Applied Intelligence

Recommender Challenge Project Report

Wenchong Chen

1 Improving Collaborative Filtering

Generally, a user-based collaborative recommender system can be improved in three ways. The improvement approaches are based on the process of how a recommender system is built.

1.1 Improving Similarity Measurement

The first step of building a recommender system is to compute the similarity between a pair of users. The base code provides the Mean Square Difference measurement, given by:

$$diff(ui, uj) \; = \frac{\sum_{\forall itemk \in corated\ (ui,uj)} (r(ui, itemk) - r(uj, itemk))^2}{|corated(ui, uj)|}$$

The similarity, also known as weight$_j$, is then calculated by:

$$\mathrm{sim(ui, uj)} = 1 - \frac{diff(ui, uj)}{MaxDiff}$$

The most commonly used measurement, which is also the most efficient one among all measurements mentioned in this report, is Pearson's Correlation Coefficient. The similarity is calculated by:

$$\mathrm{sim(ui, uj)} = \frac{\sum_{\forall itemk \in corated\ (ui,uj)} r(ui, itemk) \times r(uj, itemk)}{\sqrt{\sum_{\forall itemk \in corated\ (ui,uj)} (r(ui, itemk) - \bar{r}(ui))^2} \sqrt{\sum_{\forall itemk \in corated\ (ui,uj)} (r(uj, itemk) - \bar{r}(ui))^2}}$$

Spearman's Rank Correlation is also implemented. Based on Pearson's Correlation Coefficient, it focuses on the rankings of ratings, ignoring the values of them. In a non-linear situation with numerous ties, it does not perform better than Pearson's Correlation Coefficient. The formula of Spearman's Rank Correlation with numerous ties is given by:

$$\rho = \frac{\sum_{\forall itemk \in corated\ (ui,uj)} ranking(ui,itemk) \times ranking(uj,itemk) - n(\frac{n+1}{2})^2}{\sqrt{\sum_{i=1}^{n} ranking(ui,itemk)^2 - n(\frac{n+1}{2})^2} \sqrt{\sum_{i=1}^{n} ranking(uj,itemk)^2 - n(\frac{n+1}{2})^2}}$$

where n is the number of corated items.

As an alternative, ratings of a user pair can be treated as a vector. Cosine Similarity then calculates the cosine of the angle between them as follows:

$$\mathrm{cos(ui, uj)} = \frac{\sum_{\forall itemk \in corated\ (ui,uj)} r(ui, itemk) \times r(uj, itemk)}{\sqrt{\sum_{\forall itemk \in ui} r(ui, itemk)^2} \sqrt{\sum_{\forall itemk \in uj} r(uj, itemk)^2}}$$

1.2 Improving Neighbourhood Policy

Two most common approaches to select a neighbourhood are to select a fixed size or similarity bounded one.

A fixed size neighbourhood contains top N most similar users, while users selected for a similarity bounded neighbourhood exceed a fixed similarity threshold.

This assignment applied a fixed size neighbourhood. Statistic used to test the efficiency of similarity measurements and prediction algorithms were top 300, 100 and 50 users.

1.3 Improving Prediction Computation

To predict recommendations for a user, the system needs to predict ratings for items that user has not rated. The prediction algorithm in the base code calculates ratings by the following formula:

$$pre(\mathrm{ui}, \mathrm{itemk}) = \frac{\sum_{\forall uj\ \in Neighbourhood\ (ui) \cap r(uj,itemk) \neq 0} sim(ui,uj) \times r(uj,itemk)}{\sum_{\forall uj\ \in Neighbourhood\ (ui) \cap r(uj,itemk) \neq 0} |sim(ui,uj)|}$$

An improved prediction computation algorithm is based on Resnick's Formula given as below:

$$\mathrm{pre}(\mathrm{ui}, \mathrm{itemk}) = \bar{r}(uj) + \frac{\sum_{\forall uj\ \in Neighbourhood\ (ui) \cap r(uj,itemk) \neq 0} sim(ui,uj) \times (r(uj,itemk) - \bar{r}(uj))}{\sum_{\forall uj\ \in Neighbourhood\ (ui) \cap r(uj,itemk) \neq 0} |sim(ui,uj)|}$$

2 Evaluating Collaborative Filtering

The first step of evaluating collaborative filtering is to split the real data into two parts. One is called training data that is used to predict ratings for those taken away from the original ratings matrix. The other is test data which is used to evaluate the how close the predictions are compared to test data.

Suppose we have a ratings data matrix showed in Figure 1.

Figure 1 Ratings Data Matrix

	Item1	Item2	Item3	Item4	Item5
User1	3.5	3.8	4.6		5.0
User2	2.7	3.4	2.9	4.8	3.2
User3		4.6	2.3	4.5	3.9
User4		3.5	4.5	3.0	4.7
User5	2.8	4.2	3.7	4.9	

The ratings data matrix is split to training data (see Figure 2) and test data (see Figure 3). The data in the blocks marked blue are taken away from trainings data.

Figure 2 Trainings Data

	Item1	Item2	Item3	Item4	Item5
User1	🟦	3.8	4.6		5.0
User2	2.7	🟦	2.9	🟦	3.2
User3		4.6	2.3	4.5	3.9
User4		3.5	🟦	3.0	4.7
User5	2.8	4.2	3.7	4.9	

In test data matrix, data in blue blocks are kept and other data are taken away.

Figure 3 Test Data

	Item1	Item2	Item3	Item4	Item5
User1	3.5				
User2		3.4		4.8	
User3					
User4			4.5		
User5					

Based on the training data, ratings are predicted for each user-item pair where data is sufficient based on other users' ratings. Assume that the ratings predicted are marked red in Figure 4 and that the blocks left blank means the ratings for them can't be predicted.

Figure 4 Predictions

	Item1	Item2	Item3	Item4	Item5
User1	3.2	3.8	4.6	4.7	5.0
User2	2.7	3.1	2.9	2.5	3.2
User3	2.9	4.6	2.3	4.5	3.9
User4		3.5	2.5	3.0	4.7
User5	2.8	4.2	3.7	4.9	

The ratings in blue blocks will then be compared to those in test data respectively so as to evaluate the MSE and coverage of items that can't be rated.

3 Conclusion

In order to find out which similarity measurement performs better, control variate method is applied by using a fixed size neighbourhood (top 300, 100 and 50 most similar users) across all the implementations of similarity computation. At the same time, the two algorithms of prediction computation are also controlled as variates.

Test results shows that, when neighbourhood and prediction computation stayed the same, while both Pearson's Correlation Coefficient and Spearman's Rank Correlation were much better than the other similarity measurements, Pearson's Correlation Coefficient was slightly better than Spearman's Rank Correlation.

While the two other factors were kept the same, the bigger the size of neighbourhood was, the bigger the coverage was, but the lower the MSE might be.

Keeping neighbourhood and similarity measurement the same, it showed that Resnick's Formula was better than the prediction computation given by the base code.

In conclusion, the combination of Pearson's Correlation Coefficient and Resnick's Formula with a fixed top 200 neighbours performs better than the other combinations.