National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh

DEPARTMENT OF INFORMATION TECHNOLOGY

technical report NUIG-IT-121002

Non-Traditional Collaborative Filtering Techniques

J. Griffith (NUI, Galway)
C. O'Riordan (NUI, Galway)

Non-Traditional Collaborative Filtering Techniques

Josephine Griffith
Dept. of Information Technology
NUI, Galway
josephine.griffith@nuigalway.ie

Colm O'Riordan
Dept. of Information Technology
NUI, Galway
colmor@geminga.nuigalway.ie

Abstract

Collaborative filtering produces recommendations for some active user using the ratings of other users, where these users have similar preferences to the active user. It differs to traditional retrieval and filtering systems which return items to some user based on a comparison between the content contained in items (documents) and the content of a user query (information need).

Many authors have viewed collaborative filtering from a statistical point of view and numerous statistical techniques have been applied (including mean square difference, Pearson correlation, Spearman correlation and techniques from probability theory), with good success, to collaborative filtering domains. Early collaborative filtering systems, as well as many commercial systems today, use a statistical approach to calculate a correlation between users. However, a large body of work exists on the application of other techniques and approaches in a collaborative filtering domain, including Bayesian networks, dependency networks, aspect models and clustering. This paper reviews a number of these techniques.

1 Introduction

This paper reviews the area of collaborative filtering and describes some non-traditional collaborative filtering approaches that have been used to date.

Given a set of users, a set of items, and a set of ratings, collaborative filtering systems attempt to recommend items to users based on prior ratings of the users. The collaborative filtering system essentially automates the "word of mouth" process. The assumption upon which collaborative filtering is based is that human preferences are correlated and thus prediction is possible.

The problem space can be viewed as a matrix consisting of the ratings given by each user for items in a collection, i.e., the matrix consists of a set of ratings $u_{i,j}$, corresponding to the rating given by user i to an item j. Using this matrix, the aim of collaborative filtering is to predict the ratings of a particular user, i, for one or more items previously not rated by that user. The traditional collaborative filtering paradigm involves a centralised approach whereby users register with one particular system and provide ratings for items (explicitly, implicitly or both).

A number of issues arise with respect to collaborative filtering. These include, choosing the best method to predict preferences; evaluating the effectiveness of the predictions made; analysing the theoretical underpinnings of collaborative filtering techniques and justifying the assumptions made by the various collaborative filtering approaches as well as dealing with known collaborative filtering problems.

The paper outline is as follows: firstly an overview of collaborative filtering is given, together with a summary of the traditional approach to collaborative filtering. Section 3 details a number of "non-traditional" approaches. Conclusions are presented in Section 4.

2 History, Issues and Traditional Approaches

2.1 History

A number of researchers have investigated the construction of user models based on stereotypes whereby a stereotype is a collection of data which typifies a class of users. Rice [36] defines stereotypes as a means of making a large number of assumptions about a user based upon only a small number of observations, and she pioneered the use of stereotypes in the *Grundy* system for recommending books to users.

In general, user modelling has had a long history in many computer science domains. Traditionally, user models were created based on evidence from explicit user actions; there has been a gradual change in this approach and now the focus is on building user models using implicit information gleaned from the user's interaction with a system. User modelling techniques are used in the domain of information management where techniques are used to ascertain a given users information need with the aim of providing more relevant and personalised information to that user.

Goldberg [17] coined the term *Collaborative Filtering* and was the first to publish details of a collaborative filtering technique in his description of the *Tapestry* email filtering system. This system allowed users to annotate documents with their opinions of the documents. Users could specify mail filtering queries to select interesting documents based on the document content and the document annotations. The system was not automatic and relied on users to manually identify similar users and to select recommendations.

Resnick [35] introduced automated collaborative filtering for a system (*GroupLens*) which provided personalised predictions for Usenet articles. A neighbourhood-based algorithm, using Pearson Correlation to create neighbours, was used. Shardanand and Maes [38] developed a music recommender system (*Ringo*) using constrained Pearson correlation to calculate user correlations. Neighbourhoods were selected based on a fixed threshold. Predictions were generated based on a weighted average of ratings from all users in a neighbourhood. The Bellcore Video recommender [23] also used Pearson Correlation.

Collaborative filtering techniques have been successfully applied to several domains on the Internet, e.g. www.amazon.com, www.cdnow.com.

