Accredited Ranking SINTA 2
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online on: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 7 No. 4 (2023) 743 - 750 ISSN Media Electronic: 2580-0760

Solution to Scalability and Sparsity Problems in Collaborative Filtering using K-Means Clustering and Weight Point Rank (WP-Rank)

Mohamad Fahmi Hafidz[1], Sri Lestari[2]*
[1,2]Master of Informatics Engineering, Faculty of Computer Science, Institut Informatika dan Bisnis Darmajaya
[1]mohamad.hafidz.2021211024@mail.darmajaya.ac.id, [2]srilestari@darmajaya.ac.id

Abstract

Collaborative Filtering is a method to be used in recommendation systems. Collaborative Filtering works by analyzing rating data patterns. It is also used to make predictions of user interest. This process begins with collecting data and analyzing large amounts of information about the behavior, activities, and tendencies of users. The results of the analysis are used to predict what users like based on similarities with other users. In addition, Collaborative Filtering is able to produce recommendations with better quality than recommendation systems based on content and demographics. However, Collaborative Filtering still faces scalability and sparsity problems. It is because the data is always evolving so that it becomes big data, besides that there are many data with incomplete conditions or many vacancies are found. Therefore, the purpose of this study proposed a clustering and ranking based approach. The cluster algorithm used K-Means. Meanwhile, the WP-Rank method was used for ranking based. The experimental results showed that the running time was faster with an average execution time of 0.15 second by clustering. In addition, it was able to improve the quality of recommendations as indicated by an increase in the value of NDCG at k=22, the average value of NDCG was 0.82, so that the recommendations produced had more quality and more appropriate with user interests.

Keywords: collaborative filtering; scalability; sparsity, K-means; WP-rank

1. Introduction

Recommendation systems are often used to solve problems by seeking relevant information from the available collection of information. Generally applied to fields that have large amounts of data and continue to grow over time. With the explosive growth of online information, recommendation systems play a key role to alleviate such information overload[1]. This system will process user information which then provides recommendations according to the characteristics of the user, namely according to their specialization. The recommender system's main idea is to build relationship between the products, users and make the decision to select the most appropriate product to a specific user[2]. One of the methods used in providing recommendations according to user interest is collaborative filtering.

Collaborative filtering (CF) works by analyzing rating data patterns, which are then used to make predictions of user interest based on similarities with other users. CF has several advantages including easy to implement and can filter all kinds of information or goods without having to analyze comments from users. Collaborative filtering (CF) methods produce recommendations based on usage patterns without the need of exogenous information about items or users[3]. In addition, CF generates high quality recommendations.

CF is also a comprehensive and common method, and is widely implemented in various fields. Various companies like face book which recommends friends, LinkedIn which recommends job, Pandora recommends music, Netflix recommends movies, Amazon recommends products etc. use recommendation system to increase their profit and also benefit their customers[4]. Recommendations are obtained by looking at the history of previous purchases or by looking at similar users. Implementing the application to the platforms is pampering the users by intelligently providing a list of movies of their favors out of a huge movie collection. Many works have been done on movie recommendations[5].

Although CF is a popular method, it faces major problems, namely cold start, sparsity and scalability[6]. Cold start is a condition of new users who have never given a rating to a product, so that the information obtained for the direction of user interest is difficult to know or new items that have never received a rating

Accepted: 28-09-2022 | Received in revised: 12-06-2023 | Published: 12-08-2023

from the user. If the direction of interest is unknown, it is difficult to give recommendations. Sparsity is data in sparse conditions; this is because the data matrix contains incomplete or many data voids are found. If sparse data are found, then the resulting similarity value will be small, either on the similarity between users or the similarity between items, so that the resulting recommendations are of less quality. Scalability is a condition where recommendation systems need to increase their computing power to offer timely recommendations. This is done with large-scale data conditions and requires a lot of resources and reliable computing.

Several studies have been carried out to overcome this problem, such as that of Das, J et al who proposed a clustering-based collaborative filtering approach, by partitioning the data using CURE (Clustering using representatives). The results of the cluster are then processed using a collaborative filtering algorithm so as to produce recommendations for target users. This process is carried out for each cluster, so it does not process the entire database of user items. In this way, the time required becomes faster. In addition to overcoming the scalability problem, the clustering approach can overcome the sparsity problem by reducing the dimension of the rating matrix and reducing noise data. In addition, it significantly reduces running time and with quality recommendations [7].

