Chapter 1

Identifying and Analyzing User Model Information from Collaborative Filtering Datasets

Josephine Griffith* and Colm O'Riordan* and Humphrey Sorensen†

*Department of Information Technology
National University of Ireland, Galway, Ireland
josephine.griffith@nuigalway.ie
colmor@it.nuigalway.ie
† Department of Computer Science
University College Cork, Ireland
sorensen@cs.ucc.ie

This paper considers the information that can be captured about users from a collaborative filtering dataset. The aims of the paper are to create a user model and to use this model to explain the performance of a collaborative filtering approach. A number of user features are defined and the performance of a collaborative filtering system in producing recommendations for users with different feature values is tested. Graph-based representations of the collaborative filtering space are presented and these are used to define some of the user features as well as being used in a recommendation task.

1.1. Introduction

Modern information spaces are becoming increasingly more complex with information and users linked in numerous ways, both explicitly and implicitly, and where users are no longer anonymous, but generally have some identification and a context in which they navigate, search and browse. This offers new challenges to recommender system designers, in capturing and combining this information to provide a more personalized and effective retrieval experience for a user.

The original foundations of collaborative filtering came from the idea of "automating the word of mouth process" that commonly occurs within social networks,[1] i.e. people will seek recommendations on books, CDs, restaurants, etc. from people with whom they share similar preferences in

these areas.

Although collaborative filtering is most frequently seen as a way to provide recommendations to a set of users, collaborative filtering datasets also allow for the analysis of social groups and of individual users within a group, thus providing a means for creating a new user model, group model or for augmenting an existing user or group model.

User modelling has had a long history in many computer science domains and traditionally user models were created based on evidence from explicit user actions. There has been a gradual change in this approach and the focus is often on building a model for a user using implicit information gleaned from the user's interactions with a system, the user's interactions with data and information, and the user's interactions with other users.

A social network can be defined as a graph representing relationships and interactions among individuals.[2] Nodes in the graph represent individuals and the links between the nodes represent some relationship or relationships between individuals. Many modern social networks are found on the Internet in the form of virtual communities and the study and analysis of social networks occur in many different fields. A number of systems based on social networks and small world networks have been proposed for referral and recommendation.[3–7] Other work linking social networks and collaborative filtering has viewed the collaborative filtering dataset as a social network with the aim of analyzing properties of users and items to improve retrieval performance.[8–11] Aims other than solely improving retrieval performance have also been explored.[9]

This paper considers the ways that recommender systems bring users together and considers how the information from these recommender systems can be extracted to form user models. The motivation for this work is that although, in collaborative filtering approaches, users are often clustered into groups based on finding "similar users", there is no modelling of the features of a particular user or group. Also, with the exception of simple cases (e.g. when a user has given very few ratings), it is not clear what effect these features have on recommendation accuracy.

The goals of the work presented in this paper are to specify some of the information that can be captured about users given a collaborative filtering dataset and to provide a model that will represent these features of users. In this work, eight features that can be extracted from the collaborative filtering dataset are firstly identified and defined. Some of these features are particular to the recommendation task while some features use measures

from social network theory and information retrieval. The eight features are then analyzed with respect to their effect on recommendation accuracy using a collaborative filtering approach. This is done, for each feature, by taking sample test users that have a particular value for the feature and by testing the accuracy of a collaborative filtering recommender system in providing predictions for the test users.

The user model defined will be used in future work to ascertain if improvement in recommendation accuracy can be achieved (by allowing the development of more personalized recommender algorithms) and also the model will be used to maintain histories of users in a collaborative filtering information space.

Section 1.2 presents related work in collaborative filtering, graph-based approaches to recommendation and social networks. Section 1.3 outlines the methodology, presenting the collaborative filtering approach and the graph models used as well as specifying the user features which are extracted from the collaborative filtering dataset. Section 1.4 discusses the experiments performed and the experimental set-up. Section 1.5 presents results and Sec. 1.6 presents conclusions, discussing the potential usefulness of the features and approach and outlining future work.

1.2. Related Work

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 user ratings. Collaborative filtering systems generally make use of one type of information, that is, prior ratings that users have given to items. However, some recent work has investigated the incorporation of other information, for example, content,[12] time,[13] and trust[14] information. To date, application domains have predominantly been concerned with recommending items for sale (e.g. movies, books, CDs) and with small amounts of text such as Usenet articles and email messages. The datasets within these domains will have their own characteristics, but they can be predominantly distinguished by the fact that they are both large and sparse, i.e. in a typical domain, there are many users and many items but any user would only have ratings for a small percentage of all items in the dataset.

The problem space can be viewed as a matrix consisting of the ratings given by each user for the items in a collection, i.e. the matrix consists of a set of ratings $r_{a,i}$, corresponding to the rating given by a user a to an item i. Using this matrix, the aim of collaborative filtering is to predict the

ratings of a particular user, a, for one or more items not previously rated by that user. The problem space can equivalently be viewed as a graph where nodes represent users and items, and nodes and items can be linked by weighted edges in various ways. Graph-based representations have been used for both recommendation and social network analysis of collaborative filtering datasets.[11,15]

1.2.1. *Weighting schemes in collaborative filtering*

There has been much work undertaken in investigating weighting schemes for collaborative filtering where these weighting schemes typically try to model some underlying bias or feature of the dataset in order to improve prediction accuracy. For example, in Ref. 16 and Ref. 17 an inverse user frequency weighting was applied to all ratings where items that were rated frequently by many users were penalized by giving the items a lower weight. In Ref. 18 and Ref. 17 a variance weighting was used which increased the influence of items with high variance and decreased the influence of items with low variance. The idea of *tf-idf* weighting scheme from information retrieval was used in Ref. 19 (using a row normalisation) and in Ref. 20 (using a probabilistic framework). Work in Ref. 21, Ref. 22 and Ref. 11 involve learning the optional weights to assign to items. In Ref. 14 more weight is given to user neighbours who have provided good recommendations in the past (this weight is calculated using measures of "trust" for users) and in Ref. 23 more weight is given to items which are recommended more frequently (where the weights are calculated using an "attraction index" for items). In general, although some of the weighting schemes for items have shown improved prediction accuracy (in particular those involving learning), it has proven difficult to leverage the feature information to consistently improve results.

