Impact of Variations of Similarity and Prediction Techniques on User-based and Item-based Collaborative Filtering Recommendations

BASEL ELZATAHRY, Davidson College, USA
NATNAEL MULAT, Davidson College, USA
PAUL CHOI, Davidson College, USA
RIDA SHAHID, Davidson College, USA

Collaborative filtering recommender systems play vital roles in many online platforms to ensure the best experiences for users by personalizing their preferences. Since there are multiple constraints such as amount of resources and time, in real life application, there are trade offs made from the limited resources to produce the optimal result for recommendations. In this research paper, we have considered two factors–accuracy and coverage–to determine the best set of parameters for a recommender system algorithm by conducting different trials with varying parameters.

Additional Key Words and Phrases: Recommender Systems, Collaborative Filtering, Similarity and Prediction techniques

ACM Reference Format:
Basel Elzatahry, Natnael Mulat, Paul Choi, and Rida Shahid. 2021. Impact of Variations of Similarity and Prediction Techniques on User-based and Item-based Collaborative Filtering Recommendations. 1, 1 (March 2021), 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

There are different types of collaborative filtering algorithms for recommendation systems, such as user-based recommender algorithms which compare ratings among users for predictions, and item-based recommender algorithms which compare ratings among items for predictions. In our prior analysis of recommender systems, we designed a program to compare between these collaborative filtering algorithms and also between similarity methods such as Euclidean Distance and Pearson Correlation. However, we simplified the trials by using a small data set of ratings called “criticsratings”. This did not reflect the practical application of recommender systems in real life, where data sets involved in the filtering process consist of a substantial amount of users and items.

In order to expand upon the algorithms we have previously developed, we consider different variations and run the program on a much bigger dataset called “ML-100K” [4]. The two important variations we added, in addition to the similarity methods and algorithmic approach, are significance weighting and similarity threshold. We decided to add these variations by referring to the work of Herlocker et al. Since the user-item matrix derived from a real database

Authors’ addresses: Basel Elzatahry, Davidson College, 102 N Main Street, Davidson, North Carolina, 28036, USA, baelzatahry@davidson.edu; Natnael Mulat, Davidson College, 102 N Main Street, Davidson, North Carolina, 28036, USA, namulat@davidson.edu; Paul Choi, Davidson College, 102 N Main Street, Davidson, North Carolina, 28036, USA, pachoi@davidson.edu; Rida Shahid, Davidson College, 102 N Main Street, Davidson, North Carolina, 28036, USA, rishahid@davidson.edu.

Manuscript submitted to ACM

would be bigger and sparser, determining ways to weigh the similarities among more items and users based on the size of the dataset, while also making the overall process efficient, becomes necessary [2].

The primary goal of this research paper is to determine the best set of variations among thirty six possible combinations of parameters for a recommender system algorithm. We believe our findings would serve as the basis for future research and study of recommender systems. Our findings can be applied to other datasets of a similar size and data structure to ML-100K. This will allow future researchers to efficiently generate accurate recommendations with high coverage.

2 RELATED WORK

Our approach to significance weighting and similarity threshold derives from the work of Herlocker et al [2]. In particular, Herlocker et al. used an unprecedented significance weight of n/50 with n denoting the number of "co-rated" items [2]. We would like to determine if "n/50" is a definite significance weight that always ensures the best outcome or if there are instances where no significance weight or a significance weight of n/25 would be better. In order to determine the best result, along with accuracy, we have considered the coverage of items for each set of variations. Our definition of coverage comes from Herlocker et al., who define coverage to be the "number of items" for which evaluations can be made out of the total number of items recommendations were requested for [3]. In order to combine both accuracy and coverage, we referred to the work of Seminario and Wilson, who use the "AC measure" as a single metric for determining the best results [6].

3 THESIS

3.1 What:

The purpose of this research is to determine which set of parameters for recommender systems algorithms yields the "best" recommendations using user-based and item-based collaborative filtering. We define the best set of parameters as the one which generates the recommendations with the highest accuracy and coverage.

