Extraction of User Preference for Hybrid Collaborative Filtering

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

With the development of e-commerce and information access, recommender systems have become a popular technique to prune large information spaces so that users are directed toward those items that best meet their needs and preferences. In this paper, clustering technique is applied in the collaborative recommender framework to consider semantic contents available from the user profiles. We also suggest methods to construct user profiles from rating information and attributes of items to accommodate user preferences. Further, we show that the correct application of the semantic content information obtained from user profiles does enhance the effectiveness of collaborative recommendation.

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 [2] and Fab filtering systems [3].

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

In our approach, we integrate the information of user profiles and user ratings to calculate the user-user similarity. From the user profiles, we get the feature-based information of users (Tom likes movies with Meg Ryan with average score 2.45), while, from the user ratings, we get the atomic-item-based rating information (Tom likes a certain movie with score 3). In Fab system, feature-based user profiles replace the atomic-item-based rating information. It helps alleviate sparsity, but the abstraction does lose information. However, in our approach we use both kinds of information - those based on ratings of individuals and those based on features. Figure1 shows the overview of our approach.

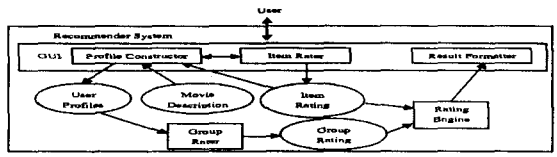

Figure 1. System Architecture

2.1 User profile creation

User profiles indicate the information needs or preferences on movies that users are interested in. In our current system, a user profile consists of five profile vectors and each profile vector represents an aspect (or dimension of his preferences). 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.

$$w_{n,m} = \frac{Num_{item \subseteq attribute_m \ of \ aspect_n | item > threshold}}{Num_{item > threshold}} \qquad (1)$$

where $Num_{item>threshold}$ is the number of items, in which ranking value is lager than the threshold ; $Num_{item \subseteq attribute_m \ of \ aspect_n}$ is the number of items containing the attribute m in the aspect n of user profile and its rating is larger than the threshold. In our present experiment, we set the value of the threshold as 3.

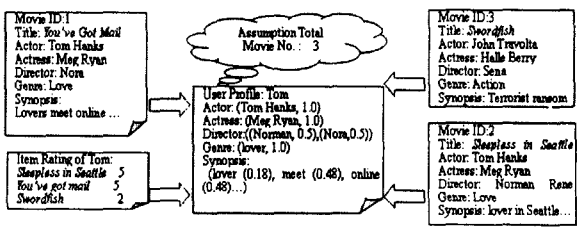

Figure 2. User Profiles

2.2 Construction of user profiles through Fuzzy Inference

The characteristics of the synopsis aspect are a little different from other aspects. So, we take a different approach. A weighted keyword vector is constructed for each user. Firstly, keywords are extracted from synopsis field of movie description record, which user shows interests on. Secondly, calculate the weight of each keyword. Although tf ×idf [4] formula is popularly used in information retrieval literature for calculating the keyword weight, we adopt the method of calculating weight through fuzzy inference from our previous work [5], which shows more effectiveness.

In order to construct the user profiles based on the synopsis, firstly, the synopses of user preferred movies are transformed into a set of candidate terms through eliminating stopwords and stemming by Porter's algorithm. The DTF, DF, and IDF of each term are calculated based on this set and used as inputs of fuzzy inference. The DTF (Distributed Term Frequency) reflects the frequency and distributed status of a term in a set of documents (user preferred movie synopses), which is calculated by dividing total occurrences of the term in a set of documents by the number of documents in the set containing the term. It needs to be normalized for being used in fuzzy inference. The following shows the normalized distributed term frequency (NDTF).

$$NTF_i = \frac{\frac{TF_i}{DF_i}}{\max_j \left[\frac{TF_j}{DF_j}\right]} \quad (2)$$

where, TF_i is the frequency of term t_i in the example documents (user preferred movie synopses), and DF_i is the number of documents having term t_i in the example document.

The DF (Document Frequency) represents the frequency of documents having a specific term within the example documents. The normalized document frequency, NDF, is defined in equation (4), where DF_i is the number of documents having term t_i in the example documents.

$$NDF_i = \frac{DF_i}{\max_j DF_j} \quad (3)$$

The IDF (Inverse Document Frequency) represents the inverse document frequency of a specific term over an entire document collection not example documents. The normalized inverse document frequency, NIDF, is defined as follows:

$$NIDF_i = \frac{IDF_i}{\max_j IDF_j}, \quad IDF_i = \log\frac{N}{n_i} \quad (4)$$

where, N is the total number of documents and n_i is the number of documents in which term t_i appears.

