Ian Battin and Derek Lance
November 30, 2018

Lab 6: Collaborative Filtering

Methods:

1. Mean Utility
2. KNN Mean Utility w/ Cosine Similarity
3. KNN Mean Utility w/ Pearson Similarity
4. KNN Adjusted Weighted Sum

Research Question/Goal:

Given a rating's matrix of jokes, try and predict the rating a given user would give to a given joke in order to recommend jokes that they have not seen before. This is done slightly different by each of the four methods but the general idea is to use the ratings of other user's by either ranking them on similar joke "tastes" or taking the joke's average overall rating's times the user's average rating.

Experiments Description:

Mean utility simply averages all of the ratings for a given joke and returns that as the prediction score. Obviously, this is very primitive and does not take into account a user's rating habits or tastes. This results in around a 74% accuracy - however, this is equal to not recommended any jokes at all (which is what our tests did). This makes sense as the average rating for a joke is very unlikely to rate above a 5 which is what's required to be recommended. What can be learned from this though is that though is that 74% of jokes would not be recommended to the average user.

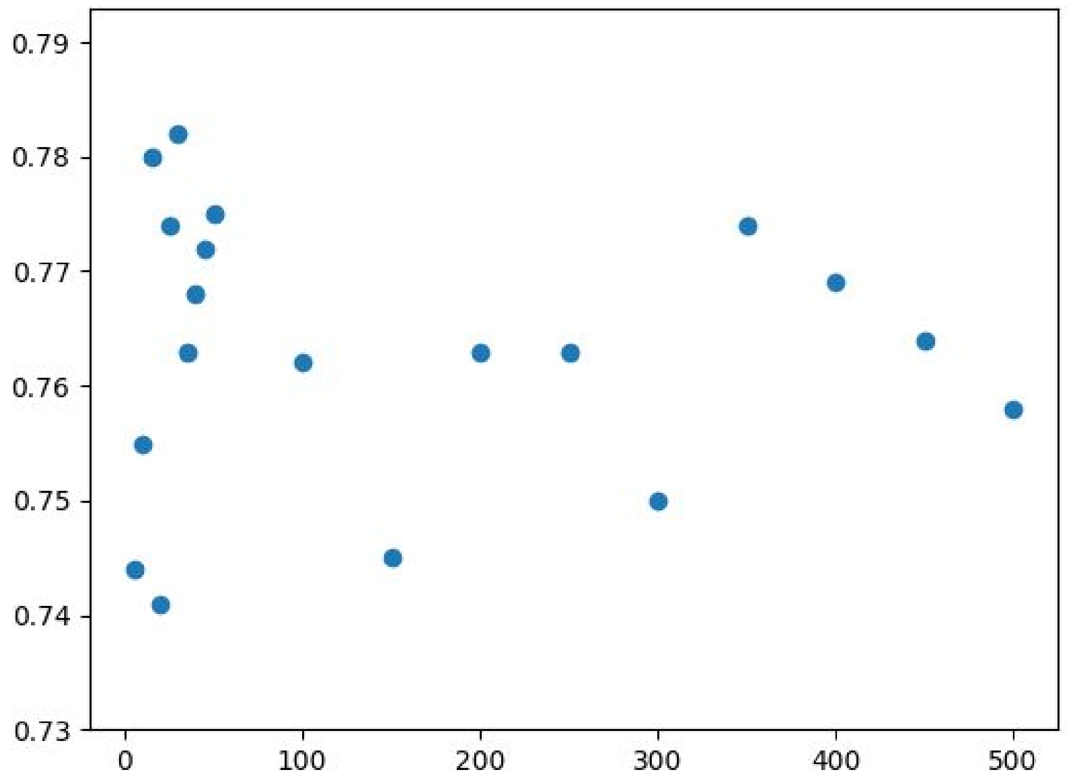
KNN Mean Utility w/ Cosine similarity and w/ Pearson similarity are both modified versions of the original mean utility method but instead of taking the average for every user in the dataset, they calculate the N most similar rating vectors to the given user, essentially finding other users with similar tastes. They then average these ratings. This gives a slight boost in accuracy, moving up to 76-77% and actually recommending jokes.

KNN Adjusted Weighted Sum is the most "tuned" method given to us in the text. It calculates the N most similar users and finds the average rating from this group, but also weights each sum by the similarity score, as well as the given user's average rating. In essence, each data point used in the prediction is weighted by how similar the user's tastes are as well as how harshly or lenient the user rates jokes in general. This resulted in an overall accuracy of around 79%, just slightly higher than the previous methods. This seems to say that the majority of the accuracy gain comes from similarity scores.

Tuning:

For mean utility, there's no tuning required.

For the KNN methods, we created a script to cycle through various values of N using Mean Utility w/ Cosine similarity to try and determine which values of N gave the best results.



As you can see, lower values seem to give better predictions than higher values (as shown by the upward trend from 15 - 45 and the downward trend starting at 350). This makes sense as you want enough variation to have better averages, but not so many that your KNN group includes users who have much different tastes.

As far as size and repeats, we tuned these manually but it's pretty easy to see that the bigger size and repeats are, the more accurate your predictions. Size increases the accuracies of individual test runs while repeats averages out the variability in random selection across test runs. We tested using 200 for size and 5 for repeats. We believe that given a high size, our results would be more accurate. However, for the sake of runtime, it grows extremely quick because of having to do similarity comparisons. If we could precompute our cosine/Pearson similarity matrix for all 27,000 users, this would likely speed this up but that would have taken days to calculate.

Results

Mean Utility	KNN Mean Utility Cosine	KNN Mean Utility Pearson	KNN Adjusted Weighted Sum Cosine
Mean MAE: 4.08 Standard Deviation MAE: 0.22 Mean Accuracy: 0.747	Mean MAE: 3.35 Standard Deviation MAE: 0.21 Mean Accuracy: 0.76	Mean MAE: 3.6 Standard Deviation MAE: 0.22 Mean Accuracy: 0.747	Mean MAE: 3.37 Standard Deviation MAE: 0.31 Mean Accuracy: 0.782

Full output of every test run is included for each method in our submission files in the outputs folder.

Conclusion:

Overall, we believe the best method of the four we chose to be KNN Adjusted Weighted Sum with its 79% accuracy. This makes sense as it takes the most variable/weights into consideration. It compares tastes of users and uses this to weight every sum value, as well as the individual rating tendencies.