Collaborative Filtering using a Spreading Activation Approach

Josephine Griffith*, Colm O'Riordan*, Humphrey Sorensen**
*Department of Information Technology, NUI, Galway
**Computer Science Department, University College Cork

Abstract

This paper describes a spreading activation approach to collaborative filtering where the collaborative filtering space is represented by a network of two sets of nodes, user nodes and item nodes, with bi-directional connections between the user nodes and the item nodes. Activation is spread from an active user node to item nodes and from item nodes to user nodes and so on in order to find groups of users with similar preferences and to recommend items to users. The approach is compared with a traditional collaborative filtering approach that uses Pearson Correlation to find similar users.

1. Introduction

Given a set of users, a set of items, and a set of ratings by users on these items, collaborative filtering systems attempt to recommend items to users based on the prior ratings that users have given to items. The problem space can be viewed as a matrix consisting of the ratings given by each user to items in a collection, i.e., the matrix consists of a set of ratings $u_{i,j}$, corresponding to the rating given by user i to an item j. 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 network where users and items are represented by nodes and the links between user nodes and item nodes represents a rating for the item by the user. Several researchers have adopted graph representations in order to develop recommendation algorithms [6], [8], [9]. A variety of graphs have been used (e.g. directed and two-layer) and a number of graph algorithm approaches have been adopted (e.g. horting [1] and spreading activation [5]).

In this paper, details are presented of another graph and spreading activation approach in a collaborative filtering domain. The approach and results are compared with a traditional collaborative filtering approach that uses Pearson Correlation.

Section 2 details the network representation and Section 3 compares the network representation and spreading activation approach with a traditional collaborative filtering approach and outlines the experiments performed and the testing methodology used. Section 4 presents results and conclusions are presented in section 5.

2. Network Representation and Spreading Activation Approach

The collaborative filtering problem space is frequently viewed as a matrix consisting of the ratings given to a set of items by a set of users. Figure 1 illustrates how the problem space can be represented as a network with bidirectional-weighted edges connecting the set of user nodes and the set of item nodes. The network consists of:

- A set of user nodes, which are not directly connected to each other. Each user node has an associated activity, output and threshold. For all but the starting node (the active user) the activity and output of all other user nodes is initially zero.
- A set of item nodes, which are not directly connected to each other. Each item node has an associated activity, output and threshold, all of which are zero initially.
- User and item nodes are connected via weighted edges. The weights on the edges represent ratings given to items by users.
- A node's output is a function of its activity.

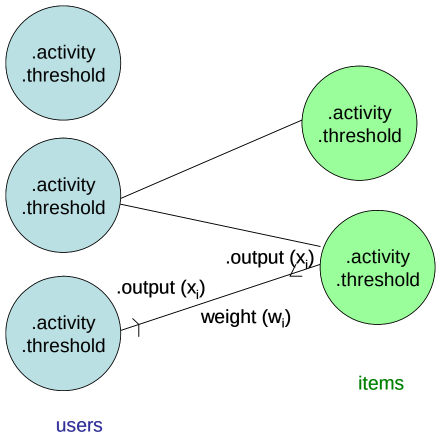

Fig. 1 Network Representation

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

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

where x_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 of a user or item node is calculated based on that node's activity and a threshold value. If the activity of a node is greater than the threshold value, the node outputs a value. The node does not output a value if the activity of the node is less than or equal to the threshold. In the initial experiments presented in this paper, no thresholds are used and a node's output is its activity. As a result, both negative and positive values are allowed as outputs.

Spreading activation involves moving activation from one set of nodes to a second set of nodes. The terminology of a hop[1] 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 or item set, updating the associated activities and outputs of the nodes.

The initial hop performed is from the active user node to all item nodes connected, with non-zero weight, to the active user node. For all item nodes connected to the active node, the item node's activities and outputs are calculated and the nodes output a value. The second hop is from a set of item nodes to a set of user nodes, updating activities and outputs of user nodes. 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. Further hops are also explored with the motivation of highlighting "transitive" neighbours and items.

Currently, outputs and activities are calculated using the same formula irrespective of whether it is the first or ninth hop. Future experiments will investigate the effect of including a "decay factor" such that as the number of hops increases, the influence is decreased. Essentially this means that items and users "nearer" (less hops away) from the active user are more important (e.g. the users who have rated the same items as the active user are more important).

3. Comparing the Spreading Activation Approach and a Traditional Collaborative Filtering Approach

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*.

[1] The terminology of a hop is also used in [8] but the definition of a hop is not the same as the definition used here.

Using the network representation and the spreading activation approach presented in Section 2, a collaborative filtering approach is defined consisting of three stages corresponding to the three stages in the traditional memory-based collaborative filtering approach. The stages in the spreading activation approach are:

1. Find neighbours of the active user *a* with two hops of the network. The first hop, from item nodes to user nodes, will initially only activate those items that the active user has rated. The second hop activates the user nodes that are connected to the item nodes highlighted in the first hop.
2. After two hops, some user nodes will have non-zero activity. These are the neighbours of user *a*. A ranked list of these neighbours, based on the activity value, exists and this can be used to select neighbours over a certain activity value or to select the *top-N* neighbours.
3. A third hop, from user nodes to item nodes, activates items not rated by user *a*. These items are recommended to the user *a* if the activity is sufficiently high (using some threshold) or the *top-N* items are recommended.

