Identifying User and Group Information from Collaborative Filtering Datasets

Josephine Griffith[1], Colm O'Riordan[1], and Humphrey Sorensen[2]

[1] Dept. of Information Technology,
National University of Ireland, Galway, Ireland
josephine.griffith@nuigalway.ie, colm.oriordan@nuigalway.ie
http://www.it.nuigalway.ie/cirg/
[2] Dept. of Computer Science, University College Cork, Ireland
humphrey.sorensen@cs.ucc.ie

Abstract. This paper considers the implicit information that can be captured about users and groups from a collaborative filtering dataset with a view to creating a user model and using this model to explain the performance of a collaborative filtering approach. A number of user and group features are defined and the performance of a collaborative filtering system in producing recommendations for users with different feature values is tested. In addition graph-based representations of the collaborative filtering space are presented and these are used to define some of the user and group features as well as being used in a recommendation task.

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 personalised 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 [24], 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 now on building user models 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 occurs in many different fields.

A number of systems based on social networks and small world networks have been proposed for referral and recommendation [25], [16], [14], [23] and [13].

Other work linking social networks and collaborative filtering has viewed the collaborative filtering dataset as a social network with the aim of analysing properties of users and items to improve retrieval performance [18], [21], [15] and [20]. Aims other than solely improving retrieval performance are also explored in [18].

This paper considers the ways that recommender systems bring users and groups 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 the information which can be captured about users and groups given a collaborative filtering dataset and to provide a model that will represent features of users and groups. In this work, seven features which 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 view the collaborative filtering dataset as a graph or network. The seven features are then analysed 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 testing the accuracy of a collaborative filtering recommender system in providing predictions for the test users.

The user model defined can ultimately be used to help improve recommendation accuracy (by allowing the development of more personalised recommender algorithms) and also to maintain histories of users and groups in a collaborative filtering information space.

Section 2 presents related work in collaborative filtering, graph-based approaches to recommendation and social networks. Section 3 outlines the methodology, presenting the collaborative filtering approach and the graph models used as well as specifying the user and group features which are extracted from the collaborative filtering dataset. Section 4 discusses the experiments performed and the experimental set-up. Section 5 presents results and Section 6 presents conclusions, discussing the potential usefulness of the features and the approach and outlines future work.

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 sys-

tems generally make use of one type of information, that is, prior ratings that users have given to items, although some recent work has investigated the incorporation of additional information, particularly content. To date, application domains have predominantly been concerned with recommending items for sale (e.g. movies, books, CDs, restaurants) and with small amounts of text such as Usenet articles and email messages. The datasets within these domains will have different 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 $r_{i,j}$, 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 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 recommendation as well as in the social network analysis of collaborative filtering datasets [11], [21].

2.1 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 [1], spreading activation [11]).

Aggarwal et al. present *horting*, a graph-based technique where nodes represent users and directed 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.

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 [11]. 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 [10]. A bipartite graph is used with one set of nodes representing items and the second set of nodes representing users. The transactions of users and user feedback are modelled as binary weighted edges connecting the nodes between the two sets. 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 [5] and more recently in the domain

of collaborative filtering [10]. Spreading activation approaches have also been used to integrate sources of evidence and information [26], [6].

2.2 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 rated and/or the actual ratings that users have given to these items [2], [20], [21]. Rashid et al. state that "In contrast to other social networks, recommender systems capture interactions that are *formal*, *quantitative*, and *observed*" [21].

A social network can be defined as a network (or graph) of social entities (e.g. people, markets, organisations, countries), where the links (or edges) between people 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 [23]; co-occurrence of names on WWW pages [13]; co-use of documents by users [19]; and matching user models and profiles [25].

Palau et al. represent the agents in a multi-agent 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 [20].

Rashid et al. [21] view the collaborative filtering dataset as a social network where users are linked to each other. The aim is to find *influential users*. Lemire also considers the 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 penalise users with a high number of ratings [15]. 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 co-ratings of the same items [18]. They define a *hammock jump* as a connection between two users in the network that will exist if the users have at least w items co-rated (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 [8].

In [12], a graph-based representation is used to analyse 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). The features analysed were group features based on clustering coefficient measures and user/item pair clustering distribution measures.

3 Methodology

In this paper the focus is to extract implicit user and group information available from the collaborative filtering dataset and 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.