Recent advances within the field of collaborative filtering have focused on the application of clustering and data mining techniques and on the combination of content and collaborative filtering techniques.

2.2 Collaborative Filtering Issues and Metrics

A number of issues with respect to collaborative filtering exist including:

1. **Voting**: The data set may be populated using explicit ratings, implicit ratings or both. Explicit ratings can be obtained using a gauge set or by allowing the user to select the items which they will rate. Implicit ratings are obtained by inferences from user actions. Ideally, as many ratings as possible are required from each user and for this reason explicit ratings are usually not sufficient on their own as users are not willing to invest a large amount of their time in rating many items. Implicit ratings may not be as accurate as explicit ratings. Ratings can be recorded as absolute values (binary or within some range) or as a relative ordering of items.

2. **Data Set**: sparsity is a recognised problem with collaborative filtering data sets. Issues with respect to the data set include: the sparsity of the matrix and any assumptions that are made with respect to missing values and noise (and the validity of these assumptions and approaches); the size of the data set and any dimension reduction techniques that are performed and the effect these techniques have on results as well as efficiency considerations.

3. **Algorithms**: numerous collaborative filtering algorithms have been proposed and tested. These include model-based approaches, memory-based approaches, machine learning approaches, statistical and probabilistic approaches and list ranking approaches as well as approaches which combine content with collaborative information. Various algorithms have also been used as a pre-processing step.

4. **Efficiency**: due to the typically large size of collaborative filtering data sets, efficiency considerations are important and may mean that certain approaches are not viable. It is important that users receive recommendations from a system in a timely manner.

5. **Cold-start problem**: this problem occurs when a new user or a new item is added to the data set on which no, or very little, data exists.

6. **one-of-a-kind problem**: this problem occurs when there is some user who is not similar to any other users in the data set.

7. **Results**: recommendations can be presented to a user in a number of ways, the most common being to present one recommendation for a particular item or present a list of ranked recommendations, e.g. *top-N*.

The ability of a system to provide quality recommendations is the main measure of effectiveness used in collaborative filtering systems. Intermediate steps within a particular approach could also be evaluated, for example, the quality of the neighbourhood formed in nearest-neighbour approaches. The main metrics used to test the predictions produced are:

- *coverage*: a measure of the ability of the system to provide a recommendation on a given item.

- *accuracy*: a measure of the correctness of the recommendations generated by the system.

Coverage is usually computed as a percentage for the items for which the system was able to provide a recommendation. Accuracy metrics may be broken down into statistical accuracy metrics and decision support accuracy metrics. **Statistical accuracy metrics** are usually calculated by comparing the ratings generated by the system to user-provided ratings. The accuracy is usually presented as the mean absolute error between ratings and predictions [38]. Root mean squared error and correlation measures between ratings and predictions can also be used. Sarwar et al. [8] claim that these metrics track each other closely. **Decision support accuracy metrics** provide a measure of the ability of the system in a decision support environment. Typically, the value of the rating is not that important—it is more important to know if the rating is a good or a bad rating. Metrics which can be used include: ROC sensitivity which is measured by plotting *sensitivity* against ($1 -$ *specificity*). *Sensitivity* is the probability of a good item being returned by the system as such, *specificity* is the probability of a *poor* item being accurately identified. Weighted errors and reversal rate may also be considered.

2.3 Traditional Approaches: Memory-based Techniques

Memory-based techniques are the most commonly used approach in collaborative filtering. The title "memory-based" is attributed to these set of techniques due to the fact that the dataset of user preferences must be kept in memory as it is accessed for each prediction/recommendation made. The majority of memory-based techniques are user-user techniques though some item-item techniques have also been investigated. Griffith and O'Riordan [20] give a detailed overview and empirical analysis of memory-based techniques.

