IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 2, Ver. I (Mar-Apr. 2016), PP 17-23
www.iosrjournals.org

Recommender Systems to Address New User Cold-Start Problem with User Side Information

M. Sunitha [1] , Dr. T. Adilakshmi [2]
[1,2]Department of Computer Science & Engineering, Vasavi College of Engineering Ibrahimbagh, Hyderabad-31, AP, India.

***Abstract**: Due to exponential growth of Internet, users are facing the problem of Information overloading. Recommender Systems (RS) serve as an indispensible tool to solve information overloading problem. Due to their great commercial value, recommender systems have also been successfully deployed in industry, such as product recommendation at Amazon, music recommendation at iTunes, movie recommendation at Netflix, etc. Collaborative Filtering (CF) is one of the popular approaches for Recommender systems based on user-item rating matrix. The main challenges faced by CF are Sparsity, Cold-Start and Scalability. In this paper we proposed a new approach to use user's side information in addition to user-item rating matrix to address new user cold-start problem. User's side information is obtained from Social Networks, which are the most active source of information now a day. First CF method is used to form user clusters based on the similarity of users. User's side information is obtained from Social networks and the same is used to form a Social matrix. Finally combine user history with social matrix to provide recommendations to new users. The experimental results proved that the performance of RS is improved over traditional CF systems.*

***Keywords**: Information Overloading, collaborative filtering, recommender system, Social Network, Social matrix*

I. Introduction

We are offered options among different things that we come across throughout a day. We hear song on a radio, see a movie, read about some books, or see different clothes/accessories. We form an opinion: we like them, don't like them or sometimes we don't even care. This happens unconsciously. Although these all seem random, we inherently follow a pattern and we call it personal taste. We tend to like similar things. For example, if someone likes bacon-lettuce-tomatoes and winches, then there are good chances that that person will also like a club sandwich because they are very similar only with turkey replacing the bacon. We follow these kinds of patterns inherently. In the crux, recommendations are all about pattern recognition and finding similarities.

Music is omnipresent. It is no surprise that there are millions of songs at everyone's fingertips. In fact, given the number of songs, bands, and artists coming up, music listeners are overwhelmed by choices. They are always looking for ways to discover new music so that it will match their taste. This has given birth to the field of music recommendations. In the past few years, there have been many services like Pandora, Spotify, and Last.fm that have come up in order to find a perfect solution, but haven't been completely successful. Choices are influenced by interests, trust, and liking towards any particular object and these emotions are very difficult to quantify especially for a machine or software. Hence, it has been a very difficult experience for these service providers to give a fulfilling experience. Every music recommendations system works on a given set of hypotheses, which they believe will result in the effective recommendations.There are two fundamental types of music recommendations: Collaborative filtering and Content-based Filtering. The major challenges with Collaborative filtering are Sparsity, Scalability and Cold-start [1][4][5].

Sparsity: The number of items each user will rate are much less compare to the total number of distinct items, i.e. a user will rate few items out of total number of available items. Because of this, the data structure User-Item matrix used in CF techniques will be sparse. Recommendations provided based on these sparse ratings will be less accurate i.e. user will be recommended many uninterested items.

Scalability: Scalability is another important problem faced by CF techniques. The time complexity of CF techniques increases exponentially with the increase in the number of users or items as they are basically dependent on similarity measures.

Cold-Start: Cold-start is the problem of not able to recommend items to new users or new items to existing users. This is because CF technique can not recommend items to new users until the new user rates sufficient number of items. Similarly CF technique will not be able to recommend new items to users until the items are being rated by sufficient number of users. In the past few years, the dramatic expansion of social networks poses new problems to the traditional Collaborative Recommender systems. Traditional CF based recommender systems do not consider social interactions of users into account. But in this paper we proposed an Algorithm to integrate the traditional Collaborative Filtering and social Information in the form of user's social interactions for recommender systems to address Cold-start problem.