3.2 Why:

The main motivation behind this work lies in the fact that collaborative filtering is still a fertile area for recommender systems research [7]. Moreover, as the size of the dataset and the number of test cases increase, testing single test cases using LOOCV becomes less efficient and can take up to several hours. In order to avoid running several variations of parameters for a large dataset every time, it would be useful to determine the sets of parameters that work best for the particular dataset. In future, accurate predictions can efficiently be generated using these known sets of parameters and further research can easily be carried out on the results. In addition, previous research has shown whether item-based filtering or user-based filtering is better suited for certain datasets but it has not shown which set of parameters (similarity methods, threshold values and significant weightings) work particularly well for them. This work seeks to determine the best set of parameters for the ML-100K dataset with an intent of applying the same parameters to other similar datasets in future for efficiently generating recommendations.

3.3 Hypotheses:

There are several criteria to consider when trying to implement the best recommender system for a given dataset. Previous research shows that when there are much more users than items, item-based methods are more accurate and efficient whereas when there are more items than users, user-based methods are more accurate and efficient [1]. Based on these findings, we hypothesize that the user-based recommender system will prove to be more accurate in our research because the ML-100K dataset has more items than users (1,682 movies and 943 users).

Similarly, we previously designed a recommender system for the smaller dataset, "criticsratings", and our results showed that Pearson Correlation predicted more accurate ratings as compared to Euclidean Similarity Distance for RS. This was shown by the MSE which was lower with Pearson Correlation than with Euclidean Distance for RS. Therefore, we hypothesize that Pearson correlation will prove to be more accurate than Euclidean Distance for RS in predicting ratings.

Oftentimes, there are highly correlated neighbors that are based on a very small number of co-rated items. Due to the small sample, these neighbors prove to be terrible predictors for the active user [2]. In order to overcome this, a significance weight can be applied which determines the amount of trust to place in a correlation with a neighbor [2]. Previous research has shown that applying the significance weighting, particularly a weight of n/50, increases the accuracy of the prediction algorithm by a relatively large amount [2]. Based on these results, we also hypothesize that adding a significance weight will improve the accuracy of our predictions. Since a significance weight of n/50 has shown an increase in accuracy in previous studies, we also expect that we will obtain the best results with a significance weight of n/50.

In order to determine which other users' data will be used in the computation of a prediction for the active user, we will use a similarity threshold. We set an absolute correlation threshold, where all neighbors with absolute correlations greater than or equal to a given threshold are selected [2]. Setting a high threshold limits the neighborhood to only contain very good correlates but for many users, high correlates are not available. This results in a small neighborhood that cannot provide prediction coverage for many items [2]. On the other hand, setting a very small correlation threshold results in a large number of lower correlates. Therefore, this diminishes the purpose of a threshold. Based on these findings, we hypothesize that a threshold of 0.3 would work the best as it is neither too low (0), nor too high (0.5).

Therefore, the overall hypothesis to be tested in this research is that a user-dased recommender system with similarities calculated using Pearson Correlation, n/50 as the similarity significance weighting and 0.3 as the similarity threshold yields the best recommendations.

3.4 How:

The first step in neighborhood-based prediction algorithms is to weigh all users with respect to similarity with an active user. We want to weigh neighbors based on how likely they are to provide an accurate prediction for the active user [2]. We utilize two methods for doing so, Euclidean Similarity Distance for RS and Pearson Correlation.

The Euclidean distance denotes the proximity of two points with lower values when the points are nearby and higher values when they are far apart. Thus, the Euclidean distance between points p and q is the length of the line segment connecting them. For our purposes, we need the distance and similarity to correlate which means that similar users should have higher distance values and dissimilar users should have a lower distance value. In order to ensure this, we add 1 to Euclidean Distance and take its inverse resulting in the following formula:

$$d(p_i, q_i) = \frac{1}{(\sqrt{\sum_i^n (q_i - p_i)^2}} \tag{1}$$

This is defined as the Euclidean Similarity Distance for RS, referred to as distance in this paper, and the values range from 0, representing no similarity, to 1, representing a high similarity between the two variables.

Pearson Correlation describes the extent to which variables vary together divided by the product of the extent to which they vary individually. Therefore, it is a measure of the strength of the association between the two variables.