Memory-based techniques provide a simple and intuitive approach to collaborative filtering and are widely used. Disadvantages include:

- Recommendations are slow to respond to changes in a user profile.
- They could be seen to be theoretically weak in that the choice of threshold for neighbourhood selection affects the recommendation results but it is difficult to ascertain the optimal threshold value. Also the similarity is based only on the known features in the dataset.
- The correlation between two user profiles can only be computed based on items that both users have rated.
- The correlation approach induces one global model of similarities between users, rather than separate models for classes of ratings (e.g., positive vs negative ratings).
- To overcome problems with the sparsity of the data set and the size of the data set, some form of preprocessing may be required. Singular value decomposition (SVD) is one possible approach. Using SVD, the initial matrix can be decomposed into 3 matrices: $A = U\sigma V^T$ where U and V are composed of orthonormal vectors that define left and right singular values of A. σ is a diagonal matrix. The highest k singular values are maintained together with the corresponding rows and columns in U and V^T. From these three reduced matrices, A' an approximation of the original matrix A can be derived. A detailed description is available in [4, 12].
- There are scalability issues as the dataset increases.

3 Non-Traditional Approaches

3.1 Description

Typical or "traditional" collaborative filtering algorithms use standard statistical measures to calculate the similarity between users (mean square difference, pearson correlation etc.). Such techniques, along with other similarity measures such as vector similarity, belong to the catrgory of memory-based approaches.

Within the grouping of "non-traditional" approaches a large set of approaches have been investigated. Probability theory has been used [33] to calculate the probability that users are of the same type.

Other algorithms construct a model of underlying user preferences from which predictions are inferred. Examples include Bayesian models [6]; dependency networks [22], aspect models [24, 25], horting [1], clustering models [39] and models of how people rate items [33].

Collaborative filtering has also been cast as a machine learning problem [3, 5, 31, 15], as a list-ranking problem [15] and as a data mining problem [1, 30]).

SVD has been used to improve scalability by dimensionality reduction [5, 37]. This section gives an overview of these many "non-traditional" approaches to collaborative filtering.

3.2 Probability Theory

Pennock et al. [33] propose **personality types** as an approach to collaborative filtering. A personality type is encoded as a vector of the user's "true" ratings for titles in the database. Given the user's known ratings of items, the probability of other users having the same personality type is computed; then the probability that the user will like some new item is computed. This approach therefore tries to acquire a model of users and groups of users to give an initial idea of neighbourhoods. Traditional collaborative filtering techniques could then be used. Thus it could be seen as a form of pre-processing. However, from a practical point of view, users may not be willing to give explicit ratings.

3.3 Model-based Approaches

Breese et al. [6] describe an approach where **Bayesian networks** can be used to create a model based on a training set where each node corresponds to an item and the states for each node correspond to the possible rating values for each item. After training, each item in the resulting network will have a set of parents that are best predictors of its ratings. Each conditional probability table is represented by a decision tree encoding the conditional probabilities for that node. Over the experiments performed, the Bayesian network approach performed as well as correlation methods. Bayesian networks have typically smaller memory requirements and allow for faster predictions than a memory-based approach [6]. However, a training phase is required that can be time consuming. It was also shown that when there are relatively few ratings, Bayesian networks perform less well. Bayesian networks may prove practical for environments in which knowledge of user preferences changes slowly with respect to the time needed to build the model but are not suitable for environments in which user preference models must be updated rapidly or frequently.

Heckerman et al. [22] use **dependency networks** as an alternative to Bayesian networks. The graph of a dependency network is potentially cyclic (unlike a Bayesian network) and the probability component of a dependency network, like a Bayesian network, is a set of conditional distributions, one for each node given its parents. They evaluated dependency networks and Bayesian networks on three datasets and used three metrics: the accuracy of the recommendations; the prediction time (time taken to create a recommendation list given what is known about a user); and the computational resources needed to build the prediction models. In general, results showed that Bayesian networks are slightly more accurate than dependency networks but dependency networks are significantly faster at prediction and require substantially less time and memory to learn.

Horting is a graph-based technique in which nodes represent users and edges between nodes correspond to the notion of predictability [1]. Predictions are produced by traversing the graph to nearby nodes and combining the ratings of the nearby users. Horting differs from nearest neighbour approaches as the graph may be traversed through other users/nodes who have not rated the item in question, thus exploring transitive relationships that nearest neighbour algorithms do not consider. A prediction for item j for user i can be computed as weighted averages computed via a few reasonably short directed paths joining multiple users. Each directed path will connect user i at one end with another user k who has rated item j. In [1], using synthetic data, horting produced better predictions than a nearest neighbour algorithm.

