Automated Playlist Extension: Music Recommendation with Million Playlist Data using Collaborative filtering Approach

Anonymous ACL submission

Abstract

Currently music service providers have generic (and popular), mood-based playlists, that are the same for all users. The goal of this paper is to introduce a music recommendation system that can provide customized music recommendation for any given playlist based on collaborative filtering. By suggesting appropriate songs to add to a playlist, a Recommender System can increase user engagement by making playlist creation easier, as well as extending listening beyond the end of existing playlists. Here, we device a system which retrieves the semantic similarity between playlists and recommend music based on the similarities. The link to github: https://github.com/zn8ae/RecSys2018

1 Introduction

Traditionally, Spotify has relied mostly on collaborative filtering approaches to power their recommendations. The idea of collaborative filtering is to determine the users preferences from historical usage data. For example, if two users listen to largely the same set of songs, their tastes are probably similar. Conversely, if two songs are listened to by the same group of users, they probably sound similar. This kind of information can be exploited to make recommendations.

Pure collaborative filtering approaches do not use any kind of information about the items that are being recommended, except for the consumption patterns associated with them: they are content-agnostic. This makes these approaches widely applicable: the same type of model can be used to recommend books, movies or music, for example.

In this work, we are given the Million Playlist Dataset (MPD) from Spotify. Each playlist contains information such as name, description, number of songs, as well as the detail about each track in this playlist. However, no user information is given, nor do we have any subjective attitude expressed towards these playlists such as ratings, listening behaviors, etc. Thus, we build our recommendation system utilizing the semantic characteristics of these playlists. Our collaborative filtering approach takes into consideration the semantic similarities between playlists and give recommendations accordingly. In this benchmark, we are only considering the similarity between two playlists' normalized song names(lower case all characters and remove punctuations), however, in order to fully measure the semantic similarity/differences, we will also take into consideration album, artist, and description data in the future.

This paper is organized as follows: First, we will describe the dataset, how do we clean and organize the data, as well as the key insight retrieved from the data set. Then, we will describe the structure of our collaborative filtering model. Lastly, we will present our experimental setup along with some results generated by our model.

2 Dataset

2.1 Source

The MPD contains a million user-generated playlists. These playlists were created during the period of January 2010 through October 2017. Each playlist in the MPD contains a playlist title, the track list (including track metadata) editing information (last edit time, number of playlist edits) and other miscellaneous information about the playlist. Playlists that meet the following criteria are selected at random:

- Created by a user that resides in the United States and is at least 13 years old
- Was a public playlist at the time the MPD was generated
- Contains at least 5 tracks
- Contains no more than 250 tracks
- Contains at least 3 unique artists
- Contains at least 2 unique albums
- Has no local tracks (local tracks are non-Spotify tracks that a user has on their local device)
- Has at least one follower (not including the creator)
- Was created after January 1, 2010 and before December 1, 2017
- Does not have an offensive title
- Does not have an adult-oriented title if the playlist was created by a user under 18 years of age

2.2 Insights

As we conduct our explanatory data analysis, we have found the following aggregate statistics about the dataset:

- number of playlists 1000000
- number of tracks 66346428
- number of unique tracks 2262292
- number of unique albums 734684
- number of unique artists 295860
- number of unique titles 92944
- number of playlists with descriptions 18760
- number of unique normalized titles 17381
- avg playlist length 66.346428

Since we will recommend according to the semantic information, it is necessary for us to retrieve the characteristics of some important characteristics about the data, such as top playlist titles, top album, top artists, etc. For example, these are what we have found in our explanatory data analysis about the top playlist titles:

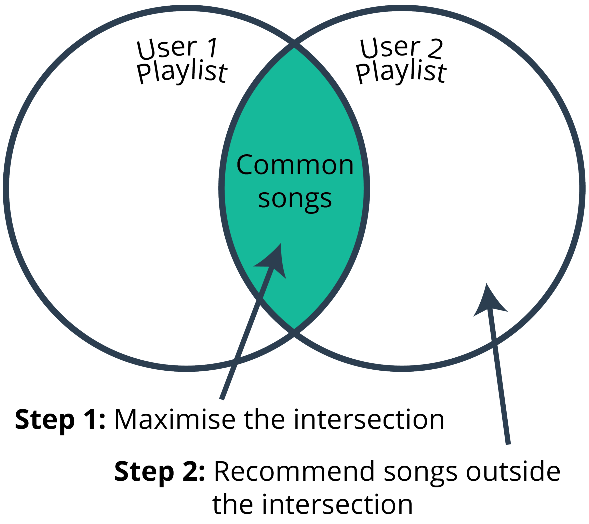
User 1 Playlist
User 2 Playlist
Common songs
Step 1: Maximise the intersection
Step 2: Recommend songs outside the intersection

