Web Search & Information Retrieval

Project 2

Xinrui Wu
Mar. 8 2018

Overview

In this project, a python application was developed to predict users' taste on movies based on their previous rates assigned to other movies. Four types of algorithms were adopted including user-based collaborative filtering algorithm using Cosine similarity, user-based collaborative filtering algorithm using Pearson Correlation, item-based collaborative filtering algorithm using Cosine similarity, and item-based collaborative filtering algorithm using Pearson Correlation. Each algorithm was also tested with different values of Inverse User Frequency and Case Amplification to improve the performance. Due to the lack of validation dataset, the program sampled 20 users (1000 movies each) randomly from the training dataset to form a validation dataset and a new training dataset without those 20 users to test the algorithm, root-mean-square error (RMSE) is used to evaluate the performance. Total 640 combinations of different values of parameters were test and each combination yielded a score reflecting the performance of the algorithm with those values of parameters. **All the scores listed in this report were produced by local python code instead of online test due to the limitation on numbers of submissions (30 times) so the values present in this report only reflect patterns for comparison purpose.**

After the comparison, an improved algorithm was developed to try to further improve the accuracy of the prediction. **These customized algorithm will be tested by online judgement and the score will be presented in this report additionally.**

1. User-Based Collaborative Filtering Algorithms

- Cosine Similarity

The result provided by user-based collaborative filtering algorithm with Cosine similarity is listed in Table 1. Similarity threshold denotes to the ratio of the minimum similarity to the maximum similarity of the included neighbor. Larger similarity threshold means more restrict on similarity and fewer neighbors are selected to make the prediction. Similarity threshold equals to 1 means only neighbors with the maximum similarity are selected and 0 means all the neighbors are selected. The table shows that as the similarity threshold grows, the prediction error is fluctuating but still relatively stable, however, when the similarity threshold goes beyond 0.8, the error decrease greatly first then increase to a very high value. This phenomenon shows that the similarities are centralized to 80%~100% of the maximum similarity, so increasing similarity threshold from 0 to 80% does not obviously influent the result, but from 80% to 85% can help remove dissimilar neighbors so as to refine the prediction. When the similarity threshold is greater than 0.9, too few neighbors are included to provide reasonable result.

Table 1. Prediction Error of User-Based Collaborative Filtering Algorithms with Cosine similarity

Similarity Threshold	<=0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0
Test-5 Error	1.12789	1.12789	1.12723	1.12723	1.12723	1.12833	1.12458	1.12347	1.15986	1.55048
Test-10 Error	1.11955	1.11932	1.11979	1.12096	1.12119	1.12119	1.11322	1.11675	1.15160	1.75547
Test-20 Error	1.12078	1.12078	1.12104	1.12052	1.12052	1.11973	1.11843	1.12078	1.18741	2.13711
Overall Error	1.12274	1.12266	1.12269	1.12290	1.12298	1.12309	1.11874	1.12033	1.16629	1.81435

- Pearson Correlation

The result provided by user-based collaborative filtering algorithm with Pearson Similarity is listed in Table 1. The table shows that as the similarity threshold grows, the prediction error is growing as well, so unlike Cosine similarity, Pearson correlation relies on a broad range of neighbors, and the neighbors with lower vote weight are actually helping improve that prediction. This phenomenon reflates the nature of Pearson correlation that it highly emphasizes on patterns and eliminates some unpredictable fluctuations. This nature may make Pearson Correlation's performance closely relate to dataset, so including larger range of neighbors is a way to improve the accuracy of Pearson correlation method. One should notice that for given dataset, Person correlation is not as accurate as Cosine similarity.

