Content based music recommendation with deep learning

Wanhe Zhao
Carnegie Mellon University
Pittsburgh, PA
wanhez@andrew.cmu.edu

Junyan Pu
Carnegie Mellon University
Pittsburgh, PA
junyanp@andrew.cmu.edu

1 Introduction

When music streaming service providers are giving music recommendation to their users, a wide varieties of machine learning and deep learning algorithms have been used. Today's music recommenders such as Spotify, Pandora and iTunes have made a great progress in learning users' taste based on streaming histories and meta-information about the songs to minimize the user's effort to provide the recommender the feedback. In this project we will investigate in some of the dominant recommendation algorithms to train our own deep learning model. Inspired by lifehouse method project, a unique and groundbreaking portraiture system, launched by one of members at the Who, we found the inspiring idea by Pete Townshend: "I believe music is our only hope. I believe music can reflect who we really are - like a mirror"[1]. Therefore, building a personalized songs recommendation system becomes such an exciting field to do. We plan to build a deep learning model using datasets from Million songs library, with access to millions of users streaming histories and song features for song-level tags and similarity. The output will be a playlist that can reflect users taste. We will use million songs library for our training and validation sets, and as for test set we will collect some of the individual users with their Spotify streaming histories over the past three months. Then we will provide some recommended songs to the user, and collect their feedbacks on whether or not they like the songs for an evaluation.

2 Background and Literature Review

2.1 Collaborative filtering

This a traditional approach to power recommendations. The historical usage data is used for determining the users' preferences. This method can be viewed as a sequence prediction problem, and the intuition is that if two users with similar history streaming data, the system will recommend the content based on these informations. However, the main drawback for pure collaborative filtering approach is content-agnostic and the reliance on usage data. This leads to the cold start problem that new content is not being recommended, and also the recommender is bias towards popular songs as there is more usage data available for them [2]. "Heterogeneity of content with similar usage patterns" is also a drawback.

2.2 Content-based recommendation

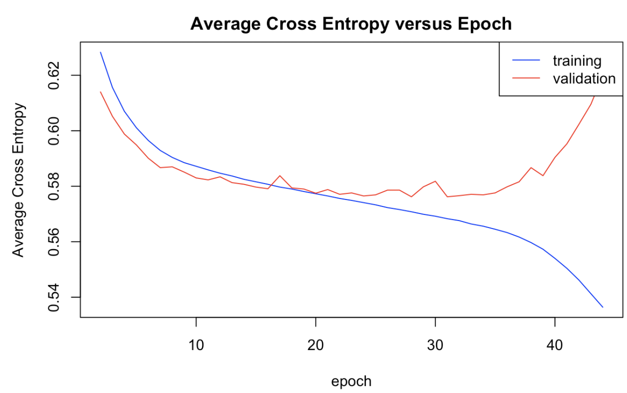
Music could also be recommended based on available metadata: lyrics, text mined from music reviews, the artist, album, year of release, and audio signal. The audio signal deep convolutional neural networks is successfully used by The Echo Nest, a music intelligence platform company acquisited by Spotify. Despite quite a large semantic gap between music audio, it is a promising idea to learn users preference through audio signal. Content-based recommendation has been tackled by many groups previously. One of the approaches train a regression model to predict the latent representations of songs that were obtained from a collaborative filtering model [3]. This method is able to provide recommendations even if no user data available, and generate playlists through the filters in the network that are picking up different properties of the audio signals. McFee et al. has defined an artist-level content-based similarity measurement using bag-of-words [4]. We plan to incorporate CNN in the model for an artist-level matrix learning, since previous works focused on song-level audio signal learning to generate similarity-based playlist, that is generating songs that sound similar. Artist-level recommendation will help us to generate theme-based playlist or radio, for example recommending similar songs to user's favorite artist, recommendation station, and daily mixes based on users' favorite artists available on Spotify.