Hofmann [24] proposes an **aspect model**–a latent class statistical mixture model–for associating word-document co-occurrence data with a set of latent variables. Hofmann et al. [25] apply the aspect model to user-item co-occurrence data for collaborative filtering. A latent class variable, $z \in \mathbf{Z} = \mathbf{z_1}, \ldots, \mathbf{z_k}$ is associated with each observation. The assumption is that x (set of persons) and y (set of objects) are independent, conditioned on z. The main motivation behind the introduction of the latent variables z is to explain the observed preferences by some smaller number of *typical* preference patterns which are assumed to underlie the data generation process (the cardinality of z is smaller than the number of items in the document, thus z is used as a "bottleneck" variable in predicting items).

Browning and Miller [7] test two **statistical modelling approaches**—Naive Bayes Model and Latent (variable) cluster model. They claim that such approaches are not affected by missing features and that the complexity has "limited" dependence on the dataset size. However, such approaches assume feature independence which is not necessarily valid in the collaborative filtering domain. They then present a "statistical modelling" approach to the collaborative filtering task—they view the task as one of learning a maximum entropy (ME) model [26]. The general idea is when only partial information about possible outcomes exist, probabilities should be chosen so as to maximise the uncertainty about the missing information. Several approaches to ME learning can be used including iterative scaling and its extensions (as used in [7]). The ME model has been used in a variety of tasks including natural language modelling, classification and inference tasks. In experiments, using the F1 measure ($\frac{2 \times precision \times recall}{precision + recall}$) and the *MS Web* dataset (using 285 features), the ME method achieved better results than naive Bayes and cluster models as well as better results than SVMs (discussed later) and the Pearson correlation method. A number of experiments were performed where x features were selected as known and ratings produced for the remaining features. With *x = 2* the ME method had an F1 value of 0.331, the Naive Bayes method gave a value of .317, the cluster model gave a value of .316, the correlation method gave a value of .198 and the SVM approach had an F1 value of .197. When x was increased to 10, all methods performed better but the relative performance of the approaches stayed the same. The ME technique does not assume feature independence and no matrix reduction is required. In addition, the ME model building complexity grows only with the number of known feature values and/or the number of ME constraints that are encoded. However, it could be an inefficient approach if changes are made to the data set over time.

3.4 Machine Learning

Various machine learning approaches have been investigated such as Bayesian networks (as described in the previous section), classification, clustering and rule-based approaches. Pre-processing is often carried out prior to the machine learning stage.

In general, **classification** involves assigning an object to some defined class. As discussed in the previous section, Browning and Miller [7] review the use of a **SVM** (support vector machine) approach whereby N SVMs are built—each one dedicated to the prediction of a single feature. They claim that even with preprocessing the approach is infeasible due to the effect missing features will have on the accuracy of predictions and the computational complexity of the learning task.

Billsus and Pazzani [5] describe a technique which combines a neural network coupled with singular value decomposition (SVD). A feed-forward neural network is created for each user. After training, each network will map a vector (representing an unseen item) to a predicted rating for that item for that user. The EachMovie data set was used and "reasonable prediction accuracy" was achieved. This approach appears to have two major problems: the approach is inefficient due to the fact that a neural network was used for each user; also SVD is computationally expensive and there is difficulty in knowing the best dimension to choose for the reduced matrix.

Boosting [16] is a general set of methods which produce a series of classifiers. The training set used for each member of the series is chosen based on the performance of earlier classifiers in the series, e.g., examples that are incorrectly predicted by previous classifiers in the series are chosen more often than examples that were correctly predicted. Thus, a classifier is forced to concentrate on the difficult examples in the training set. Two main approaches to boosting are called Arcing and Ada-Boosting. Freund et al. [15] view collaborative filtering as a list ranking problem and use a boosting algorithm, using a weak learner, for collaborative filtering. The goal of the learner is to produce a "good" ranking of all items, including those not observed in training. Results showed that it outperformed regression and nearest neighbour approaches (the nearest neighbour approach used found only one most similar user to the active user).

The EachMovie data set was used with 61625 users and 1628 movies. For experiments a subset, C, of the users were selected. Each user in C defined an ordering of the set of movies liked by that user. For each target user, 50% of these movies were used in the feedback function and 50% used for testing. Assuming that each approach produces a function H which orders movies, four measures were used to test the approaches: disagreement (the fraction of movies mis-ordered by H), predicted-rank-of-top (precision of the first good movie on $H's$ list); coverage (precision of last good movie) and average precision (how good H is at putting good movies high on the list). Three experiments were performed where the number of features, density of feedback and density of features were varied.