1.2.2. *Graph-based approaches for recommendation*

Several researchers have adopted graph representations to develop recommendation algorithms. A variety of graphs have been used, including, among others, directed, two-layer, etc. and a number of graph algorithm approaches have been adopted (e.g. horting,[24] spreading activation[15]).

Aggarwal *et al.* present *horting*, a graph-based technique where nodes represent users and directed edges between nodes correspond to the notion of predictability.[24] Predictions are produced by traversing the graph to nearby nodes and combining the ratings of the nearby users.

Huang *et al.* present a two-layer graph model where one layer of nodes corresponds to users and one layer of nodes corresponds to items.[15] Three types of links between nodes are represented: item–item links representing item similarity based on item information, user–user links representing user similarity based on user information, and inter-layer user–item links between items and users that represent a user's rating (implicit or explicit) for an item. Transitive relationships between users, using a subset of this graph representation, are explored in Ref. 25. A bipartite graph is used with one set of nodes representing items and the second set of nodes representing users. Binary weighted edges connect the nodes between the two sets where an edge has a weight of 1 if a user purchased, or gave positive feedback to, an item and a weight of 0 otherwise. The goal is to compare how well different collaborative filtering approaches deal with the sparsity problem and the cold start problem for new users.

A number of approaches have been proposed to effect retrieval and filtering using graph representations. One such approach is spreading activation which originated from the field of psychology and was first used in computer science in the area of artificial intelligence to process semantic networks. Spreading activation approaches have been used in many information retrieval applications[26] and more recently in the domain of collaborative filtering.[25] Spreading activation approaches have also been used to integrate sources of evidence and information.[27,28]

1.2.3. *Collaborative filtering as a social network*

As well as being used for recommendation, a collaborative filtering dataset has been viewed as a social network where nodes in the network represent users and the links between users can be calculated based on the items users have accessed and/or the actual ratings that users have given to these items.[2,10,11] Rashid *et al.* state that "In contrast to other social networks, recommender systems capture interactions that are *formal*, *quantitative*, and *observed*."[11]

A social network can be defined as a network (or graph) of social entities (e.g. people, markets, organizations, countries), where the links (or edges) between the entities represent social relationships and interactions (e.g. friendships, work collaborations, social collaborations, etc.). Recently, online relationships between people have also been used to create social networks.

A number of recommender systems based on social networks and small world networks have been developed. Such social networks have been built using histories of email communication,[6] co-occurrence of names on WWW pages,[3] co-use of documents by users,[29] and matching user models and profiles.[7]

Palau *et al.* represent the agents in a multiagent restaurant recommender system using a social network where the connections between agents are based on the level of trust the agents have in the recommendations of other agents. Social network theory measures of size, density, network centrality, and clique and faction substructures are used to help give an explanation of the performance of the system.[10]

Lemire considers the social network feature of *influence* and found that recommendation results were better if the system was not "too democratic", i.e. it was found that it was better not to penalize users with a high number of ratings.[8] In addition, Lemire discusses the *stability* of a collaborative filtering system, defining stability as a property which exists if a single user in a large set does not make a difference to the results for some active user.

Mirza *et al.* also induce a social network from a collaborative filtering dataset where connections between users are based on the co-ratings of the same items.[9] They define a *hammock jump* as a connection between two users in the network that will exist if the users have co-rated at least w items (where w is defined as the hammock width). Herlocker *et al.* refer to this measure as a *significance weighting* whereby they devalue the correlation value between two users if this correlation value has been calculated based on only a small number of co-rated items.[30]

In Ref. 31, a graph-based representation is used to analyze various features of a dataset in order that the suitability of a collaborative filtering algorithm to a particular dataset can be ascertained (in particular, to give an indication of whether a naive (Top-N), user–user, item–item or spreading activation collaborative filtering algorithm would work well with the dataset).

1.3. Methodology

In this paper the focus is to extract implicit user information available from the collaborative filtering dataset and to form a user model for each user. This implicit information is based on simple features which can be extracted from any recommendation dataset (e.g. number of items rated, liked, disliked, etc.) as well as extracting features which

are based on measures from social network theory and information retrieval.

The user model consists of a octet containing, for each user, the values for eight identified features. Each of the eight features is individually analyzed by considering a set of users with different values for the feature. For example, one set of users are those who have rated close to the maximum number of items rated; another set of users are those who have rated close to the average number of items rated. Test users from each of these sets are chosen as the active users of a collaborative filtering recommender system, i.e. these are the users for which a recommendation is sought. The accuracy of the recommender system in providing recommendations for each set of users is found and these results are compared for each feature (e.g. the set of users who rated close to the maximum number of items may receive better predictions than the set of users who rated close to the minimum number of items). Therefore, for an individual user, some explanation as to why the system performs poorly or well for the user can be given by looking at the feature values for that user.

