ISSN 1330-3651(Print), ISSN 1848-6339 (Online) https://doi.org/10.17559/TV-20250428002631
Original scientific paper

Enhancing Recommendation Accuracy of Item-Based Collaborative Filtering under Data Sparsity

Tao ZHANG, Jiaming PI*, Kun ZHAO

Abstract: Data sparsity remains a major challenge for collaborative filtering (CF) systems, where enhancing recommendation accuracy with limited data is crucial. Traditional CF methods depend on common ratings for similarity computation but often ignore users' bounded rationality in neighbor selection and rating behavior. To overcome these limitations, we propose an Expected Utility-based Collaborative Filtering (EUCF) method with three key contributions: (1) a positioning function computing item expected ratings to capture global patterns, (2) a dynamic similarity criterion incorporating item rating frequency and user rating saturation for adaptive neighbor selection (threshold ε), and (3) an average deviation correction model addressing imperfect ratings. Experiments on MovieLens-1M and Netflix show EUCF outperforms KNN-IBCF, improving *MAE* by 3.56% and 4.17%, and *MSE* by 6.02% and 6.45%, respectively. Moreover, EUCF boosts recommendation coverage by 9.96% and 11%, mitigating data sparsity and long-tail challenges. With reduced computational complexity, EUCF ensures better scalability for real-world deployment. The results demonstrate its effectiveness in balancing accuracy, efficiency, and robustness under sparse data conditions.

Keywords: bounded rationality; collaborative filtering; data sparsity; expected utility; recommendation accuracy

1 INTRODUCTION

With the exponential growth of online data, users struggle to extract valuable information efficiently - a phenomenon known as information overload. Collaborative filtering (CF) recommendation systems address this issue by analyzing user preferences to actively recommend relevant resources, thereby meeting personalized needs in the Big Data era [1]. The concept of CF was first introduced by Goldberg et al. in 1992 with the Tapestry system [2], sparking rapid development and widespread adoption in academia and industry. Today, leading platforms like Amazon, Google, and Yahoo employ CF-based recommenders [3-5], cementing its status as a key research area.

The core challenges in CF systems lie in similarity computation and rating prediction, which directly impact recommendation accuracy. Practical applications face hurdles such as data sparsity, cold-start scenarios, long-tail recommendations, scalability, and unpredictable user behavior. Research indicates that cold-start and long-tail issues stem from rating matrix sparsity [6-8], which remains a primary barrier to high-quality recommendations. Current solutions include dimensionality reduction (e.g., SVD) [6-8], enhanced similarity metrics [9, 10], optimized neighbor selection [11], matrix completion [12-16], and user confidence modeling [17, 18]. While these methods improve prediction accuracy, they often incur high computational costs. As data volumes surge, balancing prediction precision with algorithmic efficiency has become a critical research focus in recommendation systems.

To address the limitations of traditional recommendation methods, this paper proposes an Expected Utility-based Collaborative Filtering (EUCF, a novel recommendation method that models users' preference utility under bounded rationality constraints) approach from a bounded rationality perspective [19, 20]. Unlike conventional K-nearest neighbor item-based collaborative filtering (KNN-IBCF) that relies on computationally intensive similarity calculations and neighborhood selection, EUCF introduces two key innovations: (1) a dynamic threshold ε derived from item-level rating expectations to efficiently measure item relationships, and (2) an average deviation-corrected rating prediction model that accounts for user irrationality and data sparsity. Experimental results on MovieLens-1M and Netflix datasets demonstrate that EUCF achieves comparable accuracy while being significantly more efficient than KNN-IBCF, as it eliminates the need for complex similarity computations and neighborhood searches. By better balancing prediction quality and computational performance, while effectively handling sparse rating data and bounded rational user behavior, EUCF provides a more practical and scalable solution for real-world recommendation scenarios, offering both theoretical advancements and practical implementation benefits.

2 RELATED RESEARCH

Recent advances in collaborative filtering have focused on improving similarity computation to address data sparsity and cold-start problems. The classic similarity calculation method is the adjusted cosine similarity, which uses the same equations as the Pearson correlation coefficient, and can be regarded as an extended type of cosine similarity [9, 10]. In addition, there are many more algorithms for improving similarity calculation. For example, the Matusita coefficient was introduced in [1], taking into account all the ratings given by a user to estimate the nearest neighbors to arrive at the similarity. In [21], a trapezoidal fuzzy rating model was proposed that calculates the similarity by blurring the sharp points into trapezoidal fuzzy numbers based on the rating statistics. In [22], an improved similarity measure based on the differences between users' common preferences, rating scales, and common item ratings was proposed. In [23], a collaborative filtering recommendation algorithm that integrates user similarity and rating attributes was designed. In [24], a trust-based model, combined with user similarity, was proposed, along with integration of user evaluation with attribute similarity to perform similarity calculation. The statistical meaning similarity measure (SIS) was introduced in [25] to calculate the similarity. In [26], the user rating and time interval for each attribute from a user's rating of an item was obtained, and then two methods were generated to calculate the similarity between users; subsequently, weighting parameters were introduced, and the fusion similarity between two users was obtained by controlling the weight between the two similarity methods. In [27], a novel approach that calculates the similarity between users not only based on the items but rather on the attributes of the items was proposed. In [28], a new similarity measure method that uses a singularity factor to adjust nonlinear equation and takes into account the user scoring habits was introduced.

