2004) 128–141
16. Dietterich, D.T.: Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computing **10** (1998) 1895–1923
17. Christianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods. Cambridge University Press (2000)

Information Extraction from Calls for Papers with Conditional Random Fields and Layout Features

Karl-Michael Schneider

University of Passau, Department of General Linguistics, 94030 Passau, Germany
schneide@phil.uni-passau.de

Abstract. For members of the research community it is vital to stay informed about conferences, workshops, and other research meetings relevant to their field. These events are typically announced in call for papers (CFP) that are distributed via mailing lists. We employ Conditional Random Fields for the task of extracting key information such as conference names, titles, dates, locations and submission deadlines from CFPs. Extracting this information from CFPs automatically has applications in building automated conference calendars and search engines for CFPs. We combine a variety of features, including generic token classes, domain-specific dictionaries and layout features. Layout features prove particularly useful in the absence of grammatical structure, improving average F1 by 30% in our experiments.

1 Introduction

People actively involved in scienti c research rely on information about academic conferences, workshops, etc. in order to know when and where to publish their work. This information is typically distributed via mailing lists in so-called *call for papers* (CFP). CFPs invite the submission of papers, abstracts, posters, demos and the like and specify the date and place of an event, the deadline for paper submission, relevant topics, program committee members, contact addresses, and a meeting website, among others. Prospective authors, on the other hand, rely on this information to nd appropriate conferences to submit their papers ready for submission in due time.

Besides CFPs sent on mailing lists, conference calendars, specialised search engines and digital libraries of CFPs such as EventSeer (http://www.eventseer.net/) are useful tools for researchers. Building such online services requires techniques to extract the key information from CFPs automatically, in order to make this information accessible in a structured manner, e.g. by searching in different elds and browsing lists of CFPs ordered by date, place, deadline etc. The value of such online services depends crucially on the accuracy of the extraction techniques.

A lot of research in information extraction has focused on extracting facts from texts consisting of complete sentences such as news articles [1]. These methods rely on the grammatical structure of sentences by applying automatic tools such as POS taggers and syntactic parsers. Calls for papers are different they often consist of grammatical text that is interspersed with fragments of text that do not contain complete sentences and lack any grammatical structure. These latter sections usually contain the important

information about an event. They can be recognised visually by their physical layout, such as indented or centered lines, double-spaced lines, blank lines separating them from the rest of the text, as well as particular orthographic properties like capitalised and uppercase words. Also, many CFPs follow information-theoretic or communication-theoretic principles by placing the most important information at the beginning of the text.

This paper presents an approach for information extraction from CFPs that integrates various kinds of evidence from both content (i.e. tokens in a text) and layout (i.e. the physical structure of a text) by using *conditional random fields* (CRFs) [2]. CRFs are discriminatively-trained undirected graphical models. Like maximum entropy models they are based on an exponential form and thus can combine overlapping, non-independent features very easily. CRFs have been applied successfully to a variety of sequence labelling tasks such as shallow parsing [3], named entity recognition [4], information extraction [5] and table recognition [6].

The features we use measure generic properties of tokens (capitalisation, spelling), membership in particular token classes (years, month names, URLs, E-mail addresses), domain-specific vocabulary through dictionaries, location names through gazetteer lists, and layout properties such as empty and indented lines and the position of tokens in lines. We present experimental results on a corpus of hand-tagged CFPs using various subsets of features to measure the impact of different feature classes on extraction accuracy. Layout features prove particularly useful, improving accuracy dramatically.

2 Related Work

Layout features have been used previously in a variety of information extraction tasks. In [5] a CRF is trained to extract various fields (such as author, title, etc.) from the header sections of research papers using a combination of linguistic and layout features. The features are very similar to ours. CFPs are similar to research papers in that most (though not all) of the important information is contained in highly formatted regions (the header section at the beginning) rather than in grammatical sentences. An important difference between this task and ours is that research paper headers consist only of header fields, with no intervening material. In contrast, the field instances in a CFP comprise only a small fraction of the tokens, making extraction a harder task. Moreover, many papers use standardised document layouts (e.g. through the use of LaTeX style files), whereas CFPs exhibit greater variation in form and layout.

In [6] layout features are used to locate tables in text, identify header and data cells and associate data cells with their corresponding header cells. They use a large variety of layout features that measure the occurrence of various amounts of whitespace indicative of table rows in text lines. Layout features such as *line begins with punctuation* and *line is the last line* are also used to learn to detect and extract signature lines and reply lines in E-mails [7]. In both tasks an input text (web page with tables, E-mail) are considered sequences of lines rather than sequences of tokens, and features measure properties of lines. In contrast, we use features that measure properties of both lines and tokens.

In [8] a conditional Markov model (CMM) tagger and a CRF are trained to extract up to 11 fields from workshop calls for papers using various token features, includ-

ing orthography, POS tags and named entity tags, but no layout features.[1] In addition, domain knowledge is employed to find matching workshop acronym/name pairs and select the correct workshop date (e.g. one that occurs after the paper submission date). This improves performance over the CMM but not over the CRF.

3 Conditional Random Fields

Conditional random fields are undirected discriminatively-trained graphical models. A special case of a CRF is a linear chain, which corresponds to a conditionally-trained finite state machine. A linear-chain CRF is trained to maximise the conditional probability of a label sequence given an input sequence. As in Maximum-Entropy Markov Models (MEMM) [9], this conditional probability has an exponential form which makes it easy to integrate many overlapping, non-independent features. MEMMs maximise the conditional probability of each state given the previous state and an observation, which makes them prone to the *label bias problem* [2]. CRFs use a global exponential model to avoid this problem.

Let $\mathbf{x} = x_1 \ldots x_T$ be an input sequence and $\mathbf{y} = y_1 \ldots y_T$ be a corresponding state (or label) sequence. A CRF with parameters $\Lambda = \{\lambda, \ldots\}$ defines a conditional probability for $\mathbf{y}$ given $\mathbf{x}$ to be

$$P_\Lambda(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_\mathbf{x}} \exp\left(\sum_{t=1}^{T} \sum_k \lambda_k f_k(y_{t-1}, y_t, \mathbf{x}, t)\right) \quad , \tag{1}$$

where $Z_\mathbf{x}$ is a normalisation constant that makes the probabilities of all label sequences sum to one, $f_k(y_{t-1}, y_t, \mathbf{x}, t)$ is a feature function, and λ_k is a learned weight associated with f_k. A feature function indicates the occurrence of an event consisting of a state transition $y_{t-1} \rightarrow y_t$ and a query to the input sequence $\mathbf{x}$ centered at the current time step t. For example, a feature function might have value 1 if the current state, y_t, is B-TI (indicating the beginning of a conference title) and the previous state, y_{t-1}, is O (meaning "not belonging to any entity") and the current word, x_t, is "Fifth", and value 0 otherwise. The weight λ_k for the feature function f_k indicates how likely the event is to occur.

Inference in CRFs is done by finding the most probable label sequence, $\mathbf{y}^*$, for an input sequence, $\mathbf{x}$, given the model in (1):

$$\mathbf{y}^* = \arg\max_\mathbf{y} P_\Lambda(\mathbf{y}|\mathbf{x}) \quad .$$

This can be calculated efficiently by dynamic programming using the Viterbi algorithm, similarly to inference in HMMs.

[1] Unfortunately we became aware of this work too late to be able to obtain the corpus and test our system on it before publication of this paper.

During training, the weights λ_k are set to maximise the conditional log-likelihood of a set of labelled sequences in a training set $D = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}) : i = 1, \ldots, M\}$:

$$\begin{aligned} LL(D) &= \sum_{i=1}^{M} \log P_\Lambda(\mathbf{y}^{(i)}, \mathbf{x}^{(i)}) - \sum_k \frac{\lambda_k^2}{2\sigma_k^2} \\ &= \sum_{i=1}^{M} \left(\sum_{t=1}^{T} \sum_k \lambda_k f_k(y_{t-1}^{(i)}, y_t^{(i)}, \mathbf{x}^{(i)}, t) - \log Z_{\mathbf{x}^{(i)}} \right) - \sum_k \frac{\lambda_k^2}{2\sigma_k^2} \quad . \end{aligned} \tag{2}$$

The term $\sum_k \frac{\lambda_k^2}{2\sigma_k^2}$ is a Gaussian prior that is used for penalising the log-likelihood in order to avoid over- tting, and σ_k^2 is a variance [5]. Maximising (2) corresponds to matching the expected count of each feature according to the model to its adjusted empirical count:

$$\sum_{i=1}^{M} \sum_{t=1}^{T} f_k(y_{t-1}^{(i)}, y_t^{(i)}, \mathbf{x}^{(i)}, t) - \frac{\lambda_k}{\sigma_k^2} = \sum_{i=1}^{M} \sum_{\mathbf{y}'} P_\Lambda(\mathbf{y}' | \mathbf{x}^{(i)}) \sum_{t=1}^{T} f_k(y'_{t-1}, y'_t, \mathbf{x}^{(i)}, t) \quad .$$

The terms $\frac{\lambda_k}{\sigma_k^2}$ are used to discount the empirical feature counts. In [5] several alternative priors for regularisation in CRFs were investigated but the Gaussian prior was found to work best.

Finding the parameter set Λ that maximises the log-likelihood in (2) is done using an iterative procedure called *limited-memory quasi-Newton* (L-BFGS) [3]. Since the log-likelihood function in a linear-chain CRF is convex[2] the learning procedure is guaranteed to converge to a global maximum. CRFs could also be trained using traditional maximum entropy learning algorithms, such as GIS and IIS [10], but BFGS was shown to converge much faster [3].

4 Information Extraction from Calls for Papers

4.1 Task and Approach

We extract up to seven elds from a CFP: Name (e.g. *ACL 2005*), Title (e.g. *42nd Annual Meeting of the Association for Computational Linguistics*), Date, Location, URL, Deadline, and Conjoined (i.e. the name and title of the main conference if the event is part of a larger conference, e.g. a workshop held in conjunction with a conference).

We follow the standard methodology used in shallow parsing, named entity recognition and similar tasks and represent our extraction problem as a sequence labelling task. Following [11], each token in a text is marked as being either the beginning of an entity, inside an entity but not at the beginning, or not part of any entity. For example, the rst token in a conference title is labelled with B-TI and all subsequent tokens of the title are labelled with I-TI, and likewise for other entities. Tokens outside of any entity are labelled with O. Thus the information extraction problem can be seen as a token classi cation task, subject to the further constraint that I-*entity* can only follow B-*entity* or I-*entity*.

[2] Assuming a one-to-one correspondence between states and labels, as we do.

Each token is represented as a set of binary features that describe lexical, contextual and spatial properties of the token. Our features are summarised in Table 1. We use a linear-chain CRF to learn a labelling function from training examples and label new text. Below we describe the features used for our task.

Table 1. Description of features

Type	Feature	Definition	Example
generic	ICAP	capitalised	*International*
generic	ACAP	all uppercase	*EACL*
generic	SCAP	single uppercase letter	*A.*
generic	MCAP	mixed case	*PostScript*
generic	ADIG	all digits	*2005*
generic	PUNC	punctuation symbol	*.*
generic	URL	regular expression for URL	*http://www.aclweb.org/*
generic	EMAIL	regular expression for E-mail address	*aics05@ulster.ac.uk*
generic	HASUP	token contains uppercase letter	*T'sou*
generic	HASDIG	token contains digit	*+49(*
generic	HASDASH	token contains -	*17-21*
generic	HASPUNC	token contains punctuation symbol	*llncs.cls*
generic	ABBR	word ends with period	*Prof.*
domain	CNAME	conference name	*ACL'03*
domain	CNUMY	conference number or year	*'03*
domain	DAY	day of week or day of month	*16*, *Sunday*
domain	DAYS	range of days	*17-23*, *4th-6th*
domain	YEAR	four-digit year	*2003*
domain	SYEAR	two-digit year	*01*
domain	ROM	roman number	*IX*
domain	NTH	number attribute	*third*, *9th*
layout	BOL	first token in the line	
layout	EOL	last token in the line	
layout	BOT	first line in the text	
layout	EOT	last line in the text	
layout	BLANK	line contains no visible characters	
layout	PUNCTLN	line contains only punctuation characters	
layout	INDENT	line is indented	
layout	FIRST10	first 10 lines in the text	
layout	FIRST20	first 20 lines in the text	

4.2 Token features

Token features describe properties of individual tokens and their surrounding tokens. We use generic (i.e. domain independent) as well as domain dependent features. We use a variety of information sources to extract features from tokens:

– Orthographic properties are used to assign each token to one or more generic token classes.
– Each token is a feature by itself; however, we map capitalised words (ICAP) and words consisting of all uppercase letters (ACAP) to lowercase.
– We use a dictionary to recognise names of months and week days.
– Another (domain dependent) dictionary is used to recognise words that often occur as part of conference titles, such as *Conference*, *Workshop*, *International*, *on*, and capitalised words that regularly occur in CFPs but are rarely used in conference titles, e.g. *Call*, *Deadline*, *LaTeX* (see Table 2).
– A gazetteer list[3] is used to recognise name of cities, towns, countries, and other known locations. We look up sequences of up to ve consecutive tokens in the gazetteer and assign a feature to each token of a matching sequence.

Table 2. Domain dictionary

Class	Words
INST	*University*, *Center*, *Institute*, *School*
ORG	*Society*, *Association*, *Council*, *Consortium*, *Group*
EV	*Conference*, *Workshop*, *Symposium*, *Meeting*, *Congress*, *Track*, *Colloquium*, ...
ATTR	*Annual*, *Interdisciplinary*, *Special*, *Joint*, *European*, *International*, *National*, ...
DL	*Deadline*, *Reminder*, *Submission*, *due*
TH	*st*, *nd*, *rd*, *th*

In addition to the features representing a token, we add the features of the surrounding tokens within a window size of 2 (marked accordingly) to represent the context of the token. For example, for the token *9th* in the sequence *Call for Papers 9th EUROPEAN WORKSHOP ON NATURAL LANGUAGE GENERATION* we would extract the features *W=9th, HASDIG, DAY, NTH, W-1=papers, ICAP-1, D_NONAME-1, W-2=for, D_FOR-2, W+1=european, ACAP+1, D_ATTR+1, W+2=workshop, ACAP+2, D_EV+2*.

4.3 Layout Features

Layout features encode information about the position of a token in a line of text, such as beginning and end of line, as well as properties of whole lines in a text, such as rst/last lines and blank lines (see Table 1). For each token we add the layout features of the token and of the line in which the token occurs to the token's feature set, as well as the features of 2 preceeding and following lines. For example, the feature set *BOL, FIRST10, FIRST20, INDENT, FIRST10-1, FIRST20-1, BLANK-1, BOT-2, FIRST10-2, FIRST20-2, INDENT-2, FIRST10+1, FIRST20+1, BLANK+1, FIRST10+2, FIRST20+2, INDENT+2* would indicate that the current token appears at the beginning of a line; the current line (the line containing the current token) is the

[3] obtained from http://www.world-gazetteer.com/

third line; the previous and next line are empty; the current line and the lines two lines up (rst) and down (fth) are indented; and all of them are among the rst 10 and rst 20 lines in the text.[4]

5 Experiments

5.1 Dataset

The data consists of 263 CFPs received by the author from various mailing lists between August 2002 and January 2004, and from February 2005 to May 2005. We use only the plain text part of each message and remove mailing list signatures and email headers occurring in the text (e.g. due to manual forwarding and editing by list moderators). We avoid duplicate and near duplicate CFPs by computing their Nilsimsa digest[5] and removing all but one CFP if the number of bits that are equal in two digests is greater than 230 (90%).

We apply only minimal tokenisation. We separate punctuation, double quotes and parentheses from preceeding and following words but do not separate a period from the preceeding word if the word is a single capital letter or appears on a hand-crafted list of known abbreviations (*Dr*, *Prof*, *Int*, etc.). Also, we do not separate dashes and single quotes from preceeding and following material because these symbols are often part of conference names, e.g. *ACL'05*, *ICML-2005*.

Each CFP has been manually annotated for the seven elds described in Sect. 4.1. To reduce the amount of manual work we use an iterative procedure by training a CRF on a small number of manually annotated CFPs, then using this model to annotate more CFPs, correcting any errors manually, retraining the model, and so on. The total number of tokens is 203,151, with 7,217 tokens (3.6%) belonging to eld instances.

For the experiments, we split the data into a training and testing set. We use the rst 128 CFPs (from August 2002 to January 2004) for training and the remaining 135 CFPs (from February 2005 to May 2005) for testing.

5.2 Performance Measure

Following [5] we measure performance using two different sets of metrics: word-based and instance-based. For word-based evaluation, we de ne *TP* as the number of distinct words in all hand-tagged instances of a eld that occur in at least one extracted instance of that eld; *FN* as the number of distinct words in hand-tagged instances that do not occur in an extracted instance; and *FP* as the number of distinct words in all extracted instances of a eld that do not occur in at least one hand-tagged instance of the eld. These counts are summed over all CFPs in the test set. Word precision, recall and F1 are de ned as $prec = \frac{TP}{TP+FP}$, $recall = \frac{TP}{TP+FN}$, $F1 = \frac{2 \times prec \times recall}{prec+recall}$.

[4] Note that the feature BLANK can never occur (because all features occur with tokens, and no token occurs in a blank line). However, features BLANK-*i* and BLANK+*i* represent valuable information about the physical layout of the text.

[5] http://ixazon.dynip.com/ cmeclax/nilsimsa.html

Instance-based evaluation considers an extracted instance correct only if it is identical to a hand-tagged instance of the same eld. Thus in instance-based evaluation an extracted instance with even a single added or missing word is counted as an error. Instance precision and instance recall are the percentage of extracted instances of a eld that are identical to a hand-tagged instance, and the percentage of hand-tagged instances that are extracted by the CRF, respectively. Instance F1 is de ned accordingly as in word-based evaluation.

We report the word-based and instance-based measures for each eld. Overall performance is measured by calculating precision and recall from counts summed over all elds and calculating F1 from overall precision and recall (called micro average in the information retrieval literature). This favours elds that occur more frequently than others. In addition, we calculate the average of the per- eld F1 values (called macro average in the information retrieval literature). This gives equal weight to all elds.

5.3 Training CRFs

We use a Java implementation of CRFs [12]. Training with the full feature set took about four hours on an Athlon AMD 800 MHz CPU with Linux operating system and converged after 156 iterations.

6 Results

6.1 Performance Evaluation

Table 3 shows per- eld and overall performance. Word-based F1 is around 80% for most elds, except Conjoined and Name which are signi cantly lower. As expected, instance-based F1 is lower than word-based F1 for most elds, except Name which is 1.3% higher and URL which is equal to word-based F1 because URLs are single tokens. For Conjoined and Title instance-based F1 is much lower than word-based F1 (around 15 18%), presumably because on average instances of Conjoined and Title consist of more tokens than other elds, making them more prone to instance-based errors.

Table 3. Extraction results with the full feature set

Field	Instances	W-Recall	W-Precision	W-F1	I-Recall	I-Precision	I-F1
Conjoined	93	41.6%	66.1%	51.0%	28.0%	48.1%	35.4%
Date	168	72.7%	90.8%	80.8%	64.9%	79.6%	71.5%
Deadline	161	68.9%	92.0%	78.8%	59.6%	80.7%	68.6%
Location	120	72.1%	90.8%	80.4%	64.2%	82.8%	72.3%
Name	78	46.7%	78.1%	58.5%	48.7%	77.6%	59.8%
Title	136	80.9%	79.8%	80.3%	61.8%	63.6%	62.7%
URL	131	71.8%	87.9%	79.0%	71.8%	87.9%	79.0%
Micro average	887	70.2%	84.1%	76.5%	59.1%	75.8%	66.4%
Macro average				72.7%			64.2%

Notice also that performance is signi cantly lower than in [5] for the research paper extraction task. However, eld extraction from CFPs is a more dif cult task because most tokens in a CFP do not belong to a eld instance, whereas research paper headers consist only of header elds. In the CFP task there are three types of extraction errors: (i) assigning a word to the wrong eld, (ii) assigning a word that belongs to a eld to no eld, (iii) assigning a non- eld word to some eld. In the research paper task only the rst error type can occur.

6.2 Effects of Different Kinds of Features

To analyse the contribution of different kinds of features we trained four different models, using (i) only generic features, (ii) generic and domain features, (iii) generic and layout features, (iv) all features (the latter model is identical to that in the previous section). We compare the overall performance of the four models in Table 4. Both domain and layout features improve the performance over using only generic features, both individually and in combination. Using the full feature set increases instance-based macro averaged F1 by 38% (relative) over using only generic features. Layout features have the biggest impact, resulting in a 34% relative increase in F1 over the generic features and 30% over the combination of generic and domain features. Domain features alone contribute only a 6% improvement over the generic features.

Table 4. Contribution of different kinds of features

Features	generic	generic+domain	generic+layout	generic+domain+layout
micro Word-F1	58.8%	61.4%	74.9%	76.5%
macro Word-F1	54.0%	57.2%	70.4%	72.7%
micro Instance-F1	50.3%	52.6%	65.0%	66.4%
macro Instance-F1	46.4%	49.2%	62.3%	64.2%

Table 5 shows the per- eld improvement in instance-based F1 due to layout features. The biggest improvement (64% relative) is obtained for Name, and for Title and Location the relative improvement is 40%. These elds are highly correlated with spatial properties in CFPs. For the Deadline eld the improvement is relatively small (only 7%). This is due to the fact that deadlines are typically surrounded by unambiguous lexical material (in fact, the features with highest weights in the CRF for the beginning of Deadline are W-2=deadline, W-2=submissions, W-2=submission and W-2=due).

Table 5. Instance-based F1 improvements for individual fields through the use of layout features

Field	Conjoined	Date	Deadline	Location	Name	Title	URL
without layout	29.2%	62.6%	64.0%	50.0%	36.4%	43.5%	58.8%
with layout	35.4%	71.5%	68.6%	72.3%	59.8%	62.7%	79.0%

7 Conclusions and Future Work

This paper applies conditional random fields to a practical problem: extracting important knowledge from call for papers for academic conferences and related events. We demonstrate the effectiveness of layout features in the absence of grammatical structure, which is typical for those regions in CFPs that contain the key information about an event, obtaining an improvement in instance-based average F1 by 30%.

Extraction performance in our experiments is reasonable but not optimal, probably due to the relatively small training corpus. Increasing the amount of training data would be expected to help improve the performance. However, annotating training data manually is labour-intensive. In future work we intend to employ bootstrapping [13] to reduce the amount of manual work in obtaining training data.

References

1. Riloff, E., Jones, R.: Learning dictionaries for information extraction by multi-level bootstrapping. In: Proc. 16th National Conference on Artificial Intelligence and 11th Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI 1999), Orlando, Florida, AAAI Press (1999) 474–479
2. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proc. 18th International Conference on Machine Learning (ICML-2001), San Francisco, CA, Morgan Kaufmann (2001) 282–289
3. Sha, F., Pereira, F.C.N.: Shallow parsing with conditional random fields. In: Proc. HLT-NAACL 2003, Edmonton, Canada (2003) 134–141
4. Settles, B.: Biomedical named entity recognition using conditional random fields and rich feature sets. In: Proc. Int. Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA), Geneva, Switzerland (2004) 104–107
5. Peng, F., McCallum, A.: Accurate information extraction from research papers using conditional random fields. In: Proc. HLT-NAACL 2004, Boston, Massachusetts (2004) 329–336
6. Pinto, D., McCallum, A., Wei, X., Croft, W.B.: Table extraction using conditional random fields. In: Proc. 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2003), Toronto, Canada (2003) 235–242
7. Carvalho, V.R., Cohen, W.W.: Learning to extract signature and reply lines from email. In: Prod. First Conference on Email and Anti-Spam (CEAS), Mountain View, CA (2004)
8. Cox, C., Nicolson, J., Finkel, J.R., Manning, C., Langley, P.: Template sampling for leveraging domain knowledge in information extraction. In: PASCAL Challenges Workshop, Southampton, U.K. (2005)
9. McCallum, A., Freitag, D., Pereira, F.: Maximum entropy markov models for information extraction and segmentation. In: Proc. 17th International Conference on Machine Learning (ICML-2000), San Francisco, CA, Morgan Kaufmann (2000) 591–598
10. Della Pietra, S., Della Pietra, V.J., Lafferty, J.: Inducing features of random fields. IEEE Trans. on Pattern Analysis and Machine Intelligence **19** (1997) 380–393
11. Ramshaw, L.A., Marcus, M.P.: Text chunking using transformation-based learning. In: Proc. ACL Third Workshop on Very Large Corpora. (1995) 82–94
12. McCallum, A.K.: MALLET: A machine learning for language toolkit. http://mallet.cs.umass.edu/ (2002)
13. Lin, W., Yangarber, R., Grishman, R.: Bootstrapped learning of semantic classes from positive and negative examples. In: Proc. ICML-2003 Workshop on The Continuum from Labeled to Unlabeled Data, Washington, DC (2003) 103–110

Investigation into the use of PCA with Machine Learning for the Identification of Narcotics based on Raman Spectroscopy

Tom Howley, Michael G. Madden, Marie-Louise O'Connell and Alan G. Ryder

National University of Ireland, Galway,
thowley@vega.it.nuigalway.ie, michael.madden@nuigalway.ie,
ML.OConnell@nuigalway.ie, alan.ryder@nuigalway.ie

Abstract. The identification of narcotics using high dimensional spectral data poses an interesting challenge to machine learning, as the presence of high numbers of redundant or highly correlated attributes can seriously degrade classification accuracy. This paper investigates the use of Principal Component Analysis (PCA) to reduce spectral data and to improve the predictive performance of some well-known machine learning methods. Experiments are carried out on a high dimensional Raman spectral dataset, in which the task is to identify Acetaminophen, a pain-relieving drug, within a mixture. These experiments employ the NIPALS (Non-Linear Iterative Partial Least Squares) PCA method, a method that has been used in the field of chemometrics for spectral classification, and is a more efficient alternative than the widely used eigenvector decomposition approach. The experiments show that the use of this PCA method can improve the performance of machine learning in the classification of high dimensionsal spectral data.

1 Introduction

Automatic identification of illicit materials using Raman spectroscopy is of significant importance for law enforcement agencies. In recent years, Raman spectroscopy has enjoyed a strong resurgence in popularity due to advances in the technology that have resulted in more sensitive and lower cost instruments. The high dimensionality of spectral data can pose problems for machine learning as predictive models based on such data run the risk of overfitting. Furthermore, many of the attributes may be redundant or highly correlated, which can also lead to a degradation of prediction accuracy. Typically, methods from a field of study known as chemometrics have been applied to this particular problem [1], and these methods use PCA to handle the high dimensional spectra. PCA is a classical statistical method for transforming attributes of a dataset into a new set of uncorrelated attributes called principal components (PCs). PCA can be used to reduce the dimensionality of a dataset, while still retaining as much of the *variability* of the dataset as possible. The goal of this research is to determine if PCA can be used to improve the performance of machine learning methods in the identification of a material based on spectral data.

In the first set of experiments presented in this paper, the performance of five competitive and well-known machine learning techniques (Support Vector Machines, k-Nearest Neighbours, C4.5 Decision Tree, RIPPER and Naive Bayes) along with classification by Linear Regression are compared by testing them on a Raman spectral dataset. A number of pre-processing techniques such as normalisation and first derivative are applied to the data to determine if they can improve the classification accuracy of these methods. A second set of experiments is carried out in which PCA and machine learning (and the various pre-processing methods) are used in combination. This set of PCA experiments also facilitates a comparison of machine learning with the popular chemometric technique of Principal Component Regression (PCR), which combines PCA and Linear Regression.

The paper is organised as follows. Section 2 will give a brief description of Raman spectroscopy and outline the characteristics of the data it produces. Section 3 describes PCA and the PCR method that incorporates PCA into it. Section 4 provides a brief description of each machine learning technique used in this investigation. Experimental results along with a discussion are presented in Section 5. Section 6 describes related research and Section 7 presents the conclusion of this study.

2 Raman Spectroscopy

Raman spectroscopy is the measurement of the wavelength and intensity of light that has been scattered inelastically by a sample, known as the Raman effect [2]. This Raman scattering provides information on the vibrational motions of molecules in the sample compound, which in turn provides a chemical fingerprint. Every compound has its own unique Raman spectrum that can be used for sample identification. Each point of a spectrum represents the intensity recorded at a particular wavelength. A Raman dataset therefore has one attribute for each point on its constituent spectra. Raman spectra can be used for the identification of materials such as narcotics [1] and explosives [3].

Raman spectra are a good example of high dimensional data; a Raman spectrum is typically made up of 500-3000 data points, and many datasets may only contain 20-200 samples. However, there are other characteristics of Raman spectra that can be problematic for machine learning:

- *Collinearity:* many of the attributes (spectral data points) are highly correlated to each other which can lead to a degradation of the prediction accuracy.
- *Noise:* particularly prevalent in spectra of complex mixtures. Predictive models that are fitted to noise in a dataset will not perform well on other test datasets.
- *Fluorescence:* the presence of fluorescent materials in a sample can obscure the Raman signal and therefore make classification more difficult [1].
- *Variance of Intensity:* a wide variance in spectral intensity occurs between different sample measurements [4].

3 Principal Component Analysis

In the following description, the dataset is represented by the matrix X, where X is a $N \times p$ matrix. For spectral applications, each row of X, the p-vector x_i contains the

intensities at each wavelength of the spectrum sample i. Each column, X_j contains all the observations of one attribute. PCA is used to overcome the previously mentioned problems of high-dimensionality and collinearity by reducing the number of predictor attributes. PCA transforms the set of inputs $X_1, X_2, \ldots, X_N$ into another set of column vectors $T_1, T_2, \ldots, T_N$ where the T's have property that most of the original data's information content (or most of its variance) is stored in the first few T's (the principal component scores). The idea is that this allows reduction of the data to a smaller number of dimensions, with low information loss, simply by discarding some of the principal components (PCs). Each PC is a linear combination of the original inputs and each PC is orthogonal, which therefore eliminates the problem of collinearity. This linear transformation of the matrix X is specified by a $p \times p$ matrix P so that the transformed variables T are given by:

$$T = XP \quad \text{or alternatively } X \text{ is decomposed as follows: } X = TP^T \tag{1}$$

where P is known as the *loadings matrix*. The columns loadings matrix P can be calculated as the eigenvectors of the matrix $X^T X$ [5], a calculation which can be computationally intensive when dealing with datasets of 500-3000 attributes. A much quicker alternative is the NIPALS method [6]. The NIPALS method does not calculate all the PCs at once as is done in the eigenvector approach. Instead, it iteratively calculates the first PC, then the second and continues until the required number of PCs have been generated. See Ryder [1] and O'Connell *et al.* [4] for examples of the use of PCA in the classification of materials from Raman spectra.

3.1 Principal Component Regression

The widely used chemometric technique of PCR is a two-step multivariate regression method, in which PCA of the data is carried out in the first step. In the second step, a multiple linear regression between the PC scores obtained in the PCA step and the predictor variable is carried out. In this regression step, the predictor variable is a value that is chosen to represent the presence or absence of the target in a sample, e.g. 1 for present and -1 for absent. In this way, a classification model can be built using any regression method.

4 Machine Learning

4.1 Support Vector Machine

The SVM [7] is a powerful machine learning tool that is capable of representing non-linear relationships and producing models that generalise well to unseen data. For binary classification, a linear SVM (the simplest form of SVM) finds an optimal linear separator between the two classes of data. This optimal separator is the one that results in the widest margin of separation between the two classes, as a wide margin implies that the classifier is better able to classify unseen spectra. To regulate overfitting, SVMs have a complexity parameter, C, which determines the trade-off between choosing a large-margin classifier and the amount by which misclassified samples are tolerated. A

higher value of C means that more importance is attached to minimising the amount of misclassification than to finding a wide margin model. To handle non-linear data, kernels (e.g. Radial Basis Function (RBF), Polynomial or Sigmoid) are introduced to map the original data to a new feature space in which a linear separator can be found. In addition to the C parameter, each kernel may have a number of parameters associated with it. For the experiments reported here, two kernels were used: the RBF kernel, in which the kernel width, σ, can be changed, and the Linear kernel, which has no extra parameter. In general, the SVM is considered useful for handling high dimensional data.

4.2 k-Nearest Neighbours

k-Nearest Neighbours (k-NN) [8] is a learning algorithm which classifies a test sample by firstly obtaining the class of the k samples that are the closest to the test sample. The majority class of these nearest samples (or nearest single sample when k = 1) is returned as the prediction for that test sample. Various measures may be used to determine the distance between a pair of samples. In these experiments, the Euclidean distance measure was used. In practical terms, each Raman spectrum is compared to every other spectrum in the dataset. At each spectral data point, the difference in intensity between the two spectra is measured (distance). The sum of the squared distances for all the data points (full spectrum) gives a numerical measure of how close the spectra are.

4.3 C4.5

The C4.5 decision tree [9] algorithm generates a series of if-then rules that are represented as a tree structure. Each node in the tree corresponds to a test of the intensity at a particular data point of the spectrum. The result of a test at one node determines which node in the tree is checked next until finally, a leaf node is reached. Each leaf specifies the class to be returned if that leaf is reached.

4.4 RIPPER

RIPPER [10] (Repeated Incremental Pruning to Produce Error Reduction) is an inductive rule-based learner that builds a set of prepositional rules that identify classes while minimising the amount of error. The number of training examples misclassified by the rules defines the error. RIPPER was developed with the goal of handling large noisy datasets efficiently whilst also achieving good generalisation performance.

5 Experimental Results

5.1 Dataset

In the following experiments, the task is to identify acetaminophen, a pain-relieving drug that is found in many over-the-counter medications. The acetaminophen dataset comprises the Raman spectra of 217 different samples. Acetaminophen is present in 87 of the samples, the rest of the samples being made up of various pure inorganic materials. Each sample spectrum covers the range 350-2000 cm^{-1} and is made up of 1646 data points. For more details on this dataset, see O'Connell *et al.* [4].

5.2 Comparison of Machine Learning Methods

Table 1 shows the results of six different machine learning classification methods using a 10-fold cross-validation test on the acetaminophen dataset. The first column shows the average classification error achieved on the raw dataset (RD). The three remaining columns show the results of using each machine learning method in tandem with a pre-processing technique:

- ND: dataset with each sample normalised. Each sample is divided across by the maximum intensity that occurs within that sample.
- FD: a Savitzky-Golay first derivative [11], seven-point averaging algorithm is applied to the raw dataset.
- FND: a normalisation step is carried out after applying a first derivative to each sample of the raw dataset.

Table 1. Percentage Error in Identifying Presence of Acetaminophen, using various ML Methods in combination with various Pre-processing Techniques

	Pre-processing Technique			
Method	**RD**	**ND**	**FD**	**FND**
Linear SVM	**6.45** *(C=100)*	**2.76** *(C=1)*	**3.23** *(C=10000)*	**0.92*** *(C=0.1)*
RBF SVM	**5.07** *(C=1000, σ=0.1)*	**2.76** *(C=1000, σ=0.001)*	**1.84** *(C=10000, σ=10)*	**0.92*** *(C=10, σ=0.01)*
k-NN	11.06 *(k=1)*	7.83 *(k=1)*	**4.61** *(k=10)*	**4.15** *(k=1)*
C4.5	10.14	7.83	**1.84**	**1.38**
RIPPER	15.67	11.06	**3.69**	**2.3**
Naive Bayes	25.35	13.82	25.81	**5.53**
Linear Reg.	27.65	16.13	25.35	20.28

Table 1 shows the lowest average error average achieved by each classifier and pre-processing combination. For all these methods, apart from k-NN, the WEKA [8] implementation was used. The default settings were used for C4.5, RIPPER and Naive Bayes. For SVMs, RBF and Polynomial kernels with different parameter settings were tested. The parameter settings that achieved the best results are shown in parentheses. The Linear SVM was tested for the following values of C: $0.1, 1, \ldots, 10000$. The same range of C values were used for RBF SVM, and these were tested in combination with the σ values of: $0.0001, 0.001, \ldots, 10$. For k-NN, the table shows the value for k (number of neighbours) that resulted in the lowest percentage error. The k-NN method was tested for all values of k from 1 to 20. The results of each machine learning and pre-processing

technique combination of Table 1 were compared using a paired t-test based on a 5% confidence level and using a corrected variance estimate [12]. The lowest average error over all results in Table 1 of 0.92% (i.e. only two misclassifications, achieved by both Linear and RBF SVM) is highlighted in bold and indicated by an asterisk. Those results which do not differ statistically from the best results (according to the t-test) are also highlighted in bold.

On both the raw (RD) and normalised (ND) dataset, both SVM models perform better than any of the other machine learning methods, as there is no significant difference between the best overall result and the SVM results on RD and ND, whereas a significant difference does exist between the best overall result and all other machine learning methods on RD and ND. This confirms the notion that SVMs are particularly suited to dealing with high dimensional spectral data and it also suggests that SVMs are capable of handling a high degree of collinearity in the data. Linear Regression, on the other hand, performs poorly with all pre-processing techniques. This poor performance can be attributed to its requirement that all the columns of the data matrix are *linearly independent* [5], a condition that is violated in highly correlated spectral data. Similarly, Naive Bayes has recorded a high average error on the RD, ND and FD data. This is presumably because of its assumption of independence of each of the attributes. It is clear from this table that the pre-processing techniques of FD and FND improve the performance of the majority of the classifiers. For SVMs, the error is numerically smaller, but not a significant improvement over the RD and ND results. Note that Linear Regression is the only method that did not achieve a result to compete with the best overall result.

Overall, the SVM appears to exhibit the best results, matching or outperforming all other methods on the raw and pre-processed data. With effective pre-processing, however, the performance of other machine learning methods can be improved so that they are close to that of the SVM.

5.3 Comparison of Machine Learning methods with PCA

As outlined in Section 3, PCA is used to alleviate problems such as high dimensionality and collinearity that are associated with spectral data. For the next set of experiments, the goal was to determine whether machine learning methods could benefit from an initial transformation of the dataset into a smaller set of PCs, as is used in PCR. The same series of cross-validation tests were run, except in this case, during each fold the PC scores of the training data were fed as inputs to the machine learning method. The procedure for the 10-fold cross-validation is as follows:

1. Carry out PCA on the training data to generate a loadings matrix.
2. Transform training data into a set of PC scores using the first P components of the loadings matrix.
3. Build a classification model based on the training PC scores data.
4. Transform the held out test fold data to PC scores using the loadings matrix generated from the training data.
5. Test classification model on the transformed test fold.
6. Repeat steps 1-5 for each iteration of the 10-fold cross-validation.

With each machine learning and pre-processing method combination, the above 10-fold cross-validation test was carried out for P=1 to 20 principal components. Therefore, 20 different 10-fold cross-validation tests were run for Naive Bayes, for example. For those classifiers that require additional parameters to be set, more tests had to be run to test the different combinations of parameters, e.g. C, σ, and P for RBF SVM. The same ranges for C, σ and k were tested as those used for the experiments of Table 1.

