A Personalized Music Recommendation System Using Convolutional Neural Networks Approach

Shun-Hao Chang[a], Ashu Abdul[a], Jenhui Chen[a,b,c,*], Hua-Yuan Liao[a]

[a,*] Department of Computer Science and Information Engineering, College of Engineering, Chang Gung University, Kweishan, Taoyuan 33302, Taiwan.
[b] Department of Otorhinolaryngology, Head and Neck Surgery, Chang Gung Memorial Hospital, Kweishan, Taoyuan 33375, Taiwan.
[c] Department of Electronic Engineering, Ming Chi University of Technology, Taishan Dist., New Taipei City 24301, Taiwan.
Email: ashu.a507@gmail.com; abide0222@gmail.com; jhchen@mail.cgu.edu.tw; nmxzqw@gmail.com

Abstract

In this paper, we present a personalized music recommendation system (PMRS) based on the convolutional neural networks (CNN) approach. The CNN approach classifies music based on the audio signal beats of the music into different genres. In PMRS, we propose a collaborative filtering (CF) recommendation algorithm to combine the output of the CNN with the log files to recommend music to the user. The log file contains the history of all users who use the PMRS. The PMRS extracts the user's history from the log file and recommends music under each genre. We use the million song dataset (MSD) to evaluate the PMRS. To show the working of the PMRS, we developed a mobile application (an Android version). We used the confidence score metrics for different music genre to check the performance of the PMRS.

Key words: Collaborative filtering, CNN and music recommendation

I. 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. Researchers in the field of MIR generally use two recommendation algorithms: collaborative filtering (CF) [2] and content based (CB) [3]. The CF algorithm recommends new music which is not available in the active user's history and which is available in the other users with similar history as the active user [2]. The CB algorithm recommends music to a user based on the similar music available in the user's history [3].

In order to recommend a music to a user, first step is to classify the music according to the user's history [4]. Traditional classifiers such as the support vector machine [5] and linear regression [6] classify the music by extracting the mel-frequency cepstral coefficients (MFCC) from the audio signal of the music [4]. As the structural complexity of the music is more, the efficiency of traditional classifiers reduces in classifying the music from different genres [7].

To solve the above research issues, researchers use deep neural networks (DNN) approaches for music classification [7], [8]. The DNN approaches have shown efficient results in the tasks related to the pattern recognition (e.g., image processing, video processing etc.) [9]. These approaches have the capability to extract the classified information presented in the data. DNN approaches such as the convolutional neural networks (CNN) [7], the gated recurrent unit [10], and the long short-term memory (LSTM) [11] 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) [5] and the linear regression [6]. In [7], authors use CB recommendation algorithm with the CNN approach for recommending music to user. The CNN approach classify the music based on the audio signal presented in the music. In [8], authors use the CNN approach for classifying environmental sounds. The existing music recommendation [7], [8], are limited to recommend music based on the audio signal of the music.

In this paper, we investigate a personalized music recommendation system (PMRS) which is based on the CNN approach and a CF recommendation algorithm. The CNN approach classifies the music based on the audio signal presented in the music. The CNN extracts the hidden features presented in audio signals the music and classifies the music accordingly. In PMRS, we propose a CF algorithm to extract the user's history presented in the log file and the output of CNN for recommending music under different music genres. We developed an android music application, to demonstrate the working of the PMRS algorithm. Publically available the million song dataset (MSD) [12] is used to train the system.

II. Personalized Music Recommendation System (PMRS)

A. *PMRS system architecture*

In this section, we discuss the system architecture of the PMRS. The PMRS is a distributed architecture as shown in Fig. 1. In PMRS, we achieve the distributed functionality by using Apache Spark [13] and MongoDB [14] for distributed storage and processing of the data. Apache Nutch crawler crawls music data from the online music websites. The input to the CNN approach is the crawled music data. The output of the CNN is stored as classified-music in the database. The log file gives the history of all the users who use the PMRS. The CF algorithm uses the classified-music data and the log file data to recommend music to the user.

B. *Collaborative filtering (CF) algorithm*

The CF algorithm, extracts the user's history H_U, from the log file. The H_U contains the ratings R_M given by user U for music M. Here $M = \{M_1, M_2, M_3, ..., M_S\}$ is a set of music where

