RankPro-M Method to Alleviate the Sparsity Problem in Collaborative Filtering

Sri Lestari[1,*], Yulmaini[2], Suhendro Yusuf Irianto[3], Hari Sabita[4]

[1,2,3,4]*Faculty of Computer Science, Institute of Informatics and Business Darmajaya, Indonesia*

(Received: September 15, 2025; Revised: November 12, 2025; Accepted: February 15, 2026; Available online: March 17, 2026)

Abstract

The rapid shift from conventional commerce to online platforms has been driven by evolving consumer behavior that demands fast, accurate, and personalized services. Consequently, e-commerce has become a primary channel for product marketing and service delivery without temporal or spatial constraints. However, the continuous expansion of e-commerce platforms has led to a substantial increase in both the volume and diversity of available products, thereby complicating the task of delivering personalized recommendations aligned with user preferences. Recommender systems offer an effective solution to this challenge, with Collaborative Filtering (CF) being among the most widely adopted techniques. Despite its popularity, CF suffers from a critical limitation known as the data sparsity problem, which adversely affects recommendation accuracy and system reliability. This study proposes RankPro-M, a ranking-oriented imputation approach designed to mitigate the impact of sparsity in recommender systems. RankPro-M operates by identifying items with high rating frequency and imputing missing ratings using mode values as representations of dominant user preferences. The imputed rating matrix is subsequently processed through ranking aggregation mechanisms (Borda, Copeland, and WP-Rank) to generate item recommendations. Experimental results demonstrate that the application of RankPro-M consistently improves recommendation quality, as indicated by increased Normalized Discounted Cumulative Gain (NDCG) values across multiple evaluation scenarios. These findings confirm that RankPro-M effectively addresses data sparsity and enhances the performance of ranking-based recommender systems.

Keywords: Recommendation System, Collaborative Filtering, Sparsity, RankPro-M

1. Introduction

The continuous growth of digital commerce ecosystems has significantly increased the complexity of user decision-making processes, primarily due to the rapid expansion in both the volume and diversity of available products. In such data-rich environments, users are frequently exposed to an overwhelming number of alternatives, which can impede efficient product discovery and negatively affect overall user satisfaction. This situation highlights the growing need for intelligent decision-support mechanisms that are capable of filtering large amounts of information and facilitating personalized item selection in a scalable manner[1], [2].

Within e-commerce platforms, recommendation systems serve a strategic function in addressing these challenges by acting as an intermediary between users and extensive product catalogs. By presenting personalized suggestions, these systems aim to reduce users' cognitive load while simultaneously enabling service providers to deliver more targeted and relevant offerings. As product catalogs continue to expand, the performance of recommendation mechanisms becomes increasingly reliant on their ability to accurately model user preferences and effectively operate under conditions of limited and uneven interaction data [3], [4].

Among the various recommendation techniques, collaborative filtering remains one of the most widely adopted approaches due to its domain-independent nature and its ability to infer user preferences from collective behavioral patterns. Despite these advantages, collaborative filtering is highly sensitive to data sparsity, a common condition in real-world applications where user–item interactions are inherently limited. Such sparsity weakens similarity computations, restricts neighborhood formation, and ultimately compromises the accuracy and stability of ranking-based

***Corresponding author: Sri Lestari (srilestari@darmajaya.ac.id)**
DOI: https://doi.org/10.47738/jads.v7i2.1173

recommendations [5], [6]. To address this limitation, this study introduces RankPro-M, a sparsity-aware recommendation approach that integrates product ranking with a mode-based imputation strategy. By prioritizing frequently rated items within user groups and utilizing statistically dominant rating values that refer to the mode value to fill missing entries. The proposed method aims to enhance recommendation robustness while maintaining computational simplicity in sparse data environments.

2. Literature Review

2.1. Recommendation System

The recommendation system works by finding the most relevant information taken from a large amount of data to produce recommendations that are in accordance with the user's interests [7]. The most relevant recommendations continue to be explored with various approaches, such as content-based filtering [8], [9], demographic filtering [10], [11], collaborative filtering [10], [12], and hybrid filtering [13], [14], [15].

Content-based filtering [16] uses product content and user tendency profiles to produce recommendations. The way this method works is by analyzing text, images and sounds to measure the similarity between products. Besides studying user specialization on products in the past to find out the current trends of users. Based on the similarity of products and preferences of user interests, the product recommendations that are most suitable for user specialization are generated. It should be noted that this description is conceptual and based on previous literature [16], and does not represent an implementation carried out in this study. This section provides theoretical background for understanding content-based filtering. Content-based filtering has several advantages, which can build profiles based on assessments provided by each active user without calculating the influence/interaction with other users (user independence).

Content-based filtering provides transparency to the details of how the recommendation system works to users, in bringing up relevant products based on content features. This method is able to recommend products that have not been rated by each user. But it is very difficult to produce a product profile, because product description information is generally unstructured. Therefore, in preparing a product profile, the required preprocessing steps to extract product descriptions in order to obtain relevant and structured information. Content-based filtering also experiences excessive specialization problems because it supports the same type of product. Users will always get a product similar to what was previously recommended. Another problem is that the system cannot provide reliable recommendations to new users, because it requires searching first on user preferences [17], [18].