Table 2. Percentage Error in Identifying Presence of Acetaminophen, using various ML Methods (and Pre-processing Techniques) with PCA

	Pre-processing Technique			
Method	**RD**	**ND**	**FD**	**FND**
Linear SVM	5.07 *(P=18,C=0.1)*	**1.84** *(P=13,C=0.1)*	**3.23** *(P=14,C=0.01)*	**0.46** *(P=4,C=0.1)*
RBF SVM	6.91 *(P=19,C=100, σ=0.001)*	**2.76** *(P=16,C=10, σ=0.001)*	**2.23** *(P=12, C=10, σ=0.001)*	**0.46** *(P=5,C=10, σ=0.001)*
k-NN	11.06 *(P=17,k=3)*	5.99 *(P=10,k=1)*	**2.3** *(P=14,k=1)*	**0.0*** *(P=4,k=5)*
C4.5	7.83 *(P=20)*	7.37 *(P=19)*	7.37 *(P=5)*	**1.38** *(P=6)*
RIPPER	11.98 *(P=20)*	8.29 *(P=8)*	6.45 *(P=5)*	**2.3** *(P=3)*
Naive Bayes	38.71 *(P=1)*	10.6 *(P=8)*	11.52 *(P=5)*	3.23 *(P=2)*
PCR (PCA+Linear Reg.)	9.22 *(P=16)*	5.53 *(P=20)*	8.29 *(P=11)*	**1.38** *(P=80)*

Table 2 shows the lowest average error achieved by each combination of machine learning and pre-processing method with PCA. The number of PCs used to achieve this lowest average error is shown in parentheses, along with the additional parameter settings for the SVM and k-NN classifiers. As with Table 1, the best result over all the results of Table 2 is highlighted in bold and denoted by an asterisk, with those results that bear no significant difference from the best overall result also highlighted in bold. Again, the pre-processing method of FND improves the performance of the majority of the classifiers, Naive Bayes being the exception in this case. In comparing the best result of Table 1 with the best result of Table 2 for each machine learning method (all in the FND column), it can be seen that the addition of the PCA step results in either the same error (C4.5 and RIPPER) or a numerically smaller error (Linear SVM, RBF SVM, k-NN and Linear Regression). The improvement effected by the inclusion of this

PCA step is particularly evident with the Linear Regression technique. Note that this combination of PCA and Linear Regression is equivalent to PCR.

Despite the fact that for the SVM and k-NN classifiers, there is no significant difference between the best results with or without PCA, it is noteworthy that the SVM and k-NN classifiers with PCA were capable of achieving such low errors with far fewer attributes, only four PCs for the Linear SVM and k-NN and 5 PCs for the RBF SVM. This makes the resulting classification model much more efficient when classifying new data. In contrast, PCR required a much greater number of PCs (80) to achieve its lowest error. (This result was discovered in the experiment detailed in the next section.)

To make an overall assessment of the effect of using PCA in combination with machine learning, a statistical comparison (paired t-test with 5% confidence level) of the 28 results of Table 1 and Table 2 was carried out. This indicates that, overall, a significant improvement in the performance of machine learning methods is gained with this initial PCA step. It can therefore be concluded that the incorporation of PCA into machine learning is useful for the classification of high dimensional data.

5.4 Effect of PCA on Classification Accuracy

To further determine the effect of PCA on the performance of machine learning methods, each machine learning method (using the best parameter setting and pre-processing technique) was tested using larger numbers of PCs. Each method was tested for values of P in the range 1-640. Figure 1 shows the change in error of some of the methods versus the number of PCs retained to build the model.

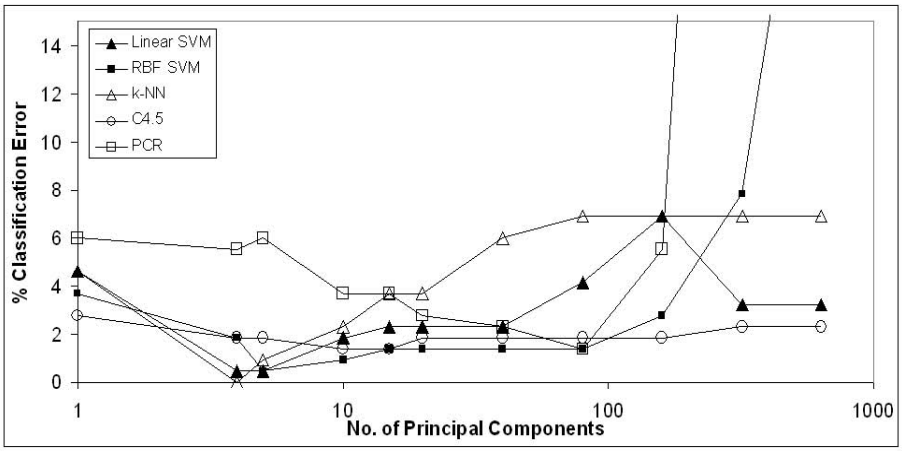


Fig. 1. Effect of changing the number of PCs on Machine Learning Classification Error

It can be seen from this graph that as PCs are added, error is initially reduced for all methods. Most methods require no more than six PCs to achieve the lowest error. After this lowest error point, the behaviour of the methods differ somewhat. Some classifiers

suffer drastic increases in error within the range of PCs tested: PCR, RBF SVM, and k-NN (although not to the same extent as the previous examples). In contrast, the error for C4.5 never deviates too much from its lowest error at six PCs. This may be due to its ability to prune irrelevant attributes from the decision tree model. The Linear SVM initially seems to follow the pattern of the majority of classifiers, but then returns to a more acceptable error with the higher number of PCs. Overall, it is evident that all of the classifiers, apart from PCR, will achieve their best accuracy with a relatively small number of PCs; it is probably unnecesary to generate any more than twenty PCs. However, the number of PCs required will depend on the underlying dataset. Further experiments on more spectral data, or other examples of high dimensional data, are required to determine suitable ranges of PCs for these machine learning methods.

6 Related Research

The most closely related research to the work presented here can be found in Sigurdsson *et al.* [13], where they report on the use of neural networks for the detection of skin cancer based on Raman data that has been reduced using PCA. They perform PCA using singular value decomposition (SVD), a method which calculates *all* the eigenvectors of the data matrix, unlike the NIPALS method that was used here. In addition, they do not present any comparison with neural networks on the raw data without the PCA step.

As far as the authors are aware, few studies have been carried out that investigate the effect of using PCA with a number of machine learning algorithms. Popelinsky [14] does analyse the effect of PCA (again, eigenvector decomposition is used) on three different machine learning algorithms (Naive Bayes, C5.0 and an instance-based learner). In this paper, the principal component scores are added to the original attribute data and he has found this to result in a decrease in error rate for all methods on a significant number of the datasets. However, the experiments were not based on particularly high dimensional datasets. It is also worth noting that there does not appear to be any evidence of the use of NIPALS PCA in conjunction with machine learning for the classification of high dimensional data.

7 Conclusions

This paper has proposed the use of an efficient PCA method, NIPALS, to improve the performance of some well-known machine learning methods in the identification of materials based on high dimensional spectral data. Experiments in the classification of Raman spectra have shown that, overall, this PCA method improves the performance of machine learning when dealing with high dimensional data. Furthermore, through the use of PCA, these low errors were achieved despite a major reduction of the data; from the original 1646 attributes to at least six attributes. Additional experiments have shown that it is not necessary to generate more than twenty PCs to find an optimal set for the spectral dataset used, as the performance of the majority of classifiers degrades with increasing numbers of PCs. This fact makes NIPALS PCA particularly suited to the proposed approach, as it does not require the generation of all PCs of a data matrix, unlike the widely used eigenvector decomposition methods. This paper has also shown

that the pre-processing technique of first derivative followed by normalisation improves the performance of the majority of these machine learning methods in the classification of the dataset used.

Overall, the use of NIPALS PCA in combination with machine learning and the first derivative with normalisation pre-processing technique appears to be a promising approach for the classification of high dimensional spectral data. Future work will involve using this approach for the identification of other materials based on Raman spectra, with tests also being carried out on other high dimensional datasets. This work will also investigate the automatic selection of parameters for these techniques, such as the number of PCs, kernel parameters for SVM and k for k-NN.

Acknowledgements

This research has been funded by Enterprise Ireland's Basic Research Grant Programme. The authors are also grateful to the High Performance Computing Group at NUI Galway, funded under PRTLI I and III, for providing access to HPC facilities.

References

1. Ryder, A.: Classification of narcotics in solid mixtures using Principal Component Analysis and Raman spectroscopy and chemometric methods. J. Forensic Sci **47** (2002) 275–284
2. Bulkin, B.: The Raman effect: an introduction. New York: John Wiley and Sons, Inc (1991)
3. Cheng, C., Kirkbride, T., Batchelder, D., Lacey, R., Sheldon, T.: In situ detection and identification of trace explosives by Raman microscopy. J. Forensic Sci **40** (1995) 31–37
4. O'Connell, M., Howley, T., Ryder, A., Leger, M., Madden, M.: Classification of a target analyte in solid mixtures using principal component analysis, support vector machines and Raman spectroscopy. In: Proc. SPIE - Int. Soc. Opt. Eng. Volume 5826 (in press). (2005)
5. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer (2001)
6. Geladi, P., Kowalski, B.: Partial Least Squares: A Tutorial. Analytica Chemica Acta **185** (1986) 1–17
7. Scholkopf, B., Smola, A.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press (2002)
8. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann Publishers (2000)
9. Quinlan, R.: Learning Logical Definitions from Relations. Machine Learning **5** (1990)
10. Cohen, W.: Fast Eeffective Rule Induction. In: Proc. of the 12th Int. Conference on Machine Learning. (2002) 115–123
11. Savitzky, A., Golay, M.: Smoothing and differentiation of data by simplified least squares procedures. Anal. Chem. **36** (1964) 1627–1639
12. Nadeau, C., Bengio, Y.: Inference for generalisation error. In: Advances in Neural Information Processing 12. MIT Press (2000)
13. Sigurdsson, S., Philipsen, P., Hansen, L., Larsen, J., Gniadecka, M., Wulf, H.: Detection of Skin Cancer by Classification of Raman Spectra. IEEE Transactions on Biomedical Engineering **51** (2004)
14. Popelinsky, L.: Combining the Principal Components Method with Different Learning Algorithms. In: Proc. of ECML/PKDD IDDM Workshop (Integrating Aspects of Data Mining, Decision Support and Meta-Learning. (2001)

Separating Heuristic Search and Statistical Inference in Decision Tree Learning: The ID3* Algorithm

Ray J. Hickey

School of Information and Software Engineering, University of Ulster at Coleraine, Co. Londonderry, N.Ireland, BT52 1SA.
E-mail: rj.hickey@ulster.ac.uk

Abstract. Using an algorithm such as ID3 to learn a decision tree for classification involves both heuristic search and statistical inference from training data. In assessing the effectiveness of such learning, it will be argued here that separating the contribution of these two activities is useful. Central to this view is the notion of a true decision tree classifier (called ID3* if ID3 is used to learn). Amongst the benefits of this perspective on learning are sharper insights into such issues as whether ID3 does tend to build small trees and whether overfitting avoidance can be properly regarded as a utility.

1. Introduction

Guided by training data, decision tree induction algorithms such as ID3 perform a greedy heuristic search in a space of decision trees in order to build a classification model. Heuristic search is fundamental to much of Artificial Intelligence but in Machine Learning there is the added complication that heuristic scores must be *estimated* from training data. In the case of ID3, class probability distributions are estimated from the corresponding class relative frequency distributions and an entropy-based information heuristic is applied to these. ID3, in common with several other learning algorithms, interfaces to the training set solely through class frequency distributions.

Thus, to be effective, Machine Learning must employ techniques of statistical inference to ensure that the inductions made from data, and which guide the search, are valid. It is this combination of the Artificial Intelligence search process and statistical inference that makes Machine Learning especially difficult.

It will be argued here that in discussions of the major issues in decision tree learning, such as *attribute selection*, *inductive bias* and *pruning*, the search and the statistical inference aspects are often confounded and that this can lead to a confused view of the learning process.

The present work, therefore, endeavours to show that, for decision tree learning, the search and statistical aspects can be considered separately. A view is adopted in which the learning process is seen as one of *estimation*, using training examples, of a ‘true’ decision tree, one constructed from the underlying probability distribution

of the examples. Central to this perspective is the clear distinction between a model (of a classifier) and its many representations.

2 The ID3 Induction Process

Given a set of classified training examples expressed in an attribute-value description language the ID3 algorithm (Quinlan [14]) induces a decision tree through a process of expansion downwards from the root node. Each expansion involves splitting on a single attribute, either using the individual values of the attribute (discrete case) or through a binary test *attribute* =< *value* (continuous case).

Each node corresponds to a concept defined as the conjunction of tests on the path from the root to the node. The examples subsumed by this concept (the *covered* examples) exhibit a relative frequency distribution of assigned classes, *the class (relative) frequency profile*.

An attribute is selected for expansion at a node, from those available, using a heuristic that assesses the profiles. The most widely adopted measure is the (expected) information gain. Information is explicated using the entropy function defined, for probability or relative frequency distribution $Q = (q_1, ...q_n)$, as:

$$entropy(Q) = -q_i \sum_{i=1}^{i=n} \log_2 q_i$$

Information gain is then:

$$infGain(A, N) = entropy(P) - \sum_{i=1}^{i=m} f_i entropy(P_i)$$

where node *N*, with class relative frequency profile, *P*, is expanded using discrete attribute, *A*, which has *m* values with relative frequencies at *N* of $(f_1, ... f_m)$ and the class relative frequency distributions of the expanded (or *child*) nodes of *N* are $(P_1, ...P_m)$. A corresponding definition exists if *A* is continuous.

A class frequency profile in which all frequency is concentrated at a single class is said to be *pure* and will minimise entropy. Information gain is zero if and only if all child nodes have identical frequency profiles.

The process of tree expansion terminates when either:

1. there are no more attributes available for expansion or
2. all unexpanded nodes have pure frequency profiles or
3. no immediate information gain is possible (a hill-climbing property).

The majority class in a leaf class frequency profile, when associated with the path from the root to that leaf provides a (*deterministic*) *classification rule*. The set of such rules is the induced classifier used to classify new examples.

Several authors (for example, Quinlan [14], Frank and Witten [3], Jensen and Cohen [7]) suggest that a formal hypothesis test of independence be used on an attribute to decide whether a node expansion is worth making (run-time pruning).

Information gain is closely related to the standard chi-squared statistic for independence (White and Liu [17]) for which statistically significant large values indicate that the class frequency profiles of the child nodes in the expansion are genuinely different. Martin [9] proposes attribute selection and run-time pruning based on the exact probability test. Alternatively post-pruning of the fully expanded tree can be performed (Quinlan [15], Breiman [2]). Post pruning will be discussed further below.

Jensen and Cohen [7] argue, further, that mis-handling of multiple comparison procedures (MCP) is often the source of problems with induction. An MCP involves selection of an item amongst several candidates where each is scored using an evaluation function, e.g. in attribute selection. Picking the item with the largest score without adjusting for the number of items may, as the number of items increases, result in selection of an item which is not the best or significantly better than the others.

3. The Class Model and its True Tree Representations

As noted above, ID3 interfaces to the example set through the node class frequency distributions. The latter are statistical estimates of the true probability distributions. If the true probabilistic model of the domain were completely known it would not be necessary to use training examples to build a tree: the true class distribution at a node could be obtained from the model. The issue of tree representation and the need for search to find an appropriate tree would, however, still remain.

In Hickey [6] the true relationship between the class and the description attributes (all assumed discrete) is called the *class model*. The latter is derivable from the joint probability distribution of the description attributes together with the class attribute and provides the probability distribution of class conditional on any description attribute vector – either fully or partially instantiated. The corresponding mapping between description vectors and the majority class in each class distribution is called the *best* (*deterministic*) *classifier*. The notion of class model can be extended to cater for continuous attributes.

Any statement specifying the class model is a *representation* of the model (Hickey [6]). The class attribute is just a dependent variable with the description attributes being the corresponding independent variables. What distinguishes class models in Machine Learning from other scientific or statistical models is the large number of possible representations. Compare the situation in statistical regression analysis: for a linear model - even one with a large number of independent variables - algebraic re-expression is not a substantial issue. Normally a mathematical or statistical model may be re-expressed in a small number of different, but logically equivalent, formulations to suit particular purposes (such as to gain insight or to facilitate further analysis) but there is not the notion of many different representations with different utilities.

In contrast, in Machine Learning and in Artificial Intelligence generally, because models tend to be descriptive, finding an appropriate representation is a substantial issue and, as a result, the distinction between a model and its representation(s) becomes vital. For the class models discussed here, possibilities

range from, at one extreme, tabulation of all fully-instantiated description vectors with their conditional class distributions to, at the other extreme, a comparatively small set of rules in which individual rules have considerable generality. Finding a representation to satisfy some requirement, e.g. that with the smallest number of rules, usually requires a substantial search.

One possibility is to use a decision tree to represent the class model. Branching tests would be as they are for induction; leaves would contain conditional class probability distributions. Rules of the form *if* <*path*> *then* <*class distribution*> could be extracted from the tree. If a domain has only discrete attributes then all possible class models can be represented in this way. If a continuous attribute is involved, however, there may be no partition of intervals of the value set of the attribute such that the class probability distributions are constant on an interval. In some cases the distribution in a leaf would be a non-constant function of the actual value of the continuous attribute. In the discussion that follows it will be assumed that such a partition exists and involves a finite number of sets.

Definition A decision tree which represents a class model will be called a *true*[1] (*decision*) *tree* representation for the model. The mapping from paths to majority class of leaf class distributions is the *best classifier* corresponding to the true tree.

A true tree representation may, in the first instance, be fully expanded, i.e. all paths are of length *n* where there are *n* attributes in the domain (called here a *complete* tree). There may, though, be nodes at the bottom of the tree where each sibling of an expansion shares the same class probability distribution. Beginning at the leaves, such nodes may be unexpanded recursively until siblings with different distributions are first encountered. This is analogous to post-pruning of an induced tree but is not a statistical procedure because all distributions are known. The remaining tree after such purging will be called a *core* tree. Expansions thus removed may (but not necessarily) have involved pure noise attributes; see Hickey [6] for a discussion of pure noise. The core may also contain pure noise attributes and is said to be *inflated* by them. A core that does not contain pure noise attributes is *deflated*. Identifying pure noise attributes internal to the core is quite complicated.

Removing unnecessary expansions from the bottom of the tree simplifies the representation but does not simplify the model: the pruned and unpruned versions are merely different correct representations of the same model. Likewise deflating a core simplifies the representation but not the model.

It is important to consider how a decision tree might wrongly represent a model. There are two elements to a decision tree representation: the tree structure and the class distributions. A fully expanded tree, is always a correct structure; only a wrong specification of a class distribution in a leaf can render the tree an inaccurate representation of the model.

On the other hand, if informative attributes are missing from the tree then it cannot represent the class model correctly even if the class distributions are the appropriate ones given the conditions specified by the paths. Such a tree is said to *under-represent* the model and would have a lower classification rate.

[1] Called a *canonical tree* in Hickey [5]

4 The ID3* Tree

One method of constructing a true tree is to use a probabilistic version of ID3: the information gain heuristic is applied to class distributions at nodes rather than to class frequency profiles of training examples. A true tree will be produced in which the (locally) most informative attributes will appear near the top and the least informative towards the bottom. If, amongst the attributes available for expansion at a node, there are some offering positive information gain, select the one with the greatest gain. There is no MCP issue.

A node that offers no immediate information gain from any attribute is called a *locally uninformative node* (LUN). LUNs sub-divide into

1. those where the Cartesian product of several of the available attributes (i.e. a conjunction) would produce a gain (called *non-terminal* LUNs) and
2. those where all available attributes are pure noise, and so there is no further information gain to be had, (called *terminal* LUNs).

At a non-terminal LUN, the tree must be expanded else it would under-represent the class model. At a terminal LUN, no expansion is required: the class distributions are correct and any further expansion merely sub-divides each distribution into specialised replicates. A terminal LUN is thus a leaf node of the core of the true tree. Unfortunately it is not possible to decide immediately at a LUN node whether it is terminal or non-terminal and so in all cases further search must be undertaken.

Because there is no immediate gain, a new criterion must be adopted for attribute selection. One method is to choose an attribute that maximises the average of the information gains across expansions of its child nodes. Of course each child node might also be a LUN, thus the criterion would have to be applied recursively until some genuine gain were obtained. If the LUN is non-terminal, this gain must occur at some stage. Such a gain will be referred to as the *minimal-lookahead information gain* (minimal being in the sense of distance ahead). At a non-LUN node, the minimal-lookahead gain is just the normal information gain. If several attributes tie for maximum minimal-lookahead gain a random choice should be made. If the minimal-lookahead gain at a LUN is zero, then it is terminal and so no expansion is required.

This probabilistic version of ID3 using maximising minimal-lookahead gain as the criterion for attribute selection, will be called ID3*.

Algorithm (ID3*)

Input : the joint distribution of description and class attributes.
Output : a decision tree, *T*, with class distributions in the leaves (ID3*).

Initialise *T* to be a root node having the unconditional class distribution
while there are nodes in *T* that are not marked as terminal do
{

- choose such a node, *N*
- if the class distribution of *N* is degenerate, mark *N* as terminal

- if there are no attributes available to expand *N*, mark *N* as terminal
- if there is a least distance, *d*, such that an attribute offers positive minimal-lookahead gain at this distance
  - then select the attribute which maximises this gain (if unique) or choose one at random from those maximising the gain (if there are several) and expand *N* creating a child node for each value of the selected attribute
- else mark *N* as terminal

}
Return *T*

Note that the ID3* true tree is automatically a core tree: pure noise attributes at the bottom will not be expanded. An ID3* tree is also deflated:

Theorem The attribute selection criterion for the ID3* will never choose a pure noise attribute for expansion and so an ID3* tree is always deflated.

Proof At a non-LUN node the attribute selected will offer a positive gain. Therefore not all class distributions at the child nodes of this attribute are identical and so the attribute is not pure noise.

At a LUN node, *N*, suppose attribute, *A*, offers maximum minimal-lookahead gain and that this is positive (if it is zero then, by definition, no expansion will be made).

Suppose *A* is a pure noise attribute. Let *d* be the least depth below *N* at which there is a node, *N''*, two of whose children have different class distributions. Let *N'* be the child node of *N* lying on the path from *N* to *N''*. (*N'* and *N''* could be identical). Let the attribute expanded by ID3* at *N'* be *A'*. *A'* is then the maximum minimal lookahead attribute at *N'*. All nodes that are descendants of *N'* (and *N'* itself) have the same value, *v*, of *A* (the value on the branch from *N* to *N'*); so too will all nodes considered for expansion by ID3* at or beneath *N'*. Because *A* is pure noise, uninstantiating this value will not affect any of the conditional probability distributions at or beneath *N'*. Thus the ID3* sub-tree rooted at *N'* can be lifted and placed at *N* with *A'* now being the expanded attribute at *N'* and having the same minimal lookahead information gain as *A*. But the distance between *N* and *N''* is now *d*-1 instead of *d*. This contradicts the maximal minimal-lookahead property for *A* and so it cannot be a pure noise attribute. □

Since it may require full expansion of a sub-tree, the minimal lookahead search undertaken at a LUN is NP-hard. The formation of the true ID3* tree is, for the most part, just a thought experiment the purpose of which is to gain insight into the induction process.

The lookahead performed by ID3* should not be confused with that sometimes adopted in variants of ID3 (see, for example, Murthy and Salzberg [12], and Norton [13]) where it is used to widen the search space in an attempt (which appears not be altogether successful) to find a smaller tree.

4.1 The ID3 Tree as an Estimate of the ID3* Tree

In an ideal world, anyone wishing to create a representation of a class model could build the ID3* tree and extract a rule set from it. In the absence of a full specification of the class model a set of classified training examples is used as a substitute. The ID3 tree built from the training set can be regarded as an estimate of the ID3* tree that would have been built from the class model.

The main tree-building inference mechanism in ID3 is the estimation of information gain. This is straightforward, based, as it is, on class frequency distributions as estimates of the corresponding class probability distributions. There is a major shortcoming, though in the treatment of LUNs. First of all, it requires an inference to establish that a node really is a LUN. This can be undertaken using a proper MCP procedure although without lookahead, as undertaken by ID3*, it cannot be established whether the LUN is terminal or not.

Equally, the continued expansion without testing for a terminal node is unsatisfactory. It amounts to assuming that there are no pure noise attributes. In the case where the node is a non-terminal LUN, selection of the attribute with the highest immediate gain can be seen as an inferior estimation of the maximum minimal-lookahead attribute. Thus, overall, it is in the handling of LUNs that ID3 is weakest in comparison to ID3*.

The ID3 tree (grown without run-time pruning) will tend to:

1. Successfully replicate the ID3* tree in the upper parts because node frequencies tend to be large thus permitting good estimation of the locally most informative attributes
2. Lower down the tree, expand on pure noise attributes (inflation) wrongly inferring that they are informative.
3. Extend the tree beyond terminal LUNs because there is no mechanism for their detection.
4. Under-represent the model because informative attributes are pushed over the horizon, i.e. beyond the leaves of the tree, as a result of the inflation in 2) above. This is a horizon effect analogous to that in minimax - an unimportant but immediate consideration pushes an important situation beyond a point where it would be visible.
5. Wrongly identify the majority class in some leaves because the sample reaching the leaf is small and biased as result of the attribute selection process.

Thus, for a domain that contains some pure noise attributes and unless the training set is sufficiently large, the induced ID3 tree will tend to be an overgrown and inflated under-representation of the model.

5 Discussion

The development above provides a basis for discussion of some of the key issues in decision tree learning: tree size, bias and overfitting.

5.1 Tree Size and Inductive Bias in ID3

Inductive bias is regarded as providing an essential framework that permits learning to take place. Mitchell [11] states that inductive bias amounts to 'some form of prior assumptions' about the identity of the target concept and that without such assumptions there can be no basis for generalising beyond the examples. This clearly is the case for a *hypothesis restriction bias* as applies to, for example, the Candidate Elimination algorithm where no generalisation is possible without the bias. ID3, though, has a *preference bias*: Mitchell [11] says

"Shorter trees are preferred over longer ones. Trees that place high information gain attributes close to the root are preferred to those that do not."

The second statement here simply states what ID3 does. Ultimately the purpose of the bias in attribute selection is, as asserted in the first statement, to produce a small tree, i.e. one with has shorter paths to its leaves.

There is some empirical evidence that this is so (see, for example Mingers [10]). The information-theoretic analysis of Goodman and Smyth [4] provides bounds on the size of binary trees built using information gain.

The theorem above shows that ID3* builds a deflated core. The omission of pure noise expansions that would otherwise inflate the core certainly contributes to the tree being smaller. Does the information gain heuristic achieve more than this: that is, amongst deflated core trees, does ID3* tend to find one that is small?

ID3* selects for expansion an attribute having the least average entropy values over child node class distributions (assuming the node is not a LUN). The (hill-climbing) argument for this is that a low entropy class distribution is nearer to a pure distribution and thus the leaf of the core tree will tend to be closer. On the face of it, though, a node class distribution reveals nothing about the depth of the tree beneath. The class distribution could be fairly pure (low entropy) yet have considerable depth beneath it or be fairly impure (high entropy) and be terminal. In short, because the probability distributions in the class model are entirely arbitrary, it would appear difficult to assert that entropy at a node tends to be correlated with depth to the core beneath that node.

There is, however, an argument in support of the claim. It follows from a consideration of how an internal node's class distribution may be calculated. The latter can be obtained by recursive backing up of the leaf class distributions of the core subtree beneath the node - a parent node's class distribution is the weighted average (convex combination) of its child nodes' distributions. The backing up process results in information loss (the opposite to the information gain obtained from expanding downwards). Generally, the greater the depth that has to be backed up, the greater the loss.

Now suppose that all leaf nodes in the model exhibit similar noise levels as measured by entropy. Nodes internal to the core must then have class distributions with higher entropies and the further up the tree they are, the greater these entropies will tend to be. An attribute that has a leaf node as one or more of its children will, therefore, tend to have lower average entropy than one that has no leaf nodes. What this amounts to is that average entropy, applied to each candidate attribute, is operating as a *depth-sounding device*.

This depth-sounding argument assumed that entropy was broadly constant across all leaves of the tree. Removal of this condition provides an example of a circumstance in which ID3* may miss the opportunity to expand on an attribute that would have yielded a shallow subtree. Suppose a leaf child node of a candidate attribute has a high entropy in comparison with other leaf nodes of the subtree. Other attributes may have child node distributions that, although heavily backed up, still manage to have lower entropy than that of this leaf. Thus our attribute may lose the competition.

Apart, then, from the circumstance just described, in can be argued that ID3* does indeed build a tree that tends to be small within the population of all deflated core trees. To the extent that it is a good estimate of ID3*, ID3 will also tend to have this property. With ID3, unlike ID3*, there is the possibility that pure noise attributes will be selected for expansion thus inflating the core and resulting in a larger tree. As noted by Liu and White [8] using information gain does guard against this to some extent with pure noise attributes tending to be pushed towards the bottom of the tree.

Rather than facilitating induction, bias can be regarded a personal choice of a representation of a model: someone who knows the class model (and so has no need of induction) must still decide on a representation. If the ID3* representation is chosen then ID3* must be run to build the representation. Learning from training examples using ID3 can thus be seen as:

1. deciding upon the ID3* tree to represent the model,
2. estimating this tree from training examples using class frequencies profiles in place of the true (but unknown) node class distributions.

Preference bias, therefore, relates primarily to representation and not to induction (although, in ID3, bias facilitates induction through preserving leaf frequencies).

5.2 Overfitting Avoidance as a Utility

A fully expanded decision tree is often regarded as an overfit of the model. Post pruning is then advocated to find a more correct fit. There is a sense in which a representation, unlike a model, *cannot* be overfitted. Extending the tree below the core results in an unnecessarily elaborate representation but one that can represent the same model provided the class frequency profiles are accurate.

In practice the frequency in the leaves of fully expanded trees tend to be small and so the class frequency profiles are poor estimates of the probabilities and, in particular, the true majority class is less likely to be discovered. Since the generalisation performance of the classifier depends heavily on correct identification of majority class, it is prudent to prune back, accumulating frequency until statistical reliable estimates are obtained. Such pruning may stray into the core and thus create, or add to, under-representation - which is counter-productive.

Schaffer [16] suggests that post pruning amounts to a bias, reflecting a preference (or utility) of the user for a simpler model. Viewed as described above, however, post pruning is a purely statistical procedure; like all such procedures it will satisfy some user-specified criterion relating to statistical significance or confidence. Unquestionably, the latter is difficult to ascertain because the MCP

tree-building activity biases the class frequency distributions in the leaves, but this does not constitute a utility. Pruning into the core would result in a simpler model being induced but this would be a by-product of the satisfaction of the statistical requirement not an explicit request on the part of the user for a simpler model.

Trading accuracy of the classifier for a simpler tree and rule set is a different problem and has been investigated by Bohanec and Bratko [1].

References

1. M. Bohanec and I. Bratko, Trading accuracy for simplicity in decision trees, *Mach. Learning* **15** (1994) 223-250.
2. L. Breiman, J. Friedman, R. Olshen and C. Stone, *Classification and Regression Trees* (Wadsworth, Belmont, CA, 1984)
3. E. Frank and I.H. Witten, Using a permutation test for attribute selection in decision trees, in J. Shavlik, ed., *Proceedings Fifteenth International Conference on Machine Learning* (Morgan Kaufmann, San Mateo, California, 1998) 152-160.
4. R.M. Goodman and P. Smyth, Decision tree design from a communication theory standpoint, *IEEE Trans. on Information Theory* **34** (1988) 979-994
5. R.J. Hickey, Artificial universes: towards a systematic approach to evaluating algorithms which learn from examples, in D. Sleeman and P. Edwards, eds., *Proceedings Ninth International Conference on Machine Learning* (Morgan Kaufmann, San Mateo, California, 1992) 196-205.
6. R.J. Hickey, Noise modelling and evaluating learning from examples, *Artif. Intell.* **82** (1996) 157-179.
7. D.D. Jensen and P.R. Cohen, Multiple comparisons in induction algorithms, *Mach. Learning* **38** (2000), 1-30.
8. W.Z. Liu and A.P. White, The importance of attribute selection measures in decision tree induction, *Mach. Learning* **15** (1994) 25-41.
9. J.K. Martin, An exact probability metric for decision tree splitting and stopping, *Mach. Learning* **28** (1997) 257-291.
10. J. Mingers, An empirical comparison of pruning methods for decision tree induction, *Mach. Learning* **4** (1989) 227-243.
11. T.M. Mitchell, *Machine Learning* (McGraw-Hill, New York, New York, 1997.
12. S.K. Murthy and S. Salzberg, Lookahead and pathology in decision tree induction, *Proceedings Fourteenth International Joint Conference on Artificial Intelligence* (Morgan Kaufmann, San Mateo, California, 1995) 1025-1031.
13. S.W. Norton, Generating better decision trees, *Proceedings Eleventh International Joint Conference on Artificial Intelligence* (Morgan Kaufmann, San Mateo, California, 1989) 800-805.
14. J.R. Quinlan, Induction of decision trees, *Mach. Learning* **1** (1986) 81-106.
15. J.R. Quinlan, Simplifying decision trees, *Int. J. of Man-Machine Studies* **27** (1987) 221-234.
16. C. Schaffer, Overfitting avoidance as bias, *Mach. Learning* **10** (1993) 153-178.
17. A.P. White and W.Z. Liu, Bias in information-based measures in decision tree induction, *Mach. Learning* **15** (1994) 321-329.

Poster Presentations

Approaches to Developing an Intelligent MultiMedia Distributed Platform Hub

Glenn G. Campbell, Tom Lunney, Paul Mc Kevitt

School of Computing and Intelligent Systems
Faculty of Engineering
University of Ulster, Magee
Derry/Londonderry, BT48 7JL, N. Ireland
{Campbell-g8, TF.Lunney, P.McKevitt}@ulster.ac.uk

Abstract. Preliminary research on the development of an intelligent multimedia distributed platform hub (MediaHub) for the fusion and synchronisation of language and vision data is presented. Related research is reviewed and a potential new approach to decision-making within MediaHub based on Bayesian Networks is proposed. A proposed system architecture, including a Dialogue Manager, Semantic Representation Database and Decision-Making Module, is outlined. Bayesian Networks will be employed in the decision-making process within the Decision-Making Module. Initial findings suggest that this is a promising approach for MediaHub.

1 Introduction

The area of intelligent multimedia has seen considerable research into creating user interfaces that can accept multimodal input. This has led to the development of intelligent interfaces that can learn to meet the needs of the user, in contrast to traditional systems where the onus was on the user to learn to use the interface. A more natural form of human-machine interaction has resulted from the development of systems that allow multimodal input such as natural language, eye and head tracking and 3D gestures [1] [2]. Considerable work has also been completed in the area of knowledge representation within multimodal systems, with the development of several semantic mark-up languages [3]. Efforts have also been made to integrate natural language and vision processing, and the main approaches in this field are described in [2].

The area of distributed computing has been exploited to create intelligent multimedia systems that are human-centred and directly address the needs of the user. DACS (Distributed Applications Communication System) [4] is a powerful tool for system integration that provides numerous features for the development and maintenance of distributed systems. Communication within DACS is based on simple asynchronous message passing, with additional extensions to deal with dynamic system reconfiguration during run-time. Other more advanced features include both synchronous and asynchronous remote procedure calls and demand streams.

1.1 Objectives of MediaHub

The principle aim of this research is to develop an intelligent multimedia distributed platform hub (MediaHub) for the fusion and synchronisation of multimodal information, specifically language and vision data. The primary objectives are to:

- Interpret/generate semantic representations of multimodal input/output.
- Perform fusion and synchronisation of multimodal data (decision-making).
- Implement and evaluate MediaHub, a multimodal platform hub with a potential new approach to decision-making.

It is envisaged that MediaHub will be tested as a plug-in within an existing multimodal platform such as CONFUCIUS [5] using multimodal input/output data.

Next, section 2 reviews research related to the development of MediaHub. Then, section 3 focuses on multimodal semantic representation. Section 4 discusses decision-making within MediaHub. Section 5 presents the proposed system architecture of MediaHub, while section 6 provides a conclusion and discussion of the future development of MediaHub.

2 Related Research

This section gives a review of related research that is relevant to the design and implementation of MediaHub. Section 2.1 provides a review of the area of distributed processing, whilst section 2.2 looks at existing multimodal distributed platforms.

2.1 Distributed Processing

Recent advances in the area of distributed systems have seen the development of several software tools for distributed processing. These tools are utilised in the creation of a range of distributed platforms. The Open Agent Architecture (OAA) [6] is a general-purpose infrastructure for creating systems that contain multiple software agents. OAA allows such agents to be developed in different programming languages and run on different platforms. All agents interact using the Interagent Communication Language (ICL). ICL is a logic-based declarative language used to express high-level, complex tasks and natural language expressions. JATLite [7] incorporates a set of Java packages that enable multi-agent systems to be constructed using Java. JATLite provides a Java agent platform that uses the KQML (Knowledge Query and Manipulation Language) Agent Communication Language (ACL) [8] for inter-agent communication. KQML is a message format and message-handling protocol used to support knowledge sharing among agents.

.NET [9] is the Microsoft Web services strategy that allows applications to share data across different operating systems and hardware platforms. The web services provide a universal data format that enables applications and computers to communicate with one another. Based on XML, the web services allow

communication across platforms and operating systems, irrespective of what programming language is used to write the applications. CORBA [10] is a specification released by the Object Management Group (OMG). A major component of CORBA is the Object Request Broker (ORB), which delivers requests to objects and returns results back to the client. The operation of the ORB is completely transparent to the client, i.e. the client doesn't need to know where the objects are, how they communicate, how they are implemented, stored or executed. CORBA uses the Interface Description Language (IDL), with syntax similar to C++, to describe object interfaces.

2.2 Multimodal Platforms

Numerous intelligent multimedia distributed platforms currently exist. With respect to these platforms, of particular interest to the design of MediaHub are their methods of semantic representation, storage and decision-making (fusion and synchronisation).

Ymir [11] is a computational model for creating autonomous creatures capable of human-like communication with real users. Ymir represents a distributed, modular approach that bridges between multimodal perception, decision and action in a coherent framework. The modules within Ymir are divided into four process collections. The Reactive Layer operates on relatively simple data. The Process Control Layer controls the global aspects of the dialogue and manages the communicative behaviour of the agent. The Content Layer hosts the processes that interpret the content of the multimodal input and generate suitable responses. The Action Scheduler within Ymir is used to coordinate appropriate actions. There are three main blackboards implemented in Ymir, and communication is achieved via message passing. The first blackboard, called the Functional Sketchboard, is primarily used for information exchange between the Reactive Layer and the Process Control Layer. The second blackboard is called the Content Blackboard. This deals with communication between the Process Control Layer and the Content Layer. The messages that are posted on the Content Blackboard are less time-critical than those on the Functional Sketchboard. The third blackboard is called the Motor Feedback Blackboard and is used to keep track of which part of a stream of actions is currently being planned or carried out by the Action Scheduler. Within the Ymir architecture, a prototype interactive agent called Gandalf has been created. Gandalf is capable of fluid turn-taking and dynamic sequencing.

CHAMELEON [12] is a platform for developing intelligent multimedia applications that makes use of DACS for process synchronisation and intercommunication. The hub of CHAMELEON consists of a dialogue manager and a blackboard. The role of the blackboard is to keep track of interactions over time, using frames for semantic representation. The architecture of CHAMELEON is shown in Fig. 1. CHAMELEON consists of ten modules, mostly programmed in C and C++, which are glued together by the DACS communications system. The blackboard and dialogue manager form the kernel of CHAMELEON. The blackboard stores the semantic representations produced by the other modules, keeping a history of all interactions. Communication between modules is achieved by exchanging semantic representations between themselves or the blackboard.