In addition we investigate the performance of a graph-based approach to recommendation with a view to incorporating the user features into a graph representation in future work. A graph-based representation is also used to define some of the user features.

The collaborative filtering system that is used to provide recommendations to different sets of users is described in Sec 1.3.1. Two graph-based representations of the collaborative filtering problem are presented in Sec 1.3.2. Each of the eight features of the user model are defined in Sec 1.3.3.

1.3.1. *Collaborative filtering approach*

The collaborative filtering problem space is often viewed as a matrix consisting of the ratings given by each user for some of the items in a collection. Using this matrix, the aim of collaborative filtering is to predict the ratings of a particular user, a, for one or more items not previously rated by that user. Memory-based techniques are the most commonly used approach in collaborative filtering although numerous other approaches have been developed and used.[16] Generally, traditional memory-based collaborative filtering approaches contain three main stages (for some active user a):

(1) Find users who are similar to user a (the neighbours of a).
(2) Select the "nearest" neighbours of a, i.e. select the most similar set of

users to user a.

(3) Recommend items that the nearest neighbours of a have rated highly and that have not been rated by a.

Standard statistical measures are often used to calculate the similarity between users in step 1 (e.g. Spearman correlation, Pearson correlation, etc.).[32] In this work, similar users are found using the Pearson correlation coefficient formula (1.1):

$$\mathrm{corr}_{a,u} = \frac{\sum_{i=1}^{m} (r_{a,i} - \bar{r_a}) \times (r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^{m} (r_{a,i} - \bar{r_a})^2} \times \sqrt{\sum_{i=1}^{m} (r_{u,i} - \bar{r_u})^2}} \tag{1.1}$$

where $\mathrm{corr}_{a,u}$ is the correlation value between users a and u (a value in the range $[-1, 1]$) for m items rated by users a and u, $r_{a,i}$ is the rating given by user a to item i, $r_{u,i}$ is the rating given by user u to item i, $\bar{r}_a$ and $\bar{r}_u$ are the average ratings given by users a and u, respectively.

An adjustment (using a *significance weighting*) is used in the Pearson correlation calculation based on the number of items that users have rated in common (co-rated items).[30] The motivation is that two users may receive a high correlation value but this might only be based on a small number of co-rated items. The adjustment ensures that users must have similar preferences over more than a few items to be considered highly correlated. The adjustment used in this work involves multiplying $\mathrm{corr}_{a,u}$ by the significance weighting if the number of co-rated items is less than twice the average number of co-rated items. The significance weighting between two users a and u is defined as (1.2):

$$\frac{\mathrm{cr}_{a,u}}{2 \times \ \mathit{average}} \tag{1.2}$$

where $\mathrm{cr}_{a,u}$ is the number of items users a and u have co-rated and *average* is the average number of items that have been co-rated by all users in the dataset.

The "nearest" neighbors of a user are selected using a low neighbour selection threshold, with any correlation value greater than 0.01 being considered. Although Breese[16] found that users with high correlation values (> 0.5) were more valuable in providing recommendations, work by Herlocker[33] using the Movie Lens dataset found that, for this dataset, such a high threshold sacrificed coverage and in addition, higher thresholds never improved the accuracy of predictions. They found that experiments using all correlation values greater than 0 always outperformed experiments with higher thresholds.[33]

Rating predictions for items for a user a (step 3) are found using the formula (1.3):

$$\text{pred}_{a,i} = \frac{\sum_{u=1}^{n} (r_{u,i} - \bar{r_u}) \times \text{corr}_{a,u}}{\sum_{u=1}^{n} \text{corr}_{a,u}} \tag{1.3}$$

where $\text{pred}_{a,i}$ is the prediction for item i for user a, n is the number of neighbors of user a, $r_{u,i}$ is the rating given by user u to item i, $\bar{r_u}$ is the average rating given by user u, and $\text{corr}_{a,u}$ is the correlation value between users a and u.

1.3.2. *Graph-based representations of the collaborative filtering space*

As discussed in the previous section the collaborative filtering problem space is often viewed as a matrix. The problem space can equivalently be viewed as a graph consisting of a set of user nodes and a set of item nodes. Two different graph representations are considered in this work.

In the first representation (see Fig. 1.1), user and item nodes are connected via weighted edges where the weights on the edges represent the ratings given to items by users. Apart from some scaling of the rating values this graph is a direct mapping of the matrix representation of the data to a graph representation of the data.

The second representation (see Fig. 1.2) is a social network representation which only considers user nodes. These user nodes are connected via weighted edges if the users are deemed sufficiently similar to each other. This similarity is calculated using the Pearson correlation formula where positive correlation values indicate similarity. A threshold value of 0.25 is used so that an edge only exists between users if their correlation value is greater than 0.25. For the collaborative filtering case, commonly used correlation measures are not commutative so therefore in the representation used, two edges can exist between two users.

Note that the two representations can be combined into a single graph representation. Currently, in the given representations, the information on user features is not represented explicitly. To represent this information explicitly, additional edges can be added to the graph to represent further relationships between users and items, relationships between items and relationships between users. For example, a relationship can exist between commonly rated items; between highly rated items, etc.

To provide recommendations, the graph representation in Fig. 1.1 is augmented such that three weighted edges connect nodes: one undirected

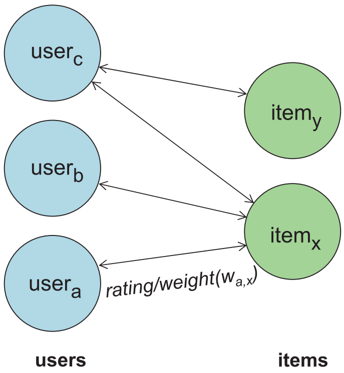

Fig. 1.1. Graph representation of users, items and ratings.