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2(y_i - \overline{y})^2}} \tag{2}$$

The correlation coefficient ranges between -1 and 1. A value of +1 signifies a perfectly positive linear relation (high correlation), a value of 0 means that there is no linear relation (no correlation) and a value of 1 signifies a perfectly negative relation (opposite correlation) between the two variables.

We calculate these similarities for both user-based and item-based filtering systems. The user-user and item-item similarities are then recorded into User-User Similarity matrix and Item-Item Similarity matrix respectively. Next, for each of these matrices, we predict ratings using Leave-One-Out Cross-Validation (LOOCV). This is done by removing each actual rating from the User-Item matrix. Then, the data we stored as either User-User or Item-Item similarity matrix is used to evaluate the removed rating. Next, the error between the predicted and actual rating is calculated and stored. The removed rating is then restored and we repeat this process for every rating in the User-Item matrix. At the end, the accuracy is calculated from the stored error values and the coverage is calculated from the number of recommendations generated. We also vary the similarity threshold values and the significance weightings as described in the previous section.

Accuracy is the most popular and accepted method of evaluating recommender systems. It can be defined as the extent to which the recommender system produces "mathematically accurate" predictions based on the U-I matrix data. For the purpose of this research, we use statistical accuracy metrics to evaluate the accuracy of our systems by comparing the numerical recommendation scores against the actual ratings for the user item pairs in our dataset [5]. We use Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). A lower error represents a more accurate recommender system as compared to that with a higher error since a lower error implies a minimized difference between the predicted and actual ratings.

The three error metrics used are as follows:

1. Mean Squared Error (MSE): squared difference between predicted ratings p and actual ratings r averaged over n observations. Formally, it is expressed as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Actual\ Rating_i - Predicted\ Rating_i)^2 \tag{3}$$

2. Root Mean Squared Error (RMSE): similar to MAE, large differences are heavily penalized, square-root of mse. Formally, it is expressed as:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Actual\ Rating_i - Predicted\ Rating_i)^2} \tag{4}$$

3. Mean Absolute Error (MAE): absolute difference between predicted ratings p and actual ratings r, averaged over n observations. Formally, it is expressed as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|Actual\ Rating_i\ \text{-}\ Predicted\ Rating_i| \tag{5}$$

We also bring coverage into consideration. For each test run, we compute coverage as the total of number of rating predictions calculated divided by the total number of ratings available for prediction. Thus, coverage is calculated as follows [6]:

$$coverage = \frac{Total\ \#Ratings\ Calculated}{Total\ \#Ratings\ Requseted} \tag{6}$$

A high coverage indicates that the recommender system is able to provide predictions for a large number of items which is considered to be a desirable characteristic of the recommender system [6].

Therefore, in order to determine the best recommender system algorithm and its corresponding parameters, a combination of high accuracy (low error values) and high coverage is needed. Impressed by the evaluation metric used in [6], we also use the “AC Measure” to determine our best recommender system based on “practical accuracy”. The AC Measure combines both accuracy and coverage into a single metric as follows:

$$AC_i = \frac{Accuracy_i}{Coverage_i} \tag{7}$$

where i indicates the $i^t h$ trial in an evaluation experiment [6].

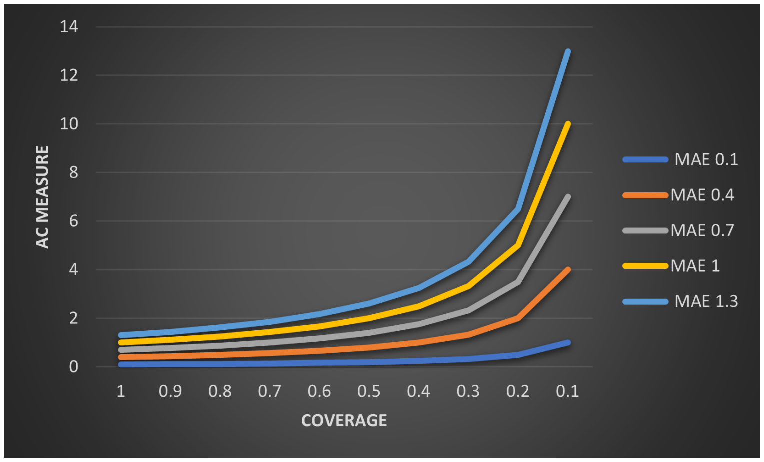


Fig. 1. Illustration of the AC Measure Created with Microsoft Excel.