In the experiments performed, a standard subset of the Movie Lens dataset is considered that contains the ratings of 943 users for 1682 movies. Ratings are in the range 1-5 where 0 indicates an un-rated movie. There are 100,000 non-zero ratings. The average number of ratings received by an item is 59 (standard deviation 80).

A proportion of the data set is removed for testing and the metric of precision is used to compare the performance of different approaches. Two sets of experiments are performed:

1. Comparing a memory-based collaborative filtering approach (using Pearson Correlation) with a spreading activation approach (using 3, 6 and 9 hops).
2. For the spreading activation approach, comparing the effect of increasing the number of hops from 3 to 6 hops and to 9 hops.

3.1 Parameters

The experiments use the following parameters involving the initialisation of edge weights based on user ratings for items; adjusting the Pearson Correlation value if users have not rated many items in common; and setting the threshold value for neighbourhood selection in the traditional collaborative filtering approach.

Edge Weights in Network Representation

The Movie Lens rating values are originally in the set {0, 1, 2, 3, 4, 5} where 0 indicates no rating has been given; 1 and 2 indicate negative ratings; and 4 and 5 represent positive ratings. A rating of 3 is considered an “I don’t mind” rating, showing neither a strong like or dislike for the item. The weights on the network edges indicate the strength of like/dislike for an item. Thus ‘dislike’ can be viewed as an inhibitory or negative rating and ‘like’ can be viewed as an excitory or positive rating. One approach to modelling liked and disliked items in the network representation is to use negative weight values for ratings which indicate ‘dislike’ and positive weight values for ratings which indicate ‘like’. The calculation of a node’s activity is then based on a sum of both positive and negative values.

Given that the original rating values 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, giving:

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

Therefore the original ratings of value 1 and 2 will now be negative values (-1.5 and -0.5) and the original ratings of 4 and 5 will be positive values (1.5 and 2.5). As 0 already has a special meaning it should not be mapped to any other value. It was considered that a slightly positive value (0.5) is a good mapping for the original rating of 3, indicating that the item is not disliked (not a negative value) but that it does not have a high positive value.

Adjustment to Correlation Values in Pearson Correlation Calculation

An adjustment is used in the Pearson Correlation calculation based on the number of items that users have rated in common (co-rated items). The motivation (as discussed in [7]) 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.

Given a correlation value calculated by the Pearson Correlation formula, this value is adjusted if the number of co-rated items between two users, *a* and *b*, is less than or equal to twice the average number of co-rated items between all users in the dataset. The adjustment, if used, involves multiplying the original correlation value between users *a* and *b* by:

$$\frac{num\_co\text{-}rated\_items_{a,b}}{avg\_num\_co\text{-}rated\_items \times 2}$$

where $num\_co\text{-}rated\_items_{a,b}$ is the number of items co-rated by both users *a* and *b*, and $avg\_num\_co\text{-}rated\_items$ is the average number of co-rated items across all users. For example, for one test the average number of co-rated items was 19. Therefore the correlation value of any two users who have co-rated 38 items or less is adjusted.

Neighbour Selection Threshold in Pearson Correlation

A low neighbour selection threshold is used in these experiments, with any correlation value greater than 0.01 being considered. Although Breese [2] found that users with high correlation values (> 0.5) were more valuable in providing recommendations work by Herlocker [3], using the Movie Lens data set, found that for this data set 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 [3].

3.2 Experimental Methodology

The data set is split into a training and test set as follows:

- 90% of users are used as a training set, and
- 10% of users are used as a test set.

Users are selected randomly but with the constraint that they must have rated over 30 items. For each user in the test set, 10% of the number of items they have rated is withheld[2]. The items removed for each user are chosen randomly but items are only chosen for the test set (to be withheld) if the user has actually rated the item (an approach also used in [2]). No distinction is made between choosing a "liked" or "disliked" rating. The data set is randomly split 100 times for 100 different runs. For each run, each user from the test set is in turn chosen to be the *active* user and each of the approaches (spreading activation or collaborative filtering using Pearson Correlation) predicts ratings for the withheld items.

3.3 Metrics

The metric of precision is used which calculates the percentage of items recommended to a user that the user actually likes, thus ignoring items that the user does not like. Precision is used because the spreading activation approach returns a ranking of recommended items, not prediction values that can be compared with actual values. In addition, work by Herlocker et al. [4] suggests that the level of accuracy provided by a metric such as Mean Absolute Error (MAE), which is often used in collaborative filtering evaluation, is unnecessary and that distinguishing between exact actual and predicted rating scores is not as important as measuring if the system correctly or incorrectly identifies "good" or "liked" items.