To summarize, the above-mentioned algorithms improved user or item similarity calculation and performed a large number of operations and analysis on the individual characteristics of users or items, as well as those of the application scenarios. The improved algorithms were able to improve the accuracy of the recommendation algorithm to a certain extent, but they greatly reduced the efficiency of the recommendation algorithm. Based on these previous studies, this paper's aim is to better solve the issue of trade-off between algorithm accuracy and algorithm efficiency performance, as well as the issues of the sparsity of rating data and the bounded rationality of user rating behaviors. From the perspective of bounded rationality, the expected utility of items is analyzed, and based on users' rating expectation (mean) obtained for items, the relationship between items are measured by defining the dynamic threshold ε (positioning accuracy). The nearest-neighbor set is then calculated, and an average deviation correction method is used to calculate the predicted rating.

3 COLLABORATIVE FILTERING RECOMMENDATION METHOD BASED ON EXPECTED UTILITY

This paper proposes an Expected Utility-based Collaborative Filtering (EUCF) method that introduces a paradigm shift from traditional similarity-based approaches by employing market positioning as its core concept, which operates on three fundamental principles. First, the Market Positioning Principle evaluates items based on their global expected ratings, representing perceived market value through aggregate user ratings rather than pairwise comparisons. Second, the Rating Consistency Hypothesis posits that users demonstrate consistent rating behavior for items with similar market positions, regardless of specific attributes. Third, the Adaptive Selection Mechanism dynamically adjusts reference item selection using a positioning accuracy threshold (ε) that intelligently balances rating frequency and user rating saturation.

The EUCF framework implements these principles through three key components: (1) a Positioning Function that calculates each item's expected rating to establish market position; (2) a Selection Function that determines reference items via an adaptive ε threshold responsive to data sparsity and user behavior; and (3) a Fitting Rating Calculation that generates predictions through deviation-corrected averaging while accounting for market position differences. This integrated approach provides a more robust and efficient alternative to conventional recommendation methods by capturing collective user wisdom while adapting to individual rating patterns.

3.1 Positioning Function

According to the above-mentioned principle, EUCF determines the positioning with the average rating of all the ratings obtained for each item, called the expected rating. Letting E_i be the expected rating of item i, then the positioning function is mathematically formulated as Eq. (1):

$$E_i = \sum_{r_i \in R_i} r_i / card(R_i) \quad (1)$$

where R_i denotes the expected rating set of item I and $card$ (R_i) represents the number of rating scores in the rating set. This formulation captures the global preference for each item.

3.2 Selection Function

EUCF selects the reference set (nearest-neighbor item) by controlling the proximity of the market positions between items by a threshold ε. For prediction user u and prediction item i, the reference set is denoted $Su(i)$, which is determined by the selection function Eq. (2) defined herein,

$$Su(i) = \{j \,||\, Ei - Ej \,| \le \varepsilon\} \quad (2)$$

Item j ($j \neq i$) that satisfies Eq. (2) becomes the rating reference item of prediction item i. The threshold ε is called the positioning accuracy.

The traditional KNN-IBCF method usually controls the selection range of neighbor items by setting a fixed number of nearest-neighbor items K and uses a set of neighbor items as a reference set. In contrast, the positioning accuracy ε of EUCF is not a fixed parameter, but is dynamically determined by a function according to the saturation of the ratings obtained by the prediction item.

