# Music Recommendation System based on Matrix Factorization technique -SVD

M. Sunitha Reddy [1]
[1] Dept. of CSE, Vasavi College of Engineering
Ibrahimbagh, Hyderabad-31, AP, India.
sashu2006@gmail.com

Dr. T. Adilakshmi [2]
[2] Dept. of CSE, Vasavi College of Engineering
Ibrahimbagh, Hyderabad-31, AP, India.
t_adilakshmi@rediffmail.com

Abstract-**Recommender systems have been proven to be valuable means for web online users to cope with the information overload and have become one of the most powerful and popular tools in electronic commerce. With the development of electronic commerce systems, the magnitudes of users and items grow rapidly, resulted in the extreme sparsity of user rating data set. Traditional similarity measure methods work poor in this situation, make the quality of recommendation system decreased dramatically. Sparsity of users' ratings is the major reason causing the poor quality. To address this issue, Item based collaborative filtering recommendation algorithm based on singular value decomposition (SVD) is presented. This Paper uses SVD for dimensionality reduction, and then uses Euclidian distance as dissimilarity measure to find the target users' neighbors, lastly produces the recommendations. The collaborative filtering recommendation algorithm based on SVD can alleviate the sparsity problems of the user item rating dataset, and can provide better recommendation than traditional collaborative filtering algorithms.**

**Index Terms- collaborative filtering, recommender system, sparsity, dimensionality reduction, singular value decomposition**

## 1. INTRODUCTION

Due to the evaluation of internet and e-commerce users are able to get large volumes of information. This problem is known as Information overloading. It is very difficult to get the information which is interesting to the users. Recommender systems are the one which serves as a good tool for information filtering. Information filtering can be done by using content based filters and Collaborative filters. Content based method depends on the content of the item whereas collaborative filtering is based on user's ratings.[1,2]

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]. 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.

Many researchers have proposed different kinds of Collaborative filtering techniques to do quality recommendations to users. CF can be performed based on two different methods. One is User based CF technique and the other is Item based CF technique [3,4]. Both these methods are based on the data structure, User-Item matrix. User based CF technique does 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.

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.

**Scalability:** Scalability is another important problem faced by CF techniques. The time complexity of CF techniques increases non linearly 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 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

978-1-4799-2352-6/14/$31.00 ©2014 IEEE

to users until the items are being rated by sufficient number of users.

This paper proposed an algorithm to solve the problem of sparsity by using Singular value decomposition (SVD) , a dimensionality reduction technique. The proposed approach uses Item based clustering for recommendations.

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 up and Results. Section V gives a brief about conclusion and future directions for research

## II. TRADITIONAL USER-BASED COLLABORATIVE FILTERING ALGORITHM

### a. User -Item Rating matrix

The main goal of traditional User-based Collaborative filtering technique is to recommend items to target users based on the ratings of the target user and the ratings of the users similar to the target user. The data structure used in traditional user based CF technique, User-Item matrix as shown in Fig. 1 represents each user as a vector which contains the ratings $R_{ij}$ given by the $i^{th}$ user for the $j^{th}$ item.

| Item /User | $Item_1$ | $Item_2$ | … … | $Item_n$ |
|---|---|---|---|---|
| $User_1$ | $R_{11}$ | $R_{12}$ | … … | $R_{1n}$ |
| $User_2$ | $R_{21}$ | $R_{22}$ | … … | $R_{2n}$ |
| … … | … … | … … | … … | … … |
| $User_m$ | $R_{m1}$ | $R_{m2}$ | … … | $R_{mn}$ |

Fig. 1 User-Item matrix

Where $R_{ij}$ denotes the Rating of $item_j$ by an active $user_i$. If user i has not rated $item_j$, then $R_{ij}$ =0. The symbol m denotes the total number of users, and n denotes the total number of items.

### b. Similarity Measures

Collaborative filtering techniques have been very widely used by researchers and practitioners which is evident from the large number of publications and implementation cases. The basic idea of User based CF techniques is to use some similarity measure to find similarity among the users and recommend items based on the ratings of the similar users. [9]

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. The similarity measure can be effectively used to balance the ratings significance in a Recommendation algorithm to improve accuracy.

The following are the different similarity measures used in user based CF technique. [1,3] Pearson correlation, cosine vector similarity, adjusted cosine vector similarity, mean- squared difference and Spearman correlation.

**Pearson's correlation**, measures the linear correlation between two vectors of ratings.

$$Sim(i,j) = \frac{\sum_{c \in I_{ij}} (R_{i,c} - A_i)(R_{j,c} - A_j)}{\sqrt{\sum_{c \in I_i} (R_{i,c} - A_i)^2 \sum_{c \in I_j} (R_{j,c} - A_j)^2}}$$

Where $R_{i,c}$ 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 0° is 1, and it is less than 1 for any other angle. Two vectors with the same orientation have a Cosine similarity of 1, two vectors at 90° have a similarity of 0, and two vectors diametrically opposed have a similarity of -1, independent of their magnitude. Cosine similarity is particularly used in positive space, where the outcome is neatly bounded in [0,1].

$$Sim(i,j) = \frac{\sum_{k=1}^{n} R_{ik} R_{jk}}{\sqrt{\sum_{k=1}^{n} R_{ik}^2 \sum_{k=1}^{n} R_{jk}^2}}$$

Where $R_{ik}$ is the rating of the $item_k$ by $user_i$ and n is the number of items co-rated by both users. And if the rating is null, it can be set to zero.