With feature size of 100, the average precision of rankboost was approx. 0.475; of nearest neighbour was 0.45 and of regression was 0.1.

Clustering involves dividing a heterogeneous group of objects into homogeneous subgroups. In a collaborative filtering domain, clustering techniques can be used to cluster users based on their similar preferences or cluster similar items. Predictions for a user in a cluster can be made using the ratings of other users in the same cluster.

Ungar and Foster [39] investigated a number of clustering approaches which could be applied to the problem. A movie domain is used and the model views classes of people and movies—movie classes are known, e.g. comedy, drama, etc., whereas people classes are unknown and must be derived. The model contains three sets of parameters:

- $P(k)$ the probability a (random) person is in class k.
- $P(l)$ the probability a (random) movie is in class l.
- $P(kl)$ the probability a person in class k is linked to a movie in class l.

Clustering approaches used were: k-means clustering; repeated clustering (clustering on clusters); and Gibbs sampling [2, 9]. Results showed that all three approaches gave comparable error rates when the classes had equal number of members and in general results depended on the nature of the data being fit. Choosing the attributes on which to cluster led to better performance. For example, CDs were clustered based on artist and the users were then clustered based on CD clusters. It was noted that repeated clustering may overcome the disadvantage of the sparsity problem but the approach also has the potential to over-generalise.

Lee [29] investigates two clustering approaches. In the first approach, it is assumed that each user is equally likely to belong to one of m clusters of users and the rating for each item is generated randomly according to a distribution that depends on the item and the cluster to which the user belongs. In the second approach each user is again assumed equally likely to belong to one of m clusters of users while each item is equally likely to belong to one of n clusters of items. The rating for a $< user, item >$ pair is then generated randomly according to a distribution that depends on the cluster to which the user belongs and the cluster to which the item belongs (similar to [39]). Heuristic algorithms to approximate Bayesian sequential probability assignment were developed where a row column clustering method and row clustering method were combined.

Experiments were performed on the EachMovie data set. In general, (using average absolute error), row clustering performed better than row column clustering and the correlation algorithm. The designed combined algorithm performed well for new users who had made very few ratings and for new items that had received very few ratings.

Mobasher et al. [30] consider a web mining domain where they aim to find overlapping aggregate profiles that can be used by recommender systems to provide recommendations. They evaluate two web mining techniques: PACT (profile aggregations based on clustering transactions), which clusters based on user transactions and association rule hypergraph partitioning, which clusters based on the viewing of a page. The *Clique* algorithm [34] was also used for comparison. The techniques were evaluated in terms of the quality of individual profiles generated and the quality of recommendations when the techniques were integrated with a "personalization engine". Results showed that both techniques are suitable for web personalisation.

O'Connor et al. [32] list three motivations for clustering items prior to using traditional collaborative filtering techniques:

- reduce dimensionality of space via clustering, thus reducing computation time.
- increase accuracy of predictions.
- increase potential for parallelism of task.

Their approach was to partition the data set (on items) and then apply traditional collaborative filtering techniques within each partition to produce recommendations. They experimented with four different clustering algorithms: average link hierarchical agglomerative [19]; ROCK (robust clustering algorithm for categorical attributes) [21]; and kMetis and hMetis [28, 27]. Some preprocessing was performed and the Pearson correlation coefficient was used to calculate the similarity between items. Results showed that kMetis was the most promising of the four clustering algorithms tested. However in most cases, the accuracy of recommendations for the partitioned model was not better than the accuracy of recommendations using the whole data set (unpartitioned). They posit that this could be due to the fact that the similarity measure used is based on rating data rather than content data. Also, due to the fact that they wanted to parallelise the task, an item could only belong to one cluster whereas certain items may have significant predictive value for a number of clusters.

