Final Report

Team member: Yaliang Wang
Haomin Zeng, Sisi Chen

Introduction and Motivation:

In this project, our task is accomplishing streaming media recommender. In other word, given a user with finite listening records, the recommender figure out a potential artist who fit the user's music taste mostly among our artist database.

It is of both high interestingness and great significance. On the one hand, this recommender benefits users of streaming media a lot. Usually, given users might have a same taste of music, the recommender can compute out new artist who satisfy users personal music taste. On the other hand, the recommender also benefits provider of streaming media. The providers can design personalized music album for specific users to cater to users' music favor then make more profits.

Recommender Construction:

To train and test the recommender, we download unprocessed dataset from Grouplen website. This is a dataset with 92834 listening records from 1892 users. Using K-Nearest Neighbor algorithm, we train and test our recommender as follow:

Data structure: For each user there is a list of his/her favorite artists, each listened artist has a listening record denoting the listening times. For each artist there is a list of his/her genre, which is a list of tags given by users that have ever listened to the music from the artist.

Training: The training procedure in KNN is trivial and easy. We training the recommender with the whole dataset, simply adding each user sample into feature space with his/her complete artist listening records. We use tag to characterize our feature space. In the feature space, each tag has a normalized weight. This weight is an accumulative result for all artists who has the given tag. Each addend is the product of normalized tagged count and listening time of corresponding artist.

Testing: As for testing, we use recommender to repeatedly predict favorite artist for all the users in the dataset, then comparing the result with their actual favorite artist. Firstly, we remove the test user from feature space, delete his/her favorite artist from record. Then, we put this processed user into feature space to find out his/her K nearest neighbors. Initially, we use Euclidean distance to compute the tag distance between test user and trained user. After figuring out the K nearest neighbors of the test user, we scan into the listening record of these K nearest neighbors and pick out the artist(the one doesn't contain in the listening record of the test user) with highest weight to recommend to the test user. This highest weight choice is based on Gaussain function: G(x) = a*exp(- (x - b) ^2 / (2 * c^2)), where x denotes the

computed Euclidean tag distance between test user and each one of its K nearest neighbor, and we implicitly set a = 1 and b =0. Multiply result from Gaussian function with listening times of corresponding artists, we compute out artist weight.

Recommender Evaluation:

At this part, we firstly design a naive algorithm as baseline for our recommender evaluation. This algorithm compute all the artists weight, and rank them in decreasing order of weight. This means we have a list of artists who is listened more by users in dataset will be ranked higher. Then, we predict the potential favorite artist for each test user(who's favorite artist has been removed before testing) according to the ranked list. Moreover, to increase the accuracy of our baseline algorithm, we only recommand the artist that does not contain in listening record of the test user.

Furthermore, to refine our recommender, we manually change the decisive parameters in our KNN algorithm, namely, the value of K, the distance function and the standard deviation in Gaussian function.

Given above evaluation assumption, our recommender performance evaluation can be denoted in following sheet:

		Baseline Accuracy	7.77%
K	Distance Function Norms	Gaussion Function Parameter c^2	Accuracy
5	2	0.4	26.80%
10	2	0.4	28.54%
15	2	0.4	28.91%
20	2	0.4	29.60%
25	2	0.4	29.49%
30	2	0.4	29.60%
35	2	0.004	30.34%
35	2	0.4	29.02%
40	2	0.004	30.39%
40	2	0.4	28.54%
45	2	0.4	28.65%
50	2	0.4	28.17%
55	2	0.4	28.28%
60	2	0.4	27.96%
65	2	0.4	27.75%
70	2	0.4	27.80%

Comparing with accuracy result using baseline algorithm, it is obvious that our recommender system does have a better prediction behavior than naive recommender.

We deliberately ignore the change in the norm value of distance function in our sheet because according our primary test, only Euclidean distance function has the best predicting performance as the value of K and c^2 fixed.

Then, we increase K by the equal interval each time, according to the table above, we find out that our recommender system has the best prediction accuracy when K = 35 or K = 40.

Finally, we do some test on learning the influence of standard deviation on our KNN algorthm based recommender, and find out the large change on the value of c^2 only result slight difference on final accuracy.

Future Work:

One of the major work we are interested in in future is to construct our recommender system with more applicable algorithms. One of our proposed potential alternative is GMM. We have already gotten some ideas about it that we use each artist as a compenent in Gaussian Mixture Model, then putting test user in this feature space, find out the most suitable Gaussian distribution and recommend the artist to the test user.

Also, we have pulled a larger dataset with timestamp from last.fm using provided API, considering using timestamp information to split the dataset into training and testing set. Based on timestamp, we could predict the most favorite artist of a user in later time given his/her previous listening records.