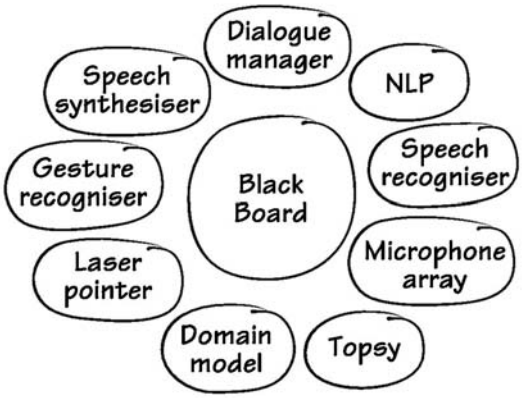


Fig. 1. Architecture of CHAMELEON [12]

SmartKom [13] is a multimodal dialogue system that is being developed to help overcome the problems of interaction between people and machines. SmartKom focuses on developing multimodal interfaces for applications in the home, public and mobile domains. The system uses a combination of speech, gestures and facial expressions to facilitate a more natural form of human-computer interaction, allowing face-to-face interaction with its conversational agent Smartakus. For example, in the public domain, the user can allocate the task of finding a library to Smartakus. MIAMM [14] is an abbreviation for Multidimensional Information Access using Multiple Modalities. The aim of the MIAMM project is to develop new concepts and techniques that will facilitate fast and natural access to multimedia databases using multimodal dialogues.

3 Multimodal Semantic Representation

One of the central questions in the development of intelligent multimedia or multimodal systems is what form of semantic representation should be used. The term 'semantic representation' refers to the method employed to represent the meaning of media representation [3]. This semantic representation must support interpretation and generation, multimodal input and output and a variety of semantic theories. The majority of the work in multimodal systems employs either frames or XML as the method of semantic representation. A review follows of both these approaches.

3.1 Frames

A frame is a collection of attributes with associated values that represent some real world entity. Minsky [15] first introduced frames as a method of semantically

representing situations in order to facilitate decision-making and reasoning. The idea of frames is based on human memory and the psychological view that, when faced with a new problem, humans select an existing frame (remembered framework) and adapt it to fit the new situation by changing appropriate details. Although frames have limited capabilities on their own, a frame system provides a powerful mechanism for encoding information to support reasoning and decision-making. Frames can be used to represent concepts, including real world objects, for example "the village of Dromore". The frames used to represent each concept have slots which represents the attributes of the concept. Frame-based methods of semantic representation are implemented in Ymir [11] and CHAMELEON [12]. Fig. 2 shows an example of the frame semantic representation that is utilised in CHAMELEON. The example frame in Fig. 2 illustrates how speech and gesture input are represented using input frames in the CHAMELEON platform. Note that although the syntax and structure of frames will vary from system to system, the basic idea of knowledge representation will remain the same.

```
[SPEECH-RECOGNISER
UTTERANCE:(Point to Hanne's office)
INTENTION: instruction!
TIME: timestamp]

[GESTURE
GESTURE: coordinates (3, 2)
INTENTION: pointing
TIME: timestamp]
```

Fig. 2. Example frame from CHAMELEON [12]

3.2 XML

Besides frames, the other most popular method of semantic representation in multimodal systems is XML (eXtensible Mark-up Language). XML, created by W3C (World Wide Web Consortium) [16], is a derivative of SGML (Standard Generalised Mark-up Language). XML was originally designed for use in large-scale electronic publishing but is now used extensively in the exchange of data via the web. XML documents contain both parsed and unparsed data, with the former being either mark-up or character data (data between a pair of start and end mark-ups). The mark-up encodes a description of the storage layout and logical structure of the document. A mechanism is provided within XML that allows constraints to be imposed on the storage layout and logical structure. The main purpose of XML is to provide a mechanism that can be used in the mark-up and structuring of documents. XML is different to HTML in that tags are only used within XML to delimit pieces of data. The interpretation of the data is left completely to the application that reads it. Another advantage of using XML is that it is possible to easily create new XML tags. With respect to semantic representation, SmartKom [13] and MIAMM [14] both use an XML-based method of semantic representation. It is common that a derivative of XML is used for semantic representation. For example, SmartKom uses an XML-based mark-up language, M3L (MultiModal Markup Language), to semantically

represent information passed between the various components of the platform. An example of M3L is shown in Fig. 3. The M3L code in Fig. 3 is used to present a list of TV broadcasts to the user in response to a user-request. The exchange of information within MIAMM is also facilitated through a derivative of XML called MMIL (Multi-Modal Interface Language). Any programming language can manipulate data in XML and a range of middleware technology exists for managing data in XML format.

```
0<presentationTask> <presentationGoal>
1     <inform> <informFocus> <RealizationType>list </RealizationType> </informFocus> </inform>
2     <abstractPresentationContent>
3<discourseTopic> <goal>epg_browse</goal> </discourseTopic>
4<informationSearch id="dim24"><tvProgram id="dim23">
5   <broadcast><timeDeictic id="dim16">now</timeDeictic>
6                 <between>2003-03-20T19:42:32 2003-03-20T22:00:00</between>
7       <channel><channel id="dim13"/> </channel>
8   </broadcast></tvProgram>
9</informationSearch>
10        <result> <event>
11<pieceOfInformation>
12   <tvProgram id="ap_3">
13<broadcast> <beginTime>2003-03-20T19:50:00</beginTime>
14        <endTime>2003-03-20T19:55:00</endTime>
15        <avMedium> <title>Today's Stock News</title></avMedium>
16        <channel>ARD</channel>
17</broadcast>……..
18          </event>   </result>
19</presentationGoal> </presentationTask>
```

Fig. 3. Example M3L code [13]

4 Decision-Making within MediaHub

The aim of this research is to develop a multimodal platform hub (MediaHub) which will use a potential new approach to decision-making over language and vision data. We will now consider the types of decisions that MediaHub will be required to make. Essentially these can be divided into two main categories:

- Decisions relating to input
- Decisions relating to output

With regard to decisions concerning input, these can be further categorised into the following three areas:

- Determining the semantic content of the input.
- Fusing the semantics of the input. That is, fuse the semantics of the language input such as "Whose office is this?" with the visual input (i.e. the pointing information/data) [12].
- Resolving any ambiguity at the input.

An example of ambiguity at the input could be if the user points three times while saying "Show me the best route from this office to this office" [12]. Here,

synchronisation (e.g. using timestamps) could be used to determine which two offices the user is referring to. Another example could be in an industrial environment where a control technician points at two computer consoles saying “Copy all files from the ‘process control’ folder of this computer to a new folder called ‘check data’ on that computer.” In this example, synchronisation of the visual and audio input may be needed to determine which two computers the control technician is referring to. Resolving ambiguity at the input will be a key objective for the decision-making component of MediaHub.

In relation to decisions at the output, synchronisation issues could arise in order to match, for example, a laser movement with a speech output. As is the case in CHAMELEON [12], a statement of the form “This is the best route from Paul’s office to Tom’s office” may need to be synchronised with the laser output tracing the route between the two offices. A decision may also need to be made on what is the best modality to use at the output (i.e. language or vision?). For example, the directions from one office to another may be best presented visually using a laser, while a response to a user’s query may be better presented using natural language output. Another example could be when the driver of a car asks an in-car intelligent system for directions to the nearest petrol station. Here the system could respond by presenting a map to the driver or by dictating directions using speech output. The system response in this case would depend on whether or not the car was moving. If the car is stopped in a lay-by, the response could be given to the user via the map. If however the car is moving (i.e. the drivers eyes are pre-occupied on the road), then the system would respond using speech output. Of course, there are numerous other possible decisions that will be needed in relation to multimodal input and output in MediaHub. Ultimately, the decisions required in MediaHub will depend on its application. The ideal scenario for a multimodal platform hub is that it will be capable of making all possible decisions that could be required in a multimodal system.

5 System Architecture

MediaHub will be an intelligent multimedia distributed platform hub for the fusion and synchronisation of language and vision data. MediaHub’s proposed architecture is shown in Fig. 4.

The key components of MediaHub are:

- Dialogue Manager
- Semantic Representation Database
- Decision-Making Module

The role of the Dialogue Manager is to facilitate the interactions between all components of the platform. It will act as a blackboard module, with all communication between components achieved via the Dialogue Manager. It will also be responsible for the synchronisation of the multimodal input and output. The Semantic Representation Database in MediaHub will be responsible for holding semantic representations of the multimodal input. Several options exist for semantic

representation including XHTML + V [17], the Synchronised Multimedia Integration Language (SMIL) [18], EMMA (Extensible MultiModal Annotation mark-up language) [19] and the Web Ontology Language (OWL) [20]. The Decision-Making Module will employ Bayesian Networks [21] for decision-making. With regard to multimodal input and output, existing input/output data structures will be assumed.

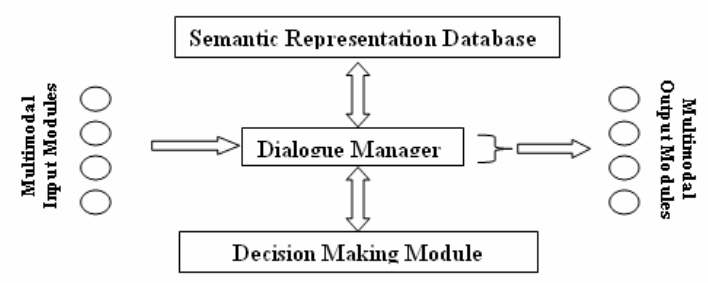


Fig. 4. Architecture of MediaHub

6 Conclusion and Future Work

The objectives of MediaHub, in providing a distributed platform hub for the fusion and synchronisation of language and vision data, have been defined. A review of various existing distributed systems and multimodal platforms has given an insight into recent advances and achievements in the area of intelligent multimedia distributed computing. The various existing methods of multimodal semantic representation, storage and decision-making, which will be of critical importance in the development of MediaHub, were also discussed. The area of Bayesian Networks has been considered with regard to the use of Bayesian decision-making in MediaHub. This provides a potential new approach to decision-making over language and vision data.

The development of MediaHub is in its infancy. Key decisions that lie ahead relate to the method of semantic representation, semantic storage, communication and decision-making. Several implementations of XML could be used by the Semantic Representation Database. XHTML + Voice and other XML-based languages such as SMIL and EMMA will be considered. OWL is also being investigated with regard to its potential use for semantic representation within MediaHub. Further analysis of distributed processing tools, such as OAA and .NET, will also be performed before a final decision on MediaHub's implementation is made.

A major focus of the future development of MediaHub will be in the area of decision-making over multimodal data. The HUGIN software tool [21], a tool for implementing Bayesian Networks as CPNs, is currently being investigated for its potential to provide MediaHub with decision-making capabilities. Hugin offers an API which is implemented in the form of a library written in the C, C++ and Java programming languages. The API can be used like any other library and can be linked to applications, allowing them to implement Bayesian decision-making.

In conclusion, this paper provides a summary of issues relating to the development of an intelligent multimodal distributed platform hub and presents the proposed future development of MediaHub.

References

1. Maybury, M.T. (Ed.): Intelligent Multimedia Interfaces. Menlo Park: AAAI/MIT Press (1993)
2. Mc Kevitt, P. (Ed.): Integration of Natural Language and Vision Processing (Vols I-IV). London, U.K.: Kluwer Academic Publishers (1995)
3. Mc Kevitt, P.: MultiModal semantic representation. In Proceedings of the SIGSEM Working Group on the Representation of MultiModal Semantic Information, First Working Meeting, Fifth International Workshop on Computational Semantics (IWCS-5), Harry Bunt, Kiyong Lee, Laurent Romary, and Emiel Krahmer (Eds.), Tilburg University, Tilburg, The Netherlands, January, 1-16 (2003)
4. Fink, G.A., Jungclaus, N., Kummert, F., Ritter, H., Sagerer, G.: A Distributed System for Integrated Speech and Image Understanding. International Symposium on Artificial Intelligence, Cancun, Mexico, 117-126 (1996)
5. Ma, M., Mc Kevitt, P.: Semantic representation of events in 3D animation. In Proceedings of the Fifth International Workshop on Computational Semantics (IWCS-5), Harry Bunt, Ielka van der Sluis and Roser Morante (Eds.), 253-281. Tilburg University, Tilburg, The Netherlands, January (2003)
6. Cheyer, A., Julia, L., Martin, J.C.: A Unified Framework for Constructing Multimodal Experiments and Applications. In Proceedings of CMC '98: Tilburg, The Netherlands, 63-69 (1998)
7. Kristensen, T.: T Software Agents In A Collaborative Learning Environment. In International Conference on Engineering Education, Oslo, Norway, Session 8B1, 20-25, August (2001)
8. Finin, T, Fritzson, R., Mc Kay, D., Mc Entire, R.: KQML as an Agent Communication Language. In Proceedings of the 3rd International Conference on Information and Knowledge Management CIKM '94, Gaithersburg, MD, USA, 456-463 (1994)
9. Fay, D.Q.M.: An architecture for distributed applications on the internet: Overview of Microsoft's .net platform. In 17th International Parallel and Distributed Processing Symposium, 7-14, Nice, France, April (2003)
10. Vinoski, S.: Distributed object computing with CORBA. C++ Report, Vol. 5, No. 6, July/August, 32-38 (1993)
11. Thórisson, K.R.: A Mind Model for Multimodal Communicative Creatures & Humanoids. In International Journal of Applied Artificial Intelligence, Vol. 13 (4-5), 449-486 (1999)
12. Brøndsted, T., Dalsgaard, P., Larsen, L.B., Manthey, M., Mc Kevitt, P., Moeslund, T.B., Olesen, K.G.: The IntelliMedia WorkBench - An Environment for Building Multimodal Systems. In Advances in Cooperative Multimodal Communication: Second International Conference, CMC '98, Tilburg, The

Netherlands, January 1998, Selected Papers, Harry Bunt and Robbert-Jan Beun (Eds.), 217-233. Lecture Notes in Artificial Intelligence (LNAI) series, LNAI 2155, Berlin, Germany: Springer Verlag (2001)
13. Wahlster, W.: SmartKom: Symmetric Multimodality in an Adaptive and Reusable Dialogue Shell. In: Krahl, R., Günther, D. (Eds.), 47-62, Proceedings of the Human Computer Interaction Status Conference, June. Berlin, Germany: DLR (2003)
14. Reithinger, N., Lauer, C., Romary, L.: MIAMM: Multimodal Information Access using Multiple Modalities. In Proceedings of the International CLASS workshop on Natural, Intelligent and Effective interaction in MultiModal Dialogue Systems, Copenhagen, Denmark, 28-29 June (2002)
15. Minsky, M.: A Framework for representing knowledge. In Readings in knowledge representation, R. Brachman and H. Levesque (Eds.), 245-262, Los Altos, CA: Morgan Kaufmann (1975)
16. World Wide Web Consortium http://www.w3.org
17. Mc Tear, M.F.: Spoken dialogue technology: toward the conversational user interface. Springer Verlag: London(2004)
18. Rutledge, L.: SMIL 2.0: XML For Web Multimedia. In IEEE Internet Computing, 78-84, Sept-Oct (2001)
19. EMMA http://www.w3.org/TR/2004/WD-emma 20041214/
20. OWL http://www.w3.org/2004/OWL/
21. Jensen, F.: Bayesian belief network technology and the HUGIN system. In Proceedings of UNICOM seminar on Intelligent Data Management, Alex Gammerman (Ed.), 240-248. Chelsea Village, London, England, April (1996)

Lexical Functional Grammar constraints and concurrent constraint programming

Peter Hancox

School of Computer Science, University of Birmingham, Birmingham, B15 2TT, England
P.J.Hancox@cs.bham.ac.uk,
WWW home page: http://www.cs.bham.ac.uk/~pjh/

Abstract. Lexical Functional Grammar allows linguistic constraints to specify attributes and their values without using unification. The satisfaction algorithm for these constraints is within the generate-and-test paradigm and has the disadvantage of not being able to detect, at minimal cost, violations of the constraints as early as native speakers. Concurrent constraint languages, of which CHR is an example, allow searches to be incrementally constrained with goals delayed until they can be properly discharged. It is shown that linguistic constraints can be implemented in CHR to give early detection of satisfaction/violation of constraints, and also allows some further detection of redundancy and inconsistency.

1 Unification and constraints

Grammars used in syntactic parsers are, now, widely based on unification. The family of unification grammars has a broad membership from the computational: Prolog's Definite Clause Grammar (DCG); to the linguistic, for instance Head-Driven Phrase Structure Grammar and Lexical Functional Grammar (LFG). These formalisms typically use both syntactic and lexical categories (noun phrase, noun) and attribute/value pairs. The latter allow the expression of grammatical information, for instance the grammatical features of number, person, tense and so on.

Unification is used as the major information combining mechanism in these formalisms. Attributes and their values can be seen as collected into bundles and bundles can be unified to produce the attribute bundles of other linguistic objects. As an example, consider the combination of features for a determiner and noun which, together, give a feature bundle for a noun phrase:

$$\text{DET/the} \quad + \quad \text{NOUN/car} \quad \rightarrow \quad \text{NP/the car}$$

$$\begin{bmatrix} \text{PERS} & \text{3RD} \\ \text{SPEC} & \text{THE} \end{bmatrix} \quad \begin{bmatrix} \text{NUM} & \text{SING} \\ \text{PERS} & \text{3RD} \end{bmatrix} \quad \begin{bmatrix} \text{NUM} & \text{SING} \\ \text{PERS} & \text{3RD} \\ \text{SPEC} & \text{THE} \end{bmatrix}$$

The intimate association between attributes/values and unification has several implications. An obvious aspect is that it is relatively easy to impose a consistency requirement on feature bundles such that, for any feature, it shall have no more than one value. Thus, it is impossible for the number attribute to appear simultaneously with both the values of singular and plural. Second, the use of unification allows attributes to be referred to although, at the point of reference, they do not have a value. In programming language terms, it is the equivalent of accessing a variable, whether or not that variable has been set to a value.

The third consequence of unification is less obvious: it tends to lead the parser implementer into search algorithms of the generate-and-test variety. Classically, parsing with unification consists of generating partial structures, the validity of which are tested when they are unified. This is clearest in the most commonly used version of unification grammar, DCG, which, using Prolog's in-built top-down depth-first search, hypothesises (generates) structures which are tested and backtracked over when the test fails.

The work described here shows the weakness of the generate-and-test model of LFG constraints and demonstrates a new model of incremental constraints implemented in the concurrent programming language CHR. Not only does this model discharge constraints in an intuitively correct way but it also allows the detection of redundant and inconsistent constraints.

1.1 LFG and unification

LFG, while firmly in the unification grammar family, also uses alternative methods of specifying the values of attributes.

LFG uses two main representations of structure: c-structure is generated from the application of essentially context-free grammar rules; f-structure is a feature/attribute structure that allows values of attributes to be either atomic values (eg SING, 3RD, +) or f-structures. Thus, an f-structure may consist, in part, of embedded f-structures, although there are theoretical restrictions on the depth of embedding. As will be shown later, f-structures (including embedded f-structures) have important well-formedness conditions imposed upon them. While c-structure represents constituent structure in the same way as phrase structure in traditional syntax, f-structure represents the functional structure of sentences, setting it out as essentially a predicate with associated information (eg tense, mood, aspect) and the functions that the predicate governs (eg subject, object).

LFG specifies the production of c- and f-structures by schemata (equations) attached to context-free-like grammar rules and lexical entries. As an example, a NP could be specified in LFG as:

$$\begin{array}{l} \text{NP} \rightarrow \text{DET NOUN} \\ \qquad \uparrow = \downarrow \; \uparrow = \downarrow \end{array}$$

the: DET, $(\uparrow \text{SPEC}) = \text{THE}$
car: NOUN, $(\uparrow \text{NUM}) = \text{SING}$
$(\uparrow \text{PRED}) = \text{'CAR'}$

The $\uparrow = \downarrow$ schema states that the attributes and their values at the current node (eg DET or NOUN) should be unified with those of the higher node (NP).

It is simple to specify that the bundled attributes of a lower node should be the embedded value of a feature at the dominating node. Two typical rules from English are:

$$\begin{array}{lccc} \text{S} \rightarrow & \text{NP} & & \text{VP} \\ & (\uparrow \text{SUBJ}) = \downarrow & & \uparrow = \downarrow \\ \text{VP} \rightarrow & \text{V} & & \text{NP} \\ & \uparrow = \downarrow & & (\uparrow \text{OBJ}) = \downarrow \end{array}$$

In the first, the resulting f-structure will have a SUBJ attribute which will have as its instantiation the f-structure generated by the NP rule. In the second rule, there will be an embedded f-structure as the instantiation of the OBJ function. An f-structure can look like the following taken from [4]:

$$\begin{bmatrix} & \begin{bmatrix} \text{NUM} & \text{SING} \\ \text{PRED} & \text{'GIRL'} \\ \text{SPEC} & \text{A} \end{bmatrix} \\ \text{SUBJ} & \\ \text{PARTICIPLE} & \text{PRESENT} \\ \text{PRED} & \text{'WASH}\langle(\uparrow \text{SUBJ})\ (\uparrow \text{OBJ})\rangle\text{'} \\ & \begin{bmatrix} \text{NUM} & \text{SING} \\ \text{PRED} & \text{'CAR'} \\ \text{SPEC} & \text{THE} \end{bmatrix} \\ \text{OBJ} & \end{bmatrix}$$

1.2 LFG and constraints

In addition to unification, LFG allows schemata to express constraints on the values attributes can take. These constraints are of two types: *constraining equations* that constrain the value an attribute can take and *existential constraints* that require a feature to be instantiated or uninstantiated. By the end of parsing, all constraints must be satisfiable.

Constraining equations It is possible to constrain a feature to be instantiated only to a specific value, while not actually causing a unification action:

$(\uparrow \text{SUBJ NUM}) =_c \text{SING}$

This states that the NUM feature embedded within the SUBJ feature bundle must be both instantiated and instantiated to SING. (It is possible to imagine a weaker version of this constraint: either the feature is instantiated to SING

or it remains uninstantiated.) Negative constraining equations are also available which, as their name suggests, require a feature either to have a value other than that given or to be uninstantiated:

$\neg$ (↑ SUBJ NUM) $=_c$ SING

The use of these constraints is often to enforce agreement without accidental unification. So, it is possible to specify that the NUM feature of an f-structure agrees with its subject's number feature, thus enforcing subject/verb agreement of English without, in the process, instantiating either feature:

(↑ SUBJ NUM) $=_c$ (↑ NUM)

Existential constraints These constraints specify that a named feature must be instantiated (without specifying what that value must be) or that a feature must remain uninstantiated.

1.3 Well-formedness conditions

LFG has three well-formedness conditions which it imposes on f-structures. As seen above, consistency requires that each feature has no more than one value. The other two, completeness and coherency, require a more detailed understanding of f-structure. Within each f-structure there should be an instantiated PRED feature. When this is contributed by a verbal lexical entity, this will include a specification of the governable functions that can occur with that verbal entity. The lexical entry used earlier:

wash: V, (↑ TENSE) = PAST
(↑ PRED) = 'WASH⟨(↑ SUBJ)(↑ OBJ)⟩'

specifies that *wash* must occur with the two fully-formed grammatical functions of subject and object. This allows *Mary washed the car* but disallows **washed the car* because it is not complete: its lack of a subject leaves one governable function unsatisfied.

**Mary washed the car the sponge* is not coherent because it contains more governable functions than the PRED specification requires.[1]

A governable function is held to be instantiated in an f-structure if its own, local PRED feature is itself instantiated (and is well-formed). So, returning to the example given above, it is complete because the f-structure's PRED feature's governable functions (SUBJ and OBJ) both have instantiated features. It is coherent because no governable functions other than those specified in the top-level PRED feature occur in the f-structure.[2]

[1] A comparison should be made with syntactically equivalent sentences such as: *Mary gave the baby the book.* It is possible to have sentences such as *Mary washed the car with a sponge* and *Mary washed the car with Jane* but the prepositional phrases act as adjuncts to the sentence.

[2] The set of governable functions is essentially defined by all those feature names that appear in PRED specifications.

The well-formedness conditions can be modelled using existential constraints. Completeness is modelled by adding an existential constraint on each governable function in the PRED specification. Coherency is modelled by adding a negative existential constraint on each governable function not given in the PRED specification. Assuming the existence of governable function OBJ2, the well-formedness example used above would be constrained as:

(↑ SUBJ PRED) ∧ (↑ OBJ PRED) ∧ ¬ (↑ OBJ2 PRED)

(Alternatively, the last constraint could be written as ¬ (↑ OBJ2).)

2 Testing constraints with generate-and-test

The original detailed description of LFG sets out a generate-and-test algorithm for constraints. The c-structure and f-structure are to be created with constraints collected. Once these structures are generated, constraints are tested for satisfaction along with testing for well-formedness. This approach has the virtues of being easy to explain and relatively easy to program.

Because of the family relationship between DCG and LFG, it is relatively straightforward to implement LFG in DCG. This suffices as a technique for experimentation and demonstration although, for practical purposes, Prolog has inherent disadvantages, for instance in its lack of structure sharing and inability to handle left-recursive rules. Here, DCG is used to model LFG's c-structure, with Prolog's term unification used to model attribute/value unification. LFG constraints, including well-formedness conditions, are collected to be solved in a post-parsing check.

Here, a small abstract grammar and lexicon is used as an illustration:

ntA → ntB ntC ntB → ta tb ntC → ta ntB

ta → a tb → b tc → c

The first grammar rule can be implemented in DCG as:

```
ntA(ntA(NTB, NTC),
    fs(fA(FA),fB(FB),fC(FC),f1(F1),f2(F2),f3(F3),f4(F4)),
                                                Const0, Const) -->
     ntB(NTB, FA, Const0, Const1),
     ntC(NTC,
         fs(fA(FA),fB(FB),fC(FC),f1(F1),f2(F2),f3(F3),f4(F4)),
         Const1, Const).
```

and a lexical entry as:

```
ta(ta(a),
   fs(fA(_),fB(_),fC(_),f1(1),f2(2),f3(_),f4(_)), Const, Const)
                                                        --> [a].
```

In a classic generate-and-test approach, constraints have to be collected, for instance in a difference list (in this example, in the variables `Const0, Const1, Const`) and then checked in a post-parsing test. The following (unrealistic) DCG clause shows how the various constraints could be coded:

```
ta(ta(a),
   fs(fA(_),fB(_),fC(_),f1(F1),f2(F2),f3(F3),f4(F4)),
   Const, [pos_const(F1, 1), neg_const(F2, 1), pos_exist(F3),
           neg_exist(F4)|Const])
                                                        --> [a].
```

(Such constraints could also be attached to grammar rules.) Well-formedness conditions can be implemented as outlined above:

```
ta(ta(b),
   fs(fA(fs(fA(_),fB(_),fC(_),f1(_),f2(FA2),f3(_),f4(_)),
      fB(fs(fA(_),fB(_),fC(_),f1(_),f2(FB2),f3(_),f4(_)),
      fC(fs(fA(_),fB(_),fC(_),f1(_),f2(FB2),f3(_),f4(_)),
      f1(_),f2(2),f3(_),f4(_)),
   Const, [pos_exist(FA2), pos_exist(FB2), neg_exist(FC2)|Const])
                                                        --> [b].
```

In the post-parse check, each member of the constraint list has to be checked to ensure it is satisfied, with code provided for testing each kind of constraint, eg:

```
test_constraint(pos_exist(Var)) :-
     nonvar(Var).
```

The generate-and-test technique doesn't seem intuitively correct and it is easy to see that some constraint violations that could be detected early are, in fact, detected late. It is possible to get an improvement in performance, most obviously by checking for constraint violations (and satisfactions) at the end of the processing of each non-terminal rule or, indeed, after processing each syntactic category. While the search tree would be pruned, there would be an increased cost in computation time. Such modifications don't change the essential generate-and-test inefficiencies of the problem.

Intuitively, the generate-and-test technique seems wrong. Suppose a constraint is used, as shown above, to impose subject/verb agreement. Then, for the sentence:

**a car are washed by the children*

the agreement constraint would not be applied and the ungrammaticality detected until the end of the sentence. However, a native speaker would detect the error as early as:

**a car are*

Looking at the problem from a programming perspective, it becomes a matter of when the satisfaction or violation of a constraint is detected. (The early detection of violations is particularly important as it triggers backtracking as soon as possible - in effect pruning the search tree as early as possible.)

3 Early detection of satisfactions and violations

Constraining equations are capable of early satisfaction and violation. For both negative and positive constraining equations, the testing of the constraint only need be delayed until the constrained feature is instantiated. In a strict interpretation of the positive constraint, it may be important that the constrained feature is not left uninstantiated: this is discussed below.

Existential constraints present a more complex problem. For a negative existential equation, violation can be detected as soon as the attribute it references is instantiated. If the attribute is left uninstantiated at the end of the parse, this is the equivalent of satisfaction. Note that there is no explicit satisfaction here, but this may not be significant in implementation. The satisfaction of positive existential constraints can be detected as soon as the attribute is instantiated. However, violations can't be detected until the whole input has been parsed and then only by some form of post-parsing check.

The strict interpretation of constraining equations becomes quite easy to model in this approach: it is formed of a weak constraining equation with existential equations added to ensure instantiation of attributes.

Thus, a successful implementation will be one where the evaluation of constraints is delayed, as far as possible, for only as long as attributes remain uninstantiated. The remainder of the paper presents and discusses an implementation in the concurrent constraint language, CHR. This allows incremental addition of constraints and also the manipulation of the constraint store which, in turn, allows for the detection of inconsistencies to an extent not envisaged in descriptions of LFG[3].

4 CHR: a concurrent constraint language

Constraint Handling Rules (CHR)[3] is a general purpose constraint specification language that can be used in conjunction with several programming languages. In this investigation it is used with SICStus Prolog. CHR is sufficiently general that it is possible to implement the widely used constraint languages, eg CLP(B), CLP($\mathcal{R}$) and CLP(FD).

CHR is a variety of concurrent constraint programming language where constraints are extended by adding dynamic scheduling. Constraints can be seen as processes which execute concurrently to the main program (in this case the DCG parser), communicating through the global constraint store[5]. The experimental implementation described below takes advantage of this to test for satisfaction and violation but also to introduce some pruning of the global constraint store and to detect logical inconsistencies.

[3] Space limitations preclude a review of an early implementation in Prolog II[2] and an explanation of why Constraint Logic Programming over Finite Domains (CLP(FD))[1] does not provide the means to solve the problem.

4.1 LFG constraints in CHR

A concurrent constraint rule has the form:

```
Process :- Guard | Goal.
```

where *Process* is the process defined by the rule (and, unlike Prolog, this can consist of more than one "head"); *Goal* is the body of the rule (in the usual Prolog rule sense), and *Guard* is the delay condition that specifies when the rule can be fired.

When a CHR rule is first called, woken (by a variable being touched) or reconsidered (in backtracking), the guard is executed. Should the guard fail, then the next rule is tried or (if there are no alternative rules) the CHR rule is delayed until a variable is again touched.

If the guard succeeds, then the rule fires. For propagation rules (written with `=>`) the body of the rule is executed without removing any constraint. Simplification rules (written using `<=>`) remove the matched constraint or constraints from the global constraint store and the body is executed. Simpagation rules (written with \) are a combination of propagation and simplification rules, such that any constraints preceding "\" are kept in the global constraint store and those following are deleted.

The weak versions of the constraining equations are based on the use of the Prolog predicates *ground/1* and *var/1*. For the positive constraining equation the constraints are:

```
pos_const_weak(Attribute, Value) <=> ground(Attribute),
     ground(Value), Attribute == Value | true.
pos_const_weak(Attribute, Value) <=> ground(Attribute),
     ground(Value), Attribute \== Value | fail.
```

The negative constraining equations are similarly coded. Strict versions of these constraining equations are coded as a combination of weak constraining equations and existential constraints:

```
pos_const_strict(Attribute, Value) :-
     pos_const_weak(Attribute, Value),
     pos_exist(Attribute),
     pos_exist(Value).
```

Existential constraints are coded very simply:

```
pos_exist(Attribute) <=> ground(Attribute) | true.
neg_exist(Attribute) <=> ground(Attribute) | fail.
```

4.2 Discussion: satisfaction and violation of constraints

The concurrent nature of CHR allows constraints to be added incrementally during search, unlike the constrain-and-generate approach of CLP(FD). Incremental constraining exactly matches the "intuitive" view of LFG constraints set out above. A computation in CHR is not so much a tree search but a series of transitions between states. The posting of a constraint moves the program into a new state and the satisfaction or violation of a constraint is yet another state.

These constraints can be added to a DCG as embedded subgoals:

```
ta(ta(a),
   fs(fA(_),fB(_),fC(_),f1(_),f2(F2),f3(_),f4(_))) --> [a],
     { pos_const_weak(F2, 2) }.
```

In this implementation, LFG's constraining equations are discharged as soon as their variables are instantiated (without the programmer having to pass messages or set flags). If the arguments are not instantiated, then the constraints remained in the store at the end of processing, without affecting the success of the parse. As has been seen above, the strict interpretation of the constraining equations is provided by simply adding existential constraints to the weak constraining equations.

The violation of negative existential constraints is detected as soon as the constraint's variable is instantiated. Positive existential constraints present more of a challenge. If they are satisfied (ie their variable is instantiated) during parsing, they are immediately removed from the constraint store. However, if they are not satisfied then they are, necessarily, violated but this can only be detected by a post-parsing check[4]:

```
constraint_check :-
     findall_constraints(pos_exist(_), List_of_Pos_Exists),
     ground(List_of_Pos_Exists).
```

If compared with the model of LFG constraint processing set out in the objections to generate-and-test, it can be seen that the CHR version succeeds in implementing the detection of constraint satisfaction or violation at exactly the correct states in the parse process. As will be shown below, the use of incremental constraints programmed in CHR allows for the detection of inconsistent combinations of constraints.

4.3 Extensions to the CHR model

The power of CHR allows the specification of other rules that either prune the global constraint store or detect inconsistencies. Simpagation rules can be used to remove duplicate constraints, as for the positive existential constraint:

[4] As the strict versions of the constraining equations are programmed using existential constraints, they also rely on this post-processing check.

```
pos_exist(Attribute) \ pos_exist(Attribute) <=> true.
```

where the guardless rule will delete the second `pos_exist(Attribute)` from the store.

More significantly, some inconsistencies can be detected. The obvious contradictions are when a positive constraint occurs in the store with a corresponding negative constraint:

```
pos_const_weak(Attribute, Value), neg_const_weak(Attribute, Value)
     <=> fail.
pos_exist(Attribute), neg_exist(Attribute) <=> fail.
```

Less obvious is the inconsistency of a negative existential constraint on an argument of a positive constraining equation which, while not exactly a contradiction, seems inconsistent:

```
pos_const_weak(Attribute, Value), neg_exist(Attribute)
     <=> fail.
pos_const_weak(Attribute, Value), neg_exist(Value)
     <=> fail.
```

This is an extension to the processing of LFG constraints not explicitly set out in the detailed description of LFG.

5 Significance and conclusions

The significance of this work is that it provides an implementation of LFG's constraint system that, rather than following the standard generate-and-test description, uses concurrent constraint programming to implement an incremental constraint model. In so doing, it demonstrates that inconsistencies in LFG constraints can be detected early. While this application is specific and detailed, the principle of modelling with incremental constraints acting as agents has wider applicability, for instance in the modelling of reference in natural language as constraints waiting to be solved.

References

1. Carlsson, M., Ottosson, G,. Carlson, B.: An open-ended finite domain constraint solver. *Proceedings of the 9th International Symposium on Programming Languages: Southampton, 3-5 September 1997.* Berlin: Springer, 1997. 191–206.
2. Eisele, A.: A Lexical Functional Grammar system in Prolog. Department of Linguistics, Stuttgart University, 1984.
3. Frühwirth, T., Abdennadher, S.: *Essentials of constraint programming.* Berlin: Springer, 2003.
4. Kaplan, R. M., Bresnan, J.: Lexical-Functional Grammar: a formal system for grammatical representation. *In: The mental representation of grammatical relations*, edited by J. Bresnan. Cambridge, Mass: MIT Press, 1982. 173–281.
5. Marriott, K., Stuckey, P.J.: *Programming with constraints : an introduction.* Cambridge, Mass: MIT Press, 1998.

Enhancing Information Retrieval Interfaces with Information Foraging

Cathal Hoare and Humphrey Sorensen

Department of Computer Science, University College Cork, Ireland

Abstract. Information Foraging, first described by Pirolli and Card, is an innovative model, based on a gain function, which allows IR practitioners and user-interface designers to describe and evaluate information retrieval interface techniques. This paper describes how under-utilised aspects of the model can be used to improve the performance of information retrieval interfaces. By combining recommender systems and user interfaces, performance can be improved. By identifying initial and subsequent clusters, identifying when a forager should leave a document cluster or patch and finally, identifying documents that provide high information gain, the productivity of an interface can be improved.

1 Introduction

Information Foraging, first described by Pirolli and Card [21], is an innovative model that allows IR practitioners and user-interface designers to describe and evaluate information retrieval interface techniques. Based on the analogy of animals in an ecosystem, it describes information as a source of nutrition and information seekers - info-vores in information foraging parlance - as consumers of that information. Information has scent, cues that help info-vores to locate it. That scent can be enhanced or diminished by the surrounding environment, e.g., if there are a lot of confusing scents, it would increase the difficulty of locating an interesting item of information.

The concept of information scent has been utilised to evaluate and enhance many interface designs [15][22]. ScentTrails[19], for example, assists users in finding relevant information in an e-commerce environment. Pirolli and Card applied information foraging to create a cluster based information retrieval interface in [21]. In traditional information retrieval interfaces, scent represents any cue that allows a user to locate information of interest. For example, many commonly used search engine interfaces use a short descriptive text summary to indicate the content of a particular result. These interfaces also indicate scent in a subtler way: by ranking results, users are implicitly drawn to information patches, areas where relevant information is located, which are considered, by one metric or another, to be useful.

Certain information may have greater nutritional content, or value, than others; information diet has a significant influence on gain. For instance, in the field of text-based information retrieval, a summary document might be

of greater value to a researcher than would a document focussing on only one detail of interest. Relevant artefact density within a patch is also significant. If the relevant artefacts within a patch are hidden by the noise of other irrelevant artefacts, the rate of gain will decrease.

Pirolli and Card examined the application of Charnov's Marginal Gain Theorem to the foraging task. By doing this, they created a model that allows comparison of information retrieval interfaces. The model is based on the following gain function

$$R = \frac{\lambda g}{1+\lambda t_w} \text{ [Eq. 1]}$$

Fig. 1 illustrates graphically the application of the model. It depicts the fact that information seekers often move between useful patches of information, spending some time browsing within a patch before moving onto another; choosing the best patch, not browsing it exhaustively, and knowing when to move on are aspects that might assist the user. Reproduced from [20], it shows three lines R^1, R^2 and R^* that intersect with the gain function $g_i(t_{wi})$. Pirolli and Card show that the slope of these lines is the average gain, that is the amount of gain divided by the time spent between clusters and time within an information path. Ideally, interfaces should seek to minimize t_b and t_w, while gain should be maximized.

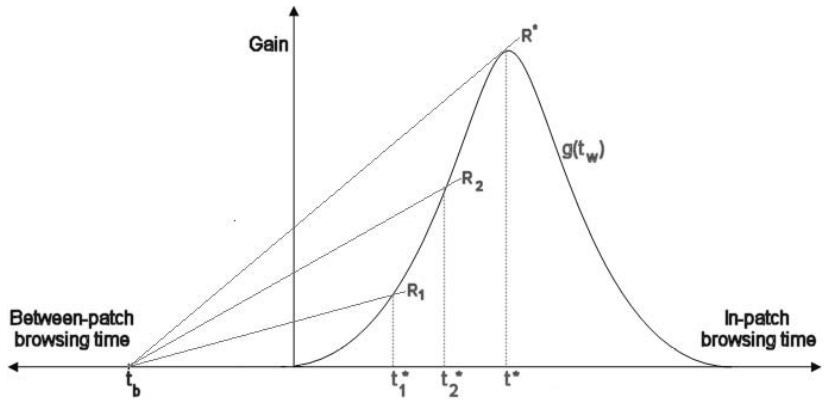


Fig. 1. Charnov's Marginal Gain Theorem

The authors believe that Charnov's Marginal Gain theorem identifies three major deficiencies in many cluster based information retrieval interfaces:

– While many interfaces allow a focussed search of an information patch, locating the best initial patch, and suitable patches subsequently, is expensive as no cue is given to indicate which patch is next best; t_b in figure 1 is not minimized.