Figure 1 illustrates the relationship between accuracy, coverage and the AC Measure. We can see that for lower coverage values, the AC Measure value increases. This is because when the coverage is high, the recommender system is able to predict ratings for a high percentage of the dataset, and the accuracy metric more closely indicates the level of performance of the recommender system [6]. When the coverage is low, the accuracy metric is no longer a valid indication of the level of performance and thus, it needs to be adjusted upwards [6]. The AC Measure is beneficial because it can be calculated for each test case and accuracy metric, and the lowest value can be chosen as our best AC Measure. Our best AC Measure corresponds to our best recommendation system.

4 EXPERIMENTAL DESIGN

4.1 Datasets and Algorithms

The data used in this study was the MovieLens 100K Dataset downloaded from GroupLens Research [4]. MovieLens is a web-based research recommender system run by GroupLens Research at the University of Minnesota. The 100K dataset consists of 100,000 ratings for 1,682 movies and 943 users with each user having rated at least 20 movies. Throughout this paper, we refer to this dataset as ML-100K. Ratings provided in this dataset consist of integer values between 1 (did not like) to 5 (liked very much).

For both user-based and item-based collaborative filtering, we use Pearson Correlation and Euclidean distance to calculate similarities and varied the values of the similarity significance weighting and similarity threshold to develop test cases. All the predictions were generated using LOOCV.

4.1.1 *Test cases:*

In order to test the overall hypothesis, the following test cases were developed and executed for both user-based and item-based recommenders using the ML-100K dataset:

1. Pearson Correlation, No similarity significance weighting, No similarity threshold
2. Pearson Correlation, Similarity significance weighted, No similarity threshold
3. Pearson Correlation, Similarity significance weighted, With similarity threshold
4. Pearson Correlation, No similarity significance weighting, With similarity threshold
5. Euclidean Distance, No similarity significance weighting, No similarity threshold
6. Euclidean Distance, Similarity significance weighted, No similarity threshold
7. Euclidean Distance, Similarity significance weighted, With similarity threshold
8. Euclidean Distance, No similarity significance weighting, With similarity threshold

We further divided these into more test cases by varying the values of the significance weighting and similarity threshold, resulting in a total of 36 test cases.

4.1.2 *Accuracy and Coverage Metrics:*

We used MSE, MAE and RMSE to measure the accuracy of the rating predictions. For measuring the coverage, we calculated the percentage of the dataset for which the recommender system was able to provide predictions. We then calculated AC Measures for all test cases using each accuracy metric.

4.1.3 *Test variations:*

Various values and combinations of similarity thresholds and similarity significance weightings were implemented for each test case in order to evaluate the corresponding behavior of the recommender systems. For both user-based and item-based recommender systems, similarity thresholds of 0.0, 0.3, and 0.5, and similarity significance weightings of 1, 25, and 50 were tested. Each case was tested using Pearson Correlation and Euclidean Distance resulting in a total of 36 different variations.

5 RESULTS

Before evaluating our results, we find it important to understand the data that we are working with so the results can be more digestible and easy to interpret.

5.1 Descriptive Analytics for ML-100K:

```
Number of users: 943
Number of items: 1664
Number of ratings: 99693
Overall average rating: 3.53 out of 5, and std dev of 1.13
```

```
Average item rating: 3.08 out of 5, and std dev of 0.78
Average user rating: 3.59 out of 5, and std dev of 0.44
User-Item Matrix Sparsity: 93.65%
```

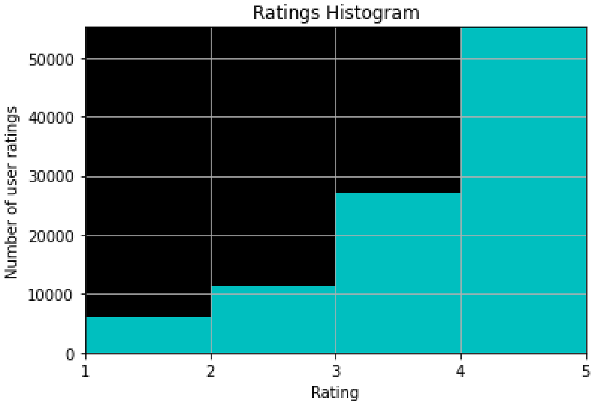


Fig. 2. Histogram of the Ratings in ML-100K Created with Matplotlib.

