IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Volume-3, Issue-10 E-ISSN: 2347-2693

Music Recommendation Based on Subjective Attributes

Priyank Jain[1*] and Vamsikrishna Patchava[2]

[1*]Department of Computer Engineering, Veermata Jijabai Technological Institute (VJTI), Mumbai, Maharashtra, India
[2]Department of Electronics & Communication Engineering, RGUKT-Nuzvid, Andhra Pradesh, India

www.ijcseonline.org

Received: Sep/28/2015 Revised: Oct/10/2015 Accepted: Oct /22/2015 Published: Oct /31/ 2015

Abstract— Majority of the music recommendation systems in use today use historical preferences of other users having similar taste to recommend songs to a particular user. Other systems use past preferences of the current user and musical attributes of songs to make recommendations. In this paper, we use a novel approach to recommend music to users based on content-based filtering. This system can be used both as a search engine and for making recommendations. Moreover, this system does not suffer from the cold start problem which most of the recommender systems suffer from. Our system has a very small learning curve. We present a simple yet fast approach to make music recommendations using echonest's music attributes. The system is based on calculating the Euclidean distance to find out top recommended songs. This system can be used in combination with traditional recommendation systems for more effective recommendation. We think music users will find this system easy to use and experiment with and therefore helpful to discover new music. This system will result in increased enjoyment of music for users.

Keywords-Music; Recommendation; Euclidean Distance; Machine Learning; Content-Based Filtering; Echonest; Danceability

I. INTRODUCTION

Music is the language of the universe. Music can create your mood and make you feel various emotions. It can put you into a trance, taking you away on an extended journey. It can make time feel frozen. If you are a music lover, you can remember a moment where a particular song made you particularly sad or very excited or happy. Most of us don't know why we like a particular kind of music. Moreover, most of us can't exactly describe the kind of music we like to hear. People hear music for the kind of emotion and feeling it evokes, which is a subjective feature.

However, most of music recommendations systems in today's times use either objective features combined with either content-based recommendation or collaborative filtering. Such systems not only take time to learn the user preferences, they primarily fail to capture the mood and emotion the users wanted to be evoked through the song. We present a system that applies content-based filtering to music using subjective features.

The rest of the paper is organized as follows: Section II explains how current music recommendation systems work. Section III goes into the details of the data source we used for our system. Section IV explains the working and algorithm used by our system. Section V discusses the results. Section VI discusses future work and concludes the paper.

II. CURRENT RECOMMENDATION SYSTEMS

Spotify [1] uses collaborative filtering for its music recommendation. The basic idea behind such a system is: First, if two tracks get similar ratings then they are probably similar. Second, if a lot of users all listen to a set of tracks, then those tracks are probably similar [2].

Such a system groups similar users into a group and suggest tracks to a user based on the historical preferences of other users in the group. The problem with this system is that a track must be liked by a substantial number of people before it can be recommended. New users and new tracks can cause the cold start problem, as there will not be enough data for collaborative filtering to work properly. The cold start problem is because the recommender system cannot draw inferences about users and tracks about which it has not yet gathered sufficient information.

Pandora [3] uses content-based recommendation [4]. Such recommendation is based on the musical attributes of a track. Contrary to spotify, this system takes into account the songs and the artists the user likes and accordingly makes recommendation to the same user. Similar to the previous approach, this approach too suffers from the cold start approach and requires quite a few likes and dislikes by the user on various tracks to understand that user's preferences. Moreover, such a system does not exactly suggest the music which the user wants to listen at a particular time. For example, let's say an user x clicks 'like' button for a track A which is loud, evokes a happy mood and is soft on the energy level (not so energetic). The user likes the fact that the track A is loud and has a happy mood. However, the user dislikes the fact that the track is not so energetic and would have preferred a song with a higher energy level. Such a system fails to take into account this preference of the user. Consequently, the system will keep on recommending loud, happy and less energetic songs, which in turn will result into decreased confidence of the user in the recommendation system.

We present a recommendation system which directly ask the user for his/her preferences about subjective attributes of music. We then present a list of top 100 recommended songs based on the subjective features the user mentioned.

III. DATA SOURCE

We use a subset of the available songs in the Million Song Dataset [5]. Our database consists of 5 thousand songs. We get features for the songs from Echonest [6]. Echonest is a music intelligence and data platform for developers and media companies. It provides various objective and subjective features about a track. Table 1 lists the features we use in our system, along with their description, value range and how the features are interpreted by the recommendation system [7].

These features have been calculated with data from track analysis and from inputs of EchoNest's Data QA Team, many of whom are qualified musicians [8]. We gathered these attributes for all the songs using python and stored the results in a tab seperated file.