According to the principle of expected utility, the positioning accuracy is closely related to the number of actual ratings obtained by the prediction item and the rating saturation of all the items evaluated by prediction users. In general, if the number of actual ratings obtained by the prediction item is small, the value of the positioning accuracy should be large and otherwise small; if the rating saturation of all the items evaluated by prediction users is low, the value of the positioning accuracy should be large and otherwise small. Based on this, the accuracy function is given, as in Eq. (3):

$$\varepsilon = \frac{1 - sat(u)}{1 + \ln(1 + card(R_j))} \quad (3)$$

where the prediction user is u and the prediction item is j. I_u is the set of all items that have been evaluated by user u. R_j is the rating set of item j ($j \in I_u$), and $card(R_j)$ is the number of its corresponding rating scores. $sat(u)$ is the average rating saturation of all items rated by user u and is defined as the ratio of the actual number of rating scores of the items evaluated by the user to the expected number of rating scores; the equation is expressed as Eq. (4):

$$sat(u) = \frac{\sum_{j \in I_u} card(R_j)}{card(I_u) \times card(U)} \tag{4}$$

3.3 Fitting Rating Score

According to the basic principle of the EUCF method, in predicting a user's rating of an item, if another item with the same positioning can be found and the user has rated it, then the rating is the rating on the prediction item by the prediction user; if multiple of such reference items are found, then the prediction user should give them the same rating. However, considering the factors, such as incomplete rationality and even complete irrationality in the user's actual rating behavior, as well as the finiteness of the evaluation items in the recommender system and the sparsity of the rating data, the predicted rating $r^p_{u,I}$ uses Eq. (5), the average deviation correction method, for estimation. Here, the prediction user is u, the prediction item is i, $r_{u,j}$ is the rating by user u for reference item j, and the reference set is generated according to Eq. (2). The fitting rating score equation is Eq. (5):

$$r^p(u,i) = \bar{u} + b \tag{5}$$

where $\bar{u}$ is the arithmetic mean of the ratings given by user u to the reference item, and b the deviation correction value:

$$\bar{u} = \frac{\sum_{j \in S_u(i)} r_{u,j}}{card(S_u(i))} \tag{6}$$

The deviation correction value b is estimated by the average difference between the market expectation of prediction item i and the market expectation of reference item j [$j \in Su(i)$], and the calculation equation is Eq. (7):

$$b = E_i - \frac{\sum_{j \in S_u(i)} E_j}{card(S_u(i))} \tag{7}$$

3.4 Calculation Process for EUCF Recommendation Method

The basic process of the EUCF method is as follows.

Step 1. Using the positioning function equation to calculate the market expectation E_i (mean) of each item in the existing rating data and the number of its corresponding rating scores $card(R_i)$.

Step 2. When calculating the prediction user's rating on the prediction item, determine the item set I_u that has been rated by the prediction user, and then calculate the positioning accuracy ε of the prediction item according to the selection function; then, from the items already rated by the prediction user, determine the neighbor items such that the difference between the market expectation of the already rated items and that of the prediction items is less than ε, which leads to the reference set $Su(i)$.

Step 3. Calculate the arithmetic mean of the ratings by prediction user u for the neighbor items, and then calculate the average difference b (deviation correction value) between the market expectation of the prediction item I and that of the neighbor item j [$j \in Su(i)$].

Step 4. Calculate the predicted rating $r^p_{u,I}$ using the fitting predictive rating calculation equation based on the arithmetic mean value $\bar{u}$ and deviation correction value b.

Step 5. Select the N items with the highest predicted rating as the recommendation results.

4 EXPERIMENTAL DESIGN

Collaborative filtering can generate recommendations through both item and user-based approaches. The technique for generating recommendations by analyzing the similarity between items is called item-based collaborative filtering (IBCF), and the technique for generating recommendations by analyzing similarities between users is called user-based recommendation (referred to as user-based collaborative filtering (UBCF)). In practical applications, the IBCF method is usually adopted. In general, the recommendation accuracy of IBCF is higher than that of UBCF. Therefore, the experiments in this paper are based on the analysis of relationships between items.

Su et al. present three main categories of CF techniques and attempt to present a comprehensive survey for CF techniques from basic techniques to the state-of-the-art [29]. Laurent Candillier et al. review the main collaborative filtering methods proposed in the literature and compare them on the same widely used real dataset [30]. Combined with the above method, we choose KNN-IBCF as a baseline to compare.

4.1 Experimental Environment

The software and hardware environment built by the experimental platform in this paper includes the following components: host processor, 8th generation intelligent Intel(R) Core(TM) i7-8565U CPU@4.6 GHz; memory, DDR4, 16 GB; operating system, Windows7; application software MATLAB.

4.2 Experimental Datasets

Movielens and Netflix are the two most commonly used film-rating datasets in the research test on recommendation techniques [31]. The MovieLens dataset is derived from movielens.umn.edu (this website is a research-based recommendation system), and it is collected by the University of Minnesota's Group-Lens. The MovieLens dataset is commonly used for recommendation system research [32]. The Netflix dataset is a standard dataset which has been used for the Netflix Prize competition. The Netflix prize competition held by Netflix was designed to rank the recommendation algorithms. The Netflix dataset from December 31, 1999 to December 31 is well known by recommendation system researchers. It contains more than 480000 randomly selected anonymous users, with more than 100 million ratings over 17 thousands movies [33]. In this paper, we randomly selected 3184912 ratings of 10385 movies from

27053 users in Netflix dataset. Based on the above analysis, we choose the Movielens-1M and Netflix datasets as experimental datasets, as shown in Tab. 1.