– Further, when browsing the information patch, a user is not 'fed' items of high nutritional value; all artefacts have the same value and $g_i(t_{wi})$ is not maximized.
– Finally, when browsing within a cluster, users are not provided with an indication that all nutritional value has been consumed; several documents could consider the same facts from the same point of view. Having read one of these documents, the others may not make a significant contribution to the information seeker's understanding of the topic. Users should receive an indication that the potential gain from the cluster has diminished, and an indication of where artefacts that provide gain can be located. This feedback would allow a reduction in t_w.

In the remainder of this paper, we will examine several types of interface from the perspective of Charnov's Marginal Gain theorem. The authors will then present their proposal for solving these deficiencies by combining recommender systems with user interfaces. Finally, a scenario of usage for the hybrid application is presented.

2 Visualisation and Information Foraging

The Information Foraging model can be applied to any information retrieval interface. The authors believe that info-vores can best explore an information space when it is represented in two or more dimensions.

1-dimensional representations are most common in information retrieval, and are exemplified by the list; usually these lists are ordered to produce a ranked list. Each item in a result set presented as a ranked list exists as a singleton; no relationship exists between them, other than their common relevance to some query. On finding an item of interest in a ranked list, the examination of the list must continue exhaustively to find other relevant items.

For higher dimensionality representations, some type of information map is employed, of which there are several variants. In an information map items that are similar are located in proximity to one another: once one item of interest is found, other similar items will be located in the same area of the visualisation. This is known as clustering, a task for which research on the cluster hypothesis [11] has developed algorithms to partition collections into sets of artefacts that are similar to one another based on some metric. In Information Foraging terminology, these clusters are known as Information Patches.

When a simple ranked list and clustered view are modelled using Information Foraging Theory, the differences are manifest. A ranked list, as explained, must be searched exhaustively. This means that the rate of gain is low as the effort expended locating relevant items is high. As each artefact in a ranked list is a singleton 'cluster', time between relevant clusters is relatively high; users must check each cluster to examine its relevance, reducing gain. When using clustering, the rate of gain is higher. Due to the consequences of the cluster hypothesis, information patches, which contain numerous artefacts, can be found

reducing the effort required to find relevant artefacts. The advantage of the clustered view is apparent, in Fig. 2, when a ranked list and clustered view are informally modelled using the Information Foraging Model. Cluster based views of artefact collections or sub-collections have been produced using both custom visualisations[1][14][24][18] and variations of the ranked list[3].

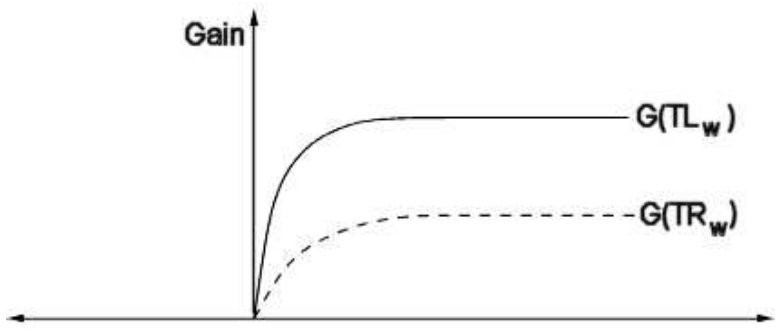


Fig. 2. Ranked List verses Clustered View

Spence [25] explained the significance of producing appropriate visualisations when he described the concept of an internal model, or cognitive map, to explain how a user develops an understanding of a concept or dataset. Often the internal model for one concept is reinforced through exploration, asking questions, and deriving conclusions from existing and trusted internal models. Internal models can be more readily interpreted if a suitable externalisation of the dataset is present. Externalisation can aid the formation of internal models by exploiting pre-attentive processing. Cognitive factors that can be exploited by pre-attentive processing include associative memory, visual memory and spatial ability which are captured by the Gestalt principles [2]. Specifically, we chose to exploit the principle of proximity which states that items placed near each other appear to form a group. Users mentally associate proximally positioned objects with one another. On laying out nodes proximally, spatially arrayed graphical symbols create a map of an artefact collection, usually based on similarity of artefacts to one another. Each symbol or glyph can be carefully encoded with further information through, for example, colouring or shape [5]. Rao [17] captures the advantage of information maps when he described the advantages of *focus + context*, where a user can examine or focus on specific items in a collection, while remaining aware of the context or relationship between that item and the collection as a whole. Thus, information maps provide a collection overview that allows rapid navigation and provides context for artefacts in the visualisation.

2.1 Information Map Implementations

Numerous novel approaches have been used to generate information maps. We examine some, though by no means all of those implementations now. Initially this work concentrated on visualising co-citation maps. VR-VIBE [1] was an early attempt to visualise the results of a query to a document search engine. Written in VRML, VR-VIBE creates a visual browsing environment that relates individual documents to a set of keywords through their proximity in the visualisation. The user-defined keywords are placed in the environment; they may be weighted according to importance. A document's position in the environment is determined by the occurrence of each keyword in that document; if in a particular document one keyword is more prevalent than others, then that document's icon will be located closer to the keyword in the virtual scene. As a result, ad-hoc clusters form. The systems relate documents to one another based on their relationship with a small set of keywords. Several issues can arise because of this arrangement, for example, homonyms can cause unrelated document to be grouped together.

Kaski et al [14] applied Kohonen networks to organise very large document collections. These Self-Organising Maps (SOMs) create contour maps, colour coding dense areas of the map to indicate groups of documents that are related to one another. Peaks are decorated with words or phrases that are indicative of the topicality of that area of the map. Navigation functions include zooming and keyword searches to enable users to focus on a particular part of the map.

Hierarchical agglomerative clustering techniques, based on term frequencies within a document have been combined with Treemaps [24] to produce visualisations of document collections. Treemaps use a space-filling algorithm to fill recursively divided rectangle areas with components of a hierarchy; in this case the hierarchy is produced by the clustering algorithm.

Lighthouse [18] present document relationships in 2 and 3-dimensions, allowing users to navigate the information space and select individual items for further examination. Like VR-VIBE, an explicit document clustering stage did not take place. In Lighthouse, the visualisation was combined with ranked lists to maintain a user's search context while browsing. By so doing, the authors highlight the retention of context information on both the documents-query and document-document relationships.

We will now introduce our information foraging tool and client; it combines visualisation, ranked list and explicit clustering of documents in a result-set to facilitate information foraging.

3 The SolonEvo Application

The SolonEvo [10] application provides users with access to an information map that explores and leverages information foraging theory. Built on a client-server architecture, the application indexes document collections allowing users to submit queries about the collection's contents. The results of the search are displayed

in a proximity based visualisation. The application incorporates two tools to aid information foraging: the proximity and thresholding tools. The architecture is easily extended allowing incorporation of new presentation techniques and tools.

The visualisation lays out artefacts proximally based on similarity to one another, an intuitive metaphor that provides a high degree of affordance to the user. Artefact similarities were calculated using the Smart IR system [23]. Taking an adjacency matrix of similarity measurements as a parameter, the algorithm first calculates the Minimum Spanning Tree (MST) for the graph using an implementation of Prim's Algorithm [9]. The MST is then partitioned by removing the weakest edge in the tree. Each partition forms a cluster. When the desired number of clusters have been created, the resulting tree is passed to the layout algorithm. The application implements a modified version of the Fructerman-Reingold [7] force-directed layout algorithm [8]. The original algorithm has been modified to emphasise attractive forces and lower repulsive forces, so as to convey proximity between elements in the collection. Further, drawing attributes such as minimizing edge-crossings and evenly distributing nodes on the drawing area are discarded in our algorithm. These features are not required in the IR domain and removing the features reduces execution complexity.

The visualisation is embedded in the application along with its controls and a standard ranked list view of the result set. The ranked list is provided to facilitate foraging by providing an initial point for exploration and also to reassure users familiar with the traditional approach. The application appears as shown in Fig 3. The proximity based visualisation appears on the left, with a proximity and thresholding controls on the top right and the ranked list on the bottom right. The proximity tool allows a user to focus their search on a particular information patch.

In the case that clusters presented in the visualisation are too finely grained or, conversely, too coarse, the thresholding tool allows a user to adjust the density of relevant artefacts in a cluster. When a user changes the threshold level, the minimum-spanning tree derived from artefact similarity measurements is repartitioned creating a new set of clusters that are displayed in the visualisation.

To date the application has been evaluated using questionnaire type usability tests that have fed into the development process. A new task-based usability test, using the TIPSTER document collection to assess the quality of the application, is underway. The information foraging based evaluation technique proposed by [13] will also be used to determine the performance of the application. The rate of information gain for users, over a fixed time period, using ranked lists, the current version of SolonEvo and proposed evolution of SolonEvo will be evaluated for a variety of tasks.

4 Recommenders and Information Maps

The use of information maps in cluster presentation achieves one objective of information foraging: it presents likely feeding patches to info-vores in an easily comprehended manner. On its own, however, this is not sufficient. A more com-

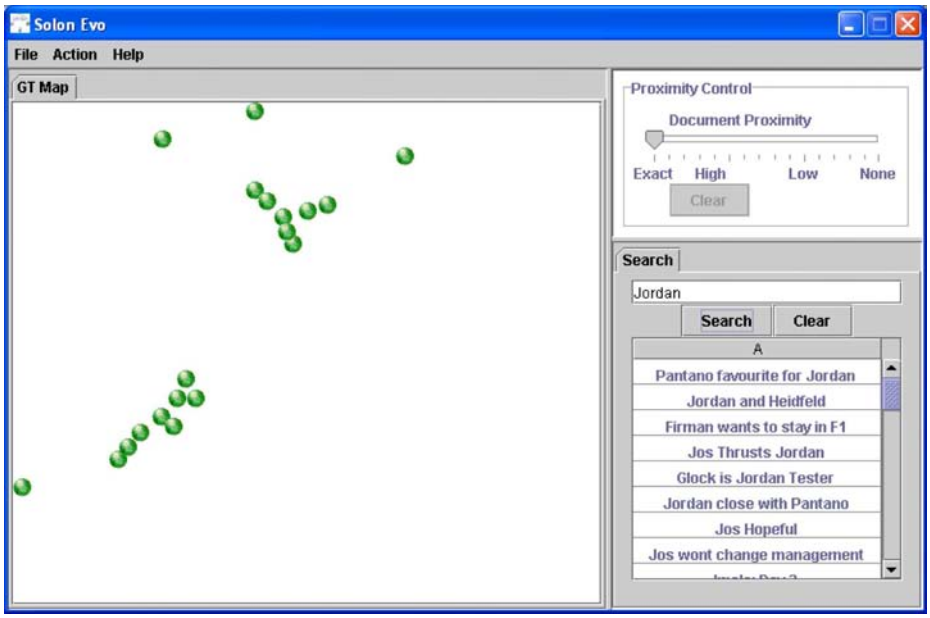


Fig. 3. Screenshot of Application

prehensive approach would guide the user toward the (next) best feeding patch and would suggest that a patch, when exhausted, should be abandoned for more fertile ground.

We believe that such objectives can be best achieved by integrating recommender systems with information maps. Recommender systems possess an ability to suggest courses of action to users. These suggestions can be based on the previous actions of individuals and/or on collaborative groups of users. These systems have evolved from single-shot models of recommendations, where all suggestions are based on a user's long-term history, to content-based recommenders that engage the user at search time to discern their immediate short-term information need. Numerous implementations have evolved, such as ISPY[6] and GroupLens[16]. These applications use web-based ranked list type interfaces. The authors are, after an initial literature review, unaware of any implementation that combines recommenders and visualisations. The authors believe that such a combination will overcome the shortcomings identified by modelling visualisations with information foraging theory.

By combining these visualisation techniques and recommender technologies, the authors believe that three important optimisations can be made to the information map concept. The ideal case requires two conditions to be met. The first is for the user to match the slope of the tangent, from the x-axis to the point of maximum rate of gain, to the slope of the gain function. This ensures that between-cluster time has been minimized, improving the rate of information gain. Secondly, the slope of the gain function should be maximised. This ensures

that users are retrieving a high rate of relevant artefacts. For the first condition to be met the between-patch time must be minimised. Also, the user should be made aware of when they have consumed all unique semantic content. For the second condition to be met, the time spent achieving a maximum rate of gain must be minimised.

4.1 Minimising Between-Patch Time

Between-patch time can be minimised by intelligently placing the clusters produced by SolonEvo. Therefore the position where users would first look at a visualisation must be identified. This may be the centre of the visualisation or another area. The authors will conduct a set of empirical experiments to identify this location. These experiments will use eye-tracking tools such as EyeLink[4] to examine whether there is a trend towards looking at a particular part of the visualisation or whether adding a glyph will assist in attracting a user's attention.

The system must also decide which cluster should be presented at this optimum area of the visualisation. The recommender system under construction is a content-based system which will profile a users short term or session goals by observing both clusters and documents that are examined by a user. This system will also provide recommendations on which cluster a user should examine next when the semantic content of the current cluster is exhausted.

4.2 Visual Encoding to Support Maximum Gain

The user should be informed through visual encoding that the semantic gain for a particular patch has peaked. So far, resizing (decreasing size of) nodes representing documents with similar semantic content seems to be the most intuitive means of achieving this. Semantic content in this context extends beyond the concept of word-content. For example, a news item relating to the Middle-East might contain similar terms and achieve a high-similarity measurement using a model such as the Vector-Space model. However, the semantic content of a report on the event produced by Fox News might differ significantly from one produced by Al Jazeera.

An initial examination of the state-of-the-art suggests that two key approaches exist. One is centred on lexical analysis using onthologies such as WordNet, while the other approach employs statistical techniques such as Latent Semantic Indexing and Support Vector Machines to determine the semantic orientation of a document.

4.3 Assisting the Forager to Choose a Healthy Diet

Certain documents will yield better knowledge than others; for a given effort, a forager will derive greater nourishment from a summary or briefing document than would be derived from a set of documents dealing with individual issues. A

user should be guided towards these summary documents when they first begin to browse a patch. As mentioned in Section 3.1, the thresholding tool allows users to adjust the density of relevant documents within a cluster.

5 Scenario of Usage

A typical scenario of usage for the evolved SolonEvo application would be as follows. Having selected a collection to search, a user submits a query. A visualisation of the result-set will be created by applying both a clustering and force-directed algorithm. Based on one of several techniques, e.g., collaborative recommendation, the most suitable cluster is intelligently placed on the interface to reduce inter-cluster search-time.

If the clusters are excessively fine-grained or are too coarse, the user can adjust the density of the information patch using the threshold tool. The tool will repartition the minimum spanning tree derived from the similarity matrix formed by artefacts in the result-set (ss 3.1). Once satisfied with the density of the clusters, the user can focus their search using the proximity tool. As users examine artefacts, nodes that contain similar semantic information will adjust their appearance to signify decreased importance relative to other potentially relevant nodes. Once all unique semantic information has been examined, the rate of gain significantly decreases. At this point, the user will probably decide to move to another cluster that will have been recommended through some visual encoding.

6 Conclusions

This paper has described how certain aspects of the information foraging model can be used to improve the performance of Information Retrieval Interfaces. Based on Charnov's Marginal Gain Theorem, the authors contend that, by combining recommender systems and appropriate visualisations, the rate of information gain of information foragers can improve.

References

1. Benford, S., Snowdon, D., Greenhalgh, R., Knox, I., Brown, C.: VR-VIBE A Virtual Environment for Cooperative Information Retrieval. In: Computer Graphics Forum, 14(3) (1995)
2. Chang, D., Dooley, L., Tuovinen, J.E.: Gestalt Theory in Visual Screen Design A New Look at an Old Subject. In: Proceedings of Seventh World Conference on Computers in Education, Copenhagen (2002) 5-12
3. Davidson, I.: Visualising Clustering Results. In: Proceedings of the Second SIAM Conference on Data Mining, Arlington, (2002)
4. www.eyetools.com
5. Foltz, M.A., Davis, R.:Query by Attention: Visually Searchable Information Maps. In:Proceedings of the Fifth International Conference on Information Visualisation (IV'01), London (2001) 85-97

6. Freyne, J., Smyth, B.: Collaborative Result Clustering in Web Search. In:Proceedings of the 15th Artificial Intelligence and Cognitive Science Conference. Castlebar, Ireland (2004) 235-244
7. Fructerman, T., Reingold, E.: Graph Drawing by Force-directed Placement. In: Software - Practice & Experience 21(11) (1991) 1129-1164
8. Tollis, I., Di Battista, G., Eades, P., Tamassia, R.:Graph Drawing. Algorithm for Visualization of Graphs. Prentice Hall (1998) Chapter 10
9. Grimaldi, R.: Discrete and Combinatorial Mathematics: An Applied Introduction. 3rd Edn, Addison Wesley (2003)
10. Hoare, C., Sorensen, H.: A Proximity Based Browsing Tool for Information Foraging. In:Proceedings of the 15th Artificial Intelligence and Cognitive Science Conference. Castlebar, Ireland (2004) 67-76
11. Jain, A.K., Murty, M.N, Flynn, P.J: Data clustering: a review. In: ACM Computer Surveys, 31(3) (1999) 264-323
12. Jasen, B. J., Spink, A., Bateman, J., Saracevic, T.: Real Life Information Retrieval: A Study of User Queries on the Web. In: ACM SIGIR Forum, 32(1) (1998) 5-17
13. Kaki, M.: Proportional Search Interface Usability Measures. In: Proceedings of the Third Nordic Conference on Human-Computer Interaction. Tampere, Finland (2004) 365-372
14. Kaski, S., Honkela, T., Lagus, K., Kohonen, T.: WEBSOM - Self organising maps of document collections. In: Neurocomputing, vol. 21 (1998) 101-117
15. Katz, C. A., Byrne, M. D.: Effects of Scent and Breath on Use of Site Specific Search on E-Commerce Web Sites. In: ACM Transactions on Computer-Human Interaction 10(3) 2003 198-220
16. Konstan, J., Miller, B., Matz, D., Herlocker, J., Gordon, L., Riedl, J.: GroupLens: Applying Collaborative Filtering to Usenet News. In: Communications of the ACM, 40(3) 1997 77-87
17. Lamping, J., Rao, R., Pirolli, P.: A Focus+Context Technique Based on Hyperbolic Geometry for Visualising Large Hierarchies. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Denver, Colorado (1995) 401-408
18. Leuski, A., Allan, J.: Lighthouse: Showing the Way to Relevant Information. In: Proceedings of IEEE Symposium on Information Visualisation 2000 (2000)
19. Olston, C., Chi, E.: ScentTrails: Integrating Browsing and Searching the Web. In: ACM Transactions on Computer-Human Interaction (TOCHI), 10(4) (2003) 177-197
20. Pirolli, P., Card, S.: Information Foraging. In: Psychological Review, 106(4) (1999) 643-675
21. Pirolli, P., Card, S.: Information Foraging Models of Browsers for Very Large Document Spaces. In: Proceedings of the Working Conference on Advanced Visual Interfaces. L'Aquilla, Italy (1998) 83-93
22. Pirolli, P., Card, S., Van Der Wege, M.: The Effects of Information Scent on Visual Search in the Hyperbolic Tree Browser. In: ACM Transactions on Computer-Human Interaction, 10(1) (2003) 20-53
23. Salton, G.: The Smart document retrieval project. In: Proceedings of the 14th annual international ACM SIGIR conference on Research and Development in Information Retrieval, Chicago (1991) 356-358
24. Shneiderman, B.: Tree visualization with tree-maps: 2-d space-filling approach. In: ACM Transactions on Graphics, 11(1) (1998) 92-99
25. Spence, R.: Information Visualisation. ACM Press and Addison-Wesley (2001)

Agent Cooperation using Simple Fixed Bias Tags And Multiple Tags

Enda Howley and Colm O'Riordan

Department Of Information Technology
National University Of Ireland, Galway.
enda.howley@nuigalway.ie
colmor@it.nuigalway.ie

Abstract. Much research in multi-agent systems has focussed on the emergence of cooperation in societies where individually optimal behaviour for agents leads to low levels of cooperation in the society. This conflict between individual and collective rationality can be modelled through the use of social dilemmas such as the prisoner's dilemma. Tagging schemes have been shown to increase levels of cooperation through biasing interactions in a manner comparable to that of kin selection and trust mechanisms. We outline some simulations involving a simple tagging system and outline the main factors which are vital to increasing cooperation. This paper also outlines the effects of multiple tags.

1 Introduction

Agent interactions are often heavily biased by certain group structures. These structures may be based upon certain models of trust or kin selection. These groups reflect an approximation of which peers are most likely to remain altruistic. Some schemes are based solely on past behaviour and therefore reflect aspects of trust and relationships. Other models use physical proximity as a guide to bias interactions.

Significant research has been conducted involving agents whose interactions are determined by spatial proximity. Nowak and May(1993) describe the significance of group structuring techniques with special attention to spatially determined interactions[6]. This structuring can also be represented through the use of tagging which is an abstract method of biasing agent interactions based on membership of certain tag groups. Holland(1993) describes tags as markings or social cues that are attached to individuals (agents) and are observable by others[5]. When determining which agents should interact, we calculate the proximity of their tag values, as opposed to their physical proximity. Tagging is an abstraction which allows us reflect the relatedness between agents based on any possible grouping analogy, not simply spatial or genetic proximity. Tagging can be used as a general case to represent all these possible grouping structures without the specific complexities which they entail.

Hales(2004)[4] states that tags can evolve from initially random values into complex ever-changing patterns that serve to structure interactions between individuals. Tagging schemes are highly accurate mechanisms for biasing agent interactions based on their relatedness with each other.

This paper will analyse a simple tagging scheme and review some of the factors which contribute to its success. This paper will describe a series of experiments which we have designed involving a simple tagging scheme. For example, we will review the levels of cooperation achieved using different amounts of tags. We also investigate multiple tags which allow us to model agent societies, where agents can exist in a number of distinct groups.

Our primary motivation throughout this paper involves studying the effects of biasing agent interactions using various parameter space values. We extend previous research on tagging schemes to allow agents participate in multiple groups. This will be outlined through a series of simulations involving multiple tags. Throughout this paper all simulations involve populations of agents competing through the iterated Prisoner's Dilemma.

2 Related Research

The Prisoner's Dilemma (PD) is a simple two-player game where each player must make a decision to either cooperate (C) or defect (D). Both players decide simultaneously and therefore have no prior knowledge of what the other has decided. If both players cooperate they receive a specific payoff. If both defect they receive a lower payoff. If one cooperates and the other defects then the defector receives the maximum payoff and the cooperator receives the minimum. The payoff matrix outlined in Table 1 demonstrates the potential payoffs for each player.

Table 1. Payoff Matrix

Players Choice	Cooperate	Defect
Cooperate	$(\lambda 1, \lambda 1)$	$(\lambda 2, \lambda 3)$
Defect	$(\lambda 3, \lambda 2)$	$(\lambda 4, \lambda 4)$

The dilemma is a non-zero-sum, non-cooperative and simultaneous game. For the dilemma to hold in all cases, certain constraints must be adhered to: $\lambda 2 < \lambda 4 < \lambda 1 < \lambda 3$. These conditions result in $\lambda 2$ being the sucker's payoff, $\lambda 1$ is the reward for mutual cooperation, $\lambda 4$ is the punishment for mutual defection, and $\lambda 3$ provides the incentive or temptation to defect. The dilemma also states $2\lambda 1 > \lambda 2 + \lambda 3$. This constraint prevents players taking alternating turns receiving the sucker's payoff ($\lambda 2$) and the temptation to defect ($\lambda 3$), therefore maximising their score. The following values were used throughout this research: $\lambda 1 = 3, \lambda 2 = 0, \lambda 3 = 5, \lambda 4 = 1$.

In the non-iterated game, the rational choice is to defect, while in the finitely repeated game, it is rational to defect on the last move and by induction to defect all the time. However, if there exists a non-zero probability the two players will play again, then cooperation may emerge. Within a society of social groups, repeated meetings are common and as with tag group members who meet repeatedly, significant levels of cooperation can emerge.

2.1 Previous Tagging Models

Holland[5] initially outlined the concept of tags and since then significant numbers of tag models have emerged[8][4]. Holland describes tags as markings or social cues that are attached to individuals (agents) and are observable by others. Riolo has described a number of tagging approaches throughout a series of papers[8][9], each focusing on the effects of tagging on levels of cooperation among players in the Iterated Prisoner's Dilemma. These papers outline basic forms of tagging: fixed-bias tagging, variable-bias tagging and evolved-bias tagging.

These models comprise two-player Iterated Prisoner Dilemma (IPD) tournaments. Players are not paired randomly as in traditional tournaments such as those researched by Axelrod[2]. Instead, each player is given a numeric tag value and this is used to probabilistically bias interactions towards players of similar tag values. In these tagging models, significant increases in cooperation can be observed. Simple replicator dynamics are used to reflect the fitness of agents over successive generations. The proportional fitness of a genome is used to determine representation of that genome in subsequent generations.

We hope to gain a greater understanding of how tagging mechanisms successfully increase cooperation among agents playing the IPD. We address the following questions:

- What factors allow tagging schemes boost cooperation among agents?
- How does multiple tagging effect levels of cooperation?

In later sections, we outline a number of experiments and the results obtained. In the following section we will discuss the design of our simulator and tagging model.

3 Experimental Setup

3.1 Strategies

The design of any IPD simulator requires the simple creation of an initial set of player strategies. The research by Nowak et. al.[7] provides a basis for our method of strategy definition. The following is a strategy genome with 3 genes representing 3 possible behaviour values when P_i *(Probability of cooperation in the initial move of a game)*, P_c *(Probability of cooperation after opponent has cooperated)*, P_d *(Probability of cooperation after opponent has defected).*

$$Genome = P_i, P_c, P_d, \tag{1}$$

Our initial population of agent genomes is randomly generated with random gene values and therefore the initial population has an initial average fitness of about 2.25[1][8]. A fourth gene representing a tag value is also given to each genome, this tag value is randomly assigned to all strategies in the population and is a simple integer in the range [1...Maximum number of tags]. Multiple tagging involves extending this model to include a number of tag genes representing each of the tag groups an agent is a member.

3.2 Tag Model

Our simulator consists of a population of 100 players; each is randomly generated from a normally distributed set of possible strategies. These strategies are each given random tag values. Games lasting 20 iterations between all members of the same tag value are conducted. Agents of different tag values do not play each other. Agents can play themselves. If an agent is the solitary holder of a tag value it may still play itself, even though there are no other members of the population it can interact with.

The representation of a genome in successive generations is based on its fitness in the current generation. Therefore this dictates representation in the next generation and so on, through successive generations. This simple replicator dynamic is similar to that used by Riolo[8][9] in his simulations. In our research and the research of Riolo, all reproduction is asexual. As a result, no crossover between agents occurs.

This simulator extends previous tag models to allow some parameters vary over certain experiments. For example, this simulator can conduct experiments across a number of possible tag parameters.

4 Results

In this section we outline the results of a number of experiments. Our primary goal is to address the questions which we outlined earlier in Section 2.1 of this paper.

4.1 Performance of Tagging Scheme

The first set of experimental results represent a fitness comparison between two populations. One implements a simple tagging scheme while the other uses no tagging technique. Across 50 simulations, 100 agents are allowed to compete for survival while playing each other. The tagged model randomly distributes up

[1] If we use the following IPD payoff matrix values $\lambda 1 = 3, \lambda 2 = 0, \lambda 3 = 5, \lambda 4 = 1$, a initial population of random agents will have an average of these fitness values $(3 + 0 + 5 + 1)/4 = 2.25$

to 50 different tag values to the 100 agents. Throughout each simulation, these players will play IPDs of 20 iterations with peers of the same tag value. 50 generations are simulated and these tag groups are represented through successive generations based on their fitness through the use of a replicator dynamic. Our non-tagged model operates in exactly the same manner but all players play IPD's of 20 iterations with every other member of the population.

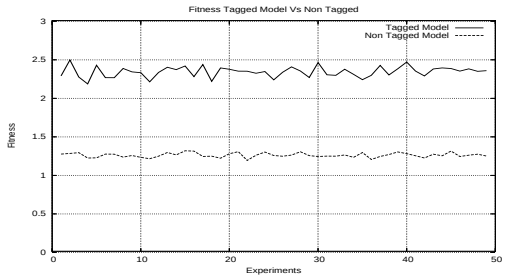


Fig. 1. Tagged Model Vs Non-Tagged Model

In Figure 1, we can clearly see the improved fitness achieved through the use of tagging. This data is comparable with previous research in the domain of tagging[8]. These results are consistent across all experiments while our simulation without tagging displays much lower levels of average fitness. The improved performance of our tagged model is as a result of tag groups insulating themselves against invasion from defecting strategies. In a non-tagged model, the effects of the defecting strategies propagates throughout the population and they dominate very quickly. This is less likely in a tagged model which partitions the population into tag groups through limiting their interactions. As a result groups containing defectors are impeded and loose representation to fitter competitive groups.

4.2 The Significance of Tag Group Size

To fully understand the factors determining the success of a tag model, we investigate the effect of varying the number of tags. The following experiment illustrates how the number of tags can influence levels of cooperation among agent populations. Four simulations represent populations using different numbers of permissible tags in their initial composition (1, 5, 20, 50) are shown in Figure 2.

In the experiment represented in Figure 2 we observe that as the number of tags increases, the partitioning effect increasingly limits the effects of defection. The model performs best when the number of tags is high. Therefore, in any one tag group, there exists a higher probability that all the members will be cooperative and as a result the group will be fitter.

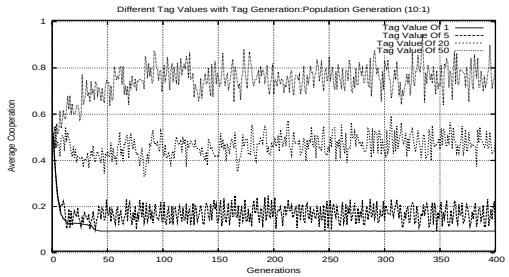


Fig. 2. Different Tag Ranges

4.3 The Evolution Of Tags

We have observed the importance of tag group size over time. The following experiment examines how the group dynamics are effected by the evolutionary process. In this simulation we record the percentage of possible tags in use at each generation. We also plot the average cooperation at each generation.

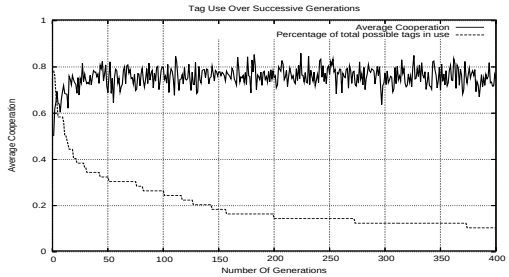


Fig. 3. Numbers Of Tags In Use

From this experiment we see that the number of total tag values falls significantly after the initial generations. The levels of cooperation remain very high. This is explained by the ability of tags to marginalise non-cooperative behaviour over the initial generations. The tagging system results in non-cooperative agents becoming extinct in the earlier generations while the most cooperative tag groups increase rapidly in size. These tag groups with the most altruistic members takeover the total population and lead to the extinction of tag groups which contain any non-cooperative agents. The resulting small number of tag groups all contain highly cooperative agents and are about equal in size. Usually this experiment resulted in about 5 of a possible 50 tag groups surviving, each with an average membership of 20.

4.4 The Effects of Multi-Tagging

In the following series of experiments, we have extended the simple tagging model through allowing each agent participate in multiple tag groups. This reflects real life interactions where individuals often participate in many social groups. Each agent may hold membership of up to five tag groups simultaneously. This is modelled through displaying each of these tag values. An agent cannot interact with peers which do not contain at least one common tag value. Therefore, they must each hold membership of at least one common tag group.

In the following experiment we outline the effects of varying the number of tags a population of agents may simultaneously hold. The actual tag values are specified at random in the range of 1 to 50 among a population of 100 agents. We ran 5 simulations allowing agent membership of different numbers of groups ranging from 1 to 5.

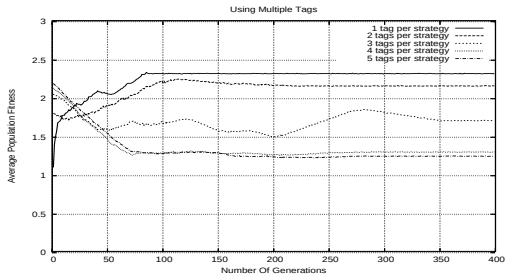


Fig. 4. Number Of Tags Per Strategy

From this experiment we observe the decreased levels of fitness among populations which permit agent membership of multiple tag groups. This is explained through the increased interactions which multiple group membership facilitates. As explained earlier altruism benefits from partitioning small groups of cooperative strategies away from defective peers. Multiple tags undermine this system and result in lower levels of fitness throughout the population.

A less contrived experiment specifies a population with a mixed degree of membership among many tag groups. For example each agent will have a high probability of holding one random tag membership, a smaller probability of holding two and so on. We model this as follows. The probability Y is calculated with respect to the number of tag groups X and some negative power N.

$$Y = x^{-N} \tag{2}$$

The following experiment shows the effects of varying the possible number of tags in use throughout the population while applying the above function with a N value of 2.2. It has been shown in many studies of various social structures and social networks, that connections and interrelationships between individuals follow this type of distribution[3]. Albert et. al.[1] show for a range of examples

including web topology, citation patterns and food webs, the degree of connection can be described by a power law. This value significantly biases agents towards containing less tags as can be seen from the following probabilities.

$$P_1, P_2, P_3, P_4, P_5 = 1.0, 0.21, 0.08, 0.04, 0.02 \tag{3}$$

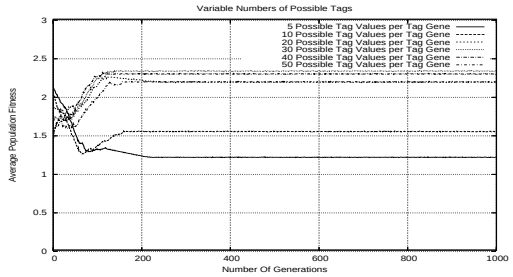


Fig. 5. Varying the number of possible tags in the population

The function specified limits the number of multiple tag groups agents may participate within. Subsequent fitness depends on the number of tag values permitted within the population. But these levels of fitness are dependant upon reducing the number of tag groups agents participate in. While this is reflected through results outlined earlier in this paper it is augmented through examining our final test case experiment which follows. We can observe the opposite effect of greater use of multiple tags through changing the values used in our formula. In the following case we can observe the direct effects of increased multiple tag use on the population.

In the following experiment we replicate its predecessor through simulating a new value of $N = 0.1$. The resulting function represents a greater probability agents will hold membership of multiple tag groups. Here we examine the negligible effect of varying the number of tags within a population which is biased towards holding membership of multiple tag groups. We observe this bias through the following probabilities:

$$P_1, P_2, P_3, P_4, P_5 = 1.0, 0.93, 0.89, 0.87, 0.85 \tag{4}$$

This experiment confirms that multiple tags undermine cooperation and as a result diminish the fitness of the agent population. Multiple tags counteract the benefits of using many tag values to boost cooperation in a population. The conflicting nature of the two parameters is confirmed through reviewing the dominant strategies which proceed to win our final simulations. In simulations depicting the emergence of cooperation, dominant strategies were predominantly single tag holders. Alternatively in non-cooperative populations multiple tag holders prevailed and displayed dominance.

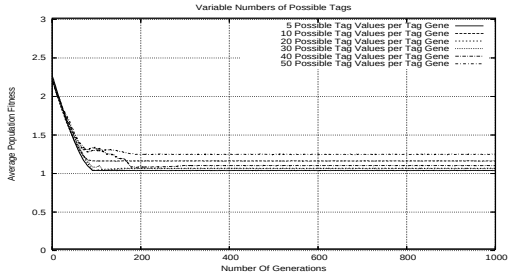


Fig. 6. Varying the number of possible tags in the population

5 Conclusions

Our experiments have highlighted the primary factors leading to the success of certain tagging models. We have observed the importance of the number of tags used in a population. This has a direct effect on the overall levels of cooperation. The success of tagging is based upon preventing invasion from greedy agents. This partitioning effect which is synonymous with tagging and spatial models is fundamentally important to their success. A tag group of only one agent never encounters the "invasion" difficulties which usually jeopardise cooperation among larger groups of agents. In successful tagging models we have observed a small set of tag groups emerging to dominate a population after a number of generations. This small number of tag groups composed of highly cooperative strategies experience rapid growth because of their fitness. Tag groups containing non-cooperative agents experience high attrition rates.

The effects of multiple tagging resulted in undermining the overall fitness of our population. This feature reflected that agents which spanned multiple tag groups were at a disadvantage and more susceptible to exploitation. Reinforcing our previous evidence that less interactions among agents improved cooperation, multiple tags increased such interactions and as a result decreased levels of cooperation.

In answer to the two questions posed earlier (Section 2.1) of this paper we conclude that the primary factors contributing to the success of tagging schemes all involve limiting the number of agent interactions to a minimum. This finding is based on results across all our experiments and is further reinforced by previous research in the domain.

Future work involves more elaborate tagging models. Various aspects of learning, evolution and communication of tags are also possible through extensions of current tag models.

Acknowledgment

The primary author would like to acknowledge the Irish Research Council for Science, Engineering and Technology(IRCSET) for their assistance through the Embark initiative.

References

1. R. Albert, A. Barabsi, and H. Jeong. Scale-free characteristics of random networks: The topology of the world wide web, 2000.
2. R. Axelrod. *The Evolution of Cooperation.* Basic Books, New York, 1984.
3. G. Csyani and B. Szendroi. Stability by mutation in evolutionary games. *Physical Review*, E 69, 2004.
4. D. Hales. Understanding tag systems by comparing tag models. In *Presented at the 2nd Model-to-Model Workshop M2M2 co-located with ESSA04*, 2004.
5. J. Holland. The effects of labels (tags) on social interactions. *Working Paper Santa Fe Institute 93-10-064*, 1993.
6. M. Nowak and R. May. The spatial dilemmas of evolution. *Int Journal of Bifurcation and Chaos*, 3:35–78, 1993.
7. M. Nowak and K. Sigmund. The evolution of stochastic strategies in the prisoner's dilemma. *Acta Applicandae Mathematicae*, 20:247–265, 1990.
8. R. Riolo. The effects and evolution of tag-mediated selection of partners in populations playing the iterated prisoner's dilemma. In *ICGA*, pages 378–385, 1997.
9. R. Riolo. The effects of tag-mediated selection of partners in evolving populations playing the iterated prisoners dilemma. *Nature 414*, pages 441–443, 2000.

Decision Tree Learning based approach for Traffic Classification in High Speed Networks

Parag Kulkarni, Nazeeruddin Mohammed, Sally McClean, Michaela Black, Gerard Parr, Bryan Scotney

School of Computing and Information Engineering
Faculty of Engineering, University of Ulster, Coleraine
Cromore Road, Coleraine – BT52 1SA
Co. Londonderry, Northern Ireland
{pg.kulkarni}@ulster.ac.uk

Abstract. Traffic management and congestion control are the rudimentary blocks of any service provider network that desires to support user differentiation through different service classes. The first step in providing such varied Quality of Service (QoS) to different classes of users is the ability to efficiently classify traffic into one of the different service classes. Such an approach can help service providers to offer better services to their clients. In this paper, we present a decision tree learning based approach for traffic classification in ATM networks and compare its performance with the existing approaches. The argument is clearly in favour of deploying decision tree learning based approach for traffic classification because they are computationally less intensive, intuitively appealing, require much less training time and can classify traffic more efficiently than the existing approaches employing neural networks. Results obtained validate these claims.

