

Personalized Suggestion for Music Based On Collaborative Filtering

¹Aniket Salunke *, ²Ruchika Kukreja, ³Jayesh Kharche, ⁴Amit Nerurkar

^{1,2,3,4}Vidyalankar Institute of Technology, Mumbai University, Mumbai, India

Abstract:

With the advancement of technology there are millions of songs available on the internet and this creates problem for a person to choose from this vast pool of songs. So, there should be some middleman who must do this task on behalf of user and present most relevant songs that perfectly fits the user's taste. This task is done by recommendation system. Music recommendation system predicts the user liking towards a particular song based on the listening history and profile. Most of the music recommendation system available today will give most recently played song or songs which have overall highest rating as suggestions to users but these suggestions are not personalized. The paper purposes how the recommendation systems can be used to give personalized suggestions to each and every user with the help of collaborative filtering which uses user similarity to give suggestions. The paper aims at implementing this idea and solving the cold start problem using content based filtering at the start.

Keywords: Collaborative Filtering, Coldstart, Recommendation System.

1. Introduction

Currently it has been estimated that there are 97 million songs in the world with each song being created every two minutes. Recommender systems aim to predict users' interests and recommend product items that quite likely are interesting for them. They are among the most powerful machine learning systems that online retailers implement in order to drive sales. Data required for recommender systems stems from explicit user ratings after watching a movie or listening to a song, from implicit search engine queries and purchase histories, or from other knowledge about the users/items themselves. Sites like Spotify, YouTube or Netflix use that data in order to suggest playlists, so-called Daily mixes, or to make video recommendations, respectively [10].

We intend to design an application to modify the way any music recommendation system provides suggestion of songs to user. In this we will be finding similarities between different users this can be done with the help of collaborative filtering. Our application will first build user profile by asking various questions to users and based on that will give content based recommendations which will

help us solve cold start problem or new user problem. There is a scale of 0-5 on which user can rate the song and based on this rating using similarity matrix suggestions which are personalized to every user are generated.

In the initial part of the paper, we will understand the process incorporated by leading music apps in order to give recommendations. We would also try to understand how their Recommender System works. In the later part of the paper, we would discuss more about our proposed system. We would present the steps that would be used by our system. And the details of concepts which would be used.

2. Background & Related Work

Any Recommender system uses either content or collaborative approach or mixture of both known as hybrid.

Collaborative filtering approaches build a model from a user's past behavior (items previously purchased or selected and/or numerical ratings given to those items) as well as similar decisions made by other users. This model is then used to predict items (or ratings for items) that the user may have an interest in. [7]

Content-based filtering approaches utilize a series of discrete, pre-tagged characteristics of an item in order to recommend additional items with similar properties [7].

Nowadays there are many applications which uses various recommendation system and provide personalized playlists to users like spotify, wynk ,pandora,etc.Some of them are discussed here.

Wynk is a music streaming app designed to give personalized playlists to millions of users worldwide. At Wynk, they have a real-time data engineering pipeline that captures user's behaviour throughout the app in real-time, be it playing tracks or skipping them, searching for specific songs/curated playlists, following artists, time of play etc. All of these events, along with the shared profile info is used in building a User Persona, which along with latent features from interactions with tracks becomes an input to their ML models[6].

Gaana uses collaborative and content based filtering systems. Gaana app analyzes the content of its entire application and recommends features based on the user's preferences or interests. If the app suggests the items and preferences which the user has already used in the past it leads to overspecialization. Collaborative filtering (CF) systems compute correlations between users; they predict song / album ratings for the current user, based on the song/ album ratings provided by other users, who have preferences for the songs which are highly correlated to the current user.[4]

Pandora utilizes a classification system that is the heart of their service. Pandora recommends by matching up the user's artist and song likes with other songs that are similar. The greatest challenge for Pandora is classifying songs in their database and building their musical taxonomy. To accomplish this, Pandora employs a team of trained musicians who perform a manual classification on each song before adding it to their database. The musicians spend their workdays listening to a collection of songs and tagging each according to approximately 400 musical attributes. Once songs are properly classified in the database Pandora compares the description of musical tastes of a station selected by an individual user with the classification of the songs in the music database. This comparison

returns a collection of songs that drive the playlist. The key to Pandora's method of recommending music from its database is to utilize an efficient and effective proximity measure algorithm to determine the neighborhood of music to play on a station [5].

3. Literature Survey

After studying the available system for music recommendation we aim to execute our proposed syetem,In order to do that we studied various papers based on the same to understand different concepts adopted in each one of them.

“Personalized Music Recommendation Algorithm Based On Hybrid Collaborative Filtering Technology [2]”

The paper proposes a hybrid model.It studies 800 pieces of songs by getting it from web crawler.It consists of two recommendation models.First is music gene model in which user preferences are calculated from users liking towards particular song and a list is generated.Secondly user similarity is also calculated and other list of recommended songs is generated.At the end finalized list of songs is generated taking into consideration both the models.