Table 1 Description of experimental datasets

	Movielens-1M	Netflix
Number of users	6040	27053
Number of movies	3706	10385
Number of ratings	1000209	3184912
Rating scale	1-5	1-5
Sparsity / %	95.53	98.87

As can be seen from Tab. 1, the Movielens-1M dataset is relatively small and the Netflix data set is relatively large. For the experimental samples of the two datasets, 80% of each were randomly selected as the training set and 20% as the test set. The user set in the training set is U_T, the item set is I_T, the user set in the test set is U_P, and the item set is I_P. Regarding the KNN-IBCF method discussed above, the training set was used to generate the related data, such as similarity between items, number of rating users, and information entropy of the rating. Regarding the EUCF method, the training set was used to generate the related data, such as the market expectation, number of rating users, and information entropy of the rating. They were all processed before making the rating prediction.

4.3 Comparison of Experimental Methods

In this paper, the traditional KNN-IBCF method and the EUCF method proposed in this paper were used for comparative experiments. The reason the traditional KNN-IBCF method was used as the comparison method is that the KNN-IBCF method is a simple and efficient method in collaborative filtering technology. It is also the basis of many rating prediction methods, so it has significant reference value. The experimental scheme is briefly described as follows.

Experiment I: For the KNN-IBCF method, the similarity between the items was calculated using the adjusted cosine similarity equation, and the base number of the neighbor item count N was set to 25. Multiple experiments were performed on the dataset, and the number of nearest neighbors taken in each experiment was appropriately adjusted, and experiments performed by setting a fixed number of neighbors.

Experiment II: For the EUCF method, the experiment was performed by calculating the predicted rating using average deviation correction according to Eq. (5) to Eq. (7) in Section 3. Ten experiments were performed on the same dataset. Each experiment randomly selected 80% of each as training sets and 20% as test sets. The results of the 10 experiments were averaged to obtain the best results.

5 ANALYSIS AND DISCUSSION OF RESULTS

5.1 Algorithm Efficiency Comparison

The KNN-IBCF recommendation method usually calculates the similarities between any two items in the system according to the similarity calculation method established in advance and stores the similarities for subsequent rating prediction. If the system has m users, the time complexity of calculating the similarity between any two items in n items is $O(n^2 \times m)$. Regardless of the storage method used, the required storage space is also on the order of $O(n^2)$. In rating prediction, each time a rating prediction is performed, $n - 1$ comparisons are needed in n items owned by the system to select k- nearest neighbor items. Therefore, the time complexity of performing a rating prediction is $O(n)$.

The EUCF recommendation method only needs to consider an item's own rating in the calculation of the item's market expectation and its existing number of rating scores, without considering the relationship with other items. Therefore, the time complexity of the rating by m users for processing n items is $O(m \times n)$, and the storage overhead is on the order of $O(n)$, which can also be processed in an offline manner. In the rating prediction, the time complexity of calculating the expected proximity between items is calculated as $O(n)$. When calculating the predicted rating, although the number of items in the reference set obtained for each prediction item is different, the time complexity does not exceed $O(n)$. The time complexity and storage overhead of the above two recommendation methods are summarized in Tab. 2.

Table 2 Time complexity and storage overhead of EUCF

Method	Offline time complexity	Online time complexity	Storage overhead
KNN-IBCF	$O(n^2 \times m)$	$O(n)$	$O(n^2)$
EUCF	$O(m \times n)$	$O(n)$	$O(n)$

In summary, compared to KNN-IBCF, the time complexity of EUCF is greatly reduced. This performance feature is extremely important when the system has a large number of items and users because it can better adapt to the needs of incremental computing.

5.2 Experimental Results and Analysis

5.2.1 Recommended Results and Evaluation

Tab. 3 shows the best experimental results for *Top-N* recommendations using the traditional KNN-IBCF algorithm and the threshold ε recommendations using the EUCF algorithm. The experimental results show that, compared with the traditional KNN-IBCF method, the *MAE*, *MSE*, precision, F1 score, and coverage of the EUCF method proposed in this paper have all been somewhat improved whether the dataset is Movielens-1M or Netflix.

