Metadata-Based Collaborative Filtering Using K-Partite Graph for Movie Recommendation

Eugene Seo, Dongyeop Kang and Ho-Jin Choi
Department of Computer Science, KAIST
291 Daehak-ro, Yuseong-gu, Daejeon 305-701, Korea
{eseo, dykang, hojinc}@kaist.ac.kr

Collaborative filtering recommends items to a user based on the interests of other users having similar preferences. However, high dimensional, sparse data result in poor performance in collaborative filtering. This paper introduces an approach called multiple metadata-based collaborative filtering (MMCF), which utilizes meta-level information to alleviate this problem, e.g., metadata such as genre, director, and actor in the case of movie recommendation. MMCF builds a k-partite graph of users, movies and multiple metadata, and extracts implicit relationships among the metadata and between users and the metadata. Then the implicit relationships are propagated further by applying random walk process in order to alleviate the problem of sparseness in the original data set. The experimental results show substantial improvement over previous approaches on the real Netflix movie dataset.

Key Words: recommendation system, collaborative filtering, k-partite graph, random walk

1. INTRODUCTION

A recommendation system is an application to provide personalized items or information to a user by analyzing the user's preference [1]. To recommend items to a user, a recommendation system builds the user's profile or preference information. Such information can be obtained usually from previously purchased or used items, or ratings for those items given by the user. Many examples of recommendation system exist such as book recommendations at Amazon (http://www.amazon.com/), movies at Netflix (http://www.netflix.com/), and music at iTunes Genius (http://www.apple.com/itunes/). Especially, Netflix opened a competition Netflix Prize [2] in 2006 for the advancement of required technology for improving its movie recommendation system, Cinematch. In this paper, we used the Netflix dataset for the purpose of our experiment.

In the Netflix Prize, many participants have tried to solve the movie recommendation problems mainly using a collaborative filtering approach [1,3] because of its efficient and effective performance. However, most works applying a collaborative filtering for recommendation suffer from data sparseness problems. In fact, the Netflix data only has less than 2% user-movie pair of ratings of whole user-movie pairs. It causes poor clustering and recommendation results. Likewise, data sparseness is a critical problem in collaborative filtering based recommendation systems. In addition to data sparseness problem, using only user-movie ratings information in collaborative filtering approaches lacks of analyzing a user's movie preference and find similar users. When a user select a movie, there are many reasons to choose the movie and the reasons are different with a user by a user. The previous collaborative filtering based movie recommendations cannot capture a user's hidden movie preference and actual similar users because they only rely on the previous movie selection or ratings information.
In this paper, we propose an approach called multiple metadata-based collaborative filtering (MMCF) for movie recommendation. Metadata is loosely defined data about data meaning other data describing original data in definition, structure, and administration aspects. Multiple metadata means more than one descriptive metadata about movie such as title, author, subject, keywords, publisher, and so on. In this thesis, genre, director, and actor are considered as multiple metadata of movies which influence users to select movies, and we believe those meta-level information improves movie recommendation performance. MMCF also considers implicit relationships of multiple metadata to capture hidden movies preference of a user and find similar users based on the hidden movie preference on multiple movie metadata. Therefore, we aim to improve collaborative filtering performance as well as movie recommendation performance. Our research mainly has two challenge issues. One is the representation of multiple metadata and their implicit relationships. Another is the way to deal with data sparseness as a common problem in collaborative filtering approaches with a huge data size. MMCF solves those two problems with proposed a movie k-partite graph representation and random walk propagation on the movie k-partite graph.

The rest of this paper is organized as follows. Section 2 summarizes briefly the existing work. Section 3 presents the overall and the detailed of MMCF approach we propose, and Section 4 shows the experimental results. Section 5 concludes.

2. RELATED WORK

During recent years, many researchers have tried to develop effective and accurate recommendation techniques to increase users’ satisfaction for the recommendation. The recommendation techniques are mainly divided into two methods [1]: content-based approach, and collaborative filtering. A content-based approach makes item recommendation by looking over contents of items a user have used or liked. In this approach, collecting and comparing all content information of items are critical issues in the sense of time complexity aspect. A collaborative filtering approach first looks for similar users or items, then uses group information to make item recommendation to each user. In this approach, user or item similarity measurement is one of key issue. Both approaches have been researched during the last decade, and the collaborative filtering approach is most popular in academic fields as well as applications due to its advantage of independence of content information, implicit association, and effective performance than a content-based approach. There are many variations of collaborative filtering to improve the performance of recommendation with proposed approaches [3,4,5,6,7,8]. In this paper, we will address the limitations of the collaborative filtering approaches and propose a novel approach based on utilizing meta-level information.