The steps involved in calculating precision are:

- Calculate precision over all items returned for each user (there is a varying number of items withheld for each user as 10% of the items rated by that user is withheld).
- To calculate precision, only the items that were liked by a user are considered. An item is defined to be *liked* if it received a rating of 3, 4 or 5 from a user. This agrees with the precious modelling of *liked* in the spreading activation approach where a value of 3 is given a positive value. Note that McLaughlin and Herlocker (2004) only use ratings of 4 and 5 to indicate *liked* (also using the Movie Lens data set). In initial experiments comparing the effect of using ratings for *liked* of 4 and 5 or ratings of 3, 4 and 5, better precision was achieved when the rating of 3 is included in the definition of *liked*.
- Average over all users (94 users).

---

[2] An alternative employed in [10] is to withhold the same number of items for all users in the test set, e.g., 30. This would mean that for some users there might be more data in the training set than there is for other users.

4. Results

Table 1 presents the final average precision values after 100 runs (each run involving a new training and test set) for each of the four approaches considered: Pearson Correlation (PC) and the three spreading activation approaches (SA) using 3, 6 and 9 hops. The standard deviation is also included.

	PC	SA: hops 3	SA: hops 6	SA: hops 9
average	0.871214	0.844506	0.843085	0.842897
std. dev.	0.01900	0.01801	0.01816	0.01820

Table 1. Average precision on test set for each of the 4 approaches

As can be seen from Table 1 the average precision for the Pearson Correlation approach is 0.871214 and for the spreading activation (3 hops) is 0.844506 and for the spreading activation (6 hops) is 0.843085. The difference between the Pearson Correlation approach and the spreading activation approaches is statistically significant (using 2-tail dependent T-Test with p value < 0.05) therefore the Pearson Correlation approach has better performance than the spreading activation approach.

Comparing the effect of increasing the number of hops in the spreading activation approach, it was found that the difference between 3 hops and 6 hops using a 2-tail dependent T-Test with p value <0.05 was statistically significant and that 3 hops had better performance than 6 hops.

It was found that the difference between 6 hops and 9 hops, with average precision of 0.843085 and 0.842897 respectively, was not statistically significant (again using a 2-tail dependent T-Test with p value < 0.05) and therefore performance has not increased or decreased by the inclusion of additional hops.

5. Summary

This paper presents an initial implementation of a spreading activation approach to collaborative filtering where the collaborative filtering space is viewed as a network consisting of two sets of nodes: user nodes and item nodes. The initial experiments have shown that a traditional collaborative filtering approach, using Pearson Correlation, performs better than the spreading activation approach and that the accuracy of the spreading activation approach does not increase as the number of hops are increased. These results could be due to the simple spreading activation approach used which does not employ thresholds when calculating node outputs and does not incorporate a decay factor as the number of hops increase (which would lessen the effect of noisy data). Ongoing work involves investigating the effect of incorporating a decay factor and investigating the effect of thresholds on outputs. In addition, further testing using other testing methodologies (using a different approach to select the testing and training sets) and calculating coverage will also be performed.

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, Proceedings of the Fifth ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'99), pp. 201-212, 1999.
2. J. Breese, D. Heckerman and C. Kadie, Empirical Analysis of Predictive Algorithms for Collaborative Filtering, Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, 1998.
3. J.L. Herlocker, Understanding and Improving Automated Collaborative Filtering Systems, PhD Thesis, University of Minnesota, 2000.
4. J.L. Herlocker, J.A. Konstan, L.G. Terveen and J.T. Riedl, Evaluating Collaborative Filtering Recommender Systems, ACM Transactions on Information Systems (TOIS), 22,(1), pp. 5-53, 2004.
5. 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), pp. 259-274, 2004.

6. T.-H. Kim, Y.-S. Ryu, S.-I. Park and S.-B. Yang, An Improved Recommendation Algorithm in Collaborative Filtering, Proceedings of the Third International Conference on E-Commerce and Web Technologies, pp. 254–261, 2002.
7. M.R. McLaughlin and J.L. Herlocker, A Collaborative Filtering Algorithm and Evaluation Metric that Accurately Model the User Experience, Proceedings of the 27th International *ACM SIGIR* Conference on Research and Development in Information Retrieval, pp. 329-336, 2004.
8. B. Mirza, B. Keller and N. Ramakrishnan, Studying Recommendation Algorithms by Graph Analysis, Journal of Intelligent Information Systems, 20(2), pp. 131-160, 2003.
9. M.F. Schwartz and C. M. Wood, Discovering Shared Interests using Graph Analysis, Communications of the ACM, 36(8), pp. 78-89, August 1993.
10. K. Yu, A. Schwaighofer, V. Tresp, X. Xu and H-P Kriegel, Probabilistic Memory-based Collaborative Filtering, IEEE Transactions on Knowledge and Data Engineering, 16(1), 2004.