COEN 169 Project2 Report

Zheqing Li

Error Rate result for different algorithms

|  |  |
| --- | --- |
| Algorithm | MSE |
| User- based Cosine similarity | 0.793055327532425 |
| User- based Pearson Correlation | 0.817394516499754 |
| User-based Pearson Correlation –  Case modification | 0.836890494171729 |
| User- based Pearson Correlation – Inverse User Frequency | 0.856058118535544 |
| Item- based Adjusted Cosine Similarity | 0.819857166310951 |
| Custom1 | 0.895706780495814 |
| Custom2 | 0.792973239205385 |

In this project, the user based cosine similarity algorithm gives the best result, and the IUF Pearson Correlation algorithm performs worst. However, conceptually, the Pearson correlation algorithms can provide lower MSE compared with cosine similarity. The user-based algorithm has a little bit better performance than item- based algorithm. For comparison purpose, all the algorithm tests use 50 neighbors for calculating the predicted rating.

The user- based cosine similarity is beneficial to abstract out the magnitude of the term vector, which takes out the influence of the document length. One of the drawbacks of this algorithm is that the different in rating scale between users are not taken into account, so the user average rating will affect the similarity. Since the cosine similarity is the most naïve algorithm tested in this project, it runs the fastest.

The user- based Pearson Correlation algorithm gives a little bit higher error rates than cosine similarity. The Pearson Correlation is like adjusted cosine similarity, which eliminates the influence of the user mean rating. Sine the cosine similarity is not fully decided only by the subset of items the two users have in common, it is not natively present in Pearson correlation. Therefore, in contexts where users tend to have very different sets of items, Pearson will perform worse.

The case modification of user- based Pearson Correlation algorithm performs worse than Pearson Correlation algorithm. It refers to a transform applied to the weights used in the prediction. This transform emphasizes high weights and punished low weights. However, it is hard to put on the most appropriate rho value in case amplification. Larger pho value will emphasizes and punishes more, sometimes it will make the predicting less accurate depending on the similar neighbors.

The IUF of user based Pearson correlation algorithm performs the worst in this test. The purpose of this algorithm is to punish universally rated movies in capturing similarity as less common movies by using IUF, based on Pearson Correlation. The IUF is used to prevent popular movies worsen the quality of the recommendations, since they are less discriminative across different users. However, this algorithm will reduce the weight of all popular movies. If the popular movie matches the users’ interest, its weight will still be reduced. Then the rating will be less accurate.

The Item based adjusted cosine similarity runs very slow since it need to find the similarity of 1000 movies instead of 200 users. The idea of this algorithm is to find similarly rated movies instead of finding similar users’ ratings. The computation is basically similar to user based, but it looks into co-rated items only. When the system has many users, it will be faster to use item based. In this project, since there are more movies, it is much faster to use user- based algorithm. Furthermore, similarity estimates between items are more likely to converge over time than similarity between users. Moreover, sometimes the item based similarity might result in totally dissimilar product from the one user is interested in. Some items with high similarity may not actually be similar products. Overall, this algorithm gives good result, since it subtract the corresponding user average from each co-rated pair.

The custom1 algorithm is similar to Pearson Correlation algorithm. It subtracts median instead of average when calculating similarity. If the system has positive and negative, it will works much better, since it considers the impact of positive and negative ratings. However, in this project, there are no negative ratings in data set, and then it is reasonable that the custom algorithm behaves worse than Pearson Correlation algorithm.

Since Cosine similarity performs pretty well in the test, I modified the average. When there is no similar user found, it will call item based algorithm to find the predicted rating. Thus, this algorithm is mainly user based, but item based is used to help to handle the special cases, such as not similar user found. Therefore, the result behaves little bit better than Cosine similarity.

Overall, in this project, the cosine similarity works the best. In general, the error rate of different algorithm depends on different features of the testing data set. In real life, it is important to consider the specialty of the data set.