***Final Project EE627***

*Group: one piece*

*Ledong Tang*

*Chengjingwen Tang*

*Qi Li*

*Instructor: Prof. Wang*

*Date: 12/18/2016*

**1. Introduction**

Humans are living in a time of information exploration now. Everyday people are forced to accept large amount of information especially in commerce. Customers are often confused about how to choose commodity. In other words, customers need someone or something to help them make a decision. Therefore, huge profits are potential in recommendation. In one conference of Amazon.com, a report shows that 20% of the interests are from commodity recommendation. In the recommendation system, however，music recommendation is a special one. The reason is that a customer will not buy an item repeatedly in a short period and prefers the favorite one at most time. Hence a good music recommendation will play a key role in improving company profit.

**2. Basic principles**

a. Hierarchy structure is used in this project. This project tried several ways about how to recommend music to user by the information provided by users’ rating score on different music. And the data set is from Yahoo and it is from real market. As commonly known, music has some special features that can be used as tracking path which is called hierarchy structure. Hierarchy structure have following 5 features:

1. A track belongs to an album.

2. An album is under an artist.

3. An artist has no less than one genre.

4. Users can rate any items with score between 0~100.

5. Hierarchy List: Track, Album, Artist, Genre 1, …, Genre K.

The whole data set is divided into two parts: training data and test data. The task is to get the rating information from training data, and do recommendation for the testing data, which has six tracks per user to be recommended, label “1” means it should be recommended while label “0” means it is not worth recommendation.

In this project, we use music data that is stored in hierarchy structure for analysis.

This structure is shown as follows:

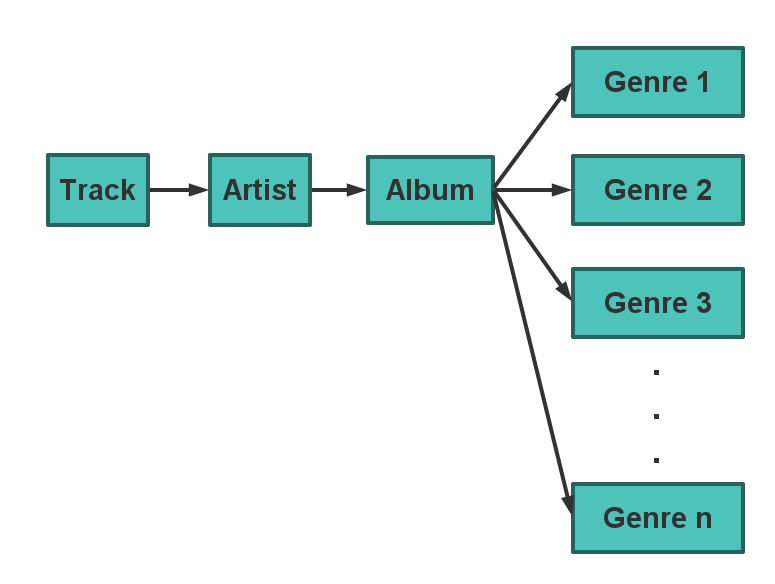


Fig.1 Hierarchy Structure of Yahoo!Music Data

In this hierarchy structure, each song track belongs to an album, each album is with an artist and each artist is in one or more genres. Users can rate Tracks, Albums and (or) Artists with scores from 0 to 100. Based on the rating, we use specific algorithm to find out which three of six unrated tracks provided in test data may be rated as highest scores and recommend them to a specific user.

**3. Training data and testing data**

Data file provided in two parts. One is training data and the other is testing data. In the file *testldx2.txt,* there is 6 tracks listed for user i, where i is the number on the left side of a “|”.

The first number of each line in file *trackData2.txt* represents track, the second number represents album to which the track belongs, the third number represents the artist who has this album and the rest ones represent genres that artist is in. They are separated by “|”.

In the file *trainldx2.txt*, user i (on the left side of a “|”) rated n items (on the right side of a “|”), the item numbers and the accordingly ratings are listed.

**4. Analysis Process**

The analysis process and recommendation is done in the following step:

Step1: Find six tracks for user i in testldx2.txt.

Step2: Find the album, artist and genres each track belongs to in trackData2.txt.

Step3: Search every items, including albums, artists and genres we get from Step2 in trainldx2.txt, record the rating of these items by user i. If there is no rating for some items, record 0.

Step4: Use items and their ratings we got from previous steps as variables in function to obtain the predict score user i may give for each track.

Step5: Compare the predict scores of 6 tracks and choose the highest 3 as recommendation.

**5. Programming Process (in JAVA)**

**a. Read files**

For preparation, use FileReader and BufferedReader to transfer the whole txt file from string to integer, then store the information in arrays.

(a) Transfer users and 6 tracks data in testldx2.txt to array Des[ ] in function read\_track.

(b) Transfer tracks, albums, artists and genres relation data in trackData2.txt to separately

store in array track[i] in function read\_track\_data.

(c) Transfer ratings for items in trainldx2.txt to array Rates[i] in function read\_rate.

**b. Calculate to obtain the prediction**

In calculation session, we come out predict scores for 6 tracks according to the recommend process described in part II, then store the scores in array tempeRate[i].score[j].

We have tried several methods to make the predict scores have top precision including calculate the middle, overall average and average without the highest rating and the lowest one and so on.

Finally, we realize that it is important to find out the best weight combination of ratings for tracks, albums, artists and genres.

After several attempts, the best weight combination we can have is:

*Rpredict = Rtrack \* 0.436 + Ralbum \* 0.912 + Rartist \* 0.591 + Rgenre \* 0.05*

When there is no rating for genres:

*Rpredict = Rtrack \* 0.436 + Ralbum \* 0.912 + Rartist \* 0.651*

**c. Compare and return the highest 3 predict scores**

When we have 6 predict scores for 6 tracks, we use 3 loops in function data\_order to obtain 3 tracks that have the highest 3 predict scores. Set these 3 tracks to value “1” and the rest to “0”. The value-1 tracks is the recommendation we made in this project.

# 6. Result records

We totally submit 8 results and their accuracy is 0.5266 0.7779 0.8161 0.8368 0.8484 0.8485 0.8674 0.8831 respectively.

# 7. Conclusion

In this final project，we use JAVA code to do the algorithm tests. We use hierarchy search to get the ratings that is related to the 6 tracks, among which we are going to make recommendation. Then we try different algorithms including change the weight of the items ratings we get to obtain a highest accuracy of recommendation. When we change the weight, the accuracy of recommendation varies and the highest one we have is 88.31%.

We used different methods to get this result. Through this project, we found that different methods mean different thoughts. You cannot easily judge which one is better, which one is not. Although our team has already tried several different methods, it’s not quite enough. However, we tried our best to get we want. We have thought about the neighborhood method and matrix factorization in processing the origin data. Even though we tried a lot, we failed because neither of us has the ability to type the code about it. If possible, I hope the professor could tell us more ways to complete this project.

We learned a lot from this project, like different methods about data mining, team work, real market analysis skill, etc. We realized that what really matters is the combination of multiple ways to solve one problem. Each one would have his own approach. Different approach means different advantages. The comprehensive result should conclude from all the possible approach, and it would be the best of best.