Week 2 – Prediction Testing

Filip Piskor

12331436

Implementation

This week we had to implement a function that would generate a mean item rating for each user rating by summing all the ratings for a given item except for the user rating and using that to predict what the user might rate the item. This function took in a userID and itemID. To calculate the mean rating I obtained all the ratings for a given item by querying my items HashMap which contained the itemID to item ratings. After obtaining the ratings I iterated over them and summed them up. After summing I subtracted the user rating and then returned the average of the sum. By storing all the ratings I feel I was able to speed up the calculation of the mean item rating. Each prediction was then evaluated by calculating the root mean squared error (rmse) between the actual and the predicted rating. By implementing a simple test loop that went through each users items and prediciting their items we were able to get an average of all the rmses and see how accurate were the generated predictions. All of the predicitions were saved to a predictions.csv file. I ran the test 10 times and got an average of the time it took to run 1 test which was on average ~900 ms.

Findings

I have found that on average the error is ~0.81453 (~16%). While this seems not to be very significant is indeed quite since if we were to guess randomly we'd have a 20% chance of getting it right. I would expect a generated prediction to be more accurate at guessing what the user might think of a given item.
I have also found that the coverage is quite substantial being 99.859% in a sample of 100,000. This means that there are very few users who were the only one to rate a specific item. Due to this coverage we were able to predict most of the items in the dataset.

Can you use this technique to predict a rating for every existing user-item pair?

No you can't since some users are the only ones that have rated the movie. If we exclude them we have no other ratings to get an average of and therefore get a NaN as the answer.