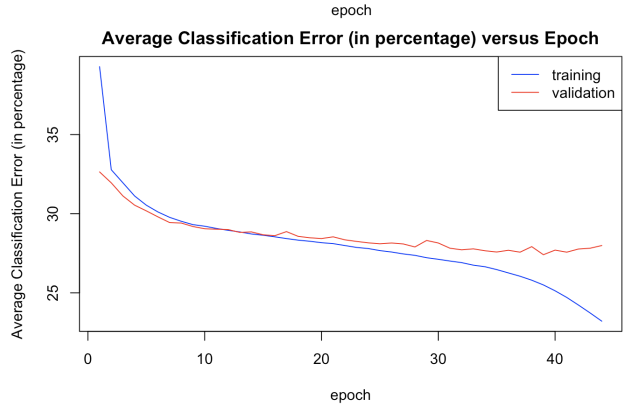
3 Methods and Model

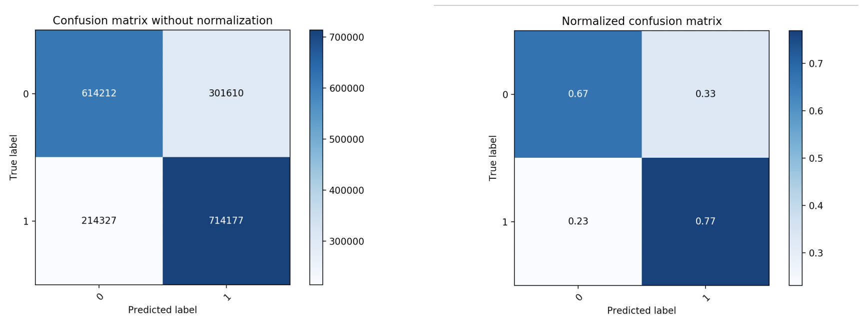
In our baseline model we used the collaborative filtering approach. We implemented our baseline algorithm on the kkbox dataset for music recommendation [5] using trainable embedding and neural network [6]. The kkbox dataset contains a total of 7377304 entries, with features including user/song pairs, features of songs, and users' information, and a target (response variable) 0/1 indicating whether the user listen to the song again within a month after user first listened to the song. We shuffle the dataset, and choose 3688652 samples as train set, 1844326 samples as validation set, and 1844326 samples as test set. In our model, we first encode member id, song id, and other categorical variables. Second, we transform member id and song id using trainable embedding to project them into a 64 dimension vector space. Then, we take the dot product of embedded member id and embedded song id as a new feature (user, song pairs). After these two steps, we feed in all features we have into a neural network with 128 hidden units with reLU activation function. Then, we apply dropout with rate 0.5 and use linear mapping with softmax to map to the target (response variable).

4 Preliminary result

During the model training, we applied early stopping with patience of 5 epochs. Our model has 0.7679 accuracy on the training set, 0.7201 accuracy on the validation set, and 0.7948 accuracy on the test set.







We can see how train and validation cross entropy loss and accuracy change over the epoch. The normalized confusion matrix and confusion matrix without normalization are both shown.

5 Evaluation of Preliminary Results

According to the confusion matrix, we have sensitivity equals 0.77 and specificity equals 0.67 on the testing set. Given that we have a large test data set (1844326 samples), we have enough evidence to say that our baseline model has an improvement compared to random guessing. Since this kkbox dataset only contains extracted information from user and songs, we have a limited number of features (18 features). Therefore, it is necessary that we do feature engineering including embeddings in order to expand the dimension of the input. What's more, the entries in the dataset have an inherent chronological order. However, in our baseline model, we did not use any time series related assumptions during the training process. Therefore, using our baseline model, we are not able to conclude whether more recent entries are more helpful in predicting future recommendation results.

6 Future work

One limitation of collaborative filtering is that we used extracted information and features instead of taking audio data as our input, which means that the spatial information are lost during the feature extraction process. Therefore, in our next step, we will implement a content based convolutional neural network which takes in audio features as input. By using the content-based algorithm, we will be able to extract unique audio features from our input without any loss of spatial information. Therefore, we believe the content-based CNN model will be more flexible and more precise than our baseline model.

7 Teammates and work division

<table>
<tr><th></th><th>March 08</th><th>By April 5</th><th>By April 22</th></tr>
<tr><td>Junyan Pu</td><td>Implemented baseline model</td><td rowspan="2">Find dataset with audio features &
Initial implementation of content-based CNN</td><td rowspan="2">Improving content-based CNN model &
Model evaluation</td></tr>
<tr><td>Wanhe Zhao</td><td>Background/literature review and Citation</td></tr>
</table>

References

[1] Pratt, C., "Lifehouse Method" *Pete Townshend's website.* Available at : `https://petetownshend.net/musicals/lifehouse-method`.

[2] Dieleman, S., "Recommending music on Spotify with deep learning" *Sander Dieleman's blog.* (2014). Available at: `http://benanne.github.io/2014/08/05/spotify-cnns.html`.

[3] Oord, A.v.d, Dieleman, S. , Schrauwen, B., "Deep content-based music recommendation" *NIPS Proceedings.* (2013). Available at: `https://papers.nips.cc/paper/5004-deep-content-based-music-recommendation.pdf`

[4] Brian M., Luke B., and Gert R. G. Lanckriet., "Learning content similarity for music recommendation." *arXiv.* (2011). Available at: `https://arxiv.org/abs/1708.02182`.

[5] Kaggle contributors, "WSDM - KKBox's Music Recommendation Challenge." *Kaggle* (2018). Available at: `https://www.kaggle.com/c/kkbox-music-recommendation-challenge`

[5] Kaggle contributors, "Beat KKBox Benchmark without using metadata" *Kaggle* (2018). Available at: `https://www.kaggle.com/lystdo/beat-kkbox-benchmark-without-using-metadata-0-62`