Music Recommendation Engine

[1]Soni Kumari. [2]Prof. B. I. Khodhanpur

[1]M.Tech Scholar, [2]Professor
Dept. Computer Science & Engineering
RNS Institute of Technology, Bangalore, India
[1]sonichiraniya@gmail.com, [2]bi.khodanpur@gmail.com

***Abstract*—A Recommender system for music data is proposed here, which assists customers in searching music data and provides results with items using his own preference. This system first extracts unique properties of music like Genre, Artist, Album, Language, Length and Year from the music file. This extracted data is stored on the database. Each stored property is analyzed using item based Collaborating filtering algorithm and cosine similarity method. After acquiring records, the system recommends items appropriate to user's own choice. The main objective of this paper is to design a recommender that exploits advantages of item based Collaborating filtering for high quality of recommendation. This approach is one of the most powerful search techniques for finding an optimized solution of matching songs to be recommended to the user.**

***Index Terms*—Component, filtering, uploading, styling, insert**

I. Introduction

Recommender systems have been widely adopted by e-Commerce websites to suggest products or services to customers. In general, they assist users to narrow down their choices for making a purchase decision from a large pool of items. One simple way of offering recommendation is based on the top selling products. This however does not differentiate customers who may have different tastes. The other approach is known as one-to-one marketing, which takes into account the information about a particular user and tries to find a Personal match of product that is predicted to best suit his or her flavor. Generally, there are three methods of the recommendation systems for one-to-one marketing: Content-based Filtering, Collaborative Filtering and Hybrid Model.

Content-based Filtering selects items based on the correlation between the content of the items and user preferences. Current search engines are based on automatic analysis of the content of documents and the content of user's query. As an example, movie recommendation application, in order to recommend movies to a user, the content based recommender system tries to understand the commonalities among the movies that the user has rated highly in the past (specific actors, directors, genres, subject matter, etc.). Movie item D is hence recommended because it is has a certain high degree of similarity to Item A by its content. For another example, a Personalized Recommender System [1] creates dynamic hyperlinks on a web site that contains a collection of advises about do it yourself home improvement. The advices are recommended based on the similarity that they have, to the ones the user has highly rated in the past. In addition to hypertext content, the same concept on content analysis was extended in [2] for recommending multimedia products such as mp3 music on a recommender website

One common problem with the Content-based recommendation system is that it can only recommend items scoring highly against the user profile; so the user is restricted to see the items similar to those already rated, new items will never be recommended due to the working on an individual user.

Collaborative Filtering (CF) based on the similarity between currently active user and other users, finds new items the active has never seen before but they were guessed to be interested by him because the other users who have similar interest to his have seen/liked. The similarity can either be measured by the same item which known as item-based CF or by the same type of user, known as user-based CF. This method is to suggest new items or to predict the utility of a certain item for a particular user based on his previous likings or the opinions of other like-minded users.

II. Background

In the early 1990s, collaborative filtering began to arise as a solution for dealing with overload in online information spaces. Tapestry [8] was a manual collaborative filtering system: it allowed the user to query for items in an information domain, such as corporate e-mail, based on other users' opinions or actions ("give me all the messages forwarded by John"). It required effort on the part of its users, but allowed them to harness the reactions of previous readers of a piece of correspondence to determine its relevance to them.

The capacity of computers to provide recommendations was recognized fairly early in the history of computing. Grundy, a computer based librarian, was an early step towards automatic recommender systems. It was fairly primitive, grouping users into "stereotypes" based on a short interview and using hard-coded information about various sterotypes book preferences to generate recommendations, but it represents an important early entry in the recommender systems space.

During this time, recommender systems and collaborative filtering became an topic of increasing interest among human–computer interaction, machine learning, and information retrieval researchers. This interest produced a number of recommender systems for various domains, such as Ringo [5] for music, the BellCore Video Recommender [6] for movies, and Jester [7] for jokes. Outside of computer science, the marketing literature has analyzed recommendation for its ability to increase sales and improve customer experience.

III. Ease of Use

Recommender system for music data is proposed which assists customers in searching music data and provides result with items resulting in own user preference. This system first extracts unique properties of music like Genre, Artist, Album, Language, Length and Year from the music file. This extracted data is stored in the database. Each stored property is analyzed using item based Collaborating filtering. After acquiring records, the system recommends items appropriate to user's own favourite.

The main objective of this paper is recommender systems by applying a different approach item-based algorithm. The bottleneck in conventional collaborative filtering algorithms is the search for neighbours among a large user population of potential neighbours. Item-based algorithms avoid this bottleneck by exploring the relationships between items first, rather than the relationships between users. Recommendations for users are computed by finding items that are similar to other items the user has liked.