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.

$$sim(i,j) = \frac{\sum_{c \in I_{ij}} (R_{i,c} - A_c)(R_{j,c} - A_c)}{\sqrt{\sum_{c \in I_{ij}} (R_{i,c} - A_c)^2 \sum_{c \in I_{ij}} (R_{j,c} - A_c)^2}}$$

Where $R_{i,c}$ 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 $user_j$.

### c. Formation of user clusters

Based on the similarity measure users are mapped into clusters. Two techniques have been employed in the collaborative filtering recommender systems.

Threshold-based selection, according to which users whose similarity exceeds a certain threshold value are considered as neighbors of the target user.

### d. Recommendation stage

The target user is mapped to the cluster with highest similarity and top-n items are given as the best recommendations.

## III. PROPOSED APPROACH FOR RECOMMENDATIONS

This section describes about the dissimilarity measure used, Singular Value Decomposition technique for dimensionality reduction, formation of item clusters, recommendation of items to new users and evaluation measures.

### a. Euclidean Distance

Euclidean distance or Euclidean metric is the distance between two points that one would measure with a ruler, and is given by the Pythagorean formula.

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

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

### b. Singular Value Decomposition (SVD)

The main goal of SVD is dimensionality reduction i.e taking high dimensional data and decomposing it to low dimensional data. Singular Value Decomposition (SVD) [7] is an approach where we factorize a matrix into a series of linear approximations. These approximations will expose the underlying structure of that matrix. SVD can be expressed from three consistent viewpoints.

First, SVD transforms a matrix of seemingly correlated variables into an uncorrelated one that provides a better understanding of the relationship between all the data points which might not be obvious in their original formations. This is helpful because in some cases the relationships might be confusing; or it may suggest another relationship between 2 songs rather than what is apparent.

Secondly, SVD is used to identify the relationship between various items (mapped as data points in the matrix) and aligns the data in the product matrix so that the data points show the maximum variation.

Thirdly, once we have figured out the vectors having most variations; it is possible to find the best approximations in the original data by using fewer dimensions.

SVD is very useful dimensionality reduction technique. SVD has a wide range of applications including signal processing, Latent Symantec Analysis, Pattern recognition, low range matrix approximation and weather prediction.

The definition of SVD theorem is as given below:

Consider a Matrix M with m rows and n columns. The SVD theorem in linear algebra states that Such a matrix can be decomposed into a product of three matrices, which can be represented by following equation :

$$\boxed{X_{mxn} = U_{m \times r} S_{r \times r} V_{r \times n}}$$

### c. Item-based Clusters

**Algorithm** Item_Cluters()
Begin
Each item is represented as a Vector ($u_1, u_2, \ldots u_{10}$)
Initialize the threshold_cutoff
Put $item_1$ into cluster $C_1$
For remaining items find the similarity with mean of $C_1$
If it is within threshold_cutoff then put item into same cluster
Else
Create a new cluster
Return the clusters $C_1, C_2, C_3 \ldots C_k$
End

### d. Recommendation of items

After getting the item clusters, we used these clusters to recommend items to new users. The following algorithm is used for recommendations

**Algorithm** Recommendation ()
Begin
For each new user
Find the similarity with each cluster mean
Find the cluster with highest similarity
Then recommend the items Rated by the users in the cluster
End

### e. 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) as the evaluation measure.

**Confusion Matrix**

A confusion matrix shows the number of correct and incorrect prediction made by the classification 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. Confusion Matrix

Precision (P)= (TP ) / (TP + FP)

Precision is calculated by taking each user as test user and remaining users as training users. Let $P_1$ , $P_2$...... $P_{10}$ are the precisions of 10 users, the average precision is calculated by the following equation

Avg. Precision= ($P_1 + P_2 + \ldots + P_{10}$ ) / 10

## IV EXPERIMENT AND RESULTS

This section describes about the Dataset used for this 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 con- temporary popular music tracks [9]. Comprising several complementary datasets that are linked to the same set of songs, 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. We used Last.fm logs as Data set for our experiment

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 10 users whose listening history has been captured anonymously for the period of 3 months. For every song that a user listens to, its activity is recorded in the following format:

User_000004 2009-04-09T12:49:50Z
078a9376-3c04-4280-b7d7-b20e158f345d
A Perfect Circle 5ca13249-26da-47bd-
bba7-80c2efebe9cd People Are People

Fig.3. 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 which will be
used in our algorithm to determine the session in which it will belong.
**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.