Basu et al. [3] develop an **inductive learning approach** using *Ripper*[10] which can learn rules from data with set-valued attributes. The test set used is from the movie recommendation domain. Pre-processing of the data set involves converting each user/movie rating into a tuple of two set-valued features: movies liked by a user and users who liked a movie. The notion of *like* is defined to be any rating that is in the top quartile of the ratings made by a user. In addition, content information was added whereby a set was created of movie *genres* with three possible values — comedy, drama and action. The first experiment uses the following collaborative features for each movie: users who liked the movie; users who disliked the movie and movies liked by the user. In general precision was reasonable (78%) but there was a lower level of recall (27%) in comparison to traditional collaborative filtering techniques—in particular the *Recommender* system [23]. *Recommender* achieved 78% precision and 33% recall. A second experiment added 26 content features to the list of collaborative features. No improvements in precision and recall were noticed (73% and 33% respectively). A further experiment combined collaborative with content information relating to the genre of a movie. The hybrid features used were: comedies liked by user; dramas liked by user; action movies liked by user. In addition, features for each genre were used: users who liked *many* movies of genre X; users who liked *some* movies of genre X; users who liked *few* movies of genre X; users who disliked movies of genre X. With this data set, precision and recall were increased (83% and 34% respectively).

Nakamura et al. [31] cast the collaborative filtering problem as one of learning a binary relation between the users (rows) and items (columns). Prediction is carried out by the use of weighted majority binary prediction algorithms which are based on learning binary relations using weighted-majority voting [18]. Initially, each user is related to each item in the dataset. A value is calculated for the relation of a user to an item using a prediction function or set of prediction functions. In this process, users' preferences are considered to be the learnt *target function*.

Delgado [13] uses a pool of independent prediction algorithms—one for each user, where a prediction is made in each trial. The idea is, for some user, to find other target functions that consistently behave in a neutral, opposite or similar way to the active target function that the system is trying to learn. The algorithms prediction is a function of the original target function and a similarity measure between users (similarity calculated using correlation measure). Several learning techniques can be used to update weights, the most common being weighted majority voting which updates weights only when the prediction is wrong. The approach was not tested in [13].

3.5 List Ranking

Freund et al. [15] view collaborative filtering as an ordering task. They use a ranking approach where the output from a collaborative filtering system is a ranking of all items (per user) which accurately predicts which items a user will like more or less than other items. This list will include items that the user has not already rated.

Problems that involve ordering and ranking have been investigated in various fields such as decision theory, social science, information retrieval and mathematical economics. The problem of learning to rank is closely related to the I.R. problem of combining results from different search engines. A number of techniques can be used to learn the ranking model including, classification (using boosting) as carried out in [15]. Cohen et al. [11] also use a list ranking approach. They develop a framework for manipulating and combining multiple rankings with the aim of minimising the number of disagreements between rankings. In their framework, the rankings are used to construct preference graphs and the problem is reduced to an optimisation problem which is NP-complete and thus an approximation is used to combine the different rankings. Results were reported in Section 3.4 but are difficult to evaluate as the methods chosen with which to compare RankBoost were not standard ones (i.e., most other approaches choose Pearson correlation as the nearest neighbour approach with which to compare a new technique; this was not done in [11]).

3.6 Item-based Techniques

Item–based techniques [14, 8] identify relationships between different items, and then use these relationships to compute recommendations for users.

Fisk [14] presents a system *MORSE* which makes personalised film recommendations based on information about users' film preferences. The approach taken involves calculating the correlation between item j (on which a recommendation is required for user i) and all other items in the data set. Then for each user k in the data set, the ratings given by i to the N films most closely correlated with j is plotted against ratings given to the same films by the current k. The best-fit straight line is determined. The correlation between i and k (for the n films most closely correlated to with j) is also plotted. Using the same data set, results showed that the approach produced more accurate predictions than Pearson-r.

Sarwar et al. [8] looks at the set of items that the user has rated and firstly computes how similar they are to the target item i and then selects k most similar items $\{i_1, i_2, \ldots, i_n\}$. At the same time their corresponding similarities $\{s_{i1}, s_{i2}, \ldots, s_{in}\}$ are also computed. Once the most similar items are found the prediction is then computed by taking a weighted average of the target users ratings on these similar items. Each item pair in the co-rated set corresponds to a different user. A number of approaches can be used to compute the similarity between items including:

- correlation: compute the similarity between two items by calculating the Pearson-r correlation.
- cosine similarity: the two items are thought of as two vectors in the m dimensional user-space. The similarity between them is measured by computing the cosine of the angle between the two vectors.
- adjusted cosine similarity: the ratings are normalised by subtracting the user average from each co-rated pair before using the cosine similarity.