There are a number of different ways to compute the similarity between items. Here we present three such methods. These are cosine-based similarity, correlation-based similarity and adjusted-cosine similarity. But here for getting similarity using cosine-based similarity.

In this case, two items are thought of as two vectors in the m dimensional user-space. The similarity between them is measured by computing the cosine of angle between these two vectors, Formally, in the m*n rating matrix, similarity between item i and j, denoted by *sim(i,j)*.

IV. Proposed System

Most recommenders today rely on an "Item-based collaborating filtering" algorithm, which calculates the distance between each pair of items or songs by Lata mangeshkar are likely to rate song by Asha Bhosle highly, so songs by Lata mangeshkar and Asha Bhosle are in the same neighbourhood. Distances between pairs of items, which may be based on the likings of thousands or millions of users, tend to be relatively stable over time, so recommenders can recomputed distances and generate recommendations more quickly. Both Amazon and Netflix have said publicly that they use variants of an item-item algorithm, though they keep the details secret.

A. Item-Based Collaborative Filtering Algorithm

Item-based recommendation algorithms for producing recommendations to users. Unlike the user-based collaborative filtering algorithm. The item-based approach looks into the set of items the target user has liked and computes how similar they are to the target item i and then selects k most similar item {i1, i2,....., ik}. At the same time their corresponding similarities {si1, si2,... sik} are also computed.

B. Item Similarity Computation

One critical step in the item-based collaborative filtering algorithm is to compute the similarity between items and then to select the most similar items. The basic idea in similarity computation between two items i and j is to first isolate the users who have liked one of these item and then to apply a similarity *sim(i,j)*. Figure1 illustrates this process; here the matrix rows represent users and columns represent items.

There are a number of different ways to compute the similarity between items. Here we present three such methods. These are cosine-based similarity, correlation-based similarity and adjusted-cosine similarity. But in this project for getting similarity we are using cosine-based similarity.

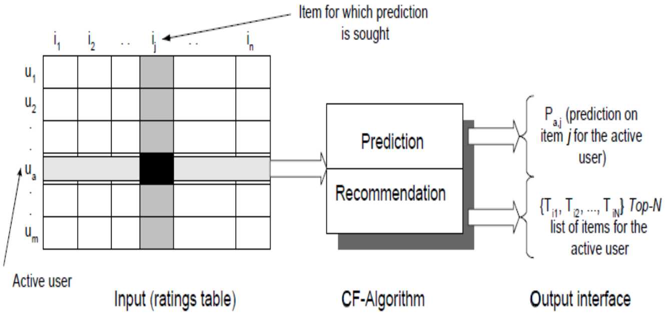

Item for which prediction is sought
i_1
i_2
. .
i_j
. .
i_n
u_1
u_2
u_a
u_m
Active user
Prediction
Recommendation
$P_{a,j}$ (prediction on item j for the active user)
{T_{i1}, T_{i2}, ..., T_{iN}} Top-N list of items for the active user
Input (ratings table)
CF-Algorithm
Output interface

Figure1 Collaborating filtering

C. Cosine Similarity

In this method, two items are thought of as two vectors in the m dimensional user-space. The similarity between them is measured by computing the cosine of angle between these two vectors, Formally, in the m*n rating matrix in Figure1, similarity between item i and j, denoted by *sim(i,j)* is given by.

$$sim(i,j) = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\|_2 * \|\vec{j}\|_2}$$

where "·" denotes the dot-product of the two vectors.

For example

There are five songs with four features Genre, Artist, Composer and Language which have vector values suppose any user likes song s1 so we will check similarity between all the songs so here we have calculated similarity between s1 and s2 which shows the distance between these two songs like that we will calculate similarity between all songs in that which songs have less distance that we will recommend to the user.

Song	Genre	Artist	Composer	Language
S1	1	1	0	1
S2	2	0	1	1
S3	1	3	2	4
S4	0	2	1	2
S5	3	1	4	5

$$\frac{1\cdot 2+1\cdot 0+0\cdot 1+1\cdot 1}{\sqrt{1^2+1^2+0^2+1^2}\sqrt{2^2+0^2+1^2+1^2}} \simeq 0.72$$

Here, the value 0.72 shows similarity between two songs S1 and S2.

V. WORKING

With item-based collaborative filtering as shown in Figure b, if many users (User2 and User3 in this example) like Item A and Item D, we assume that Item A is highly correlated with Item D. If User1 likes Item A, he should also like Item D given the strong link of association between Items A and D.