Figure 3 shows the membership functions of the input/output variables - 3 inputs (NDTF, NDF, NIDF) and 1 output (TW) - used in our method. As you can see in Figure 3(a), NDTF variable has { S(Small), L(Large) }, and NDF and NIDF variables have { S(Small), M(Middle), L(Large) } as linguistic labels (or terms). The fuzzy output variable, TW (Term Weight) which represents the importance of a term, has six linguistic labels as shown in Figure 3(b).

The 18 fuzzy rules are involved to infer the term weight (TW). The rules are constructed based on the intuition that the important or representative terms may occur across many positive example documents (user preferred movies) but not in general documents, i.e., their NDF and NIDF are very high. As shown in Table 1, the TW of a term is Z in most cases regardless of its NDF and NDTF if its NIDF is S, because such term may occur frequently in any document and thus its NDF and NDTF can be high. When NDF of a term is high and its NIDF is also high, the term is considered as a representative keyword and then the output value is between X and XX. The other rules were set similarly.

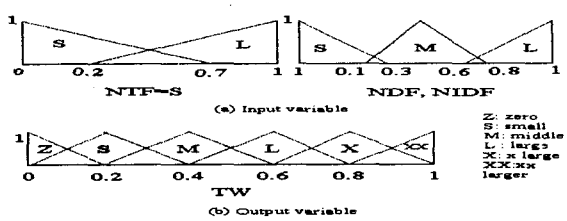

Figure 3. Fuzzy variables

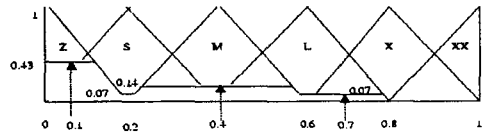

Figure 4. Defuzzify the output

Table 1. Fuzzy inference rules

NTF = S

NIDF / NDF	S	M	L
S	Z	Z	S
M	Z	M	L
L	S	L	X

NTF = L

NIDF / NDF	S	M	L
S	Z	S	M
M	Z	L	X
L	S	X	XX

We can get the term weight TW through the following procedure. But, the output is in the form of fuzzy set and thus has to be converted to the crisp value. In this paper, the centre of gravity method (COG) is used to defuzzify the output.

i. Apply the NTF, NDF, and NIDF fuzzy values to the antecedent portions of 18 fuzzy rules.
ii. Find the minimum value among the membership degrees of three input fuzzy values.
iii. Classify every 18 membership degree into 6 groups according to the fuzzy output variable TW.
iv. Calculate the maximum output value for each group and then generate 6 output values.

For instance, let us assume, there is one term, which NIDF is 0.35, NDF is 0.2 and NDTF is 0.3. The degree of membership is determined by plugging the selected input parameter (NIDF, NDF or NDTF) into the horizontal axis and projecting vertically to the upper boundary of the membership function(s) in Figure 4. So the result is as follows.

NIDF=0.35 : S=0.00, M=0.57;
NDF=0.20 : S=0.43, M=0.14;
NDTF=0.30 : S=0.53, L=0.07.

Now referring back to the rules, only 8 rules out of 18 rules need to be selected, which are marked in Figure 4. The effective rules are listed as follows.

If(NIDF=S,NDF=S,NDTF=S) then TW=Z min{S=0.00,S=0.43,S=0.53}=0.00
If(NIDF=S,NDF=M,NDTF=S) then TW=Z min{S=0.00,M=0.14,S=0.53}=0.00
If(NIDF=M,NDF=S,NDTF=S) then TW=Z min{M=0.57,S=0.43,S=0.53}=0.43

If(NIDF=M,NDF=M,NDTF=S) then TW=M min{M=0.57,M=0.14,S=0.53}=0.14
If(NIDF=S,NDF=S,NDTF=L) then TW=Z min{S=0.00,S=0.43,L=0.07}=0.00
If(NIDF=S,NDF=M,NDTF=L) then TW=Z min{S=0.00,M=0.14,L=0.07}=0.00
If(NIDF=M,NDF=S,NDTF=L) then TW=S min{M=0.57,S=0.43,L=0.07}=0.07
If(NIDF=M,NDF=M,NDTF=L) then TW=L min{M=0.57,M=0.14,L=0.07}=0.07

Then we calculate the maximum output value for each group and generate 6 output values as Figure 4 shows, which consists a fuzzy set of TW as follows.