Approaches which can be used for prediction include:

- weighted sum: computes the prediction for an item i for a user u by computing the sum of the ratings given by the user on items similar to i. Each rating is weighted by the corresponding similarity (s_i, j) between items i and j.
- regression: instead of directly using the ratings of similar items, an approximation of the ratings based on a regression model is used. In practice the similarities computed using cosine or correlation measures may be misleading in the sense that two rating vectors may be distant (in Euclidean sense) yet may have high similarity. In such a case, using the raw similarity ratings may result in poor prediction.

Two types of experiments were performed: those testing the quality of predictions and those testing the performance of the approaches. Three types of similarity algorithms were tested (basic cosine, adjusted cosine and correlation). The weighted sum algorithm was used to generate predictions. The adjusted cosine showed lowest MAE and was used in the remaining experiments (MAE using the adjusted cosine measure was approx 0.733; the MAE using the basic cosine measure was approx. 0.835 and the MAE using the correlation measure was 0.83 approx.).

Results showed that the item-item algorithms slightly outperformed a Pearson user-user algorithm. With the training set/test set ratio set at 0.5 and the neighbourhood size set at 30 the item-item approach has an MAE of .749 and the user-user approach has an MAE of .755. Performance experiments showed that the item neighbourhoods are "fairly static" and thus can be pre-computed which will facilitate good online performance.

3.7 Advantages and Disadvantages

Some of the approaches could be viewed as theoretically stronger than neighbourhood-based approaches.

Modelling approaches in general provide flexibility and have scaled best with large datasets. Some can also handle missing values.

Many of the machine learning techniques are more computationally expensive that memory-based techniques. However, machine learning approaches suffer from being seen as a "black box" where it is difficult to explicitly state what has been learnt.

There is a potential problem with the application of statistical techniques due to the fact that these techniques may require certain features of data to hold and this may not be feasible. For this reason, machine learning approaches may be more robust.

4 Summary and Future Directions

Collaborative filtering has attracted much interest in the research domain since the publication of details of the *Tapestry* system in 1992 [17]. Collaborative filtering has been viewed as a complementary approach to traditional I.R. systems and many collaborative filtering approaches have been proposed. Early collaborative filtering techniques use standard statistical measures to calculate the correlation between users. Numerous other techniques have been proposed to improve on the performance of these techniques and to try overcome some of the problems inherent in the collaborative filtering domain, such as sparsity and high dimensionality. This paper has given an overview of these techniques.

Research in the area has by no means reached a steady-state and much work can yet be performed on evaluating existing approaches as well as the potential of investigating new models and new approaches.

References

[1] C.C. Aggarwal, J.L. Wolf, K.-L. Wu, and P.S. Yu. Horting hatches an egg: A new graph-theoretic approach to collaborative filtering. In *Proceedings of the Fifth ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'99), San Diego, CA*, pages 201–212, 1999.

[2] M. Aldenderfer and R. Blashfield. *Cluster Analysis*. Sage Publishers, 1984.

[3] C. Basu, H. Hirsh, and W. Cohen. Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, pages 714–721, 1998.

[4] M. Berry, S. Dumais, and G. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4):573–595, 1995.

[5] D. Billsus and M. Pazzani. Learning collaborative information filters. In *AAAI Workshop on Recommender Systems*, pages 24–28, July 1998.

[6] J. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, July 1998.

[7] J. Browning and D.J. Miller. A maximum entropy approach for collaborative filtering. In *IEEE International Workshop on Neural Networks for Signal Processing*, 2001.

[8] B.Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the Tenth International World Wide Web Conference on World Wide Web*, pages 285–295, 2001.

[9] G. Casella and E.I. George. Explaining the gibbs sampler. *The American Statistician*, 46:167–174, 1992.

[10] W. Cohen. Fast effective rule induction. In *Machine Learning: Proceedings of the Twelfth International Conference*. Morgan Kaufmann, 1995.

[11] W.W. Cohen, R.E. Schapire, and Y. Singer. Learning to order things. *Advances in Neural Information Processing Systems*, 10, 1998.

[12] S. Deerwester, S.Dumais, T. Landauer, G. Furnas, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[13] J. Delgado and N. Ishii. Memory-based weighted-majority prediction. In *ACM SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation*, 1999.