The user model consists of a septet containing, for each user, the values for seven identified features (six user features and one group feature). Each of the features is individually analysed by considering a set of users with different values for the feature (e.g. one set of users may have rated around the maximum number of items rated, another set of users may have rated around the average number of items rated). For each feature, test users from each set are chosen as the active users of a collaborative filtering recommender system, i.e. these are the users for which a recommendation is sought. Results are analysed and the accuracy of the recommender system in providing recommendations for each set of users is compared (e.g. the set of users who rated around the maximum number of items may receive better predictions than the set of users who rated around 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 septet values for that user.

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

The collaborative filtering system which is used to provide recommendations to different sets of users is described in Section 3.1. Two graph-based representations of the collaborative filtering problem are presented in Section 3.2. Each of the seven features of the user model are defined in Section 3.3.

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 previously not 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 [3]. 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 have rated highly 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.) [22]. In this work, similar users are found using the Pearson Correlation coefficient formula:

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

where $corr_{a,u}$ is the correlation value between user a and user u (a value in the range [-1, 1]), $r_{a,i}$ 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*) as already mentioned is used in the Pearson Correlation calculation based on the number of items that users have rated in common (co-rated items) [8]. 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 here involves multiplying $corr_{a,u}$ 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 here as:

$$\frac{cr_{a,u}}{2 \times average}$$

where $cr_{a,u}$ 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" neighbours of a user are selected using a low neighbour selection threshold, with any correlation value greater than 0.01 being considered. Although Breese [3] found that users with high correlation values ($>$ 0.5) were more valuable in providing recommendations, work by Herlocker [9] 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 with no threshold (using all correlation values $>$ 0) always outperformed experiments with higher thresholds [9].

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

$$pred_{a,i} = \frac{\sum_{u=1}^{n}(r_{u,i} - \bar{r_u}) \times corr_{a,u}}{\sum_{u=1}^{n} corr_{a,u}}$$

where $pred_{a,i}$ is the prediction for item i for user a, n is the number of neighbours 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 $corr_{a,u}$ is the correlation value between users a and u.

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 here.

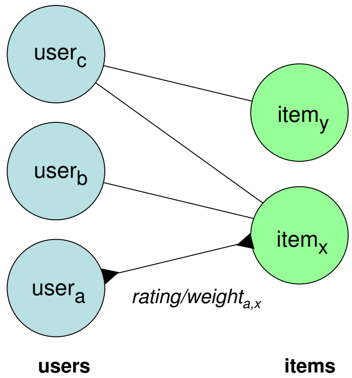

Fig. 1. Graph Representation of Users, Items and Ratings

In the first representation (see Fig. 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 value users have given to items, this graph is a direct mapping of the data in the matrix representation to a graph representation of the data.

The second representation (see Fig. 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.

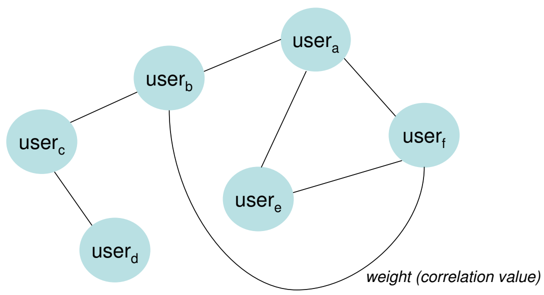

Fig. 2. Graph Representation of Users and their Similarity

Note that the two representations can be combined into a single graph representation. Currently in these representations, the information on user and group features is not represented explicitly. To represent this information explicitly, additional edges can

be added to the graph that 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 is augmented such that three weighted edges connect nodes: one undirected 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 (Fig. 3 shows a portion of the graph from Fig. 1).

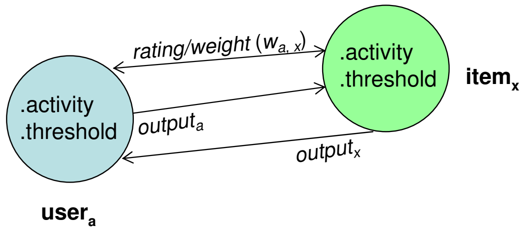

Fig. 3. Extended Graph Representation of a single User and Item Node

The activity of a user or item node a, for N nodes connected to the node a with non-zero weight, is calculated by:

$$activity_a = \sum_{i=1}^{N} output_i w_i$$

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 node a. The output, $output_a$, of a user or item node is calculated using a threshold function:

$$output_a = \begin{cases} activity_a & \text{if } activity_a > \tau \\ 0 & \text{otherwise.} \end{cases}$$

where the threshold function uses the node's activity and a threshold value, τ, where each node may have its own threshold value depending on its importance.

