User-Based Collaborative Filtering Using Agglomerative Clustering on Recommender System

Malim Muhammad
{malim.muhammad@gmail.com}

Department of Mathematics Education, Faculty of Teacher Training and Education, Universitas Muhammadiyah Purwokerto

Abstract. Content-based, collaborative filtering, demographic, knowledge-based, and hybrid recommender systems are the five categories of recommendation systems. User-based collaborative filtering and item-based collaborative filtering are the two types of collaborative filtering. However, the user-based approaches can be claimed to represent the user; researchers will employ them here. This method is more concerned with the user's likeness, or similarity than with the user's evaluated item. The accuracy of user-based collaborative filtering approaches employing agglomerative Clustering with similarity computations, i.e., cosine similarity, is improved in this study. MovieLens (https://grouplens.org/datasets/movielens/) provided the researchers with the data they needed. Between January 9, 1995, and October 16, 2016, a total of 100004 ratings for 9,125 films were collected from 671 individuals. At least 20 movies have been rated by each user. Each rating has a value of 1 to 5. The data utilized for testing is five value data from each user. In other words, 3,355 data points were tested in total. Using the single linkage clustering approach to cluster films in the use-based method has been shown to improve the accuracy of results that differ significantly between scenarios one and two, namely 3,409 and 3.26. MAE and RMSE are the accuracy gauges utilized in the analysis, and the smaller the value (closer to zero), the better the program results. The findings of two trials (2 Scenarios) revealed significant differences between scenario 1 and scenario 2, namely 3,409 and 3.26. This is because in scenarios 1 and 2, only neighbors with similarity values greater than zero are utilized to find predictions, regardless of whether the neighbor has scored the film to be forecasted or not. In scenario 1, however, the results produced by adding the single linkage clustering approach to the user-based method as mentioned above are not as good. As the value obtained grows larger, the system's level of accuracy decreases. However, the results achieved in scenario 2 are smaller, but the differences are not significant.

Keywords: Recommended System, User-Based Collaborative Filtering, Agglomerative Clustering

1 Introduction

Advances in technology have made the digital search easier. Over time various sites that use search engines, be it selling sites or other sites, also use a recommendation system. The recommendation system can be used in various areas, such as movies, news, music, books, and others. According to Casey (2014), the recommendation system utilizes the history of user behavior such as articles that have been read, products that have been assessed or purchased,

ISTED 2021, July 17-18, Purwokerto, Indonesia

DOI 10.4108/eai.17-7-2021.2312410

music that is often played, and so on to identify the user's preferences which are then used as references to produce final recommendations in the form of items or products [1]. The recommendation system works based on previously stored user information. This information itself can be a numeric value, an ordinal value, or a binary value [2].

According to Ricci et al. (2015), the recommendation system is divided into five types: content-based, collaborative filtering, demographic, knowledge-based, and hybrid recommender system [2]. The general principle of content-based filtering is to identify common traits of an item that gets a high rating from the user and recommend that using the item's characteristics. Collaborative filtering uses assessment information from users and other goods. Demographics recommend items based on the demographic profile or region of that user. Knowledge-based systems recommend goods based on specific information of how much an item meets needs and is useful to users. In comparison, the Hybrid Recommender System is a combination of the recommendation systems above.

Collaborative filtering is divided into two types: user-based collaborative filtering and item-based collaborative filtering [3]. However, researchers will use user-based methods because this method can be said to represent the user. This method pays more attention to the similarity or similarity of the user than the item that the user has assessed. In contrast, item-based pay more attention to the assessment of goods. This method is included in the neighborhood model that directly uses stored assessments to predict.

Some of the advantages of using neighborhood-based methods [2] are

1) Simplicity, neighborhood-based methods are relatively easy to implement
2) Justifiability, this method also provides a concise and intuitive basis of truth on the computing of its predictions.
3) Efficiency, one of the advantages of this method is its efficiency. Because this method does not require pre-computing and storage for data, its determination is not too large.
4) Stability, this method is not unduly affected by additional users, items, and ratings.

