COEN 272 Web Search and Information Retrieval

Project II (100 points + 20 bonus points)

Deadline 1: First submission, 5pm,

Wednesday, May 22, 2024

Deadline 2: Full report with code, 5pm,

Wednesday, May 29, 2024

Introduction

In this project, you will develop different algorithms to make recommendations for movies.

You are free to choose *any programming language that you like* such as Python, Ruby, R, Java, C/C++, Matlab, etc.

Movie Recommendation System

The Training Data

[train.txt] The training data: a set of movie ratings by 200 users (userid: 1-200) on 1000 movies (movieid: 1-1000). The format of the data is as follows: the file contains 200 blocks (users) of lines. Each line contains a triple : (U, M, R), which means that user U gives R points to movie M. In other words, U is the User ID, M is the Movie ID, and R is the corresponding rating. A rating is a value in the range of 1 to 5, where 1 is "least favored" and 5 is "most favored".

Please download the training data here: train.txt.

The Test Data

There are three test files: test5.txt, test10.txt and test20.txt.

[test5.txt] A pool of movie ratings by 100 users (userid: 201-300). Each user has already rated 5 movies. The format of the data is as follows: the file contains 100 blocks of lines. Each block contains several triples : (U, M, R), which means that user U gives R points to movie M. **Please note that in the test file, if R=0, then you are expected to predict the best possible rating which user U will give movie M.** The following is a block for user 276. (line 6545-6555 of test5.txt)

```
276 42 4      // user 276 gives movie 42 4 points.
276 85 2      // user 276 gives movie 85 2 points.
276 194 5     // user 276 gives movie 194 5 points.
```

```
276 208 5     // user 276 gives movie 208 5 points.
276 585 1     // user 276 gives movie 585 1 point.
276 4 0       // need to predict user 276's rating for movie 4
276 26 0      // need to predict user 276's rating for movie 26
276 33 0      ...
276 56 0
276 63 0
276 67 0
276 72 0
```

ATTENTION: Please make the prediction block by block: every time when you are making predictions for user U, please assume that you **ONLY** know the knowledge of the training data (train.txt) and the existing 5 ratings for this user. In other words, please DO NOT use the knowledge of any other blocks in the test file when making predictions.

The format of test10.txt and test20.txt is nearly the same as test5.txt, the only difference is that: in test10.txt, 10 ratings are given for a specific user; in test20.txt, 20 ratings are given for a specific user.

How to get the accuracy?

To get the accuracy of your predictions, please submit the predicted ratings to our online grading system. (For more information, please check the help page of our grading system.).
You can access the grading system via submit.html.

Hint: You can do some cross validations to pretest the performance of your algorithm. (that is to split the training data into your own training data and your own test data...)

Tasks

Your task is to design and develop collaborative filtering algorithms that predict the unknown ratings in the test data by learning users' preference from the training data.

Please complete the following experiments:

1. User-Based Collaborative Filtering Algorithms (40 points)

1.1 Implement the basic user-based collaborative filtering algorithms

Please implement two versions of the basic user-based collaborative filtering algorithm as the Cosine similarity method and Pearson Correlation method.

1.2 Extensions to the basic user-based collaborative filtering algorithms

Please implement the following two modifications (individually and both) to the standard algorithm (using Pearson Correlation): 1. Inverse user frequency; 2. Case modification.

2. Item-Based Collaborative Filtering Algorithm (30 points)

Please implement the item-based collaborative filtering algorithm based on adjusted cosine similarity.

3. Implement your own algorithm (15 points)

You can implement your own algorithm to improve your recommendation performance. The grading will be based on the performance and novelty of the algorithm. Please describe your own algorithm in detail in the report.

4. Results Discussion (15 points)

Please provide the following information

Compare the accuracy of the various algorithms. Please include a table to show the Mean Absolute Error (MAE) of your different algorithms that you obtain from the online submission system. Do you think your results are reasonable? How can you justify the results by analyzing the advantages and disadvantages of the algorithms?

Bonus:

Results Competition (20 points)

20 points will be assigned according to the accuracy of your recommendation system. The best performance of all the algorithms from each student will be recorded and compared.

1: 20

2-3: 18

4-6: 16

7-10: 12

11-15: 10

16-18: 8

19-22: 6

23-25: 4

26-28: 2

29-30: 1

Rest: 0

What to turn in

Please submit the report of your experiments with the code to Camino. Please include some instructions about how to run your program. Please also give a hard copy of your report to the instructor right before the class on the second deadline.