Table 3 Comparison of recommendation results

Data set	Method	*MAE*	*MSE*	Precision / %	Recall / %	F1 / %	Coverage / %
Movielens-1M	KNN-IBCF	0.786	0.929	59.67	39.72	44.36	73.16
	EUCF	0.758	0.873	62.63	38.81	47.16	80.45
Netflix	KNN-IBCF	0.743	0.868	62.53	37.83	49.32	77.35
	EUCF	0.712	0.814	65.37	36.92	53.94	85.92

The following discussion is based on the above experimental results.

First, from *MAE* and *MSE*, the EUCF algorithm further improved the overall error of the prediction. For the Movielens-1M dataset, compared with the traditional KNN-IBCF algorithm, *MAE* of the EUCF algorithm was improved by 3.56% and *MSE* by 6.02%. For the Netflix dataset, *MAE* of the EUCF algorithm was improved by 4.17% and *MSE* by 6.45%. This indicates that the EUCF algorithm proposed in this paper mined the user's overall preference information, actual ratings of the prediction items, and rating saturation of the items already rated by the prediction user. It can filter out neighbor items with higher similarity as a whole, thus effectively improving the overall error of the prediction.

Secondly, from the perspective of precision, recall, and F1 indicators, the EUCF algorithm improved the comprehensive performance of recommendation. Whether for the Movielens-1M or Netflix dataset, the EUCF algorithm improved the recommendation precision, even though the recall decreased. This is consistent with the previous research finding that recall generally decreases with increasing precision. The F1 value can integrate the evaluation indicators of both precision and recall. For the two datasets, the F1 of the EUCF algorithm increased by 6.31% and 9.36%, indicating that the EUCF algorithm further improved the comprehensive performance of recommendation.

Furthermore, the coverage of the EUCF algorithm on the Movielens-1M and Netflix datasets increased by 9.96% and 11%, respectively, both exceeding 80%. The coverage of the EUCF algorithm is therefore more reasonable than that of KNN-IBCF. Because coverage is an important indicator with which to measure the long-tail recommendation of the recommender system, it means that the EUCF algorithm can make better long-tail recommendations through the recommender system.

Finally, the sparsity of the Movielens-1M dataset was 95.53% and that of the Netflix dataset was 98.52%. The *MAE* of the KNN-IBCF algorithm was 0.786 and 0.743. The same method improved the *MAE* of the different datasets by 5.47%, while the EUCF algorithm decreased the *MAE* by 6.06%. The EUCF algorithm can therefore better solve the data sparsity problem of the recommender system than KNN-IBCF because EUCF takes into account the rating saturation of the items already rated by the prediction user.

5.2.2 Analysis of Influence of Parameters *Top-N* and Threshold *ε*

In the collaborative recommendation process, the number of neighbors *N* of the prediction items also has a significant impact on the quality of the recommendation results. In the general recommendation process, when the *Top-N* value is relatively small, the recommendation result will not be satisfactory because of the limited information available. When the *Top-N* value is too large, the recommendation error will also increase due to the low reference value of the introduced neighboring items. This problem was further verified in the experiment. Fig. 1 contains the *MAE* and *MSE* results returned by the KNN-IBCF algorithm when different *N* values were taken on the Movielens-1M dataset. It shows that when the *Top-N* value is taken from 5 to 30 gradually, *MAE* and *MSE* become smaller and smaller. When the *Top-N* value is taken from 30 to 60, *MAE* and *MSE* become increasingly larger. When *Top-N* = 30, the minimum *MAE* value is 0.786 and the minimum *MSE* value is 0.929. The number of *Top-N* neighbors recommended for the prediction item should therefore not be too small or too large. By the same token, as shown in Fig. 2, on the Netflix dataset when *Top-N* = 50, the minimum *MAE* value is 0.743 and the minimum *MSE* value is 0.868.