The several recommendations contained in this method, there are also shortcomings in the scope of its recommendations [2]. This becomes more visible when the data used has a fairly high sparsity. This will then reduce the scalability of this approach itself. There has been a lot of research done to improve the accuracy performance of collaborative filtering. Leben (2008) uses adjusted cosine to calculate similarity. The comparison of both collaborative filtering methods has been made by Sarwar et al. (2001) using adjusted cosine to calculate similarity on item-based and Pearson correlation on user-based [4]. The result obtained is a user-based method with Pearson correlation has higher accuracy. Based on the above, the author will try to contribute to improving the accuracy of user-based collaborative filtering methods using agglomerative Clustering with similarity calculations, i.e., cosine similarity.

A recommendation system is a technique that advises or suggests goods that are in demand by certain users [2]. There are three main processes of this technique, namely: object data collections and representations, similarity decisions, and recommendation computation. Collaborative filtering methods collect and analyze large amounts of information about a user's behavior, activity, or preferences and predict what users will like. This method does not rely on content that can be analyzed. Therefore, this method can recommend complicated items such as movies without understanding the movie itself. This is the advantage of collaborative filtering methods. One of the best-known examples of collaborative filtering is item-to-item or item-based (people who buy x and buy y), an algorithm popularized by the recommendation system Amazon.com [5]. Last.fm recommends music based on comparisons of the same user's listening habits, while Readgeek compared book ratings to recommendations. Facebook, MySpace, LinkedIn, and other social networks use collaborative filtering to recommend friends, groups,

and other social connections (by checking the network of connections between users and their friends). Twitter uses a lot of signals and calculations in memory to recommend to its users who to follow [2],[6].

In collaborative filtering, there is also a user-based method, where this method will recommend goods to users x, which is also liked by other users similar to x [2]. So, between the user-based method and the item-to-item method, it is almost the same. However, the difference lies in what the recommendations are. User-based sees the user's resemblance to other users, while item-based looks at it in terms of goods.

Agglomerative Clustering has a way of working with the assumption that n items want to be clustered and matrix distance or similarity N *N. The basic process, according to Johnson (1967), is as follows:

1) Create clusters for each item so that if it has n items, it will now have n clusters, which contain one item. Let the similarity between clusters equal the similarity between the items in it.
2) Find the nearest (most similar) cluster pair and combine the two into one cluster, so we now have one fewer cluster
3) Calculate the similarity between the new cluster and each of the old clusters.
4) Repeat steps 2 and 3 until all items are grouped into one group with size N

Step 3 can be done in several different ways, namely minimum proximity, maximum proximity, average proximity, and centroid proximity. At minimum proximity (single-linkage), the shortest distance (largest similarity) is used to create clusters, while at maximum proximity (double-linkage) used is the largest distance (smallest similarity).

Pearson Correlation Coefficient and Cosine Similarity are commonly used and well-known similarity methods. That's why I use both methods. Pearson correlation coefficient will be used for calculation of final prediction of goods (recommendations) and cosine similarity for similarity calculations to create clusters.

The similarity is a method in machine learning that serves to calculate similarities between 2 or more data. Similarity calculations use algorithmic methods such as Pearson correlation, cosine similarity, and many more. Pearson Correlation is one of the algorithms commonly used in calculating the similarity between users and other users. Correlation is a measurement technique that determines the closeness of relationships between two sets of different numbers. Correlation calculations have a condition where the set of numbers calculated must have a fixed order and pair with each other between the two sets. The results of measurement can be either positive relationships or negative relationships. Positive relationships indicate that both sets have a tendency to increase or increase equal values. In contrast, the negative relationship shows that both sets have a tendency to decrease or decrease in equal value [8]. Here is the Pearson Correlation equation for calculating the similarity between users:

$$PC(u,v) = \frac{\sum_{i=1}^{n}(r_{ui} - \overline{r}_u)(r_{vi} - \overline{r}_v)}{\sqrt{\sum_{i=1}^{n}(r_{ui} - \overline{r}_u)^2 (r_{vi} - \overline{r}_v)^2}} \quad 1)$$