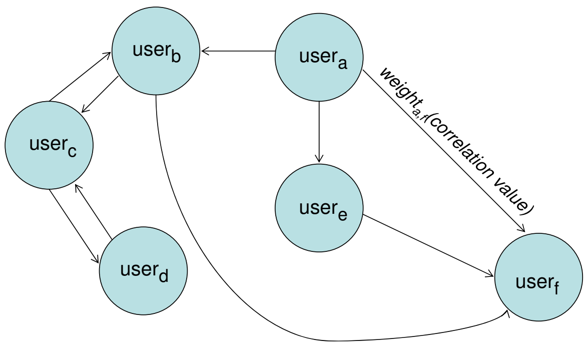

Fig. 1.2. Graph representation of users and their similarity.

edge representing the rating (or weight, w_i) and the second and third directed edges representing node outputs (output_i). Associated with each user node and item node is an activity and a threshold (see Fig. 1.3 which shows this augmentation for a portion of the graph from Fig. 1.1).

The activity of a user or item node a, for N nodes connected to the node a with nonzero weight, is calculated by (1.4):

$$\text{activity}_a = \sum_{i=1}^{N} \text{output}_i w_i \tag{1.4}$$

where output_i is the output of the node i that is connected to node a and w_i is the weight on the edge connecting node i to a. The output, output_a,

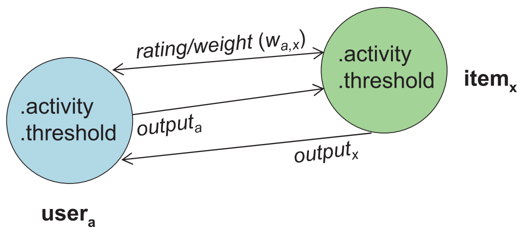

Fig. 1.3. Extended graph representation of a single user and item node.

of a user or item node is calculated using a threshold function (1.5):

$$\text{output}_a = \begin{cases} \text{activity}_a & \text{if activity}_a > \tau, \\ 0 & \text{otherwise,} \end{cases} \tag{1.5}$$

where the threshold function uses the node's activity and a threshold value, τ. Each node may have its own threshold value.

The terminology of a hop is used in this paper to define the activation spreading from one set of nodes to a second set of nodes. A hop involves the calculation of all node outputs in either the user set or item set, updating the associated activities and outputs of the nodes. The steps involved in the spreading activation approach are as follows:

1. Hop 1: Calculate the activities of all item nodes connected, with nonzero weight, to the current active user node. For each activated item node, calculate the output of the node using the threshold function.
2. Hop 2: Calculate the activities of all user nodes connected, with nonzero weight, to item nodes where the item nodes have nonzero output. For each activated user node, calculate the output of the node using the threshold function.
3. Hop 3: Calculate the activities of all item nodes connected, with nonzero weight, to user nodes where the user nodes have nonzero output. For each activated item node, calculate the output of the node using the threshold function.
4. Following three hops, items with the top-N highest positive activities are recommended to the active user.
5. Steps 2 and 3 can be repeated any number of times before recommendations are given (step 4).

Two hops result in activating a set of user nodes constituting a user neighborhood of the original active user node. The third hop, from user nodes

to item nodes, provides item recommendations for the active user.

1.3.3. *User features*

A user model is defined which consists of eight features. For some user a the features are defined as follows:

(1) *rated* is the number of items rated by the user a.
(2) *liked* is the percentage of items rated by the user a that the user a liked, and is calculated by (1.6):

$$\frac{\mathrm{num}_{\mathrm{liked}}}{\mathrm{rated}} \tag{1.6}$$

where $\mathrm{num}_{\mathrm{liked}}$ is a count of the number of items liked by the user a and *rated* is the number of items rated by the user a. In this work, an item is considered to be liked by a user if it receives a value greater than the middle value of the rating range, as also used in Ref. 34 (e.g. if the rating range is $[1, 5]$ a liked item is an item that receives a value of 4 or 5).
(3) *disliked* is the percentage of items rated by the user a that the user a disliked and is calculated by (1.7):

$$\frac{\mathrm{num}_{\mathrm{disliked}}}{\mathrm{rated}} \tag{1.7}$$

where $\mathrm{num}_{\mathrm{disliked}}$ is a count of the number of items disliked by the user a and *rated* is the number of items rated by the user a. An item is considered to be disliked by a user if it receives a value less than the middle value of the rating range.
(4) *avg-rating* is the average rating value given to items by the user a.
(5) *std-dev* is the standard deviation of the ratings of user a.
(6) *influence* is a measure of how influential a user is in comparison to other users. As also considered in Refs. 11 and 9, *influence* is defined in this work by using measures from social network theory. In particular, degree centrality is used where the dataset is viewed as a graph (or social network) where nodes represent users and the values of weights on edges between users are based on the strength of similarity of users to each other (as shown in Fig. 1.2). Degree centrality is then measured by counting the number of edges a node has to other nodes. Essentially this is a count of the number of neighbours (above a correlation threshold of 0.25) a user has.