Table 2. Prediction Error of User-Based Collaborative Filtering Algorithms with Pearson Correlation on Various Neighbor Radius

Similarity Threshold	0	0.1	0.2	0.3	0.4	0.5	0.55	0.6
Test-5 Error	1.17583	1.17626	1.17520	1.18195	1.18048	1.17626	1.17943	1.18280
Test-10 Error	1.10021	1.10045	1.10021	1.10235	1.11416	1.13026	1.12515	1.14293
Test-20 Error	1.07904	1.08175	1.08687	1.09731	1.11895	1.14988	1.16453	1.17777
Overall Error	1.11836	1.11949	1.12076	1.12721	1.13786	1.15213	1.15637	1.16783
Similarity Threshold	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0
Test-5 Error	1.18678	1.18532	1.19597	1.20075	1.20900	1.21802	1.22169	1.22799
Test-10 Error	1.15840	1.16606	1.18895	1.20122	1.20644	1.20731	1.21660	1.20340
Test-20 Error	1.18913	1.19477	1.19526	1.20234	1.20598	1.19477	1.18962	1.13812
Overall Error	1.17810	1.18205	1.19340	1.20143	1.20714	1.20670	1.20930	1.18984

2. Modification on User-Based Collaborative Filtering Algorithms

- Inverse User Frequency

Inverse user frequency are used to decrease the weight of movies that are universally rated by users, and to increase the weight of movies that are less commonly rated. Table 3 and Table 4 shows the effect of IUF on user-based collaborative filtering algorithms with Cosine similarity and Pearson Correlation. The result turns out that in these two tests, generally IUF did not yield a better accuracy. Probably the reason is that the training dataset is not large enough to distinguish universally rated movies from less commonly rated movies, enforcedly applying IUF on small dataset may lead to larger prediction error.

Table 3. Prediction Error of User-Based Collaborative Filtering Algorithms with Cosine Similarity w/ and w/o IUF

Similarity Threshold	<=0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0
With IUF Error	1.12302	1.12294	1.12376	1.12376	1.12337	1.12217	1.11915	1.12209	1.14057	2.43828
Without IUF Error	1.12274	1.12266	1.12269	1.12290	1.12298	1.12309	1.11874	1.12033	1.16629	1.81435

Table 4. Prediction Error of User-Based Collaborative Filtering Algorithms with Pearson Correlation w/o IUF

Similarity Threshold	0	0.1	0.2	0.3	0.4	0.5	0.55	0.6
With IUF Error	1.12883	1.12916	1.13196	1.13671	1.14622	1.15398	1.15617	1.16286
Without IUF Error	1.11836	1.11949	1.12076	1.12721	1.13786	1.15213	1.15637	1.16783
Similarity Threshold	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0
With IUF Error	1.17439	1.18172	1.19388	1.20143	1.21129	1.21413	1.21849	1.18873
Without IUF Error	1.17810	1.18205	1.19340	1.20143	1.20714	1.20670	1.20930	1.18984

- Case Amplification

Case amplification is a method to emphasizes high weights and punishes low weights. Table 5 shows effect of case amplification on both Cosine similarity and Pearson correlation with similarity threshold of 0. From the table it can be observed that the fluctuation is tiny. For Cosine similarity, amplification power of 1.5 yields the worst error and all the results with amplification power of 2 to 3 are better than the result without case amplification. However, for Pearson correlation, amplification power of 1.5 yields the lowest error and all the results with amplification power of 2 to 3 are worse than the result without case amplification, so it can be concluded that case amplification do improves prediction but the value should be chosen carefully. It enlarges the weight of close neighbors but it is not guaranteed that these neighbors always have positive effect on the prediction.

