**A music recommend system based on**

**deep neural network**

**1.ABSTRACT**

Music recommendation has become a rapid booming field in recent years. However, the widely used traditional collaborative filtering suffers from the cold-start problem that the usage data is not always available, so that new issued songs and unpopular songs are unlikely been recommended. In the other hand, the deep learning has achieved great success in many fields, especially in computer vision and speech recognition. It, however, rarely used in music recommendation system. Our project is trying to propose a new hybrid music recommendation system based on deep learning.

The data of the project is based on Kaggle’s Million Songs Dataset Challenge [1], which provides with 386213 songs. According to the dataset, the universal recommendation scoring system is defined. Then, We are trying to use convolutional automatic encoder to process MFCC and Chroma audio features, while lyrics features are obtained by word embedding method. Finally, we would develop our hybrid recommendation system and evaluate the quality of recommendation. The hybrid system combines the content features and collaborative filtering.

**2.Motivations**

**2.1 Background**

With the growth of serious problem of information overload, the recommendation system came into being. Recommendation system is actually a kind of information filtering system, which aims to predict user’s preference for articles, commercials, products and so on. Now, recommendation system is widely used in industry. For instance, 80% of the clicks on online video Netflix come from recommendations, while around 60% clicks come from recommendation in Youtube. In Internet age, recommendation systems bring with great convenience to acquire personal preference for users.

The project focus on the application of recommendation in music field since music industry is turning to online music service, such as iTunes, Spotify, Google Play, Netease Music, etc. As users soaring, the online music services are paying more and more attention to music recommendation algorithm, which make it much easier for users to retrieve the music that fit their tastes.

**2.2 Issue**

The music recommendation system could automatically match the user’s preference, and recommend music that fit their tastes. The quality of recommendation is influenced by many factors, such as age, mood, lyrics, singers, and so on. Besides, the audio content itself is vital. Whether we like a song is because of the content of the music, including voice, melody, rhythm, tone, and instruments, etc. The content of music largely determines whether we like it or not. So we believe the content-based recommendation would have good performance. However, the traditional music recommendation system is based on collaborative filtering, which fails when usage data is not available.

The project is committed to develop music recommendation system based on deep learning. The research objectives are list as followings:

1. Exploring the advantages of collaborative filtering using deep learning versus traditional collaborative filtering.

2. Applying convolutional neural network to learn audio features.

3. To build hybrid music recommendation system.

**3 Related Topics**

Firstly, we can use Hadoop or Spark to process data. We have studied Hadoop in week 2. Besides, we will use recommendation systems to recommend music to users. Recommendation systems are mainly used to estimate the users' preference for their unknown pair of items. We will study Recommender Systems and Matrix Factorization in Week 6. The recommendation system is divided into two different groups: Collaborative filtering systems and Content-based systems [2]. Collaborative filtering systems collect the users' preferences or taste information to automatically predict the users' interests to achieve the purpose of recommending items. Content-based systems analyze the attributes of some products that the users have been paying attention to, and then recommend related things to the users.

**4 Plan to Submit**

To better evaluate the recommendation model, we will use Kaggle's Million Song Dataset Challenge [1], which provided an additional test set. Because the number of times users listen to the song is different, in order to eliminate the impact of this data between the users, we will normalize the data.

After that, we will use Mean Average Precision (MAP) to evaluate our project because Kaggle used it as an evaluation standard in this competition. The MAP is given by:

where represents the average accuracy of the music recommended to the i-th user for the Top 500, and I represents the number of all users. We will first count the MAP of the top teams in the Kaggle competition, then we compare our MAP results with these results. We will be able to submit relevant line graphs to indicate the impact on the performance of the recommendation system when adjusting different parameters or calculation methods of our algorithm.

In the meantime, for better comparison, we will use part of the train set as validation set if possible. We can calculate Root Mean Square Error (RMSE) to evaluate the performance of our recommendation system. RMSE is a commonly used indicator of the recommendation systems to indicate the performance of the prediction. The RMSE is given by:

where represents all data in the observation dataset, and represents data predicted by the recommendation system, and n represents the number of observations.

**5. Dataset**

We use **Million Song Dataset** as our project dataset [3]. The Million Song Dataset is a freely available collection of audio features and metadata for a million contemporary popular music tracks. The core of the dataset is the feature analysis and metadata for one million songs, provided by Echo Nest. And the Million Song Dataset is a cluster of datasets including the dataset of songs, the dataset of lyrics, the user data and so on.

For our project, we mainly use the Echo Nest Taste Profile Subset and the Last.fm dataset. The Echo Nest Taste Profile Subset provides play counts for over 380,000 songs gather from 1 million users. The Last.fm dataset provides tags over 500,000 songs.