(7) *clustering-coeff* is also a measure taken from social network theory and measures how similar users in a group are to each other using the clustering coefficient measure. This measures how connected the neighbours of the user a are to each other using the graph representation in Fig. 1.2. For example, if none of user a's neighbours are connected to each other, the clustering coefficient is 0 whereas if this subgraph has a clustering coefficient of 1 then all of user a's neighbours are connected to each other. The clustering coefficient is calculated by (1.8):

$$\frac{\text{actual}}{\text{possible}} \tag{1.8}$$

where *actual* is the number of actual links between neighbour nodes and *possible* is the number of possible links which can exist between neighbour nodes. In the representation described the total number of possible links that can exist between n nodes is $(n^2 - n)$.
In addition, in the collaborative filtering case it is possible that small sub-groups (small values of n) will have high clustering coefficients and therefore comparisons using clustering coefficient values may not always be meaningful. To overcome this the formula is extended to also include the active user in the calculation.[35] Thus the formula for the clustering coefficient for a user a with degree, $deg(a)$, and n neighbour nodes with degree greater than 1 becomes (1.9):

$$\frac{actual + deg(a)}{(n+1)^2 - n + 1} \tag{1.9}$$

Considering the graph shown in Fig. 1.2 with the active user being $user_a$, who has degree 3 and three neighbours ($n = 3$) who are connected to each other as follows: user e is connected to user f and user b is connected to user f. Therefore the number of actual links is 2 and the clustering coefficient for this group is 0.42.

(8) *importance* is a measure taken from Information Retrieval. Some collaborative filtering weighting schemes incorporate the idea from Information Retrieval of a *term frequency, inverse document frequency* (*tf-idf*) weighting.[19,20] The idea is to find terms with high discriminating power, i.e. terms which “describe” the document well and also distinguish it from other documents in the collection. Mapping the idea of *tf-idf* to collaborative filtering, a “term” can be viewed as a user with associated ratings for M distinct items. The more ratings a user has the more important the user is, unless the items that the user has rated have been rated frequently in the dataset. Note that the value a user

gives an item is not a frequency or weight - it is an indication that the item has been rated and thus the actual rating value is not used in the following formula. The formula used to calculate the importance, w_i, of a user i is:

$$w_i = \frac{1}{M} \times \sum_{j=1}^{M} \left(1 + \log \frac{n}{n_j}\right) \tag{1.10}$$

where n is the total number of users in the dataset; M is the number of ratings by user i and n_j is the number of users who rated item j.

1.4. Experiments

This section presents details of the experiments performed using the collaborative filtering approach and the graph-based approach outlined previously. The first set of experiments involve analyzing each of the eight features identified in the previous section using a collaborative filtering approach. The final experiment involves testing the performance of the graph-based representation illustrated in Fig. 1.3 using a spreading activation approach to collaborative filtering.

1.4.1. *User model features*

The main experiments involve checking the relative performance of a collaborative filtering approach using different sets of users for each of the eight features. A set of users consists of the users who have the same value, or nearly the same value, for an identified feature. The aim is to ascertain which sets of users will be more likely to have better or worse predictions (measured using the mean absolute error (MAE) metric).

A standard subset of the Movie Lens dataset is considered that contains 943 users and 1682 movies. A proportion of the dataset is removed for testing, as described below, and the metric of mean absolute error is used to compare the performance of the collaborative filtering approach using different sets of users, for each feature, with different feature values.

For each feature, the range of values for that feature (e.g. [0.12, 1] for the *liked* feature) is broken into regular intervals (typically 8 intervals) and users belong to a particular interval based on their value for that feature. All users in a particular interval then form a set. Intervals are chosen such that the set size (the number of users in each interval) is close to 100.

For testing, 30 users are chosen randomly from each set as the test users and 10% of their ratings for items are removed to yield the items to test (i.e. the system should return predictions for these items). MAE results are averaged over 10 runs for each set of users, for each feature. In addition, for each feature a control set of 30 users is chosen randomly from the entire dataset as test users (i.e. the users are chosen without considering the feature value of these users).

1.4.2. *Spreading activation*

The purpose of this experiment is to test whether the graph representation and spreading activation approach are sufficiently accurate to be used in future work involving the incorporation of the user features into the graph representation. The experiment involves the comparison of a spreading activation approach and a traditional memory-based collaborative filtering approach. The reason for choosing a traditional memory-based collaborative filtering approach is that it has been shown to perform well in comparison to many other collaborative filtering techniques.[33,36] Also, as can be seen from the descriptions of the approaches in Secs. 1.3.1 and 1.3.2, it is quite similar to the spreading activation approach outlined. The important difference between the representations and approaches is in terms of the flexibility of the graph-based representation and spreading activation approach in allowing the incorporation of additional information.

Again, the Movie Lens dataset is used. Weights on the network edges indicate the strength of like/dislike for an item where "dislike" can be viewed as an inhibitory or negative rating and "like" can be viewed as an excitatory or positive rating. Given that the original rating values in the Movie Lens dataset are all positive numbers, the approach adopted maps the ratings to positive and negative values to indicate positive and negative influences. The mapping chosen is to subtract 2.5 from all nonzero values which will give ratings around 0, giving:

$$\{0,\ 1,\ 2,\ 3,\ 4,\ 5\} \rightarrow \{0,\ -1.5,\ -0.5,\ 0.5,\ 1.5,\ 2.5\}.$$

The value 0 is not changed in this mapping as it has a special meaning, being used to indicate that no rating has been given to an item.

A proportion of the dataset is removed for testing and the metric of precision is used to compare the performance of the two approaches at different recall points. Precision is used because the spreading activation approach returns a ranking of recommended items, not prediction values

that can be compared with actual values.