Spreading activation involves moving activation from one set of nodes to a second set of nodes. 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 are:

1. Hop 1: Calculate the activities of all item nodes connected, with non-zero 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 non-zero weight, to item nodes where the item nodes have non-zero 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 non-zero weight, to user nodes where the user nodes have non-zero 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 neighbourhood of the original active user node. The third hop, from user nodes to item nodes, provides item recommendations for the active user.

3.3 User and Group Features

A user model is defined which consists of seven 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 user a that user a liked and is calculated by:
$$\frac{num_{liked}}{rated}$$
where num_{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 (e.g. if the rating range is [1,5] a liked item is an item that receives a value of 4 or 5 [17]).
3. *disliked* is the percentage of items rated by user a that user a disliked and is calculated by:
$$\frac{num_{disliked}}{rated}$$
where $num_{disliked}$ 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 score 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 [21] and in [18], *influence* is defined here 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. 2). Degree centrality is then measured by counting

the number of links a node has to other nodes. A node can be considered central if it has a higher degree than any of the other nodes whereas a node with a low degree is isolated from most of the other users in the network [7].

7. *clustering-coeff*, the final feature, 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. 2. For example, if none of a's neighbours are connected to each other, the clustering coefficient is 0 whereas if this sub-graph has a clustering coefficient of 1 then all of a's neighbours are connected to each other.
   The clustering coefficient can be calculated by:

$$\frac{actual}{possible}$$

   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. As already described, only user nodes who are connected to each other with a correlation value greater than 0.25 are considered. In this representation, edges are undirected so therefore the total number of possible links that can exist between n nodes is:

$$possible = \frac{(n^2 - n)}{2}$$

   Thus the formula for the clustering coefficient becomes:

$$\frac{actual \times 2}{(n^2 - n)}$$

   Considering the graph shown in Fig. 2 with the active user being $user_a$ who has three neighbours (b, e and f): user e is connected to user f and user f is connected to user b (there is no edge between user b and user e). Therefore the number of actual links is 2 and the number of possible links (given that there are three neighbour nodes) is 3. The clustering coefficient for this group is therefore 0.66.

In addition to the group feature specified, a number of other group features can be extracted from the collaborative filtering dataset. In general, groups can be considered from two perspectives: when a is the active user, the group model tells us about the users which influence a; when a is a member of some active user's group the group model tells us about the influence a exerts on other users in that group. Using the graph representation in Fig. 1, a group for an active user a can be defined as the user nodes which are two steps, or links, away from the user node a, possibly where links above a certain threshold weight are only considered. Some work has also considered nodes which are more than two links away from the active node as neighbour nodes (e.g. when considering transitive relationships [1], [10], [11]). Future work will explore these other group features in more detail.

4 Experiments

This section presents details of the experiments performed using the collaborative filtering approach and graph-based approaches outlined in the previous section. The first set of experiments involve analysing each of the seven features identified in the previous section using a collaborative filtering approach. The final experiment involves testing the performance of the graph-based representation in Fig. 3 using a spreading activation approach to collaborative filtering.

4.1 User Model Features

The main experiments involve checking the relative performance of a collaborative filtering approach using different sets of users for all seven features where a set consists of users who have around the same value for an identified feature. The aim is to see which sets of users will be more likely to have better or worse predictions (measured using the mean absolute error (MAE) metric). It is also of interest to see if there is some combination of feature values which result in "strong" or "weak" groups with respect to recommendation accuracy. Future work, possibly involving a machine learning approach, will consider this.

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. [0,1] for the *liked* feature) is broken into regular intervals (typically 7 to 10 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) must be at least 30.

For testing, 30 users are chosen randomly from each set as the test users and 10% of their ratings for items are removed as the test items (i.e. the system should return predictions for these items). MAE results are averaged over 50 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).

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 and group 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 [4], [9]. Also it is quite similar to the spreading activation approach outlined. The important difference

between the representations and approaches is 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 non-zero 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 non-zero 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 outputting a value.

5 Results

5.1 User Model Features

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

Fig. 4 shows the MAE results when the *rated* feature was analysed for seven sets of users. The *rated* value ranges from 0 to 668. The users in the first set (0-24 interval) have rated 0-24 items; the users in the second set (25-49 interval) have rated 25-49 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 0.8836.