1 Introduction

The internet was not envisaged to be a multi-service network. However voice, video and data all share the same infrastructure today. The Quality of Service (QoS) requirements of each of these traffic types is different. These requirements generally take the form of a certain degree of packet loss, end to end delay, delay variation etc. In order that the traffic sent by the end user gets the treatment that it has requested, a mechanism to differentiate this traffic from other traffic is required. Thus, the need of the day is a fast and efficient traffic classification mechanism.

As is evident from existing work ([1], [2], [3], [4], [5], [6]), data mining has proved to be a viable alternative in the context of this problem and can provide fast, adaptive and intelligent control. The benefits accrued out of addressing network control problems using data mining have been realised long since the early nineties. Most of the initial effort was directed towards solving the fault management problem. However, to date, several broad classes of problems in network

design and control have been addressed using data mining techniques. Sekercioglu et al in [7] have surveyed the various Computational Intelligence strategies used in the management of ATM networks – typically the use of fuzzy logic, neural networks (NN) and genetic algorithms in routing, fault management, call admission control (CAC), parameter control, congestion and rate control. A neural network based connection admission controller for ATM networks has been demonstrated in [8] and [9]. Solutions to several traffic control problems such as routing, service coding, traffic policing, traffic prediction and fault detection using neural networks have been proposed in [5], [2], [4], [3] and [6]. A neural network based method for estimating the QoS in ATM networks is presented in [1]. [10] has improved upon the results (improved classification accuracy and reduced training time) of [1] by introducing a novel divide and conquer approach for QoS estimation. Table. 1 highlights some of the data mining tasks and techniques that have been used to address problems in networking. A thorough survey of the various data mining tasks, techniques and applications can be found in [11]. Note that there are a raft of data mining techniques each with their own distinct capabilities. In the context of this work however, *classification* has been the main focus.

Table 1. Data Mining techniques used in Network Management so far

Data Mining Technique	*Network Management Problem to which it was applied*	*Data Mining Context in which the technique was applied*
Bayesian Belief Networks	Alarm Correlation, Fault Diagnosis	Feature Selection, Model Building
Neural Networks	Alarm Correlation, Fault Diagnosis, Admission Control, Congestion Avoidance, *Traffic Classification*	Model Building
Expert Systems	Fault Management	Building knowledge base
Time Series Analysis	Fault Prediction	Model Building
Hypothesis Testing	Fault Diagnosis	Feature Selection, Model Building
Reinforcement Learning	Routing, Admission Control	Model Building
Fuzzy Logic	Admission Control, Congestion Control	Model Building
Discriminant Functions	Fault Classification	Model Building

2 Framework for QoS Estimation

Congestion control is one of the most important aspects of a high speed network. If heavy traffic (more than what the link is capable of carrying) is destined for a link, the waiting time (Cell[1] Delay) and hence the Cell Delay Variation at the output queue will increase. In addition to this, the queue may discard cells (Cell Loss) if it overflows. Increased Cell Delay Variation (CDV) and Cell Loss Ratio (CLR) are undesirable as they severely impact not only the network performance but also the user perception of the underlying service. Since QoS in ATM networks is specified through these two parameters, some means of traffic management is necessary to regulate these two parameters so that QoS requirements of the end users can be satisfied. Table 2 shows the various QoS Classes as defined by the ATM Forum. It should be noted that depending on the requirements of the service provider, there could exist multiple levels of service within a service class. The values of CDV and CLR define a service class and the levels within a service class. Needless to mention, the lower the values of CDV and CLR, the better the class of service. The experiments described in this paper will make use of the seven classes shown in Table 3 as outlined in [10].

Table 2. QoS Classes Specified by the ATM Forum

QoS Class	*Service class*	*Applications Supported*
1	Circuit Emulation, Constant Bit Rate (CBR) Video	Digital Private Line
2	Variable Bit Rate (VBR) Audio and Video	Packet Video and Audio in Teleconferencing, Multimedia Applications
3	Connection oriented data transfer	Support interoperation of connection oriented protocols (e.g. Frame Relay)
4	Connection-less data transfer	Support interoperation of connectionless protocols (e.g. IP)

2.1 Problem Formulation

Fig. 1 shows the QoS estimation framework. As shown in Fig. 1, the CDV and CLR values of the incoming traffic are fed to the data mining algorithm. The data mining algorithm, based on the values of CDV and CLR, predicts the QoS class

[1] Note that the notion of a 'Cell' is similar to that of a 'Packet' in an IP network.

Table 3. Seven Levels of QoS

QoS Level	*Meaning of the QoS Level*
1	Excellent
2	Degrading from Excellent to Good
3	Good
4	Degrading from Good to Fair
5	Fair
6	Degrading from Fair to Bad
7	Bad

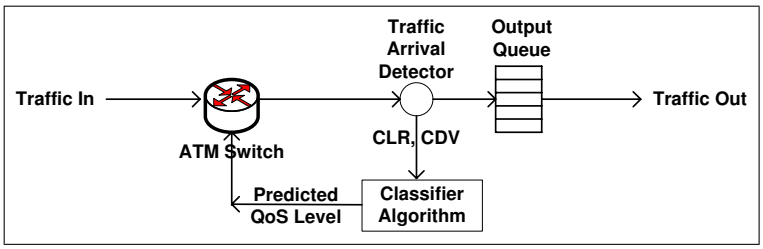


Fig. 1. Framework for QoS Estimation

of the incoming traffic. This prediction is passed on to the ATM switch which then deals with this traffic accordingly. Supervised Inductive learning (which uses pre-classified examples to train the algorithm) is employed. A common inference task consists of making predictions about a concept (also known as a class). The prediction problem under consideration is known as 'classification'. The concept to be learned in this case, is the level of QoS which has well defined values (one of seven classes) as described in Table 3. The aim is to find the most appropriate algorithm that can accurately find relationships among the attributes which distinguishes values of the concept. The key performance quantifiers for judging the performance of such an algorithm are classification accuracy, complexity and training time.

3 Experimentation and Results

3.1 Training and Test Data

Usually training data is decided by the service provider based on customer service quality requirements [1]. Training data is comprised of pre-classified examples (data that contains input and target values that are used to train a model). Training data is normalized into the range of [0 1] using simple normalization equation $(I - I_{Min})/(I_{Max} - I_{Min})$. I_{Max} and I_{Min} are maximum and minimum values of input data (I). Once the model is generated, its performance is evaluated by passing a test set through it. A test set is a set of examples which

the model has not seen before. To provide a ground for comparing the various approaches, the same training and test patterns were used as those described in [1] [10] (for more details see [1] and [10]). For the sake of validation, two additional test sets were generated using MATLAB ([12]). The equations used to generate these test sets were $CLR = asin(CDV)/b$ and $CLR = acos(CDV)/b$ for different values of a and b. CDV values were generated randomly. Test data was also normalized into the range of [0 1].

3.2 Choice of classification techniques

As mentioned in section 2.1, the aim is to find a classifier that delivers well on accuracy without trading off complexity and training time. The three main approaches described in this paper are – Linear classifier (based on the Mahalanobis distance algorithm), decision trees and neural networks. These are some of the best known multivariate techniques that deal with separating distinct sets of attributes and allocating new attributes to groups defined apriori. Note that the attributes in this case are CDV and CLR. Each of these approaches are summarised below.

- *Linear classifiers* try to differentiate between the various classes by obtaining a linear combination of the attributes. In the context of this work, the approach based on Mahalanobis distance (implemented by the classify function in Matlab [12]) was chosen. This is the simplest of all the approaches considered in this paper and is computationally least intensive.
- *Neural networks* are classifiers capable of finding non-linear combinations of attributes. Most of the existing knowledge based approaches to traffic classification (section 1) employ neural networks. In the context of this work, neural network based approaches described in [1] and [10], have been used as a baseline for comparison. For the sake of discussion, we will refer to the approach described in [10] as approach-DaC (divide and conquer). The basic idea behind approach-DaC involves dividing the m input, n output neural network model into n neural network blocks with m inputs and 1 output. Where n is the number of QoS levels to be estimated (seven in this case). For a given input vector, if the p_{th} block's output is one then that input vector belongs to QoS level p otherwise that input vector belongs to another QoS level. Since any given input vector can belong to only one QoS level, for any given input vector the output of one of the n blocks will be 1 and that of the other $n - 1$ blocks will be 0. In addition to their own approach (approach-DaC), the authors of [10] have also provided results of their experimentation with approaches based on Multi-layer Perceptron (MLP) Networks and Radial Basis Function Networks. We refer the interested reader to [10] for details.
- *The Decision tree* is a powerful classifier that identifies the attributes that are important in connection with an outcome in a prediction task (concept / class) and how each attribute is associated with the different possible outcomes. The decision tree is constructed through a process called induction

which requires a small number of passes through the training data. Decision tree induction generally involves growing the tree and pruning it (e.g. C4.5 algorithm). The tree grows from root to the leaves i.e. top to bottom through iterations. The instances at the root node comprise the whole training set. During every iteration, the training set is partitioned into smaller subsets. The algorithm, for each level of the tree, tries to find an independent attribute (CDV and CLR in this case) which when used as a splitting node will result in rules that are very different from each other with respect to the concept / class (QoS Class in this case). There are several ways to measure this difference - entropy, information gain etc. Once a node is split, the same process is repeated for each of its child nodes, with the dataset further subdivided based on the split criteria at the parent node. This process continues until a node is reached where a further split isn't possible either because there are no more variables to split or there is no information gain in doing so. The rules to stop building the tree at a point also depends on factors like the maximum depth of the tree. Once the tree is built, it is necessary to prune it to avoid over-fitting the data. This process, called 'post-rule pruning', is in-built in most of the tree building algorithms. Before deploying this tree, it is necessary to test its accuracy and generality by passing a test set through it and observing the outcome. Once ready for deployment, this tree can be used for predicting a new case by starting from the root node down to one of the leaf nodes. The path taken through the tree is governed by the rules (associated with each non-leaf node) applied to the values of the independent variables in the new case. For details we refer the interested reader to [13], [14] and [15].

3.3 Data Analysis - Results and Summary

Table 4 quotes results of experiments with neural networks described in [10] whereas Tables 5, 6 and 7 highlight the results from the experiments that were carried out as a part of this work.

Table 4. Results of QoS estimation using Neural Networks from [10]

	% Correct	*%Wrong*
RBFN	87.23	12.77
MLP	88.39	11.61
Divide and Conquer	92.73	7.27

Following are some of the key observations:

– The linear approach is not a universal approximater and may not perform well in terms of accuracy when the attributes from the various classes have fuzzy boundaries, as is the case here. This is reflected in the results that we

Table 5. Results of QoS estimation using Linear Classifier

Test Set	*Samples*	*% Correct*	*% Wrong*
1	876	64.612	35.388
2	752	54.566	31.164
3	727	70.20	29.80

Table 6. Results of QoS estimation using C4.5 algorithm

Test Set	*Total Cases in Test Set*	*No. of Cross Validation Folds*	*% Correct*	*% Wrong*
1	876	5	93.84	6.16
1	876	10	93.84	6.16
1	876	15	93.84	6.16
2	752	5	96.54	3.46
2	752	10	96.54	3.46
2	752	15	96.54	3.46
3	727	5	94.22	5.78
3	727	10	94.22	5.78
3	727	15	94.22	5.78

Table 7. Results of QoS estimation using CaRT algorithm

Test Set	*Total Cases in Test Set*	*No. of levels below root in the Tree*	*% Correct*	*% Wrong*
1	876	9	85.39	14.61
1	876	10	92.92	7.08
1	876	11	94.18	5.82
1	876	12	94.18	5.82
2	752	9	94.41	5.59
2	752	10	96.28	3.72
2	752	11	96.28	3.72
2	752	12	96.28	3.72
3	727	9	71.94	28.06
3	727	10	90.51	9.49
3	727	11	92.3	7.7
3	727	12	92.3	7.7

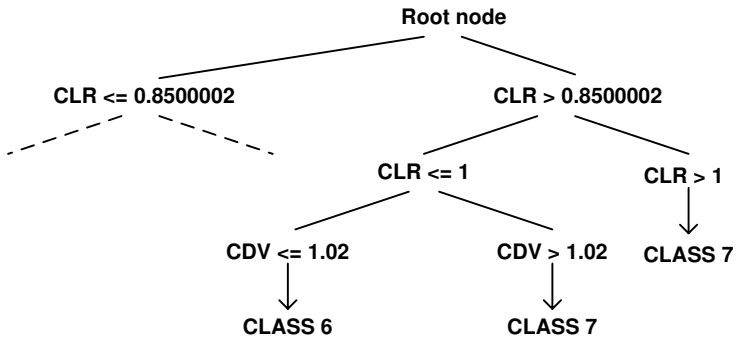


Fig. 2. Subtree of a tree built by the C4.5 algorithm

obtained which are shown in Table. 5. Observe that the linear classifier is able to achieve a best case accuracy of only 70.20% compared to that of the other approaches.

- The best results obtained by [10] have been listed in Table. 4. Observe from Table. 4 that most of these approaches perform considerably well in terms of classification accuracy. Even though neural networks are well known for picking up non-linearities from the underlying system and delivering on accuracy, they suffer from several problems, the key ones being - complexity (in terms of network structure as in [1]), scalability, black box approach and a significant training time.
- Recall from section 3.1 that same data as described in [10] was used for our experimentation in order to provide a baseline for comparison of results. Tables. 6 and 7 show results using two most popular decision tree learning algorithms - C4.5 [16] and Classification and Regression Trees (CaRT [15]) available as a part of the software package Clementine [17]. Fig. 2 shows a subtree of an example tree generated by the C4.5 algorithm. Table. 6 shows the prediction accuracy for different cross validation folds and Table. 7 shows the effect of increasing the number of levels in a tree below the root. As shown in Table. 6, the prediction accuracy is similar for different cross validation folds. An interesting observation here is that even for a 5-fold cross validation, the C4.5 algorithm outperforms all the other approaches (linear classifier and neural network based approaches). As for the CaRT algorithm, as the number of levels below the root is increased the prediction accuracy improves. However the accuracy reaches a steady value and does not improve when the number of levels below the root is increased beyond 11. On the whole, C4.5 and CaRT algorithms outperform all the existing approaches.
- During the tree building process, the computational complexity to find the best splits among p-real valued variables typically scales as O(p*n*log(n)) [15]. The process of classification on the other hand involves tree traversal

which in the worst case is known to take O(log(n)) steps for a tree size of depth n.

In summary, decision trees are known to be computationally efficient (hence require less training time than neural networks) and are intuitively appealing due to their tree structure (simpler to understand as opposed to the black box approach of neural networks). Moreover in the context of this work, decision trees have outperformed neural networks while predicting the traffic class to which the incoming traffic belongs. The aforementioned arguments clearly advocate the use of decision tree learning for traffic classification in high speed networks.

4 Conclusion

In this paper we demonstrated that the decision tree learning based approach outperforms existing approaches for traffic classification in high speed networks. The main advantages are an intuitively simple model in the form of a tree of *if-then-else* rules, less computational complexity, less training time and also an increased predictive accuracy.

References

1. W. Sheng, J. Rueda, and D. Blight. Neural networks based atm qos estimation. In *Conference on Communications, Power and Computing WESCANEX-97, Winipeg*, pages 1–6, May 1997.
2. N. Kotze and C. Pauw. The use of neural networks in atm. In *Communications and Signal Processing (COMSIG)*, pages 115–119, July 1997.
3. A. Sarajedini and P. Chau. Quality of service prediction using neural networks. In *Military Communications Conference MILCOM-96*, pages 567–576, 1996.
4. E. Nordstrom, J. Carlstrom, O. Gallmo, and L. Asplund. Neural networks for adaptive traffic control in atm networks. *IEEE Communications Magazine*, pages 43–49, October 1995.
5. A. Tarraf and I. Habib. A novel neural network traffic enforcement mechanism for atm networks. *IEEE Journal on Selected Areas in Communications*, 12(6):1088–1096, 1994.
6. A. Aussem, A. Mahul, and R. Marie. Queueing network modelling with distributed neural networks for service quality estimation in b-isdn networks. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks 5, Como, Italy*, pages 392–397, July 2000.
7. Y.A.Sekercioglu, A.Pitsillides, and A.Vasilakos. Computational intelligence in management of atm networks. *Soft Computing Journal*, 5(4):257–263, August 2001.
8. C-K. Tham and W-S. Soh. Atm connection admission control using modular neural networks. In *Proceedings of the IEEE INFOCOM*, pages 1022–1029, 1998.
9. S. Leng, K. Subramanian, N. Sundararajan, and P. Saratchandran. Novel neural network approach to call admission control in high-speed networks. *International Journal of Neural Systems*, 13(4):251–262, 2003.
10. M. Nazeeruddin, M. Mohandes, and H. Cam. Atm qos prediction using neural-networks. In *Proceedings of the 6th International Conference on Neural Information Processing (IEEE-ICONIP)*, pages 532–537, 1999.

11. S.McClean. Data mining and knowledge discovery. *Encyclopaedia of Physcial Science and Technology, Third Edition, Kluwer Publishers*, 4, 2002.
12. The Mathworks Inc. Matlab. http://www.mathworks.com/, May 2005.
13. StatSoft Inc. Electronic textbook. http://www.statsoftinc.com/textbook/stathome.html, September 2004.
14. T. Mitchell. *Machine Learning.* MacGraw Hill, 1997.
15. D. Hand, H. Mannila, and P. Smyth. *Principles of Data Mining.* Prentice Hall of India Pvt. Ltd., 2001.
16. J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.
17. SPSS Inc. Clementine. http://spss.com/clementine/index.htm, September 2004.

Emergence of Cooperative Societies in Structured Multi-Agent Systems

Marguerite Moran and Colm O'Riordan

National University Of Ireland, Galway.
m.moran4@nuigalway.ie
colmor@it.nuigalway.ie

Abstract. In this paper we investigate the emergence of cooperation in spatially organised games. We extend traditional spatial models and use a graph to model the environment. In the graph representation of the environment, each node represents an individual, and an edge between two individuals represents a neighbourhood relationship.
In our model, players interact in a Prisoner's Dilemma. We examine various learning mechanisms where the agent's strategies are selected and propagated. We investigate the effect of allowing agents learn from their neighbours to improve their individual performance.
We also explore the evolution of neighbourhoods by enabling them to grow or shrink depending on their relative fitness to other neighbourhoods.

1 Introduction

Many approaches have been investigated in an attempt to understand how cooperation may emerge in societies of autonomous, rational agents. The Prisoner's Dilemma has been adopted as the standard for studying cooperative behaviour [1], [2], [12], [11]; and has also been used in work focussing on spatially organised games [6], [9], [3], [13].

Classical game theory [8] doesn't include the effect of spatial structures on a population. In many populations, both real and artificial, individuals are more likely to interact with their neighbours than interact with a player chosen at random from the population. In order to model such scenarios more realistically, it is necessary to spatially organise agents in a multi-agent system, and for interactions to take place according to these spatial constraints. Furthermore, it is unlikely that neighbourhoods will be of uniform size throughout the population. We adopt a graph model to represent the neighbourhoods.

While our model encompasses many features and extensions of traditional models, our primary focus is on the effect of defining different social structures and neighbourhoods based on our graph representation of the environment.

In the experiments discussed in this paper, we investigate the emergence of cooperation or defection in a spatially structured society. We compare the spread of cooperation in a graph based model to traditional grid representations. We

investigate a range of neighbourhood sizes and learning mechanisms. Furthermore, we investigate the spread of cooperation in a spatially structured society where the neighbourhood structure itself is allowed to evolve and change, i.e., fit neighbourhoods are allowed to increase in size. Much research in traditional models deals with patterns of cooperators and defectors [9] — clusters of cooperators and defectors situated on a grid with cooperator-defector interactions along the edges of the clusters. They don't assess the overall fitness of each neighbourhood. Our model allows fit neighbourhoods to propagate and conversely, if a neighbourhood has a low fitness value it is reduced in size.

2 Related Research

2.1 The Prisoner's Dilemma

The Prisoner's Dilemma is described as a non-cooperative, non-zero-sum, two person simultaneous game. In the prisoner's dilemma, two players are separated and faced with a decision. Each has two alternatives — to cooperate or defect. Neither has knowledge of the other player's choice. If they both cooperate, they receive a payoff, R. If both defect, they receive a smaller payoff, P. If one defects and the other cooperates, the defecting strategy receives the largest possible payoff, T, and the cooperator the smallest possible payoff, S.

For a dilemma to exist, the following must hold: $T > R > P > S$. The constraint $T + S \leq 2R$ must also hold. The constraint $T + S \leq 2R$ prevents a form of cooperation where two players obtain an average payoff greater than cooperation by alternating between cooperation and defection.

2.2 Spatial Prisoner's Dilemma

Nowak and May [9] model a simple, deterministic, spatial version of the Prisoner's Dilemma, with no memories among players and no strategic elaboration; showing that cooperators and defectors and co-exist indefinitely for a subset of the parameter space.

Many researchers have investigated the effect of varying payoffs [13]. Some have suggested that if agents are indistinguishable from each other, genuine cooperation cannot emerge (Frank [4], Maynard-Smith [8], Hofbauer [7], Weibull [18], Samuelson [15]). Evidence in social science shows that cooperation can emerge even if creatures cannot recognise individual players [17], [11]. Epstein[3] shows that cooperation can emerge and endure if negative payoffs are introduced. He requires that $T > R > 0 > P > S$, and his results depend largely on the values of T, R, P and S.

Cooperators in the population will benefit those close to them and flourish if the bonus for defection is not too large [5], [11]. Hence, groups of cooperators situated adjacent to each other can allow cooperation to flourish through mutual cooperation. Conversely, for defectors, they hurt their own kind and hence groups of defectors will not do well.

Nowak et al. [10] also experimented with the grid size and topology and also the neighbourhood type. They extended their analysis to explore simulations where players are randomly distributed on a two-dimensional plane. Players are said to be neighbours of each other if their distance is less than a certain radius of interaction r; so the number of neighbours each player has can vary. The simulations were deterministic and in discrete time. Results showed that populations on random grid were more static than on rectangular lattices and that cooperation can emerge and endure.

Ashlock [16] focuses on a static population and the effectiveness of a choice / refusal strategy undertaken by some agents in the population. He implements a graph based model, where each node represent an individual, and an edge connecting two vertices represent some relationship between the two individuals. Initially, the graph is fully connected and each edge is assigned a weight of zero. When two individuals play, the weight is incremented by 1. These weights are plotted in order to show population characteristics. Analysis is performed on a significant play graph. Edges are considered significant if they are greater than a given threshold value; and these edges determine the significant play graph. The significant play graph adds valuable information about the social behaviours of a system.

3 Simulator Design and Description

Our simulator allows us to examine the emergence (or not) of cooperation for a wide range of parameters. We compare different topologies, neighbourhood types and update strategies for varying radius values.

For a finite number of iterations (unknown to the player), each agent plays a PD game against each of its neighbours. Each player is either a pure cooperator or a pure defector. At the end of each game, depending on the payoffs received by each agent and the update strategy in place, the agent adopts the strategy of one of its neighbours. Following the completion of each game, and the application of the update rules, the number of cooperators is recorded.

3.1 Parameter Range Modelled

The parameter space within which agents play the PD game is very important. In the following section we discuss the parameters that are modelled and studied in our system. We discuss topology, initial configuration of the population, neighbourhood type and learning mechanisms.

Topology. A common example of a topology used in simulations is that of a $N \times M$ rectangular lattice. An $N \times M$ grid is a planar graph with $N \times M$ vertices arranged on a rectangular grid, and with edges connecting horizontally and vertically adjacent vertices.

We compare two types of topology. In our model, the population is placed randomly on either an $N \times M$ lattice or on a graph type structure. The $N \times M$

lattice is standard for traditional spatial models. This planar grid type structure is not an accurate or flexible representation of real social systems so we have also simulated a more general graph based environment. Our graph is made up of nodes — one node denotes one member of the population — and depending on the connectivity, edges which connect nodes at random.

A general unidirectional graph is a better representation of real systems. Individuals in our simulated environment are connected to other individuals via an edge. Rather than representing distance or adjacency, this edge can be a representation of a different type of relationship. An individual's neighbourhood can be built depending on some social relationship taken into account. Kinship, similarity, mutual interest are typical criteria used when establishing the social components of a community [14].

Initial configuration of cooperators and defectors. The configuration of the population can also be defined. This includes the ratio of cooperators to defectors and where they are located in their environment. They can be placed either randomly, or a specific configuration can be imposed.

Neighbourhood type. The neighbourhood refers to the set of agents with which a player interacts. There exist two traditional neighbourhood types: Von Neumann and Moore. In the Von Neumann neighbourhood, each player interacts with its four nearest neighbours to the north, south, east and west. In the Moore neighbourhood each individual interacts with the eight neighbours reachable by a chess-king's move - i.e. one square in any direction.

Our model simulates these neighbourhoods types as well as a graph type network. In the graph network, individuals are connected to a set of individuals, chosen at random. The size of this neighbourhood depends on the connectivity imposed at the outset of the experiment. Our simulator also allows us to experiment with different radius values in order to determine what effect this has on a system. As the radius, and thus the neighbourhood size increases, we can determine the resulting impact on cooperation.

Player Update. The model enables us to compare many different update rules, some of which are deterministic, some stochastic. Depending on its own score and the scores of its neighbours, each player may update its strategy. Different update rules exist: e.g. Best Takes Over, Imitate the Better and Proportional. Our model is capable of simulating each of these. If strategies are updated using Best Takes Over, a player adopts the strategy of its most successful neighbour. Imitate the Better involves a player comparing its own score with that of it neighbours. If the difference is positive, the player imitates the neighbour with a probability proportional to the difference. The proportional update is where a player adopts the strategy of one of their neighbours with a probability proportional to their scores. Other learning models can also be employed. Update rules can be either deterministic or stochastic. In the stochastic game, update rules will be executed with some degree of probability.

We implement stochastic update rules to discourage swift update convergence which can sometimes lead to a sub-optimal outcome. Deterministic local player update means that after the first round, players are more likely to face opponents playing the same strategy as themselves. This can tend to a bias in favour of cooperators if defectors end up playing defectors. This can quickly drive to extinction strategies that would normally survive and eventually dominate under more stochastic update rules.

Neighbourhood Update. As well as individual learning mechanisms described in the previous section, our model allows us to update and evolve neighbourhoods. The evolution is executed at discrete time steps and in a stocahstic nature to prevent swift update convergence which could lead to a sub-optimal outcome. Depending on a neighbourhood's fitness relative to the average neighbourhood fitness, a new agent can be propagated or indeed an agent can die off. Fitter neighbourhoods are allowed to grow while less fit neighbourhoods decrease in size.

4 Results

In our experiments, we aim to compare agent behaviour in grid and graph topologies. We then extend our graph experiments and examine behaviour as a result of a neighbourhood's ability to grow or shrink relative to neighbourhood fitness.

Initially, we varied the environment size and ran simulations with 100, 400 and 900 agents. Results for the varying sizes showed similar graph patterns emerging for mean values. The results in the rest of this section are based on simulations run for 900 agents.

4.1 Grid topologies

In our experiments, we vary the radius size (R = 1, 2, 3). We bias the initial configuration such that 90% of agents are cooperators. We explore the behaviour of the agents given different update strategies (Best Takes Over and Imitate the Better), across different neighbourhood types (Von Neumann and Moore). The results depicted in Figures 1, 2, and 3 show us that for the Von Neumann neighbourhood with $R = 1$, cooperation can coexist with defectors for all update strategies. This is due to the small neighbourhood size defined and results are consistent with traditional models.

For the Moore neighbourhood type with $R = 1$, levels of cooperation can be maintained for deterministic update rules. However, the stochastic update mechanism results in the emergence of defection. For both deterministic and stochastic updates, the *imitate the better* update strategy showed slightly better results in terms of cooperation for Moore and Von Neumann neighborhoods with a radius of 1.

Due to the increase in neighbourhood size, for all neighbourhood types with a radius value of $R > 1$, defection becomes the norm, with one exception —

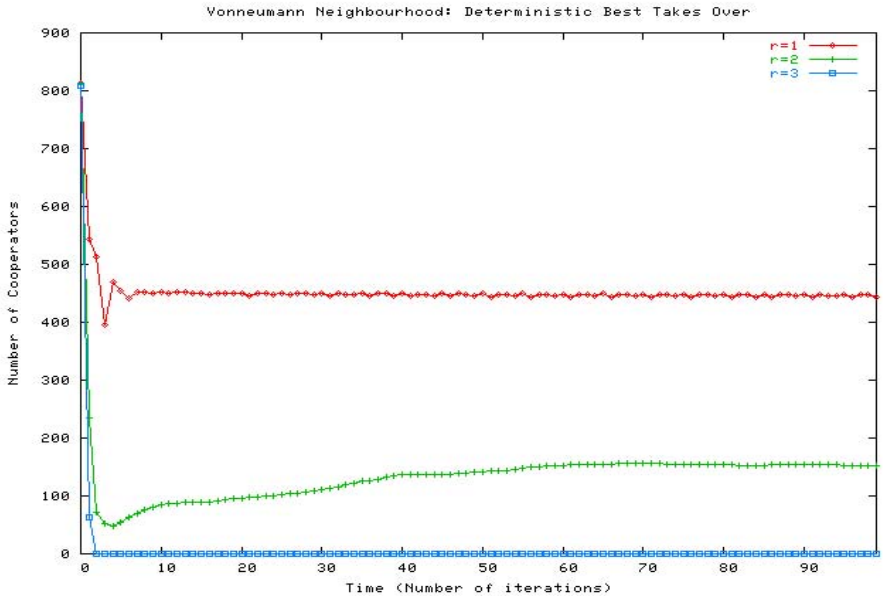


Fig. 1. Von Neumann: Deterministic Best Takes Over

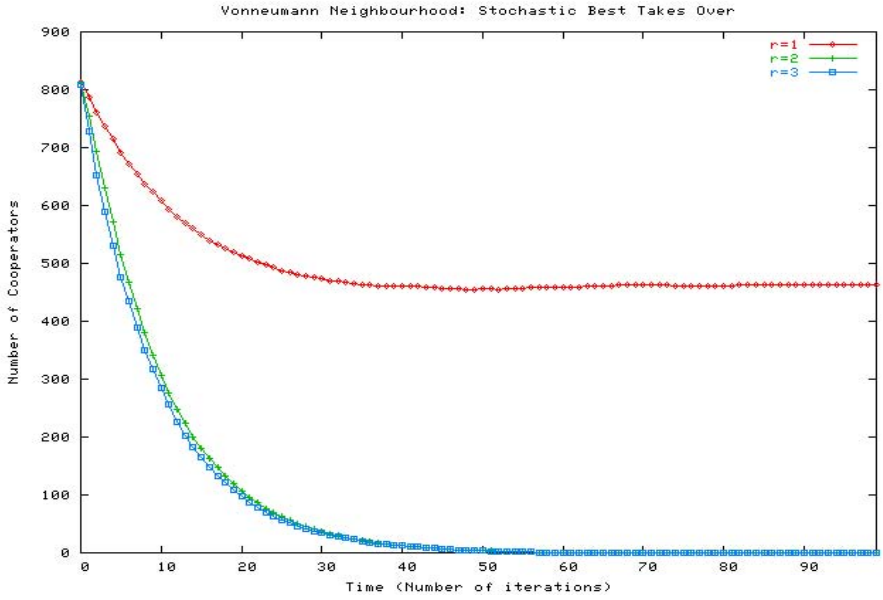


Fig. 2. Von Neumann: Stochastic Best Takes Over

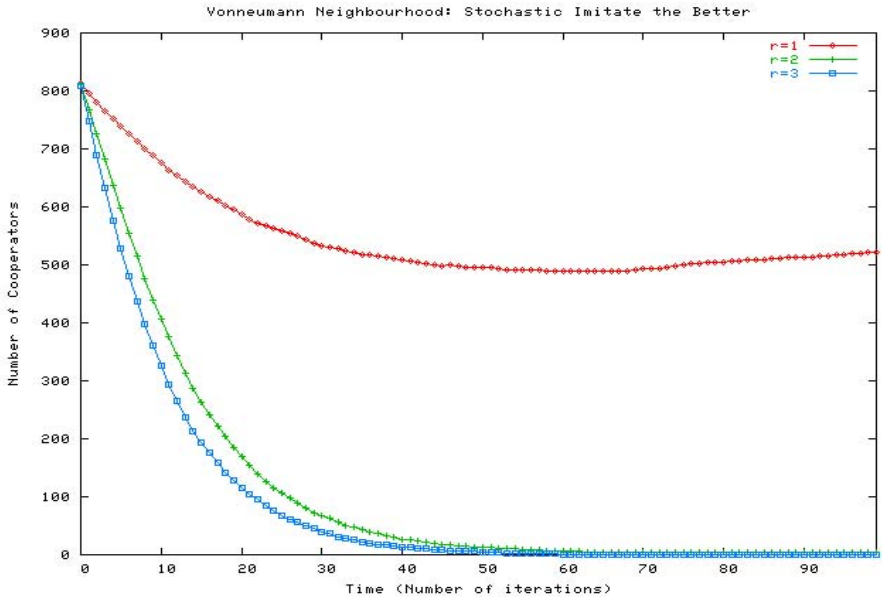


Fig. 3. Von Neumann: Stochastic Imitate the Better

at $R = 2$, the Von Neumann neighbourhood, with a deterministic Best Takes Over update mechanism, a small degree (10% - 15%) of cooperators survive. As expected, the rate at which cooperation decreases is proportional to the increase in R.

4.2 Graph topology

We also explore the behaviour of agents given different update strategies, using the graph topology and neighbourhoods. For deterministic updates widespread defection is almost immediate. For some runs of the experiment, a small number of cooperators coexist with defectors. However, this could be a result of a small neighbourhood of cooperators being disconnected from the rest of the graph.

Using stochastic update rules has little effect on our simulated graph environment (Figures 4 and 5). Convergence is similar but occurs over a longer period of time.

Defection spreads more quickly through our graph topology for a number of reasons. Firstly, depending on the seed, the graph neighbourhood can be much larger than that of the Moore or Von Neumann, and since defection is proportional to neighbourhood size this seems reasonable. Also, our graph topology represents a social system where behaviour can spread more quickly than on traditional lattice structures.

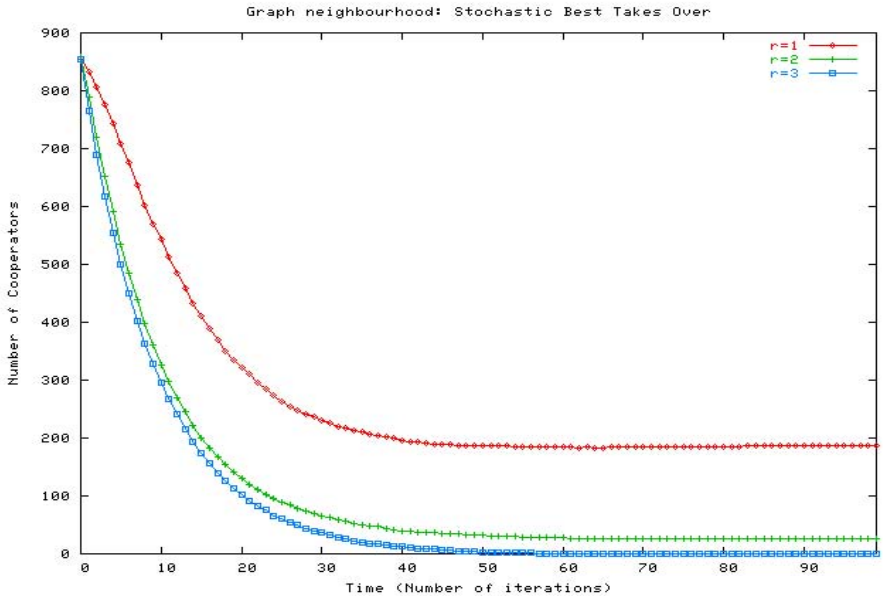


Fig. 4. Graph: Stochastic Best Takes Over

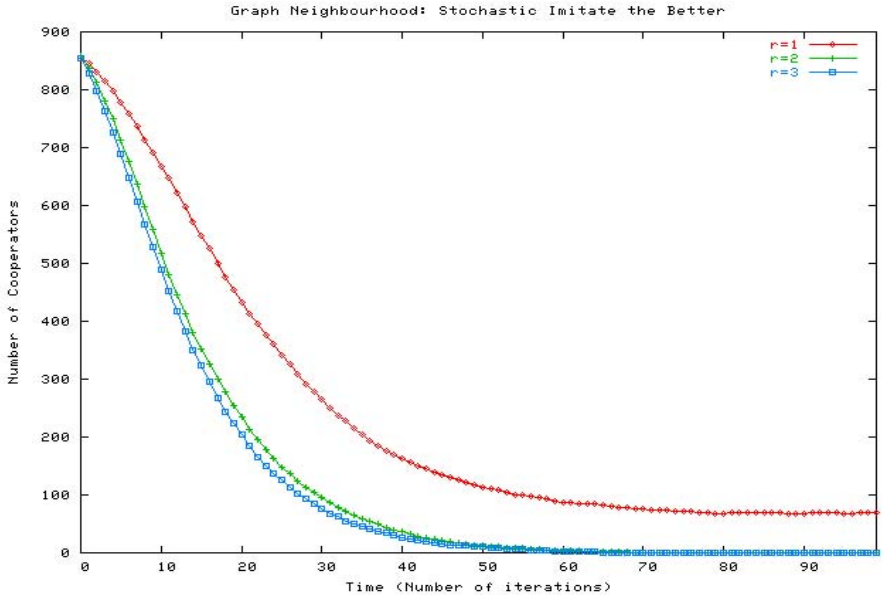


Fig. 5. Graph: Stochastic Imitate the Better

4.3 Graph neighbourhood propagation

In the previous experiments, we updated strategies by allowing agents learn from their neighbours, i.e. strategies which are fit in comparison to the neighbours are propagated. We paid no attention to the fitness of neighbourhoods relative to other neighbouhoods in the population. In this experiment, we allow fit neighbourhoods to grow. A neighbourhood which is fitter than the average neighbourhood is allowed to grow in proportional to its fitness above the average. Similarly, less fit neighbourhoods decrease in size. We investigated the effect of neighbourhood propagation on our graph topology. Figure 6 shows the outcome for one of our experiments. Here we initiated our population to 900 with 95% cooperators and experiments were run for 1000 generations. The resulting graph shows that, initially, defecting strategies quickly flourish and propagate. However, as the simulation continues and neighbourhoods evolve by propagating or shrinking depending on their fitness, we can see the decline of defecting strategies and emergence of a cooperative environment.