ISBN 978-1-5386-4342-6

|M|=S. The H_U is a vector of R_M provided by the user U towards music M shown as

$$H_U = [R_M], \ \forall M = \{M_1, M_2, M_3, \ldots, M_S\}. \quad (1)$$

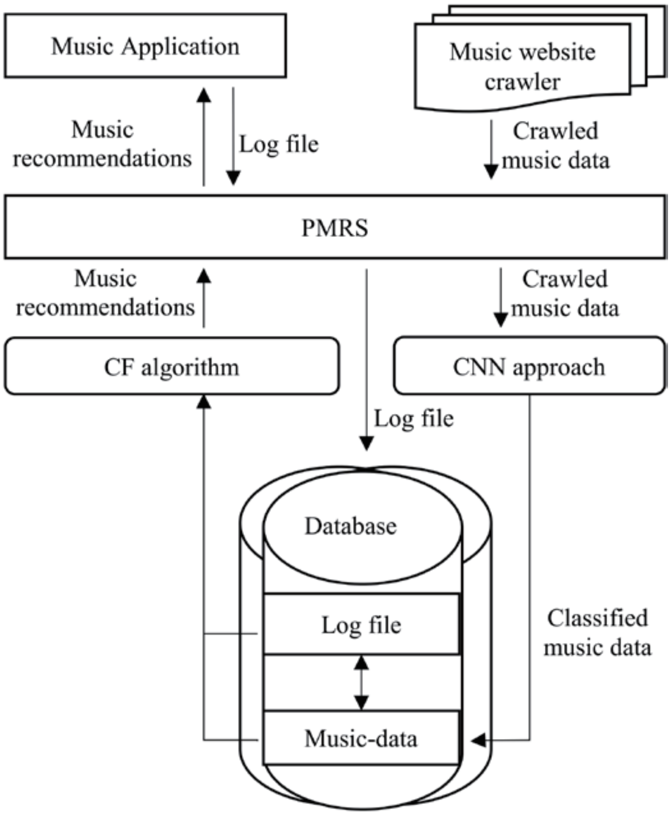


Fig. 1 The system architecture of the PMRS

Let U_M be the music vector selected from the H_U whose rating is greater than a threshold value T_R and represented as

$$U_M = [R_M] \forall R_M \in H_U \ \ and \ \ R_M > T_R. \quad (2)$$

Let C_M be the music classification obtained from the CNN approach. The vector $V_{U,M}$ containing the cross product of the U_M and the C_M is calculated as

$$V_{U,M} = [U_M \otimes C_M]. \quad (3)$$

The PMRS selects the top N music from the $V_{U,M}$ which are not presented in the H_U and recommends to the user U. Let *TopN* denote the set of top N music represented as

$$Top_N = \{M_1, M_2, \ldots, M_N\}$$
$$\forall M_X \in V_{U,M} \ and \ M_X \notin H_U. \quad (4)$$

C. Convolutional Neural Networks (CNN) approach

The CNN approach collects the crawled music data from Internet for classifying the music. In order to classify the music, firstly we should represent the audio signals in the form of the MFCCs. These MFCCs denote the structural and hierarchical features of an audio signal. In order to obtain the MFCCs, we sampled the audio signal of the music into a set of small audio clips of 4 second. Later we convert these audio signals into log-compressed mel-spectrogram with 128 components [8.8] to generate the MFCCs. Based on these MFCCs the CNN classifies the music. Let C_M be set classified music classification of music set M containing F features F= {1, 2, 3,…, F}. The C_M is a vector containing the music and the music features and is represented as

$$C_M = [M_F],$$
$$\forall M = \{M_1, M_2, M_3, \ldots, M_S\}, F = \{1, 2, 3, \ldots, F\}. \quad (5)$$

III. Results

A. Dataset

We use the publicly available million song dataset (MSD) [12] to evaluate the performance of the PMRS. We use the distributed architecture shown in Fig. 1 to deploy the MSD. The MSD is a dataset of size 280 GB containing the audio signal of one million music tracks.

B. Music Application

To demonstrate the working of the PMRS we developed a mobile application (an Android version). This mobile application acts as a user interface to collect the user's history in the log file. Fig. 2(a) shows the login page of the mobile application. Fig. 2(b) shows the top 5 music recommendations along with the music genre.

C. Performance evaluation

We use Python programming language to implement the PMRS. The TensorFlow library [15] is used to implement the CNN approach for music classification. We used the system configuration settings provided in [7] for implementing the CNN approach. The audio signal of the music is converted in to mel-spectrogram of 599 frames with 128 frequency bins. The CNN approach uses the rectified linear units (ReLUs) with MaxPool (i.e., max(0,x)) as the activation function. In case of the CF algorithm, we used the threshold value T_R = 1 to implement the equation (2). We do not consider the music M for a user U with a rating R_M = 0. In PMRS, we assume that a music M with ratings R_M = 0, signifies that the user U is unaware of that music. For this reason, we consider the ratings R_M >= 1 given by user U with a threshold value T_R = 1 to create the U_M music vector as shown in equation (2).