Where:

- $PC(u,v)$ is the similarity value between user u and user v
- r_{ui} and r_{vi} is the user rating ui and vi to item-i

- $\overline{r}_u$ and $\overline{r}_v$ is the average rating u and v to item-i
- n is the number of items

Cosine similarity is a method used to calculate the similarity of each user with each other. Different rating scales between different users will result in different similarity values [8]. Here is the Cosine Similarity equation for calculating the similarity between users:

$$Sim(u_i, u_k) = \frac{r_i . r_k}{|r_i||r_k|} = \frac{\sum_{j=1}^{m} r_{ij} r_{ik}}{\sqrt{\sum_{j=1}^{m} r_{ij}^{\,2} \sum_{j=1}^{m} r_{kj}^{\,2}}} \quad 2)$$

Where:

- $Sim(u_i, u_k)$ is the similarity value between the ith user and the kth user
- u_i and u_k is the ith user and the kth user
- r_i and r_k is the ith user rating and the kth user
- m is the number of items

Different similarity indicators will result in different prediction scores. Mean absolute error (MAE) and root mean square error (RMSE) are two common indicators for measuring the accuracy of similarity methods. The smaller the value, the better the prediction accuracy. It is defined as follows:

1) Mean absolute error (MAE) is the average of the absolute error of the user's predicted score and the true score in the scoring test set q_i p_i.

$$MAE = \frac{\sum_{i=1}^{n} |p_i - q_i|}{n} \quad 3)$$

2) Root-mean-square error (RMSE) is the mean square root of the true score value and the predicted score value of the user in the test set: p_i q_i

$$RMAE = \sqrt{\frac{\sum_{i=1}^{n} (p_i - q_i)^2}{n}} \quad 4)$$

2 Research Methods

The data used by the researchers was obtained from a site called MovieLens (https://grouplens.org/datasets/movielens/). The dataset consists of 100004 ratings of 9,125 films obtained from 671 users between January 9, 1995, and October 16, 2016. Each user has rated at least 20 movies. Each rating is worth from 1 to 5. For testing, the data used is five value data from each user. In other words, the total data testing used is 3,355 data.

The test scenario is done by conducting the experiment twice. First, implement the user-based method directly by using 100 neighbors regardless of whether the neighbor has judged the film to be predicted. Second, implement the user-based method directly by using a maximum of 100 neighbors who have assessed the film to be predicated on the implementation of Clustering. In experiments one and two, all films will be clustered first into 5 clusters because the rating value used is nominal 1 to 5. Clustering is done using the agglomerative clustering technique (single linkage). After that, it will be implemented user-based methods in the same way in the first and second scenarios. For more details, you can see it in figure 1.