Table 5. Prediction Error of User-Based Collaborative Filtering Algorithms with Case Amplification

Case Amplification	1	1.5	2	2.5	3
Cosine Similarity	1.12274	1.12287	1.12251	1.12215	1.12132
Pearson Correlation	1.12302	1.12294	1.12376	1.12376	1.12337

3. Item-Based Collaborative Filtering Algorithms

- Cosine Similarity

The result provided by item-based collaborative filtering algorithm with Cosine similarity is listed in Table 6. The table shows that as the similarity threshold grows, the prediction error is growing as well, so unlike the user-based Cosine similarity algorithm, item-based Cosine similarity algorithm relies on a broad range of neighbors, and the neighbors with lower vote weight are actually helping improve the prediction. One should notice that comparing with the user-based Cosine similarity algorithm, item-based Cosine similarity algorithm performs worse on Test-5 but better on Test-10 and Test-20, this result is reasonable because for Test-5, user-based algorithm predicts based on 200 users however item-based algorithm only relies on 5 input, as input size grows, the item-based algorithm is performing better than the user-based one.

Table 6. Prediction Error of Item-Based Collaborative Filtering Algorithms with Cosine Similarity on Various Neighbor Radius

Similarity Threshold	<=0.4	0.4~0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1
Test-5 Error	1.16947	1.16883	1.16990	1.16990	1.17011	1.17181	1.17351	1.18006	1.17985	1.18867	1.30898
Test-10 Error	1.09497	1.09497	1.09497	1.09449	1.09712	1.09831	1.09950	1.10354	1.11322	1.12329	1.33303
Test-20 Error	1.07333	1.07333	1.07333	1.07306	1.07388	1.07388	1.07470	1.07958	1.08012	1.09170	1.32431
Overall Error	1.11259	1.11238	1.11273	1.11248	1.11370	1.11467	1.11590	1.12106	1.12440	1.13455	1.32211

- Adjusted Cosine Similarity

The result provided by item-based collaborative filtering algorithm with adjusted Cosine similarity is listed in Table 7. The table shows that adjusted Cosine similarity shares the same pattern with raw Cosine similarity. One should notice that adjusted Cosine similarity perform worse on Test-5 than raw Cosine but better on Test-10 and Test-20 when similarity threshold equals to 0, as similarity threshold increases, the gap is shrinking but with different speed, adjusted Cosine similarity get worse on Test-10 when similarity threshold equals to 0.3, and on Test-20 when similarity threshold equals to 0.5. It can be concluded that adjusted Cosine similarity is an ideal candidate when the data is abundant.

Table 7. Prediction Error of Item-Based Collaborative Filtering Algorithms with Adjusted Cosine Similarity on Various Neighbor Radius

Similarity Threshold	0	0.1	0.2	0.3	0.4	0.5	0.55	0.6
Test-5 Error	1.21291	1.21352	1.21475	1.21700	1.22515	1.23022	1.23244	1.24009
Test-10 Error	1.08125	1.08319	1.08873	1.10544	1.11979	1.14201	1.15772	1.16313
Test-20 Error	1.05548	1.05742	1.05797	1.06211	1.06073	1.07633	1.09251	1.11581
Overall Error	1.11655	1.11804	1.12048	1.12818	1.13522	1.14952	1.16089	1.17301

Similarity Threshold	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0
Test-5 Error	1.24649	1.25425	1.26747	1.27158	1.28810	1.29619	1.29925	1.31164
Test-10 Error	1.17031	1.18432	1.19510	1.21120	1.22581	1.24762	1.26636	1.28849
Test-20 Error	1.12676	1.14274	1.16654	1.19453	1.22594	1.25050	1.28031	1.30742
Overall Error	1.18119	1.19377	1.20970	1.22577	1.24662	1.26477	1.28197	1.30252

4. Modification on Item-Based Collaborative Filtering Algorithms

- Inverse User Frequency

Table 6 and Table 7 shows the effect of IUF on item-based collaborative filtering algorithms with Cosine similarity and adjusted Cosine similarity. The result turns out that raw Cosine similarity algorithm's prediction is refined a little bit with IUF while adjusted Cosine similarity algorithm's prediction drops obviously. It probably can be explained by previous discussion: the training dataset is not large enough to distinguish universally rated movies from less commonly rated movies, enforcedly applying IUF on small dataset may not have too much effect.