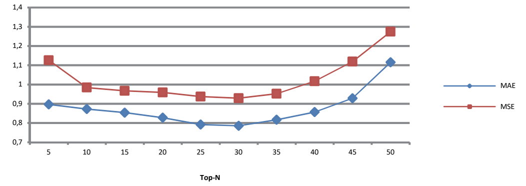

Figure 1 Effect of *Top-N* on recommendation results for dataset Movielens-1M

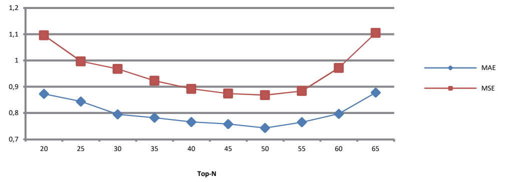

Figure 2 Effect of *Top-N* on recommendation results for dataset Netflix

For the threshold *ε* recommendation of the EUCF algorithm, in the actual calculation process the threshold *ε* (positioning accuracy) of the EUCF algorithm is not set to a fixed parameter, but is dynamically determined by a function according to the related information such as the number of actual ratings for the prediction item and the rating saturation. Considering this, 10 EUCF experiments were used on the Movielens-1M dataset to find the mean value of the threshold *ε* of each experiment, and the threshold *ε* mean retained three decimal places so as to study the effect of the threshold *ε* mean on the quality of the recommendation results. Fig. 3 contains the returned *MAE* and *MSE* results when the 10 different values of the threshold *ε* mean were taken using the EUCF algorithm on the Movielens-1M dataset. The threshold *ε* mean falls in [0.215, 0.235]. When the threshold *ε* mean is taken from 0.215 to 0.225, *MAE* and *MSE* become increasingly smaller, and when the threshold *ε* is taken from 0.225 to 0.235, *MAE* and *MSE* become increasingly larger. This shows that when the threshold *ε* mean is 0.225 the minimum *MAE* value is 0.769 and the minimum *MSE* value is 0.884, close to the *MAE* and *MSE* of the best recommendation result under the dynamic threshold *ε*. Similarly, as shown in Fig. 4, when the threshold value *ε* is 0.235, the minimum *MAE* value is 0.723 and the minimum *MSE* value is 0.821.

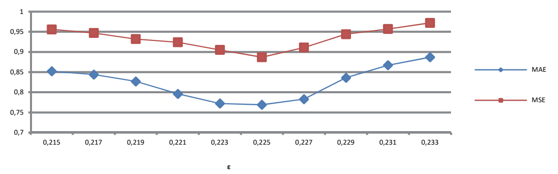

Figure 3 Effect of threshold *ε* on recommendation results for dataset Movielens-1M

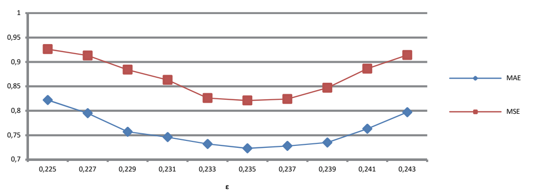

Figure 4 Effect of threshold ε on recommendation results for dataset Netflix

To summarize, the *Top-N* value of the traditional KNN-IBCF algorithm and the threshold ε of the EUCF algorithm have a direct impact on the quality of collaborative filtering recommendation results. The experimental results show that, overall, the EUCF algorithm proposed in this paper is better than the traditional KNN-IBCF algorithm. In addition, the EUCF algorithm is more effective in optimizing the larger of two datasets tested, i.e., Netflix.

6 CONCLUSIONS

This paper presents the Expected Utility-based Collaborative Filtering method, which significantly improves recommendation accuracy under data sparsity by incorporating global item expectations, dynamic thresholding (ε), and deviation correction. Experimental results on MovieLens-1M and Netflix datasets show EUCF reduces *MAE* by 3.56-4.17% and *MSE* by 6.02-6.45% compared to KNN-IBCF, while increasing coverage by 9.96-11%, demonstrating superior handling of long-tail items. With $O(m \times n)$ offline and $O(n)$ online complexity, EUCF offers a scalable solution for real-world systems where efficiency and sparse data are critical. These advancements provide practical guidance for developing high-performance recommender systems, emphasizing the balance between accuracy and computational feasibility.

While EUCF demonstrates superior performance in handling data sparsity, several promising directions warrant further investigation. First, integrating deep learning techniques could enhance the modeling of complex user-item interactions beyond bounded rationality. Second, exploring temporal dynamics in user preferences may improve long-term recommendation stability. Finally, extending EUCF to cross-domain recommendations could address cold-start problems more effectively. These extensions could further advance the practical utility of expectation-based collaborative filtering approaches.

Acknowledgements

This work was supported by the National Natural Science Foundation Program of China (Grant No. 71462036), the Scientific Research Foundation of Yunnan Education Department (Grant No. 2023J0653), the School-level Project of Yunnan University of Finance and Economics (Grant No. 2021B02, No. 2024H72, No. 2023YUFEYC079), and the Open Foundation of Engineering Research Center of Cyberspace (Grant No. KJAQ202112004), and the Yunnan Fundamental Research Projects (Grant No. 202501AT070461).

