International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 06 | June 2019 www.irjet.net p-ISSN: 2395-0072

A Personalized Music Recommendation System

Shravanthi K[1], Sneha Rao S[2], Swati Joshi[3], Vandana R[4]

[1,2,3,4] *Students of Bangalore Institute of Technology*

[5] *Guided by Prof. Maya. B. S, Dept. of Computer Science and Engineering, V. V. Puram, Bangalore, Karnataka, India*

Abstract - *With growing number and variety of songs, choosing a few favorite songs according to one's taste has become a challenging task. Therefore we intend to develop a music recommendation system to recommend songs to the user based on some previous data. This system recommends the most popular songs among all songs. It also gives personalized recommendation based on similar songs and similar genre from the user's history. The system is based on convolutional neural network which classifies songs into different genre. The system also uses collaborative filtering algorithm to match songs from user history to the similar songs from all the songs. The front-end consists of a simple demonstration of the recommendation system with log-in for existing users and sign-up for new users.*

Key Words: *Collaborative filtering, CNN and music recommendation*

1. INTRODUCTION

Personalized music recommendation is a challenging task in the field of music information retrieval (MIR) [1]. A personalized music recommendation algorithm recommends new music to a user is similar to the previously music listened by the user. In order to recommend a song to a user, first step is to classify the music according to the user's history [2]. Traditional classifiers such as the support vector machine [3] and linear regression [4] classify the music by extracting the mel-frequency cepstral coefficients (MFCC) from the audio signal of the music [2]. As the structural complexity of the music is more, the efficiency of traditional classifiers reduces in classifying the music from different genres [5].

To solve the above research issues, researchers use deep neural networks (DNN) approaches for music classification [5], [6]. The DNN approaches have shown efficient results in the tasks related to the pattern recognition (e.g., image processing, video processing etc.) [7]. These approaches have the capability to extract the classified information presented in the data. DNN approaches such as the convolutional neural networks (CNN) [5], the gated recurrent unit [8], and the long short-term memory (LSTM) [9] can work on large data in a distributed manner. The classification efficiency of these DNN approaches are better than the traditional classifiers such as the support vector machine (SVM) [3] and the linear regression [4].

1.1 Convolutional Neural Network (CNN)

Convolutional networks were inspired from the biological processes of the human or an animal brain. It consists of one input layer, many hidden layers and one output layer. Each layer consists of one or more neurons based on the application just like in a brain. The input data given at the input layer is modified at each layer and passed onto other layers and the final output is produced. It is used for many applications like classification, clustering etc. This approach can be used with visual and audio data.

1.2 Collaborative Filtering (CF)

User-based collaborative filtering is based on implicit observations of normal user behavior (as opposed to the artificial behavior imposed by a rating task). These systems observe what a user has done together with what all users have done (what music they have listened to, what items they have bought) and use that data to predict the user's behavior in the future, or to predict how a user might like to behave given the chance.

Item-based collaborative filtering calculates the similarity between items calculated using people's ratings of those items. Item-item models use rating distributions per item, not per user. With more users than items, each item tends to have more ratings than each user, so an item's average rating usually doesn't change quickly. This leads to more stable rating distributions in the model, so the model doesn't have to be rebuilt as often. When users consume and then rate an item, that item's similar items are picked from the existing system model and added to the user's recommendations. First, the system executes a model-building stage by finding the similarity between all pairs of items. This similarity function can take many forms, such as correlation between ratings or cosine of those rating vectors. Second, the system executes a recommendation stage. It uses the most similar items to a user's already-rated items to generate a list of recommendations. Usually this calculation is a weighted sum or linear regression. Item-based CF had lesser errors compared to user-based CF method.

2. THE FRAMEWORK

2.1 Personalized Music Recommendation System (PMRS) System Architecture

In this section, we discuss the system architecture of the PMRS. The PMRS architecture is as shown in Fig. 1. The log file which has the user history is sent to CF algorithm as input. The CF algorithm gives two kinds of recommendations based on popularity and similar songs. The new music consists of audio signals from which features are extracted and given to CNN as input. CNN classifies the audio signals into genres and gives the genre as output for the corresponding audio signal. Using this, the genre-based recommendations are given.

