Data Science

Term Project

컴퓨터전공

2013011695 정태화

1. Goal

Predict the ratings of movies in test data by using the given training data containing movie ratings of users.

2. Summary of algorithm

This program goes through following procedure :

1. Read the entire training data in the data type of map.
2. Use this training map to make 2D-array and average rating of each user's rating info.
3. Using PCC and cosine similarity, make neighbor list of each user in descending order.
4. With this neighbor list, predict rating of every items in test file and write it into output file.

3. Detailed description of code

A. `map<int, map<int, int> > readTrain(string path)`

This function reads data from given input base file, and make each user's rating data to map type. Each data is consisted of its user id, item id, its rating, and timestamp.

B. `vector< vector<int> > mapToArray(map<int, map<int, int> > &)`

This function transforms map data, the result of `readTrain()`, into 2D-array, and also calculates

each user's average rating to predict rating of items which have never been rated.

C. `double cosine_similarity(int user_a, int user_b)`

Calculates cosine similarity value of 2 different users.

D. `pair<double, int> pcc(int user_a, int user_b)`

Calculates Pearson correlation coefficient value of 2 different users. This returns similarity value and total count of items rated by both users.

E. `map<int, vector< pair<double, int> > > getPccNeighbor()`

Get neighbors of every users using similarity values. In this case, I used both Pearson correlation coefficient and cosine similarity to get similarity values by multiplying those two values. And then, sort it into descending order to see which one is most similar and which one is not. Also, if I cannot find any similarity between 2 users, it is not considered as a neighbor. For example, if similarity value of user a and b is calculated as NaN, that user is not in neighbor list at all.

F. `double predict(int user, int item, map<int, vector< pair<double, int> > > &similarity_list)`

Predict specific item's rating in the view of specific user using that user's neighbors' rating. I used K-nearest neighbor(KNN) with means algorithm to predict rating.

$$\hat{r}_{ui} = \mu_u + \frac{\sum_{v \in N_i^k(u)} sim(u,v) \cdot (r_{vi} - \mu_v)}{\sum_{v \in N_i^k(u)} sim(u,v)}$$

G. `void writePrediction(string basePath, string testPath, map<int, vector< pair<double, int> > > &similarity_list)`

Get every prediction results of test items and write it into output file. If predicted rating is lower than 1, I considered it as 1, and did same to the ratings which are over 5.

4. Result

<RMSE>

- u1 : 1.010416

- u2 : 1.002108

- u3 : 0.9943647

- u4 : 0.9898998

- u5 : 0.9917092

5. Instructions for compiling this code

- This project contains ‘recommender.cpp’, ‘Makefile’, every data files, and all results of predictions.

- In project folder path, just type ‘make’ in terminal, or please type below line. This will generate executable file ‘recommender’ for linux.

```
$ g++ -O2 -o recommender recommender.cpp --std=c++11
```

- Now you will be able to execute this file with specified arguments.

```
$ ./recommender u1.base u1.test
```

6. Any other specifications

- This code is written in C++11.

- Compiler must support C++11 standard.

- This program is compiled with g++ and xcode.

- This program compilation is tested on macOS High Sierra.