“Music Recommendation Based on Feature Similarity [8]”

In this paper similarity between songs is calculated.In order to do that some features are extracted from the song,this is done by calculating mel frequency cepstral coefficient(MFCC) feature quantity.Now this feature quantity can have vast range of values these values are then clustered to compress thje features.Now with the help of these features similarity is calculated between songs and similar songs are recommended to user.

“Music Recommendation Based on Content and Collaborative Approach & Reducing Cold Start Problem[9]”

In this paper spotify api is used to extract various features of the song like loudness,dancibility,tempo,liveness,energy, acousticness,instrumentalness .After these features are extracted various songs having same features are grouped together and user are recommended songs according to their taste.

4. Proposed System

The diagram below shows the overall flow of how the recommendation system is going to work.

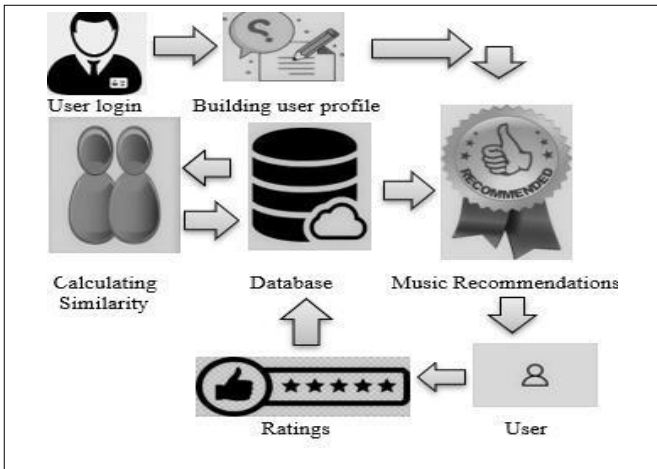


Fig 1: Work flow of the system

Step 1: First the user is supposed to register in our system if it is for the first time he is using our system.

Step 2: After successful Sign up user will be asked different questions like genre they are interested in and the artist they like in order to get the general idea of the user taste and build user profile.

Step 3: Then based on this choices of selection done by user he/she will be displayed songs.

Step 4: Now its users chance to rate individual songs..

Step 5: This rating of song for each user is stored in database and based on this rating similarity is calculated between user using cosine similarity and songs are recommended accordingly.

Step 6: User can now access the recommended song and rate it if he likes it and this process continues. Adding to this feature is the feature of searching the song using song name or movie name is also provided.

As each user is rating the song based on their liking this rating with corresponding user id and song id is stored in database forming n to n relationship .With the help of this data similarity is calculated between users using cosine similarity and this results in finding top number of similar user to a particular user and based on that songs are recommended .With each rating getting updated in the database similar users vary accordingly.And the list of recommended songs is displayed to user.

5. Overview of System

In our system we have used a subportion of kaggle’s Bollywood Song dataset .This dataset consists of song id,artist name,movie name,song name.This metadata about the song is used for displaying song details.Here we are going to see in detail how different concepts are used in our system for giving recommendations.

A. Content based filtering

At the start system doesn’t have any knowledge about user;s liking or what it prefers.So in our system we have explicitly asked user for preferences.So after logging in user is asked various questions like what genre they are interested in?,what artists they are interested in? etc.So this will help our system to create user profile.And based on this initial suggestions will be made for the user

B. Collaborative filtering

Collaborative filtering is a technique that can filter out items that a user might like on the basis of reactions by similar users.It works by searching a large group of people and finding a smaller set of users with tastes similar to a particular user. It looks at the items they like and combines them to create a ranked list of suggestions.There are many ways to decide which users are similar and combine their choices to create a list of recommendations. Here in our proposed system we are using cosine similarity to do that.

C. Matrix Formation:

Considering there are m number of users and n nuber of songs available in our system.Users are supposed to rate these songs.thus forming user-song matrix.Each row denotes rating given by particular user to various songs and each column denotes different ratings acquired by a particular song.So each element in a matrix is rating given by that particular user to that particular song.Given below is the user-song matrix:

	Song 1	Song 2	Song 3	Song n
				

User 1	4	4	3		5
User 2	1		3		3
User 3		3	2		
....					
.....					
User m	4	4	2	1

Fig.2: User-Song Matrix

Since, it is not possible that every time user listens a song he/she will rate it. If the user does not rate a particular song, this creates sparsity in the above matrix. As it can be seen that user 2 has not given any rating to song 2 and user 3 to song 1 and n. So this problem is also handled by cosine similarity.