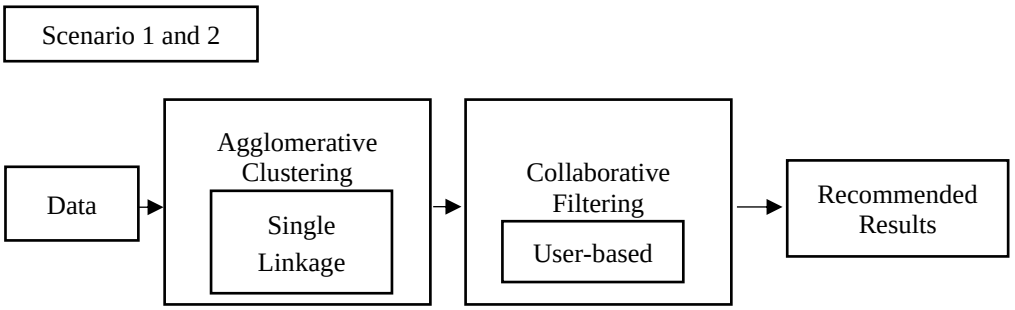

Figure 1. Scenario Illustration

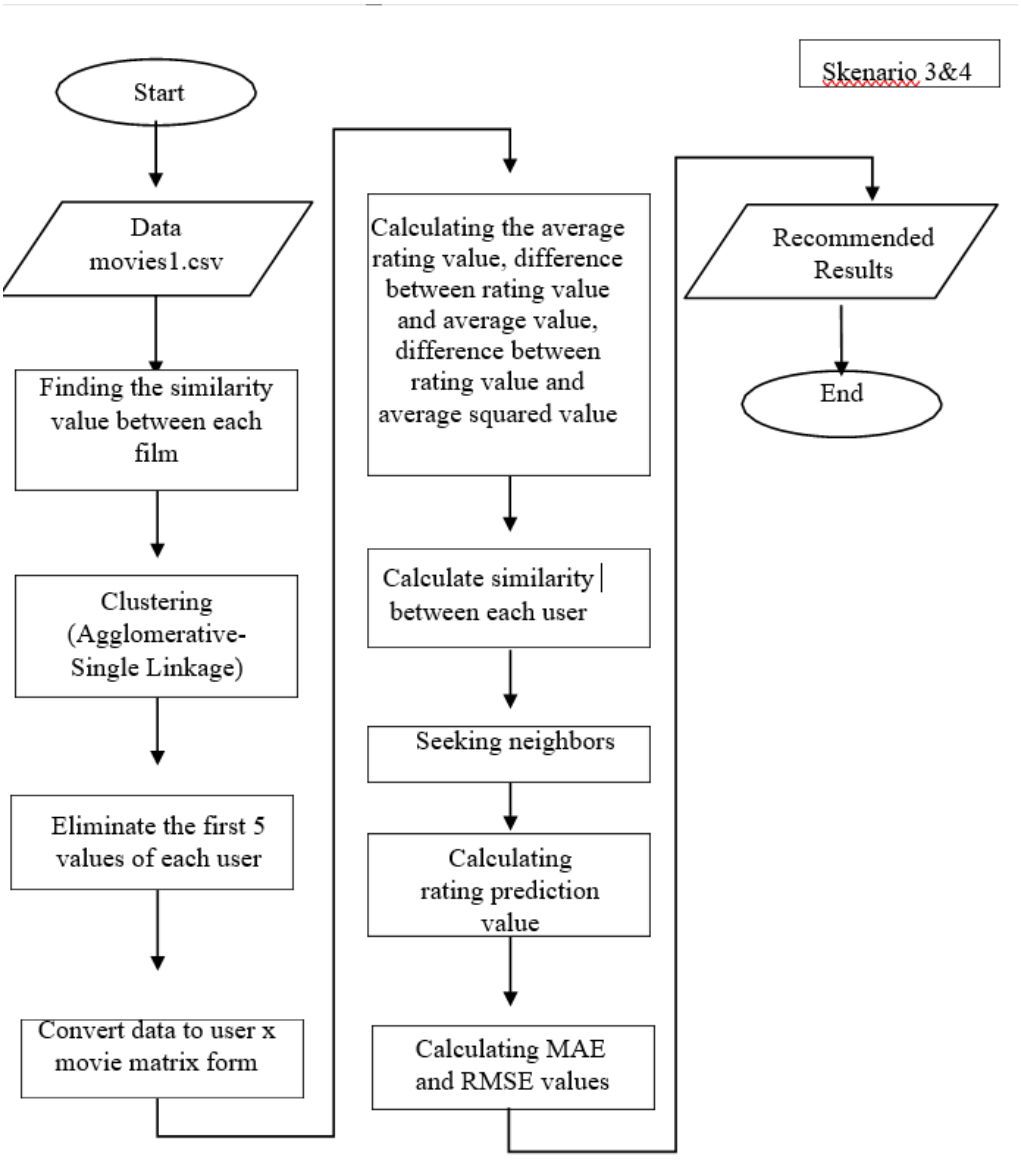

Figure 2. Flowchart process design with Clustering

3 Results and Discussion

The implementation of process blocks in a class is in table 1, and the implementation of process blocks in functions is in table 2. While in table 3 contains the implementation of functions in the following classes:

