QUALITATIVELY PREDICTING USER'S MUSIC PREFERENCES USING INTEGRATED COLLABORATIVE FILTERING

Prachi Patel[1], Minubhai Chaudhari[2]

[1] *Student, Computer Engineering Department, Government Engineering College, Gandhinagar, India*
[2] *Head of Department, Computer Engineering Department, Government Engineering College, Gandhinagar, India*

ABSTRACT

Collaborative Filtering based recommenders are very popular in electronic commerce. Main challenges are the sparsity of the user-item rating matrix, scalability of the system and cold-start. In this paper, a new hybrid method has been proposed where user-based collaborative filtering and item-based collaborative filtering are combined together, which reduces sparsity problem. Clustering is applied as a preprocessing step to make the system more scalable. Item features based clusters and profile based clusters are introduced to generate the quality predictions. Experimental results show that combined user-based and item-based filters produce quality predictions while using clustering as preprocessing.

Keyword: - *Collaborative filtering, Recommender Systems, Content-based filtering, Clustering*

1. INTRODUCTION

The intense development of online environment has raised an issue of Information Overloading. Users have many options to consider and they can't evaluate each option. There is a trend of online shopping and people also listen to music and watch movies online. As the huge data related to a product is available, it becomes tedious for a user to filter the useful information. To help the customers, many e-commerce companies use the Recommender Systems so that customers can acquire relevant information. This is a Win-Win strategy where customers get useful information regarding the product which they are likely to buy and companies get benefit with an increase of sale. Recommender Systems estimate a utility function which predicts a user's preference. i.e., how a user will like an item [1]. To predict the preference of a user, system considers past behavior of the user, relation of a user to other users of the system, similarity between items, contextual information, etc. depending on the problem domain. Preferences of users are taken either explicit or implicit. In explicit preferences, user's ratings for items are considered while in implicit method, user's click through data or his interaction history with system are taken into consideration. There are broadly three types of Recommendation systems: Collaborative filtering, Content-based filtering and Hybrid approaches [2].

Collaborative Filtering methods make use of the past ratings of the users and predict the missing rating [3]. There are two approaches to collaborative filtering: User-based collaborative filtering and item-based collaborative filtering [3]. User-based collaborative filtering finds the neighborhood containing most similar users for a given user and generates the predictions by considering the users in the neighborhood [3]. Item-based method finds the items

which are similar to the given item and form the neighborhood. Then predict the ratings using the neighborhood of the item [3]. Collaborative filtering techniques suffer from many potential challenges such as: Sparsity, Scalability and Cold-start [3].

Sparsity: Large commercial web sites contain thousands of products and users. In such scenario, even an active user has rated under 1% of products. Thus user-item matrix used in collaborative filtering is very sparse [1]. A Problem called Neighbor Transitivity is also present in sparse database where users having similar preferences cannot be identified because they didn't have rated the same item [1]. To overcome the data sparsity problems, dimensionality reduction techniques are used. For Example, Singular Value Decomposition (SVD) reduces the dimensionality of user-item matrix by removing insignificant items or users but it undergoes expensive matrix factorization steps [1]. Principal Component Analysis can also be used to reduce the dimensionality [1]. In these techniques where certain items are discarded useful information may get lost [1].

Scalability: As number of users and products increases, a recommender system experiences severe scalability problems. Techniques for dimensionality reduction like SVD (Singular Value Decomposition) are also used to solve the scalability problem. SVD involves matrix factorization [1]. Item based Pearson Correlated Collaborative filtering can be used to tackle scalability problem because it does not consider the similarity between each pairs of items but instead it considers the similarity between co-rated items by a user [1]. Model-based CF algorithms like K-means can also be helpful because they find the similarity for recommendations within smaller clusters rather than using the entire database [1].

Cold-start: To generate the predictions for a user, recommender system uses interaction history of a user with the system i.e., ratings given by a user. New users does not have past ratings. So, it is difficult to know the taste of the user without interaction history of the user [2]. Likewise, new items are not rated by enough number of users, system cannot recommend them to the users [2]. This is the problem of cold-start [2]. Content based approaches are useful in cold-start problem as they rely on attributes of the items instead of ratings of users [2].

2. TRADITIONAL COLLABORATIVE FILTERING METHODS

Collaborative filters consider similarity among users and items for recommendation [4]. In collaborative filtering, User-Item rating matrix is created having sparse rating values, as shown in Fig.1 and missing ratings are calculated [4].