Item based Collaborative Filtering is quite common for current recommendation systems, which have been wildly used by Movie Lens and Yahoo. According to papers [3][4], item-based techniques first analyze a user-item matrix to identify relationships between different items, and then use these relationships to indirectly compute recommendations for users. With user-based collaborative filtering as shown in Figure c, the taste of User3 is very similar to that of User1 because they have preferred Item A and Item B in common. They are deemed to be like-minded users. Hence User 3 likes Item D so will User1.

Furthermore, it was argued in [5] that in real life the way in which two people are said to be similar is not based solely on whether they have complimentary opinions on a specific subject, e.g., Songs ratings, but also on other factors, such as their background and lifestyles. Therefore, when doing the profile matching, issues such as age, gender and preferences of Songs genres must also be taken into account.

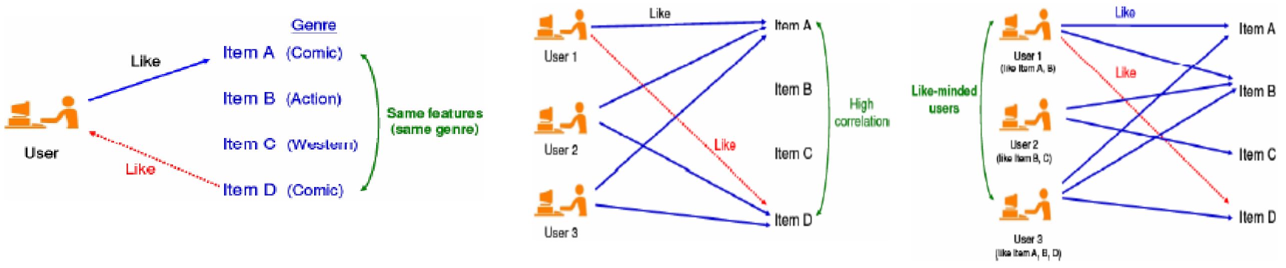
A novel framework for user-based collaborative filtering is proposed in [6] that enables recommendation by groups of closely relate individuals. By using the rating information from a group of closely related users, unrated items of the individual user in a group can be predicted.

VI. CONCLUSION

The recommender system that was able to accurately recognize the trend in user's preferences and adaptively provide an appropriate recommendation, in an efficient manner. This application allows users to get accurate recommendation on the basis of user's preferences and his likes.

Various modules were designed using flow charts. Each module was tested i.e. unit testing was carried out and testing results were satisfactory. Integration of various modules was carried out. Finally the application was tested using item-based collaborating filtering algorithm.

The proposed system initially extracts the unique features of each song using the item-based filtering. next, cosine similarity operates on the song to discover the most appropriate songs to be recommended to the user. to this end, the proposed system combined the strengths of item-based collaborative filtering algorithm and cosine similarity method



(a) Content-based Filtering (b) Item-based Collaborative Filtering (c) User-based Collaborative Filtering

REFERENCES

[1] R. Meteren, M. Someren, "Using Content-based Filtering for Recommendation", *Proceedings of MLnet / ECML2000 Workshop*, 2000

[2] K. Iwahama, Y. Hijikata, S. Nishida, "Content-based Filtering System for Music Data", *Proceedings of the 2004 International Symposium on Applications and the Internet Workshops*, pp. 480 - 487, 2004

[3] B. Sarwar, G. Karypis, J. Konston and J. Riedl, “Item- Based Collaborative Filtering Recommendation Algorithms”, *Proceedings of the 10th International Conference on World Wide Web,* pp. 285 - 295, 2001

[4] M. Deshpande, G. Karypis, “Item-Based Top-N Recommendation Algorithms”, *ACM Transactions on Information Systems*, Volume 22, Issue 1 (January 2004), pp. 143 – 177, 2004.

[5] S. Ujjin, J. P. Bentley, “Particle swarm optimization recommender system”, *IEEE International conference on Evolutionary Computation*, pp. 124-131, 2003.

[6] G. Xue, C. Lin, Q. Yang, W. Xi, H. Zeng, Y. Yu and Z. Chen, “Scalable Collaborative Filtering Using Cluster-based Smoothing”, *International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR ’05*, pp. 114 – 121, 2005

[7] Y. Shih, D. Liu, “Hybrid Recommendation Approaches: Collaborative Filtering via Valuable Content Information”, *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, Track 8 - Volume 08, pp. 217-220, 2005

[8] M. Claypool, A. Gokhale, T. Miranda, “Combining Content-Based and Collaborative Filters in an Online Newspaper”, *ACM SIGIR Workshop on Recommender Systems*, August 19, 1999, USA.

[9] K. Ali and W. van Stam, “TiVo: Making show recommendations using a distributed collaborative filtering architecture,” in *ACM KDD ’04, pp. 394–401,* ACM, 2004.