Table 6. Prediction Error of Item-Based Collaborative Filtering Algorithms with Cosine Similarity w/ and w/o IUF

Similarity Threshold	<=0.4	0.4~0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1
With IUF	1.11222	1.11222	1.11229	1.11282	1.11236	1.11324	1.11566	1.11840	1.12318	1.13563	1.33928
Without IUF	1.11259	1.11238	1.11273	1.11248	1.11370	1.11467	1.11590	1.12106	1.12440	1.13455	1.32211

Table 7. Prediction Error of Item-Based Collaborative Filtering Algorithms with Adjusted Cosine Similarity w/ and w/o IUF

Similarity Threshold	0	0.1	0.2	0.3	0.4	0.5	0.55	0.6
With IUF	1.12345	1.12487	1.12851	1.13310	1.14374	1.15806	1.16373	1.17154
Without IUF	1.11655	1.11804	1.12048	1.12818	1.13522	1.14952	1.16089	1.17301
Similarity Threshold	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0
With IUF	1.18967	1.20158	1.20755	1.23073	1.24700	1.26058	1.28006	1.32288

Without IUF	1.18119	1.19377	1.20970	1.22577	1.24662	1.26477	1.28197	1.30252

- Case Amplification

Table 8 shows effect of case amplification on both raw and adjusted Cosine similarity with similarity threshold of 0. From the table it can be observed that the errors is getting worse as the amplification power increases. This result is identical to what has been discussed before that item-based algorithms rely on a broad range of neighbors, blindly increasing weights of close neighbors and decreasing weights of far neighbors does not effect the prediction positively.

Table 8. Prediction Error of Item-Based Collaborative Filtering Algorithms with Case Amplification

Case Amplification	**1**	**1.5**	**2**	**2.5**	**3**
Cosine Similarity	1.11259	1.11288	1.11386	1.11511	1.11554
Adjusted Cosine Similarity	1.11655	1.12777	1.13743	1.15214	1.16559

5. Result Comparison

- Accuracy

Figure 1. Prediction Error Plot
(1 = User-Based Cosine, 2 = User-Based Pearson, 3 = Item-Based Cosine, 4 = Item-Based Adjusted Cosine)