7 REFERENCES

[1] Guan, J., Chen, B., & Yu, S. (2024). A hybrid similarity model for mitigating the cold-start problem of collaborative filtering in sparse data. *Expert Systems with Applications*, *249*, 123700. https://doi.org/10.1016/j.eswa.2024.123700

[2] Goldberg, D., Nichols, D., Oki, B., & Terry, D. (1992).Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, *35*(12), 61-70. https://doi.org/10.1145/138859.138867

[3] Mishra, K. N., Mishra, A., Barwal, P. N., Lal, R. K., & Barwal, M. (2024). Natural Language Processing and Machine Learning-Based Solution of Cold Start Problem Using Collaborative Filtering Approach. *Electronics*, *13*(11), 1024-1045. https://doi.org/10.3390/electronics13112104

[4] Al-Hasan, T. M., Sayed, A. N., Bensaali, F., Himeur, Y., Varlamis, I., & Dimitrakopoulos, G. (2024). From Traditional Recommender Systems to GPT-Based Chatbots: A Survey of Recent Developments and Future Directions. *Big Data and Cognitive Computing*, *8*(4), 36. https://doi.org/10.3390/bdcc8040036

[5] Park, S. & Pennock, D. (2007). Applying collaborative filtering techniques to movie search for better ranking and browsing. *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 550-559. https://doi.org/10.1145/1281192.1281252

[6] Koohi, H., Kobti, Z., Farzi, T., & Mahmodi, E. (2024). UDIS: Enhancing Collaborative Filtering with Fusion of Dimensionality Reduction and Semantic Similarity. *Electronics*, *13*(20), 4073. https://doi.org/10.3390/electronics13204073

[7] Mehta, R., Mishra, S., & Saha, S. (2025). Onto Proximality in Non Negative Matrix Factorization for Recommender Systems. *IEEE Access*, *13*, 67476-67487. https://doi.org/10.1109/access.2025.3557945

[8] Movafegh, Z. & Rezapour, A. (2023). Improving collaborative recommender system using hybrid clustering and optimized singular value decomposition. *Engineering Applications of Artificial Intelligence*, *126*, 107109. https://doi.org/10.1016/j.engappai.2023.107109

[9] Al-Hassan, M., Abu-Salih, B., Alshdaifat, E., Aloqaily, A., & Rodan, A. (2024). An Improved Fusion-Based Semantic Similarity Measure for Effective Collaborative Filtering Recommendations. *International Journal of Computational Intelligence Systems*, *17*(1), 45. https://doi.org/10.1007/s44196-024-00429-4

[10] Abdi, M., Okeyo, G., & Mwangi, R. (2025). Improved Collaborative Filtering Recommender System Based on Hybrid Similarity Measures. *The International Arab Journal of Information Technology*, *22*(1). https://doi.org/10.34028/iajit/22/1/8

[11] Kulvinder Singh, Dhawan, S., & Bali, N. (2024). An Ensemble Learning Hybrid Recommendation System Using Content-Based, Collaborative Filtering, Supervised Learning and Boosting Algorithms. *Automatic Control and Computer Sciences*, *58*(5), 491-505. https://doi.org/10.3103/s0146411624700615

[12] Behera, G., Nain, N., & Soni, R. K. (2024). Integrating user-side information into matrix factorization to address data sparsity of collaborative filtering. *Multimedia Systems*, *30*(2), 64. https://doi.org/10.1007/s00530-024-01261-8

[13] Abubakar, A. M. & Almu, A. (2024). Enhancing User-Item Matrix Using Principal Component Analysis and User Profiling Techniques for User Based Collaborative Filtering Recommender System. *International Journal of Science for Global Sustainability*, *10*(3), 140-152. https://doi.org/10.57233/ijsgs.v10i3.726

[14] Mishra, S., Singh, T., Kumar, M., & Satakshi. (2024). Lazy learning and sparsity handling in recommendation systems. *Knowledge and Information Systems*, *66*(12), 7775-7797. https://doi.org/10.1007/s10115-024-02218-z

[15] Fan, R., Wang, Z., Guo, Y., Xu, Y., Wang, Z., & Li, W. (2024). Robust enhanced collaborative filtering without explicit noise filtering. *The Journal of Supercomputing*, *80*(11), 15763-15782. https://doi.org/10.1007/s11227-024-06086-w

[16] Hamidi, H. & Moradi, R. (2024). Design of a dynamic and robust recommender system based on item context, trust, rating matrix and rating time using social networks analysis. *Journal of King Saud University - Computer and Information Sciences*, *36*(2), 101964. https://doi.org/10.1016/j.jksuci.2024.101964

[17] Li, D. & Esquivel, J. A. (2025). Trust-Aware Hybrid Collaborative Recommendation with Locality-Sensitive Hashing. *Tsinghua Science and Technology*, *30*(4), 1421-1434. https://doi.org/10.26599/tst.2023.9010096

[18] Ngwawe, E., Abade, E., & Mburu, S. (2023). Trust Enhanced Collaborative Filtering Recommendation Algorithm. *International Research Journal of Computer Science*, *10*(04), 88-96. https://doi.org/10.26562/irjcs.2023.v1004.10