3. PROPOSED METHOD

3.1 Overall Approach
Figure 1 shows the overall framework in our movie recommendation. To make movie recommendation we obtain base information such as user given movie ratings as well as movie metadata which could be genre, director, or actor. The information is represented in matrix format. That is, we have user-movie rating matrix and three movie-metadata matrix which are movie-genre, movie-director, and movie-actor in this case. The proposed collaborative filtering approach shows how users are clustered in the aspects of each user's movie preference. Our movie recommendation system aims to figure out how a user has chosen movies as considering movies' features.

3.2 A Movie K-Partite Graph

A k-partite graph [9] is a graph whose vertices can be partitioned into k disjoint sets. A movie k-partite graph models the relationship between users, movies, and multiple metadata, in our case, represented by five sets of vertices: users, movies, movies' genres, directors, and actors (see Figure 2 for example). Unlike a general k-partite graph, however, we differentiate edges into two categories, explicit edges and implicit edges. The explicit edges mean the explicitly known relationships from given data such as a user's movie selection information between a user and movies and a movie's multiple metadata information between a movie and multiple metadata. The information is explicitly obtained from a user's movie selection behavior and a movie's meta-level information from Web site. However, we do not know about a user's movie preference, especially, in various movie metadata aspects.
Our movie recommendation system extracts two types of implicit relationships. First, it extracts user-metadata implicit relationship by assuming a user's movie selection behavior implies the user's preference for movie metadata. This relationship shows a user's different interest or preference for each movie metadata. The system also extracts metadata-metadata implicit relationship from the movie-metadata information. In this implicit relationship, we assume that metadata appeared in the same movie are correlate each other. For example, particular genres have similar characteristics such as mystery and thriller, or a particular director frequently work with a particular actor in his or her movies. Those interconnected relationships among multiple metadata influence to capture a user's hidden preference and identify similar groups. The information ultimately would help to improve collaborative filtering to find more similar users as well as to cause better movie recommendation performance.

The k-partite graph is transformed into an adjacency matrix for mathematical computation. The size of adjacency matrix for a movie k-partite graph is a sum of total number of users, movie, genres, director, and actors.

The explicit relationships between a user and movies are represented by a user's rating for movies and the explicit relationships between a movie and metadata are represented by binary value indicating whether they are relevant or not. For the implicit relationships, values of all edges mean the frequency of two elements related in movies. For example, the value of an edge between a genre and a director means that the director has made movies of that genre in the number of value times.

The user-metadata implicit relationship explains that which genre, director, or actor of movie a user is interested in. A user's interest in genre, director, actor of movies is obtained from the movies the user have selected and watched since the user would choose a movie based on his or her interest of movie metadata. Therefore a user's movie preference in movie metadata is generated by matrix multiplication of user-movie matrix with movie-metadata matrix.

The results of matrix multiplication indicate frequencies of a user's choice for the multiple metadata. The high value implies a strong interest for corresponding movie metadata and the smaller value means less interest.

Another implicit relationship is metadata-metadata implicit relationship which means the correlation of movie multiple metadata by the occurrence of metadata in a movie. That is, those elements are regarded to share similar characteristics if two
of elements are appeared in a movie. Therefore we also obtain the metadata-multiple metadata implicit relationship by matrix multiplication of each movie-metadata matrix through movies as medium.

3.3 Random Walk Propagation

A user-metadata information is still sparse because the number of movies a user has seen is too little so that the initially inferred a user's interest for movie multiple metadata is not sufficient to identify user similarity among users. Thus, we find the hidden semantic interests of each user from the extracted two implicit relationships. This is called propagation process on a user-metadata implicit relationship from metadata-metadata implicit relationship information. We use a random walk technique to propagate the metadata-metadata implicit relationship into the user-metadata implicit relationship.

Random walk [9,10,11] is a mathematical formalization of consisting of successive random steps. Random walk on graph is a kind of probabilistic process to move from one node to another. We define the random steps in a movie k-partite graph as movie selection steps or movie browsing steps of a user. The random walk on graph formulates the way of movie selection process based on a user's interests in movie multiple metadata. From the initial state which is a particular user, we model the stochastic process to choose other attributes of one of multiple movie metadata and find the probability of having interest in other attribute of movie multiple metadata through the steps. In other words, we formulate the stochastic process of a user's choice for different genre, director, and actor based on the initial his or her movie interest in multiple metadata and relationship between the metadata a user have shown interest in and others from metadata-metadata implicit information.

3.4 User Clustering and Movie Recommendation

A k-nearest neighbor (k-NN) algorithm is used to find similar user groups in movie multiple metadata spaces. Closely located k nearest users of a particular user implies that they have a similar pattern of interest in movie multiple metadata so that they are considered as similar users. The k-NN algorithm aims to classify k users for n target users based on closeness among users. After classifying k similar users in three different groups of genre, director, and actor, we finally determine similar user groups to be used for collaborative filtering. Basically, there are two ways to collect similar users: combining all similar users as a final group or selecting only the users belonging to three groups.