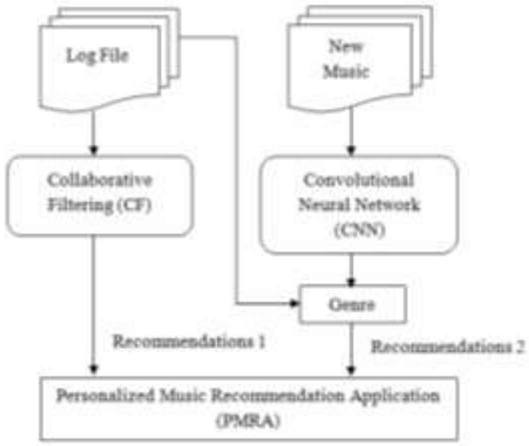

Fig -1: The system architecture of the PMRS

2.2. Collaborative filtering (CF) algorithm

A. Popularity-based CF

Input is the log file and output is the most popular song recommendation. It takes the count of user-ids (as score) for each unique song. Sorts it based on score (descending). Ranks it as higher score assigned to lower rank. Takes the top ten ranks and gives them as recommendations to the user.

B. Item-based CF

Input is the Log File and output are the recommendations based on similar songs in the user history. We create a pivot table with SongID and UserID as a matrix with listen count as entry in the matrix. The pivot table is converted into count matrix using cosine similarity function. The count matrix consists of similarity between all the songs as values in the matrix. The songs that have the highest similarity values for a particular song X is taken from the count matrix and recommended to the user.

Table -1: Sample pivot table

	User U1	User U2	User U3
Song S1	1	3	4
Song S2	2	1	2

The values in the matrix are the listen count i.e., the number of times the user has listened to a particular song. The assumption here is that the user has listened to a song at least once. Therefore the minimum listen count in the pivot table will be 1.

Cosine Similarity Formula:

$$\text{Cos}\theta = \frac{\sum_1^n a_i b_i}{\sqrt{(\sum_1^n a_i^2)(\sum_1^n b_i^2)}}$$

The equation specifies the product of entries of songs S1 and S2 from the table for n = 3 users. Therefore:

Cosɵ = ((1 * 2) + (3 * 1) + (4 * 2)) / (sqrt(1^2 + 3^2 + 4^2) * sqrt(2^2 + 1^2 + 2^2))

Cosɵ = 0.85

Cosɵ is the angle between the two songs. Higher the angle less similar they are and lesser the angle more similar the two songs are. Therefore higher similarities between songs are given as recommendations to the user.

Table -2: Count matrix derived from the pivot table using cosine formula

	S1	S2
S1	0	0.15
S2	0.15	0

The diagonal entries (S1, S1) and (S2, S2) is made zero so that the same song will not be recommended to the user.

2.3. Convolutional Neural Networks (CNN) approach

Inputs to the CNN are the audio signals and output is the probability of audio signal belonging to each genre. We extract the features from audio signals using librosa library. The extracted features are given as input to the network. Network has 5 layers: 1 input, 3 hidden and 1 output layers. Activation function used in each hidden layers is rectified Linear Unit (ReLU). The output function used in the output layer to provide the probabilities is the Softmax function. The genre-based recommendations are given using the output of CNN and log file. If user U1 has listened to songs S1, S2, S3. We find the genres of S1, S2, S3 as G1, G2, G3. Identify songs set S from output of CNN having genre G1, G2 and G3. Recommend songs from S to the user.

3. RESULTS

3.1 Dataset

We use the publicly available million song dataset (MSD) [10] from which the log file is taken. The log file consists of one lakh twenty four thousand rows one for each user and song details. The audio signal dataset consists of 1000 signals.

