Status Report of EECS 349 Machine Learning

Team Members:
Yaliang Wang (ywo130)
Sisi Chen (sci963)
Haomin Zeng (hzy075)

Our task is to recommend an artist that is of high possibility to be liked by a target user. This result is based on training input dataset which contains users' listenning records and artists' information. This task is important because it gives users suggestion about the potential artists they might like but have never known about, which helps users extend their listenning list.

Initially, we use the dataset that contains 92834 listening records from 1892 users that we gain from http://grouplens.org/datasets/hetrec-2011/. Each user in the dataset has a list of listened artists with listenning counts. For each artist, there is a list of tags labeled by the users in this dataset. In total, 186480 tags are assigned by 1892 users to the 18745 artists. Furthermore, the total number of different tags is 12648.

In order to validate the recommender's performance, we firstly sample part of users as test user. The remaining data, which include the records of the training users as well as the tags information of all artists, are used as the trainning set of the recommender system. Secondly, we remove test user's most favorite artist record, which we consider as the artist with highest listening count. For each test user, we predict an artist that is most likely favorited by the test user and set that as the recommendation. We exam the predicted one with the artist we previously removed.

The system is a collaborative filtering recommender which firstly find the K nearest user in feature space. In order to represent a user, we calculate the tag attributes for each user. The tag attributes of a user equals to the sum of tag attributes of an artist times the corresponding listening count. The listening record of a neighbour is weighted by the distance in the feature space. By combining all neighbours' weighted record, we recommend the top 1 artist. The average accuracy where the recommended artist matches with the artist which the test user most listened is 10.15%. The percentage of user which the recommended artist matched with the artist that the test user listened before is 63.58%. For the rest users, we recommended an artist that the test user never listened.

The dataset we used lose the timestamp of listen record. Right now, we use partial listening record to predict the favorite artist of a given processed test user. In this way, we evaluate our recommender's performance by comparing the initial most favorite aritist with the recommendatory artist. This method is kind of unreasonable, as we remove the most favorite artist, we influence the user’s taste to a large extent.

Thus, we currently download the data by using the API on last.fm and try to reevaluate the performance. We devide the dataset into two parts according to their time order, and using the previous listening records of a user to predict the most likely favorite aritist which is not listened, then check whether the predicted artist has been listened by the given user in the later time. This method is kind of more reasonable, but still tricky.

There are two possible techniques we may be able to use for the comparison. First one is the Gaussian Mixture Model. We set each artist as a component of the model and try to learn the parameter of the model. The users later are clustered into these n components based on their listening record. For a test user, we can calculate the posibility that belongs to each artist and recommend the artist with the highest posibility. The second one is the nerual network. The input layer of the NN can be the tag attributes of a user. Each node in the output layer is the posibility which represents the likelihood that an artist being the user’s most faviourite. For the trainning user, the node in the output layer representing the most listened artist is set to be 1 and the other nodes in the output layer are set to be 0. For the testing user, we pick up the the node with highest value and recommend the artist corresponding to this node.