Using User Model Information to support Collaborative Filtering Recommendations

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
[2] Dept. of Computer Science, University College Cork, Cork, Ireland
sorensen@cs.ucc.ie

Abstract. This paper considers some of the information that can be captured about users and groups from a collaborative filtering dataset and uses this information to provide a more personalised recommendation experience. The idea is that features of users are used to provide evidence as to whether good or poor recommendations are likely for a given user. This evidence will be returned to a user together with a set of recommendations and will give the user more information with which to judge if the recommendations are likely to be accurate or of interest to the user, e.g., on occasion the user may choose to discard the recommendation if it has been produced by the system using a "weak" set of evidence.

1 Introduction

Collaborative filtering systems automate the "word of mouth" process that commonly occurs within social networks [16], i.e. people will seek recommendations from people with whom they share similar preferences in an area. Within the field of collaborative filtering many models and techniques have been proposed, tested and compared. Additional features of users, items and the recommendation task have also been considered (e.g., product information, demographic information, time, trust). Studies involving new models, techniques and incorporating additional information often have different foci where the aim has not always been to improve performance. For example, various studies have focused on dealing with scalability issues [5], dealing with the issue of dataset sparseness [7], including additional information [1], incorporating trust [12] and dealing with shilling attacks [13].

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. We believe that such analysis can also be used to provide an explanation of how accurate the system predictions are. Mirza et al. [11] list four desirable aspects of recommendation:

1. "Recommendation is an indirect way of bringing people together."
2. "Recommendation, as a process, would emphasize modelling connections from people to artifacts, besides predicting ratings for artifacts."
3. "Recommendations should be explainable and believable."
4. "Recommendations are not delivered in isolation, but in the context of an implicit/explicit social network."

The approach taken here concentrates on making recommendations more "explainable and believable" by providing users with an information relating to whether a recommendation may be accurate or not. The motivation for this work is that although users are often clustered into groups based on finding "similar users" and much is known about the effect of various user, item and group features on the accuracy of predictions, this information has not been used to support the output of recommendation systems. Although it is unlikely that sufficiently clear evidence, and thus an explanation, can be found to support all recommendations and users, it could still be useful to highlight the cases when a set of recommendations are formed using particularly poor evidence or particularly strong evidence.

In this work, six features that can be extracted from the collaborative filtering dataset are firstly identified, defined and analysed with respect to their effect on recommendation accuracy. Some of these features are particular to the recommendation task while some features use measures from social network theory and information retrieval. Each feature can provide one piece of "evidence" if its values are above or below a certain threshold. Thresholds are chosen based on the analysis of the features effect on prediction accuracy. The more positive or negative pieces of evidence that exist for a given user, the more likely that the recommendation results will be accurate (for positive evidence) or inaccurate (for negative evidence).

2 Related Work

Collaborative filtering techniques produce recommendations for some active user using the ratings of other users, where these users have similar preferences to the active user. Collaborative filtering datasets 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 ratings only exist 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. The problem space can equivalently be viewed as a graph where nodes represent users and items, and nodes can be linked by weighted edges in various ways (e.g., user-item links; user-user links).

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 [2] and [18] an inverse user frequency weighting was applied to

all ratings where items that were rated frequently by many users were penalised by giving the items a lower weight. In [6] and [18] 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 [10] (using a row normalisation) and in [8] (using a probabilistic framework). More recent work in [3], [9] and [14] involve learning the optional weights to assign to items. In [12] 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 [4] 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. There may be a number of reasons for this including the fact that the dataset is sparse and also that the data may not always be correct. Even if the data is correct the underlying preferences that the data "describes" may not always be consistent as user tastes and opinions may change over time.

3 Methodology

In this paper, the focus is to extract implicit user and group information available from the collaborative filtering dataset, and to form a user model for each user and use this model to provide evidence as to how likely the system is to produce good or poor recommendations for a user. The implicit information extracted from the dataset is based on simple features which can be extracted from any recommendation dataset (e.g. number of items rated, average rating value) as well as extracting features which are based on measures from social network theory (degree, clustering coefficient) and from information retrieval (*tf-idf*).