The rest of the paper is organized as follows. Section 2 explains Related work. Section 3 discusses about the proposed algorithm. Section 4 describes conclusion and future directions of research.

II. Related Work

1.1 Collaborative based filtering

Collaborative Filtering (CF) is an approach in which information is gathered about the user's preferences for any particular item (books, videos, news articles, songs, etc.). The information captured is maintained in a data structure known as User-Item matrix as shown in Fig 2.1. This data structure will be used in predicting ratings for unknown items and users.

Item /User	**Item$_1$**	**Item$_2$**	**… …**	**Item$_n$**
User$_1$	S_{11}	S_{12}	… …	S_{1n}
User$_2$	S_{21}	S_{22}	… …	S_{2n}
… …	… …	… …	… …	… …
User$_m$	S_{m1}	S_{m2}	… …	S_{mn}

Fig.2.1.1 User-Item Matrix

The interaction between user and item in CF can be obtained by using two different methods.

a. Explicit Ratings

Here the user should be willing to express his/her preferences for an item. The preference can be a simple true (like), false (dislike) method or it can be a rating system (e.g., Rate a book on the scale of 1-5). This method comes with an assumption that a user will be actively participating to provide the feedback. The data they provide is to the best of their knowledge as an induction of false data/ noise can hamper the performance of the algorithms.

b. Implicit Ratings

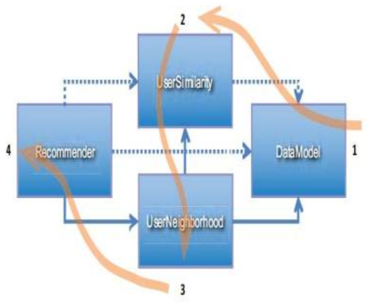
These are the inferred ratings which are interpreted as a result of user interactions with the items. These are the subtle algorithms which are used by many web-portals that are working behind the curtains. Data collected by this method needs to be pre-processed as there is a greater probability

Many variations of this algorithm have also been proposed. The basic idea behind CF is that users who agree in the past also tend to agree again in the future. Therefore, CF first finds users with similar taste to target user's. CF will then provide recommendations to the target user by predicting the target user's rating to the target item based on the ratings of his/her top-K similar users.

Collaborative filtering can be further be categorized into following two groups:

2.1.1 User Based Filtering

In the user based collaborative filtering approach, we use user ratings for each item in his profile to infer interests and make recommendations



.

Fig. 2.1.2 A simple user-based recommendation system

The bottom line of this approach is to find all the neighboring users for the current target user and try to fill in the missing pieces in order to guess the items that the target user would like. A simple user based collaborative filtering is shown in Fig.2.1.2

2.1.2 Item Based Filtering

In the item based collaborative approach, we construct item-profiles instead of user profiles and find

similarities between any two given items using various measures like Euclidean distance similarity, Tanimoto coefficient similarity, and Log likelihood similarity.
For any given item i, we compute its similarity with the one which is already present in the user profile to predict if the target item is worth recommending to the user or not. This type of approach is useful when new items are being added to the system too often.

2.2 Content Based Filtering

In Content-Based filtering, we analyze the attributes or the content of a song in order to make recommendations. In the case of a song, we analyze the kind of instruments used, tempo, pitch of the song, and store all those information in a structured format. Now, when a user listens to a particular song, the system analyses that song and finds the neighboring similar songs to make active recommendations. This approach is a content dependent approach because the methodology that is used to analyze or recommend songs would not work for analyzing books or videos since those items has different sets of attributes. Therefore, they should be approached differently. Pandora is one of the music services that uses content based filtering for their music recommendations.

2.3 Other Approaches

2.3.1 Automatical Playlist Generation

Automatically playlist generation focuses on recommending songs that are similar to chosen seeds to generate a new playlist. These approaches ignore user's feedback when the user listens to the songs in the playlist. They have an underlying problem that all seed-based approaches produce excessively uniform lists of songs if the dataset contains lot of music cliques. In iTunes, Genius employs similar methods to generate a playlist from a seed.