	Item-1	Item-2	Item-3	Item-4
User-1	?	4	?	2
User-2	?	?	3	?
User-3	2	5	?	3
User-4	?	2	?	?
User-m5	3	?	4	1

Fig- 1: CF calculates the missing values given a User-Item rating matrix

There are two approaches to collaborative filtering based on k-Nearest algorithm: User-based collaborative filtering approach and Item-based collaborative filtering approach [2].

2.1 User-based Approach

For a target user, ratings of target user and ratings of users who are similar to the target user are considered to generate the predictions. Similarity between the target user and each other users are calculated. Based on the similarity, a neighborhood of similar users is formed. Predictions for target user is generated using the ratings of the users in the neighborhood [2]. An algorithm to generate such predictions is as follows [5]:

1. Calculate the similarity between each two users.
2. Generate the neighborhood by selecting k top most similar users with the target user.
3. Compute the prediction using the weighted combination of ratings of neighborhood users.

In step 1, similarity between the target user and other user is represented using a weight measure as shown in equation (1) [5]. Pearson correlation method is used to find the similarity $w_{u,v}$ between two users *u* and *v* [5]. Pearson correlation between two users *u* and *v* is defined as

$$w_{u,v} = \frac{\sum_{i \in I} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I} (r_{v,i} - \bar{r}_v)^2}} \quad (1)[5]$$

Where $i \in I$ represents that summations are over the items rated by the both users *u* and *v* [5]. $\bar{r}_u$ is the average rating of items rated by user *u* [5].

In step 2, k most similar user to the target user are chosen [5].

In step 3, Predictions are generated using weighted aggregate of the ratings of users chosen in step 2 [5]. To generate prediction for user *t*, on a certain item *i*, following formula is used [5]

$$p_{t,i} = \bar{r}_t + \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_u).w_{t,u}}{\sum_{u \in U} | w_{t,u} |} \quad (2)[5]$$

Where $\bar{r}_t$ and $\bar{r}_u$ represents the average ratings of user *t* and user *u* on other rated items [5]. Summations are carried out on all users $u \in U$ who have rated item *i*, $w_{t,u}$ represents weight between the user *t* and *u* [2].

2.2 Item-based Approach

Item-based collaborative filtering approach works like user-based approach but instead of finding similar users, it finds similar items rated by the user under consideration [2]. An algorithm to generate item-based predictions is as follows [5]:

1. Calculate the similarity between each item and the item that is rated by target user.
2. Select the *k* items having the highest similarity with the item rated by target user to represent the neighborhood.
3. Prediction for given item are calculated using the weighted combination of ratings of selected similar items.

In step 1, Similarity between two items *i* and *j* are computed using Pearson Correlation as follows:

$$w_{i,j} = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_i)^2} \sqrt{\sum_{i \in I} (r_{u,j} - \bar{r}_j)^2}} \quad (3)[5]$$

Where $r_{u,i}$ is the rating on item *i* by user *u*, $\bar{r}_i$ is the average ratings on item *i* [5].

Similarity between two items is more stable than similarity between two users. In step 2, k most similar items are selected [5].

In step 3, Ratings for item *i* for a target user *u* can be calculated using a simple weighted average as shown in Eq. (5)

$$p_{u,i} = \frac{\sum_{n \in N} r_{u,n} w_{i,n}}{\sum_{n \in N} | w_{i,n} |} \quad (4)[5]$$

Where $p_{u,i}$ is prediction of rating of item *i* by user *u* [4]. Summations are over all other rated items $n \in N$ for user u. $w_{i,n}$ is the weight between items *i* and *n*, $r_{u,n}$ is the rating of user *u* on item *n* [5].

3. PROPOSED APPROACH

Traditional user and item based collaborative filtering produces good results but these methods make use perform poor when the data are very sparse. So, a new hybrid collaborative filtering algorithm is presented here where user-based collaborative filtering and item-based collaborative filtering techniques are integrated into a single method. User-based collaborative filtering and item-based collaborative filtering methods make use of the entire database, which reduces the scalability of the system. Our approach apply clustering as a preprocessing step before calculating the predictions. Clustering is useful for dimensionality reduction. Also the user-based collaborative filtering and item-based collaborative filtering methods do not consider the item features which are important for personalized recommendations. In our clustering approach, we consider the item features also for more personalized and quality recommendations. We are making user profiles for more relevant matching of users.