Wang, L et al proposed a diversified and scalable recommendation method (DR_LT) to overcome problems in neighborhood-based collaborative filtering (CF). Some of these problems include an increase in the volume of rating item data from users, so the resulting recommendations are less efficient. This is because the recommendation system will analyze all ranking data when searching for similar users or similar items. In addition, neighborhood-based collaborative filtering (CF) pays more attention to recommendation accuracy, while key indicators such as recommendation performance are often ignored such as recommendation diversity (RD) which will have an impact on recommendation results and reduce user satisfaction. By using a DR_LT, which is utilize locality-sensitive hashing and cover trees to optimize the list of recommendations so that performance becomes effective, besides producing item recommendations that are accurate, diverse and able to solve scalability problems [8].

Several studies have also been conducted to overcome the problem of sparsity, including those conducted by Andra, D and Baizal, Z proposed Principal Component Analysis (PCA) and K-Means Clustering to overcome the sparsity problem. PCA is used to reduce data dimensions and improve the performance of K-Means clustering. While the K-Means Algorithm is used to form data clusters and reduce the amount of data processed. Using PCA and K-Means results in a lower RMSE value compared to other models [9]. The same thing was also done by Ardimansyah, MI et al., proposing Matrix Factorization to fill in the empty rating values to overcome the problem of sparse rating data [10].

In addition, study to overcome sparsity was also carried out by Lestari, S., et al who proposed Weight Point Rank (WP-Rank) which maximizes the use of ranking data to generate product weights. The experimental results show that the WP-Rank method is superior to the Borda method [11]. They were followed by proposing the PoratRank method to generate product rankings by optimizing rating data so that the aggregation results are in the form of product rankings recommended to users according to their interests. This process is able to produce higher quality recommendations[12].

Meanwhile, our next study is to combine a clustering approach with a ranking based approach to overcome the problems of scalability and sparsity. The K-Means clustering algorithm is used to overcome scalability problems, while WP-Rank is used to overcome the sparsity problem by performing an aggregation process so as to produce higher quality recommendations that are in accordance with user preferences.

2. Research Methods

This study solved the problem of scalability and sparsity in Collaborative Filtering using clustering and ranking based approaches. The cluster algorithm used K-Means. Meanwhile, the WP-Rank method was used for ranking based. The stages of the study was seen in Figure 1.

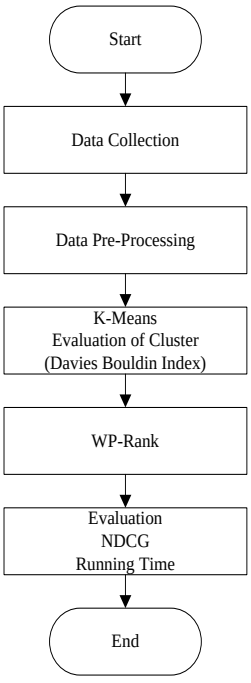

Figure 1. Research Stages

DOI: https://doi.org/10.29207/resti.v7i4.4543

Figure 1 explains the stages in this research, namely first conducting data collection by accessing it from MoviLens.org, namely ml-100k data. The next step is pre-processing by removing the zip code data. After that, the clustering process was carried out using K-Means. The resulting cluster data is then ranked using WP-Rank to produce a movie ranking. The final stage is to evaluate the ranking quality using the NDCG.

The initial stage of this study was started by collecting datasets from movielens.org, namely movielens 100k with 943 users and 1682 movies. The demographic information includes age, gender, occupation, zip. User demographic information was seen in Table 1.

Table 1. The Example of Information Data for Demography User

User Id	Gender	Age	Occupation	Zip Code
1	M	56	16	70072
2	M	25	15	55117
3	M	45	7	2460
4	M	25	20	55455
5	F	50	9	55117
6	M	35	1	6810
7	M	25	12	11413
8	M	25	17	61614
9	F	35	1	95370
10	F	25	1	4093

The list of occupations can be seen in Table 2. After the data collection stage was complete, it was followed by pre-processing data. It was done by changing the empty rating data with a value of 0, in addition to changing the demographic data to numeric, as in gender to 1 and 2, while in occupation to 1-21, as shown in Table 3.

The next step was to perform clustering using the data as shown in Table 4 with the K-Means algorithm, and evaluation using the Davies Bouldin Index method to determine the optimal number of clusters. The results of the cluster were ranked using the WP-Rank method, so as to produce recommendations in the form of film rankings.

