

Final Project Progress Report

Report two

Xingbang Liu, Matt Jones, Travis Thomas

April 14, 2018

So far, we have finished implementing the basic structure of the program. We have main class which is a do while loop, if user don't want to quit, the program will continue load conversion class. conversion class is a class that will require commands from user. Based on different commands, the class will redirect users to different classes. For example:

Menu: 1.Like | 2.Explore | 3.Daily Selection | 4.quit

If user type "1", then the program will direct user to liked song list, then like class will show user the music list. The like class will also ask the user to sort the list if the user wants. In the rest of important classes, such as *explore()* and *dailySelection()*, we will implement our random suggesting system and forecast suggesting system.

In explore class, the program will read the Cloud musiclist CSV file and randomly select few songs and feed it to user based on song's name initial letter sequence. If the user likes any of the songs, we will ask if the user want to add it to his liked music list. If he do, the program will mark a song with y in the last column. and add it to like list. *need more stuff here*

In the daily selection class, the program will select songs that the user may like based on the forecasting of whether the user will like it (The algorithm will be explained later). *need more stuff here*

Algorithm Picking

For the algorithm for forecasting potential user like song, which is Collaborative filtering, there are two algorithms that we know.

Memory-based

The first one is user grouping method. In this method, the algorithm will first track down people who have similar activities, then recommend songs to each other by their playlists. For example, scores are assigned to different actions:

Repeat song = 5, Share = 4, Like = 3, Play = 2,
Played completely = 1, Skip = -1, Dislike = -5

Then a person's preference would be a N-Dimensional vector. Here N is the number of songs by default. By using the cosine of the vector angle, which is generated by two vectors, we can know how similar two users can be. Cosine of 0 degree, which means two people are exactly the same, is 1. Cosine of 180 degree, which means two people have the opposite preference, is -1. The equation is:

Assume we have vector \vec{a} and \vec{b} ,

$$\cos(A) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| \times |\vec{b}|}$$

This prediction method is very accurate, but there are too many calculations, and it is not accurate for new user.

Model-based

As for the second method, the algorithm will analyse the playlist of a person, then do recommendations. Models such as latent factor is used in this method. ***I would explain more here***

Of course, in the real world, if we implement algorithm based on song, the second dimension factor will rapidly increase. Plus there will be too many songs. Therefore it will be more efficient if we calculate based on tags of the song. For example, for “The Sound of Silence” by Simon & Garfunkel, tags can be Folk Rock, 60s, and Movie Track. Due to the time we have is limited, we would not build tags databases, therefore we would analyse by songs. Additionally, we would use the second method to calculate scores because we do not have time to build multiple users.