Steps of proposed method:

1. Dataset contains users, items and the ratings given to different items by different users. Items have their own attributes, i.e., each song has genre, artist and album.
2. As a preprocessing process, feature-based clusters are formed. For each user in the dataset, all the songs rated by user are considered, and based on the feature of each song, clusters for genre, artist and album are created. For example, to form a genre based cluster for a user, genre of all the songs rated by the user are compared with genre of songs rated by other users of the dataset and if they match, those users are put in the genre cluster of the current user. Likewise, album based and artist based clusters are formed. According to the size of clusters, alpha, beta and gamma values are decided respectively for album, artist and genre based clusters. These alpha, beta and gamma are constant values.
3. All the three clusters generated for each user in the dataset are merged into a single cluster so that one single cluster for each user is generated.
4. User Profiles are created for each user by keeping score of genre, artist and album. The songs rated by all other users in the currents user's cluster are compared with given user's rated songs one by one. If any of the song's feature is matched with the currents user's rated song, rating is added to the score of that feature of the current user.
5. Three clusters of based on genre, artist and album score are generated. Based on alpha, beta and gamma values, top k users from each of the three clusters are selected and all selected users are merged into single cluster.
6. Item based co-occurrence is applied into the cluster obtained in the step3. For item co-occurrence, for each item we consider each other item in the cluster and find the frequency of their being rated together by a single user. Top k most similar items are selected.
7. User based collaborative filtering is applied into the cluster obtained in the step 5. Similarity among users is calculated using equation (1).
8. Item-based collaborative filtering is applied to clusters obtained in step 6. Similarity among users is calculated using equation (3).
9. Results of User-based collaborative filtering and Item-based collaborative filtering are merged in Hybrid collaborative filtering method according to their weight and final prediction is obtained. In this step, final prediction is generated using the weighted combination of equation (2) and equation (4).

$$p_{a,i}(hybrid) = \delta \times p_{a,i}(user) + (1-\delta) \times p_{a,i}(item) \quad (5)[5]$$

$$p_{a,i}(hybrid) = \delta \times \left(\bar{r}_a + \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_u).w_{a,u}}{\sum_{u \in U} | w_{a,u} |} \right) + (1-\delta) \times \left(\frac{\sum_{n \in N} r_{u,n} w_{i,n}}{\sum_{n \in N} | w_{i,n} |} \right) \quad (6)[5]$$

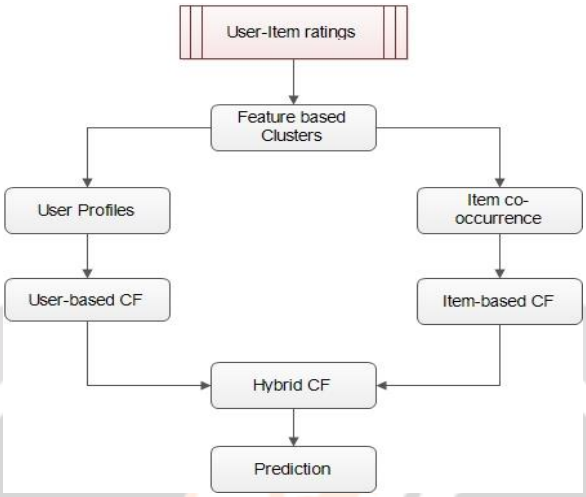

Fig- 2: System Diagram

4. EXPERIMENTAL RESULT

Dataset and experimental setup are described in this section. Result shows the transcendence of proposed method over the traditional methods.

4.1 Dataset Description

For this experiment, Yahoo! Music user ratings of songs with song attributes, version 1.0 dataset has been used. The dataset contains over 717 million ratings of 136 thousand songs given by 1.8 million users of Yahoo! Music services [6]. Each song in the dataset contains artist, album, and genre attributes. The mapping from geners id's to geners and a genere hierarchy, is given. This dataset has been partitioned into 10 training files and 10 testing files [6].

4.2 Evaluation Metric

Mean Absolute Error (MAE) is used to evaluate different methods used to generate the predictions. Mean Absolute Error is the average of absolute error between the predicted ratings and actual ratings [2].

$$MAE = \frac{\sum_{\{u,i\}} | p_{u,i} - r_{u,i} |}{n} \quad (7)[2]$$

Where n is total number of ratings, $p_{u,i}$ is the prediction of rating of user *u* on item i and $r_{u,i}$ is the actual rating of user *u* on item *i* [2].