The collaborative filtering approach and settings used are those already described. In the spreading activation approach to collaborative filtering, three stages corresponding to the three stages in the traditional memory-based collaborative filtering approach are used. Neighbours of some active user are found after two hops of the approach, at which stage user nodes that have nonzero activity are the neighbours of the active user. When activation is spread again, from user nodes to item nodes, items not rated by the active user will be highlighted. These items are recommended to the user if the activity is sufficiently high. The threshold value used in these experiments is 0 for all nodes, i.e. all positive activities will result in a node outputing a value.

1.5. Results

1.5.1. *User model features*

Results are presented for each of the eight features using the experimental methodology outlined in the previous section.

Figure 1.4 shows the MAE results when the *rated* feature was analyzed for eight sets of users. The *rated* value ranges from 0 to 737. The users in the first set (0–24 interval) have rated 0–24 items; the users in the second set (25–30 interval) have rated 25–30 items, etc. A random group of 30 users (with varying *rated* values) was also chosen (and are not included on the graph). This random group had an average MAE value of 0.7624. As expected, the worst MAE value for any set was for the users in the set who have rated between 0 and 24 items, i.e. these users have provided the very minimum number of ratings. Although we would expect that the accuracy should steadily increase as the number of ratings users have given increases, this was not necessarily the case. However, users who have rated closer to the maximum number of items have the best MAE values.

Figures 1.5 and 1.6 show the MAE results when the *liked* and *disliked* features were analyzed for nine sets of users. For the *liked* feature, values range from 0.12 to 1 where a value of 1 indicates that a user liked all the items that they rated and a value of 0.12 indicates that a user liked very few of the items that they rated. A random group of 30 users (with varying *liked* values) was also chosen and had an average MAE value of 0.7495. For the *disliked* feature values range from 0 to 0.87 where a value of 0 indicates that a user liked all the items that they rated. A random group of 30 users

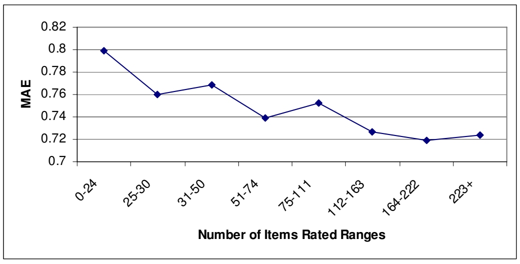

Fig. 1.4. *rated* MAE analysis.

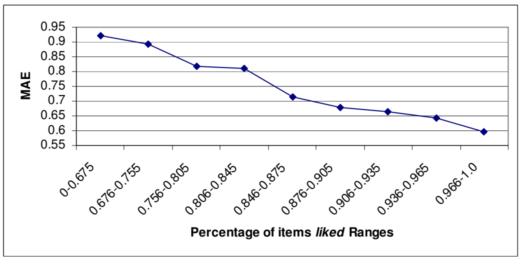

Fig. 1.5. *liked* MAE analysis.

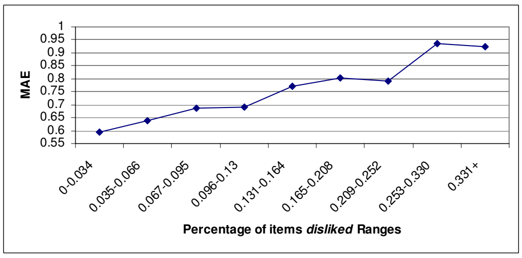

Fig. 1.6. *disliked* MAE analysis.

(with varying *disliked* values) were also chosen and had an average MAE value of 0.7507. As can be seen from both graphs, the results improve when the percentage of positively rated items (i.e. those liked by a user) increases.

Figure 1.7 shows the MAE results when the *avg-rating* feature was analyzed for eight sets of users. The *avg-rating* value ranges from 1.0 to 4.869. The MAE for 30 randomly chosen users was 0.7321. The users with lowest

averages (from the minimum to 3.03) have the worst MAE and the users with the highest averages (> 4.10) have the best MAE.

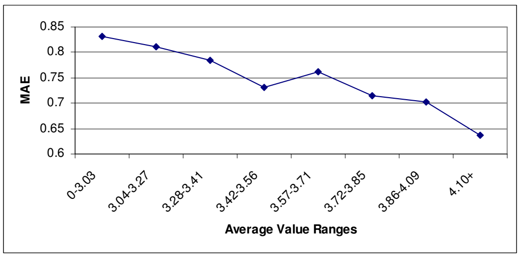

Fig. 1.7. *avg-rating* MAE analysis.

Fig. 1.8 shows the MAE results when the standard deviation feature (*std-dev*) was analyzed for eight sets of users. The *std-dev* value ranges from 0.3499 to 1.718. The users with low standard deviation (< 0.779) exhibited the best MAE value (0.5595 in comparison to the MAE of the randomly selected group which was 0.7779) while the users with the highest standard deviation had the worst MAE. This suggests that better recommendations can be found for users with lower variance in their ratings.

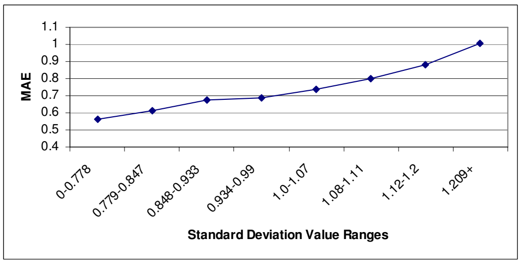

Fig. 1.8. *std-dev* MAE analysis.