Table 2. List of Occupation

Id	Occupation	Id	Occupation
0	Other Or Not Specified	11	Lawyer
1	Academic / Educator	12	Programmer
2	Artist	13	Retired
3	Clerical Admin	14	Sales Marketing
4	College / Grad Student	15	Scientist
5	Customer Service	16	Self-Employed
6	Doctor / Health Care	17	Technician /
7	Executive Managerial	18	Tradesman /
8	Farmer	19	Unemployed
9	Home maker	20	Writer
10	Student		

Table 3. The example of pre-processing data result

Id	Age	Gender	Occupation
1	24	2	20
2	53	1	14
3	23	2	21
4	24	2	20
5	33	1	14
6	42	2	7
7	57	2	1
8	36	2	1
9	29	2	19
10	53	2	10

The next step was to evaluate the ranking quality using NDCG, and evaluate the time used for method execution (running time).

Table 4. The Example of clustered data using K-Means algorithm

Id	Age	Gender	Occupation	movie1	movie2	movie3	…	movie1682
1	24	2	20	5	3	4	…	3
2	53	1	14	4	0	0	…	0
3	23	2	21	0	0	0	…	0
4	24	2	20	0	0	0	…	0
5	33	1	14	4	3	0	…	0
6	42	2	7	4	0	0	…	0
7	57	2	1	0	0	0	…	0
8	36	2	1	0	0	0	…	0
9	29	2	19	0	0	0	…	0
10	53	2	10	4	0	0	…	0
.	.	.	.	.	.	.	…	.
.	.	.	.	.	.	.	…	.
.	.	.	.	.	.	.	…	.
943	22	2	19	4	5	3	…	0

2.1 K-Means

Clustering is a process to group data into several clusters or groups so the data in one cluster has a maximum level of similarity and data between clusters has a minimum similarity[13]. K-means clustering algorithm is considered one of the most powerful and popular data mining algorithms in the research community. K-means is a well-known unsupervised, iterative, partitioning learning algorithm in the field of data mining[14]. K-means was a simple algorithm and the process is fast, so it is widely used in study. The K-Means algorithm used a partitioning system to group data into two or more clusters [15]. The K-Means algorithm worked by grouping objects based on the

DOI: https://doi.org/10.29207/resti.v7i4.4543

cluster center point (centroid) closest to the object. The goal was to group objects by maximizing the similarity of objects in one cluster and minimizing the similarity of objects between clusters. The measure of similarity in the cluster used a function of distance. So that the similarity of the object is calculated based on the shortest distance between the object and the centroid point. One of the methods used to calculate the distance Euclidean Distance Space by knowing the shortest distance between two points. The stages of the K-Means algorithm were:

Step 1: Determine the number of clusters (k=…) to be formed; Step 2: Determine the centroid (initial initiation can be done by selecting data randomly); Step 3: Calculate the distance on each data to the centroid. This study used Euclidean Distance Space, using Equation 1.

$$D(x_i, c_j) = \sqrt{\sum_{i=1}^{n}(x_i - c_j)^2} \quad (1)$$

Step 4: Grouping data based on proximity to the centroid. The smaller the distance value, the closer the data is to the cluster centroid; Step5: Determine the new centroid, by finding the average value of the data that was a member of the cluster, using Equation 2.

$$C_{kj} = \frac{\sum_{h=1}^{p} y_{hj}}{p}; y_{hj} \in cluster\ to - k \quad (2)$$

Repeat steps 2 to 5, this loop was able to stop if the data position does not change anymore.

2.2 WP-Rank

The working steps of the Weight Point Rank (WP-Rank) method were:
Step 1: Counting the number of equal ratings using Equation 4 and 5.

$$S_{(u_g,p_h)} = \sum_{k=1}^{n} SR(R_{(u_g,p_h)}, R_{(k,p_h)}) \quad (4)$$

$$SR(R_{(u_g,p_h)}, R_{(k,p_h)}) = \begin{cases} 1, if\ R_{(u_g,p_h)} = R_{(k,p_h)} \\ 0,\ if\ R_{(u_g,p_h)} = 0 \\ 0, otherwise \end{cases} \quad (5)$$

U = {u_1, u_2, ..., u_g, …, u_{l-1}, u_l} (U was User) dan P = {p_1, p_2, …, p_h, …, p_{m-1}, p_m} (P was a produk).