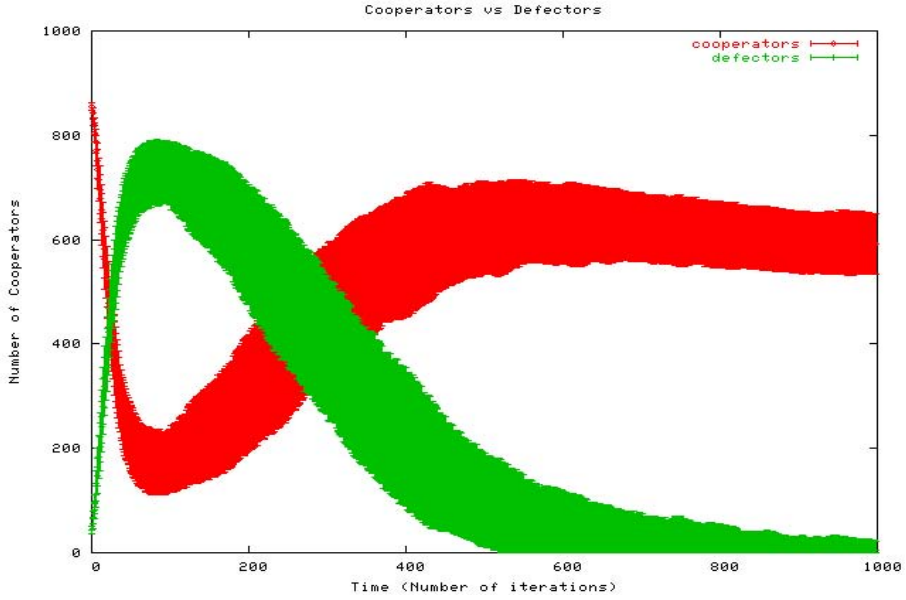


Fig. 6. Graph: Stochastic Imitate the Better including Propagation

5 Conclusion & Future Work

In this paper we explored the parameter space, with particular emphasis on the environment topology. This paper describes a graph topology within which

agents can participate in a Prisoner's Dilemma game. A comparison of the graph topology and traditional topologies for varying parameter ranges shows that behaviour spreads more quickly through the graph type environment. We see that in most cases, defection spreads. However, by modifying the graph structure and allowing neighbourhoods to grow or shrink depending on relative fitness, cooperation dominates.

Future work will involve exploring a larger strategy set and fuller investigation into the evolution of the social structure (i.e. the graph topology).

References

1. Hamilton W.D. Axelrod, R. The evolution of cooperation. *Science*, 211(4489):1390–1396, Mar. 1981.
2. R Axelrod. *The Evolution of Cooperation.* Basic Books, New York, USA, 1984.
3. J. M. Epstein. Zones of cooperation in demographic prisoner's dilemma. Technical Report 97-12-094e, Santa Fe Institute, December 1997. available at http://ideas.repec.org/p/wop/safire/97-12-094e.html.
4. R. H. Frank et. al. *Principles of Micro Economics.* McGraw-Hill, 1st edition, 1997.
5. Ch. Hauert. Fundamental clusters in 2 x 2 games. In *Proceedings of the Royal Society B: Biological Sciences*, volume 268, pages 761–769, 2001.
6. Ch. Hauert. Effects of space in 2 x 2 games. *International Journal of Bifurcation and Chaos*, 12:1531–1548, 2002.
7. J. Hofbauer and K. Sigmund. *Evolutionary Games and Population Dynamics.* Cambridge University Press, 1998.
8. J Maynard Smith. *Evolution and the Theory of Games.* Camridge University Press, Cambridge, England, 1982.
9. M. A. Nowak and R. M. May. Evolutionary games and spatial chaos. *Nature*, 359:826–829, 1992.
10. M. A. Nowak, R. M. May, and S. Bonhoeffer. More spatial games. *International Journal of Bifurcation and Chaos*, 4(1):33–56, 1994.
11. May R.M. Sigmund K. Nowak, M.A. The arithmetics of mutual help. *Acientific American*, 272, 1995.
12. Sigmund K. Nowak, M. Chaos and the evolution of cooperation. In *Proc. Natl. Acad. Sci. USE*, volume 90, pages 5091–5094, June 1993.
13. M. Oliphant. Evolving cooperation in the non-iterated prisoner's dilemma: The importance of spatial organisation. In Maes P. Brooks, R., editor, *Proceedings of the fourth artificial life workshop*, pages 349–352, 1994.
14. J. M. Pujol, R. Sang, and J. Delgado. Extracting reputation in multi agent systems by means of social network topology. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 467–474, New York, NY, USA, 2002. ACM Press.
15. L. Samuelson. *Evolutionary Games and Equilibrium Selection.* MIT Press, Cambridge, MA, March 1997.
16. M. D. Smucker, Stanley, E. Ann, and D. Ashlock. Analyzing social network structures in the iterated prisoner's dilemma with choice and refusal. Technical Report CS-TR-94-1259, University of Wisconsin - Madison, Dept. of Computer Science, December 1994.
17. R. L. Trivers. The evolution of reciprocal altruism. *Quarterly Review of Biology*, 46(1):35–57, 1972.
18. J. Weibull. *Evolutionary Game Theory.* MIT Press, Cambridge, MA, 1995.

A HOTAIR Scalability Model

A. Mur, L. Peng, R. Collier, D. Lillis, F. Toolan, J. Dunnion

Department of Computer Science,
University College Dublin (UCD), Belfield, Dublin 4, Ireland.
{ mur.angel, liu.peng, rem.collier, david.lillis, fergus.toolan,
john.dunnion}@ucd.ie

Abstract. This paper describes a scalable mathematical model for dynamically calculating the number of agents to optimally handle the current load within the Highly Organised Team of Agents for Information Retrieval (HOTAIR) architecture.

1 Introduction

Indexing the World Wide Web is a complex task that requires a software infrastructure that has the ability to crawl through millions of web pages, extracting their content, and storing representations of that content in a form that is most appropriate for retrieval. Traditionally, research in this area has focused upon the development of information retrieval techniques that improve (1) the location and extraction of content, and (2) the representation of that content in forms that engender higher levels of precision and recall.

While this area of research remains one of the key research areas in Computer Science, it is becoming increasingly acknowledged that the design of the architecture in which these techniques are embedded is equally important. In fact, a recent news article on the success of Google made the point that "many people consider the company's operations expertise more valuable than the actual search algorithms that launched the enterprise" [15]. This is reflected in the fact that Google have been able to develop a robust and reliable distributed search architecture [1] that has cost millions of dollars, rather than the tens of millions of dollars that it has cost other competitors.

The design of robust and reliable search engine architectures that can scale effectively over large numbers of machines is a significant engineering problem. This paper presents one approach to solving this problem through the use of intelligent agents [32]. Specifically, it introduces the HOTAIR Search Engine architecture, an *extensible* and *scalable* architecture for the discovery, retrieval and indexing of documents from multiple heterogeneous information sources.

Within the HOTAIR architecture, extensibility is engendered through the design of an architecture that provides support for: (1) the plugging in of multiple indexing strategies such as the Vector Space Model [25] and the Extended Boolean Model [26]; and (2) the ability to rapidly and seamlessly integrate diverse sources of information. This requires the use of an open infrastructure that is able to dynamically

adapt its configuration to seamlessly integrate new techniques and information sources into the system.

Conversely, scalability is engendered through the design of an architecture that can be easily expanded as requirements increase. Typically, this will take the form of increasing the number of machines on which the architecture is deployed. Underlying this is the assumption that adding more machines will deliver an improvement in performance of the system. However, achieving this improvement is often a non-trivial task for a system administrator. It often requires detailed knowledge of both the expected load that will be placed on the system, and the most appropriate configuration for handling that load. Their task is further complicated by the fact that the actual load on such a system will change over time as the number of searches increases and decreases. This can result in the application undergoing significant periods of non-optimal performance. Thus, supporting scalability requires a solution that is flexible, aware of the current level of demand, and which can dynamically adapt its configuration to reflect both changes in demand and the availability of resources.

The key characteristic that the implemented architecture must conform to is the ability to dynamically adapt its configuration as requirements, demand, and resources change. These characteristics are synonymous with the types of system that agent technologies are most suited to. This has led to the development of the HOTAIR Document Indexing System, a multi-agent system that has been designed to adapt its configuration in response to changes in demand, and which supports the seamless integration of new techniques and information sources.

Scalability is an important, yet under-researched, aspect of agent platforms. The dynamics of multi-agent systems are hard to predict and the number of agents in large-scale distributed applications can vary considerably over time.

This paper aims to implement a mathematical model that can be used to estimate the number of agents required based on the available resources.

2 Related Work

The first Internet search engines began to appear in the mid-1990s. One of the first was the World Wide Web Worm (WWWW) [18]. Since their emergence, the main focus of research in this area has been on the development of better information retrieval techniques. Perhaps the most successful of these has been PageRank, the information retrieval technique that underpins the Google search engine [5].

Traditionally, the implementation of search engines, such as Google, was based on cluster-based architectures, with large numbers of low-cost servers located at one or a few locations and connected by high-speed LANs [4]. Their robustness and reliability is commonly achieved through the replication of services across many different machines, and the implementation of an infrastructure that automatically detects and handles failures [1].

Some researchers, such as ODISSEA [29], have explored the potential of Peer-to-Peer technology in the design of a next generation of distributed search architectures. Others have focused on the concept of meta-search [17][27], focusing on the definition of strategies for combining results from numerous search engines.

Another approach to Internet Search is through the use of software agents [13] [14] to perform tasks such as discovering, indexing and filtering documents and routing relevant information to users. By far the most prominent agent-based approach is through the use of single agent systems, which act as assistants that do all the tasks by themselves. For example, POIROT [24] is a web search agent based on relevance, LETIZIA [16] is an agent that assists Web browsing, and CITESEER [2] is an autonomous citation index finding relevant research publications on the WWW.

In contrast, multi-agent systems are decentralized and distribute tasks among a number of agents. ACQUIRE [9], is an example of a mobile agent-based search engine for retrieving data from heterogeneous, distributed data sources. In contrast, AMALTEA [19] is a search tool that discovers and filters information using a multi-agent evolving ecosystem.

Multi-agent systems are highly dynamic. The number of agents can scale up or down to ensure optimal performance [20] [3]. Scalability is also a term that is often used to refer to extensible functionality. SAIRE [21] is a scalable agent-based information retrieval engine because it supports heterogeneous agents.

The problem of scalability and some scaling techniques are described in [31]. An overview of multi - agent system scalability and a labor market application to model scalability can be found in [28].

3 The HOTAIR Indexing System

The HOTAIR Document Indexing System has been implemented using Agent Factory [8], a cohesive framework that delivers structured support for the development and deployment of multi-agent systems, which are comprised of agents that are autonomous, situated, social, intentional, rational, and mobile [7].

A diagrammatical overview of the agents that make up the system architecture is presented in figure 1. The actual number of agents that exist at any time varies depending upon the demand on and the resources available to the system. In addition, these agents are deployed over a number of different agent platforms that reside on different physical machines.

The creation of agents is a service that is provided by the Platform Manager (PM) system agent. Each agent platform contains a PM, which is responsible for handling requests to create more agents. Upon receipt of a request, a PM negotiates with its counterparts to decide on which machine(s) the requested agent(s) should be created. If there are insufficient resources to create all of the requested agent(s), then the PM agents can either refuse or partially fulfil the request.

3.1 The HOTAIR Agents

Within the HOTAIR architecture, the *Data Gatherer* (DG) agents are charged with the task of analyzing information sources. In the current version of the architecture, two types of DG have been implemented: the Collection DGs are used to process documents stored within static Document Collections, while the Web DGs are, in essence, web spiders that are crawling the World Wide Web. All DG agents follow a common behaviour, they search their assigned information source, discovering new

documents and downloading them into a temporary cache. Internally assigned document identifiers are added to an internal queue and the Broker agent is informed of the existence of new documents.

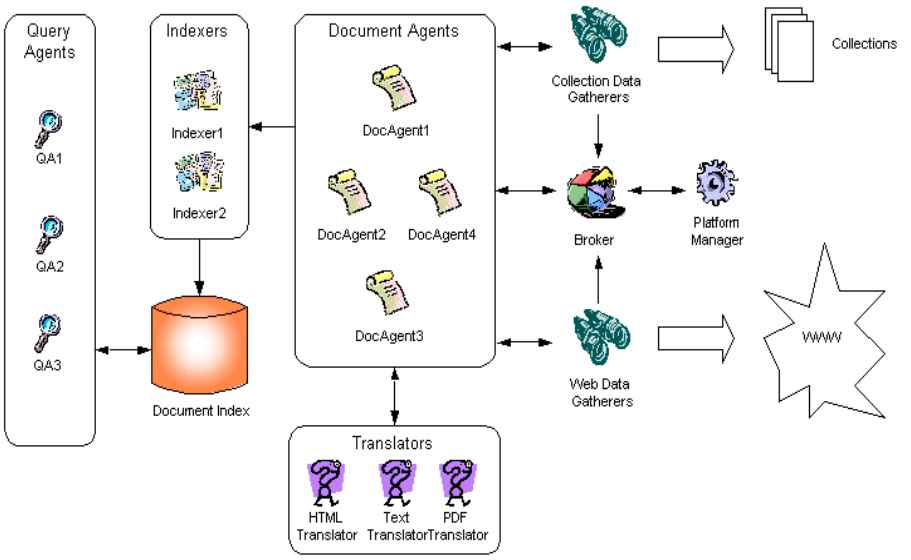


Fig. 1. The HOTAIR Document Indexing System

The *Broker* agent is responsible for monitoring the status of the DG's. This status is currently represented as the size of each DG's document queue. Periodically, the Broker requests a status update from each DG. Whenever a DG's status changes, the Broker reviews how many Document Agents (DAs) to assign to it. If the Broker decides that there are not currently enough DAs, then it asks the local AMS agent to create more DAs. As discussed earlier, this request may be refused. Thus, in cases where the Broker has an insufficient number of DAs, the Broker assigns DAs to DGs that are most in need of additional DAs. When significant disparities exist, the Broker re-assigns some existing DAs to different DGs.

Document Agents (DAs) encapsulate the workflow of the system, that is, they know how to get a document indexed. Currently, indexing a document involves: (1) getting a document from the DG, (2) getting the document translated by a Translator agent, and (3) getting the document indexed by an Indexer agent. Once assigned to a DG, each DA follows the prescribed workflow until either the DG has no more documents or the Broker re-assigns it to another DG. Once an assignment finishes, the Broker either re-assigns the DA or instructs it to terminate itself.

The *Translator* agents are responsible for translating documents from their native format into an internal format, known as the Hotair Document Format (HDF), that is understood by the Indexers. Each Translator specializes in translating one type of document. Currently supported formats include: Portable Document Format (PDF), HTML, Postscript (PS), and plain text. Should demand for a translation service become excessive, a Translator is able to use the Agent Factory cloning capability to clone itself [30]. Excessiveness is currently measured by demand passing a prescribed

threshold. Once created, the load is spread between the original and the clone. The clone is terminated if demand falls below a second lower threshold.

The *Indexer* agent is responsible for indexing documents. Eventually, HOTAIR will support numerous indexing strategies, however, currently it supports only the Vector Space Model. As with the Translators, Indexers are able to clone themselves should demand pass a given threshold.

The final set of agents is the *Query* agents. These agents query the document index on behalf of the user. They provide an agent-oriented interface to the HOTAIR system. In future versions of the architecture, these agents will perform a number of additional activities, including query expansion and user modelling.

4 HOTAIR Scalability Model

The HOTAIR architecture specifies three key points of adaptation: (1) through the cloning of Indexer Agents, (2) through the cloning of Translator Agents, and (3) through the creation of Document Agents.

Document Agent Scalability impacts the speed at which documents are indexed [22]. For a collection of documents, there will be a specific number of Document Agents, for which the document indexing speed is optimal. These agents process this collection more efficiently than other number of agents.

The Broker agent decides the optimal number of Document Agents to process a collection of documents. It uses a formula that represents a Scalable Document Agent Model.

4.1 Scalability model using Multiple Linear Regression

There are two main features of a document collection or group of documents : their size (total number of occurrences i.e. total number of words with repetition) N_o and the number of documents N_d. For each collection, it is possible to explore manually which is the optimal number of agents N_{da} that performs best in terms of time.

The objective of the experiment presented below is to find an equation that allows us to calculate automatically the optimal number of agents from any group of documents.

A solution can be found using a Multiple Linear Regression (MLR) [23]. MLR is a method used to model the linear relationship between a dependent variable and one or more independent variables or predictor variables. MLR is based on least squares: the model is fit such that the sum of squares of differences of observed and predicted is minimized. A general model expresses the value of a variable Y as a linear function of one or more variable X_i and an error term: $Y = b_0 + b_1 X_1 + \ldots + b_k X_k + \varepsilon$ where $\varepsilon \sim N(0,\sigma^2)$. b_0 is a regression constant, b_i is the coefficient on the i^{th} predictor variable X_i, k is the number of variables and ε is the error term.

If we have n experiences, the model $Y_j = b_{0j} + b_{1j} X_{1j} + \ldots + b_{kj} X_{kj} + \varepsilon_j$, (j=1…n), can be compactly written using a matrix notation **Y=XB+E** where

Y= $(Y_1, Y_2, \ldots Y_n)$ **; B=** $(b_1, b_2, \ldots b_k)$ **;**

$$\mathbf{E}=(\varepsilon_1,\varepsilon_2,.....\varepsilon_n)\ ;\ \mathbf{X}=\begin{pmatrix} 1 & X_{11} & . & . & X_{k1} \\ 1 & X_{12} & . & . & X_{k2} \\ . & . & . & . & . \\ . & . & . & . & . \\ 1 & X_{1n} & . & . & X_{kn} \end{pmatrix}\ ; \qquad \textbf{(1)}$$

B values can be estimated using the equation $\mathbf{B_s= (X'X)^{-1}X'Y.}$

4.2 Scalable Document Agent Model

We validate the MLR scalability model using Document Agents. Our response would be the optimal number of Document agents N_{da} and our predictor variables N_d and N_o : N_{da} = f (N_d, N_o).The optimal number, N_{da}, represents the number of a group of Document Agents that process documents quicker than other groups for every combination (N_d , N_o). The best group was chosen analysing time processing $t_{(Nd\ ,\ No)}$ of 50 agent groups from 1 agent to 50 agents. Every $t_{(Nd\ ,\ No)}$ was calculated several times and N_{da} is the integer nearest the mean of the results.

The results were obtained using 3 document collections, each of which contained 1000 documents. The first two collections are subsets of the Cranfield and Med collections, while the 3rd collection is comprised of single word documents (i.e. one word per document). This collection is unusual but necessary to get a general model valid for any kind of document.

The table 1 shows a selection from n=30 observations of N_{da} from different combinations (N_d , N_o). The n observations have been selected independently of one another.

N_d	N_o	N_{da}
…….	…….	……
1	26	1
50	227	7
100	6411	15
600	39404	24
1000	65454	32
1000	1000	18
……..	……..	…….

Table 1: Table of the different combination (N_d , $N_{o,}$ N_{da}).

The model found is : $N_{da} = b_0 + b_1 \times N_d + b_2 \times \ln(N_d) + b_3 \times N_d \times N_o$ where b_0 = 1.6563; b_1 = -0.0097; b_2 = 3.5143; b_3 = 2.3364e-007. The Figure 2 shows a 3D representation of the model.

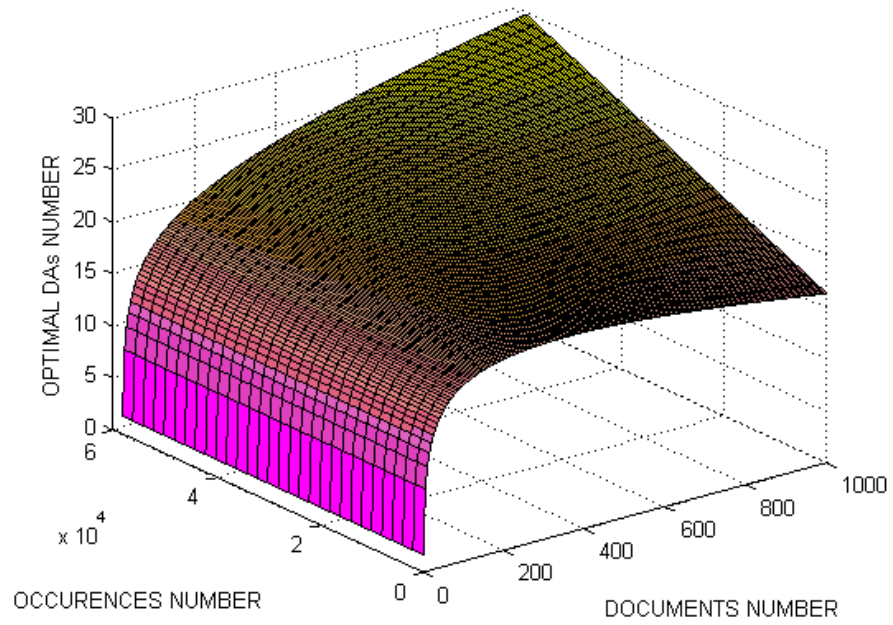


Fig. 2: 3D representation between the Optimal DAs number and the occurrences and documents number.

Table 2 shows the parameters used to validate the model. A Fisher test [23] is used to explain the model utility and a coefficient of correlation R is used to calculate the explanatory power of the regression (2).

R	**F – ratio**	**Significant**
0.9671	F = 110.8083	1%

Table. 2: Validation of the model

$$\mathbf{R} = \frac{\left\| \hat{N}_{da} - \overline{N}_{da} \right\|}{\left\| N_{da} - \overline{N}_{da} \right\|} \; ; \; \mathbf{F} = \frac{(n-k-1)\,\Sigma\left(\hat{N}_{da} - \overline{N}_{da}\right)^2}{(k)\,\Sigma\left(N_{da} - \hat{N}_{da}\right)^2} \; ; \qquad (2)$$

$\hat{N}_{da}$: fitted values, $\overline{N}_{da}$: mean of the $\mathbf{N_{da}}$ observations, N_{da} : observations

The value of R compared with the value 1 suggests that the chosen model has been very successful in relating $\mathbf{N_{da}}$ to the predictors. The Fisher test (F-ratio) shows that we have a significant model and it means there is a useful linear relationship between $\mathbf{N_{da}}$ and at least one of the predictors.

A Student test [23] (t-ratio $T_i = b_i$ / SD(b_i) where SD is the standard deviation of b_i) was used to determinate if all the coefficients of the predictor variables are useful.

Term	b1	b2	b3
Significant	*	**	**

Table. 3: ** means very significant , * significant.

The Student test shows that all our coefficients are useful. Consequently, we have a significant model with the minimum number of predictors.

4.3 Discussion

The model found represents a general view of how the Document Agent community scales as the number of documents to be process changes.

A document for any quantity of occurrences needs one DA always. From a collection of 5 documents the number of agents begins to change in relation to the number of occurrences.

The 3D graph shown in figure 2 shows that the optimal DAs number increases when the documents and occurrences number increases. Document number is more important than the occurrence number. But both are significant from the Student test.

In practice the optimal number of agents is an integer, consequently the estimated float value from the model is rounded to the nearest integer.

This model for documents uses the number of occurrences N_o like a predictor. Due to the high correlation between the N_o and the number of bits of each document N_b, other similar model could be found using N_d and N_b instead of N_d and N_o. This new model should be better for indexing web pages et/or documents dynamically. N_d and N_b can be obtained before processing the documents.

5 Conclusions / Future Work

This paper presents a Document Agent Scalability model for the HOTAIR architecture. This architecture is able to dynamically reconfigure itself to reflect changes in demand through either the creation of additional DAs or through the cloning of Indexer or Translator agents.

The model allows us to study HOTAIR Scalability and automatically gives the optimal number of DAs for any collection of documents. The performance of the HOTAIR architecture improves from a priori knowledge of the optimal number of agents using a model. The Broker agent assigns to the system the optimal number of Document Agents to process a collection of documents.

It is our intention to use the same procedure to build a Scalable Model for other types of HOTAIR agents, namely: Indexer Agents and Data Gatherer Agents.

While these experiments are based on a simplified HOTAIR Document Indexing System, we believe that the results are still valid. In particular, it would seem sensible to assume that, once the optimal number of DAs has been reached for a given indexer, then performance can only be improved by adding another indexer.

References

1. Barroso, L.A, Dean, J., and Holzle, U., *Web Search for a Planet: The Google Cluster Architecture*, in IEEE Micro, 23(2):22-28, 2003.
2. Bollacker k. ; Lawarence, S; Giles C.L.: *Citeseer: An Autonomous Web Agent for Automatic Retrieval and Identification of Interesting Publications*, Agents'98, 2nd International ACM Conference on Autonomous Agents, 116. 1998.
3. Brazier, F., Van Steen, W.: *On MAS Scalability*. Second International Workshop on Infrastructure for Agents, MAS and Scalable MAS. 2001.
4. Brewer, E.: *Lessons from giant scale services*, IEEE Internet Computing, pages 46-55, August, 2001.
5. Brin, S., and Page, L., The Anatomy of a Large-Scale Hypertextual Web Search Engine, Computer Networks and ISDN Systems, 30(1-7): 107-117, 1998.
6. Chen, L; Sycara, K.: *WebMate : Personal Agent for Browsing and Searching*, Proceedings of the Second International Conference on Autonomous Agents, St. Paul, MN, May, 132-139. ACM Press, New York, NY, 1998.
7. Collier, R., Agent Factory: A Framework for the Engineering of Agent-Oriented Applications, PhD Thesis, Dept. Computer Science, University College Dublin, 2001.
8. Collier, R., O'Hare, G. M. P. Lowen, T. D., and Rooney, C. F. B., *Beyond Prototyping in the Factory of Agents*, In Proc. 3rd Int. Central and Eastern European Conference on Multi-Agent Systems (CEEMAS), Prague, Czech Republic, 2003.
9. Das, S., Shuster, K., and Wu, C.: *Agent-Based Complex Query and Information Retrieval Engine*. AAMAS'02. July 15-9,Bologna, Italy, 2002.
10. Doorenbos, R. B., Etsioni, and Weld, D.S.: *A Scalable Comparison-Shopping Agent for the WWW*, in W.L. Johnson and B. Hayes –Roth (eds). Proc, Proceedings of the First International Conference on Autonomous Agents pp. 39-48, Marina del Rey, CA, USA. ACM Press, 1997.
11. Fischmaister, S., Vigna, G., and Kemmerer, R.A.: *Evaluating the Security of Three Java-based Mobile Agent Systems*, Proceedings of the Fifth International Conference on Mobile Agents, Springer, pp,31-41, 2001.
12. FIPA, The FIPA 2000 Specifications, FIPA Website URL: http://www.fipa.org, Accessed May 2005.
13. Julian, V; Rebollo, M; Carrascosa, C.: *Agentes de Informacion*. Revista Base 37. ISSN 1135-0695, 56-62., 2001
14. Klusch, M.: *Information agent technology for the Internet: a survey*, Data and Knowledge Engineering, volume 36 (3), 337-372. 2001.
15. LaMonica, M., *Google's Secret of Success? Dealing With Failure*, CNET News.com,URL:http://news.com.com/Googles+secret+of+success+Dealing+with+failure/2100-1032_3-5596811.html, 2005.
16. Lieberman H. *Letizia: An Agent That Assists Web Browsing*, Proceedings of the International Joint Conference on Artificial Intelligence, Montreal, 1995.
17. Mamma, The Mamma Meta Search Engine, URL: http://www.mamma.com, 1996
18. McBryan O. A'*GENVL and WWWW: Tools for Taming the Web.* First International Conference on the World Wide Web, CERN, Geneva (Switzerland), May 25-26-27 1994.

19. Moukas A., Maes P., *Amalthea: An Evolving Multiagent Information Filtering and Discovery System for the WWW*, invited paper for the first issue of the Journal of Autonomous Agents and Multiagents 1998
20. Neuman, B.: *Scale in Distributed Systems*. In T.Casavant and Singhal (eds). Readings in Distributed Computing Systems, pp 463-489. IEEE Computer Society Press, Los Alamitos, CA., 1994.
21. Odubiyi, J.B., Kocur, D. J., Weinstein S. M., Wakim, N., Srivastava, S., Gokey, C., and Graham, J.: *SAIRE- a scalable agent-based information retrieval engine*, in Proceedings of the first international conference on Autonomous agents, pp. 292-299, Marina del Rey, CA USA, feb. 1997.
22. Peng L., R. Collier, A. Mur, D. Lillis, F. Toolan, J. Dunnion. Self-Configuring Agent-based Document Indexing System, CEEMAS 2005, Budapest, Hungary, 15-17 September, 2005.
23. Peck R, Devore J. Statistics, the exploration and analyses of data, Duxbury Press. 1997.
24. Ramirez, J., Donadeu, J., and Neves, F., *Poirot a relevance-based web search agent*, AAAI Workshop Artificial Intelligence for Web Search. 2001.
25. Salton, G. and Lesk, M.E.: *Computer evaluation of indexing and text processing*. Journal of the ACM, 15(1):8-36, January 1968.
26. Salton, G., Fox, E. A., and Wu, H.. *Extended Boolean information retrieval*. Communications of the ACM, 26(11):1022-1036, 1983.
27. Smyth, B., Freyne, J., Coyle, M., Briggs, P., Balfe, E. (2003) I-SPY - Anonymous, Community-Based Personalization by Collaborative Meta-Search, Proceedings of the 23rd SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence (AI-2003). Cambridge, 2003.
28. Song, R., and Korba, L., *The Scalability of A Multi-agent System in Security Services*. NCR/ERB-1098, NRC 44952,August, 2002,.
29. Suel, T., Mathur, C., Wu, J., Zhang, J., Delis, A., Kharrazi, M., Long, X., Shanmugasundaram, K., *A Peer-to-Peer Architecture for Scalable Web Search and Information Retrieval*, in Proc. 12th International World Wide Web Conference, Budapest, Hungary, 2003.
30. Tong,Y., O' Hare, G. M. P., and Collier, R., *Using agent uml to design cloning in agent factory*, in Proceedings of the 1st Workshop on COnceptual MOdelling for Agents (COMOA 2004), Shanghai, China, 2004.
31. Wijngaards, N.J.E., Brazier, F.M.T., van Steen, M., Distributed Shared Agent Representations. Multi – Agent –Systems and Applications II Vol: 2322, pp.213-220, July, 2002, Springer Verlag, Lecture Notes in Computer Science.
32. Wooldridge, M., and Jennings, N. R., *Intelligent Agents: Theory and Practice*, Knowledge Engineering Review 10(2), 1995.

Reuse and arbitration in diverse *Societies of Mind*

Ciarán O'Leary

School of Computing, Dublin Institute of Technology
Kevin St., Dublin 8, Ireland
www.comp.dit.ie/coleary

Abstract. One popular multi-disciplinary view on the architecture of the minds of natural creatures is of a mind sub-divided and organized into a set of specialised and semi-autonomous agents, each with insufficient intelligence to drive the whole creature alone, but with an ability to work with other agents in what has generally become known as a *society of mind*. Though such models are popular in AI, building truly diverse minds is difficult since it must require the integration of the work of multiple independent authors. Since reuse is not a key concern for AI developers, at least to the degree that it is considered important by software engineers, non-intrusive mechanisms are required to reuse software and integrate agents into societies. One such attempt, using a common interface and simple arbitration is described here.

1 Introduction

The term *Society of Mind*, originally coined by Minsky [18], has come to refer to any type of modular intelligence where multiple, specialised, semi-independent, individually mindless agents contribute to a single, whole intelligence which emerges through their individual expression and collective competition and cooperation.

Agents in such societies employ heterogeneous algorithms, specialised for the goal they are addressing or the task they are designed to carry out. Within the society however, all agents generally present a standard interface to each-other which masks the differences between the algorithms neatly encapsulated inside the agent.

The popularity of this model in artificial intelligence research is due to both the psychological plausibility [7, 10, 23] and the demands placed on the engineer of complex artificial intelligences. From an engineering perspective, it is widely accepted that the modular approach to systems development facilitates incremental development, software reuse, increased understanding of components and improved maintenance [17]. Given the strengths of the modular, or object oriented approach to software engineering, many advances have been made in the development of object oriented patterns, standards, platforms and languages [8, 9, 25].

Outside of the field of multi-agent systems (MAS) [31], particularly Internet agents, the AI community has paid little attention to the standardisation efforts taking place in software engineering. Specifically, those areas of AI closely aligned with

the Society of Mind (SOM) approach have much to gain from standardisation of agent interfaces, although little work has been done in the area.

The most important benefit of standardisation of SOM agents is the ability to take an agent from one society and easily integrate it into another society, effectively *reuse* the agent (and consequently its algorithm) elsewhere. This simplifies the task of the AI researcher who wants to develop systems with truly diverse components, as the details of the implementation are hidden behind an easily understood, standard public interface.

This paper describes a first attempt at developing such a standard interface. We review a number of different approaches to developing agents for an SOM and show the key commonalities that can be easily captured in a standard interface. In brief, all agents in an SOM can provide the following information:

1. An expression of whether or not it is interested in the current state of the world i.e. does it want control of the body.
2. An action, or course of action to pursue, or to avoid.
3. An estimate of the duration of the course of action.
4. An estimate of how urgently it requires control.

In section 2 we describe a standard *Society of Mind*. There are many differences between the numerous implementations described in the published literature over the past decades, so our description is at a relatively high level. The remaining sections discuss the World-Wide-Mind project, and various modular approaches to AI, before describing our attempts to standardise an interface that can be used by diverse agents in an SOM.

2 A *Society of Mind*

The term *Society of Mind* was first introduced in Marvin Minsky's famous book of that title in the mid-1980s [18]. Minsky proposed that the mind is far too complex to be described by a single, clean set of rules, rather the mind must be composed of a large, interconnected set of mindless, focused, diverse specialists, too useless on their own to take command of the creature they were hosted by, but valuable citizens nonetheless in a society of agents. Two of the main points in the proposal, modularity and diversity, had received earlier treatment from Minsky and had served as an alternative view on the ongoing debate between the symbolic and connectionist approaches to AI [19].

In the Society of Mind, and in subsequent work [20], Minsky defined a variety of types of agents, each involved in the management, selection, admission and censorship of other agents who are selfishly trying to express themselves and exploit others. The theory was relatively high level and served as a foundation for much work in AI, with many authors describing their architectures as *Societies of Mind* [26], but Minsky's proposal never turned into an actual implementation. Work directly influenced by this theory is, however, currently being undertaken under Minsky's own supervi-

sion, where the focus is on cognitive diversity, or supporting multiple "ways to think" [27].

2.1 Characteristics

Some of the most important points of the Society of Mind theory are:

1. **Diversity**. This is the single most crucial dimension to the theory. No complex, adaptive mind could be limited to a single approach. Animals display a variety of behaviours, both innate and learned, use different types of memory for different problems, and learn different solutions in different ways. No single set of rules, akin to Newton's Laws in physics, could capture this diversity.
2. **Specialists**. Individual agents in the Society of Mind are specialists, capable of a single small contribution to the society. Some agents specialise in managing other agents by turning them on, or suppressing or censoring their output.
3. **Communication**. No single communication language could prevail throughout the society. Individual agents are not sophisticated enough to be able to speak the same language as all other agents in order to communicate, rather agents exploit or use each other by becoming activated at the same time.
4. **Lack of Centralised Control**: Variously referred to as the *homunculus* problem, or the Cartesian Theatre [10], the Society of Mind rejects the notion of a single centralised agent who is responsible for the management of the entire society.
5. **Redundancy**: Given the diversity of the agents in the society as well as the lack of centralized control, there are a variety of ways to think about, or approach any problem, so the society can continue to function in the absence of any of its agents.

3 SOM Implementations

In this section we describe some AI implementations that align closely with Society of Mind theory, in particular the five key points identified in section 2.

3.1 The Subsumption Architecture

Brooks *subsumption architecture* (SA) [3] represented a new departure for AI. Occurring around the same time as the publication of the Society of Mind, its introduction challenged much established practise in robotics and more generally in AI, as the first implementation of a behaviour based system. Much attention in behaviour based robotics, and more generally in behaviour based AI focussed on the removal of knowledge representation and reasoning [4] as core components of an intelligent system, but equally important was the idea of dividing an intelligent system into individual components that assumed responsibility for taking sensory input directly

from the environment, and producing behaviour by directly influencing the actuators of the system. The components of the system, termed behaviours, were organised into layers where it was assumed that behaviours in the same layer would not conflict with each other. Behaviours in higher layers could over-ride the inputs to, or outputs from, behaviours at lower layers. In this way conflict was averted since higher layer behaviours always took precedence.

3.2 Behaviour Based AI

The past two decades have seen considerable advances in the behaviour based approach to AI [1, 16]. Various implementations have violated the early restrictions on knowledge representation by incorporating state into the behaviours [6]. Of particular interest however, are the wide variety of conflict resolution strategies that have been tried and tested [24]. One of the first deviations from the SA was Steel's architecture [28], where the outputs from the behaviours were summed to produce the cumulative output of the whole system. In his system all output was realised, and the complex behaviour of the robot emerged.

Maes' network of competencies [14] marries together a connectionist approach to intelligent systems with symbolic representation. Competencies represented activation controlled agents which triggered when a set of preconditions were met. Activation spread between competencies along links established according to conditions and lists associated with each competency. The action of the system was determined according to the activation levels, so like Brooks' and Steels' architecture, there was no centralised action selection or arbitration.

Tyrrell's [30] extension of roboticists Rosenblatt and Payton's architecture provided action selection by passing activation within a hierarchy of nodes, where internal nodes represented neither primitive actions nor whole agents. Leaf nodes represented primitive actions, where the action with the highest level of activation was chosen by the system.

Bryson's *Behaviour Oriented Design* (BOD) [6] is a methodology based on Object Oriented Design [9], with a centralised action selection mechanism. Behaviours are designed in an iterative fashion according to the requirements of the system, and encapsulate all the perceptive and action producing state and behaviour required. Action selection is centralised in a dedicated module, which identifies prioritised drive collections, competencies and action patterns from which it selects a behaviour to execute.

3.3 Modular Reinforcement Learners

Reinforcement learning (RL) [29] is a technique that allows an agent develop its own model of action in an environment by trying out actions according to a policy which is updated as the agent receives rewards and punishments. It removes the need for hand coding courses of action into the agent, but is limited by the memory requirements for large state-action spaces. A frequently used technique for addressing

the problem of large state spaces is the division of a single agent into a society of agents, each member of which takes responsibility for learning sub-sections of the state space, or learning about individual goals or sub-goals. In behaviour based AI it has been used to both learn how to behave, and how to co-ordinate behaviour [15], often separately but sometimes together [11]. An advantage of combining learning within the individual agent with learning co-ordination in the society is that the reinforcement values learned internally by the individual agents can be propagated to higher levels where they can be considered as expressions of *how good* an agent considers an action to be.

Hierarchical RL [2] has become popular in recent years, where learning occurs simultaneously at multiple levels. In some cases the agent is permitted to follow a course of action without interruption, whereas in other cases decisions are made at higher levels, or collectively among the agents using their reinforcement values. This covers instances of both centralised and decentralised action selection.

3.4 Action Selection

Action selection is an SOM is the problem of choosing, at any point in time, the best behaviour to execute, or the best agent to listen to, or ultimately the best action for the body (animal, robot) to take. An agent in an SOM is any component that can proactively choose an action, or course of action, and suggest it, or try to execute it. In the extensive literature on the subject, these agents are variously referred to as behaviours, layers, systems, modules, competences, drives, beings, demons and homunculi among others. The key concern for us is the type of information that these agents make available to each-other in order for action selection to take place in the society as a whole.

Tyrrell [30] and Maes [14] both treat the action selection problem, by providing lists of the characteristics of effective action selection. A summarised list is given here:

1. Action selection must try to satisfy all the goals of the creature.
2. It must persist with a goal until completion, unless there is a much greater benefit for switching to another goal. This must be balanced with the requirement to be opportunistic and reactive to allow goals to be quickly spotted and satisfied where possible.
3. It must be highly adaptive to changing environments.
4. It must be able to choose *compromise candidates*, or actions that can satisfy more than one goal.