2.3.2 Dynamic Music Recommendation

Dynamic music recommendation improves automatic play list generation by considering the user's feedback. In the method, playlist generation starts with an arbitrary song and adjusts the recommendation result based on user feedback. This type of method is similar to Pandora.

2.3.3 Hybrid Approaches

Hybrid approaches, which combine music content and other information, are receiving more attention lately. Leverages both spectral graph properties of an item-base collaborative filtering a well as acoustic features of the music signal. Use both content features and user access pattern to recommend music.

2.4 Similarity Measures

Similarity measures are used to find the degree of similarity between two users. Each user is represented in the form of vectors. When the values of these vectors are associated with a user's model then the similarity is called user based similarity. When the values of these vectors are associated with the item's model then the similarity is called item based similarity. The similarity measure can be effectively used to find recommendations for target users to improve accuracy[1][2].

The following are the various types of similarity measures used in CF technique [1][3]. Pearson correlation, cosine vector similarity and adjusted cosine vector similarity etc[7][8][9]. Pearson's correlation, measures the linear correlation between two vectors of ratings.

$$Sim(i,j) = \frac{\sum_{c\in j}(S_{i,c} - A_i)\,(S_{i,c} - A_j)}{\sqrt{\sum_{c\in i}(S_{i,c} - A_i)^2 \sum_{c\in j}(S_{i,c} - A_i)^2}}$$

Where $S_{i,c}$ is the rating of the item c by $user_i$, A_i is the average rating of user i for all the co-rated items, and I_{ij} is the items set both rating by $user_i$ and $user_j$. The **cosine** is a measure of similarity between two vectors as the cosine angle between them.

Given two vectors of attributes, A and B, the cosine similarity, cos(θ), is represented using a dot product and magnitude as

$$Similarity = \cos(\theta) = \frac{A.B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i X B_i}{\sqrt{\sum_{i=1}^{n} A_i^{\,2}}\; X \sqrt{\sum_{i=1}^{n} B_i^{\,2}}}$$

The **adjusted cosine** similarity is used as a measure of similarity when each users scale of rating is different.

$$Sim(i,j) = \frac{\sum_{c\in j}(S_{i,c} - A_c)\,(S_{i,c} - A_c)}{\sqrt{\sum_{c\in i}(S_{i,c} - A_c)^2 \sum_{c\in j}(S_{i,c} - A_c)^2}}$$

Where $S_{i,c}$ is the rating of the item c by user $_i$, A_c is the average rating of user $_i$ for all the co- rated items, and $I_{i,j}$ is the items set both rating by user$_i$ and user$_j$.

III. Proposed Work

Social Recommender systems

We proposed an Algorithm in this section to combine user-based collaborative filtering with user's side information obtained from Social networks. Social influence on users not only comes from immediate friends but from distant friends also. The techniques for handling these types of influences are different. In this paper, we proposed an algorithm to combine the social influence of users as rich side information in addition to user's ratings.

Let us first revise the terminology used in recommender systems. In recommender systems we have a set of N users represented as U = {u1, ... uN } and a set of M items represented as I = {i1, ... iM}. The ratings expressed by users on items are given in a rating matrix R = [Ru,i]N×M. R is known as User-Item Matrix. In this matrix Ru,i denotes the rating of user u on item i. Ru,i can be any real number, but often ratings are integers in the range[6].