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 [2]. Generally, traditional memory-based collaborative filtering approaches contain three main stages (for some active user a):

- Find users who are similar to user a (the neighbours of a).
- Select the "nearest" neighbours of a, i.e. select the most similar set of users to user a.
- 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.) [15]. In this work, similar users are found using the Pearson correlation coefficient formula.

3.2 User and group features

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

- *rated* is the number of items rated by the user a.
- *avg-rating* is the average rating value given to items by the user a.
- *std-dev* is the standard deviation of the ratings of user a.
- *influence* is a measure of how influential a user is in comparison to other users. As also considered in [14] and in [11], *influence* is defined in this work by using measures from social network theory. In particular, the idea of 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 with users linked if their Pearson correlation value is above 0.25). 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.
- *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. For example, if none of user 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 user a's neighbours are connected to each other.

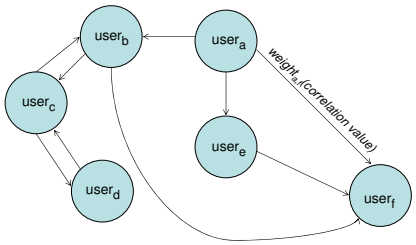

Fig. 1. Graph representation of users and their similarity.

The clustering coefficient is calculated by dividing the number of actual links by the number of possible links between neighbour nodes for all neighbour nodes with degree greater than 1. Only user nodes that are connected to

each other with a correlation value greater than 0.25 are considered neighbour nodes. 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. Therefore 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 subgroups (small values of n) will have high clustering coefficients and therefore comparisons using clustering coefficient values may not always be meaningful. To overcome this (**??**) is extended to also include the active user in the calculation [17]. 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):

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

Considering the graph shown in Fig. 1 with the active user being $user_a$ who has three neighbours (b, e and f): user e is connected to user f and user b is connected to user f. Therefore the number of actual links is 2. The degree of the active node a is 3, therefore the clustering coefficient for this group is 0.42.

– *importance* Some collaborative filtering weighting schemes incorporate the idea from Information Retrieval of a *term frequency, inverse document frequency* (*tf-idf*) weighting [10],[8]. The idea in information retrieval 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:

$$w_i = \frac{1}{M} \times \sum_{j=1}^{M} \left(1 + \log \frac{n}{n_j}\right) \tag{2}$$

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.

3.3 User and group features and their effect on accuracy

In this section the relative performance of a collaborative filtering approach is tested using different sets of users for each of the six 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). For each feature, the range of values for that feature (e.g. [0,1] for the *clustering-coeff* 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) must be at least 30 and it is usually around 100.

For testing, a standard subset of the Movie Lens dataset is considered. 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).

Fig. 2 shows the MAE results when the *rated* feature was analysed for eight 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 (25-30 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 close to the maximum number of items have the best MAE values.

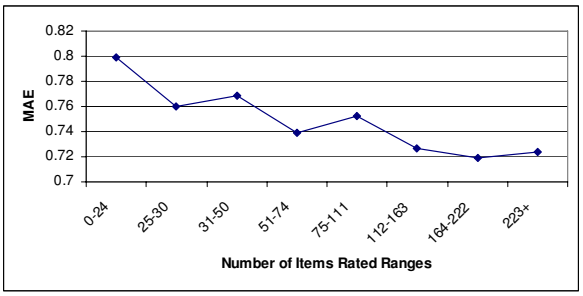

Fig. 2. *rated* MAE analysis.

Fig. 3 shows the MAE results when the *avg-rating* feature was analysed for eight sets of users. 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 (> 4.85) have the best MAE.

Fig. 4 shows the MAE results when the standard deviation feature (*std-dev*) was analysed for eight sets of users. The *std-dev* value ranges from 0 to 1.718. The users with low standard deviation (< 0.06) exhibited the best MAE value

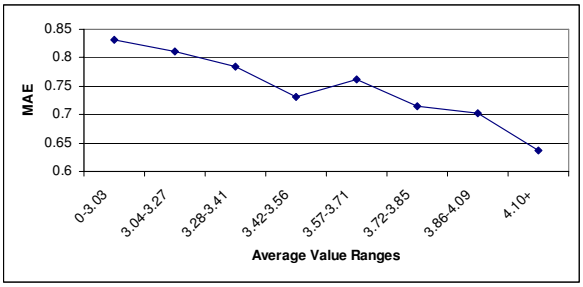

Fig. 3. *avg-rating* MAE analysis.

