Lab 6: Collaborative Filtering

CSC 466-01

Hanson Egbert ~ Dmitriy Timokhin

Methods Implemented

(NOTE: The order in this list corresponds to the order for the readme)

1. Mean Utility
2. Adjusted weighted sum
3. Adjusted weighted Nnn sum

All of these methods were covered in class and our implementation does not change them in anyway. Furthermore the Mean Utility was calculated as user-based approach and the other two methods were calculated using an item-based approach. Furthermore all of these methods were implemented using the pearson coefficient distance metric.

Research Question

In this lab we were curious to study which of our methods would have the smallest overall MAE and the smallest standard deviation of MAE's. In essence we were measuring our overall average error rate for predictions. The method which performs the best will have the smallest MAE and the smallest MAE standard deviation. It is critically important to observe the standard deviation because a method may have the same average MAE but may be predicting very good on some locations and very poorly on others, so we would like to have the method that may not do the best on some of the predictions but does a good job overall.

Experiments

To answer this question we will analyze a series of tests to see which method we believe does the best job of predicting values.

First Test

Here we will test each of the methods we implemented on a given list of points to Evaluate. The points were chosen at random but are meant to test the methods on predicting the values in columns 10 and 49 for different users. Note this first test does not all us to look at SD this will be accounted for in other tests.

1, 10 1, 49 2, 10 2, 49 3, 10 3, 49 4, 10 4, 49 1000, 10 1000, 49	5000, 10 5000, 49 7500, 10 7500, 49 10000, 10 10000, 49 15000, 10 15000, 49 20000, 10 20000, 49

This test is simply made to give us quick glance at the average performance of each of the methods. Note: N=7 for this test, for the Adjusted weighted Nnn sum.

Dima-the-penguin:Lab6CSC466 Dima$ python EvaluateCFList.py 1 testing_list.txt
Dima-the-penguin:Lab6CSC466 Dima$ python EvaluateCFList.py 2 testing_list.txt
Dima-the-penguin:Lab6CSC466 Dima$ python EvaluateCFList.py 3 testing_list.txt

Results

Mean Utility	MAE = 3.9603463302031647
Adjusted weighted sum	MAE = 2.7025239523826534
Adjusted weighted Nnn sum	MAE = 3.9368185940132476

It looks as though in this test that the Adjusted weighted sum performed the best by approximately a one point margin. This is interesting to consider because my original hypothesis was that the Nnn sum would perform the best since we are only looking at the closest N values. Nonetheless, this first test has a clear winner with Adjusted weighted sum.

Second Test

Now, since we have explored the methods' performance on values given to it we can explore how each of these methods performs on random points along the Matrix. To see this we will randomly generate 30 points and perform 8 tests for each of the methods described. We will compare the mean MAE and the sd of the MAE's and see which method performed the best. Note N = 7 once again.

Dima-the-penguin:Lab6CSC466 Dima$ python EvaluateCFRandom.py 1 30 8
Dima-the-penguin:Lab6CSC466 Dima$ python EvaluateCFRandom.py 2 30 8
Dima-the-penguin:Lab6CSC466 Dima$ python EvaluateCFRandom.py 3 30 8

Results

Mean Utility	Mean MAE = 4.19	SD MAE = 0.4916
Adjusted weighted sum	Mean MAE = 3.79	SD MAE = 0.587
Adjusted weighted Nnn sum	Mean MAE = 4.11	SD MAE = 0.22

These results prove quite interesting as the lowest MAE score for these tests belongs to the Adjusted weighted sum method but the lowest SD MAE belongs the Adjusted weighted Nnn sum method. This may be able to be explained by the fact that the closest N points have a similar distance to each-other for the N different items being considered to the prediction item.

It's quite curious to compare the two methods but I believe that which one you use will depend on which errors you are more comfortable with. Customers are more comfortable with consistently scoring slightly farther away but having a smaller range of predictions may prefer Adjusted weighted Nnn sum, but customers who prefer higher scores on average may prefer the Adjusted weighted sum. The mean utility does not perform as well as either of the other two metrics which shows that not accounting for similarity metrics between users or items will not be as accurate.

Third Test

After seeing the first two test we wanted to test the methods on very large groups of points in order to evaluate the methods better under conditions where you have a lot of data to predict. This will also allow the MAE's and SD MAE's to converge to their actual values for this dataset without having to run tests on every point.

To do this we will run 30 tests on 50 predictions each to evaluate each method.

Dima-the-penguin:Lab6CSC466 Dima$ python EvaluateCFRandom.py 1 50 30
Dima-the-penguin:Lab6CSC466 Dima$ python EvaluateCFRandom.py 2 50 30
Dima-the-penguin:Lab6CSC466 Dima$ python EvaluateCFRandom.py 3 50 30

Results

Mean Utility	Mean MAE = 4.133	SD MAE = 0.339
Adjusted weighted sum	Mean MAE = 3.451	SD MAE = 0.416
Adjusted weighted Nnn sum	Mean MAE = 3.98	SD MAE = 0.318

Once again Adjusted weighted sum performed the best on the data. Adjusted weighted Nnn sum though can be a close second though. Also, the SD MAE of the Nnn sum is the lowest as seen in the previous test. It is much closer though to the SD MAE for the mean utility which makes sense because Mean Utility should have a lower SD MAE when performing more tests since you are aggregating more data.

Full Results:

From our experiments, it appears that the adjusted weighted sum method performed the best overall by around .5 - 1 MAE better than the other methods.

Here are our results for adjusted weighted sum:

Test #1

Adjusted weighted sum	MAE = 2.7025239523826534

Test#2

Adjusted weighted sum	Mean MAE = 3.79	SD MAE = 0.587

Test #3

Adjusted weighted sum	Mean MAE = 3.451	SD MAE = 0.416

As you can see the adjusted weighted sum performed better than the other two methods for all of our tests, interestingly though it's MAE increased dramatically when the tests were changed to random ones. This may be a reason for more testing if given the time.

Furthermore the SD MAE is not unreasonably high which shows that this method is fairly consistent in its predictions of the data. It might be interesting to consider a different dataset and see which method may perform better. I personally believe there are many datasets in which the adjusted weighted Nnn sum will outperform the others if the matrix is less empty.