Table1. Implementation of process blocks in a class

Process Block	Class
Eliminates the first 5 values from each user	pearson.m, pearson1.m, pearson_cluster.m and pearson_cluster1.m
Convert data to user matrix form x movie	pearson.m, pearson1.m, pearson_cluster.m and pearson_cluster1.m
Calculate the predicted value rating	Pearson.m, pearson1.m, pearson_cluster.m and pearson_cluster1.m
Calculate the MAE value and RMSE	Pearson.m, pearson1.m, pearson_cluster.m and pearson_cluster1.m
Find the similarity value between each movie	nilai_cosine.m
Clustering (Agglomerative - Single Linkage)	AHC.m

Table2. Implementation of process blocks on functions

Process Block	Function
Calculate the average rating	m_nilaiRata2
the difference in rating value with the average value	m_rai
the difference in rating value with the average square	m_raikuadrat
Hitung similarity	m_pearson
Looking for neighbors	finding_topN

Table3. Implementation of functions in the class

Function	Class
m_nilaiRata2	pearson.m, pearson1.m, pearson_cluster.m and pearson_cluster1.m
m_rai	pearson.m, pearson1.m, pearson_cluster.m and pearson_cluster1.m
m_raikuadrat	pearson.m, pearson1.m, pearson_cluster.m and pearson_cluster1.m
m_pearson	pearson.m, pearson1.m, pearson_cluster.m and pearson_cluster1.m
finding_topN	pearson.m, pearson1.m, pearson_cluster.m and pearson_cluster1.m

The detailed usability of each class is in table 4, while table 5 contains the usability of each function in the system.

Table 4. Class and its uses

Class	Uses
Pearson	Search for rating predictions using 100 neighbors without seeing if the neighbor has passed the movie. It will be predicted or not.
Pearson1	Search for rating predictions using a maximum of 100 neighbors who have rated the film to be predicted
Pearson_cluster	Search for rating predictions using 100 neighbors without seeing if the neighbor has passed the film to be predicted or not by implementing agglomerative Clustering
Pearson1_cluster	Search for rating predictions using a maximum of 100 neighbors who have rated the film to be predicted by implementing agglomerative Clustering
Nilai_cosine	Search for the cosine similarity value of all movies
AHC	Useful for clustering movies using techniques agglomerative clustering

Table5. Its functions and uses

Function	Uses
AND	Useful for finding weight values between two films
m_nilaiRata2	Search for the average rating of each user
m_rai	Find the difference in rating value by the average value
m_raikuadrat	Searches for the difference in the value of the rating with the average square.
m_pearson	Search for Pearson value(similarity)between all users
finding_topN	Search for the 100 neighbors with the greatest similarity value of all users
cari_cluster	Search for movie clusters to predict
cari_user	Search for users – users who have rated movies in the movie cluster to predict

Class pearson_cluster (scenario 1)

In the pearson_cluster class will be used a user-based method by applying agglomerative Clustering (single linkage). Where the steps to cluster the entire film have been done first.

1) Read data from ratings1.csv.
2) Eliminate the first five ratings from each user that will be used for data testing.
3) Convert to the form of user matrix x movie.
4) Look for the movie cluster to predict and the users-users who have ranked the movie–the movie that is in the cluster. Then convert the data into the form of a user matrix x movie.
5) Calculate the average rating of each user, the difference in the rating value with the average value, the difference in the rating value with the average value of the square.
6) Calculates the similarity between each user using the Pearson correlation algorithm
7) Search for 100 neighbors of all users using the similarity value you've searched for
8) Look for the user index and the movie to predict. This is because the presence of users and movies that will be predicted cannot match the value of rows or columns.