TABLE I. LIST OF FEATURES USED IN OUR SYSTEM, THEIR DESCRIPTIONS, RANGE AND INTERPRETATION

No.	Feature	Description	Range	Interpretation
1	Loudness	The overall loudness of a track in decibels (dB)	-60 to 0	Extremely soft: -60 Soft: -45 Moderate: -30 Loud: -15 Extremely loud:0
2	Speechiness	Detects the presence of spoken words in a track. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.	0 to 1	Spoken song: 1 Both speech and music: 0.5 Mostly music: 0
3	Liveness	Detects the presence of an audience in the recording. A value above 0.8 provides strong likelihood that the track is live. Values between 0.6 and 0.8 describe tracks that may or may not be live or contain simulated audience sounds at the beginning or end. Values below 0.6 most likely represent studio recordings.	0 to 1	Live song: 0.9 Partially live: 0.7 Studio recordings: 0.3
4	Acousticness	Represents the likelihood a recording was created by solely acoustic means such as voice and acoustic instruments as opposed to electronically such as with synthesized, amplified, or effected instruments	0 to 1	Highly acoustic: 1 Hardly acoustic: 0
5	Valence	A measure of how much positivity the song conveys	0 to 1	Positive: 1 Neutral: 0.5 Negative: 0
6	Danceability	Represents how suitable a track is for dancing	0 to 1	Highly danceable: 1 Not danceable: 0
7	Energy	Represents how intense and powerful the track is	0 to 1	High Energy: 1 Moderate Energy: 0.5 Low energy: 0

IV. ALGORITHM AND WORKING

Our system uses content-based filtering since we use the musical attributes to recommend songs. However, our system differs from traditional content-based filtering systems that we do not guess the feature vector for a user based on the songs he/she has listened to in the past, but rather ask the user for the feature vector in an interactive way. The feature vector represents the preferences of the user for each of the musical attributes.

Once we have constructed the feature vector for the user using interactive user input, we calculate the euclidean distance of this vector from the vector of the musical attributes of each song in the database. Then top 100 songs are chosen which have the minimum value of the euclidean distance. These are the top 100 songs which the user should like listening to based on the subjective attribute preference he/she gave as input. For example, let (1) and (2) represent the feature vectors for two songs, where the seven elements correspond to the values for the seven musical attributes listed in table 1. The euclidean distance between these two songs can be calculated as shown in (3). This can be viewed as a customized version of k- nearest neighbour algorithm, with entire dataset being the training set and k being 100.

$$p = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \end{bmatrix} \quad (1)$$

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \end{bmatrix} \quad (2)$$

$$\text{Euclidean Distance} = \sqrt[2]{\sum_1^7 (p_i - q_i)^2} \quad (3)$$

V. Results

Fig. 1 shows the interactive input screen which captures the user preferences. The first input is the energy level, which can be one of “high energy”, “moderate energy” and “not so energetic”. The second input is the liveness of the song, which can be one of “live”, “partially live” and “studio recording”. In the same manner, one has to provide input for speechiness, accousticness, danceability, loudness and mood (valence) describing the song they want to hear. These inputs are then internally mapped to attributes of the songs as shown in Table 1. Code for the same is written in Python programming language. Then, the top 100 songs having smallest Euclidean distance from the vector of attributes corresponding to the input are displayed to the user as recommended songs as shown in Fig. 2. The top 100 songs are shown in groups of five to keep a clean user interface.

From the user point of view, the number of inputs to be taken is not much. Moreover, the advantage is that the system allows completely opposite types of music to be searched by the user in subsequent searches. This is not possible with traditional recommendation systems, which will not show a soft track to a user who prefers extremely loud music. To add to this, the interactive input does not require the user to have any special music knowledge, so it can easily be used by the general public.

Currently the algorithm scans through the entire set of songs in the database to select the top recommended songs. Since our database contains only 5000 songs for now, the process is very fast still. However, if the number of songs reaches millions or more, this process can become a possible bottleneck. One way to overcome this would be to use music clustering based on these attributes and then recommend songs from the best matching clusters.

Another advantage of the current system is that it does not require the user to have listened to a few songs or the user to have liked/disliked a few songs before it can make a recommendation. The system can make recommendation for a new user without a learning curve. This eliminates the cold start problem that comes with traditional recommendation systems.

VI. Future Work and Conclusion

Clustering can be used to make quick recommendations when the total number of songs in the database is very large.

This system can be used in user friendly devices like smartphones or raspberry pi for making a music system that recommends/plays music based on the sentiment of the user or the audience in the room.

This approach can easily be applied by online music streaming websites like spotify.com, pandora.com, gaana.com, saavn.com and so on. For best recommendation, this system can be used in combination with current content-based filtering or collaborative filtering systems, where the traditional systems can be used for users who have been on the website for quite some time and have listened to plenty of songs, and our system can be used for new users or for experienced users who want to try a different flavor of music.