In general it is assumed that a single agent in an SOM would address a single goal, although this is not necessarily always the case. When constructing an SOM from diverse agents, it is important that each agent can provide sufficient information so that the action selection mechanism, or more generally, the arbitration scheme, can attempt to satisfy the criteria listed.

4 World-Wide-Mind

The work described in this paper is taking place as part of the World-Wide-Mind (WWM) project [12, 13], a project which attempts to use the World-Wide-Web as a mechanism for supporting the scaling up of AI research. Using WWM technology, authors of new algorithms, or agents, can make their software available online as a web service which can be interacted with remotely. Third parties can then build agents which reuse existing agents by incorporating them into societies and arbitrating between their action choices. Communication between agents takes place by exchanging simple XML over HTTP, meaning that the technical knowledge required of authors is reduced to a familiarity with some basic web technologies such as CGI or Java Servlets.

At its most basic level, WWM agents are queried by client software at each time-step. The purpose of the client is to take data from a single agent, and present it to a body which can use the data to execute an action or behaviour in a real or simulated world. Societies of agents are constructed by providing a high level agent that arbitrates between a set of pre-existing agents – effectively carrying out action selection akin to standard behaviour based architectures. Both the world and the agent are available online as web services, thus making them available to all web users. All the components in a society could be developed independently by multiple authors, thus facilitating the type of diversity that is rare in a single research group.

5 Interface

The interface that is presented by each of the minds, or agents, at the WWM entry level just requires each agent to provide an action at each time-step. This requires that the higher level arbitration does not have a great deal of information which it can use to perform action selection, thus requiring the author of the arbitration mechanism to either perform a detailed analysis of the performance of each agent separately [22], or contact the author of the individual agent to establish how it is making its decision. Without this, it is difficult to perform the type of action selection described in section 3.4.

The interface we describe in this section can be used by agent developers to provide more sophisticated information to arbitration mechanisms, which can in turn perform better action selection.

When queried at each time-step, an agent will return as much of the following information as possible:

For each action that it is interested in, either to take or avoid (if it is not interested in taking or avoiding any actions i.e. it does not trigger, it returns nothing), it returns an action tuple made up of `<a, g, p, t, d, w>` where

- `a` is the action proposed.
- `g` is the main goal currently being pursued which can take a null value.

- `p` is the priority (between 0 and 1). All actions proposed by an agent should be prioritised relative to each-other. If only one action is being proposed, or if all actions are of equal priority then no value is required.
- `t` is a boolean value which is used to capture whether or not the agent is seeking to take the action (set to `true`), or avoid the action (`false`).
- `d` is the distance to goal. If this takes a value of 0 then the proposed action should result in the agent achieving goal `g`. Otherwise the agent is indicating that there is no point selecting this action now, if its actions for the next `d` steps are not selected.
- `w` is the maximum waiting time. Using this value, an agent can state that it is willing to let another agent have a go once it can get control within `w` steps.

Using these tuples, agents can express the following:

1. Whether or not they are interested in the competition
2. A group of actions that they would be equally satisfied with.
3. The beginning of an action pattern, or sequence.
4. The achievement of a goal.
5. The urgency of its action choice.

6 Experiment

The problem world used for our experiment is a well known environment used in animat (animal/robot) research [32]. Tyrrell's SE [30] models a small animal in a heavily populated dynamic environment. As well as the animal, the world contains fruit, cereal food and water (any of which can be toxic), prey, two types of predator, other animals, cover where the animal can hide, shade where the animal can cool down, dangerous places where the animal can be injured or killed, landmarks the animal can use for navigation, a den where the animal can sleep, and creatures with which the animal can mate. The animal can choose from a set of 35 different actions (9 looking to improve perception, 16 moving, 4 eating or drinking, 2 courting or mating, cleaning, sleeping, resting and freezing to avoid detection by predators). The animal can only choose one action, but must over time satisfy each of its goals (cleaning, obtaining food, obtaining water, temperature regulation, predator avoidance, vigilance, hazard avoidance, irrelevant animal avoidance, sleeping at night, staying close to cover, not getting lost, reproduction). Failure to satisfy each of its goals to some degree will negatively affect the animal's health resulting in its death. The measure of the success in the SE is the number of times the animal successfully mates, but the animal must ensure it lives long enough to be presented with mating opportunities.

6.1 A Society of Mind for Tyrrell's Simulated Environment

Five agents (*hunter*, *lookout*, *maintenance*, *mater* and *navigator*) were implemented using simple motivational based algorithms, hidden behind the interface described above. Arbitration could have been conducted according to any number of algorithms, but to demonstrate how information from the action tuples can be employed in arbitration, we designed a priority based algorithm, described here.

Each of the agents is given a *level*. At every time step the actions of the highest level agent are considered first. If any action is suggested which has a waiting time of 0, it is selected. If more than one action fits this category then actions from the next level are used to break the tie. If no action at the top level has a waiting time of 0 then the second level is considered. Where numerous actions are of equal priority, action choices from the top level down are considered. If no action is selected after going through all agents then the process is repeated for a higher waiting time. This simple algorithm does not make use of all of the information available from the agents but it is able to make good choices with the limited information it uses.

```
agents = { lookout, mater, maintenance, hunter, navigator }
wait_max = 0
while true
  for each agent in agents
    for each action in agent
      if waiting <= wait_max
        if no other <= wait_max return action
        else examine other levels return action
    end for_each
  end for_each
  wait_max = wait_max + 1
end while
```

6.2 Results

Tyrrell implemented five different minds for the creature in his simulated environment. Results are shown here for the two algorithms that achieved the best results in four versions of his world. (ER&P is the Extended Rosenblatt and Payton algorithm, a monolithic algorithm based on the work of the two named roboticists. Drives is a simple ethologically inspired motivation based algorithm)

Version	Standard	Version 1	Version 2	Version 3
ER&P	8.09	3.61	8.16	13.38
Drives	6.44	3.29	6.41	8.78

These values are the average number of times the animal mated over 1,650 runs in the world. Tyrrell's code is freely available online, so each of his algorithms was re-tested along with the *five agent society of mind* giving the following results.

Version	Standard	Version 1	Version 2	Version 3
ER&P	7.74	3.55	7.88	13.03
Drives	7.11	3.55	7.14	8.95
SOM	6.86	3.54	6.35	8.3

Bryson [5] produced a modified version of Tyrrell's world where food was made more scarce – in which she tested her own POSH algorithm as well as Tyrrell's ER&P. Results for her algorithm and the society of mind in this world are below.

Version	Standard	Version 1	Version 2	Version 3
ER&P	4.77	2.46	4.56	12.53
POSH	8.17	3.56	10.79	10.74
SOM	4.18	2.47	4.6	6.98

The society of mind was able to perform as well as other minds in some worlds. The difficulty of reusing agents not specifically designed to work together remains but with well designed agents, arbitration can be improved once sufficient information is available.

7 Summary

The importance of diversity in intelligent systems is well understood. For certain AI problems, diversity is a key requirement so mechanisms must be provided that facilitate the smooth integration of diverse components. It is important when diverse components are being integrated, that a balance is reached between the need to provide a standard interface to each of the components, and the need to express information about how decisions were made internally in the component, or agent.

We have described how a standard interface is provided for developing societies of mind. We have built on early work on the World-Wide-Mind project by showing that a more sophisticated set of parameters at the interface to an agent can capture information that is useful for arbitration. We also provide a straightforward priority based arbitration mechanism that was used to select between agents and their action choices.

Designing for reuse and integrating reusable components can lead to novel and interesting combinations of agents in societies - resulting in novel and interesting behaviour in robots and other artificial intelligences. It also presents opportunities for diverse and distributed groups of people to collaborate on the AI problem. Given the scale of the problem being dealt with, perhaps reuse and integration is the only way to keep our eyes on the prize [21].

References

1. Arkin, R.: Behavior-Based Robotics. MIT Press, Cambridge, MA, (1998)
2. Barto A. and Mahadevan S.: Recent Advances in Hierarchical Reinforcement Learning, Discrete Event Dynamic Systems: Theory and Applications, 13, 41-77 (2003)
3. Brooks, R.: Robust layered control system for a mobile robot. IEEE Rob. & Auto. (1986)
4. Brooks, R.: Intelligence without Representation, Artificial Intelligence 47, (1991)
5. Bryson, J.: The Study of Sequential and Hierarchical Organisation of Behaviour via Artificial Mechanisms of Action Selection, MPhil: University of Edinburgh (2000)
6. Bryson, J., Intelligence by Design: Principles of Modularity and Coordination for Engineering Complex Adaptive Agents, PhD Thesis, MIT, 2001
7. Buss, D.:, Evolutionary Psychology: New Science of the Mind, Allyn & Bacon (2003)
8. Capretz L.F.: A Brief History of the Object-Oriented Approach, ACM SIGSOFT Software Engineering Notes, 28(2), March 2003
9. Coad, P. and Yourdon, E.: Object Oriented Analysis (2nd Edition), Prentice Hall (1990)
10. Dennett, D.: Darwin's Dangerous Idea, Simon and Schuster, (1995)
11. Humphrys, M.: Action Selection methods using Reinforcement Learning, PhD Thesis, University of Cambridge (1997)
12. Humphrys, M.: The World-Wide-Mind: Draft Proposal, Dublin City University, School of Computing, Tech Report CA-0301 computing.dcu.ie/~humphrys/WWM/ (2001)
13. Humphrys, M. and O'Leary, C.: Constructing complex minds through multiple authors, 7th International Conference on the Simulation of Adaptive Behavior (SAB-02). (2002)
14. Maes, P.: How to do the right thing. A.I. Memo 1180, MIT, Cambridge, MA (1989)
15. Martinson, E.: Stoytchev, A. and Arkin, R.: Robot Behavioral Selection Using Q-learning, IEEE/RSJ Conference on Intelligent Robots and Systems (IROS), (2002)
16. Mataric, Maja.: Behavior-Based Robotics as a Tool for Synthesis of Artificial Behavior and Analysis of Natural Behavior, Trends in Cognitive Science, Mar 1998, 82-87 (1998)
17. Mili, H., Mili, A., Yacoub, S., Addy, E.: Reuse Based Software Engineering: Techniques, Organizations, and Measurement, Wiley (2002)
18. Minsky, M.: The Society of Mind, Simon and Schuster (1985)
19. Minsky, M.: Logical vs. Analogical or Symbolic vs. Connectionist or Neat vs. Scruffy, Artificial Intelligence at MIT., Expanding Frontiers, Patrick H. Winston (Ed.). (1990)
20. Minsky, M.: The Emotion Machine, online at web.media.mit.edu/~minsky/ (2005)
21. Nilsson, N.J.: Eye on the Prize, AI Magazine 16(2):9-17, Summer (1995)
22. O'Leary, C.: Humphrys, M. and Walshe, R., Constructing an animat mind using 505 agents from 234 different authors, 8th Int. Conf. Simulation of Adaptive Behavior (2004)
23. Pinker, S.: How the Mind Works, W. W. Norton & Company (1999)
24. Pirjanian, P.: Behavior Coordination Mechanisms - State of the Art, Ph.D., USC (1999)
25. Schmidt, D., Stal, M., Rohnert, H. and Buschmann, F.: Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects, Wiley and Sons (2000)
26. Singh, P.: Examining the Society of Mind. Computing and Informatics, (2003)
27. Singh, P. and Minsky, M.: An architecture for cognitive diversity. Visions of Mind, Darryl Davis (ed.), London: Idea Group Inc. (2005)
28. Steels, L.: A Case Study in the Behavior Oriented Design of Autonomous Agents, Proc. 3rd Int. Conf. on Simulation of Adaptive Behavior (SAB-94) (1994)
29. Sutton, R. and Barto, A.: Reinforcement Learning: An Introduction, MIT Press, (1998)
30. Tyrrell, T.: Computational Mechanisms for Action Selection, PhD Thesis, Edinburgh, Centre for Cognitive Science (1993)
31. Weiss, G.: Multiagent Systems: Modern Approach to Distributed AI. MIT Press (1999)
32. Wilson, S.: The animat path to AI, 1st Int. Conf. on Sim. of Adaptive Behavior (1990)

Improving Incremental Critiquing

Maria Salamó, James Reilly, Lorraine McGinty, and Barry Smyth

Adaptive Information Cluster, Smart Media Institute,
Department of Computer Science, University College Dublin (UCD), Ireland
{maria, james.d.reilly, lorraine.mcginty, barry.smyth}@ucd.ie

Abstract. In order to be useful, conversational recommender systems should be able to guide users through a product space in the most efficient manner possible, making the most of elicited user feedback to discover suitable products in the least number of recommendation cycles. *Critiquing* is one form of feedback for case-based recommender systems that has recently emerged as a common way to tackle this problem. A *critique* is a user's directional preference in the product space. For example, a user might ask for a '*less* expensive' vacation in a travel recommender; '*less* expensive' is a critique over the *price* feature. Previous work on critiquing-based recommenders has utilized critiques to construct an explicit model of user preferences. It is used to positively influence the recommendations made to the user over the course of a session. In this paper, we propose an alternative strategy, using a Reinforcement Learning technique to discover suitable products for recommendation based on the elicited critiques. Our results show that this technique can better utilize user preferences, reducing recommendation session length, while being more computationally efficient than previous techniques.

1 Introduction

As information has become more abundant and information access more pervasive, people are becoming overwhelmed with the number of choices regarding how to access, navigate through, and select options in complex information spaces; the so-called *information overload problem* [1]. Recommender systems are designed to alleviate this problem by assisting the user to make choices, by helping to guide and inform the decision making process. Conversational recommender systems [2] help users navigate through complex product spaces in pursuit of suitable products using a cyclical recommendation process. In each recommendation cycle, the recommender system will suggest new products to the user and solicits feedback in order to refine its search for an appropriate suggestion for the user in the next cycle [3]. At each recommendation cycle, the recommender utilizes user preferences in order to prioritise products from all the available options, moving the user to a different part of the product space, closer to their target product than they were previously. The main advantage of such systems is that users have more control over the navigation process [4] and at the same time guide users to target products faster than standard browsing and/or alternative recommendation approaches [4–6].

In this paper we will use a well-known form of user feedback called *critiquing* [4]. A *critique* is a directional feature preference made by the user in relation to a presented recommendation. For example, in a travel vacation recommender, a user might indicate that they are interested in a vacation that is *longer* than the currently recommended option; in this instance, *longer* is a critique over the *duration* feature. Traditional implementations of critiquing can lead to protracted recommendation sessions as they tend to focus only on the most recent critique, ignoring preferences supplied earlier. For example, a holiday-maker might have received a recommendation for a 2-week package in Spain for €1500. She might be interested in a 1-week holiday for around €900 and so indicates that she would like something *cheaper*. The system then recommends a 2-week vacation for €800. The user decides to tackle the *duration* feature in order to get a shorter holiday. However, on this occasion, she receives a recommendation for a luxury 1-week holiday costing €1550 which satisfies the most recent critique but not the earlier critique.

To address this issue, Reilly *et al.* introduced a technique called *Incremental Critiquing* [7], which maintains a model of user preferences over the course of a recommendation session and uses this model to make better recommendations by taking into account *all* of the user preferences. This approach has shown significant improvements on recommendation efficiency over standard critiquing systems. In this paper, we propose an alternative approach to incremental critiquing that combines user profiling and reinforcement learning to good effect. In particular we show that this new approach offers considerable computational efficiency benefits when compared to conventional incremental critiquing, while at the same time marginally improving recommendation quality.

2 Background

Critiquing, as a form of feedback, is best known by association with the FindMe recommender systems [4, 8]. The original motivation for critiquing included the need for a feedback mechanism that was simple for users to understand and apply, and yet informative enough to focus the recommender system. For instance, the Entrée restaurant recommender [4] presents users with a fixed set of directional critiques in each recommendation cycle. In this way users can easily request to see further suggestions that differ from the current recommendation, in terms of some specific feature. For example, the user may request another restaurant that is *cheaper* or *more formal*, by critiquing its *price* and *style* features, respectively.

One problem that tends to occur with the standard critiquing system like Entrée is that when it comes to making a recommendation, no direct consideration is given to the feedback provided by users during previous cycles of the same session but rather the system only focuses on the last critique. *Incremental critiquing* [7] seeks to deal with this problem by giving due consideration to past critiques during future recommendation cycles. It does this by maintaining a session-based *user model* that is made up of those critiques chosen by the user

so far. This model is given by $U = \{U_1, ..., U_n\}$, where U_i is a single unit critique. It is used during recommendation to influence the choice of a new product case, along with the current critique. Maintaining an accurate user model, however, is not quite as simple as storing a list of previously selected critiques. Users may not always provide consistent feedback. Sometimes they make mistakes or change their mind. To eliminate preference inconsistencies, the incremental critiquing strategy updates the user model by adding the latest critique only after pruning previous critiques. Specifically, prior to adding a new critique all existing critiques that are inconsistent with it are removed, as are all existing critiques for which the new critique is a refinement. The basic idea behind the user model is that it should be used to influence the recommendation process, prioritising those product cases that are compatible with the user's critiques. The standard approach to critiquing is a two step procedure. First, the remaining cases are filtered, eliminating those cases that fail to satisfy the current critique to leave the set of *relevant* cases. Next, these relevant cases are rank ordered according to their similarity to the current recommendation. Incremental critiquing makes one important modification to this procedure. Instead of ordering the filtered cases on the basis of their similarity to the recommended case, it also computes a *compatibility* score (see Equation 1) for each candidate case. The compatibility score is essentially the percentage of critiques in the user model that this case satisfies. It is important to note that $satisfies(U_i, c')$ returns a score of 1 when the filtered case, c', satisfies the critique, U_i, and returns 0 otherwise. Thus a case that satisfies 3 out of the 5 critiques in a user model obtains a compatibility score of 0.6.

$$Compatibility(c', U) = \frac{\sum_{\forall i} satisfies(U_i, c')}{|U|} \tag{1}$$

$$IQual(c', c, U) = \beta * Compatibility(c', U) + (1 - \beta) * Sim(c', c) \tag{2}$$

The compatibility score and the similarity of candidate c' to the current recommended case, c, are then combined in order to obtain an overall *quality* score (In equation 2 above, the parameter β is usually set to 0.75). The quality score is used to rank the filtered cases prior to the next recommendation cycle; of course, the case with the highest quality is then chosen as the new recommendation. The above formulation allows us to prioritise those candidate cases that: (1) satisfy the current critique; (2) are similar to the previous recommended case; and (3) satisfy as many as possible of the previous critiques. In so doing we are implicitly treating the past critiques in the user model as *soft constraints* for future recommendation cycles; it is not essential for future recommendations to satisfy all of the previous critiques, but the more they satisfy, the better they are regarded as recommendation candidates. Moreover, given two candidates that are equally similar to the previously recommended case, the algorithm prefers the one that satisfies the greater number of critiques.

3 An Alternative Approach

While incremental critiquing is quite effective, we believe that there is room for further improvement. In this section, we describe a similar strategy which also considers all user preferences when making recommendations, but does not need to retain an explicit model of user preferences. This new strategy has the potential to produce better recommendations while also being more computationally efficient. It differs from incremental critiquing in two aspects: (1) it uses a different case compatibility measure based on Reinforcement Learning; and (2) it uses a different case selection algorithm.

3.1 Compatibility based on Reinforcement Learning

The incremental critiquing compatibility measure suffers from two basic problems. First of all, it treats all user model critiques equally when computing case compatibility. For example, a case that satisfies the first and third of three possible critiques, and a case that satisfies the second and third will have the same compatibility value of 0.66. We believe that the more recent critiques should hold more sway that earlier critiques because over time, the critiques will become more refined, better expressing user preferences than earlier critiques. Secondly, the compatibility values produced are highly dependant on the number of critiques in the user model. This could make it difficult to adapt to different domains where the size of the user model is very large or very small. Ideally, we would like a compatibility score that does not suffer from these problems.

An alternative way to view case recommendation is as an optimization problem in which we are trying to recommend cases that maximise the satisfiability of user preferences. Accordingly, we can evaluate the remaining cases as if they were a set of states in a *Reinforcement Learning Problem* (RLP) [9], which consists of maximising the sum of future rewards in a set of states. Reinforcement Learning theory is usually based on *Finite Markov Decision Processes* (FMDP). The use of FMDP allows a mathematical formulation of the RLP and therefore the suitability of RL algorithms for optimization can be mathematically proven. Each case is treated as a state whose compatibility score is updated at each cycle using a Monte-Carlo value function (see Equation 3). This function evaluates the *goodness* of each state — for us the possible states are the complete set of remaining cases we want to enhance — according to the critiques the user has selected.

$$Compatibility(c', U_f) = \begin{cases} comp(c') + \alpha \times (1 - comp(c')) \; \textit{if } c' \textit{ dissatisfies } U_f \\ comp(c') + \alpha \times (0 - comp(c')) \; \textit{otherwise} \end{cases} \tag{3}$$

Our goal is to maximally satisfy all the user preferences. Thus, we are looking for a set of maximally compatible cases (i.e., those cases which have the highest compatibility *(comp)* value considering all the user preferences (U) or past critiques). At the beginning of each session each candidate case, c', has a default

compatibility value (i.e., $comp(c')= 0.5$). The compatibility value for each case is updated on every cycle, taking into account whether or not it satisfies the current critique for that cycle.

The α parameter in Equation 3 is the learning rate which is usually set up to 0.1 or 0.2 values; a larger value leads to a larger gap between cases in early stages. In our case, the learning rate is not important since we are looking for levels of satisfaction. In other words, we are not trying to obtain a set of states that arrive as quickly as possible to a 1.0 value, as usually is done in RLP.

It is important to note that Equation 3 updates the compatibility value stored by each case according to the last user critique (U_f) as opposed to computing all the set of critiques like the incremental approach (see Equation 1). The $Compatibility(c', U_f)$ value computed in the current cycle will be the $comp(c')$ in the next cycle. At the start of a recommendation session each case is assigned a default value. At every cycle, after the user specifies a critique, the compatibility for each case is updated taking into account it's previous compatibility value and the current critique only. The compatibility function only evaluates the n remaining cases to one critique, which produces a computational cost of $O(n)$.

For evaluating case compatibilities, the RL strategy is much more computationally efficient than incremental critiquing. When computing the compatibility of cases, an incremental recommender needs to check the compatibility of all relevant cases (n) to *all* of the critiques contained in the user model (U) (see Equation 1). Put differently, if the user-model contains (u) critiques and there are (n) cases relevant to the current critique, the computational cost of evaluating the compatibility is $O(n \times u)$. This cost may seem small but if the product space is large, the cost of computing the compatibility score may be significant.

However, users may not be entirely consistent in their feedback behavior, they may make mistakes or make sub-optimal critique selections. Incremental critiquing caters for this situation by pruning the user model of critiques that contradict previously supplied critiques and critiques for which the current selection is a refinement. The RL strategy doesn't deal explicitly with contradictory and refining critiques, but the RL compatibility score does prioritise those cases that satisfy the most recent critiques. So for example, if a user supplies a critique [$<$ €500] in one cycle and then a few cycles later supplied [$>$ €500], the latter, more recent critique will increase the score for compatible cases while reducing the score of the incompatible cases. In this way, the RL strategy intrinsically controls the contradictory critiques and so will still promote the most suitable cases for recommendation.

3.2 Highest Compatibility Product Recommendation

One of the drawbacks with the original incremental critiquing strategy is that it does not always return recommendations that maximally satisfy the user's preferences. With incremental critiquing, cases are ranked according to a *quality* metric, which is a simple function combining case compatibility *and* similarity (see Equation 2). However this is not ideal, as we would prefer to retrieve those cases which maximally satisfy the user preferences and recommend from that set,

the cases that are most similar to current query. By combining compatibility and similarity together to rank-order cases, the incremental critiquing quality metric does not always exhibit this desired behaviour, sometimes preferring a more similar case rather than the most compatible. Another issue with the incremental strategy is that the similarity of every relevant case must be computed even if that case is not very compatible with user model.

We propose a new strategy for product recommendation, *Highest Compatibility Selection* (HCS) that aims to eliminate this problem. The new strategy is presented below in Figure 1. When selecting cases for recommendation, there are two steps. Firstly, we rank order the list of relevant cases according to the compatibility score and select from that list those cases with the highest compatibility score (CB''). Secondly, we sort that list in order of decreasing similarity to the currently recommended case. The net result of this new recommendation strategy is that cases that are most compatible with the user's preferences are recommended, without letting the similarity score adversely effect the recommendations made. This strategy also reduces the computational effort required to recommend a case, as the system only computes the similarity for those cases with the highest compatibility score, whereas previously it was computed for every case.

```
q: query, CB: CaseBase, cq: critique, c_r : current recommendation, U: User Model

1.define ItemRecommend(q, CB, cq, U)
2.  CB' ← {c ∈ CB | Satisfies(c, cq)}
3.  CB' ← sort cases in CB' in decreasing compatibility score
4.  CB'' ← selects those cases in CB' with highest compatibility
5.  CB'' ← sort cases in CB'' in decreasing order of their sim to q
6.  c_r ← most similar case in CB''
7.return c_r
```

Fig. 1. Adapting the incremental critiquing algorithm *ItemRecommend* procedure to use the Highest Compatibility Selection strategy

4 Evaluation

So far we have argued that the traditional and incremental forms of critiquing are limited by their tendency to recommend cases that do not always maximally satisfy user preferences. We have proposed two modifications. Firstly, we have proposed a different case-compatibility measure using Reinforcement Learning. Secondly, we have proposed a new case recommendation strategy that encourages the system to recommend cases that maximially satisfy user preferences. In this section we describe the results of our evaluation which show that these modifications combined can lead to efficiency improvements in terms of reduced session length for the user and reduced computational time for producing recommendations. This evaluation uses the same methodology as described in [10, 11], the details of which are provided in the following sections.

4.1 Setup & Methodology

The evaluation was performed using the standard Travel dataset (available from *http://www.ai-cbr.org*) which consists of 1024 vacation cases. Each case is described in terms of 9 features including *price*, *duration*, etc. The dataset was chosen because it contains numerical and nominal features and it also provides a wide search space. We are evaluating a system using highest compatibility selection *HCS* strategy with the Reinforcement Learning compatibility score (which we will term HCS-RL) and compare it to an incremental critiquing system (IC). The efficiency improvements of incremental critiquing over standard critiquing have already been documented [7]. For this reason, we only compare our new system to an incremental system.

In our experiments, each case (*base*) in the case base is temporarily removed and used in two ways. First, it serves as a basis for a set of queries by taking random subsets of its features. We focus on subsets of 1, 3 and 5 features to allow us to distinguish between hard, moderate and easy queries respectively. Second, we select the case that is most similar to the original base. These cases are the recommendation targets for the experiments. Thus, the base represents the ideal query for a user, the generated query is the initial query provided by the 'user', and the target is the best available case for the user. Each generated query is a test problem for the recommender, and in each recommendation cycle the 'user' picks a critique that is compatible with the known target case; that is, a critique that when applied to the remaining cases, results in the target case being left in the filtered set of cases. Each leave-one-out pass through the case-base is repeated 10 times and the recommendation sessions terminate when the target case is returned.

4.2 Recommendation Efficiency

We are interested in how the proposed HCS-RL strategy affects recommendation efficiency, that is, the number of cycles before the target case is returned to the user. Importantly, we are not necessarily expecting HCS-RL to significantly improve recommendation efficiency by producing shorter sessions. The real benefit of HCS-RL is likely to be derived from its computational efficiency compared to incremental critiquing, because the case selection strategy is similar in its basis for both strategies but different in computational complexity. However, it is important, if this is the case, to ensure that the computational efficiency benefits do not come at a session length cost. Figure 2 (A and B) indicates that HCS-RL does not incur such a cost. In every case, session length remains at incremental critiquing levels. In fact, we witness minor but consistent reductions in session length across all query sizes.

Figure 2(A) shows that the HCS-RL setup consistently improves on the recommendation efficiency of incremental critiquing across all query difficulties. For example, for the easiest queries (5 features specified), the HCS-RL results in an average session length of 8 cycles, compared to the 8.3 cycles for the IC setup – a session length decrease of nearly 4%. For the most difficult queries (only 1

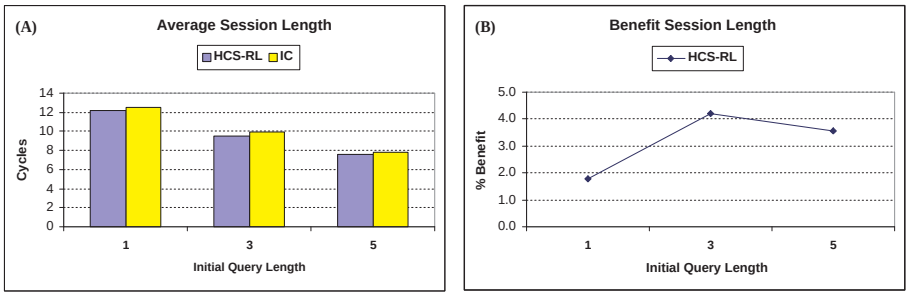


Fig. 2. Average session length per recommender

feature specified), HCS-RL produces sessions of length 12.0 cycles, whereas the IC system takes on average 12.8 cycles and the benefit here falls to around 2%. In summary, HCS-RL marginally outperforms incremental critiquing in terms of session-lengths, the average session length reduction is between 2% and 4%.

4.3 Computational Time

The results so far show that our HCS-RL manages to achieve similar session lengths to the standard approach to incremental critiquing. However, as we illustrated earlier, the HCS-RL system should be more efficient in terms of computational complexity. This can be quite important if the system has to deal with thousands of users simultaneously searching a large product catalogue. When carrying out the above experiments, we also measured the time spent by both setups in milliseconds to verify that HCS-RL is more computationally efficient.

Figure 3 graphs the average number of milliseconds spent by each setup per cycle and the benefit relative to the incremental critiquing approach for easy, moderate and difficult queries. The computational time for the IC setup remains static, with each cycle taking roughly 3.0 ms for all query difficulties. As expected, the HCS-RL setup is much more efficient than IC, taking around 2.25 ms for sessions starting with the easiest queries reducing down to 2.0 ms for the hardest queries. From the benefit graph, it's quite clear to see that HCS-RL is much more efficient than IC as query difficulty and session length increases – with queries of length 5 the benefit is around 25% and this rises to 35% for the most difficult queries.

To assess the overall savings in terms of computational time offered by the HCS-RL system, we calculated the average computational time per session ($t_{session} = \#cycles \times t_{cycle}$) which is the sum of the computational time of all the cycles in a session. This gives us a much clearer indication of the benefit of using HCS-RL in comparison to IC. Figure 4(A) graphs the total session time in milliseconds for each approach for different query sizes. The results are similar to the ones presented above on the per cycle-basis, but we can see that time per session decreases for moderate and easy queries because the number of cycles is also reduced. From the benefit graph we can see that overall computation time is

decreased by between 43% and 46%. Put differently, the the HCS-RL approach can search for nearly double the number of products in the same time as IC.

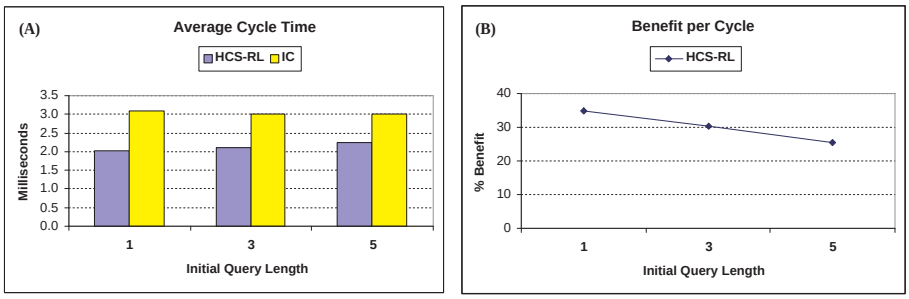


Fig. 3. Computational time per cycle for each recommender

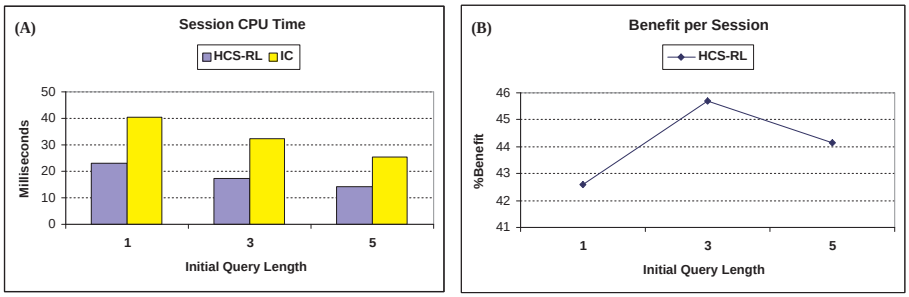


Fig. 4. Computational time per session for each recommender

5 Conclusions

Critiquing is an important form of user feedback that is ideally suited to many recommendation scenarios. It is straightforward to implement, easy to understand and use, and it has been shown to be effective at guiding conversational recommender systems. There has been a lot of research focusing on improving the efficiency of critiquing in conversational recommender systems. Incremental critiquing is one strategy for achieving this, by maintaining a model of user preferences in order to make recommendations that are compatible with all of the user's requirements, rather than just the most recently elicited preference. The net result is that by using the incremental critiquing strategy, the recommender focuses quickly on the areas in the product-space that are more in line with the user's preferences.

In this paper, we have proposed an alternative strategy for incremental critiquing that aims to achieve the same recommender efficiency results but without the overhead of maintaining an explicit user model. This strategy is made up of two techniques. Firstly, we have introduced a more efficient case recommendation strategy, *highest compatibiblity selection*, which selects those cases that maximally satisfy the user's preferences. Secondly we have introduced a new case-compatibility score, which uses a Reinforcement Learning technique to rank cases according to their compatibility with the user's preferences. This technique is also more computationally efficient than incremental critiquing.

Our experiments indicate that the proposed product recommendation strategy has the potential to significantly reduce the computational time and also to slightly improve recommendation efficiency. These computational improvements allow the proposed strategy to support much larger product case-bases without suffering any loss in performance when compared to incremental critiquing. Importantly, our proposal is sufficiently general to be applicable across a wide range of recommendation scenarios, especially those that assume a complex product-space where recommendation sessions are likely to be protracted.

References

1. Smyth, B., Cotter, P.: A Personalized TV Listings Service for the Digital TV Age. Journal of Knowledge-Based Systems **13(2-3)** (2000) 53–59
2. Aha, D., Breslow, L., Muñoz-Avila, H.: Conversational Case-Based Reasoning. Applied Intelligence **14** (2000) 9–32
3. McSherry, D., Stretch, C.: Automating the Discovery of Recommendation Knowledge. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, Morgan-Kaufmann (2005)
4. Burke, R., Hammond, K., Young, B.: The FindMe Approach to Assisted Browsing. Journal of IEEE Expert **12(4)** (1997) 32–40
5. McGinty, L., Smyth, B.: Comparison-Based Recommendation. In Craw, S., ed.: Proceedings of the 6th European Conference on Case-Based Reasoning, Springer (2002) 575–589 Aberdeen, Scotland.
6. Shimazu, H.: ExpertClerk: A Conversational Case-Based Reasoning Tool for Developing Salesclerk Agents in E-Commerce Webshops. Artificial Intelligence Review **18(3-4)** (2002) 223–244
7. Reilly, J., McCarthy, K., McGinty, L., Smyth, B.: Incremental Critiquing. In Bramer, M., Coenen, F., Allen, T., eds.: Research and Development in Intelligent Systems XXI. Proceedings of AI-2004, Springer (2004) 101–114 Cambridge, UK.
8. Burke, R., Hammond, K., Young, B.: Knowledge-Based Navigation of Complex Information Spaces. In: Proceedings of the 13th National Conference on Artificial Intelligence, AAAI Press/MIT Press (1996) 462–468 Portland, OR.
9. Harmon, M., Harmon, S.: Reinforcement learning: A tutorial (1996)
10. Salamó, M., Smyth, B., McCarthy, K., Reilly, J., McGinty, L.: Reducing Critiquing Repetition in Conversational Recommendation. In: Proceedings of the Workshop on Multi-Agent Information Retrieval and Recommender Systems at the 19th International Joint Conference on Artificial Intelligence. (2005) To Appear
11. McCarthy, K., McGinty, L., Smyth, B., Reilly, J.: On the Evaluation of Dynamic Critiquing: A Large-Scale User Study. In: Proceedings of the 20th National Conference on Artificial Intelligence, AAAI Press / The MIT Press (2005) 535–540

A Framework for the Automatic Description of Musical Structure Using MPEG-7 Audio

Elaine Smyth, Kevin Curran, Paul Mc Kevitt, and Tom Lunney

School of Computing and Intelligent Systems
Faculty of Engineering
University of Ulster, Magee
BT48 7JL, Derry/Londonderry, Northern Ireland
E-mail: {smyth-e1, kj.curran, p.mckevitt, tf.lunney}@ulster.ac.uk

Abstract. A great deal of research has been conducted in the area of musical analysis. However, very little has been done with respect to utilising the MPEG-7 metadata standard to describe the structural content of music at a fine level of granularity. The aim of this paper is to provide an overview of the field of structural musical analysis and to present an architecture for the AMUSED system for automated structural analysis and description using the MPEG-7 standard.

1. Introduction

A great deal of research has been conducted in the area of musical analysis, both non-automated and automated, and some with particular emphasis on musical structure. However, very little has been done with respect to using the relatively new MPEG-7 standard as a basis for the description of hierarchical musical structure. MPEG-7 is particularly useful as it produces a standardised output which can potentially be used within other systems. We are developing a system called AMUSED (Automated MUsical StructurE Description) which performs real-time structural analysis using the MPEG-7 standard as a basis for the identification and the subsequent description of musical structure. Output from AMUSED will be in the form of an MPEG-7 compliant XML file. This output will be useful to a number of applications which seek to recognise musical structure, such as interactive music performance, visual based music applications or error concealment modules within streaming media algorithms, among other applications. This paper aims to provide an overview of previous research in the field of musical analysis and the use of XML in music notation, leading up to an introduction to the MPEG-7 standard and presentation of the proposed Automated MUsical StructurE Description (AMUSED) system architecture for identification and description of musical structure using MPEG-7 Audio.

2. Automated Music Structure Analysis

The automated discovery of patterns within music is an important problem within computational music analysis. Patterns emerge from repetitions within the music itself, and these patterns and repetitions are generally a good indication to musical structure. Repetitions can range in size from simple repeating note sequences, to recurring sections or phrases within a musical piece. In much the same way, structure

can be defined at differing levels of granularity based on these patterns and repetitions; from high level song form down to the single note level, if required. In addition, each of these structures in turn can be interrelated and higher level structures can contain complex organization within themselves.

There have been many techniques developed to discover patterns and structure within music which have resulted in a number of varying and sometimes complex approaches. To outline them all here would be a lengthy process, therefore only a brief overview of a selection of these approaches will be provided. Patterns have been extracted in variety of ways; using multi-dimensional datasets [19], a String-Joining approach [13], matrices [1], multiple viewpoints and a suffix tree data structure [6]. Some even allow the discovery of patterns in retrograde and/or inversion [21], and some simply extracts patterns from polyphonic MIDI sequences [20]. Perhaps the most novel approach is that in [5]; they use what they call a Spiral Array model for recognising and visualising tonal patterns based on pitch/time structures. Of the approaches mentioned thus far none actually deal with the identification of structure, only pattern recognition; however, this is a crucial element within the eventual identification of structure. The approaches mentioned next not only identify patterns, but also infer some level of coarse musical structure.