9) Perform a neighbor selection process to calculate a user's rating prediction of a movie (out of 100 neighbors of that user, will be used whose similarity value is only large from zero)
10) Then calculate the rating prediction using the user-based method. In a way, all neighbor ratings are multiplied by the similarity value, then divided by the same amount of the entire neighborhood. This step will continue until all data testing has been predicted.
11) Calculate MAE and RMSE to determine the system's error value or accuracy rate after all the testing data is calculated or predicted.

Class pearson_cluster1 (scenario 2)

The steps in this class are nearly identical to those in the Pearson cluster class. It's only that this class has a different approach to selecting neighbors. The neighbors sought are the users who will be predicted's neighbors, with a maximum of 100 persons, each of whom has judged the movie to be forecasted. MAE and RMSE are the accuracy gauges utilized in the analysis, and the smaller the value (closer to zero), the better the program results. From two trials (two situations), a significant difference was found between scenario one and scenario 2, namely 3,409 and 3.26. This is because, in scenarios 1 and 2, only neighbors with similarity values greater than zero are utilized to check for predictions, regardless of whether the neighbor has scored the movie to be forecasted or not. In scenario 1, however, the results produced by adding the single linkage clustering approach to the user-based method as mentioned above are not as good. As the value obtained grows larger, the system's level of accuracy decreases. However, the results produced in scenario two do get smaller, but the differences are not significant.

4 Conclusion

The employment of the single linkage clustering approach to cluster movies using the use-based method has been shown to enhance the accuracy of findings that differ significantly between scenarios 1 and 2, namely 3,409 and 3.26. MAE and RMSE are the accuracy gauges utilized in the analysis, and the smaller the value (closer to zero), the better the program results. From two trials (two situations), a significant difference was found between scenario one and scenario 2, namely 3,409 and 3.26. This is because, in scenarios 1 and 2, only neighbors with similarity values greater than zero are utilized to check for predictions, regardless of whether the neighbor has scored the movie to be forecasted or not. In scenario 1, however, the results produced by adding the single linkage clustering approach to the user-based method as mentioned above are not as good. As the value obtained grows larger, the system's level of accuracy decreases. However, the outcomes achieved in scenario 2 grow smaller, although the changes are not too significant.

Acknowledgments. The authors would like to express their gratitude to Universitas Muhammadiyah Purwokerto for allowing us to assist and support in completing this research.

References

[1] Casey, E. 2014. Scalable Collaborative Filtering Recommendation Algorithms on Apache Spark. Claremont: CLAREMONT McKENNA COLLEGE.
[2] Ricci, F., Rokach, L. & Shapira, B. 2015. Recommender Systems Handbook (second edition). New York: Springer Science+Business Media LLC.
[3] Schafer, J.B., Frankowski, D., Herlocker, J., & Sen, S. 2007. Collaborative Filtering Recommender Systems. Dalam Brusilovsky, P., Kobsa A. & Nejdl, W. The Adaptive Web (hlm. 291-324). Berlin: Springer.

[4] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. 2001. Item-based collaborative filtering recommendation algorithms. Proceedings Konferensi internasional ke-10 dari World Wide Web, halaman 285–295, New York, NY, USA. ACM.
[5] Linden, dkk. 1998. Collaborative Recommendations Using Item-to-Item Similarity Mappings.
[6] Gupta, P. dkk. WTF: The who-to-follow system at Twitter. Proceedings konferensi internasional ke-22 dari World Wide Web. May 13 - 17, 2013. Pages 505-514.
[7] Jhonson, K. 1967. Notes on regression and inheritance in the case of two parents.
[8] Theodorus, A., & Budiyanto Setyohadi, D. (2016). User-Based Collaborative Filtering by Utilizing Pearson-Correlation To Find Nearby Neighbors In The Recommendation System. Master's Thesis in Information Technology, Atma Jaya University Yogyakarta, 1–6. http://e- journal.uajy.ac.id/8924/