Demographic filtering [16],[19] produces recommendations by utilizing information on the user's demographic profile consisting of age, gender, country and other attributes. This is because individuals with similar user attributes will have the same preferences. The way to do that is by grouping users according to their attributes. Based on these groups, users generally have a similar preference for a product as a reference for recommendations. The demographic filtering method will provide significant power to other recommendation systems as a component of a hybrid or ensemble model. However, standing alone does not give the best results [20]. For example, recent studies have shown that combining collaborative filtering with user demographic attributes such as age and gender enables the system to create segmented user profiles, which improves recommendation accuracy by providing more personalized suggestions tailored to each user group [21]. This hybrid approach demonstrates how demographic information can complement other recommendation methods to enhance the overall performance of the system.

Hybrid filtering is done by combining several existing methods, such as a combination of content-based filtering with collaborative filtering. Hybrid filtering has the advantage of being able to overcome problems in other methods such as cold start, sparsity and scalability. However, hybrid filtering is complicated and expensive in its implementation [20], [22], [23].

Collaborative Filtering (CF) works by analyzing rating data patterns to make predictions [24]. Prediction is obtained after several stages, namely collecting and analyzing large amounts of information about the user's behavior, activities, and preferences. The results of the analysis are then used to predict what the user likes based on the similarity with other users. Collaborative filtering has several advantages including being easy to implement and being able to filter any type of information or items without having to analyze the comments of users. Besides collaborative filtering produces high quality recommendations rather than recommendation systems based on content and demographics [25], [26]. This causes

the popular collaborative filtering method and is widely used in recommendation systems [27]. But it faces crucial problems, namely cold start, sparsity and scalability. Cold start is a condition where new users who have never given a rating of a product, so the information obtained for user specialization is difficult to know. If the direction of specialization of the user is unknown then it cannot recommend the product [28]. Sparsity is a rare data condition, it is caused by the majority of users who do not rate products. This condition results in a low similarity value so that it cannot produce accurate recommendations [29]. Scalability is a condition where recommendation systems need to increase their computational power to offer accurate and timely recommendations even with large-scale data conditions [28]. The focus of this research is handling the problem of sparsity in collaborative filtering.

2.2. Sparsity

Some research has been done to solve the problem of sparsity such as that done by Tang and Tong who developed the BordaRank method [30]. This method consists of two steps: filling sparse data with CF items and aggregating using the Borda method to produce product rankings. Furthermore, Z. Lin, et al., in their research used the K-medoid algorithm to solve the sparsity problem [26]. Sharifi, et al, proposed on Negative Matrix Factorization (NMF) methods in pre-processing data to solve sparsity problems [27]. With this method produces better predictions than the original data.

2.3. K-Means

Clustering algorithm is widely used in the recommendation system, because it makes it easy to identify users who have similar preferences. One of the important clustering algorithms in data mining is K-means, so many researchers use this method [31]. As did Xue et.al. who uses the K-means algorithm in collaborative filtering to facilitate the filling of unrated data according to the cluster [32]. Next, Dakhel and Mahdawi who use the K-means algorithm to categorize users based on their interests [31]. Min and Shuzhen, use the K-means algorithm because the algorithm is simple and the process is fast [33].

How the K-means algorithm works is to divide a number of objects into partitions based on certain categories, by looking at the midpoint given. The distance of the object with the closest midpoint is used to classify the object, so that the object becomes a member of the formed category. Euclidean Distance Space is one equation for calculating distance, by knowing the shortest distance between two points. Distance calculation using Equation (1).

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

In accordance with Equation (1) the K-means algorithm begins by determining the number of clusters (k) and proceed by determining the centroid (cj) of each cluster randomly. The next step is to calculate the distance of each object (xi) to the centroid (cj) denoted by D (xi, cj). This process is carried out for all clusters from the first cluster to the k-cluster. After calculating the distance, the next step is to group objects based on the nearest cetroid. The next step is to update the centroid, by calculating the average distance of all objects to the centroid. The process will be repeated until there is no data to move the cluster.

3. Methodology

This research will propose a new method that we called RankPro-M, the workings of this method is to use the product ranking approach and the Mode function. Product ranking is used to filter out products that are in great demand by users by taking Top-N products, based on these data, it is then used as a reference in filling in many sparse data. To fill in the blank data is done by searching for ratings that often appear using the mode function, the rating is then used to fill in data that is still empty. In this way, it is expected that the quality of the recommendations produced will be better so that it is in accordance with the user's specialization. The stages of the research can be seen in figure 1.

This methodology is designed to develop and evaluate a RankPro-M based recommendation approach for addressing data sparsity. The research workflow includes dataset collection, data pre-processing, user clustering, user–item matrix construction, ranking-based recommendation, and performance evaluation using the NDCG metric, as illustrated in figure 1.

3.1. Dataset collections

This research uses the dataset of MovieLens, which can be accessed via the GroupLens website (https://grouplens.org/datasets/movielens/). DataLens dataset is a datasetopen source that can be used freely but must follow the rules set by GroupLens, which is a laboratory research at the University of Minnesota, USA.

The dataset used in this experiment is MovieLens 100k. Dataset100k has around 1,682 movies, is used by 943 users and has 100,000 rating. The dataset has a general characteristic that is the presence of user demographic information (user id, age, gender, occupation, and zipcode) and 19 genres. In addition, each user at least gives a rating of 20 movies containing 93.7% sparsity, this is because many users do not give ratings to movies.

