Session Aware Music Recommendation System with Matrix Factorization technique-SVD

M. Sunitha [1], Dr. T. Adilakshmi [2]

[1,2] *Dept. of CSE, Vasavi College of Engineering*
Ibrahimbagh, Hyderabad-31, TS, India.

Abstract- *Recommender systems (RS) serve as valuable information filtering tools for web online users to deal with huge amount of information available on the Internet. RS can be used in making decision in various fields like which books to purchase or which music to listen and so on. In this paper we have proposed and implemented an algorithm based on the Collaborative filtering method and Matrix Factorization technique -SVD. Collaborative filtering is one of the traditional method for Recommendation Systems based on the user feedback. Matrix factorization is a method to address the problem of Sparsity. In this paper , first sessions are formed based on the timestamps of user logs. Collaborative filtering is used to form clusters based on users and items. SVD is applied for the user-item matrix formed from the clusters to address the Sparsity problem. Finally recommendations are given to the new test users by using user and item clusters. Experiments are performed on the benchmark data set for the proposed algorithm and results shows improvement of the recommendation system accuracy over traditional collaborative filtering method.*

Keywords- *Collaborative filtering, recommender system, Item-based clusters ,user-based clusters, Matrix factorization technique, SVD*

I. INTRODUCTION

Large amount of information is available in digital libraries because of the evaluation of Internet. It is difficult for the users to find the information which is interesting and useful for them[5]. This information overload problem leads the use of Recommender Systems (RS) which allows personalization and provides recommendations and suggestions to users in finding useful information.
RS are basically categorized into two main paradigms, Content based and Collaborative recommendation systems[6]. Content based RS are based on the items and recommend the items similar to those items which previously the user liked. Collaborative Filtering RS are based on the user feedback (explicit or implicit). Collaborative Filtering (CF) methods are further divided into two types: User-based CF and Item-based CF.

User-Item matrix is the major data structure for User-based and Item-based CF methods[6]. User based CF technique provides the recommendations based on the user's interest and their neighbor's ratings i.e first we will take user interest into consideration and then the neighbor's ratings who are similar to the target user. The basis for this method is if a test user is similar to some $user_i$, and $user_i$ has rated items { I_1 , I_2, ……} , then recommend those items to the test user.

Music recommender systems are decision support tools that solves the information overload problem by recommending the items that are interesting and relevant to the user, based on the user's music preferences[8][9]. For example , Last.fm a popular Internet radio and recommender system that recommends songs to users based on their interest and other user's rating on those items. It also allows users to get recommendations based on the artist, album and so on.

The main challenges faced by CF techniques are Sparsity, Scalability and Cold-Start [5][6].

Sparsity: As we compare the number of users with the number of items, 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[4][9].

Scalability: Scalability means ability of RS to work with increasing data sets i.e increase in the number of users or items. The time complexity of CF techniques increases exponentially with the increase in the number of users or items as CF techniques are basically dependent on similarity measures[4[9]].

Cold-Start: Cold-start is the problem of not able to recommend items to new users and new items to existing users. This is because CF technique can not recommend items to new users until the 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[4][9].

This paper addresses the problem of Sparsity by using dimensionality reduction technique- SVD[10]. In this paper we used user listening history for collaborative filtering system based on user clusters and item clusters for music recommendation. We also proposed and implemented an algorithm for music recommendation by taking Sessions and SVD into consideration.

The rest of the paper is organized as follows. Section II deals with traditional collaborative filtering methods. Section III describes about the proposed approach. Section IV explains about the experimental set up and Results. Section V describes about conclusion and future directions for research.

II. TRADITIONAL COLLABORATIVE FILTERING ALGORITHMS

A. User -Item Rating matrix

User-Item rating matrix is the heart of collaborative filtering technique. This is obtained from the ratings of m users for n items as shown in Fig 2.1. Unique Users are represented on rows and distinct items are represented on columns. CF technique uses the ratings of the observed item by all users in order to predict the rating for the same item by target user's [1][2]. Each row in the user-item matrix is used to represented users in vector space model i.e each row is a vector representation of a user and can be summarized in a user-item matrix, which contains the Scorings Sij that have been provided by the ith user for the jth item, the matrix as following

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 User-Item Matrix

Where S_{ij} denotes the score of item j rated by an active user i. If user i has not rated item j, then S_{ij} =0. The symbol m denotes the total number of users, and n denotes the total number of items.

B. *Similarity Measures*

CF methods are very popular in RS and are evidenced from the numerous research publications in this area. The basic idea in many of these publications is use similarity measures to find similar items or similar users. CF algorithms that use similarity measures to find similar users is known as user-based CF and similar items is known as Item-based CF[4][5].