4.3 Experimental Setup

System has been implemented using JAVA. Value of parameter alpha has been fixed to 0.5, beta is 1.5 and gamma is 8.0. We calculated Mean Absolute Errors in predictions for two users using all the methods, user-based collaborative filtering, item-based collaborative filtering, Hybrid collaborative filtering and Proposed method. Average of values of MAE for both users are considered to generate the graph as shown in Chart- 1. We have fixed

the value of parameter δ to 0.3 and calculated the MAE for different methods with different neighborhood parameter k. Proposed method gives the best result.

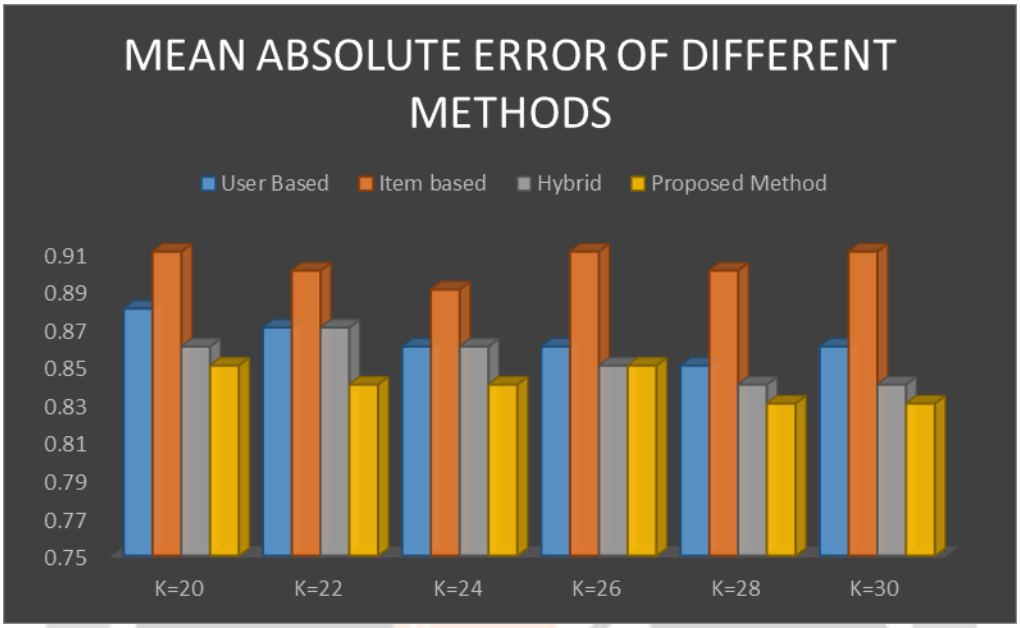

Chart -1: MAE of different methods

5. CONCLUSIONS

In this paper, we discussed user-based collaborative filtering and item-based collaborative filtering, which suffers from sparsity and scalability issues. We applied hybrid method by combining user-based collaborative filtering and item-based collaborative filtering to reduce sparsity problem. To make system more scalable, we applied clustering as a preprocessing step. User of personal user profiles gives more personalized recommendations. Experimental results shows robustness of proposed method.

6. REFERENCES

[1]. Su, Xiaoyuan, and Taghi M. Khoshgoftaar. "A survey of collaborative filtering techniques." *Advances in artificial intelligence* 2009 (2009): 4.

[2]. P. Melville and V. Sindhwani. Recommender systems. In Encyclopedia of Machine Learning, pages 829–838. 2010

[3]. Sunitha Reddy, M., and T. Adilakshmi. "Music recommendation system based on matrix factorization technique-SVD." Computer Communication and Informatics (ICCCI), 2014 International Conference on. IEEE, 2014.

[4].Liang, Zhang, Xiao Bo, and Guo Jun. "An Approach of Finding Localized Preferences Based-On Clustering for Collaborative Filtering." Web Information Systems and Mining, 2009. WISM 2009. International Conference on. IEEE, 2009.

[5]. Ji, Hao, et al. "Hybrid collaborative filtering model for improved recommendation." Service Operations and Logistics, and Informatics (SOLI), 2013 IEEE International Conference on. IEEE, 2013.

[6]. https://webscope.sandbox.yahoo.com/catalog.php?datatype=r