3.2. Pre-processing Dataset

Pre-processing the dataset aims to eliminate data that is not used in this study. For example, in the demographic information, the user uses only id_user and age, so other attributes are removed. Demographic attributes were limited to user ID and age to maintain model simplicity and ensure compatibility with the numerically based K-means algorithm. Age was selected because it is relevant to variations in user preferences, whereas other demographic attributes are categorical and may increase model complexity. This restriction improves the efficiency and stability of the clustering process. In addition to movies that are not rated by all users are not involved in the calculation process. Although the MovieLens dataset is valid data and is updated regularly, pre-processing of the dataset still needs to be done to improve the performance of the recommendation system.

3.3. Clustering Dataset

In this study, the clustering process employs the K-means algorithm using user demographic information, specifically age. User clustering is performed based on Equation (1), which generates groups of users with similar characteristics and preferences."

3.4. Proposed Method

This research proposes a new method, called RankPro-M, which combines product-ranking approaches with mode-based imputation to address the problem of data sparsity. RankPro-M operates in several stages: (1) the process begins by constructing a user–item matrix based on the available rating data. (2) Next, the number of ratings for each item is calculated to represent its popularity. (3) The items are then sorted in descending order according to the number of ratings. (4) From the sorted list, the Top-N items with the highest number of ratings are selected. (5) For each selected item, the most frequently occurring rating value (mode) is identified as the representative value. (6) This value is subsequently used to fill in all missing ratings for the corresponding item, resulting in a more complete rating matrix that is ready for further analysis.

The Top-N strategy is used to focus the imputation process on products with sufficient rating density, resulting in more stable and reliable mode estimates. The value of N is treated as an adjustable parameter and is determined empirically through initial testing, taking into account the balance between data coverage and robustness to noise arising from sparse data.

Among various statistical imputation techniques, this study adopts a mode-based approach owing to its robustness in handling sparse and discrete rating data. Mean- and median-based imputations are susceptible to distributional skewness and may yield interpolated values that lack correspondence with actual user-assigned ratings. In contrast, the mode captures the most frequently observed rating, thereby representing dominant user preference while preserving semantic validity within the predefined rating scale, which makes it particularly suitable for alleviating sparsity in collaborative filtering frameworks. For example, in a sparse item where ratings are limited to extreme values (e.g., 1 and 5), mean- or median-based imputation may yield interpolated values that do not reflect actual user preferences, whereas mode-based imputation retains the most dominant observed rating.

3.5. Similarity Measure

Cosine similarity is used in this study to calculate the proximity between users [34], [35], [36], [37]. The details of Cosine similarity can be found in Equation 2. Based on the similarity values, the Top N users were taken. The data was then aggregated using the WPRank, Borda, and Copeland methods.

 ISSN 2723-6471

$$s(u,v) = cos\left(\vec{R}(u,*)\right), \left(\vec{R}(v,*)\right) = \frac{\vec{R}(u,*).\vec{R}(v,*)}{\left\|\vec{R}(u,*)\right\| * \left\|\vec{R}(v,*)\right\|} \quad (2)$$

The cosine value is the angle formed by the row vectors of user rating u and user v, denoted by $\cos\left(\vec{R}(u,*)\right),\left(\vec{R}(v,*)\right)$. Meanwhile, $\vec{R}(u,*)$ is a row vector consisted of rating values of user u.

3.6. Recommendation Product

Based on the ranking results obtained through the application of Cosine Similarity, RankPro-M, and ranking methods (Copeland, Borda, WP-Rank), the system generates product recommendations that match the user's preferences. The ranking process ensures that the recommended products are prioritized according to relevance, providing users with more accurate and personalized outcomes.

3.7. Evaluation

An evaluation is carried out to determine the quality of the recommendations. This study uses NDCG. NDCG accounts for the positions of items within the recommendation list by assigning higher weights to relevant items appearing at higher ranks, making it well suited for evaluating ranking-based recommendation systems such as RankPro-M.

Figure 1 illustrates the research workflow, which begins with the collection of movie rating data from the MovieLens dataset. The next stage involves data preprocessing to ensure data cleanliness and consistency, followed by user clustering using the K-Means algorithm to construct groups of users with similar preference profiles. Based on the clustering results, a user–item interaction matrix of size 20 users × 50 movies is constructed to represent user–item relationships.

Subsequently, three experimental scenarios are applied to this matrix: direct utilization of the original matrix, the application of Cosine Similarity, and the proposed RankPro-M method. The outputs of these three approaches are then processed using three ranking aggregation techniques Copeland, Borda, and WP-Rank to generate ordered recommendation lists. The final stage involves evaluating recommendation quality using the Normalized Discounted Cumulative Gain (NDCG) metric at ten evaluation cut-off points. This metric assesses recommendation relevance based on item positions within the ranking, thereby enabling a comparative analysis of the effectiveness of each method.