Similarity measures are evaluated as a metric of similarity between two users by using 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 balance the ratings significance in a Recommendation algorithm to improve accuracy[1][2].

The following are the different similarity measures used in CF technique. [1][3] Pearson correlation, cosine vector similarity and adjusted cosine vector similarity etc.

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 of the angle between them. The cosine of two vectors can be derived by using the Euclidean dot product formula:

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \, \|\mathbf{b}\| \cos\theta$$

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** as the formula given below, is used in some collaborative filtering methods to find similarity among users where the difference in each user's rating scale is taken into account.

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

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 APPROACH FOR RECOMMENDATIONS

This section describes about the dissimilarity measure used, formation of sessions, applying dimensionality reduction technique SVD on sessions, formation of user-based clusters and item-based clusters on the reduced user-item matrix, recommendation of items by taking sessions into consideration to new users and evaluation measures.

A. Euclidean Distance

Euclidean distance is a measure of dissimilarity. It measures the distance between two points by the Pythagorean formula given below.

The distance from **p** to **q** is given by

$$d(p,q) = d(q,p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

a. Formation of Sessions

User logs are divided into sessions. A session is defined as fixed time slot of a day. We have taken four sessions for each day of equal intervals i.e from 0 a.m to 6 a.m as S1, 6a.m to 12 p.m as S2 , 12 p.m to 18 p.m as S3 and 18p.m to 24p.m as S4.

B. Applying SVD

SVD is a matrix factorization technique for dimensionality reduction. SVD van be viewed from three different point of views. First is SVD transforms data represented in correlated attributes into non-correlated attributes. Second is SVD identifies and orders the dimensions used to represent the data in point representation. Third is identifying the best and few number of dimensions to represent the data[10].

The third feature of SVD is also known as dimensionality reduction. Dimensionality reduction is useful in case of data with large number of dimensions. As the number of dimensions increases the data becomes Sparse and data points will be scattered over the space. If we apply traditional clustering algorithms on the sparse data, it will not form any useful clusters. To make the clustering meaningful in high dimensional data, generally dimensionality reduction techniques are applied. These techniques represent high dimensional data with few dimensions which can represent original data i.e without loss of information.

SVD decomposes the matrix A of size mxn into three marices U,S and V as given below

$$A_{mxn} = U_{mxr} \; X \; S_{rxr} \; X \; V_{rxn}$$

Where U represents the users, S is an identity matrix in the order of Eigen values and V represents items in transpose form.

In this paper we have used SVD as the dimensionality reduction technique at the session level. After forming sessions SVD is applied on user-item matrix to reduce the size of the matrix.

C. User-based Clusters

Each user from the user-item matrix of each session (S_1,S_2,S_3 S_4) is considered as a user vector. User clusters for a session are formed by using the following hierarchical agglomerative clustering algorithm.[1]

Algorithm User_clusters_with SessionsandSVD()
Input: Reduced User-Item Matrix of a particular session
Output: User Clusters
Method:
begin

1. Consider each user vector I_1, I_2,..I_k where k is the number of distinct items rated by all users
2. Initialize threshold_cutoff value
3. Consider the first user and put in C_1
4. For all remaining users repeat the steps from 4 to 8
5. Find the similarity of the $user_i$ with all the clusters formed so far
6. Put the $user_i$ in the cluster with more similarity
7. If the $user_i$ is not in the threshold value of any cluster
8. Create a new cluster

end

Fig. 2.1 Pseudocode for User_clusters_with Sessions

D. Item-based Clusters

Each item from the user-item matrix of each session (S_1,S_2,S_3 S_4) is considered as an item vector. Item clusters for a session are formed by using the following hierarchical agglomerative clustering algorithm.[1]

Algorithm Item_Clusters with SessionsandSVD()
Input: Reduced User-Item Matrix of a particular session
Output: Item Clusters
Method:

begin

1. Consider each item vector U_1, U_2,..U_k where k is the number of distinct users rated an item
2. Initialize threshold_cutoff value
3. Consider the first item and put in C_1
4. For all remaining items repeat the steps from 4 to 8
5. Find the similarity of the $item_i$ with all the clusters formed so far
6. Put the $item_i$ in the cluster with more similarity
7. If the $item_i$ is not in the threshold value of any cluster
8. Create a new cluster

end

Fig. 2.2 Pseudocode for Item_clusters_with Sessions

E. Recommendation stage

After getting the user clusters and item clusters for each session, we use these clusters to recommend items to new users by using the following Algorithm for recommendations.[1]