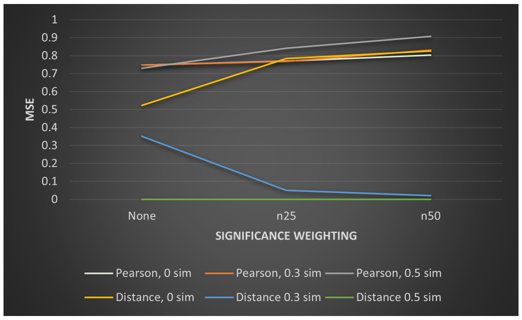
5.2 Popular Items Analytics for ML-100K:

```
Popular items -- most rated:
Title                  #Ratings   Avg Rating
Star Wars (1977)            583        4.36
Contact (1997)              509        3.80
Fargo (1996)                508        4.16
Return of the Jedi (1983)   507        4.01
Liar Liar (1997)            485        3.16

Popular items -- highest rated:
Title                                 Avg Rating   #Ratings
They Made Me a Criminal (1939)                  5.00   1
Star Kid (1997)                                 5.00   3
Someone Else's America (1995)                   5.00   1
Santa with Muscles (1996)                       5.00   2
Saint of Fort Washington, The (1993)            5.00   2

Overall best rated items (number of ratings>=20):
Title                                                   Avg Rating      #Ratings
Close Shave, A (1995)                                       4.49          112
Schindler's List (1993)                                     4.47          298
```

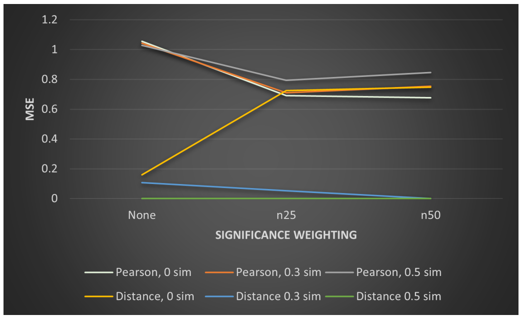
Wrong Trousers, The (1993)	4.47	118
Casablanca (1942)	4.46	243
Wallace & Gromit: The Best of Aardman Animation (1996)	4.45	67

5.3 MSE, MAE, RMSE results:

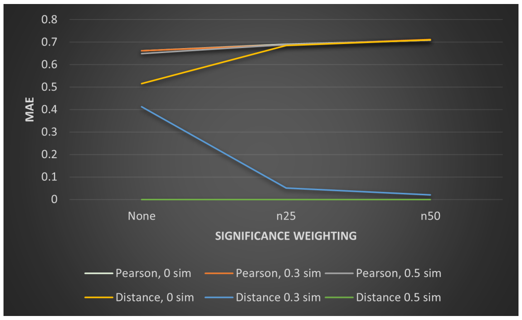
As mentioned above, we use different similarity methods, algorithms, and parameters resulting in 36 total variations. The results for the accuracy measures we have used, in order to determine the best variation, are shown in the following plots:



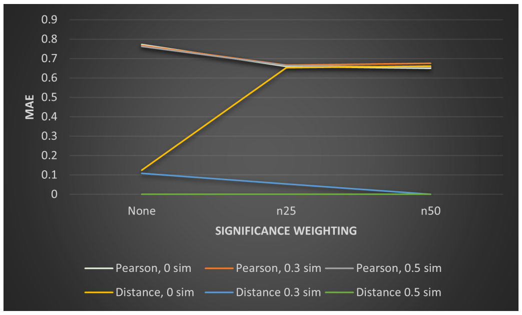
(a) Fig. 3. User-based MSE Created Using Excel.



(b) Fig. 4. Item-based MSE Created Using Excel.