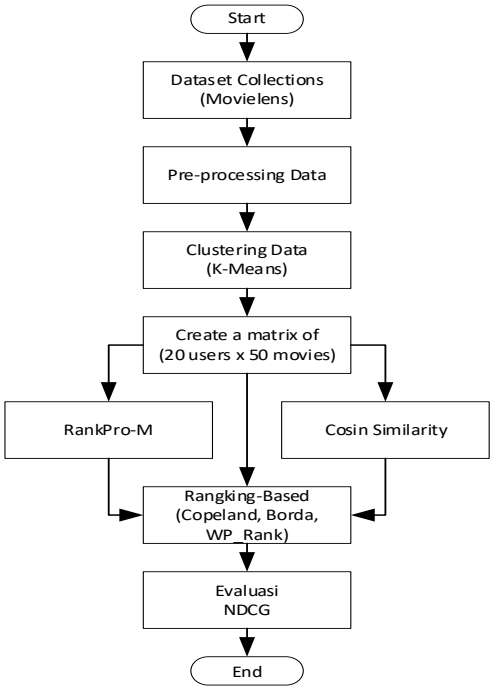

Figure 1. Flow Chart of Research Activities

4. Results and Discussion

Initial experiments were carried out on sample data with a size of 20 users x 50 movies. This is done to simplify the process of calculating RankPro-M. RankPro-M is used to fill in empty data (sparsity). Sparsity is one of the problems faced by Collaborative Filtering, it will affect the quality of the resulting recommendations. Therefore, this study will try to overcome this problem by proposing a method that we call RankPro-M. The workings of the RankPro-M method are:

Algorithm 1. RankPro-M

Input: UserMovieRatings (matrix U × M, with U = number of users, M = number of movies)

Output: CompletedRatings (matrix with missing values filled)

Begin

// Step 1: Form user × movie matrix

Matrix ← UserMovieRatings

// Step 2: Calculate the number of ratings for each movie

For each movie m_i in Matrix:

CountRatings[m_i] ← Σ_{u=1}^{U} I(r_{u,i} ≠ Ø)

// I(condition) = 1 if condition is true, 0 otherwise

// Step 3: Sort movies by number of ratings (descending)

SortedMovies ← sort movies by CountRatings[m_i] descending

// Step 4: Select Top-N movies

TopNMovies ← first N movies from SortedMovies

// Step 5: Calculate mode rating for each movie

For each movie m_i in TopNMovies:

ModeRating[m_i] ← argmax_{r ∈ R} freq(r, m_i)

// freq(r, m_i) = frequency of rating r in movie m_i

// Step 6: Fill missing ratings using mode

For each movie m_i in TopNMovies:

For each user u in Matrix:

If r_{u,i} = Ø:

r_{u,i} ← ModeRating[m_i]

CompletedRatings ← Matrix

End

The experiments we conducted employed several methods, namely K-Means clustering, Cosine Similarity, RankPro-M for imputation, and three ranking methods (Copeland, Borda, WP-Rank). K-Means was used to cluster users based on age. From this clustering, a dataset of 20 users × 50 movies was taken as a matrix for initial testing. Three scenarios were applied to the matrix: 1) Using the 20 × 50 user–movie matrix directly. 2) Applying Cosine Similarity to obtain the Top-N users, in this case the top 10, resulting in a 10 × 50 matrix. 3) Using a 20 × 50 matrix, it is then processed using RankPro-M to generate the imputed matrix. Based on the matrices from these three scenarios, ranking was performed using Copeland, Borda, and WP-Rank, and the results were evaluated using NDCG. The ranking results are then evaluated using the NDCG, as shown in figure 2 to figure 4.

The use of a 20 users × 50 items rating matrix in this study is intended to support the initial evaluation stage and to facilitate a clearer illustration of the working mechanism of the proposed RankPro-M method. This limited data subset

enables a more focused analysis of the imputation process and ranking behavior under controlled sparsity conditions, thereby improving result interpretability. Nevertheless, we acknowledge that the relatively small dataset size limits the generalizability of the findings. Therefore, the results obtained at this stage should be regarded as preliminary evidence of the effectiveness of RankPro-M. Future research will evaluate the proposed method on larger and more diverse datasets to assess its scalability, robustness, and applicability in real-world settings.

Figure 2 presents a comparison of NDCG performance for three ranking methods—Copeland, Borda, and WP_Rank—across ten evaluation levels (NDCG@1–NDCG@10) using untreated data. The visual results indicate that WP_Rank consistently achieves the highest NDCG scores at all evaluation levels. On average, WP_Rank attains an NDCG value of 0.70, outperforming Copeland (0.57) and Borda (0.53). This advantage becomes more pronounced at higher evaluation levels (NDCG@7–NDCG@10), with performance gaps ranging from 0.08 to 0.18, indicating that WP_Rank delivers more stable and effective ranking quality, particularly in capturing relevant items at deeper positions within the recommendation list.