**6. Algorithms**

For music recommended system, one appreciable algorithm is using the collaborative filtering method. It relies on usage patterns, which is the combinations of items that users have consumed or rated. These lead to the users’ preferences and the relevance of the items. But for collaborative filtering, new items that have not been consumed before cannot be recommended because it recommends items consumed by other users with similar preference.

We would like to implement an algorithm related to deep convolutional neural networks for our project [4]. The Taste Profile Subset contains play counts per song and per user. The weighted matrix factorization (WMF) algorithm can be used as a modifies matrix factorization algorithm for the implicit feedback datasets. Let be the play count for user and song . For each user-item pair, we define a preference variable and a confidence variable ( is the indicator function, α and are hyperparameters):

The preference variable indicates whether the user has ever listened to the song . If it is 1, we will assume the user enjoys the song. The confidence variable measures how certain we are about this particular preference. And the WMF objective function is given by:

where is a regularization parameter, is the latent factor vector for user , and is the latent factor vector for song . Hu et al [5]. propose an efficient alternating least squares (ALS) as an optimization method for this function.

For predicting latent factors for a given song, there are two methods can be used to achieve this, bag-of-words and deep convolutional network. The bag-of-words methods first extract MFCCs from the audio signals, vector quantize the MFCCs and aggregate them into a bag-of words representation. The CNNs applied into the music recommendation system include ReLUs, parallelization and large amount of training data.

For hybrid music recommendation model, we’d like to combine ACE(Convolutional Auto-Encode) and MF(Matrix Factorization). The system will include three parts, that is item side, user side and MF. For the first part, we’ll input the two-dimension parameters of MCFF and Chroma and the characteristic of lyrics, which is one-dimension parameter. We’ll also define the loss function to evaluate the model, which includes loss of signal, lyrics and and so on. For the second part, since the dataset has not provided with the features that could create user profile, the only thing we have is the scoring vector. So we plan to build a autoencoder in user side, taking scoring vector as input. For the last part, we define the characteristic of item side and user side as Eigenvector U and V, therefore, according to Deep Collaborative Filtering [7] and [8], we develop Tightly Coupled MF model. At last, we will define the Loss function to evaluate this kind of model.

**7. Function and Result**

We expected the following results:

1.Exploring the advantages of collaborative filtering using deep learning versus traditional collaborative filtering.

In the traditional collaborative filtering, normal designers recommend similar item to similar user, based on Linear similarity about user or item, e.g. decomposition factorization. However, popular technology deep learning can discover nonlinear relationship on user or item without complex process.

2. Applying convolutional neural network to learn audio features.

CNN(convolutional neural network) is a kind of FNN(Feedforward Neural Network), but the difference between CNN and MLP(Multilayer Perceptron) is the way of connect. The latter is full-connected and the former is aimed to solve the high dimensioned problem. Perhaps we will focus on the two key parameter to do the two dimensioned calculation.

3. To develop hybrid music recommendation system

we expected to come up with CAE(Convolutional Auto-Encode) , which is in order to deal with the characteristic of MFCC and Chroma. Therefore, we will develop hybrid music recommendation system based on CAE and DF(decomposition factorization). We will also create the Loss function to evaluate the model.

**8. Compare Our Project with Existing Work**

In the mainland China, more and more traditional application turns to online music platform, and they all provided the music recommendation system.

Fabio Aiolli [6] participates the competition of Kaggle in Million Song Dataset Challenges and get the champion. The competition is aimed to develop optimal offline recommendation system. The data includes two parts, more specific, the half of the datasets are training sets, and the rest are testing sets. But the most difference between their project and mine is that we take the nonlinear factors into consideration.

**9. Timeline**

1. Configuring the environment platform Week5
2. Reviewing related work and looking for feasible models and algorithms Week6
3. Processing data Week7
4. Implementation Week8, Week9, Week10, Week11
5. Training and testing dataset Week12
6. Conclusion Week 13

**10. Reference**

[1] Kaggle. (2019). Million Song Dataset Challenge, [online] Available at: https://www.kaggle.com/c/msdchallenge/overview [Accessed 27 Nov. 2019]

[2] Mining of Massive Datasets, Rajaraman, Leskovec, and Ullman, 2013

[3] Million Song Dataset <http://millionsongdataset.com/>

[4] Van den Oord A, Dieleman S, Schrauwen B. Deep content-based music recommendation[c]//Advance in neural information processing systems. 2013:2643-2651

[5] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, 2008.

[6] Fabio Aiolli is an outstanding researcher in the competition of Kaggle. Available at: <https://www.kaggle.com/aiolli>

[7] Li S, Kawale J, Fu Y. Deep collaborative filtering via marginalized denoising auto-encoder[C]//Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. ACM, 2015:811-820

[8] Dong X, Yu L, Wu Z, et al. A Hybrid Collaborative Filtering Model with Deep Structure for Recommender Systems[C]//AAAI. 2017: 1309-1315