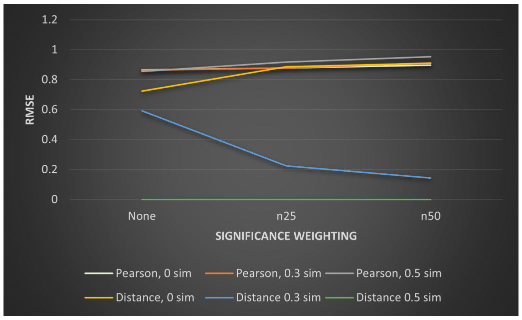
(a) Fig. 5. User-based MAE Created Using Excel.



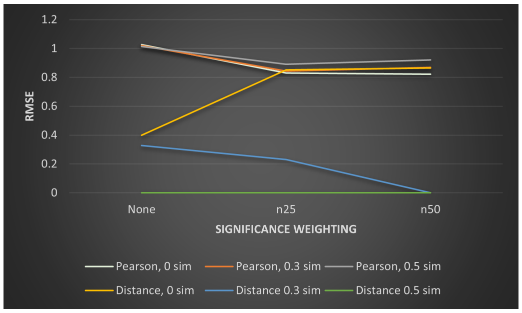
(b) Fig. 6. Item-based MAE Created Using Excel.

The key results of our work and overall trends in accuracy are:

1. As we increase the value of the similarity threshold, the MAE decreases which implies that the accuracy increases.
2. No predictions are generated when the level of threshold is 0.5 and the similarity method is distance because the threshold may be too high to the extent of dismissing all the similarity values. In some cases when predictions are generated, the coverage of the results is too small leading to an error of 0. These results do not accurately represent our recommender systems and thus, are not included in our analysis.
3. For User-based systems, the MAE increases as we add a similarity significance weighting.
4. For Item-based systems, the MAE decreases as we add a similarity significance weighting which agrees with previous research [2].



(a) Fig. 7. User-based RMSE Created Using Excel.



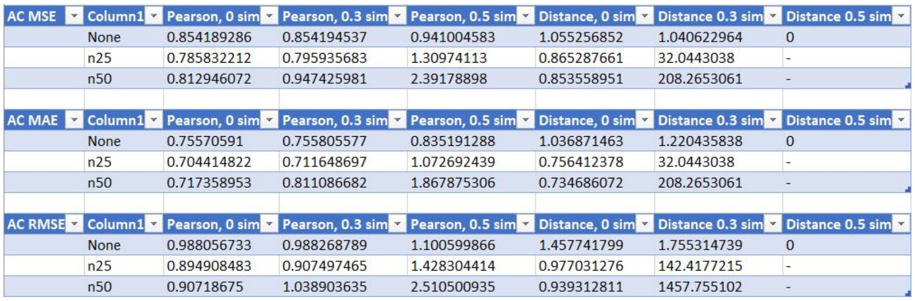
(b) Fig. 8. Item-based RMSE Created Using Excel.

In all the cases, there are a few outliers but we have described the overall trend of varying these parameters.

Moreover, based on these graphs, we find that MSE, MAE, and RMSE are lower using Euclidean distance, a similarity threshold of 0.3 and a significance weighting of n/50. Also, it is clear that the item-based recommender is a little more accurate than user-based recommender.

5.4 AC measurements:

The previous 6 graphs show the different results for accuracy using all 36 variations. However, we asked ourselves if these answers were reliable enough, and we ended up realizing that they are not. We decided to consider every variation's coverage as well and calculate its AC measures to find the one with lowest (the best) and the one with the highest (the worst). The following tables show our AC calculations:



AC MSE	Column1	Pearson, 0 sim	Pearson, 0.3 sim	Pearson, 0.5 sim	Distance, 0 sim	Distance 0.3 sim	Distance 0.5 sim
	None	0.854189286	0.854194537	0.941004583	1.055256852	1.040622964	0
	n25	0.785832212	0.795935683	1.30974113	0.865287661	32.0443038	-
	n50	0.812946072	0.947425981	2.39178898	0.853558951	208.2653061	-