D. Cosine Similarity

Cosine similarity is the measure of similarity between two vectors, by computing the cosine of the angle between two vectors projected into multidimensional space. It can be applied to items available on a dataset to compute similarity to one another via keywords or other metrics. Similarity between two vectors (A and B) is calculated by taking the dot product of the two vectors and dividing it by the magnitude value as shown in the equation below.

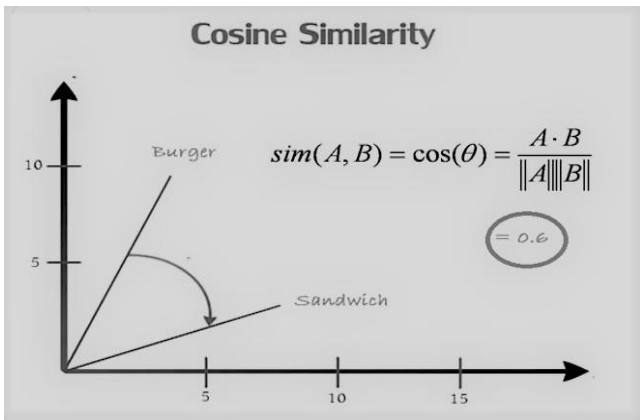


Fig.3: Cosine Similarity

Here the user-song matrix formed is used to calculate similarity between two users. This is done by plotting users in three dimensional space and then calculating cosine similarity using above formula. This results in formation of user-user matrix with values in each box between 0 and 1 as value of

cosine is always between 0 and 1. Following is the m*m user-user matrix:

	User 1	User 2	User m
			...	
User 1	1	0.8882	0.954
User 2	0.3526	1	0.325
User 3	0.852	0.5032	0.251
User 4	0.2565	0.936	0.054
...
...
..
User m	0.756	0.526	1

Fig.4. User-User Matrix

It can be seen from the above matrix that the value of similarity of user with itself is 1 since cosine(0) is 1 that means that these two vectors overlap each other. From this it can be concluded that greater the value of similarity between users less will be the angle between them when they are plotted in space and more will be the similarity.

In order to solve the sparsity issue what cosine similarity does before calculating similarity is that it replaces the blank values with the average of rating of all the songs that particular user has rated.

6. Conclusion

We have developed a progressive web application for music recommendation system with the help of hybrid filtering. Since we made a prototype of it for just 200 songs computation speed is very high. We have stored all these songs on firebase which makes the retrieval very easy. However when there are millions of songs available listened by millions of users the work of maintaining the computation speed and storing such massive amount of data is still remaining.

7. Future Scope

In our application we are building a prototype of the system using 200 songs out of the dataset to provide recommendations. Our proposed systems approach was limited due to computation factors. Since there are millions of songs available use of parallel systems with the help of apache Hadoop is an alternative. This can help compute bigger matrices and issue of storage will also be solved and recommendations out of millions of songs will be available to user within seconds.

8. References

- [1.] Wu Dan, "Music Personalized Recommendation System Based on Hybrid Filtration", 2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS).
- [2.] Wang Wenzhen, "Personalized Music Recommendation Algorithm Based On Hybrid Collaborative Filtering Technology", 2019 International Conference on Smart Grid and Electrical Automation (ICSGEA).
- [3.] Dong-Moon Kim, Kun-Su Kim, Kyo-hyun Park, Jee-Hyong Lee and Keon Myung Lee, "A Music Recommendation System with a Dynamic K-means Clustering Algorithm", Sixth International Conference on Machine Learning and Applications.
- [4.] Study of Recommender System - www.gaana.com ! My First Blog Available: <https://tejaswinidevappa.wordpress.com/2015/01/20/study-of-recommender-system-www-gaana-com/>
- [5.] Michael Howe, Pandora's Music Recommender
- [6.] MusicRecommendations@wynk-LinkedIn. Available: <https://www.linkedin.com/pulse/music-recommendations-wynk-prabhat-kumar-saraswat>
- [7.] Prem Melville and Vikas Sindhwani, Recommender System, Encyclopedia of Machine Learning, 2010[E-book]
- [8.] Huihui Han, Xin Luo, Tao Yang, Youqun Shi, "Music Recommendation Based on Feature Similarity", 2018 IEEE International Conference of Safety Produce Informatization (IICSPI).
- [9.] Parmar Darshna, "Music Recommendation Based on Content and Collaborative Approach & Reducing Cold Star", Proceedings of the Second International Conference on Inventive Systems and Control (ICISC 2018) IEEE Xplore Compliant - Part Number:CFP18J06-ART, ISBN:978-1-5386-0807-4; DVD Part Number:CFP18J06DVD, ISBN:978-1-5386-0806-7
- [10.] Tryolabs blog, "Introduction to recommender systems in 2019", Available: <https://tryolabs.com/blog/introduction-to-recommender-systems/>