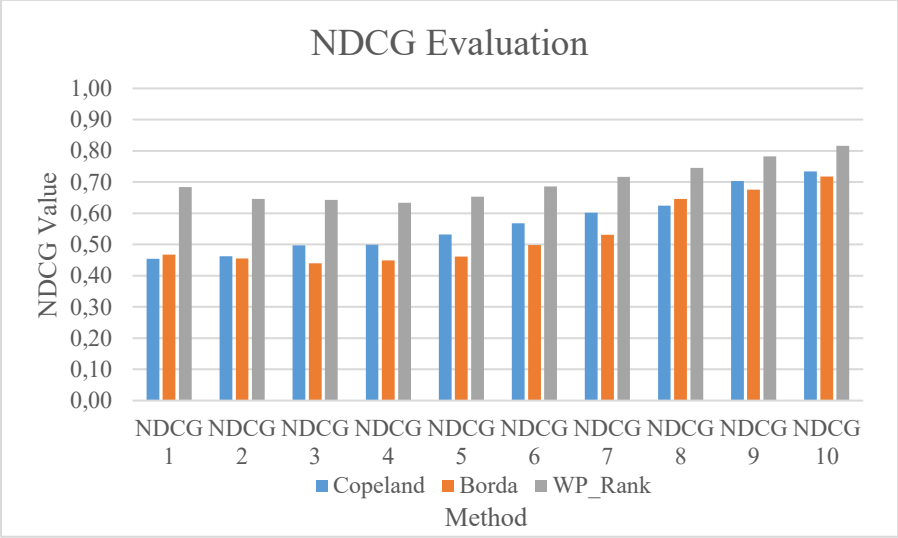

Figure 2. NDCG Evaluation Results on Untreated Data

Figure 3 presents the NDCG performance comparison of Copeland, Borda, and WP_Rank across ten evaluation levels (NDCG@1–NDCG@10) using data processed with Cosine Similarity. The results indicate that both Borda and WP_Rank consistently outperform Copeland at all evaluation levels. Quantitatively, the average NDCG value achieved by Copeland is 0.56, while Borda and WP_Rank each reach an average NDCG of 0.76. This performance improvement demonstrates that the incorporation of cosine similarity enhances ranking effectiveness, particularly for the Borda and WP_Rank methods, by producing more accurate and stable recommendation rankings.

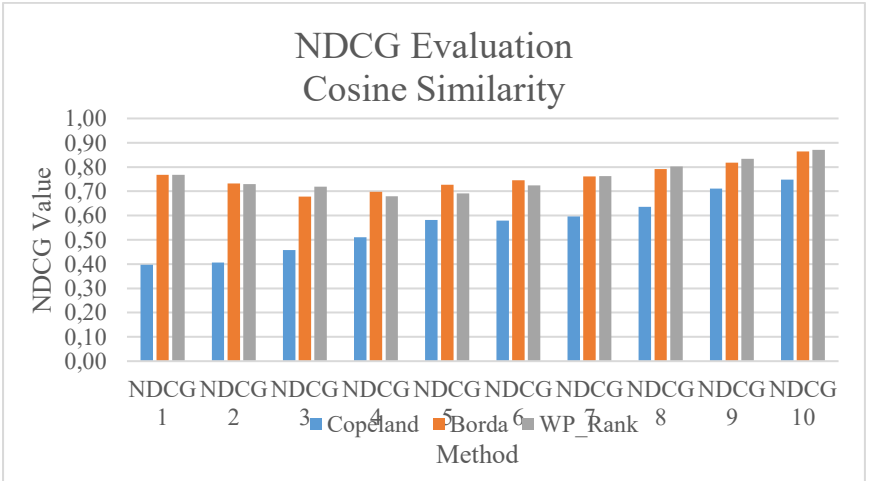

Figure 3. NDCG Evaluation Result on Data from Cosine Similarity Process

Figure 4 The bar chart illustrates the performance comparison of three ranking methods, Copeland, Borda, and WP_Rank, across 10 NDCG metrics in the implementation of Scenario 3. Based on experiments, it shows that the average NDCG value is for the Copeland method which is 0.85, for the Borda method it is 0.88, and for the WP_Rank method it is 0.92. The WP_Rank method is superior to the Borda and Copeland methods. Furthermore, this shows an increase in the NDCG value for each method, so that with RankPro-M, it is proven to improve the quality of recommendations and to overcome the sparsity problem.