In Social network, each user u has a set of friends (immediate & distant). The social influence of each user's immediate friends is captured in the form of Social matrix. The accuracy of recommender system with social influence will improve compared to the traditional recommender systems [6]

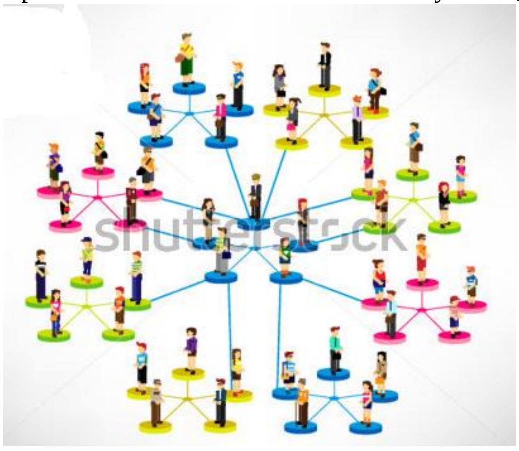


Fig.3.1.1 Social Network

Social Matrix

Social matrix is formed by extracting the information of each user from social Networks. For each user we get the list of immediate friends from user's social network profile. Distinct items in the item-set are considered as the columns in the matrix and target users friends are considered as the rows. If a user Ui rates an item j then the corresponding cell Uij is set as 1 otherwise 0.

Item	**Item$_1$**	**Item$_2$**	**Item$_3$**	**Item$_4$**
User$_1$	1	0	0	0
User$_2$	1	1	0	0
User$_3$	0	0	1	1
User$_4$	1	0	1	0

Fig.3. 2 .1Social Matrix

Combining U-I matrix with Social matrix

This Section discusses how the Collaborative filtering is combined with social networks. If the user is existing user then, we will have enough information to suggest recommendations. In case if the user is a new user we don't have sufficient/no information regarding him/her. In this scenario social contacts of the new user will give valid information about the user, which will be captured in the form of Social Matrix.

The following algorithm Collaborative_social recommender will take User-Item matrix and social matrix as input. For any target user, we will find the user's most similar cluster and get the social influence in the form of social matrix. User will be getting the recommendations from both.

```
Algorithm Collabarative_social Recommender()
Input: User-Item Matrix, Social Matrix
Output: Set of recommendations for target User
begin
1.  Consider the users as U1, U2,U3….Un
```

```
2.  Consider the items as I1, I2, I3,….In
    // Form user clusters based on Threshold based clustering algorithm
3.  Threshold_User Clusters()
4.  For a given target_user
5.  Find the user cluster to which the target_user is most similar
6.  Construct the social matrix by finding the immediate friends of the target_user
7.  Let the recommendations from the user clusters are Ru1,Ru2…. And from the social matrix are Rs1, Rs2,…
8.  Recommend the common items from both sets
End
```

Pseudocode for Collaborative_social Recommender

```
Algorithm Threshold_User Clusters()
Input: User-Item Matrix, Similarity_threshold
Output: User Clusters
begin
1.  For each user in $u_1$, $u_2$ ……………$u_n$
2.  Put $u_1$ into $C_1$ cluster and find the similarity with $u_2$
3.  Put $u_2$ into $C_1$ if the similarity is within the similarity _threshold
4.  Otherwise create a new cluster $C_2$
5.  Repeat this for all users and all Clusters
6.  Return the clusters $C_1$, $C_2$ ………… $C_k$
end
```

Evaluation Measures

Evaluating the data mining task is fundamental aspect of machine learning. Many methods have been proposed for assessing the accuracy of collaborative filtering methods. We have used Precision (P) and Recall (R). These measures are obtained from confusion matrix shown in Fig.

Confusion Matrix

A confusion matrix shows the number of correct and incorrect prediction made by the clustering model compared to the actual outcomes (target value) in the data.

	Actual – True	Actual- False
Predicted-True	True Positives (TP)	False Positives (FP)
Predicted-False	False Negatives (FN)	True Negatives (TN)

Fig.3.4.1 Confusion Matrix

Precision determines the fraction of relevant items retrieved out of all items retrieved. Recall determines the fraction of relevant items retrieved out of all relevant items. F-measure is the measure which stabilizes the changes in Precision and Recall.