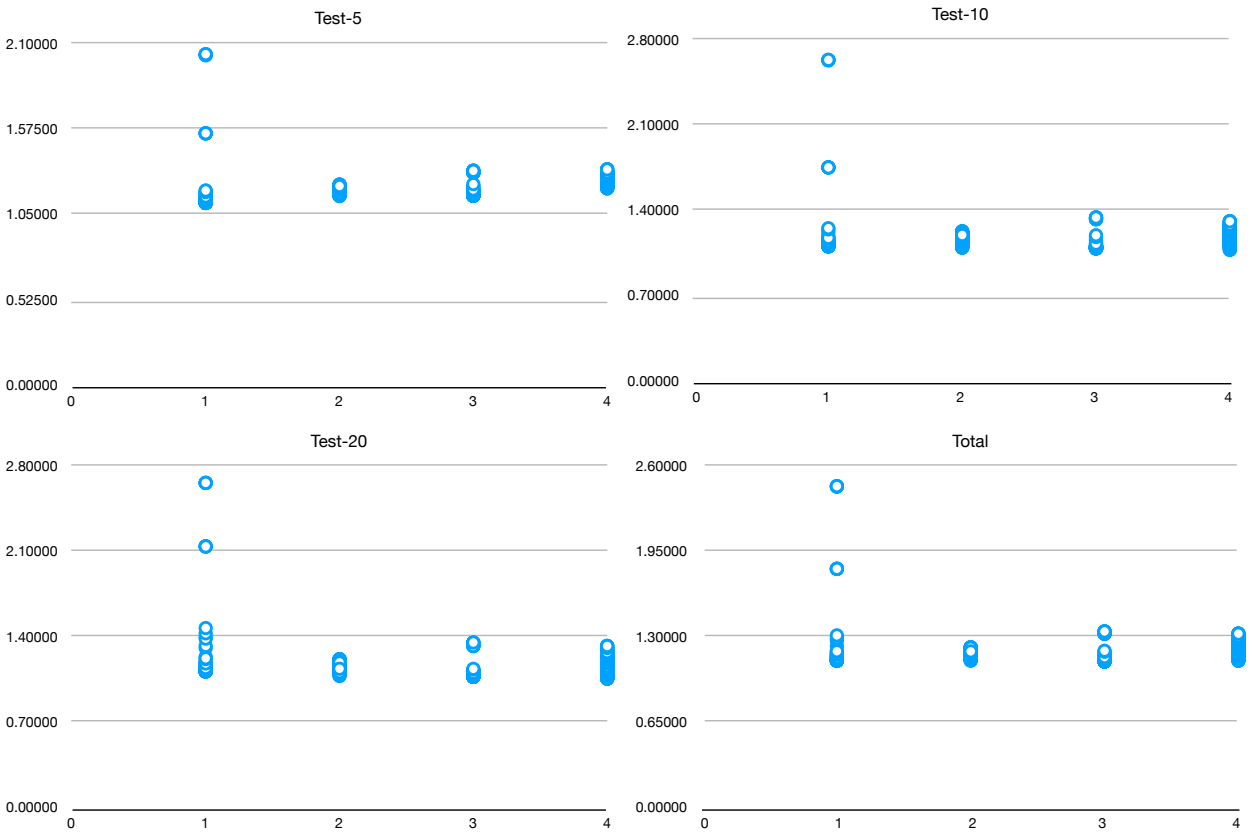

Figure 1 shows a overview of prediction error of four presented algorithms. Total 640 combinations of different values of parameters were test and each combination yielded a set of score reflecting the performance of the algorithm with those values of parameters. One could notice that user-based algorithms work well for insufficient input (Test-5), while as the input grows (Test-10, Test-20), item-based algorithms will get better than user-based ones. Considering the overall performance, all four algorithms yield accuracy approximately in the same level, however, user-based Pearson correlation algorithm is the stablest one that is not much sensitive to parameters such as numbers of neighbors, IUF, and case Amplification, while other three algorithms' accuracy vary a lot when parameters change. The top result of four algorithms is present in table 9.

Table 9. Lowest Prediction Error of All Tested Algorithms

Method	Neighbor	Radius	Amplify	IUF	Test-5	Test-10	Test-20	Total
User-Based Cosine	0.6		3	False	1.12414	1.11087	1.11476	1.11659
User-Based Pearson	0		1	False	1.17583	1.10021	1.07904	1.11836
Item-Based Cosine	0		1	True	1.16883	1.09449	1.07333	1.11222
Item-Based Adjusted Cosine	0		1	False	1.21291	1.08125	1.05548	1.11655

- Time Efficiency

The running time of four presented algorithms along with the customized algorithm about to be introduced next is listed in Table 10. It shows that user-based algorithms are sharing the same running time level, which is also true for item-based algorithms. Noticing that the dataset contains items five times to users, so user-based algorithms run faster than item-based ones. The situation would be totally opposite if the dataset contains more users than items. Customized algorithm is one that be developed for further improvement on prediction, it yields the best accuracy (to be discussed in Section. 6) however it is the slowest algorithms, the running time is approximately the sum of that of previous four algorithms.

Table 10. Running Time of All Tested Algorithms (in million seconds)

User-Based Cosine	User-Based Pearson	Item-Based Cosine	Item-Based Adjusted Cosine	Customized Algorithm
100764	99520	1007188	1023767	2210725

6. Customized Algorithm

I. Deal with empty weight set

The presenting of empty weight set is reasonable because there is a chance that no one has rated the same movie with the target user, or anyone who has rated the same movie with the target user

has not rated the target movie. Previous algorithms with Cosine similarity have no ability to deal with this situation and consequently assign 0 to those movies. To solve this problem, an average calculation function is added and it will assign an average of known ratings to target movie. The refined result is presented in Table 11.