Step 2: Determine product points was done using Equation 6 and 7.

$$P_{(u_g,p_h)} = 1 + \sum_{k=1}^{m} PR(u_g, p_h, k) \quad (6)$$

$$PR(u_g, p_h, k) = \begin{cases} 1, if\ R_{(u_g,p_h)} > R_{(u_g,k)}, \\ 1, if\ R_{(u_g,p_h)} = R_{(u_g,k)}, S_{(u_g,p_h)} > S_{(u_g,k)}, \\ 1, if\ R_{(u_g,p_h)} = R_{(u_g,k)}, S_{(u_g,p_h)} = S_{(u_g,k)}, u_g < k, \\ 0, otherwise \end{cases} \quad (7)$$

Step 3: Determine Process for *ranking* product ($P_{(u_g,p_h)}$) acquired from value 1 added by the result from the calculation required point was seen in equation 7.

Step 3: Counting Weight Point using Equation 8

$$WP_{(u_g,p_h)} = \left(S_{(u_g,p_h)} + R_{(u_g,p_h)}\right) P_{(u_g,p_h)} \quad (8)$$

Step 4: Counting Weight Point Rank (WP-Rank) was done using Equation 9.

$$WP - Rank_{(p_h)} = \sum_{k=1}^{n} WP_{(k,p_h)} \quad (9)$$

The ranking results from WP-Rank are then taken by Top-K to be recommended to users.

2.3 Evaluation

The Davies Bouldin Index (DBI) was used for the purpose of measuring by maximizing the distance between clusters, and at the same time minimizing the distance between clustered points. The value of DBI is used to measure the quality of clustering[16]. The value of the DBI indicated the quality of the cluster, the smaller the value of the DBI. It stated that the better the "k" value and it was the optimal criterion for the number of clusters [17]. DBI is one of the methods used to evaluate the internal cluster generated by the clustering algorithm. The smaller DBI value indicates that the number of clusters formed is the best. DBI is used to maximize the distance from one cluster to another. In addition, it is used to minimize the distance between points in a cluster. If the value of the similarity of characteristics in each cluster shows a smaller value, it indicates that there are differences between clusters, so the maximum distance is obtained. If the intra-cluster distance shows a minimal value, then the level of similarity of cluster characteristics is high[18].

NDCG, namely Normalized Discounted Cumulative Gain, is a widely used ranking metric in information retrieval and machine learning[19]. NDCG is one of the most commonly used measures to quantify system performance in retrieval experiments[20]. Compared to other measures, NDCG has the advantage that handles multiple levels of relevance, and includes a position dependence for results shown to the user[21]. NDCG served to measure the performance of the recommendation system by looking at the relevance value of the entity [22]. The quality of the ranking was evaluated using GCG, which was evaluating and the top product from the ranking results[23]. The NDCG equation was written as Equation 10 and Equation 11.

$$DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i - 1}}{log_2(i+1)} \quad (10)$$

$$NDCG_p = \frac{DCG_p}{IDCG_p} \quad (11)$$

Several studies have evaluated the running time to see the algorithm's performance. There are several ways to do this, namely by calculating real-time or by calculating time complexity. When the algorithm is run (execution), it will calculate how long it will take, which is generally measured in seconds[24]. The environment and the complexity of the algorithm greatly affect the results of the running time evaluation.

DOI: https://doi.org/10.29207/resti.v7i4.4543

This experiment uses MATLAB R2014a software, with Intel Core I7 hardware, 1TB HD capacity, and 8GB of RAM. These specifications are then used to evaluate the running time of the WP-Rank method. In addition to calculating real time, running time was measured by calculating the time complexity T(n). The algorithm is run with a number of computational steps according to the input size n so that T(n) is obtained. By using the time complexity of the algorithm, it was possible to determine the rate of increase in the time required for the algorithm with increasing input size n. Input size n: amount of data processed by an algorithm.

3. Results and Discussions

This study used Movielens 100k data, so it was clustered using the K-Means algorithm. The results of the cluster were then processed using the WP-Rank method so as to produce a product ranking (movie) as a basis for recommendations to the user in the form of a list of movies according to their specialization.

Experiments were carried out by clustering datasets based on demographic data, especially user age using the K-Means algorithm. The number of "k" starts from 2-25 and was evaluated using the Davies Bouldin Index to find out the most optimal number of clusters. Experiments were carried out using RapidMiner. Rapidminer is a software platform that provides an incorporated environment for data mining, predictive analysis and is used for firms, commercial applications and also for exploring, training and learning[25]. Rapidminer can be used to determine the quality classification of rice and small and medium enterprises[26], [27]. Rapidminer used to build a model using K-Means clustering and the Davies Bouldin Index as shown in Figure 2 and the evaluation results was seen in Table 5.