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP + FP}}$$

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP + FN}}$$

$$F - measure = 2 * \frac{(\mathrm{Precision} * \mathrm{Recall})}{(\mathrm{Precision} + \mathrm{Recall})}$$

IV. Experimental Setup

This section describes about the Dataset used for experiment, experimental set up and results.

Data set

Million Song Dataset (MSD) a freely-available collection of audio files and their corresponding meta-data for a million contemporary popular music tracks. MSD contains meta-data, audio features, tags at artist-level and song-level, lyrics, cover songs, similar artists, and similar songs. It consists of four datasets namely Last.fm, Second hand data set, Musixmatch and Taste profile data set.

For this experiment, the **Last.fm** dataset has been used. Last.fm is a music web portal that allows its user base, which has more than 30 million active users, to listen to millions of songs from its music library. All the users' activity is recorded in the Last.fm database, which in turn used by the portal to make music

recommendations. The dataset for this experiment contains activities of 48 users whose listening history for the period of 3 years. For every song that a user listens to, its activity is recorded in the following format:
User_000004   2009-04-09T12:49:50Z 078a9376-3c0442807d720e158f345d
 A Perfect Circle
 5ca13249-26da-47bd- bba7-80c2efebe9cd People Are People
Fig. 4.1.1 User Record tuple in the dataset
The above record contains the following fields:
User id (User_000004) – Since the data is captured anonymously, we assigned each user, a user-id of the format user_000004.
Date–Time (2009-04-09T12:49:50Z) – Time of activity is recorded
Album Id (078a9376-3c04-4280-b7d720e158f345d) – A unique identifier is Attributed to each Album.
Album name (A Perfect Circle) – An album to which that song belongs to.
Track id (5ca13249-26da-47bd-bba7-80c2efebe9cd) – A unique identifier is attributed to each track song.
Track name (People are People) – The songs which the user listened to.
Experimental steps
1. To conduct the experiment we have considered the history of 50 students for a period of 3 years
2. To filter the data two constraints are used on the users and items. Consider only those users who listened more than 30 items and only those items who have been listened by at least 3 users.
3. After applying the constraints user-item matrix is obtained
4. Use Threshold_user clusters() algorithm to form user clusters
5. For a given target user find the most similar user cluster
6. Use social network to find immediate friends of the target user. The history of the immediate friends will give the social matrix
7. Find the common items from user cluster and user's side information in the form of social matrix for recommendation

Result

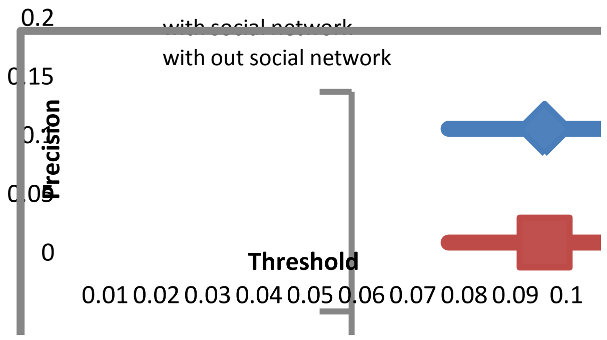


Fig 4.3.1 Threshold Vs Precision with and without Social Network integration

From the above graph we can conclude that as the threshold increased, Precision also increased and experimental analysis proved that the performance of recommender system with Social network integration is improved compared to baseline popular recommender system for Cold-start problem.