3.2 Music Applications

The front-end music application is only a demonstration of the working of the backend. It consists of user log-in and sign-up. When an existing user logs in he gets a list of his previously listened songs and also three kinds of recommendations – popularity-based, item-based and genre-based. When a new user signs up he gets only the popularity-based recommendation.

Fig -3: Webpage when existing user logs in.

Fig -4: Webpage when new user logs in.

3.3. Performance evaluation

We use Python programming language to implement the PMRS. The TensorFlow library is used to implement the CNN approach for music classification. A thirty second segment of the song that best represents the song is taken for a new song. This segment is converted into monoaural signal from general stereo signal for easy extraction of features. For this converted signal, the audio feature are extracted and sent to the CNN algorithm for genre classification. The CNN approach uses the rectified linear units (ReLUs) with MaxPool (i.e., max(0,x)) as the activation function.

In order to evaluate the performance of the PMRS, we selected the audio signals of the top 1000 music for top 10 genres. The dataset is divided as 800 train data and 200 test data. The 800 train data is further divided as 600 train data and 200 validation data. We use the training data to train the PMRS. After training the PMRS, we use the testing data to evaluate the performance of the system.

4. CONCLUSION

We have presented a personalized music recommendation system based on the CNN approach and collaborative filtering algorithm. CNN approach was used to classify music based on corresponding audio signals of the music into genre and gives genre-based recommendation. The CF algorithm uses log file to provide recommendation to users by calculating item-based similarity. As a part of our future work, we would like to on work on the efficiency of the genre classification. We would also like to extract the user's information (like geographical location, time, emotions, etc.) to provide better music recommendation that match the user's preference. The usage of the latest song can also be considered in the dataset to keep the system updated.

ACKNOWLEDGEMENT

A special thanks to my Guide Prof. Maya. B. S who supported us in completion of the project and paper.

We also thank our HOD Dr. Asha T for her motivation and support.

Lastly we would also like to thank all the faculty members and non-teaching staff member of the Department of Computer Science and Engineering, Bangalore Institute of Technology.

REFERENCES

[1] Z. Hyung, J. S. Park, and K. Lee, "Utilizing context-relevant keywords extracted from a large collection of user-generated documents for music discovery," *Info. Processing and Management*, vol. 53, no. 5, pp. 1185-1200, 2017.
[2] E. J. Humphrey, J. P. Bello, and Y. LeCun, "Moving beyond feature design: deep architectures and automatic feature learning in music informatics," in *Proc. 13th Int'l Conf. Music Info. Retrieval*, pp. 403-408, October 2012.
[3] C. W. Hsu, and C. J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Tran. Neural Networks*, vol. 13, no. 2, pp. 415-425, 2002.

[4] R. E. Fan, K. W. Chang, C. J. Hsieh, X. R. Wang, and C. J. Lin, "LIBLINEAR: a library for large linear classification," *J. Machine Learning Research*, vol. 9, pp. 1871-1874, 2008.

[5] A. V. Oord, S. Dieleman, and B. Schrauwen, "Deep content- based music recommendation," in *Proc. 26th Int'l Conf. Neural Info. Process. Systems*, pp. 2643-2651, December 2013.

[6] J. Salamon, and J. P. Bello, "Deep convolutional neural networks and data augmentation for environmental sound classification," IEEE Sig. Processing Letters, vol. 24, no. 3, pp. 279-283, March 2017.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Info. Processing Sys*, pp. 1097-1105, 2012.

[8] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *Int'l Conf. Machine Learning*, pp. 2342-2350, June 2015.

[9] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *15th Annual Conf. Int'l Speech Comm. Assoc.*, pp. 338-342, September 2014.

[10] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," *ISMIR*, vol. 2, no. 9, p. 10, October 2011.

[11] S. Chang, A. Abdul, J. Chen and H. Liao, "A personalized music recommendation system using convolutional neural networks approach," *2018 IEEE International Conference on Applied System Invention (ICASI),* Chiba, pp. 47-49, 2018.