TW={Z/0.43, S/0.07, M/0.14, L/0.07, X/0, XX/0}

At last, the COG is used to defuzzify the output into one value. Details can be referred to [6].

$$TW = \frac{0.43 \times 0.1 + 0.07 \times 0.2 + 0.14 \times 0.4 + 0.07 \times 0.7}{0.1 + 0.2 + 0.4 + 0.7} = 0.116$$

2.3 Grouping ratings

After representing the user profiles, the next task for us is to group the user profiles to form group-rating matrix. Since the result of clustering aims at applying to build the relationship among all users, we build cliques over all users instead of the nearest neighbor between all possible pairs. Detail can be referred to [7].

2.4 Similarity computation & prediction

As we can see, after grouping the user profiles, we get a new rating matrix. We can use the user-based collaborative algorithm to calculate the similarity and make predictions for users. In our approach, we use the Pearson correlation-based algorithm to calculate the user-user similarity from user-rating matrix, and then calculate the user-user similarity from group-rating matrix by the adjusted cosine algorithm. At last, the total user similarity is the linear combination of the above two as like in [7]. Prediction for an item is then computed by performing a weighted average of deviations from the neighbor's mean [8]. This suggested recommendation mechanism is called UCHM, and the detail of this approach can be referred to [7].

3. Experimental evaluation

Currently, we perform experiments on a real movie rating data collected from the MovieLens web-based recommendation system. The data set 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 were 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% of users are randomly selected to be the test users, and 80% of users are randomly selected to be the training set. MAE (Mean Absolute Error) is used for evaluating the accuracy of a recommender system [8].

In the MovieLens data, there are no user preferences and only genre information of movies. Therefore, we collect the movie semantic information, that is genre, actor, actress, director and Synopsis, from Internet Movie Database (http: //www.imdb.com) to construct the user profiles from the training data set.

Recall that, user profiles consist of terms with term weights from the semantic information of movies (genre, actor, actress, director, and synopsis). Since there are so many terms in the movie synopsis, the effectiveness of clustering method is sensitive to the number of terms selected for clustering. It can be observed from Figure 5 that when the number of selected terms is about 50, it shows a good performance.

As we discussed in Section 2.2, in order to enhance the performance,

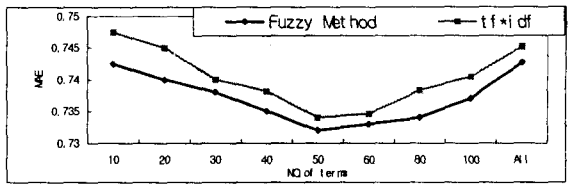

Figure 5. Terms of Synopsis

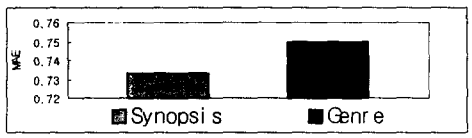

Figure 6. Comparison

we calculate the term weights through fuzzy inference rule, instead of the simple tf ×idf method. As Figure 5 shows, our suggest method does improve the performance.

In order to observe the contribution of each content information. We construct user profiles by movie synopsis and genre respectiveiy. In our experiment, we found that the semantic information extracted from synopsis contributed greatly to the recommendation performance, as Figure 6 shows.

4. Conclusion

In this paper, we describe our movie recommender system, provide our new approach for movie recommendation and evaluate it via experiment on a large and 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.

5. Reference

[1] Claypool, M., Gokhale, A., Miranda, and Sartin, M.(1999). Combining content-based and collaborative filters in an online newspaper. *ACM SIGIR '99 Workshop on Recommender Systems*

[2] Delgado, J., Ishii, N. and Ura, T. (1998). Content-based Collaborative Information Filtering: Actively Learning to Classify and Recommend Documents. *CIA' 98.*

[3] Balabanovic, M., Shoham, Y. (1997). Fab: Content-Based, Collaborative Recommendation. *Communications of the ACM.*

[4] Ricardo Baeza-Yates, Berthier Riberio-Neto. Modern Information Retrieval. New York:Addison-Wesley Publishers (1999).

[5] B.M. Kim, Qing Li (2003). "Extraction of User Preferences from a Few Positive Documents", IRAL2003, workshop of ACL2003, Sapporo, Japan.

[6] C.C. Lee (1990). Fuzzy logic in control systems: fuzzy logic controller-part I , IEEE Trans. On Systems, Man, and Cybernetics, 20 (2) , p408-418.

[7] Qing Li, Byeong Man Kim (2003). Clustering Approach for Hybrid Recommender System, In Proc. of the 2003 IEEE/WIC International Conference on Web Intelligence. Canada.

[8] J. S. Breese, D. Heckerman, and C. Kardie (1998). "Empirical Analysis of Predictive Algorithms for Collaborative Filtering". In Proc. Of the 14th Conference on Uncertainty in Artificial Intelligence, pp.43-52.