As expected, the worst MAE value for any set was for 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 users who have rated around the maximum number of items would have the best MAE value, this was not necessarily the case.

Fig. 5 and Fig. 6 shows the MAE results when the *liked* and *disliked* features were analysed for seven sets of users. Both ranges are from 0 to 1 where for the *liked* feature

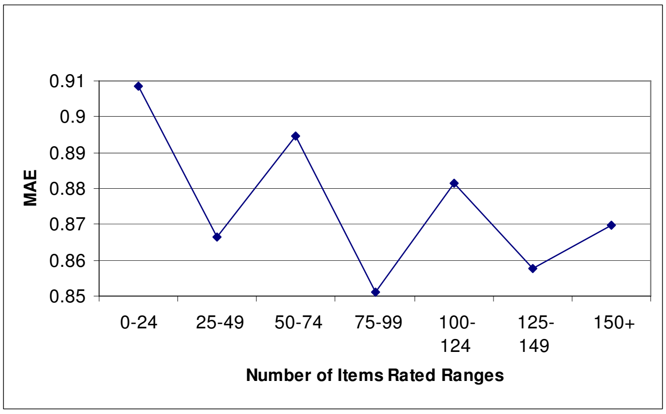

Fig. 4. *rated* MAE Analysis

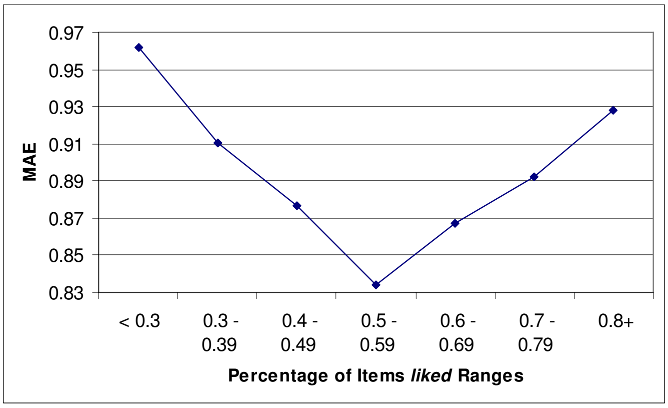

Fig. 5. *liked* MAE Analysis

a value of 1 indicates that a user liked all items that they rated and a value of 0 indicates that a user liked none of the values they rated. Similarly, a value of 1 for the *disliked* feature indicates that a user disliked all items that they rated and a value of 0 indicates that a user disliked none of the values they rated. A random group of 30 users (with varying *liked* and *disliked* values) were also chosen and had an average MAE value of 0.8999 for the random *liked* group and an average MAE value of 0.900 for the random *disliked* group.

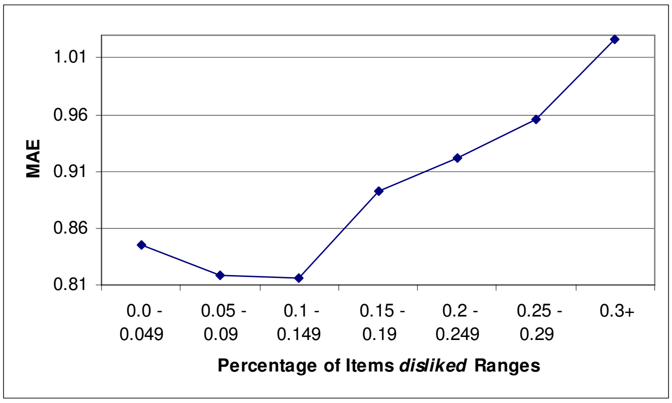

Fig. 6. *disliked* MAE Analysis

Fig. 7 shows the MAE results when the *avg-rating* feature was analysed for eight sets of users. The *avg-rating* value ranges from 1.509 to 4.857. The MAE for 30 randomly chosen users was 0.8704.

For these three features (*liked*, *disliked* and *avg-rating*), the lowest MAE values correspond to the ranges that contain the largest number of users. For example, the majority of users had an average rating in the range from 3.5 to 3.74 and it can be seen from the graph that this group had the lowest MAE value. Therefore, for these three features the actual feature value was not as important as the number of users who shared this value with the active user.

Fig. 8 shows the MAE results when the standard deviation feature (*std-dev*) was analysed for ten sets of users. The *std-dev* value ranges from 0 to 0.239. The users with low standard deviation (< 0.06) exhibited the best MAE value (0.7958 in comparison to the MAE of the randomly selected group which was 0.8669) while the users with the highest standard deviation (with variance above 0.22) had the worst MAE (1.023). This suggests that better recommendations can be found for users with lower variance in their ratings.