Fig. 1.9 shows the MAE results when the *influence* feature was analyzed for eight sets of users. The $influence$ value ranges from 0 to 392 where an $influence$ value of 0 means that a user has no neighbours. As expected, the users with fewest neighbours (0 or 1) have the worst MAE values and as the neighbourhood size grows there is a general trend towards lower MAE values. The MAE of the random group was 0.7508.

The clustering coefficient feature (*clustering-coeff*) was analyzed for eight sets of users with values ranging from 0 to 0.864 where a value of

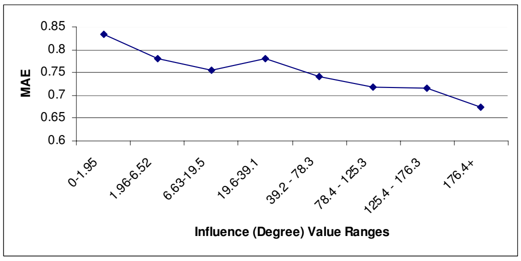

Fig. 1.9. *influence* MAE analysis.

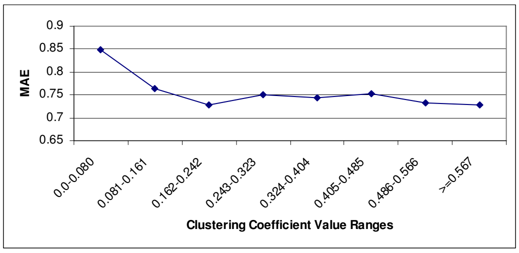

Fig. 1.10. *clustering-coeff* MAE analysis.

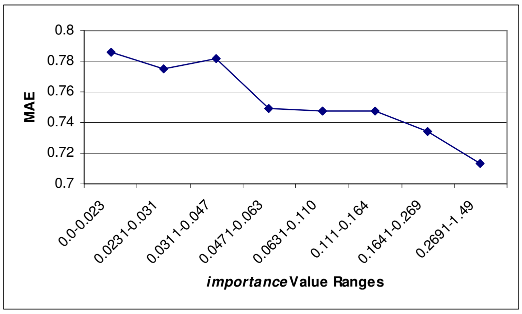

Fig. 1.11. *importance* MAE analysis.

0 means that none of the active user's neighbours are linked to each other (with a correlation value above 0.25). The MAE of the random group was 0.7479. The graph (Fig. 1.10) shows that as the *clustering-coeff* value increases towards 1 (i.e. the active user's neighbours are more similar to each other) the prediction accuracy very slightly improves. The poorest results

are seen for users who have very low clustering coefficient values.

The importance feature (*tf-idf*) was also analyzed for eight sets of users with values ranging from 0.015 to 1.485 (see Fig. 1.11). Results are poorer when a user has a low importance (*tf-idf*) value and results are better when a user has a high importance value. The MAE of the random group was 0.7255.

1.5.2. *Spreading activation*

Figure 1.12 illustrates the precision recall graph for the spreading activation approach and the traditional memory-based approach to collaborative filtering. Results were averaged over 100 runs. It can be seen that the spreading activation approach outperforms the traditional memory-based approach at all recall points other than the first. These results were shown to be statistically significant using a 2-tailed paired t-test at p-values $<$ 0.05.

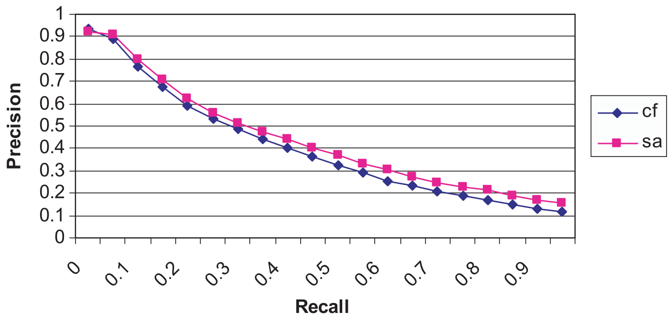

Fig. 1.12. Comparing spreading activation (sa) and traditional memory-based approaches (cf) to collaborative filtering.

This suggests that the graph-based representation and spreading activation approach give as good (and slightly better) performance as a traditional memory-based approach which has been shown to perform well. The advantage of the graph-based representation and spreading activation approach over other representations and approaches is their flexibility in allowing the incorporation of additional information. Given these results, future work can proceed in using the graph-based representation and spreading activation approach to incorporate information on user models.

1.6. Conclusions and Future Work

In this paper we have reviewed work in collaborative filtering, social networks and graph-based recommendation, highlighting the similarities between the work. We have defined a user model containing eight features of users that can be identified from a collaborative filtering dataset. We have shown how the prediction accuracy of a traditional memory-based collaborative filtering approach varies depending on the value of these features for certain users. This provides a first step at more personalized recommendations for users by providing some explanation for the relative good or poor performance of the collaborative filtering system (based on the values that users have for the identified features).

We have also shown some initial experimental evaluation of the usefulness of a graph-based representation of the collaborative filtering space using a spreading activation approach for recommendation.

We believe that more personalized and accurate recommendations can be obtained by incorporating the features identified in this paper into the graph models presented. Future work will explore these and other user features in more detail and will also consider the combination of these feature values. In addition, future work involves demonstrating that a graph-based representation of the collaborative filtering space allows the incorporation of these features and also other information on users, items and groups. This will strengthen the case for the application of graph-based recommendation algorithms.

