Collaborative Movie Recommender Considering User Profiles Explicitly

Qing Li°, Byeong Man Kim, Yoon Sik Shin, En Ki Lim
Dept. of Computer Science, Kumoh National Institute of Technology
(liqing°, bmkim,ysshin,eklim) @se.kumoh.ac.kr

Abstract

We are developing a web-based movie recommender system that catches and reasons with user profiles and ratings to recommend movies. In the paper, we outline the current status of our implementation with particular emphasis on the mechanisms used to provide effective recommendations. Social recommender systems collect ratings of items from many individuals and use nearest-neighbor techniques to make recommendations to a user. However, these methods only depend on the ratings and ignore other useful information. Our primary concern is to provide an approach that can recommend the movies based on not only the user ratings but also the significant amount of other information that is available about the nature of each items - such as cast list or movie genre. We experimentally evaluate our approach and compare them to conventional social filtering, which suggests merits to our approach.

1. Introduction

In this paper we describe a web-based movie recommending agent system, which can recommend the movies based on not only the user ratings but also the significant amount of other information.

Recent years, several movie recommendation systems have been developed. However most of the movie recommender systems are based on the principal of collaborative filtering technology, which denies any useful information from contents -such as cast list, movie genre and synopsis of movie etc.

For this reason, hybrid recommender systems have been provided, which can exploit both user preferences and content. There are several hybrid-filtering approaches. One is a simple combination of the results of collaborative and content-based filtering, such as system that is described by Claypool [1]. The other is a sequential combination of the content-based filtering and collaborative filtering. In such system, firstly, content-based filtering algorithm is applied to find the similar interesting clique of users. Secondly, collaborative algorithm is applied to make predictions, such as RAAP and Fab filtering systems.

However, as for the linear combination model, it just recommends based on the two different recommender - content-based recommender and collaborative recommender. As for the sequential combination model, although it considers the useful information from user profiles, it denies the important information from ratings. We have developed a online movie recommender and applied our novel seamless approach to combine them, which takes the advantages of content-based filtering to solve the new user problem, and combines, at the same time, the content information of items and user ratings to make predictions.

2. Our approach

The architecture of our movie recommender system can be viewed as like Fig1, which consists of three layers.

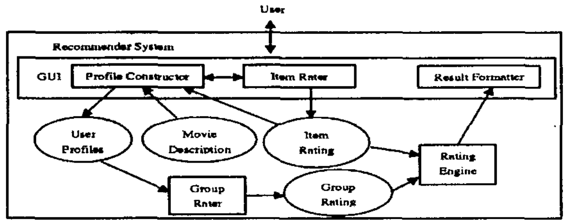

Figure 1. System Architecture

The first layer: GUI (Graphic User Interface) is used for user login, registration and obtaining preference information from users. Two types of user preference are obtained: one is content-based preference through the profile constructor and the other is the rating information on some preferred movies through the item rater. It would be difficult for users to exactly or correctly specify their information needs and so an alternative way is provided, where they only indicate movies that they are interested in. The actual user preferences are generated from that information together with movie description data. The details will be described in following part.

The second layer: Main application layer implements recommendation strategy. It consists of two function modules: group rater and rating engine. Group rater groups the user profiles to create a group-rating matrix for recommending. Rating engine is the function module for calculating the similarities between users and making predictions for users based on the item rating and group-rating matrix.

The third layer: database layer. It is a storage place for system information, such as user profiles, movie description, item ratings and group ratings

In our approach, we integrate the user profile information and user ratings to calculate the user-user similarity. The detail procedure of our approach is described as follows:

1. Applying grouping algorithm to group the user profiles, then using the result to create a group-rating matrix.
2. Compute the user-user similarity: firstly, calculate the user-user similarity of group-rating matrix using adjusted-cosine algorithm, then calculate the user-user similarity of item-rating matrix using Pearson correlation-based algorithm. At last, the total user-user similarity is the linear combination of the above two.
3. Make a prediction for an item by performing a weighted average of deviations from the neighbor's mean.

This work was supported by Korea Research Foundation Grant (KRF-2002-041-D00459).

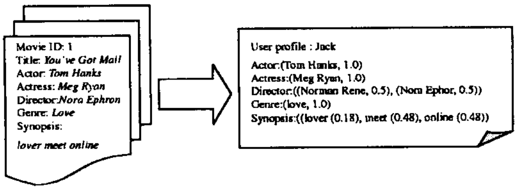

Figure 2. User profiles

2.1 User profile creation

User profiles indicate the information needs or preferences on movies that users are interested in. A user profile consists of five profile vectors and each profile vector represents an aspect (or dimension of his preferences). In our current system, five aspects - actor, actress, director, genre and synopsis - have been considered. A profile vector consists of several attribute-value pairs as Fig 2 shows.