(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.

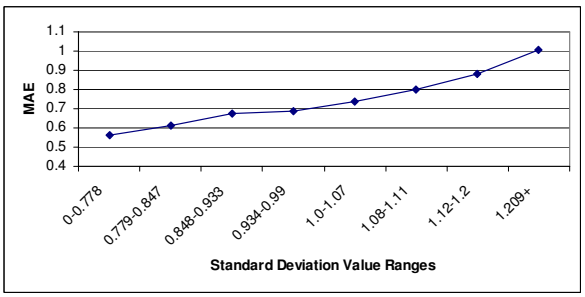

Fig. 4. *std-dev* MAE analysis.

Fig. 5 shows the MAE results when the *influence* feature was analysed for eight sets of users. 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 analysed for eight sets of users with values ranging 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). 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 user importance feature (*tf-idf*) was also analysed for eight sets of users. Results were poorer when a user has a low *tf-idf* weighting value and results are better when a user has a high *tf-idf* weighting value.

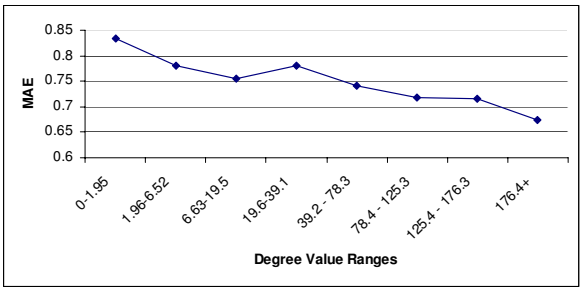

Fig. 5. *influence* MAE analysis.

3.4 Forming the evidence

Based on the graphs in the previous section thresholds were set for each of the features where, for each feature, a *evid-feature* parameter is set to 0 if the evidence is poor; *evid-feature* was set to 1 if the evidence is good; and *evid-feature* was set to -1 otherwise, i.e., in this case it is not possible to clearly say whether this feature will have an affect on recommendation accuracy. Therefore some users may receive item recommendations along with an indication that the recommendations have been found with mostly poor evidence (where *evid-feature* is mostly 0) or mostly strong evidence (where *evid-feature* is mostly 1) or there may be no explanation if the *evid-feature* values are mostly -1 or there is a mixture of values for *evid-feature*.

4 Experiments and Results

The testing methodology involved checking if the evidence generated by the system, in terms of poor and good evidence, is supported by the MAE result for that user and per run, calculating:

- the number of users correctly and incorrectly identified as having good or poor evidence.
- the number of users not identified as having good or poor evidence.

We suggest that it is equally important to know when predictions are poor (have been formed using weak evidence) as to know when predictions are strong (have been formed using stronger evidence). In order to test whether the system had correctly identified users with good and poor evidence, the mean absolute error (MAE) metric was used to analyse results where the MAE of each user was taken as an indication of whether there is sufficient evidence in the dataset to form good recommendations. A user with MAE below the average MAE was considered to have sufficient (good) evidence and a user with MAE above the average was considered to have poorer evidence.

In the experiments performed, the Movie Lens dataset is used with 10% of users chosen randomly as test users and 10% of their items chosen randomly as

test items. In addition to returning recommendations for all users and items in the test set the system indicates whether poor or good evidence is available for the user. Fig. 6 summarises the comparison of the MAE values with the values of the *evid-feature* for seven sample users. These users were correctly identified as having good and poor evidence.

		evid_feature						
evidence	User_ID	ratings	avg_rating	std-dev	influence	clus-coeff	importance	MAE
Good	454	1	-1	1	1	-1	-1	0.54
Good	266	1	-1	1	1	-1	-1	0.51
Good	757	1	-1	-1	1	-1	1	0.58
Good	199	1	-1	1	1	-1	-1	0.57
Poor	514	0	0	-1	0	-1	-1	0.81
Poor	178	0	0	0	-1	-1	-1	1.69
Poor	723	-1	0	0	-1	0	-1	0.96

Fig. 6. Sample users with *evid-feature* values and MAE.