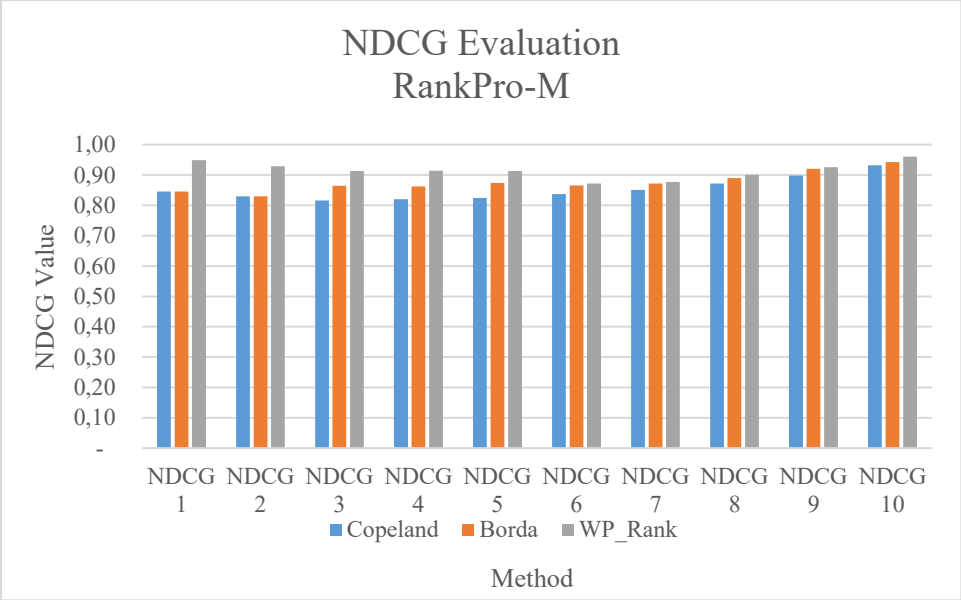

Figure 4. NDCG Evaluation Result on Data with RankPro-M Treatments

WP_Rank consistently achieves the highest NDCG values compared to Copeland and Borda across all evaluation levels after the application of RankPro-M. This superiority arises from the alignment between the WP_Rank mechanism and the characteristics of RankPro-M, which employs mode-based imputation for missing values. The imputation process stabilizes the rating distribution and reduces noise caused by data sparsity, thereby strengthening dominant user preference signals. Under these conditions, the weighted aggregation mechanism of WP_Rank is able to exploit the stabilized information more effectively than the pairwise comparison approach of Copeland or the simple aggregation strategy of Borda. Consequently, WP_Rank better preserves ranking consistency and delivers higher recommendation quality, as reflected by improved NDCG values across all cut-offs.

A one-way ANOVA was conducted to analyze differences in NDCG values across three experimental scenarios using the Copeland, Borda, and WP-Rank methods. For the Copeland method, the results yielded an F-value of 33.34 with $p < 0.001$, indicating statistically significant differences among the scenarios. Descriptively, Scenario 3 consistently achieved higher NDCG values than Scenario 1 and Scenario 2, confirming that the application of the RankPro-M approach in Scenario 3 leads to a significant improvement in recommendation relevance.

Similarly, the Borda method produced an F-value of 58.37 with $p < 0.001$, demonstrating significant performance differences across the three scenarios. Descriptive analysis revealed that Scenario 3 delivered the best performance, with consistently higher NDCG values compared to Scenario 1 and Scenario 2. These findings indicate that scenario variation has a substantial impact on the effectiveness of the Borda method in generating relevant recommendation rankings. For the WP-Rank method, the ANOVA results also showed statistically significant differences ($F = 42.12$; $p < 0.001$). Descriptively, Scenario 3 achieved the highest NDCG values relative to the other scenarios. This result further confirms that applying RankPro-M in Scenario 3 provides the most optimal improvement in recommendation performance.

The analysis revealed statistically significant differences among the three scenarios. Consistently, Scenario 3 achieved the highest NDCG values and differed significantly from both Scenario 1 and Scenario 2. Using the Copeland method, Scenario 3 demonstrated significantly superior performance, while the difference between Scenario 1 and Scenario 2

was not statistically significant. Using the Borda method, all pairwise comparisons exhibited statistically significant differences, with a progressive improvement observed from Scenario 1 to Scenario 2 and further to Scenario 3. Using the WP-Rank method, Scenario 3 again proved to be significantly superior to the other two scenarios, with Scenario 2 outperforming Scenario 1.

The one-way ANOVA results indicate statistically significant differences across the three experimental scenarios for all ranking methods ($p < 0.001$). Post-hoc Tukey HSD analysis reveals that Scenario 3 significantly outperforms Scenarios 1 and 2 for Copeland and WP-Rank, while no significant difference is observed between Scenarios 1 and 2. For the Borda method, all pairwise scenario comparisons are statistically significant. These results confirm that the application of RankPro-M in Scenario 3 leads to a substantial and consistent improvement in recommendation performance. Therefore, it can be concluded that the implementation of the ranking method in Scenario 3 (RankPro-M) provides a statistically significant performance improvement across all evaluated approaches, resulting in more relevant and accurate product recommendations compared to the previous two scenarios.

5. Conclusion

Data sparsity remains a fundamental challenge in Collaborative Filtering, as it directly degrades recommendation accuracy. To address this issue, this study proposes RankPro-M, a ranking-based imputation approach that exploits frequently rated items and mode-based rating aggregation to reduce missing data. Experimental results from this initial evaluation stage indicate that RankPro-M consistently improves recommendation quality, as evidenced by statistically significant increases in NDCG across multiple testing scenarios.

However, the proposed method is still at an early experimental stage and has been validated on a limited dataset. Its reliance on rating frequency patterns may restrict its effectiveness in environments with extremely sparse or highly imbalanced user–item interactions. Despite these limitations, RankPro-M offers a lightweight and practical mechanism that can be integrated into existing recommender systems, particularly in domains where sparse feedback and recurring preference patterns are common. To address these limitations, future work will evaluate RankPro-M on larger and more diverse datasets with varying sparsity levels, explore adaptive imputation strategies to better handle highly imbalanced user–item interactions, and integrate the proposed approach with hybrid recommendation models to improve robustness and real-world applicability.

6. Declarations

6.1. Author Contributions

Conceptualization: S.L.; Methodology: S.L.; Y, S.YI. Software: H.S.; Y.; Validation: S.L., S.YI., and H.S.; Formal Analysis: S.L., and Y.; Investigation: Y.; Resources: Y.; Data Curation: S.L. and H.S; Writing Original Draft Preparation: S.L, S.YI., Y and H.S.; Writing Review and Editing: S.L., and Y; Visualization: H.S.; All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] R. Garapati and M. Chakraborty, "Recommender systems in the digital age: a comprehensive review of methods, challenges, and applications," *Knowl. Inf. Syst*., vol. 67, no. 8, pp. 6367–6411, 2025, doi: 10.1007/s10115-025-02453-y.