In order to evaluate the performance of the PMRS, we selected the audio signals of the top 4000 music for top 7 genres rated by top 50 users from the MSD. The top 7 genres which we selected are: Breakbeat, Dancehall, Drum and bass, Funky house, Hip Hop, Rock, and Trance. We divide the audio signals of the 4000 music into two parts: the training data with 3000 audio signals and the testing data with 1000 audio signals. 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 in terms of confidence score (CS) value as

$$CS_{PMRS} = \sum \begin{pmatrix} \text{Top N correct music recommedations} \\ \text{by the PMRS to the user} \end{pmatrix}$$
$$CS_{MSD} = \sum \left(\text{Top N music rated by the user in the MSD}\right)$$
$$CS = \frac{CS_{PMRS}}{CS_{MSD}}, \ \text{for all music genres.} \quad (6)$$

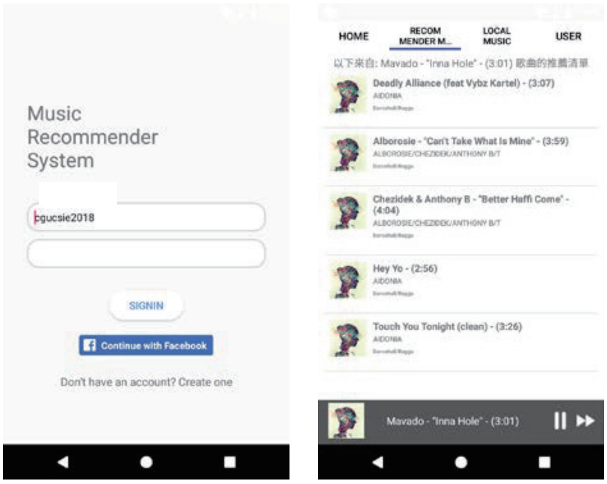
The confidence score (CS) value is calculated with respect to the fitness of music predictions for each user and each genre available in the testing data. We calculated the average of the CS values obtained from the testing data. Fig. 3 shows the mean CS values of the music recommendations obtained with respect to original MSD. From Fig. 3 we can analyze that the confidence score for Breakbeat and Funky house is less than 90% of the original MSD value. In other *5* genres the CS value

ISBN 978-1-5386-4342-6

is greater than 90% of the original MSD. We think that the structural and hierarchical features presented in the audio signals of these music genres as the cause for the differences in music recommendations.

IV. Conclusion

In this paper, we have presented a personalized music recommendation system based on the CNN approach and collaborative filtering algorithm. We used the CNN approach to classify music based on the corresponding audio signals of the music. The CF algorithm uses the classified music data and the log file to provide music recommendation to users. As part of our future work, we would like to investigate the usage of meta-data of the music along with the audio signal of the music to classifying the music. We would like to extract the user's information (like geographical location, time, ambience, emotions, etc.) to provide a better music recommendation that match with the user's preference. Moreover, the usage of other DNN approaches such as the GRU and the LSTM for performing music classification with respect to the CNN approach can be consider to enhance the precision of user's music preference prediction.



(a) (b)

Fig. 2 (a) Login page of the PMRS. (b) Top 5 music recommendations to the user by the PMRS

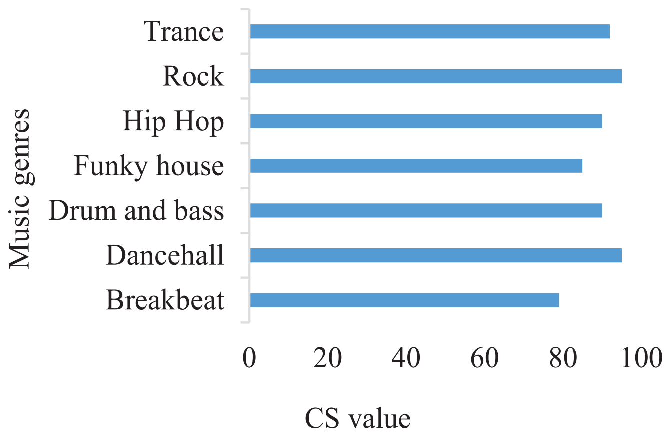


Fig. 3 The CS value obtained by the PMRS for music recommendations in different genres.

Acknowledgement

This work was supported in part by the Ministry of Science and Technology, Taiwan, R.O.C., under Contract MOST 106-2221-E-182-065.

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] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," *The Adaptive Web: Methods and Strategies of Web Personalization*, pp. 291-324, 2007.

[3] M. J. Pazzani and D. Billsus, "Content based recommendation systems," *The Adaptive Web: Methods and Strategies of Web Personalization*, pp. 325-341, 2007.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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

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

[13] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. B.Tsai, M. Amde, S. Owen, and D. Xin, "Mllib: machine learning in apache spark," *J. of Machine Learning Research*, vol. 17, no. 1, pp.1235-1241, 2016

[14] K. Chodorow, "MongoDB: the definitive guide: powerful and scalable data storage," *O'Reilly Media, Inc.*, 2013.

[15] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, and M. Kudlur, "TensorFlow: a system for large-scale machine learning," *OSDI*, vol. 16, pp. 265-283, 2016.