Foote and Cooper [10] use a somewhat original matrix style approach to music structure visualisation. They represent a piece of music as a 2-dimensional matrix. Within this matrix levels of self-similarity at any one point are indicated by light (similar) and dark (dissimilar) shading. High level structure can be inferred from the matrix in the form of intro, verse, chorus, etc. Dannenberg and Hu [7], Lu and Zhang [18], and others also utilise a matrix approach which result in the high level identification of structure. Finally, Vercoe and Chai [27], present a system that can automatically analyse the repetitive structure of musical signals; structure is discovered through repeating segments within a piece of music. The output from their system is, again, a relatively high level representation of structure in the form of ABBA, for example.

Most of these systems appear to work sufficiently well in identifying patterns and, where applicable, structure. However, they all suffer the inability to produce some level of universally standardised output, somewhat limiting their use outside their current project scope. In addition, those which do aim to identify structure do so at a relatively high, non-hierarchical level. Furthermore, the actual representation of music varied from system to system; e.g. strings, vectors, matrices and even spheres. They were also based on a variety of musical information; some on pitch, duration, interval, and others on spectral energy, rhythm and timbre. In addition, some required pre-processing before analysis could be performed, which is not ideal for use with real-time processing environments. It would be extremely valuable to have a system which analyses and represents all musical information in a standardised way; the output of which would be useful to any number of applications, even those outside

music theory. Wiggins, et al [28], go some way towards providing a standard representation, but it is not on par with the complex representation possible with XML-based notation. In addition, XML is already standardised and can be easily parsed for specific information because it is text-based.

3. XML and Music Notation

XML (eXtensible Markup Language) is a standardised, text-based method for describing the structure of data. It provides a very rich schema for defining complex documents and data structures. Text-based representations are ideal for music. Music does not lend itself to easy interpretation by computer, unlike language which can be easily represented and understood by computers because of the ANSI standard. Of particular benefit to music applications is the fact that XML allows support for multiple descriptions of the same data, and hierarchies are as fundamental to XML as they are to music notation.

There are a variety of XML-based markup projects in existence that are specifically aimed at music. XScore [11], Music Encoding Initiative (MEI) [22], MusicXML [26], and Music Content Markup Language (MCML) [25], to name a few. Although all are based on XML, they do have varying purposes. E.g. XScore is an application of XML for describing the musical score, and MCML was developed as part of another project (MIDILib) for content-based queries and navigation. While MEI strives to meet a broader range of music applications, it also avoids the confusion of other XML standards by using familiar names for elements and attributes, e.g. <note> and <chord>. MusicXML is perhaps the most mature endeavour for music encoding using XML. It has its roots in academia and has made its way into a number of commercial applications and has even been put forward for standardisation.

There are a few tools which harness the power of XML-based markup languages; as mentioned, the MIDILib project utilizes MCML; in addition, MiXA, VExPat and Sharpeye, among others, make use of MusicXML. MiXA is a web based musical annotation system [17], whereas VExPat is pattern extraction system which uses the MusicXML representation of music as a basis for analysis [24]. Sharpeye is a music reader which converts a scanned image of printed music into a MIDI, NIFF or MusicXML file [16]. There do not appear to be any XML applications which deal explicitly with the automatic identification and representation of structure within music.

A potential drawback is that most current DTDs restrict themselves to Common Western Music Notation (CWMN), with some including tablature. MusicXML, e.g. was designed to represent musical scores and sheet music, specifically common western musical notation from the 17th century onwards [26]. MPEG-7 is a further XML-based standard, but is more general in its approach; it is not based on the representation of sheet music or scores, but can deal with the representation of music directly from the audio file; in addition it can also deal with spoken content. Analysis of the sound file directly frees the representation somewhat from the restrictions of a

specific notational approach, like CWMN. Furthermore, it allows for significantly more detailed information to be stored about a particular piece of music.

4. Overview of MPEG-7 Audio

The MPEG-7 standard currently consists of 8 parts, part 4 of which deals specifically with the description of audio data; formally recognised as standard 15938-4. The main MPEG-7 elements are Descriptors (D), Description Schemes (DS), and a Description Definition Language (DDL). Ds are intended to describe low-level audio features; they are the building blocks of the system. DSs are designed to describe higher level audiovisual features. DSs produce more complex descriptions by combining multiple Ds and DSs and declaring relationships among the description components. The DDL provides the descriptive foundations through which users can create their own Ds and DSs [23].

The Audio Framework tools are applicable to the description of general audio; a graphical representation of the Framework is provided in Fig. 1. The generic Audio Framework contains low-level descriptors designed to provide a basis for the construction of higher level audio description schemes. The Low-Level Descriptors (LLDs) permit the description of an audio signal's spectral and temporal features.

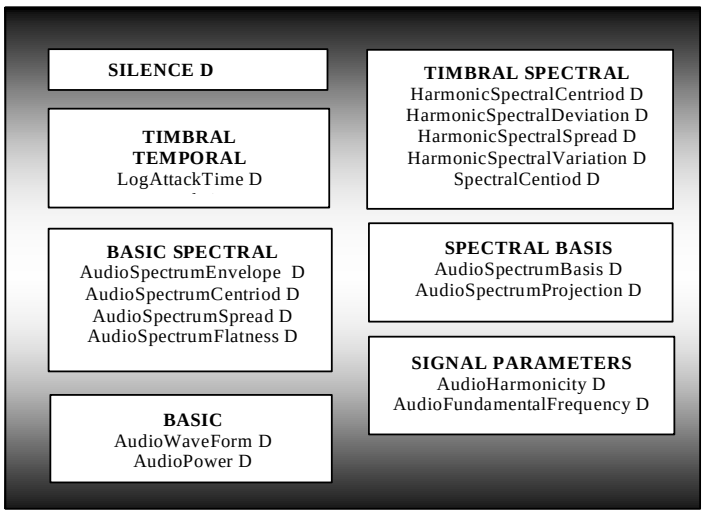


Fig. 1. Audio Framework [14]

In the first version of the standard there were seventeen LLDs for general use in a variety of applications. Since 2004, there has been an extension to the original standard (Amendment 1) to include additional Ds for such things as background noise level, balance, bandwidth, etc, with proposals for a second amendment to include Ds

to describe audio intensity, rhythmic patterns, chord patterns, and so forth [15], [12]. In addition to the LLDs there are five general sets of high-level audio description tools, which aim to encompass some application areas, such as sound recognition, musical instrument timbre, spoken content, melodic contour and melody. These specialised tools may be used in conjunction with the other tools within the standard. The high-level tools provide both functionality and also serve as examples of how to use the low-level framework [14].

MPEG-7 Ds and DSs have been successfully utilised by a number of projects over the last few years in a variety of application areas [2], [4], [8], [9]. These can be broadly categorised into one or more of the following areas: Query-By-Example (QBE), fingerprinting for audio identification, indexing and archiving, MPEG-7 authoring tools, audio analysis, audio classification, and Digital Rights Management (DRM). Musicstructure.com is perhaps the only existing MPEG-7 based application which deals solely with the analysis and representation of musical structure, but again this system only performs partitioning into a relatively high level structure [3]. MPEG-7 holds great promise in relation to the useful description of audio. It will have many applications, both within music theory and within information search and retrieval, as well as many as yet unknown application areas.

5. AMUSED System Architecture

The Automated MUsical StructurE Description (AMUSED) project seeks to overcome the inadequacies identified and provide a novel approach for the automated extraction, analysis and description of digital audio structural characteristics. It aims to develop a Java-based real-time architecture which will take digital musical audio (in most common formats) as input and produce standardised MPEG-7 compliant descriptions of musical structure as output. A diagram representing an overview of the proposed system architecture is shown in Fig. 2. The main focus will be on the accurate description of musical structure, based on within-song similarity, at a fine enough granularity to make it universally useful to all manner of applications; e.g. use within a visual based music application or an error concealment module within a streaming media application, or for use within music theory or musical analysis. Since the MPEG-7 output file is in a standardised format (XML) it creates the possibility of any system so designed being able to hook into and use it for further processing.

A digital audio signal is received into the system as a data stream. Spectral analysis is performed on the incoming data stream to accumulate low level information about the audio signal; signal power, spectrum spread, fundamental frequency, etc. This information is ‘extracted’ from the signal and described using a selection of the MPEG-7 Audio LLDs. Extracting all the Low-Level Descriptors (LLDs) would be a wasteful use of processing resources, therefore, a careful selection will be made with the project objectives in mind. Only those Descriptors which will be useful in pattern discovery, comparison and structural analysis will be chosen, as well as those with the ability to, thereafter, accurately describe structure. For example, the *SoundModelStateHistogram* is used to compare sound segments using histograms of

their state activation patterns, thus it will be useful for self-similarity comparisons, segmentation and structural analysis. The *SoundModelStatePath* could be a good candidate for modelling audio data at a micro level. The series of states could be used as 'markers' within an audio stream to identify particular elements of interest. It is proposed that perhaps a speciality Description Scheme be constructed from a choice of Descriptors in part 5 of the MPEG-7 standard. Spectral representations have the potential for accurately identifying lower level structure due to their relationship with the actual power of the signal. The power of the signal is the lowest level representation of audio therefore any descriptor which corresponds to this will demonstrate some form of low level structure over time.

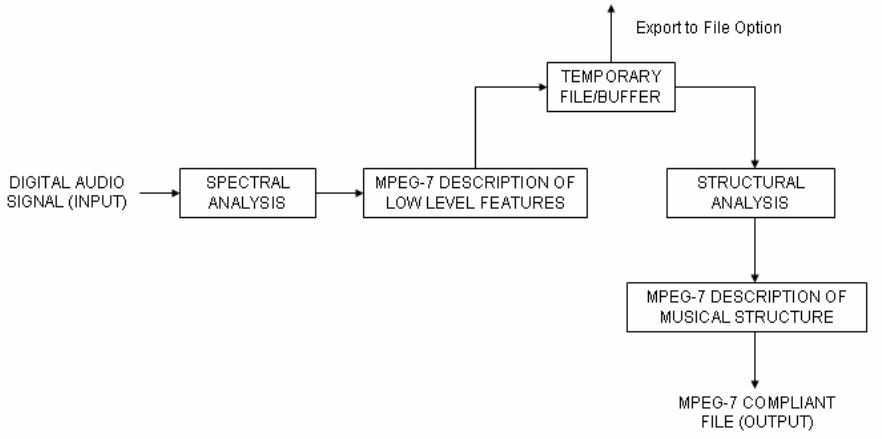


Fig. 2. Overview of Proposed System Architecture

Obviously some type of buffer/temporary file will be required to store the accumulative information produced by the Low Level Description module prior to their analysis by the structural analysis module. This file will be held in working memory and will also have the option of being exported as an XML file at this stage. Following on from the Low Level Description module, the structural analysis module performs further analysis to identify patterns, repetition and regions of self-similarity within the information contained within the extracted LLDs. This information will then used as a basis for the construction of a hierarchical model of musical structure, which is consequently described by the MPEG-7 structural description module. The final output from the system is in the form of an MPEG-7 compliant XML file, which can potentially be used within other systems.

6. Conclusion

This paper provided a review of previous research and an overview of the MPEG-7 standard, together with a brief synopsis of the proposed system architecture for the

AMUSED system. Many of the systems reviewed lack a standardised output and are therefore not particularly useful outside their own project scope. The use of XML for music notation is a step in the right direction; however, the projects outlined did not capture anything that significant about the music. E.g. XScore describes the written musical score which is really only useful for archiving and data exchange. Built on top of the XML base is the MPEG-7 standard. MPEG-7 offers a standardised scheme for the description of audio, among other media. Some applications are already in existence which utilise MPEG-7; however, very few offer the automated description of musical audio structure at a fine level of granularity. Indeed, most are QBE-style applications. The AMUSED system aims to overcome the inadequacies identified and provide a novel system for the automated extraction, analysis and description of digital audio structure and produce standardised, MPEG-7 output.

References

1. Aucouturier, J., and Sandler, M.. *Finding Repeating Patterns in Acoustic Musical Signals*, AES, 22nd International Conference on Virtual Synthetic and Entertainment Audio, Espoo, Finland, (2002), 145-152
2. Batke, J., Eisenberg, G., Weishaupt, P., Sikora, T. *A Query By Humming System using MPEG-7 Descriptors*, AES Convention Paper 6137, AES 116th Convention, Berlin, Germany, (2004)
3. Casey, M. Musicstructure.com, http://www.musicstructure.com/intro.html
4. Celma, O., Gomez, E., Janer, J., Gouyon, F., Herrera, P., Garcia, D. *Tools for content-based retrieval of audio using MPEG-7: the SPOffline and the MD Tools*, AES 25th International Conference, London, (2004)
5. Chew, E. *MuSA: Music Information Processing*, Proceedings of the 2nd International Conference on Music and Artificial Intelligence, Edinburgh, Scotland, (2002), 18-31
6. Conklin, D. and Anagnostopoulou, C. *Representation and Discovery of Multiple Viewpoint Patterns*, International Journal of New Music Research, Vol. 24, No. 1, (2001), 51–73
7. Dannenberg, R. and Hu, N. *Pattern Discovery Techniques for Music Audio*. ISMIR 2002 - 3rd International Conference on Music Information Retrieval, IRCAM – Centre Pompidou Paris, France, (2002)
8. Dumouchel, P., *MPEG-7 Audiovisual Document Indexing System (MADIS*), Testbed Development, RISQ 2003, 14th Edition, Annual Event of the Network of Scientific Information of Quebec (RISQ), Annual Workshop of CANARIE, Montreal, Canada, (2003)
9. Eisenberg, G., Batke, J., Sikora, T. *BeatBank – An MPEG-7 Compliant Query by Tapping System*, AES Convention Paper 6136, AES 116th Convention, Berlin, Germany, (2004)
10. Foote, J. and Cooper, M. *Visualizing Musical Structure and Rhythm via Self-Similarity*. International Conference Computer Music, Habana, Cuba, (2001), 140-148
11. Grigaitis, R. eXtensible Score Language (XScore) 0.01. http://grigaitis.net/xscore
12. Gruhne, M. *MPEG-7 Audio.* Workshop Music Network, Barcelona, Spain, (2004)
13. Hsu, J., and Liu, C. *Discovering Nontrivial Repeating Patterns in Music Data*. IEEE Transactions on Multimedia, Vol. 3, No. 3, (2001), 43-52
14. ISO/IEC. *Information Technology – Multimedia Content Description Interface – Part 4: Audio.* ISO-IEC JTC 1/SC 29/WG 11, ISO-IEC FDIS 15938-4:2001(E), http://projekt.rz.tu-ilemnau.de/~kfn/seminar13/MPEG-7-Dokument.pdf, (2001)
15. ISO/IEC. *Information Technology – Multimedia Content Description Interface – Part4: Audio*, Amendment1:Audio Extensions, ISO/IEC JTC1/SC29/WG11 N4769, http://www.itscj.ipsj.or.jp/sc29/open/29view/29n4854t.doc, (2002)
16. Jones, G., (2005), *SharpEye Music Reader*, http://www.visiv.co.uk/about.htm

17. Kaji, K., and Nagao, K., *MiXA: A Musical Annotation System*, Proceedings of 3rd International Semantic Web Conference, Hiroshima, Japan, (2004)
18. Lu, L. Wang, M., and Zhang, H. *Repeating Pattern Discovery and Structure Analysis from Acoustic Music Data*, Proc. of IEEE International Conference on Multimedia and Expo (ICME '04), Taipei, Taiwain, (2004)
19. Meredith, D., Lemstrom, K., and Wiggins, G. *Algorithms for discovering repeated patterns in multi-dimensional representations of polyphonic music*. Journal of New Music Research, Vol. 28, No. 4, (2003), 334–350
20. Meudic, B. *Automatic pattern extraction from polyphonic MIDI files*, *Les* Journees d'Informatique Musicale, 10th Edition, L'ecole Nationale De Musique Du Pays De Montbeliard, (2003)
21. Ren, X., Smith, L., and Medina, R. *Discovery of Retrograde and Inverted Themes for Indexing Musical Scores*, ACM/IEEE Joint Conference on (JCDL'04), Tucson, AZ, USA**,** (2004), 252-253
22. Roland, P. *The Music Encoding Initiative (MEI)*, Proceedings of 1st International Conference Musical Applications using XML, Milan: State University of Milan, (2002), 55–59
23. Salembier, P., and Smith, J. *MPEG-7 Multimedia Description Schemes*, IEEE Transactions Circuits and Systems for Video Technology, Vol. 11, No. 6, (2001), 65-73
24. Satana, H., Dahia, E. L., and Ramalho, G. *VExPat: An Analysis Tool for the Discovery of Musical Patterns*, Proceedings of IX Brazilian Symposium on Computer Music, Campinas, SP, (2003)
25. Schimmelpfenning, J and Kurth, F. *MCML: Music Contents Markup Language*, Proceedings of International Symposium of Music Information Retrieval, Plymouth, MA, USA, (2000)
26. Stewart, D. *XML for Music*, http://emusician.com/mag/desktop/emusic_xml-music/
27. Vercoe, B., and Chai, W. *Structural Analysis of Musical Signals for Indexing and Thumbnailing*, ACM/IEEE Joint Conference on Digital Libraries, New York, (2003), 27– 34
28. Wiggins, G., Harris, M., and Smaill, A. *Representing Music for Analysis and Composition*, Proceedings of the 2nd IJCAI AI/Music Workshop, (1989)

Scheduling with Uncertain Release Dates

Christine Wei Wu[1], Kenneth N. Brown[1], and J. Christopher Beck[2]

[1] Cork Constraint Computation Center,
Computer Science, University College Cork, Cork, Ireland
{cww1, k.brown}@cs.ucc.ie
[2] Toronto Intelligent Decision Engineering Laboratory,
Department of Mechanical and Industrial Engineering,
University of Toronto, Canada.
jcb@mie.utoronto.ca

Abstract. The aim of the paper is to examine whether we can adapt Bent and van Hentenryck's combination methods to our problem, and to determine how effective and significant the methods are. In particular, we develop four ways of applying Bent and Van Hentenryck's consensus approach[3] to our problem. In addition, since we assume the uncertainty distribution is known, we propose a probabilistic sampling method to handle lead-time uncertainty. That is we use this knowledge to select samples, and associate with them weights corresponding to their probability. This method allows us to generate fewer samples but have a more accurate model of future scenarios.

1 Introduction

The question of how to schedule tasks efficiently is an important problem for many applications. There are many off-line optimization algorithms for generating good schedules to satisfy different optimization criteria. However, these algorithms almost always assume a fully specified problem that doesn't change. The real world is rarely that well behaved. Many problems have uncertain specifications, and are subject to change, but still an immediate solution is needed at the outset. In particular, when the company doing the scheduling is part of a bigger supply chain, one of the main causes of uncertainty is the late arrival of raw material or components from a supplier (known as lead-time uncertainty). If the supplies have not arrived, then the job cannot be released for execution. The question then becomes how to construct an initial schedule that is robust to this variability in the release dates.

A number of approaches have been proposed to handle uncertain scheduling problems. In particular, Bent and van Hentenryck [2–4] have studied on-line packet scheduling and vehicle routing, where packet values and delivery requests are uncertain at the start. They show that sampling from a black-box distribution, solving each sample and combining the results using one of three methods (i.e. expectation, consensus or regret), does improve decision making. In this paper, we consider a different problem - job shop scheduling, where the jobs are

sequences of tasks with constraints between them, where the jobs are known but have uncertain release dates, and where we have knowledge of the uncertainty distribution. The aim of the paper is to examine whether we can adapt Bent and van Hentenryck's combination methods to our problem, and to determine how effective and significant the methods are. Since the uncertainty distribution is known, we propose a probabilistic sampling method to handle lead-time uncertainty. That is we use this knowledge to select samples, and associate with them weights corresponding to their probability. This method allows us to generate fewer samples but have a more accurate model of future scenarios. Also, we consider four ways of applying Bent and Van Hentenryck's consensus approach[3] to our problem.

The paper is organized as follows. Section 2 briefly describes Bent and Van Hentenryck's online stochastic algorithms and the problems (i.e. packet scheduling and multiple vehicle routing) they experimented on. Section 3 presents the uncertain release date job-scheduling problem, and highlights the differences from vehicle routing. Section 4 defines probabilistic sampling in general. Section 5 and 6 present various ways of adapting the combination methods to our problem. Section 7 presents experimental results on instances of a job-scheduling problem. Finally, section 8 concludes the paper.

2 Background

Bent and van Hentenryck [2–4] considered two online resource allocation problems (packet scheduling and vehicle routing), where new tasks arrive every time step, and an immediate decision must be made whether to accept or reject them. They assume that they can obtain samples of future arrival patterns, and they propose a number of methods for using these samples to compute the best immediate decision. *Expectation* considers each possible decision, computes the reward over all the samples, and chooses the decision with the highest reward. *Consensus* considers each sample in turn, computes the optimal schedule with respect to that sample, and then selects the decision that appears most often in the optimal schedules of the samples. *Regret* is similar to consensus, but as well as computing the optimal schedules also computes the loss of reward for each other possible decision, and then chooses the decision that has the lowest total loss. Regret relies on having an accurate estimate of the cost of completing a schedule, to avoid the overhead of re-optimizing. They show that although expectation may give the best results, it is infeasible since it requires too much time to make its decisions. Consensus is much faster, since it has significantly fewer optimizations to consider. Regret is a more robust method, approaching expectation when there is sufficient computation time, and approaching consensus when time is limited. For the vehicle routing problems, the optimization time is significant, so to try to reduce the number of optimizations, they maintain a set of possible plans. Each decision is then restricted to extending the possible plans - i.e. when a new task arrives, it must be slotted into an existing delivery schedule, rather than completely re-optimizing the schedule. Some of the deliv-

ery plans will become infeasible, and are replaced by new variants of the feasible plans. Similarly, they treat each vehicle in turn, and do not attempt to reallocate the customer requests between vehicles.

There has been some previous research reasoning directly about uncertainty in combinatorial problems like scheduling [1, 5–9], but this approach is not considered further here.

3 The problem

We consider the problem of scheduling jobs where some of the jobs have uncertain release dates. We have N jobs, each with K tasks. There are M machines, and each machine can only process one task at a time. The tasks of a job have to be processed in a given order, and each task must be processed on a specified machine for a known duration. Once a machine has started processing a task, it must complete it. We assume P of the jobs have an uncertain release date (arrival time) - i.e. there is uncertainty about the date on which the first task of the job can be started. We assume that the actual start date is characterized by a known probability distribution. The remaining $N - P$ jobs have a release date of 0 - i.e. they can be started immediately. The aim is to construct and execute an initial schedule for the $N - P$ jobs, so that online during execution, the schedule can be adapted to include the arriving P jobs, while minimizing *makespan* (i.e. the total time until the last job is completed). For this paper, as an initial test of our methods, we set $P = 1$, $M = N - 1$, and we assume each job must visit each machine. The uncertain job arrives no earlier than t_0, and no later than t_n. That is, we have a job-shop scheduling problem, with the addition of one extra job with an uncertain release date.

Note that there are some significant differences from the packet scheduling and vehicle routing problems considered by Bent and van Hentenryck. In their problems, each job consisted of a single task, whereas in the JSP each job has multiple tasks with precedence constraints between them. That means for the JSP that there are more constraints between the tasks and the resources, so each decision will have more consequences further down the line (i.e. a decision to delay the start of one job will have an impact on all resources). Secondly, their problems are focused on maximizing the reward obtained for tasks completed (e.g. number of customers served or packets scheduled) - at each time step, they have a probability distribution for what tasks will arrive, and they must choose immediately whether to accept or reject the tasks. In the JSP, we know the jobs that we will be required to process, and must process them all, but the aim is to minimize the finishing time of the last task. We also consider an initial lead time before time 0, which we may spend trying to optimize the schedules.

4 Probabilistic sampling

Bent and van Hentenryck assumed that the arrival process for their tasks could be modelled by a black-box simulator. They had no knowledge of the distribution

inside the simulator, but could ask for samples to be generated, and they used those samples to make their decisions. In this work, we assume that we do have access to the probability distribution from some historical data analysis. We can then use that distribution directly to select samples, and we can give them a weight corresponding to their probability. This should allow us to generate fewer samples, and to have a more accurate picture of the consequences of our decisions. Further, if we have limited time, we can generate samples in the order of probability, and thus do anytime reasoning. For example, suppose the arrival time distribution of a job is $Prob(\text{arrive at } (5, 6, 7, 8, 9)) = (0.1, 0.2, 0.4, 0.2, 0.1)$. There are five possible arrival times, so we only need to generate 5 samples - one for each arrival time. We can then compute the best decision for each sample by whatever method we choose, but then we combine them by weighting the samples by their probability. Thus when combining the decisions, a decision that is made for the sample where the job arrives at time 7 is given twice the weight of the decision made for the sample with arrival time 8. In the sections that follow, we discuss how the expectation and consensus methods can be adapted to the JSP with uncertain release dates, for use with probabilistic sampling.

5 Probabilistic Expectation

The idea of expectation is to find the decision that produces the lowest expected makespan. By assumption, nothing is going to change in the problem until time t_0. Therefore, we can consider a decision to be a schedule of tasks which start before t_0. For every feasible schedule up to t_n of the released jobs, for each possible arrival time, fix all the activities that start before that time, and run the optimizer on all the other activities, including the new job. This gives us a makespan for each possible partial schedule for each arrival time. We then obtain the expected makespan for the partial schedule by taking the weighted average over the arrival times of the makespans, using the probabilities as weights.

```
1 for each feasible solution s up to tn
2    for each possible arrival time a
3       fix all tasks starting before a
4       optimize schedule for remaining tasks to get s'
5       obtain the makespan
6       f(s) += makespan(s') * prob(a)
7 return s with minimum f(s)
```

The optimization is called $|S| \times |A|$ times, where $|S|$ is the number of feasible initial schedules and $|A|$ is the number of possible arrival times. It also requires us to generate the $|S|$ initial feasible schedules.

6 Probabilistic Consensus:

The aim of the consensus approach is to reduce the number of optimizations required, so rather than starting with all possible decisions, we start with all

possible arrival times. We compute the optimal schedule with respect to each possible arrival time, then construct 'consensus' early decision from those optimums. There are a number of different ways of defining consensus and early decisions, and we examine four below.

6.1 *C-schedule*: Consensus feasible schedule to t_0

The simplest method is the dual of expectation. For each arrival time, compute the optimal schedule up to t_0. Weight each schedule by the probability of the arrival time that generated it, and select the schedule with the highest weighted occurrence. For problems with a long gap from 0 to t_0, it is likely that each schedule will be different, and so this may default to selecting the optimal schedule for the most probable arrival time. This algorithm requires $|A|$ optimizations.

```
1 for each arrived time a
2    compute optimal schedule s wrt. a
3    f(s) += prob(a)
4 return s with maximum f(s)
```

6.2 *C-resource*: Consensus independent resource schedules up to t_0

Since the algorithm in 6.1 may not result in any schedule appearing twice, we could consider treating each resource as independent. For each arrival time, we will thus compute the optimal schedule as before, but now we will maintain a weighted count of the different tasks that start on a resource at each time step. To get the consensus decision, we will then examine each resource independently, and get a series of consensus schedules for each resource. This may introduce further problems, though, since the resource-independent consensus decisions may not be consistent - therefore, at each time step when building the schedule, we have to select the decision with the highest weighted count that is consistent with previous decisions. This algorithm also requires $|A|$ optimizations.

```
1  for each possible arrival time a
2     compute optimal schedule wrt a
3     for each resource M
4        for each time step t
5            if act starts at t
6               M[act][t] += prob(a)
7  for each time step t
8     for each resource M
9         find consistent act with highest value in M
10        schedule act on M at t
```

6.3 *C-point*: Consensus independent-resource schedules by time step

Although the algorithm above is likely to make more consensual decisions, those decisions are getting steadily less informed as the time steps increase (since the

schedules on which they were based may have had different task allocations). Therefore we developed an approach which breaks decisions down, and considers each resource and each time step independently. That is, the consensus decision for each resource independently is computed for time 0 and allocated, the optimal schedules are then recomputed from time 1 onwards to produce new consensus decisions for time 1, and so on. This algorithm requires up to $|t_0| * |A| + |A - 1| + ... + 1$ optimizations.

```
1  for each time step t before to
2     for each possible arrival time a
3        compute optimal schedule s from t wrt a
4        for each resource M
5           if act starts at t
6              M[act] +=  prob(a)
7     for each resource M
8        find consistent act with highest value in M
9        schedule act on M
10 compute and execute optimal completion when job arrives
```

6.4 *C-tuple*: Consensus schedules by time step

We can also consider an alternative to *C-point* (6.3) in which we do not treat the resources independently - that is, each decision is a tuple of allocations of tasks to machines at a single time-step (i.e. the size of a tuple is equal to the number of machines). This algorithm requires the same number of optimizations as *C-point*.

```
1 for each time step t before to
2    for each possible arrival time a
3        compute optimal schedule s from t onwards
4        find highest weighted tuple of task allocations
5        schedule the tuple of tasks
6 compute and execute optimal schedule when job arrives
```

6.5 Discussion

The effect of the different definitions of consensus is shown in Figure 1. We have three resources (M_0, M_1, and M_2), three jobs ready to start (A, B and C), and another job due to arrive at times 5, 6, 7 or 8, with probabilities 0.4, 0.3, 0.15 and 0.15 respectively. We have four possible arrival times, so we generate four possible samples, and compute the optimal schedule for each one. These optimal schedules (up to time 4) are shown in the figure. Each schedule is different, so *C-schedule* selects the one generated by the highest weighted sample. *C-resource* sums the weights for each new task over each sample independently by resource,

and generates a schedule that is different from any of the others. *C-point* treats the resources independently, and chooses a task with the highest weight (on each resource) to execute at time step 1 (and re-optimize for subsequent time steps). Finally, *C-tuple* selects the tuple of task allocations at time step 1 with the highest weight (and will also re-optimize).

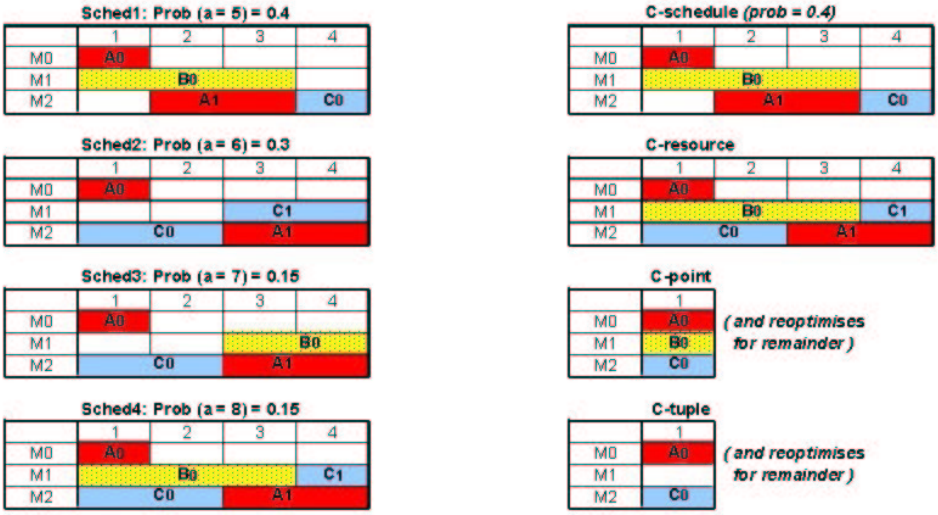


Fig. 1. LHS is four optimal schedules for each sample (up to time 4). RHS is the decisions made by four kinds of consensus at time 1.

All four methods conduct off-line scheduling to compute the best partial schedule up to the arrival of the new job, and then react to the arrival on-line. The off-line parts of *C-schedule* and *C-resource* both return a single partial schedule, and require all their off-line computation to be carried out before the first task is executed. *C-point* and *C-tuple* produce their partial schedule iteratively, updating their decisions each time step, even though nothing has changed in the problem. They require more optimizations, but spread them over the time period from 0 to t_0.

7 Experimental Results

We have implemented *expectation*, *C-schedule*, *C-point* and *C-tuple* in ILOG Scheduler 6.0, a C++ library for constraint-based scheduling.

Since *C-point* and *C-tuple* update their decisions at each time step, after constructing an off-line schedule up to the earliest arrival t_0, they can also act as online methods to react to the new job's arrival. However, *expectation* and *C-schedule* are only off-line optimization methods and we choose *C-tuple* as

the online reacting method for both. We also implemented two pure reaction methods (i.e. no lookahead) - *re-optimize* and *right-shift*. Both of the methods find a best schedule for the released jobs and execute the schedule. When a new job arrives at t, *re-optimize* optimizes the unscheduled tasks and the new job after t, while *right-shift* inserts the new job into the original schedule by shifting relevant tasks to the right.

We have tested the algorithms on JSP benchmark problems [10]. At first, we tried to use a 10 by 5 benchmark with the last job having an uncertain release date, but their results showed little difference with different approaches. Then, we adjusted the problems to 6 by 5 (with the last job having the uncertain lead-time) and scaled the durations of tasks, in order to make the problems more tightly constrained (i.e. more sensitive to the uncertainty).

Table 1 shows the results for one typical problem (la03) using two different distributions for the arrival time. *Absolute optimal* is an optimized solution knowing the release date of the new job (i.e. no uncertainty). The figures beside each method are the makespan of schedules generated by that method. The figures in bold indicate the schedules with same makespan as absolute optimal solutions. For example, in distribution 1 our four methods all produce very good schedules at the most probable (probability 0.6) arrival time 30, unlike those two pure reaction methods. Table 2 is the mean relative error (weighted by arrival probability) of each method with respect to absolute optimal.

Table 1. The experimental results on scheduling 6 jobs with one job having uncertain arrival time. It shows the makespan on all possible arrivals wrt. two probability distributions.

	Distribution 1			Distribution 2		
Arrival time	18	24	30	54	60	66
Probability	0.1	0.3	0.6	0.6	0.3	0.1
Abs Optimal	**93**	**96**	**100**	**121**	**127**	**133**
C-point	99	100	**100**	124	**127**	**133**
C-tuple	99	100	**100**	**121**	**127**	**133**
C-schedule + C-tuple	99	100	**100**	**121**	**127**	**133**
Expectation + C-tuple	94	**96**	**100**	**121**	**127**	**133**
Re-Opt	94	**96**	102	**121**	135	135
R-Shift	98	100	102	135	135	135

From Table 2, we can see that the four adapted methods all provide good results, not being more than 2% over the optimal. Because *expectation* must examine all possible initial schedules, it takes much longer time than the other methods - approximately 5 minutes as opposed to just over 1 second. However, when we look at the reactive methods, we see that the performance is not that

Table 2. This table shows the mean weighted relative errors (i.e. difference from absolute optimal) of using different methods in the two experiments in Table 1.

Method	Mean weighted relative error	
	Experiment 1	Experiment 2
C-point	1.02	1.01
C-tuple	1.02	1
C-schedule + C-tuple	1.02	1
Expectation + C-tuple	1.001	1
Re-Opt	1.02	1.02
R-Shift	1.03	1.09

significant, since it is possible to get very close to the optimal without reasoning about the uncertain events in advance. We believe that this is a feature of the problems we have looked at, in that the resource contention is generally not high, and so they are not sensitive to changes in the release date of a single job.

8 Conclusion and Future work

We considered various ways of adapting Bent and Van Hentenryck methods to the uncertain release date job-scheduling problem. We proposed the probabilistic sampling idea that allows us to use probability distribution to select samples, and give them a weight corresponding to their probability. Then we use the samples and their weights to make decisions. This gives us a more accurate picture of the consequences of decisions by generating fewer samples. Also, we devised probabilistic *expectation*, *C-schedule*, *C-point* and *C-tuple* methods. We experimented with these approaches on scaled benchmark problems, and showed the methods can solve uncertain release date job shop scheduling problems effectively. However, the benchmark problems appear to be too loosely constrained for us to draw any strong conclusions about the methods, since a reactive optimizer with no model of the future is also able to perform well on these problems.

For future work, we intend to investigate what properties of the problem make it sensitive to the uncertainty. Further, we intend to adapt the *regret*[2] method to the job-scheduling problem and also to consider breaking the expectation method down by time step as we have done for consensus.

9 Acknowledgments

This work is funded by Enterprise Ireland (SC/2003/81), and has received support from Science Foundation Ireland (00/PI.1/C075) and ILOG, S.A.

References

1. C. J. Beck, N. Wilson: Job Shop Scheduling with Probabilistic Durations. Proceedings of the Sixteenth European Conference on Artificial Intelligence (2004).
2. R. Bent, P. Van Hentenryck: Regret Only! Online Stochastic Optimization under Time Constraints. American Association for Artificial Intelligence (2004).
3. R. Bent, P. Van Hentenryck: The Value of Consensus in Online Stochastic Scheduling. Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling (2004).
4. R. Bent, P. Van Hentenryck: Online Stochastic and Robust Optimization. Ninth Asian Computing Science Conference Chiang Mai University (2004)
5. A. J. Davenport, C. Gefflot, J. C. Beck: Slack-based Techniques for Robust Schedules. Proceedings of the Sixth European Conference on Planning (2001).
6. Hélène Fargier, Jérŏme Lang, Thomas Schiex: Mixed constraint satisfaction: A framework for decision problems under incomplete knowledge. Proceedings American Association for Artificial Intelligence, pp175-180 (1996).
7. D. W. Fowler, K. N. Brown: Branching constraint satisfaction problems and Markov Decision Problems compared. Annals of Operations Research, Volume 118, Issue 1-4, pp85-100 (2003).
8. N. Yorke-Smith: Reliable Constraint Reasoning with Uncertain Data. PhD thesis, IC-Parc, Imperial College London (2004).
9. T. Walsh: Stochastic Constraint Programming. Proceedings of 15th European Conference on Artificial Intelligence (2002).
10. S. Lawrence: Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques (Supplement), Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania (1984).

Author Index

Author Index

Al Momani, B.M. 83
Beck, J.C. 203, 397
Black, M. 339
Bridge, D..... 29
Briggs, P. 181
Brown, K.N. 203, 397
Campbell, G.G. 299
Carthy, J. 115, 125
Cater, A. 105
Collier, R. 147, 359
Costello, F.J. 41
Cousineau, D. 61
Cummins, F. 51
Cummins, R. 137
Curran, D. 225
Curran, K. 73, 389
Delany, S.J. 257
Doherty, J. 73
Doran, W. 125
Dunnion, J 115, 147, 359
Fyfe, C. 245
Giguere, G. 61
Glass, DH. 191
Gros, C. 235
Hancox, P. 309
Healy, M. 257
Helie, S. 61
Hickey, R.J. 287
Hoare, C. 319
Howley, E. 329
Howley, T. 277
Kelleher, J. 159
Kelly, J.P..... 29
Kulkarni, P. 339
Lillis, D. 147, 359
Lunney, T. 299, 389

Ma, M 169
Madden, M.G. 93, 277
Maguire, P. 105
Mc Kevitt, P. 73, 169, 299, 389
McCarthy, K. 19
McClean, S.I. 83, 339
McGinty, L. 19, 379
McSherry, D. 9
Mohammed, N. 339
Moran, M. 349
Morrow, P.J. 83
Munroe, D.T. 93
Mur, A. 147, 359
Newman, E. 115
O'Connell, M.L. 277
O'Leary, C. 369
O'Riordan, C. 137, 225, 329, 349
Parr, G. 339
Pena, M. 245
Peng, L. 147, 359
Proulx, R. 61
Reilly, J. 379
Ryder, A.G. 277
Salamo, M. 379
Schneider, K.M. 267
Scotney, B. 339
Shackleton, M. 5
Sharkey, N. 3
Smyth, B 19, 181, 379
Smyth, E. 389
Sorensen, H. 319
Stokes, N. 115, 125
Toolan, F. 147, 359
Vidotto, A. 203
Wallace, R.J. 213
Wang, R. 125
Wu, C.W. 397
Zamolotskikh, A. 257