Table 11. Effect of Average Value Assignment on Prediction Error

Method	Test-5	Test-10	Test-20	Total
User-Based Cosine	1.12414	1.11087	1.11476	1.11659
User-Based Cosine (refined)	1.11011	1.09329	1.09518	1.09953
Item-Based Cosine	1.16883	1.09449	1.07333	1.11222
Item-Based Cosine (refined)	1.15535	1.07664	1.05298	1.09499

II. Method Combination

As discussed before, each of these four algorithms has its own strength and environment in which it yields good accuracy. It is a reasonable guess that combining these four algorithms could potentially refine the prediction accuracy, an algorithm is designed that takes the average of those four algorithms' prediction with each one using their best parameter combination, the result is presented in Table 12. It shows that the error decreases greatly for all three tests plus the final result, the accuracy improved by 7%.

Table 12. Comparison among Customized Algorithm with Standard Collaborative Filtering Algorithms

Method	Neighbor	Radius	Amplify	IUF	Test-5	Test-10	Test-20	Total
User-Based Cosine	0.6		3	False	1.12414	1.11087	1.11476	1.11659
User-Based Pearson	0		1	False	1.17583	1.10021	1.07904	1.11836
Item-Based Cosine	0		1	True	1.16883	1.09449	1.07333	1.11222
Item-Based Adjusted Cosine	0		1	False	1.21291	1.08125	1.05548	1.11655
Combination Algorithm	-				1.05990	1.02814	1.02055	1.03619

This algorithm has been tested online and the result is presented in the first row of Table 13. It shows that the algorithm performs well on test data especially for Test-10 and Test-20. Noticing that as discussed before, due to the lack of input, item-based algorithms do not performs well on Test-5 dataset, so it is fair to say that removing item-based algorithms from Test-5 could potentially yield a better result. The second row shows the result of modified algorithms, the prediction error of Test-5 decreased from 0.79630 to 0.77654, and the overall error reduces from 0.75710 to 0.75062. To further improve the prediction, user-

based Pearson correlation algorithm is removed from Test-10 due to the statistic result from Figure 1, and the final prediction error decreases to 0.74951.

Table 13. Comparison among Customized Algorithm
(UC = User-Based Cosine, UP = User-Based Pearson, IC = Item-Based Cosine, IA = Item-Based Adjusted Cosine)

Method			Error			
Test-5	**Test_10**	**Test-20**	**Test-5**	**Test-10**	**Test-20**	**Total**
UC + UP + IC + IA	UC + UP + IC + IA	UC + UP + IC + IA	0.79630	0.75450	0.72837	0.75710
UC + UP	UC + UP + IC + IA	UC + UP + IC + IA	0.77654	0.75450	0.72837	0.75062
UC + UP	UC + IC + IA	UC + UP + IC + IA	0.77654	0.75000	0.72837	0.74951

7. Conclusion

This report discussed performance of different type of collaborative filtering algorithms as well as their strength and weakness. After the discussion, a customized algorithm was designed in order to further improve the prediction, and the final error is 0.74951. Through discussion, it is safe to say that there is no single algorithm that universally fit to all types of dataset, combining them could potentially produce more accurate prediction, however, the complexity of the algorithm could grows and be less time efficient.

For validation purpose, a python code is developed to randomly sampling 20 users from the training set to do validation. Through the experiment, one can find that patterns revealed by validations do not necessarily 100% fit the test result. It is unsurprising to see algorithms performing well on validation set but not on test set, so it is unwise to closely follow the validation result to tune the parameters which makes the algorithms too good for training data to generally fit broad test dataset.