[14] D. Fisk. An application of social filtering to movie recommendation. In H.S. Nwana and N. Azarmi, editors, *Software Agents and Soft Computing*. Springer, 1997.

[15] Y. Freund, R. Iyer, R.E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. In *Proceedings of 15th International Conference on Machine Learning*, pages 170–178. Morgan Kaufmann, San Francisco, CA, 1998.

[16] Y. Freund and R.E. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, 1999.

[17] D. Goldberg, D. Nichols, B. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–69, 1992.

[18] S. Goldman and M. Warmuth. Learning binary relations using weighted majority voting. In *Machine Learning 20*, pages 245–271, 1995.

[19] E. Gose, R. Johnsonbaugh, and S. Jost. *Pattern Recognition and Image Analysis*. Prentice Hall, 1996.

[20] J. Griffith and C. O'Riordan. Collaborative filtering. Technical report, Deptartment of Information Technology, NUI, Galway, 2000.

[21] S. Guha, R. Rastogi, and K. Shim. Rock: a robust clustering algorithm for categorical attributes. In *Proceedings of the 15th International Conference on Data Engineering*, 1999.

[22] D. Heckerman, D.M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie. Dependency networks for collaborative filtering and data visualization. In *Proc. 16th Conference on Uncertainty in Artificial Intelligence*, pages 264–273, 2000.

[23] W. Hill, L. Stead, M. Rosenstein, and G. Furnas. Recommending and evaluating choices in a virtual community of use. In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, pages 194–201, 1995.

[24] T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Twenty-Second Annual International SIGIR Conference on Research and Development in Information Retrieval*, 1999.

[25] T. Hofmann and J. Puzicha. Latent class models for collaborative filtering. In T. Dean, editor, *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*. Morgan Kaufmann, 1999.

[26] E.T. Jaynes. The maximum entropy formalism. In *Where Do We Stand on Maximum Entropy?*, pages 15–118. The MIT Press, Cambridge, Massachusetts, 1979.

[27] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. Multilevel hypergraph partitioning: Applications in vlsi domain. Technical report, Department of Computer Science, University of Minnesota Technical Report, 1997.

[28] G. Karypis and V. Kumar. Unstructured graph partitioning and sparse matrix ordering. Technical report, Department of Computer Science, University of Minnesota Technical Report, 1995.

[29] W. Lee. Online clustering for collaborative filtering. Technical report, Department of Computer Science, National University of Singapore, Singapore, 2000.

[30] B. Mobasher, H. Dai, T. Luo, M. Nakagawa, and J. Witshire. Discovery of aggregate usage profiles for web personalization. In *Proceedings of the WebKDD Workshop*, 2000.

[31] A. Nakamura and N. Abe. Collaborative filtering using weighted majority prediction algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning*, 1998.

[32] M. O'Connor and J. Herlocker. Clustering items for collaborative filtering. In *Proceedings of the ACM SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation*, 1999.

[33] D. Pennock, E. Horvitz, S. Lawrence, and C.L. Giles. Collaborative filtering by personality diagnosis: A hybrid memory-and model-based approach. In *Proceedings of the 16th Conference on Uncertainty in AI (UAI-2000)*, pages 473–480. Morgan-Kaufmann, San Francisco, 2000.

[34] M. Perkowitz and O. Etzioni. Adaptive web sites: automatically synthesizing web pages. In *Proceedings of Fifteenth National Conference on Artificial Intelligence*, 1998.

[35] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM 1994 Conference on CSCW*, pages 175–186. Chapel Hill, 1994.

[36] E. Rice. User modelling via stereotypes. *Cognitive Science*, 2:225–266, 1979.

[37] B.M. Sarwar, G. Karypis, J.A. Konstan, and J.T. Riedl. Application of dimensionality reduction in recommender system–a case study. In *ACM WebKDD, Web Mining for e-Commerce Workshop*, 2000.

[38] U. Shardanand and P. Maes. Social information filtering: Algorithms for automating word of mouth. In *Proceedings of the Annual ACM SIGCHI on Human Factors in Computing Systems (CHI '95)*, pages 210–217, 1995.

[39] L. H. Ungar and D.P. Foster. Clustering methods for collaborative filtering. In *Workshop on Recommender Systems at the 15th National Conference on Artificial Intelligence*, 1998.