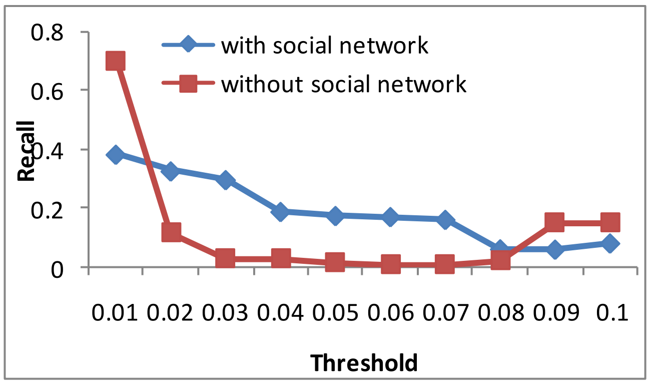


Fig 4.3.2 Threshold Vs Precision with and without Social Network integration

Threshold	Precision	Recall
0.01	0.0355047	0.3820892
0.02	0.0549396	0.3255506
0.03	0.054887	0.2960225
0.04	0.0973042	0.1881605
0.05	0.086108	0.1744613
0.06	0.1122663	0.1708858
0.07	0.1123151	0.1613805
0.08	0.1484416	0.06164
0.09	0.1475431	0.060515
0.1	0.1665505	0.081498

Table 4.3.1 Threshold Vs Precision and Recall

V. Conclusion & Future Scope

In this paper we have proposed and implemented an algorithm which will take both collaborative filtering and social network information into account in order to improve the accuracy of a Recommender system to address the Cold-Start problem. As a future work we can extend this with the communities in social networks and can assign weight or trust measure to the users in Social matrix.

References

[1]. Jovian Lin, Kazunari Sugiyama, Min-Yen Kan, Tat-Seng Chua, Addressing Cold-Start in App Recommendation: Latent User Models Constructed from Twitter Followers, SIGIR’13, July 28–August 1, 2013, Dublin, Ireland

[2]. Daqiang Zhang, Qin Zou, Haoyi Xiong, CRUC: Cold-start Recommendations Using Collaborative Filtering in Internet of Things, Elsevier ESEP, 9-10 December 2011, Singapore

[3]. F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds., Recommender systems handbook. Springer US, October 2011

[4]. Xiwang Yang, Yang Guo, Yong Liu,Bayesian-Inference-Based Recommendation in Online Social Networks, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 24, NO. 4, APRIL 2013

[5]. Ko-Jen Hsiao, Alex Kulesza, and Alfred O. Hero,Social Collaborative Retrieval, arXiv:1404.2342v1 [cs.IR] 9 Apr 2014

[6]. Abbassi Z, Amer-Yahia S, Lakshmanan LVS, Vassilvitskii S, Yu C (2009) Getting recommender systems to think outside the box. In: Proceedings of the third ACM conference on recommender systems RecSys 09, ACM Press, pp 285–288

[7]. Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng 17(6):734–749

[8]. Konstan JA, Riedl J (2012) Recommender systems: from algorithms to user experience. User Model User Adapt Interact 22(1–2):101–123

[9]. Tintarev N, Masthoff J (2007) A survey of explanations in recommender systems. 2007 IEEE 23rd international conference on data engineering workshop, vol 7, pp 801–810

[10]. Zhou X, Xu Y, Li Y, Josang A, Cox C (2011) The state-of-the-art in personalized recommender systems for social networking. Artif Intell Rev 37(2):119–132

[11]. Marinho Balby L, Nanopoulos A, Schmidt-thieme L (2011) Social tagging recommender systems. In: Francesco Ricci et al.(ed) Recommender systems handbook. Springer, pp 615–644

[12]. M. Sunitha , Dr. T. Adilakshmi, Session Aware Music Recommendation System with User-based and Item-based Collaborative Filtering Method, International Journal of Computer Applications, June ,2014

[13]. M. Sunitha Reddy, Dr. T. Adilakshmi, User Based Collaborative Filtering For Music Recommendation System, International Journal of Innovative Research and Development, Dec 2013, Volume 2, Issue 12 pg no 185-190

[14]. M.Sunitha Reddy ,Dr. T. Adilakshmi, Music Recommendation System based on Matrix Factorization technique –SVD, International Conference on Computer Communications and Informatics (ICCCI-14), Coimbatore, 3-5 January, 2014

[15]. Last. FM – A popular music web portal http://www.last.fm