CPE627 final project report

Yang Li
Ziao Ye
Minqi Guan

For this project I use two methods to get the prediction value:

1. pyspark mllib recommendation algorithm
2. logistic regression

1. pyspark mllib recommendation algorithm

Pyspark offers us a model, pyspark.mllib.recommendation module, which could help us to make recommendation by using matrix factorization. To make prediction, we need training data. For Yahoo Music data set, we could get rates of some items for each user. And these data will be the training data. Moreover, I classified it into tracks' rate of all users, albums' rate of all users, artists' rate of all users, and genres' rate of all users, and call them Track_Training_Data, Album_Training_Data, Artist_Training_Data, and Genre_Training_Data respectively.

Now, I need to make predictions for 6 tracks for each user. Following is the steps to get the prediction value:

1. Find out each track belongs to which album, artist, and genres.
2. Use spark get prediction rate for the track, and the prediction rate for album, artist, and genres which it belongs to.
3. Apply different weight on prediction rates. In my way, weight for rate, album, artist and genres are 0.325, 0.275, 0.225 and 0.175. Because one track may belong to multiple genres, I use mean value as the rate of genres.
4. Find out the higher 3 ones and make them prediction values are “1” and the lefts 3 ones' prediction values are “0”

Problem For This Method:

Because the Yahoo Music data set is not big enough, there would be some items which we need to make prediction are never be rated by all users. In this case, we cannot get the prediction of this items. Pay more attention, item could be track, album, artist, and genre. Therefore, sometimes we could not get all the needed prediction rate, album rate, artist rate, and genre rate, for each track. I have no other choice but just use mean rate value replace them. Of course this will influence the result of prediction.

2. logistic regression

This method also need training data. Following is the step to get the prediction value:

- Use spark get Following relationship file:

album_tracks	List all albums and tracks belong to them
artist_tracks	List all artists and tracks belong to them
artist_albums	List all artists and albums belong to them

genre_tracks	List all genres and tracks belong to them
genre_albums	List all genres and albums belong to them
genre_artists	List all genres and artists belong to them

- Use spark get mean rate value of tracks, albums, artists, and genres for each user
- Find out each track belongs to which album, artist, and genres for test data
- Generate the predictor matrix for test data set:

    - To see if user rate on the album. If does, store the rate into track's data feature. If does not, store the album mean rate of this user into this track's data feature.
    - Fetch all the tracks of the album. And find out the the *Eight_Data_Feature** for these tracks and add these *Eight_Data_Feature** to this track's data feature.
    - To see if user rate on the artist. If does, store the rate into track's data feature. If does not, store the artist mean rate of this user into this track's data feature.
    - Fetch all the tracks of the artist. And find out the the *Eight_Data_Feature** for these tracks and add these *Eight_Data_Feature** to this track's data feature.
    - Fetch all the albums of the artist. And find out the the *Eight_Data_Feature** for these albums and add these *Eight_Data_Feature** to this track's data feature.
    - To see if user rate on the genres. If does not, make its rate value equal to mean genre rate of the user. And then find out the the *Eight_Data_Feature** for these genres and add these *Eight_Data_Feature** to this track's data feature.
    - Fetch all the tracks of all the genres. And find out the the *Eight_Data_Feature** for these tracks and add these *Eight_Data_Feature** to this track's data feature.
    - Fetch all the albums of all the genres. And find out the the *Eight_Data_Feature** for these albums and add these *Eight_Data_Feature** to this track's data feature.
    - Fetch all the artists of all the genres. And find out the the *Eight_Data_Feature** for these artists and add these *Eight_Data_Feature** to this track's data feature.

  In this way, I generate 58 data feature for each track in test data set.

- Generate a training data set.
- Find out each track belongs to which album, artist, and genres for test data
- Just as the test data set, to generate 58 data feature for each track in training data set
- Use training data set get the factors. And then use logistic regression to get the probability to recommend for each track in test data set.
- Find out the higher 3 ones and make them prediction values are “1” and the lefts 3 ones' prediction values are “0”

Eight_Data_Feature: Whenever we get a list of items, we will find out their max value, min value, median, mean, high_median, low_median, high_range(number of rates which is bigger than mean value), and low_range(number of rates which is bigger than mean value). I call these 8 values *Eight_Data_Feature*

Problem For This Method:
For logistic regression, the training data is very important. However, because the Yahoo Music data set is not big enough, it impossible to fetch 3 tracks with high rate and 3 tracks with low rate. For example, user have a high rate on a genre, but this genre ID cannot be found in trackData2.txt file. And for some

user, for example user 171, it has few rate on items and just one item in the trackData2.txt file. Therefore, I did not generate myself training data and just use the training data offered on canvas.

Both these two methods did not perform well on predication, but I have tried my best. And I have analyzed the possible problem. Sorry for that I have no time to write more comments on my code. If you have any questions, please take easy to ask me.