To reduce the burden of users, the user only need indicate interesting movies instead of specifying his preference explicitly; the profile vectors are automatically constructed from movie description data by the following simple equation.

$$A = m/n \qquad (1)$$

where, n is the number of movies, in which ranking value is lager than the threshold, m is the number of movies containing attribute A among n movies and its ranking is larger than threshold. In our present experiment, we set the value of the threshold as 3.

The characteristics of the aspect synopsis are a little different from other aspects. So, we take a different approach, i.e., a keyword vector is calculated for each synopsis of movies. To calculate it, keywords are extracted from synopsis field in movie description record and weighted by the $tf \times idf$ formula, which is popularly used in information retrieval literature.

2.2 Group rating

The goal of group ratings is to group the items into several cliques and provides content-based information for collaborative similarity calculation. Each item has it's own attribute features, such as movie items, which may have actor, actress, director, genre, and synopsis as its attribute features. Thus, we can group the items based on those attributes.

Our group-rating algorithm is derived from K-means Clustering Algorithm. The difference is that we apply the fuzzy set theory to represent the affiliation between object and cluster. The possibility of one object j (here one object means one user) belonging to a certain cluster is calculated as follows.

$$\text{Pro(j,k)} = 1 - \frac{\text{S(j,k)}}{\text{MaxS(i,k)}} \qquad (2)$$

where Pro(j,k) means the possibility of object j belonging to cluster k ; The S(j,k) means the similarity between object j and cluster k ; Max S(i,k) means the maximum similarity over a certain cluster k . The similarity S(j,k) can be calculated by Euclidean distance or Cosine method. In our experiment, the later show little better performance.

After applying group-rating algorithm the group-rating matrix is combined with the item-rating matrix to form a new rating matrix for later similarity computation.

2.3 Item rating

When a new user enters into the system, he or she is required to rate predefined movies. This rating information is used to construct the user profile and also used as item rating information. User can indicate a rating value from 1 to 5 according to user's preferences. In our system, 1 means very bad movie, 2 means bad, 3 means ordinary, 4 means good, 5 means very good. Rating engine uses group rating and item rating to make the recommendation.

2.4 Similarity computation

Due to difference in value range between item-rating matrix and group-rating matrix, we use different methods to calculate the similarity. As for item-ratings matrix, the rating value is integer; As for group-rating matrix, it is the fuzzy set value ranging from 0 to 1. The natural way is to enlarge the continuous data range from [0 1] to [1 5] or reduce the discrete data range from [1 5] to [0 1] and then apply Pearson correlation-based algorithm [2] or adjusted cosine algorithm [3] to calculate similarity. We call this *EUCHM (enlarged user-based clustering hybrid method)*. We also propose another method: firstly, use Pearson correlation-based algorithm to calculate the similarity from item-rating matrix, and then calculate the similarity from group-rating matrix by adjusted cosine algorithm, at last, the total user similarity is linear combination of the above two, we call this *CUCHM (combination user-based clustering hybrid method)*.

2.5 Collaborative prediction [2]

Prediction for an item is then computed by performing a weighted average of deviations from the neighbor's mean. Here we use top *N* rule to select the nearest *N* neighbors based on the similarities of users. The general formula for a prediction on item i of user k is:

$$P_{k,i} = \bar{R}_k + \frac{\sum_{u=1}^{n}(R_{u,i} - \bar{R}_u) \times sim(k,u)}{\sum_{u=1}^{n} | sim(k,u) |} \qquad (3)$$

where $P_{k,i}$ represents the predication for the user k on item i ; n means the total neighbors of user k ; $R_{u,i}$ means the user u rating on item i ; $\bar{R}_k$ is the average ratings of user k on items; $sim(k,u)$ means the similarity between user k and neighbor u ; $\bar{R}_u$ means the average ratings of user u on items.

3. Experimental evaluation

3.1 Data set

Currently, we perform experiment on a subset of real movie rating data collected from the MovieLens web site. The data subset contained 100,000 ratings from 943 users and 1,682 movies, with each user rating at least 20 items. The ratings in the MovieLens data are explicitly entered by users, and are integers ranging from 1 to 5. We divide data set into a training set and a test data set. 20 percent of MovieLens data are used as a training data set, the other 80 percent are used as a test data set. We only use the genre information of movie to create the user profiles, because the MovieLens data set do not contain any other information of movies except the genre information.

3.2 Evaluation metrics [3]

MAE (Mean Absolute Error) has widely been used in evaluating the accuracy of a recommender system by comparing the numerical recommendation scores against the actual user ratings in the test data.

The *MAE* is calculated by summing these absolute errors of the corresponding rating-prediction pairs and then computing the average. The lower the *MAE*, the more accurate.