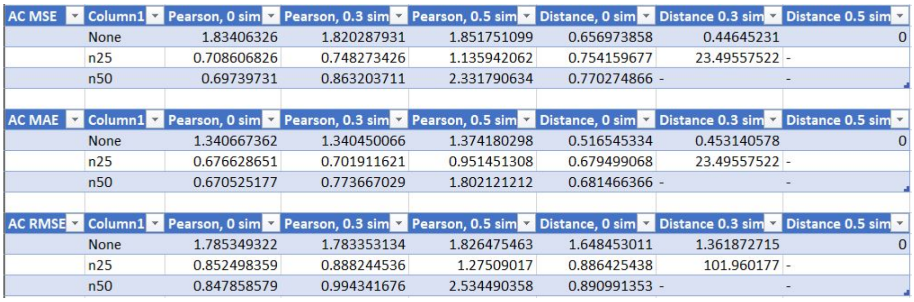
AC MAE	Column1	Pearson, 0 sim	Pearson, 0.3 sim	Pearson, 0.5 sim	Distance, 0 sim	Distance 0.3 sim	Distance 0.5 sim
	None	0.75570591	0.755805577	0.835191288	1.036871463	1.220435838	0
	n25	0.704414822	0.711648697	1.072692439	0.756412378	32.0443038	-
	n50	0.717358953	0.811086682	1.867875306	0.734686072	208.2653061	-

AC RMSE	Column1	Pearson, 0 sim	Pearson, 0.3 sim	Pearson, 0.5 sim	Distance, 0 sim	Distance 0.3 sim	Distance 0.5 sim
	None	0.988056733	0.988268789	1.100599866	1.457741799	1.755314739	0
	n25	0.894908483	0.907497465	1.428304414	0.977031276	142.4177215	-
	n50	0.90718675	1.038903635	2.510500935	0.939312811	1457.755102	-

Table 1. User-based AC Measurements Created Using Excel

The overall trends in the AC values were:

1. For User-based systems: As we increase the similarity threshold, the AC values increase as well. A similar trend is seen when a significance weighting is added except for the case with Euclidean distance and no similarity threshold. In this case, the AC value decreases when a significance weighting is added.



AC MSE	Column1	Pearson, 0 sim	Pearson, 0.3 sim	Pearson, 0.5 sim	Distance, 0 sim	Distance 0.3 sim	Distance 0.5 sim
	None	1.83406326	1.820287931	1.851751099	0.656973858	0.44645231	0
	n25	0.708606826	0.748273426	1.135942062	0.754159677	23.49557522	-
	n50	0.69739731	0.863203711	2.331790634	0.770274866	-	-

AC MAE	Column1	Pearson, 0 sim	Pearson, 0.3 sim	Pearson, 0.5 sim	Distance, 0 sim	Distance 0.3 sim	Distance 0.5 sim
	None	1.340667362	1.340450066	1.374180298	0.516545334	0.453140578	0
	n25	0.676628651	0.701911621	0.951451308	0.679499068	23.49557522	-
	n50	0.670525177	0.773667029	1.802121212	0.681466366	-	-

AC RMSE	Column1	Pearson, 0 sim	Pearson, 0.3 sim	Pearson, 0.5 sim	Distance, 0 sim	Distance 0.3 sim	Distance 0.5 sim
	None	1.785349322	1.783353134	1.826475463	1.648453011	1.361872715	0
	n25	0.852498359	0.888244536	1.27509017	0.886425438	101.960177	-
	n50	0.847858579	0.994341676	2.534490358	0.890991353	-	-

Table 2. Item-based AC Measurements Created Using Excel

2. For Item-based systems: As we increase the similarity threshold, the AC Measure (calculated using MAE) increases. However, this trend does not hold for the variation with Euclidean distance and no significance weighting. For this case, the AC value decreases with a higher similarity threshold. Adding the significance weighting decreases the AC for Pearson Correlation. However, for Euclidean Distance, the AC increases with an added significance weighting. A weighting of n/25 works best with Pearson Correlation.