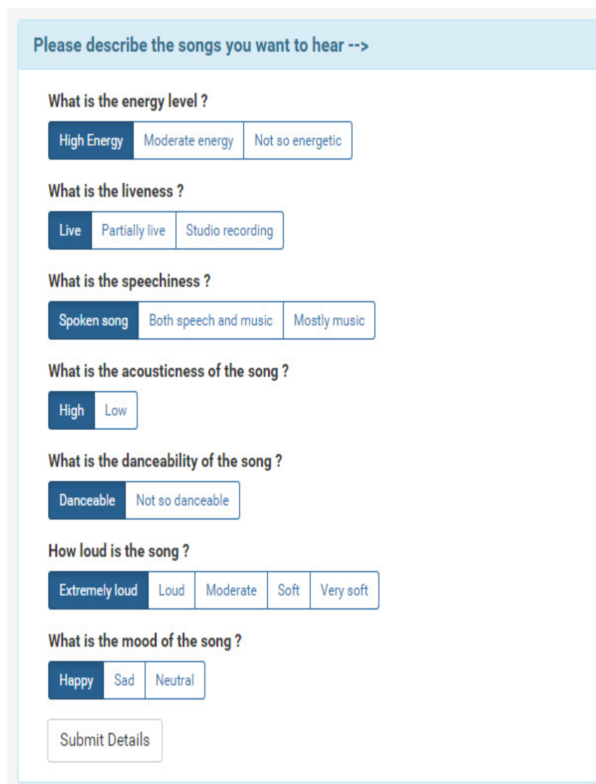

Figure 1. Interactive input screen

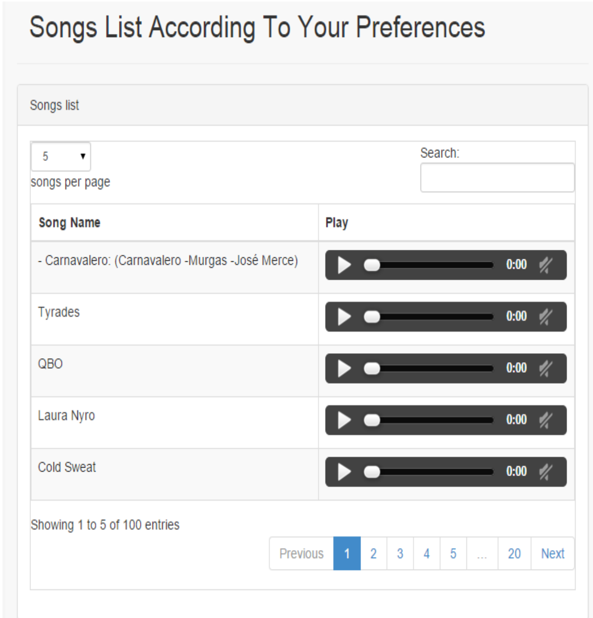

Figure 2. List of Recommended Songs

REFERENCES

[1] Spotify, music for everyone. https://www.spotify.com, Aug 2015

[2] Collaborative Filtering at Spotify, slide 4 of 63. http://www.slideshare.net/erikbern/collaborative-filtering-at-spotify-16182818, Aug 2015

[3] Pandora Internet Radio, listen to free music you'll love. http://www.pandora.com, Aug 2015

[4] Michael Howe. "Pandora's Music Recommender". http://courses.cs.washington.edu/courses/csep521/07wi/prj/michael.pdf

[5] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The Million Song Dataset. In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), 2011.

[6] Echonest. http://developer.echonest.com/docs/v4, Sep 2015

[7] Acoustic Attributes, The Echo Nest Developer Center. http://developer.echonest.com/acoustic-attributes.html, Sep 2015

[8] Danceability and Energy: Introducing Echo Nest. http://runningwithdata.com/post/1321504427/danceability-and-energy, Sep 2015

AUTHOR PROFILES

Priyank Jain is a B. Tech. in Computer Engineering from Veermata Jijabai Technological Institute, Mumbai, India in 2014. He is a software developer with keen interest in Machine Learning and Natural Language Processing.

Vamsikrishna Patchava was born in Tangutur, India, in 1994. He completed the B.Tech. Degree in Electronics and Communication Engineering from Rajiv GandhiUniversity of Knowledge Technologies, Nuzvid, India, in 2015. His current research interests include Embedded Systems and Bio Electronics. He has done many research projects using Raspberry Pi during his academic career in Rajiv Gandhi University of Knowledge Technologies, Nuzvid. He is continuing his research in Bio medical engineering and Embedded System by developing new innovative solutions for real time problems.