3.3 Our method performance

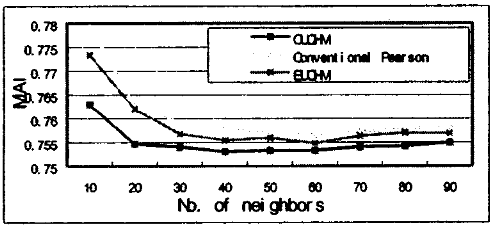

Figure 3. Comparison

In order to find the optimal combination coefficient in the *CUCHM*, we implement a serial of tests changing the value of combination coefficient from 0 to 1 with a constant step 0.1. When the coefficient arrives at 0.4, an optimal recommendation performance is achieved.

The size of the neighborhood has significant effect on the prediction quality [4]. In our experiments, we vary the number of neighbors and compute *MAE*. It can be observed from Fig.3 that the size of neighborhood does affect the quality of prediction. When the number of neighbors changes from 30 to 50 in our approach, it arrives at the optimal *MAE* value.

Nowadays many commercial collaborative systems are based on the classic Pearson method, which means that the group-rating matrix is not added into item-rating matrix, and only Pearson correlation-based algorithm is applied to calculate the similarity based on item-rating matrix. Comparing with the classic Pearson method, our approach shows a better performance, which can be observed in Fig.3.

3.4 New User Problem

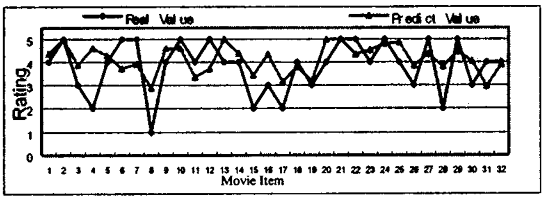

Figure 4. New user problem

In traditional collaborative filtering approach, it is hard for pure collaborative filtering to recommend any items to new user since the new user does not make any ratings on items. However, in our approach, based on the item group information, we can make predictions for new users. In our experiment, it shows a good recommendation performance. In Equation 3, $\bar{R}_k$ is the average rating of user k on items. As for the new user, who does not make any ratings on items, $\bar{R}_k$ should be the zero. Since $\bar{R}_k$ is the standard baseline of user ratings and it is zero, it is unreasonable for us to apply Equation 3 to new users. Therefore, for new user, we use the $\bar{R}_{neighbors}$, the average rating of all ratings on the new users' nearest neighbour instead of $\bar{R}_k$, which is inferred by the group-rating matrix.

In our experiment, we randomly select users, and delete all of their ratings, thus we can treat them as new users. First, we randomly selected user No.73. In training data, user No.73 makes ratings for 32 items, which is described by line *real value* in Fig. 4. We can observe that the prediction for new user can partially reflect the user preference. To generalize the observation, we randomly selected the number of users from 10 to 50 with the step of 10 and 100 from the test data, and delete all the ratings of those users and treat them as new users. Table 1 shows that our method can successfully solve the new user problem.

Table 1: MAE of new users

	10	20	30	40	50	100
MAE	0.819	0.677	0.756	0.786	0.745	0.695

4. Conclusions and future work

In this paper, we describe our movie recommender system, provide our new approach for movie recommendation and evaluate it via experiment on a large, realistic set of ratings. Since this mechanism is not limited to the movie domain, it can be extended to other domain, such as CD, music and books.

We look into applying clustering method to afford information from user profile contents for recommendation, which improves performance over the purely collaborative approach although only genre information is used. Further improvement may be archived if other information, especially synopsis is available. Since item-based collaborative filtering recommendation algorithms can further improve the performance of recommendation [3], we will apply clustering method to group item contents instead of user profiles and combine it with collaborative filtering to achieve a better performance.

REFERENCES

[1] Claypool, M.,Gokhale, A., Miranda, T., Murnikov, P., Netes, D. and Sartin, M.:Combining content-based and collaborative filters in an online newspaper. In ACM-SIGIR Workshop on Recommender Systems:Algorithms and Evaluation. University of California, 1999

[2] Resnick, P., Iacovou, N., Suchak, M. GroupLens: An open architecture for collaborative filtering of Netnews. *In Proc. ACM Conf. on Computer-Supported Cooperative Work.* pp.175-186,1994

[3] Sarwar, B. M., Karypis, G., Konstan, J. A. Item-based Collaborative Filtering Recommendation Algorithms. *In Proc. Tenth Int. WWW Conf.* pp. 285-295,2001.

[4] Herlocker, J., Konstan, J., Borchers A., and Riedl,J.: An algorithmic framework for performing collaborativefiltering. In Proc. ACM-SIGIR Conf. Berkeley, CA:ACM Press pp.230-237,1999