[19] Bsoul, Q., Zawaideh, F., Alqadi, B. S., Almusfar, L. A., Khalaf, O. I., Alattas, A. S., Alali, M., & AbdElminaam, D. S. (2025). From User Preferences to Accurate Predictions: Enhancing Movie Recommendation Systems with Neural Collaborative Filtering and Sentiment Analysis. *SN Computer Science*, *6*(3). https://doi.org/10.1007/s42979-025-03742-7

[20] Gao, R. (2024). Exploring the landscape of recommendation systems: A qualitative analysis of types, challenges, and potential solutions. *Applied and Computational Engineering*, *64*(1), 217-222. https://doi.org/10.54254/2755-2721/64/20241444

[21] Deng, J., Chen, J., Wang, S., Ye, J., & Wang, Y. (2024). A novel fuzzy neural collaborative filtering for recommender systems. *Expert Systems with Applications*, *258*, 125153. https://doi.org/10.1016/j.eswa.2024.125153

[22] Abdi, M., Okeyo, G., & Mwangi, R. (2025). Improved Collaborative Filtering Recommender System Based on Hybrid Similarity Measures. *The International Arab Journal of Information Technology*, *22*(1). https://doi.org/10.34028/iajit/22/1/8

[23] Rismala, R., Maulidevi, N. U., & Surendro, K. (2024). Personalized neural network-based aggregation function in multi-criteria collaborative filtering. *Journal of King Saud University - Computer and Information Sciences*, *36*(1), 101922. https://doi.org/10.1016/j.jksuci.2024.101922

[24] Tang, C., Zhao, S., Chen, B., Lu, X., & Zhang, Q. (2024). A two-dimensional time-aware cloud service recommendation approach with enhanced similarity and trust. *Journal of Parallel and Distributed Computing*, *190*, 104889. https://doi.org/10.1016/j.jpdc.2024.104889

[25] Al-Hassan, M., Abu-Salih, B., Alshdaifat, E., Aloqaily, A., & Rodan, A. (2024). An Improved Fusion-Based Semantic Similarity Measure for Effective Collaborative Filtering Recommendations. *International Journal of Computational Intelligence Systems*, *17*(1). https://doi.org/10.1007/s44196-024-00429-4

[26] Mahesh, T. R., Vinoth Kumar, V., & Lim, S.-J. (2023). UsCoTc: Improved Collaborative Filtering (CFL) recommendation methodology using user confidence, time context with impact factors for performance enhancement. *PLOS ONE*, *18*(3), e0282904. https://doi.org/10.1371/journal.pone.0282904

[27] Li, A., Liu, X., & Yang, B. (2024). Item Attribute-aware Graph Collaborative Filtering. *Expert Systems with Applications*, *238*, 122242. https://doi.org/10.1016/j.eswa.2023.122242

[28] Abdalla, H. I., Amer, A. A., Amer, Y. A., Nguyen, L., & Al-Maqaleh, B. (2023). Boosting the Item-Based Collaborative Filtering Model with Novel Similarity Measures. *International Journal of Computational Intelligence Systems*, *16*(1). https://doi.org/10.1007/s44196-023-00299-2

[29] Singh, R., Dwivedi, P., & Kant, V. (2024). Comparative analysis of collaborative filtering techniques for the multi-criteria recommender systems. *Multimedia Tools and Applications*, *83*(24), 64551-64571. https://doi.org/10.1007/s11042-024-18164-55

[30] Mateos, P. & Bellogín, A. (2024). A systematic literature review of recent advances on context-aware recommender systems. *Artificial Intelligence Review*, *58*(1). https://doi.org/10.1007/s10462-024-10939-4

[31] Li, L., Ai, J., Su, Z., & Yanyan, L. (2018). Collaborative filtering recommendation algorithm combining user behavior and item tags. *Computer Applications and Software*, *35*(6), 248-253.

[32] Wang, Y., Wang, X., & Tang, W. (2018). Collaborative filtering recommendation algorithm fusing users' natural nearest neighbors. *Computer Engineering and Applications*, *54*(7), 77-83.

[33] Su, Q., Zhang, J., Lin, Z., Xiaomei, L., Zhaoquan, C., & Yong'an, Z. (2019).Collaborative filtering recommendation algorithm based on improved fuzzy partition clustering. *Computer Engineering and Applications*, *55*(5), 118-123.

Contact information:

Tao ZHANG
1) Information School,
Yunnan University of Finance and Economics, Kunming, China
2) Engineering Research Center of Cyberspace, Kunming, China

Jiaming PI
(Corresponding author)
Graduate school,
Yunnan University of Finance and Economics, Kunming, China
E-mail: pijiaming_ynufe@163.com

Kun ZHAO
Information School,
Yunnan University of Finance and Economics, Kunming, China