The following summarises the results for one run: 26 users were not identified as having poor or good evidence. 36 users were identified as having good evidence (37.5% of the users in the test set). Of these users, 6 users were incorrectly classified as having good evidence (16%) and the remaining were correctly classified. Considering the top 40 best results (best MAE for 40 users), 10 users within this top-40 were not identified as having good evidence, 3 users were identified as having poor evidence and the remaining were correctly identified as having strong evidence.

34 users were identified as having poor evidence (35% of dataset). Of these 34 users, 8 were incorrectly classified as having poor evidence and the remaining were correctly classified. Of the 38 poorest results returned by the system (with MAE values greater than the average MAE), 8 users were not identified as having poor evidence, 5 users were identified as having good evidence and the remaining 25 users were correctly identified as having poor evidence.

5 Conclusions and Future Work

In this paper the idea proposed is that a collaborative filtering system often has the information available to provide an explanation as to whether the recommendations produced by the system are likely to be poor or not. A user is thus provided with more information with which to judge the recommendations presented. The information used to obtain these explanations is already available in the collaborative filtering dataset and some of the information is calculated as part of the recommendation process. This information includes: the number of ratings given by a user, the average rating value given by a user, the standard deviation of user ratings, the number of neighbours a user has, the clustering coefficient value of a user and the importance of a user.

Results show that a large percentage of users are correctly identified as having poor or good evidence. However further sample runs need to be performed and

further analysis performed for the cases that were incorrectly identified. Future work will also consider the parameters and thresholds in more detail.

References

1. M. Balabanovic and Y. Shoham. Fab: Content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.
2. J. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Uncertainty in Artificial Intelligence*, 1998.
3. K. Cheung and L.F. Tian. Learning user similarity and rating style for collaborative recommendation. *Information Retrieval*, 7:395–410, 2004.
4. A. de Bruyn, L. Giles, and D.M 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.
5. T. George and S. Merugu. A scalable collaborative filtering framework based on co-clustering. In *Fifth International Conference on Data Mining*, 2005.
6. J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR*, pages 230–237, 1999.
7. R. Hu and Y. Lu. A hybrid user and item-based collaborative filtering with smoothing on sparse data. In *16th Intl. Conf. on Artificial Reality and Telexistence*, 2006.
8. M. Reinders J. Wang, A. de Vries. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *SIGIR*, pages 501–508, 2006.
9. R. Jin, J.Y. Chai, and L. Si. An automatic weighting scheme for collaborative filtering. In *SIGIR*, 2004.
10. G. Karypis. Evaluation of item-based top-n recommendation algorithms. In *CIKM*, 2001.
11. B. Mirza, B. Keller, and N. Ramakrishnan. Studying recommendation algorithms by graph analysis. *Journal of Intelligent Information Systems*, 20:131 – 160, March 2003.
12. J. O'Donovan and B. Smyth. Trust in recommender systems. In *Proceedings of the 10th international conference on Intelligent user interfaces*, pages 167–174, 2005.
13. M.P. O'Mahony, N. Hurley, and G.C.M. Silvestre. Detecting noise in recommender system databases. In *Proceedings of the 11th International conference on intelligent user interfaces*, pages 109–115, 2006.
14. 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.
15. 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.
16. U. Shardanand and P. Maes. Social information filtering: Algorithms for automating word of mouth. In *CHI '95*, pages 210–217, 1995.
17. D.D. Wu and X. Hu. Mining and analyzing the topological structure of protein-protein interaction networks. In *Symposium on Applied Computing*, pages 185–189, 2006.
18. K. Yu, X. Xu, J. Tao, M.E. Kri, and H.-P. Kriegel. Feature weighting and instance selection for collaborative filtering: An information-theoretic approach. *Knowledge and Information Systems*, 5(2), 2003.