Table 5. The Result of Davies Bouldin Index

k=	Davies Bouldin Value	k=	Davies Bouldin Value
2	2,980	14	3,163
3	3,402	15	3,061
4	3,466	16	3,111
5	3,936	17	2,379
6	3,291	18	2,825
7	3,831	19	2,728
9	3,133	20	2,829
9	3,621	21	2,921
10	3,795	22	2,216
11	3,392	23	2,262
12	3,193	24	2,754
13	2,809	25	2,646

Table 5 showed that the results of the evaluation using the Davies Bouldin Index, the largest value was 3,936 with k = 5, while the smallest Davies Bouldin Index value was 2,216 with k = 22. The smaller the Davies Bouldin Index value indicated that the number of clusters was getting better (optimal) in the experiment. This meant that the best number of clusters was at k=22.

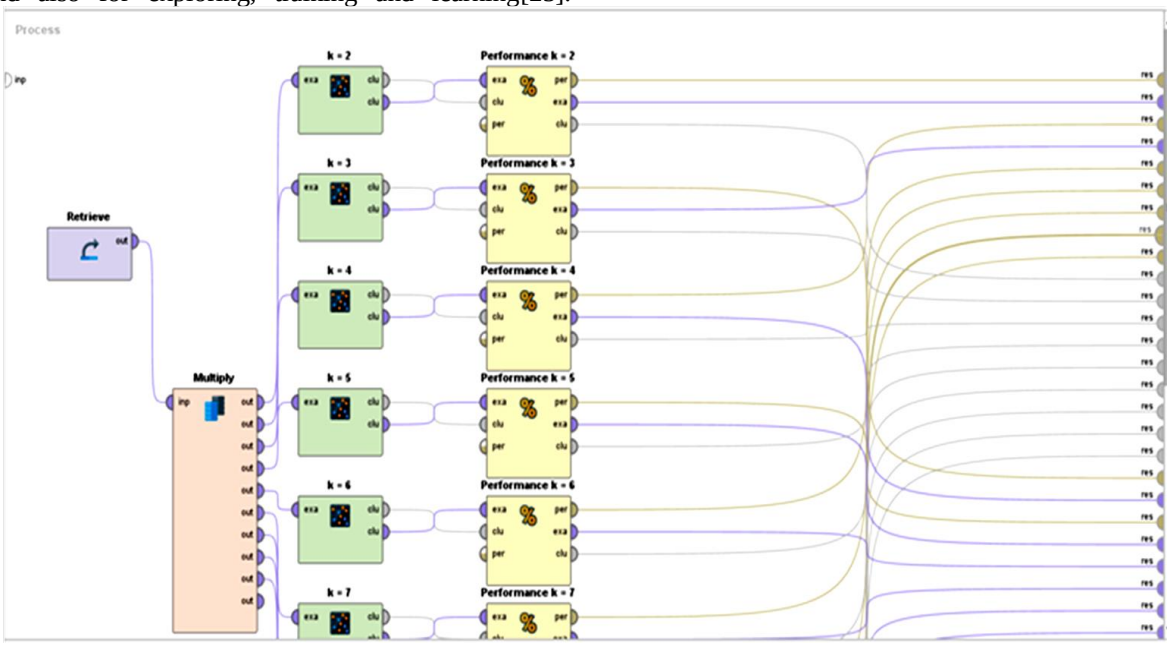

Figure 2. Model Clustering K-Means and Performance Evaluation

Figure 2 was a model for the clustering process using the K-Means algorithm and evaluation using the Davies Bouldin Index. Determination of the value of "k" starting from k=2 to k=25 to determine the effect of the number of clusters on the results of the Davies Bouldin Index. The evaluation results using the Davies Bouldin Index found that there was a significant change in value, namely in the number of clusters 2, 3, and 4; the average

DOI: https://doi.org/10.29207/resti.v7i4.4543

Davies Bouldin Index value was above 3. Meanwhile, for the number of clusters 5-25, the average Davies Bouldin Index value was above 2. However, the most optimal number of clusters in this experiment is shown in the number of clusters 22, with the smallest value of 2,216.

Furthermore, this study evaluated the quality of the ranking generated by the WP-Rank method based on the cluster formed using NDCG. The results of the evaluation was seen in Figure 3.