Fig. 9 shows the MAE results when the *influence* feature was analysed for ten sets of users. The $influence$ value ranges from 0 to 389 where a value of 0 means that a user has no neighbours. As expected, the users with fewest neighbours (0-24 range) have the

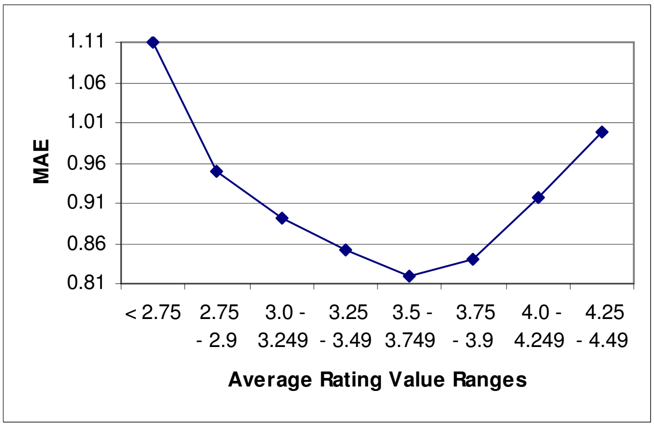

Fig. 7. *avg-rating* MAE Analysis

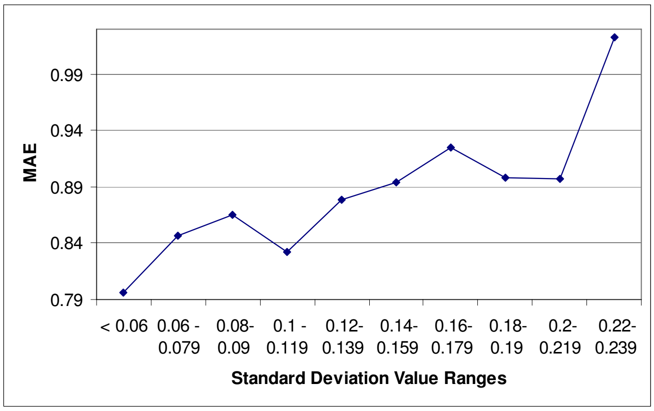

Fig. 8. *std-dev* MAE Analysis

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 0.8845.

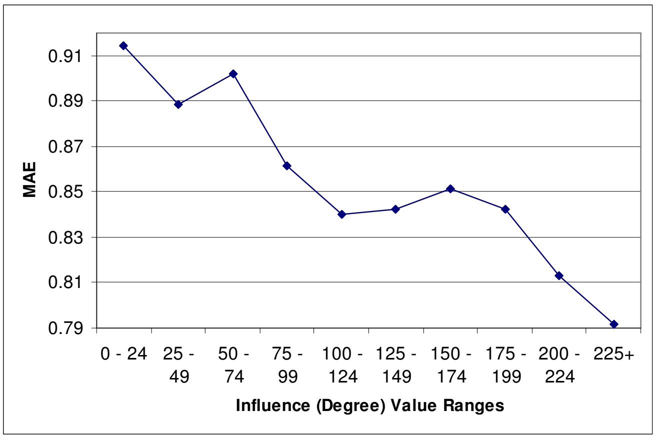

Fig. 9. *influence* MAE Analysis

Fig. 10 shows the MAE results when the clustering coefficient feature (*clustering-coeff*) was analysed for ten sets of users. The value ranges from 0 to 1 where a value of 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 0.8699. The graph 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 improves.

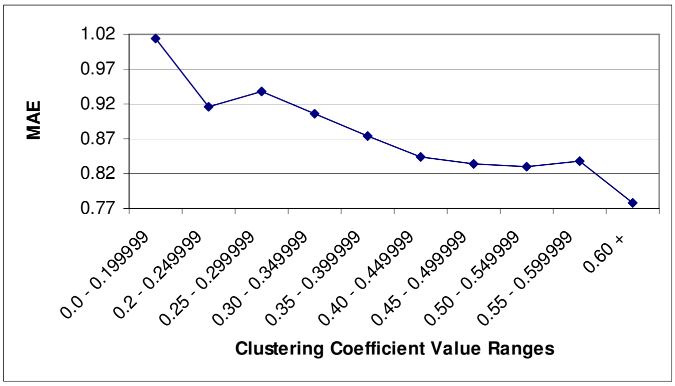

Fig. 10. *clustering-coeff* MAE Analysis

5.2 Spreading Activation

Fig. 11 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.