From all collected similar users from three different movie metadata, our movie recommendation system predicts ratings of unseen movies for each user by calculating a average value of ratings of neighbor who already watched and gave a rating for the particular movies. Finally, our movie recommendation system recommends top predicted movies to users assuming they will like the movies.

4. EXPERIMENT
4.1 Dataset and Environment

MMCF is evaluated on movie data from Netflix Prize Dataset. The dataset consists of 17 thousand movies, 480 thousand customers, and 100 million ratings. We also obtain movie metadata which is genre, director, actor information from IMDB website (http://www.imdb.com/). Among the 17 thousand movies, meta-level information for about 7 thousand movies is collected. We also filter out other types of media such as series or drama so that we only deal with pure movie type. Finally multiple metadata for 5 thousand movies are used. In total we have 21 genres, 2 thousand directors, and 3 thousand actors appeared at least once in 5 thousand movies. In our filtered data sample, the number of users who have rated movies at least once is 46 thousand. In summary, experiment dataset consists of 46 thousand users' movie rating information for 5 thousand movies and the movie's metadata from 21 genres, 2 thousand directors, and 5 thousand actors. Due to the memory problem for collaborative filtering computation, we divided whole user-movie rating dataset into 20 sub-datasets to experiment, then calculate average value of them for our evaluation. The experiment is mainly working on single PC which supports 64 bit-Window 7 Enterprise K OS, Intel(R) Core(TM)2 Duo 3.00 GHz CPU, and 8G main memory. The development environment is MATLAB version 7.6.0.324 (R2008a).

As for the evaluation criteria, Root Mean Squared Error (RMSE) will be used in our experiments as in the case of Netflix Prize [2]. RMSE measures the differences between actual movie ratings and predicted movie ratings.

4.2 Experimental Results

The proposed MMCF is evaluated with the traditional CF and genre-based CF. Traditional CF cluster users on user-movie rating matrix relying on the similar pattern of ratings over movies. The genre-based CF finds similar users in a genre space. Our proposed approach clusters users in movies multiple metadata aspects with extracted hidden preference in metadata. We also evaluate MMCF in two different ways: one is based on users' movie ratings (see Figure 3); another is based on users' movie selection (see Figure 4).

![Figure 3. RMSE comparison based on users' movie ratings.](image-url)
Mainly the proposed MMCF is evaluated in movie recommendation accuracy measured in RMSE values. The CF based movie recommendation is evaluated in different number of similar users such as 50, 100, 200, and 300 in a group. It is interpreted that movie recommendation in smaller number of similar users is more efficient than recommendation with larger number of similar users because of computation time as well as memory cost, in addition, the larger number of similar users ends up general opinions rather than personalized information. In this consideration, our proposed CF approach generates accurate movie recommendation performance within smaller number of similar users than other CF approaches. Moreover, our MMCF outperforms over other CFs every experiment. Those results mean that MMCF can generate reasonably accurate movie recommendation in smaller similar users, thus it saves time and memory computation as well as increase movie recommendation performance.

We also evaluate the data sparseness in user matrixes which are used for collaborative filtering (Table 1). Traditional CF uses user-movie matrix to cluster users, and genre-based CF clusters users on user-genre matrix while MMCF clusters users on dense form of user-metadata matrix. Therefore, we compare the degree of data sparseness among user-movie, user-genre, user-metadata matrixes. As we expected, MMCF alleviates data sparseness problem using random walk propagation than other CF approaches. Especially, our user matrix is denser than user-genre matrix used in genre-based CF even though our user vector size is 300 times bigger than a user-genre vector.

Table 1. Comparison of data sparseness.

<table>
<thead>
<tr>
<th></th>
<th>user-movie</th>
<th>user-genre</th>
<th>user-metadata</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparseness</td>
<td>98.32%</td>
<td>29.55%</td>
<td>13.50%</td>
</tr>
</tbody>
</table>
5. CONCLUSION
In this paper, we have proposed an advanced collaborative filtering method with multiple movie metadata, called MMCF for movie recommendation. Our proposed MMCF contributes on not only alleviating data sparseness but also capturing a user's hidden semantic preference of movie in multiple metadata aspects. MMCF analyzes the relationships between a user and movie metadata such as genre, director, and actor. The proposed MMCF represents a user's movie preference via a movie k-partite graph, and then discover the hidden distribution of a user's movie preference using random walk propagation on implicit relationships among multiple metadata. Finally, our algorithm generates a dense form of user-metadata matrix and obtained 10% RMSE improvement of movie recommendation over traditional collaborative filtering based movie recommendations on real Netflix data.

ACKNOWLEDGEMENTS
This work was partially supported by Defense Acquisition Program Administration and Agency for Defense Development under the contract (UD060048AD).

REFERENCES