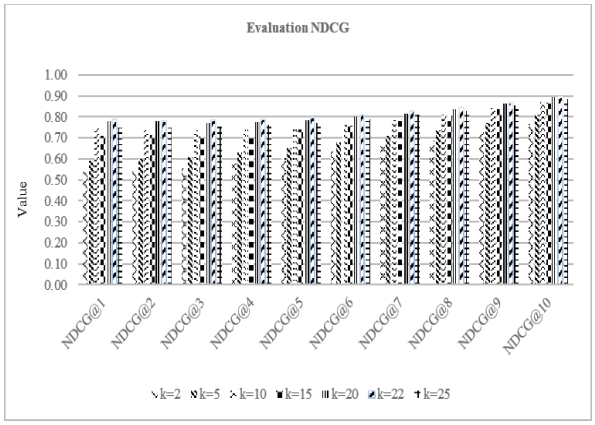

Figure 3. The Evaluation of NDCG on WP-Rank Implementation

Figure 3 showed that the results of the NDCG evaluation of the ranking quality was generated by the WP-Rank method, based on the number of clusters formed from the K-Means algorithm. In this experiment, the number of "k" set was 2-25 clusters, and samples were taken at k=2, k-5, k=10, k=15, k=20, k=22, and k=25. Based on the experimental, results showed that there was an increase in the value of NDCG from NDCG 1-10 at k=2 to k=22. The average NDCG values were 0.63, 0.68, 0.78, 0.76, 0.81, and 0.82, respectively. Meanwhile, for k=25, there was a decrease in the average value of NDCG, which was 0.79. There was a significant difference at k=2 and k=5 when compared to k=22, namely 0.19 and 0.14. Meanwhile, for k=10, k=15, k=20, they are 0.04, 0.06, and 0.01. However, there was a decrease in the average value of NDCG at k=25 although it was not too significant, namely 0.02. The average value of NDCG was k=22. It showed that the highest value because k=22 was the most optimal number of clusters according to the results of the evaluation using the Davies Bouldin Index. In addition, it showed that the optimal number of clusters affects the quality of the recommendations as indicated by a better NDCG value.

The next evaluation was running time by calculating the real time the process takes to execute the input data to produce a ranking. The results of the running time evaluation was seen in Figure 4.

Figure 4 showed the results of the running time evaluation for k=2 to k=25. The average time required for execution was 1.923, 0.507, 0.239, 0.165, 0.145, 0.164, and 0.110 second. There was a significant decrease in the execution time required at k=2 and k=5, with a difference of 1.42 seconds. Meanwhile, at k=5 and k=10, there was also a decrease in execution time, namely 0.27 seconds, but for k=15, k=20, k=22, k=25, the execution time required was relatively stable, namely 0.15 seconds on average. Based on this, it concluded that the implementation of the K-Means clustering algorithm was able to overcome the scalability problem which is one of the problems faced by Collaborative Filtering. The K-Means algorithm partitions data so that the execution process was faster and produces recommendations with better quality.

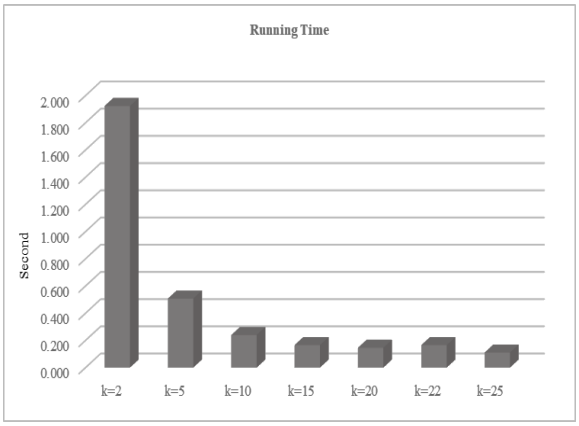

Figure 4. The Result of Running Time Evaluation

4. Conclusion

This study implemented the K-Means algorithm and the WP-Rank method to overcome scalability and sparsity problems. Based on the experiment it concluded that:

The scalability problem in Collaborative Filtering can be overcome by implementing the K-Means algorithm, namely by partitioning the data into several clusters. The experimental results showed that the optimal number of clusters was k=22. It was indicated by the results of the evaluation using the Davies Bouldin Index with the smallest value of 2.216. In addition, the results of the evaluation of running time were faster with an average execution time of 0.15 second.

The combination of clustering and ranking based on the approaches of the WP-Rank method overcome the sparsity problem, because clustering it was able to reduce the dimensions of the rating matrix and empty data. Furthermore, the aggregation process of the WP-Rank method was to produce quality recommendations as indicated by the average value of NDCG at k = 22 of 0.82.

The further study will compare it with other clustering algorithms to determine the effect on the quality of recommendations.

DOI: https://doi.org/10.29207/resti.v6iX.xxx

Acknowledgment

This study was supported by research grants organized by the Directorate of Research, Technology, and Community Service (DRTPM), Master Thesis Research Scheme (PTM) with number 1367/LL2/PG/2022.