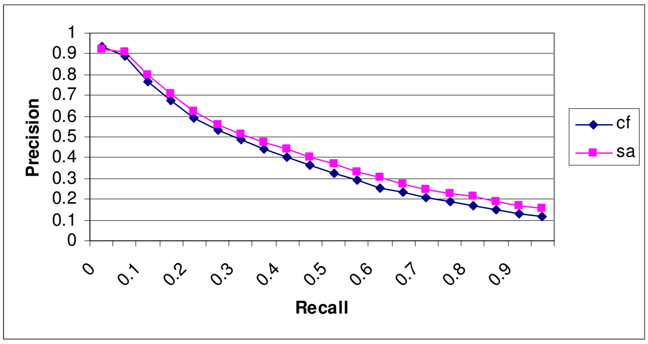

Fig. 11. Comparing Spreading Activation (sa) and Traditional Memory-Based Approaches (cf) to Collaborative Filtering

This suggests that the graph-based representation and spreading activation approach give as good (and slightly better) performance than a traditional memory-based approach which has been shown to perform well. The important difference between these representations and approaches is the flexibility of the graph-based representation and spreading activation approach 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 and group models.

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 seven features of users and groups 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 personalised 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 feature).

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 for more personalised and accurate recommendations, the features identified in this paper can be incorporated into the graph models presented. Future work will explore these and other user and group features in more detail and will also consider the combination of these feature values (possibly by use of a machine learning approach). 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.

7 Acknowledgements

The authors would like to acknowledge the contributions of Patrick Flood and David Wall for their experimental implementation.

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. J. Barnes. *Social Networks*. MA: Addison-Wesley, 1972.
3. 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.
4. M.L. Calderon-Benavides, C.N. Gonzalez-Caro, J. Perez-Alcazar, J.C. Garcia-Diaz, and J. Delgado. A comparison of several predictive algorithms for collaborative filtering on multi-valued ratings. In *Proceedings of the 2004 ACM symposium on Applied computing*, pages 1033–1039, 2004.
5. P. Cohen and R. Kjeldsen. Information retrieval by constrained spreading activation on semantic networks. *Information Processing and Management*, 23(4):255–268, 1987.
6. F. Crestani and P.L. Lee. Searching the web by constrained spreading activation. *Information Processing and Management*, 36:585–605, 2000.
7. L. Freeman. Centrality in social networks: Conceptual clarification. *Social Networks*, 1:215–239, 1979.
8. 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.
9. J.L. Herlocker. *Understanding and Improving Automated Collaborative Filtering Systems*. Phd thesis, University of Minnesota, 2000.
10. 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.
11. 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.

12. 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*.
13. H. Kautz, B. Selman, and M. Shah. Referral web: combining social networks and collaborative filtering. *Communications of the ACM*, 40:63–65, March 1997.
14. B. Krulwich and C. Burkey. The contactfinder: Answering bulletin board questions with referrals. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, 1996.
15. D. Lemire. Scale and translation invariant collaborative filtering systems. *Information Retrieval*, 8(1):129–150, 2005.
16. 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*, pages 231–240, 2000.
17. M.R. McLaughlin and J.L. 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*, pages 329 – 336, 2004.
18. B. Mirza, B. Keller, and N. Ramakrishnan. Studying recommendation algorithms by graph analysis. *Journal of Intelligent Information Systems*, 20:131 – 160, March 2003.
19. U. Mukhopadhyay, L.M. Stephens, M.N. Huhns, and R.D. Bonnell. An intelligent system for document retrieval in distributed office environments. *Journal of the American Society for Information Science*, 37(3), 1986.
20. 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*, pages 137–151, 2004.
21. A.M. Rashid, G. Karypis, and J. Riedl. Influence in ratings-based recommender systems: An algorithm-independent approach. In *SIAM International Conference on Data Mining*, 2005.
22. 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.
23. M.F. Schwartz and C.M. Wood. Discovering shared interests using graph analysis. *Communications of the ACM*, 36:78 – 89, August 1993.
24. 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.
25. A. Vivacqua and H. Lieberman. Agents to assist in finding help. In *ACM Conference on Computers and Human Interface (CHI-2000)*, 2000.
26. 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*, 2004.