6 DISCUSSION

In terms of accuracy, our results agree with some aspects of our hypothesis, while also contradicting other parts of our expectations. As we hypothesized, using a threshold of 0.3 and significance weight of n/50 is significantly more accurate than other thresholds and significance weights. On the other hand, Euclidean Distance for RS surprisingly has a higher accuracy than Pearson Correlation. Similarly, the Item-based recommender system proved to be superior to the User-based recommender system. Using our error metrics of MSE, MAE, and RMSE, we determined how accurate each variation is. However, an MAE that is less than 0.1 sounds really promising, but it could be useless if it only predicts 100 ratings out of 100,000 ratings. The users would not be satisfied if they obtained extremely accurate results but for only 1 percent of the possible ratings. Therefore, we used the AC Measure as described previously. Moreover, it does not matter if we use MAE, MSE or RMSE to determine our AC Measure values since all three should yield similar results. In our analysis, we base our conclusions on the results with MAE. The best variation is defined as the one with the lowest AC value and the worst one is defined as the one with the highest AC value. Based on the results that we obtained and showed in the previous section, Item-based recommender that uses Euclidean Correlation, 0.3 similarity threshold, and no significance weighting would have an MAE of 0.10908 and the AC for that would be 0.453, which is the best recorded AC for all the Item-based variations. Meanwhile, the lowest recorded MAE of 0.0531 that uses Euclidean distance, 0.3 threshold, and n/25 has the highest AC measurement of 23.496.

We determine that our hypothesis that a user-based recommender would work better than an item-based one was incorrect because the difference between the users and items is not large enough in the ML-100K dataset. Previous research was conducted on a dataset with much more items than users so the same expectations can not be applied to this dataset. Generally, item-based collaborative filtering is known to produce more accurate results. Moreover, there may not have been a significant difference between the accuracy values with Euclidean distance and Pearson Correlation for our experiments on the "criticsratings" dataset. This may have resulted in one similarity method seemingly working better for one dataset while the other works better for another dataset. This difference may also be due to the large difference between the sizes of the two datasets. In addition to this, we observe that the AC Measure is better with

Pearson Correlation when using a user-based recommender and vice versa for an item-based recommender. Furthermore, in our hypotheses, we determined our best set of parameters based on only the accuracy values. We did not take the coverage and AC Measure into consideration. Therefore, our final results vary from our hypotheses.

In summary, the MAE AC Measure is the lowest (best) when using an item-based recommender with Euclidean distance, 0.3 similarity threshold, and no similarity significance weighting, and it is the highest (worst) when using a User-based recommender with Euclidean distance, 0.3 threshold, and significance weight of n/50. The main factors that make the difference here are the algorithm of the recommender and the significance weighting. We also noticed that better AC measurements are more likely to occur with more coverage and lower levels of threshold. This means that our hypothesis was partially correct in using a significance weight of n/50 and 0.3 similarity threshold for more accuracy, but the best outcome that depends on both accuracy and coverage was yielded with no significance weighting and using Euclidean distance. The best accuracy coverage for Item-based was 0.453, while the best one for User-based was only 0.704; this indicates that using an item-based recommender system is better than using a user-based recommender system.

7 CONCLUSION

Our research on determining the best set of recommender systems algorithm parameters experiments with 36 different parameter combinations. We varied the filtering method based on the item-based and user-based algorithms. In addition, we varied the similarity methods, the similarity threshold and the similarity significance weighting to run our analysis. After determining the accuracy and coverage of each variation, we determined the best set of parameters as the one with the lowest AC Measure value. We also reported the cases with the best accuracy (lowest MAE) and the overall impact of varying the different parameters. This work will provide future researchers guidance on how to choose between an item-based and user-based recommender system and the parameters that work best with them. We believe that these results can be applied to other similar datasets.

8

ACKNOWLEDGMENTS

We would like to give special thanks to Dr. Seminario for providing resources including but not limited to research papers and important algorithms to implement some methods in the recommender system.

9 REFERENCES

[1] C. Desrosiers and G. Karypis. A comprehensive survey of neighborhood-based recommendations methods. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, Recommender Systems Handbook. Springer, 2011.

[2] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In Proceedings of the ACM SIGIR Conference, 1999.

[3] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. Riedl. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems, 22(1):553, 2004.

[4] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. GroupLens: an open architecture for collaborative filtering of netnews. In Proceedings of the ACM CSCW Conference, 1994.

[5] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In Proceedings of the World Wide Web Conference, 2001.

[6] C.E. Seminario and D.C. Wilson. Case Study Evaluation of Mahout as a Recommender Platform, July 2012.

[7] G. Shani and A. Gunawardana. Evaluating recommendation systems. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, Recommender Systems Handbook. Springer, 2011.