References

[1] S. Wu, F. Sun, W. Zhang, X. Xie, and B. Cui, "Graph Neural Networks in Recommender Systems: A Survey," *ACM Comput. Surv.*, vol. 55, no. 5, Dec. 2022, doi: 10.1145/3535101.

[2] M. H. Mohamed, M. H. Khafagy, and M. H. Ibrahim, "Recommender Systems Challenges and Solutions Survey," in *2019 International Conference on Innovative Trends in Computer Engineering (ITCE)*, Feb. 2019, pp. 149–155. doi: 10.1109/ITCE.2019.8646645.

[3] S. and B. R. Koren Yehuda and Rendle, "Advances in Collaborative Filtering," in *Recommender Systems Handbook*, L. and S. B. Ricci Francesco and Rokach, Ed., New York, NY: Springer US, 2022, pp. 91–142. doi: 10.1007/978-1-0716-2197-4_3.

[4] M. M. Goyani and N. Chaurasiya, "A review of movie recommendation system: Limitations, Survey and Challenges," *Electronic Letters on Computer Vision and Image Analysis*, vol. 19, pp. 18–37, 2020.

[5] N. Ifada, T. F. Rahman, and M. K. Sophan, "Comparing collaborative filtering and hybrid based approaches for movie recommendation," in *Proceeding - 6th Information Technology International Seminar, ITIS 2020*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020, pp. 219–223. doi: 10.1109/ITIS50118.2020.9321014.

[6] K. Nanthini M. and Pradeep Mohan Kumar, "Cold start and Data Sparsity Problems in Recommender System: A Concise Review," in *International Conference on Innovative Computing and Communications*, A. and B. S. and H. A. E. and A. S. and J. A. Gupta Deepak and Khanna, Ed., Singapore: Springer Nature Singapore, 2023, pp. 107–118.

[7] J. Das, T. H. Academy, M. Banerjee, T. H. Academy, and S. Majumder, "Scalable Recommendations using Clustering based Collaborative Filtering," in *International Conference on Information Technology (ICIT)*, 2019, pp. 1–6. doi: 10.1109/ICIT48102.2019.00056.

[8] L. Wang, X. Zhang, T. Wang, S. Wan, G. Srivastava, and S. Member, "Diversified and Scalable Service Recommendation With Accuracy Guarantee," *IEEE Trans Comput Soc Syst*, pp. 1–12, 2020, doi: 10.1109/TCSS.2020.3007812.

[9] Z. Zhao, Y. Sheng, M. Zhu, and A. J. Wang, "A Memory-Efficient Approach to the Scalability of Recommender System With Hit Improvement," *IEEE Access*, vol. 6, pp. 67070–67081, 2018, doi: 10.1109/ACCESS.2018.2878808.

[10] N. Ifada, "Employing Sparsity Removal Approach and Fuzzy C-Means Clustering Technique on a Movie Recommendation System," in *International Confrence on Computer Engineering, Network and Intelligent Multimedia (CENIM)*, 2018, pp. 329–334. doi: 10.1109/CENIM.2018.8711270.

[9] D. Andra and A. Baizal, "E-commerce Recommender System Using PCA and K-Means Clustering," *JURNAL RESTI (Rekayasa Sistem dan Teknologi Informasi*, vol. 6, no. 158, pp. 57–63, 2022, doi: https://doi.org/10.29207/resti.v6i1.3782.

[10] M. I. Ardimansyah, A. F. Huda, and Z. K. A. Baizal, "Preprocessing Matrix Factorization for Solving Data Sparsity on Memory-Based Collaborative Filtering," in *3rd International Conference on Science in Information Technology (ICSITech) Preprocessing*, 2017, pp. 521–525.

[11] S. Lestari, T. B. Adji, and A. E. Permanasari, "WP-Rank : Rank Aggregation based Collaborative Filtering Method in Recommender System," *Internationa Journal of Engineering & Technology*, vol. 7, pp. 193–197, 2018.

[12] S. Lestari, R. Kurniawan, and D. Linda, "Porat Rank to Improve Performance Recommendation System," in *Proceedings of the 1st International Conference on Electronics, Biomedical Engineering, and Health Informatics, Lecture Notes in Electrical Engineering 746*, 2021, pp. 1–14.

[13] M. Mughnyanti, S. Efendi, and M. Zarlis, "Analysis of determining centroid clustering x-means algorithm with davies-bouldin index evaluation," in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Jan. 2020. doi: 10.1088/1757-899X/725/1/012128.