[2] S. Huang and Z. Liu, "The Impact of Personalized Recommendation Systems on Consumer Purchase Decisions Under Data Law Frameworks:," *Journal of Organizational and End User Computing*, vol. 37, no. 1, pp. 1-12, 2025, doi: https://doi.org/10.4018/JOEUC.385731.

[3] A. Valencia-Arias, H. Uribe-Bedoya, J. D. González-Ruiz, G. S. Santos, E. C. Ramírez, and E. M. Rojas, "Artificial intelligence and recommender systems in e-commerce. Trends and research agenda," *Intell. Syst. Appl.*, vol. 2024, no. Dec., pp. 1–12, 2024. doi: 10.1016/j.iswa.2024.200435.

[4] K. Xu, H. Zhou, H. Zheng, M. Zhu, and Q. Xin, "Intelligent Classification and Personalized Recommendation of E-commerce Products Based on Machine Learning," in *Proceedings of the 6th International Conference on Computing and Data Science*, 2024, pp. 143–149. doi: 10.54254/2755-2721/64/20241365.

[5] M. F. Aljunid, M. D.H., M. K. Hooshmand, W. A. Ali, A. M. Shetty, and S. Q. Alzoubah, "A collaborative filtering recommender systems: Survey," *Neurocomputing*, vol. 617, no. 1, pp. 1-18, Feb. 2025, doi: 10.1016/J.NEUCOM.2024.128718.

[6] T. M. A. U. Gunathilaka, P. D. Manage, J. Zhang, Y. Li, and W. Kelly, "Addressing sparse data challenges in recommendation systems: A systematic review of rating estimation using sparse rating data and profile enrichment techniques," *Intell. Syst. Appl.*, vol. 2025, no. Mar., pp. 1–15, 2025. doi: 10.1016/j.iswa.2024.200474.

[7] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: A survey," *Decis. Support Syst*., vol. 74, no.1, pp. 12–32, Jun. 2015, doi: 10.1016/j.dss.2015.03.008.

[8] L. Yao, Q. Z. Sheng, A. H. H. Ngu, J. Yu, and A. Segev, "Unified collaborative and content-based web service recommendation," *IEEE Trans. Serv. Comput.,* vol. 8, no. 3, pp. 453–466, May 2015, doi: 10.1109/TSC.2014.2355842.

[9] Y. Xu and J. Yin, "Collaborative recommendation with user generated content," *Eng. Appl. Artif. Intell*., vol. 45, no.1, pp. 281–294, Oct. 2015, doi: 10.1016/j.engappai.2015.07.012.

[10] B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades, "Facing the cold start problem in recommender systems," *Expert Syst. Appl.*, vol. 41, no. 4 PART 2, pp. 2065–2073, 2014, doi: 10.1016/j.eswa.2013.09.005.

[11] L. Yun, Y. Yang, J. Wang, and G. Zhu, "Improving Rating Estimation in Recommender Using Demographic Data and Expert Optimations," in *IEEE 2nd International Conference on Software Engineering and Service Science, IEEE, IEEE Access*, vol. 2011, no. Jul., pp. 120–123, 2011. doi: https://doi.org/10.1109/ICSESS.2011.5982269.

[12] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, "A collaborative filtering approach to mitigate the new user cold start problem," *Knowl. Based. Syst*., vol. 26, no.1, pp. 225–238, Feb. 2012, doi: 10.1016/j.knosys.2011.07.021.

[13] Z. Zhang, H. Lin, K. Liu, D. Wu, G. Zhang, and J. Lu, "A hybrid fuzzy-based personalized recommender system for telecom products/services," *Inf. Sci. (N Y)*., vol. 235, no.1, pp. 117–129, Jun. 2013, doi: 10.1016/j.ins.2013.01.025.

[14] L. H. Son, "HU-FCF++: A novel hybrid method for the new user cold-start problem in recommender systems," *Eng. Appl. Artif. Intell*., vol. 41, no.1, pp. 207–222, May 2015, doi: 10.1016/j.engappai.2015.02.003.

[15] L. H. Son, "HU-FCF: A hybrid user-based fuzzy collaborative filtering method in Recommender Systems," *Expert Syst. Appl.*, vol. 41, no. 15, pp. 6861–6870, Nov. 2014, doi: 10.1016/j.eswa.2014.05.001.

[16] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowl. Based. Syst.*, vol. 46, no.1, pp. 109–132, 2013, doi: 10.1016/j.knosys.2013.03.012.

[17] K. Shah, A. Salunke, S. Dongare, and K. Antala, "Recommender Systems: An overview of different approaches to recommendations," in *2017 International Conference on Innovations in information Embedded and Communication Systems (ICIIECS), IEEE Access*, vol. 2017, no. Feb., pp. 1–4, 2017. doi: https://doi.org/10.1109/ICIIECS.2017.8276172.

[18] F. Christyawan, A. N. Rohman, and A. D. Hartanto, "Application of Content-Based Filtering Method Using Cosine Similarity in Restaurant Selection Recommendation System," *Journal of Information Systems and Informatics,* vol. 6, no. 3, pp. 1559–1576, Sep. 2024, doi: 10.51519/journalisi.v6i3.806.

[19] B.-H. Huang and B.-R. Dai, "A Weighted Distance Similarity Model to Improve the Accuracy of Collaborative Recommender System," *IEEE Access*, vol. 2015, no. Jun., pp. 104–109, 2015. doi: 10.1109/MDM.2015.43.