Figure 1: Similarity Venn Diagram

- 10000 country
- 10000 chill
- 8493 rap
- 8481 workout
- 8146 oldies
- 8105 christmas
- 6848 rock

In our notebook demo, we will demonstrate a full statistic about characteristics we have found about the dataset.

3 Collaborative Filtering Model

To recommend items via the choice of other similar users, collaborative filtering technique has been proposed .

As one of the most successful approaches in recommendation systems, it assumes that if user X and Y rate n items similarly or have similar behavior, they will rate or act on other items similarly. Instead of calculating the similarity between items, a set of nearest-neighbor users for each user whose past ratings have the strongest correlation are found. Therefore, scores for the unseen items are predicted based on a combination of the scores known from the nearest neighbors.

In our case, two playlists are similar if they have shared many of the same songs. The method used to recommend songs will be to recommend songs that are similar to songs already in a given playlist. This is shown using the Venn diagram in Figure 1.

3.1 Algorithm

First, to process our training set, we will create a dictionary that maps each playlists' pid which is unique to its track list. The reason to do this is that in the future it will be very convenient for us to locate all the tracks of a playlist given its pid.

Then, given a testing playlist which needs recommendation, we will first collect all the names of tracks, normalize them in a list, and then compare the list to all playlists in our training set, calculating the similarity score for all training playlists.

Assume for training list L, the number of songs shared between testing list and L be M, and the total number of songs in the testing list be T. The score function is calculated as follow:

$$S = M/T \tag{1}$$

We will calculate the score for all playlists in our training set and then rank the scores from high to low. Finally, we will recommend songs that are not in the testing playlist from the ranked playlists, where the most similar album will contribute first.

4 Experiment Setup

We will select the first 1000 playlist in our experiment. We will randomly split the data into 750 playlists as our training set and 250 playlists as our testing set. First, we will perform aggregated explanatory data analysis on the whole 10000 playlists, and then we will process to data to prepare for running our model such as building a dictionary mapping pid to tracks for our Collaborativer Filtering Algorithm. Then, we will perform our algorithm by going through each of the testing playlist and produce recommendations according to similarity calculated between the testing playlists and training playlists.

5 Result and Analysis

In order to test the data, we calculated the recommendation accuracy for testing playlists. In order to calculate the accuracy, we first split the data into training set and testing set, then for each playlist in the testing set, assume the playlist has N tracks, we will divide the tracks in half and withheld the latter N/2 of the tracks from the playlists. After withholding the tracks, we will perform recommendations and suggest N candidate song recommendations. We will then compare the withheld songs and candidate generated from our model to calculate accuracy. Let the number of shared songs between withheld playlist and our recommendations be S, and let the length of original testing list be N, The accuracy is calculated as follow:

$$Accuracy = S/N \tag{2}$$

We found our Collaborative Filtering approach producing accuracies averaging at 0.15. We believe this is due to the simplicity of our model - we are only considering the names of the songs in each playlists. By constructing a more factors and building a more complex scoring model, we will be able to significantly improve the result. After all, by achieving 0.15 accuracy using only the name of the songs, we believe Collaborative Filtering is a fundamentally sound apporoach.

6 Future works

In the future work, we will improve the complexity of our model by taking into consideration more complex attributes such as the shared album, artists as well as the number of playlist follower, etc. We believe these attributes will help our Collaborative Filtering model to reach a much better performance.

In the meanwhile, we believe the semantic characteristics of the playlists could be better represented by word2vec model. To be more specific, in word2vec model, we can assign vector value to each song in the playlists so that songs that share playlists will sit closer in the coordinate system. Then by calculating the cosine similarity / euclidean distance of the vectors, we will be able to make song recommendations according to the existing songs in a playlist.

We are also interested in considering more extreme cases in music recommendation. For example: what if a playlist only contains one or two songs? what if a playlist contains no songs but only provides a name or a piece of text description? By looking into more semantic cues, we believe these problem could also be effectively addressed.