### b. Experimental setup

We have taken 22000 records from Last.fm data set for this experiment. It consists of 10 users listening history for 3 months and 8240 unique items. We have taken only those items which are listened by at least two users. With these constraints we got 10 unique users and 242 unique items. We formed item clusters by taking 10 X 242 user-item matrix into consideration.

| | $Song_1$ | $Song_2$ | $Song_3$ | $Song_4$ | $Song_5$ |
|---|---|---|---|---|---|
| $User_1$ | 2 | 0 | 0 | 0 | 0 |
| $User_2$ | 0 | 4 | 0 | 1 | 0 |
| $User_3$ | 1 | 3 | 0 | 0 | 0 |
| $User_4$ | 0 | 0 | 0 | 1 | 1 |

Fig.4. Part of User-Item Matrix for 10 X 242

### c. Applying SVD

Once we got User-Item matrix, we applied SVD to it by using JAMA (Java Matrix package). The following are the results that we got for U, V, and S matrices.

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 done the experiment with various values of thresholds such 0.1, 0.15 and so on till 0.4.

We plotted the graph for threshold vs Precision and threshold vs number of clusters. We can conclude from this experiment that as the threshold value increases the Precision decreases and number of clusters also decreases.

**Threshold Vs Average Precision**

Threshold x=[0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4];

Average Precision y=[ 0.8999 0.7999 0.76321 0.6563 0.23493 0.20796 0.198499]

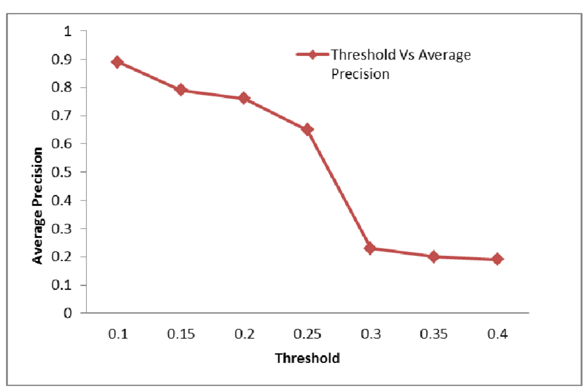


Fig. 5. Threshold Vs Average Precision

Threshold vs no of Clusters

Threshold x=[0.1 0.15 0.2 0.25 0.3 0.35 0.4]

Clusters y=[45 40 35 30 26 20 14]

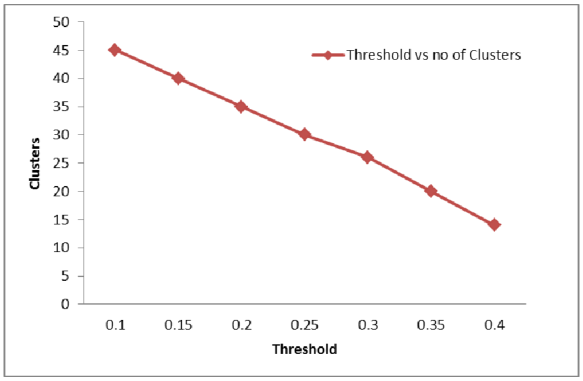


Fig. 6. Threshold vs no of Clusters

## V. CONCLUSION AND FUTURE SCOPE

We have discussed about the item based collaborative filtering method for music recommendation system with matrix factorization technique-SVD. This system is taking the user interest into consideration without taking the user feedback explicitly as user logs are one of the implicit feedback. We addressed the problem of Sparsity by using SVD which is a dimensionality reduction technique. We also evaluated our system on benchmark dataset. This work can be extended for recommendations by taking the sessions into consideration.

## VI. REFERENCES

[1] Breese J, Hecherman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI'98). 1998. 43~52.

[2] Chong-Ben Huang, Song-Jie Gong, Employing rough set theory to alleviate the sparsity issue in recommender system, In: Proceeding of the Seventh International Conference on Machine Learning and Cybernetics (ICMLC2008), IEEE Press, 2008, pp.1610-1614.

[3] Sarwar B, Karypis G, Konstan J, Riedl J. Item-Based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International World Wide Web Conference. 2001. 285-295.

[4] Manos Papagelis, Dimitris Plexousakis, Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents, Engineering Application of Artificial Intelligence 18 (2005) 781-789.

[5] Hyung Jun Ahn, A new similarity measure for collaborative filtering to alleviate the new user cold- starting problem, Information Sciences 178 (2008) 37-51.

[6] SongJie Gong, The Collaborative Filtering Recommendation Based on Similar-Priority and Fuzzy Clustering, In: Proceeding of 2008 Workshop on Power Electronics and Intelligent Transportation System (PEITS2008), IEEE Computer Society Press, 2008, pp. 248-251.

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

[8] Duen-Ren Liu, Ya-Yueh Shih, Hybrid approaches to product recommendation based on customer lifetime value and purchase preferences, The Journal of Systems and Software 77 (2005) 181–191.

[9] Million Song Dataset, official website by Thierry Bertin-Mahieux, available at: http://labrosa.ee.columbia.edu/millionsong/.