[20] F. Ricci, L. Rokach, and B. Shapira, "Introduction to Recommender Systems Handbook," *Springer J. Recomm. Syst.*, vol. 2011, no. Jan., pp. 1–35, 2011. doi: 10.1007/978-0-387-85820-3_1.

[21] A. Yassine, L. Mohamed, and M. Al Achhab, "Intelligent recommender system based on unsupervised machine learning and demographic attributes," *Simul. Model. Pract. Theory*, vol. 107, no. 1, pp. 1–16, Feb. 2021, doi: 10.1016/j.simpat.2020.102198.

[22] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 261–273, Nov. 2015, doi: 10.1016/j.eij.2015.06.005.

[23] B. Patel, P. Desai, and U. Panchal, "Methods of recommender system: A review," in *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, vol. 2017, no. Mar., pp. 1–4, 2017. doi: 10.1109/ICIIECS.2017.8275856.

[24] G. Adomavicius and J. Zhang, "Improving stability of recommender systems: A meta-algorithmic approach," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 6, pp. 1573–1587, Jun. 2015, doi: 10.1109/TKDE.2014.2384502.

[25] M. A. Ghazanfar and A. Prugel-Bennett, "A scalable, accurate hybrid recommender system," in *3rd International Conference on Knowledge Discovery and Data Mining, WKDD* 2010, vol. 2010, no. Jan., pp. 94–98, 2010. doi: 10.1109/WKDD.2010.117.

[26] D. T. Lien and N. D. Phuong, "Collaborative filtering with a graph-based similarity measure," in *2014 International Conference on Computing, Management and Telecommunications (ComManTel)*, vol. 2014, no. Apr., pp. 251–256, 2014. doi: 10.1109/ComManTel.2014.6825613.

[27] J. P. Lucas, N. Luz, M. N. Moreno, R. Anacleto, A. Almeida Figueiredo, and C. Martins, "A hybrid recommendation approach for a tourism system," *Expert Syst. Appl.*, vol. 40, no. 9, pp. 3532–3550, Jul. 2013, doi: 10.1016/j.eswa.2012.12.061.

[28] S. Sharma and A. Mahajan, "Suggestive Approaches to Create a Recommender System for GitHub," *International Journal of Information Technology and Computer Science*, vol. 9, no. 8, pp. 48–55, Aug. 2017, doi: 10.5815/ijitcs.2017.08.06.

[29] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egypt. Inform. J.*, vol. 2015, no. Nov., pp. 1–12, 2015. doi: 10.1016/j.eij.2015.06.005.

[30] Y. Tang and Q. Tong, "BordaRank: A Ranking Aggregation Based Approach to Collaborative Filtering," in *2016 IEEE/ACIS 15th International Conference on Computer and Information Science, IEEE*, vol. 2016, no. Jun., pp. 1–6, 2016. doi: https://doi.org/10.1109/ICIS.2016.7550761.

[31] G. M. Dakhel and M. Mahdavi, "A new collaborative filtering algorithm using K-means clustering and neighbors' voting," in *2011 11th International Conference on Hybrid Intelligent Systems (HIS*), vol. 2011, no. Sep., pp. 179–184, 2011. doi: 10.1109/HIS.2011.6122101.

[32] L. Ziqi, N. I. Wancheng, H. Zhang, Z. Meijing, and Y. Yiping, "A K-medoids Algorithm Based Method to Alleviate the Data Sparsity in Collaborative Filtering," in *2015 34th Chinese Control Conference (CCC), vol. 2015, no. Jul., pp. 4974–4979, 2015*, doi: 10.1109/ChiCC.2015.7260413.

[33] L. Zahrotun and M. F. Akbar, "K-Means Centroid Optimization with Genetic Algorithm for Clustering Micro, Small, Medium Enterprises in Yogyakarta," *JUITA: Jurnal Informatika*, vol. 13, no. 2, pp. 87–97, 2025, doi: https://doi.org/10.30595/juita.v13i2.25480.

[34] L. Edi Nugroho, L. Lazuardi, and A. Satria Prabuwono, "Context-aware-based Location Recommendation for Elderly Care," *International Journal on Advanced Science Engineering Information Technology*, vol. 7, no. 5, pp. 1667–1677, 2017, doi: https://doi.org/10.18517/ijaseit.7.5.3382.

[35] Z. Yu, H. Xu, Z. Yang, and B. Guo, "Personalized Travel Package with Multi-Point-of-Interest Recommendation Based on Crowdsourced User Footprints," *IEEE Trans. Hum. Mach. Syst.*, vol. 46, no. 1, pp. 151–158, Feb. 2016, doi: 10.1109/THMS.2015.2446953.

[36] R. Pereira, H. Lopes, K. Breitman, V. Mundim, and W. Peixoto, "Cloud based real-time collaborative filtering for item-item recommendations," *Comput. Ind.*, vol. 65, no. 2, pp. 279–290, 2014, doi: 10.1016/j.compind.2013.11.005.

ISSN 2723-6471

[37] M. N. Noori, J. Ahamed, and M. Ahmed, “Matrix Factorization and Cosine Similarity Based Recommendation System For Cold Start Problem in E-Commerce Industries,” *International Journal of Computing and Digital Systems*, vol. 15, no. 1, pp. 775–787, 2024, doi: 10.12785/ijcds/150156.