[14] Institute of Electrical and Electronics Engineers, *2nd International Conference on Electrical, Computer and Communication Engineering (ECCE) 07-09 February 2019, Cox's Bazar, Bangladesh : conference digest*.

[15] M. Jumarlis and Mirfan, "Detecting Diseases on Clove Leaves Using GLCM and Clustering K-Means," *JURNAL RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 4, pp. 624–631, 2022, doi: https://doi.org/10.29207/resti.v6i4.4033.

[16] A. K. Singh, S. Mittal, P. Malhotra, and Y. V. Srivastava, "Clustering Evaluation by Davies-Bouldin Index(DBI) in Cereal data using K-Means," in *Proceedings of the 4th International Conference on Computing Methodologies and Communication, ICCMC 2020*, Institute of Electrical and Electronics Engineers Inc., Mar. 2020, pp. 306–310. doi: 10.1109/ICCMC48092.2020.ICCMC-00057.

[17] A. K. Singh, S. Mittal, Y. V. Srivastava, and P. Malhotra, "Clustering Evaluation by Davies-Bouldin Index (DBI) in Cereal data using K-Means," in *International Conference on Computing Methodologies and Communication (ICCMC)*, 2020, pp. 306–310. doi: 10.1109/ICCMC48092.2020.ICCMC-00057.

[18] H. Santoso and H. Magdalena, "Improved K-Means Algorithm on Home Industry Data Clustering in the Province of Bangka Belitung," in *International Conference on Smart Technology and Applications (ICoSTA)*, 2020, pp. 1–6. doi: 10.1109/ICoSTA48221.2020.1570598913.

[19] Z.-H. Qiu, Q. Hu, Y. Zhong, L. Zhang, and T. Yang, "Large-scale Stochastic Optimization of NDCG Surrogates for Deep Learning with Provable Convergence," Feb. 2022, [Online]. Available: http://arxiv.org/abs/2202.12183

[20] L. Gienapp, M. Fröbe, M. Hagen, and M. Potthast, "The Impact of Negative Relevance Judgments on NDCG," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, in CIKM '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 2037–2040. doi: 10.1145/3340531.3412123.

[21] Y. Wang, Y. Xiao, and J. Qiu, "Bi-Rank: A New Bi-Directional Ranking Method for Goods Selection," in *Proceedings - 2020 Chinese Automation Congress, CAC 2020*, Institute of Electrical and Electronics Engineers Inc., Nov. 2020, pp. 7566–7569. doi: 10.1109/CAC51589.2020.9327762.

[22] N. Ifada, T. F. Rahman, and M. K. Sophan, "Comparing Collaborative Filtering and Hybrid based Approaches for Movie Recommendation," in *Information Technology International Seminar (ITIS)*, 2020, pp. 219–223.

[23] L. Niu, Y. A. N. Peng, and Y. Liu, "Deep Recommendation Model Combining Long - and Short-Term Interest Preferences," *IEEE Access*, vol. 9, pp. 166455–166464, 2021, doi: 10.1109/ACCESS.2021.3135983.

[24] J. Chen, H. Wang, and Z. Yan, "Evolutionary heterogeneous clustering for rating prediction based on user collaborative fi ltering ☆," *Swarm Evol Comput*, vol. 38, no. April 2017, pp. 35–41, 2018, doi: 10.1016/j.swevo.2017.05.008.

[25] M. M. Shabtari, V. Kumar Shukla, H. Singh, and I. Nanda, "Analyzing PIMA Indian Diabetes Dataset through Data Mining Tool 'RapidMiner,'" in *2021 International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE 2021*, Institute of Electrical and Electronics Engineers Inc., Mar. 2021, pp. 560–574. doi: 10.1109/ICACITE51222.2021.9404741.

[26] S. Lestari, Yulmaini, Aswin, Sylvia, Y. A. Pratama, and Sulyono, "Implementation of the C4.5 algorithm for micro,

small, and medium enterprises classification," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 6, pp. 6707–6715, Dec. 2022, doi: 10.11591/ijece.v12i6.pp6707-6715.

[27] M. R. Fahlevi, D. R. D. Putri, F. A. Putri, M. Rahman, L. Sipahutar, and M. Muhatri, "Determination of Rice Quality Using the K-Means Clustering Method," in *2020 2nd International Conference on Cybernetics and Intelligent System, ICORIS 2020*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020. doi: 10.1109/ICORIS50180.2020.9320839.

DOI: https://doi.org/10.29207/resti.v6iX.xxx