Ellen Walsh

Task 1 – Read Prediction

For the read prediction task, I first implemented popularity mixed with Jaccard similarity, from question 5 in homework 3. I then tried to implement only popularity, which gave the lowest score of all my attempts, only an accuracy of 50%. My third method was to implement only Jaccard similarity, which gave a lower accuracy than the first prediction method. I tweaked my first model by changing the threshold. The threshold that yielded the highest accuracy on the validation set was 0.6. When I reimplemented the model with a threshold of 0.6, the accuracy slightly improved and this model showed the best accuracy on Kaggle. When the unseen portion of the test set was revealed, this model did not perform as well, and my leaderboard ranking dropped a couple spots.

The methodology for combined popularity and Jaccard first sorted through the training data and ranked each book's popularity based on the number of readers it had. This trained the data before implementing the model on the pairs_read.txt file. The most popular books and their user pairs were compared at a threshold of 0.6. Next, for all books in the mostpopular list, the Jaccard similarity was computed of the two vectors. The Jaccard similarity was implemented by creating a function that divided the intersection of the two vectors by the union of the vectors. This was a rational approach for this prediction task because I was comparing two binary vectors, which in this case were sets. I did not choose to implement cosine similarity because there was no rating data. I would have used cosine similarity if the books that were read also had a thumbs up or a thumbs down to indicate the reader's satisfaction. Most popular was not sufficient alone because the reader's preferences were not taken into account, only majority preferences.

Task 2 – Rating Prediction

For the rating prediction, all the models I implemented followed a latent factor model, because I needed to account for bias terms under the readers and the books. The first model I uploaded was not regularized, and it generated the worst accuracy on Kaggle of all my submissions. This first model came from question 9 of homework 3. The second model I uploaded regularized with a value of 0.00009, because that value of lambda resulted in the best accuracy on my validation set. The accuracy on Kaggle was not much better than the unregularized model. I then began to play with different values of lambda, and the value which generated the lowest MSE for the validation set was 0.00001. When I uploaded this prediction on Kaggle, it was better than my previous models.

To implement the latent factor model, I computed the average rating of the training set and set this as the value for alpha. I then defined the unpack, cost, prediction, derivative, and gradient descent functions as the Professor did in workbook 4, to determine the bias terms for each reader and each book. I ran this trained model on the validation set. I then created a loop of different lambda values and performed gradient descent. I then used the entire dataset to train the model and get more accurate bias terms and set alpha to the average rating for all 200000 datapoints. I ran this model on the pairs_prediction.txt file with the lambda of 0.00001.