Algorithm Recommendation_ with SessionsandSVD ()
Input: User Clusters and Item Clusters
Output: Set of Recommendations for new users
Method:

begin

1. map the new user to the user clusters to which he/she is most similar
2. map the new user to the item cluster based on the items listened
3. consider the recommendations from step1 i.e user clusters and step2 i.e item clusters for the new user
4. Let I_1, I_2, ….I_k are the items which are common in both recommendations (user clusters and item clusters)
5. recommend the common items to the new user

end

Fig.2.3.Pseudocode for Recommendation_with Sessions

F. 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) and f-measure. These measures are obtained from confusion matrix shown in Fig. 2.4.[1][9]

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.5 Confusion Matrix

Precision is a measure of exactness [1], determines the fraction of relevant items retrieved out of all items retrieved. Recall is a measure of completeness, 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.

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

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

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

IV EXPERIMENT AND RESULTS

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

A. Data set

Million Song Dataset (MSD) a freely-available collection of audio features and meta-data for a million contemporary popular music tracks [8]. The MSD contains extensive meta-data, audio features, tags on the artist- 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. 6 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
AlbumId (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.
Trackid(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.

B. Experimental setup

We have taken 100000 records from Last.fm data set for this experiment. It consists of 48 users listening history for 3 months. We have taken all the items which are listened by at least 2 users. With this constraint on the data set we got 22000 unique items. From 48 users 33 users are taken as training data and 15 users are taken as test data..

	Song 1	**Song 2**	**….**	**Song 22000**
User 1	2	0	....	0
User 2	0	4	....	1
....	....	....	....	....
User 38	0	0	....	1

Table 4.1. User-Item Matrix for 38 X 22000

C. Applying SVD on User-Item matrix of a session

SVD is applied on each session's user-item matrix by taking k=2 i.e. two most significant Eigen values into consideration. With this step the user-item matrix is decomposed into three matrices U,S and V as shown in the fig .

Matrix U

0.036697123	0.001598783	0.087417695	0.004824194
0.00504383	0.045794877	0.046870156	0.003087831
0.003826221	1.49E-04	1.27E-04	7.53E-04
0.004220399	0.001268878	0.10532634	0.021120419
0.001713249	0.003035872	4.57E-03	0.006122999

Matrix V

0.56444245	-0.024768703	0.82310289	-0.038961494
0.014803129	-0.031332759	0.02376704	4.13E-04
0.002896579	0.013851497	0.060281442	0.036031687
0.704171698	-0.704036471	-0.046819038	0.016944794

Matrix S

92.65579181	0	0	0
0	87.696145	0	0
0	0	56.97938947	0
0	0	0	27.5252295

D. **Results**

We have repeated the experiment with various values of thresholds such as 0.01, 0.02 and so on till 0.1 without sessions and with sessions and SVD.

We plotted the graph for threshold vs Precision for traditional Collaborative filtering and new session based Collaborative filtering with SVD. We can conclude from the experimental results that as the threshold value increases the Precision also increases. We can also conclude from the experiment that the Precision of new proposed session aware method has improved over traditional method.

Threshold / Avg. Precision	with session	without session
0.01	0.0355047	0.0262619
0.02	0.0549396	0.0450402
0.03	0.054887	0.0460341
0.04	0.0973042	0.0795551
0.05	0.086108	0.0606562
0.06	0.1122663	0.0707849
0.07	0.1123151	0.083704

Table 4.2. Threshold Vs Average Precision

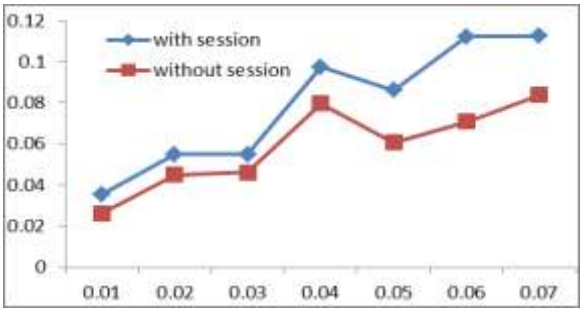

Fig.4.1 Threshold Vs Average Precision

Threshold / Avg. Recall	with session	without session
0.01	0.3820892	0.3029991
0.02	0.3255506	0.1189198
0.03	0.2960225	0.0283582
0.04	0.1881605	0.0271492
0.05	0.1744613	0.011988
0.06	0.1708858	0.0082911
0.07	0.1613805	0.0080912