References

1. U. Shardanand and P. Maes. Social information filtering: Algorithms for automating word of mouth. In *CHI '95*, pp. 210–217, (1995).
2. J. Barnes, *Social Networks*. (MA: Addison-Wesley, 1972).
3. H. Kautz, B. Selman, and M. Shah, Referral web: combining social networks and collaborative filtering, *Communications of the ACM*. **40**, 63–65 (March, 1997).
4. B. Krulwich and C. Burkey. The contactfinder: Answering bulletin board questions with referrals. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, (1996).
5. D. McDonald and M. Ackerman. Expertise recommender: a flexible recommendation system and architecture. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pp. 231–240, (2000).
6. M. Schwartz and C. Wood, Discovering shared interests using graph analysis, *Communications of the ACM*. **36**, 78 – 89 (August, 1993).

7. A. Vivacqua and H. Lieberman. Agents to assist in finding help. In *ACM Conference on Computers and Human Interface (CHI-2000)*, (2000).
8. D. Lemire, Scale and translation invariant collaborative filtering systems, *Information Retrieval.* **8**(1), 129–150, (2005).
9. B. Mirza, B. Keller, and N. Ramakrishnan, Studying recommendation algorithms by graph analysis, *Journal of Intelligent Information Systems.* **20**, 131 – 160, (March 2003).
10. J. Palau, M. Montaner, and B. Lopez. Collaboration analysis in recommender systems using social networks. In *Cooperative Information Agents VIII: 8th International Workshop, CIA 2004*, pp. 137–151, (2004).
11. A. Rashid, G. Karypis, and J. Riedl. Influence in ratings-based recommender systems: An algorithm-independent approach. In *SIAM International Conference on Data Mining*, (2005).
12. M. Balabanovic and Y. Shoham, Fab: Content-based, collaborative recommendation, *Communications of the ACM.* **40**(3), 66–72, (1997).
13. Y. Ding and X. Li. Time weight collaborative filtering. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM '05)*, pp. 299–300, (2005).
14. J. O'Donovan and B. Smyth. Trust in recommender systems. In *Proceedings of the 10th international conference on Intelligent user interfaces*, pp. 167–174, (2005).
15. Z. Huang, W. Chung, and H. Chen, A graph model for e-commerce recommender systems, *Journal of the American Society for Information Science and Technology.* **55**(3), 259–274, (2004).
16. J. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Uncertainty in Artificial Intelligence*, (1998).
17. K. Yu, X. Xu, J. Tao, M. Kri, and H.-P. Kriegel, Feature weighting and instance selection for collaborative filtering: An information-theoretic approach, *Knowledge and Information Systems.* **5**(2), (2003).
18. J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR*, pp. 230–237, (1999).
19. G. Karypis. Evaluation of item-based top-n recommendation algorithms. In *CIKM*, (2001).
20. M. R. J. Wang, A. de Vries. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *SIGIR*, pp. 501–508, (2006).
21. K. Cheung and L. Tian, Learning user similarity and rating style for collaborative recommendation, *Information Retrieval.* **7**, 395–410, (2004).
22. R. Jin, J. Chai, and L. Si. An automatic weighting scheme for collaborative filtering. In *SIGIR*, (2004).
23. A. de Bruyn, L. Giles, and D. Pennock. Offering collaborative-like recommendations when data is sparse: The case of attraction-weighted information filtering. In *Adaptive hypermedia and adaptive web-based systems*, (2004).
24. C. Aggarwal, J. Wolf, K.-L. Wu, and P. 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, pp. 201–212, (1999).
25. Z. Huang, H. Chen, and D. Zeng, Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering, *ACM Transactions on Information Systems*. **22**(1), 116–142, (2004).
26. P. Cohen and R. Kjeldsen, Information retrieval by constrained spreading activation on semantic networks, *Information Processing and Management*. **23**(4), 255–268, (1987).
27. F. Crestani and P. Lee, Searching the web by constrained spreading activation, *Information Processing and Management*. **36**, 585–605, (2000).
28. G.-R. Xue, S. Huang, Y. Y., H.-J. Zeng, Z. Chen, and W.-Y. Ma. Optimizing web search using spreading activation on the clickthrough data. In *Proceedings of the 5th International Conference on Web Information Systems*, pp. 409–414, (2004).
29. U. Mukhopadhyay, L. Stephens, M. Huhns, and R. Bonnell, An intelligent system for document retrieval in distributed office environments, *Journal of the American Society for Information Science*. **37**(3), 123–135, (1986).
30. J. Herlocker, J. Konstan, and J. Riedl, An empirical analysis of design choices in neighbourhood-based collaborative filtering algorithms, *Information Retrieval*. **5**, 287–310, (2002).
31. Z. Huang and D. Zeng. Why does collaborative filtering work? - recommendation model validation and selection by analyzing bipartite random graphs. In *15th Annual Workshop on Information Technologies and Systems*, (2005).
32. 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*, pp. 175–186. Chapel Hill, (1994).
33. J. Herlocker. *Understanding and Improving Automated Collaborative Filtering Systems*. Phd thesis, University of Minnesota, (2000).
34. M. McLaughlin and J. Herlocker. A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In *Proceedings of the 27th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 329–336, (2004).
35. D. Wu and X. Hu. Mining and analyzing the topological structure of protein-protein interaction networks. In *Symposium on Applied Computing*, pp. 185–189, (2006).
36. M. Calderon-Benavides, C. Gonzalez-Caro, J. Perez-Alcazar, J. Garcia-Diaz, and J. Delgado. A comparison of several predictive algorithms for collaborative filtering on multi-valued ratings. In *Proceedings of the ACM symposium on Applied computing*, pp. 1033–1039, (2004).