Table 4.3. Threshold Vs Average Recall

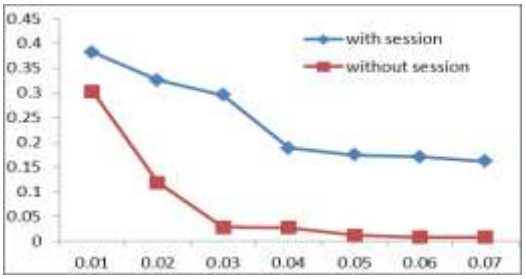

Fig.4.2. Threshold Vs Average Recall

User/ True Positives	with session	without session
U_1	8	0
U_5	2	0
U_{13}	8	2
U_{20}	3	2
U_{22}	5	1
U_{24}	101	80
U_{29}	6	0
U_{38}	0	0
U_{49}	0	0
U50	45	0

Table 4.4. Users Vs no. of True Positives for $Session_4$

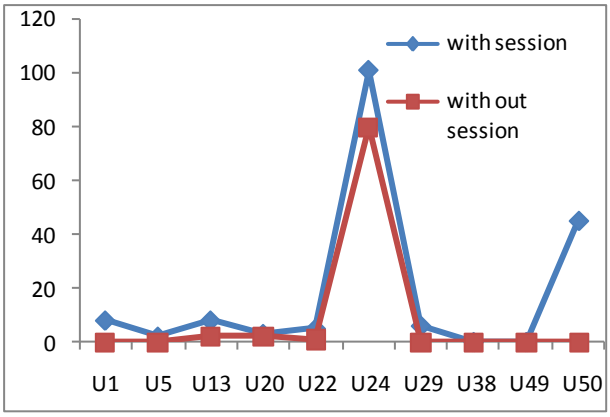

Fig.4.3. Users Vs no. of True Positives for Session$_4$

Threshold/ P, R, F	Precision (P)	Recall (R)	F-measure (F)
0.01	0.0762619	0.7029991	0.1375972
0.02	0.0850402	0.1189198	0.0991661
0.03	0.0860341	0.0283582	0.0426563
0.04	0.0795551	0.0271492	0.0404831
0.05	0.0606562	0.011988	0.0200195
0.06	0.0707849	0.0082911	0.0148436
0.07	0.083704	0.0080912	0.014756
0.08	0.1102565	0.0196916	0.0334153
0.09	0.1394788	0.1519046	0.1454268
0.1	0.1549326	0.1513282	0.1531092

Table 4.5. Precision ,Recall and F-measure for without session

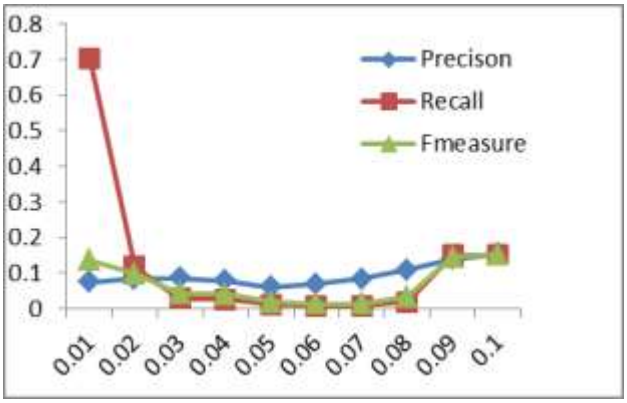

Fig.4.4 Precision ,Recall and F-measure for without session

Threshold/ P, R, F	Precision(P)	Recall (R)	F-measure (F)
0.01	0.0355047	0.3820892	0.064972
0.02	0.0549396	0.3255506	0.0940136
0.03	0.054887	0.2960225	0.0926039
0.04	0.0973042	0.1881605	0.1282737
0.05	0.086108	0.1744613	0.1153053
0.06	0.1122663	0.1708858	0.1355082
0.07	0.1123151	0.1613805	0.1324498
0.08	0.0884416	0.06164	0.0726477
0.09	0.0975431	0.060515	0.0746918
0.1	0.1065505	0.081498	0.0923554

Table 4.6. Precision ,Recall and F-measure for with session

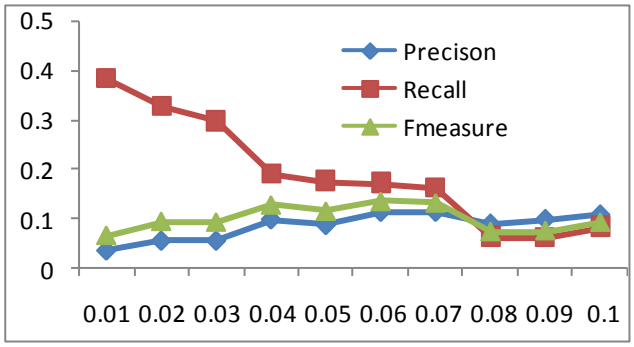

Fig. 4.5. Precision ,Recall and F-measure for with session

Threshold / Avg. Precision	with Session and SVD	with session	without session
0.01	0.0432131	0.0355047	0.0262619
0.02	0.0589237	0.0549396	0.0450402
0.03	0.0598542	0.054887	0.0460341
0.04	0.0987531	0.0973042	0.0795551
0.05	0.0989273	0.086108	0.0606562
0.06	0.0172853	0.1122663	0.0707849
0.07	0.0183862	0.1123151	0.083704

Table 4.7 Avg. Precision with SVD, with Session and with out Session

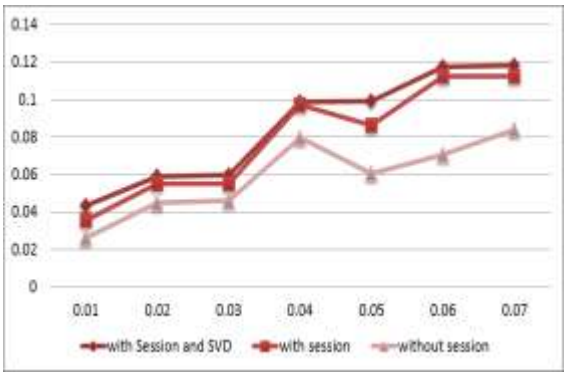

Fig 4.6. Threshold Vs Average Precision

Threshold / Avg. Recall	with Session and SVD	with session	without session
0.01	0.3960432	0.3820892	0.3029991
0.02	0.3865521	0.3255506	0.1189198
0.03	0.3060325	0.2960225	0.0283582
0.04	0.2781026	0.1881605	0.0271492
0.05	0.2544761	0.1744613	0.011988
0.06	0.2303228	0.1708858	0.0082911
0.07	0.2112871	0.1613805	0.0080912

Table 4.8. Avg.Recall with SVD, With Session and with out Session

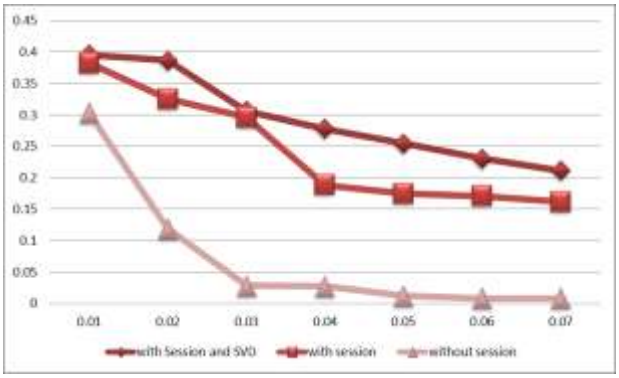

Fig 4.7. Threshold Vs Average Recall

V. CONCLUSION AND FUTURE SCOPE

We have discussed about the Session aware music recommendation system with matrix factorization technique- SVD. The proposed algorithm takes the user interest in the form of user logs into consideration without taking the user feedback explicitly to address the Sparsity problem. We also evaluated our system on benchmark dataset. We showed that session aware recommendation system with SVD, the dimensionality reduction technique improved traditional collaborative filtering technique. This work can be extended for recommendations to address Cold start problem by taking other user related information such as user demographic information into consideration.

VI. REFERENCES

[1]. 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

[2]. 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

[3]. 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

[4]. Context-aware item-to-item recommendation within the factorization framework, Balázs Hidasi, Domonkos Tikk, CaRR'13, February 5, 2013, Rome, Italy

[5]. Introduction to Recommender Systems, Markus Zanker, Dietmar Jannach, Tutorial at ACM Symposium on Applied Computing 2010 ,Sierre, Switzerland, 22 March 2010

[6]. A Collaborative Filtering Recommendation Algorithm Based on User Clustering and Item Clustering, SongJie Gong, Journal Of Software, Vol. 5, No. 7, July 2010

[7]. NetflixPrize, http://www.netflixprize.com/, 2012.

[8]. The Million Song Dataset Challenge, Brian McFee, Thierry Bertin-Mahieux, Daniel P.W. Ellis, Gert R.G. Lanckriet, WWW 2012 Companion, April 16–20, 2012, Lyon, France

[9]. Adomavicius, G., Tuzhilin, A. 2005. Toward the Next Generation of Recommender Systems:A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactionson Knowledge and Data Engineering 17, 734–749.

[10]. Singular Value Decomposition http://en.wikipedia.org/wiki/Singular_value_decomposition