ISSN 2616-5775 Vol. 7, Issue 5: 199-206, DOI: 10.25236/AJCIS.2024.070526

Reproduction Study of Item-CF Algorithm Based on Penalty Factor Correction

Chen Shuangyu, Zhang Bin, Chen Junhao, Zhou Zihan, Shen Yibo, Xiang Weitao*

Zhejiang Yuexiu University, Shaoxing City, Zhejiang Province, 312000, China
**Corresponding author*

Abstract: *With the continuous growth of China's e-commerce industry, the user base for online shopping is expanding, and e-commerce platforms are encountering bottlenecks in their development. The issue of information overload has made it increasingly challenging for consumers to locate the specific products they intend to purchase among the vast array of available options. Given the extensive range of products on e-commerce platforms, traditional collaborative filtering methods struggle with sparse user-item matrices, and less popular items face challenges in receiving recommendations. This paper introduces several algorithms, including collaborative filtering and cosine similarity-based algorithms.To enhance the quality of the recommendation system, this paper incorporates users' feedback coefficients on products into the penalty factor. Additionally, by integrating user portraits obtained through cluster analysis, a collaborative filtering algorithm based on penalty factor correction is proposed.*

Keywords: *Item-CF algorithm; Recommendation system; Disciplinary value; Matrix decomposition; Collaborative filtering algorithm*

1. Introduction

As early as the last century, the internet rapidly developed globally. The widespread use of personal computers and communication devices such as mobile phones has completely transformed the ways people handle and collect vast amounts of information. Regardless of location, as long as there is internet access and possession of these internet terminal devices, one can find a plethora of valuable resources shared by people worldwide.

The latest report on China's internet development status, released by the China Internet Network Information Center (CNNIC) in December 2022, indicates that the number of internet users in China reached 1.067 billion, showing a year-on-year increase of 3.4%.The internet penetration rate has 75.6%.The aforementioned statistics highlight the increasing amount of time people spend online and growing number of activities taking place on the internet.

According to the International Data Corporation (IDC), the global data volume is expected to reach 120ZB by 2025, with China's data volume increasing to 15ZB[1].

In the context of such a massive scale of internet users, data has consistently shown an exponential increase.Meanwhile, as China stands as the world's largest online retail market, the continuous development of e-commerce has made it increasingly challenging for consumers to discover products that meet their needs on online shopping platforms.The importance of recommendation systems ine-commerce platforms has been further emphasized. In the short term, these systems can save users time and effort in searching for products, while in the long term, they contribute to enhancing user satisfaction and loyalty to the website.

This has prompted major e-commerce platforms, including JD.com, to actively engage in research recommendation systems. These platforms have developed recommendation systems based on users' historical behavioral records, such as searches, favorites, additions to the cart, browsing product details, and sharing with friends. These systems recommend to users products similar to those previously purchased or products with potential interest, thereby meeting user needs and increasing the desire to make purchases. Assisting consumers in finding products that meet their needs from a vast amount of information and helping sellers increase the visibility of their products have become crucial challenges addressed by recommendation system algorithms in the application of e-commerce platforms. The

ISSN 2616-5775 Vol. 7, Issue 5: 199-206, DOI: 10.25236/AJCIS.2024.070526

diagram below illustrates the basic concept of recommendation systems in Figure 1.

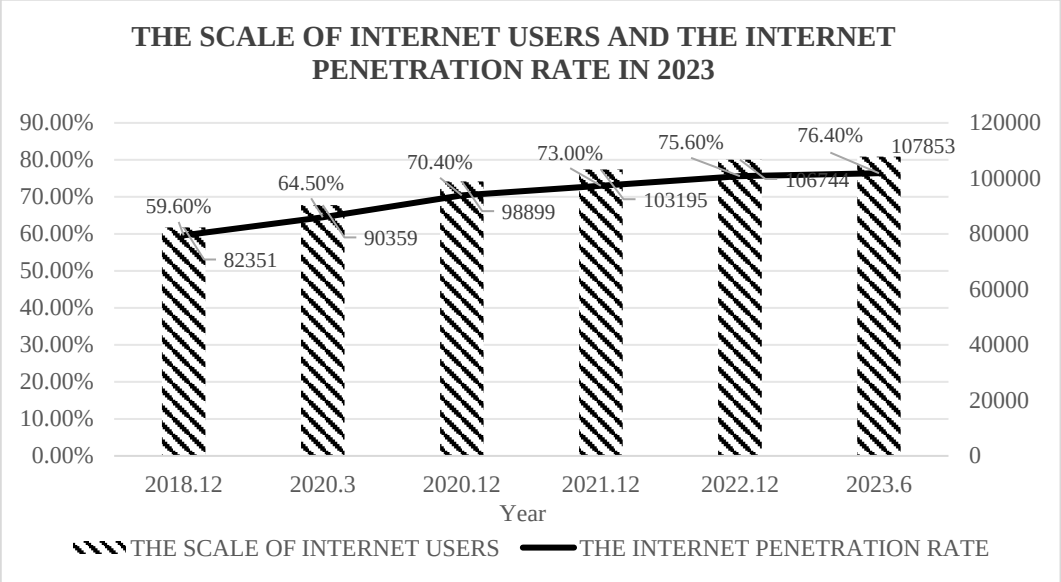

Figure 1. The scale of internet users and the internet penetration rate in 2023

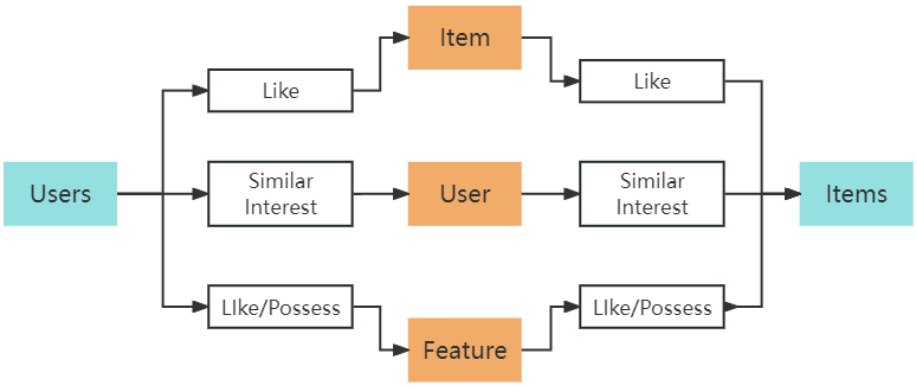

Figure 2. The basic conceptual diagram of a recommendation system

From Figure2,e-commerce platforms have witnessed a surge in sales after adopting personalized recommendation systems. They leverage personalized recommendation systems as a distinctive feature and advantage of the platform. Leading e-commerce platforms like Taobao and JD.com, based on various user behavioral information such as browsing, favoriting, adding to the shopping cart, and purchasing, conduct data analysis. While the field has made progress, challenges like data sparsity, real-time and scalability issues in recommendation algorithms, and the cold start problem remain unresolved.

In improving traditional recommendation algorithms and enhancing recommendation effectiveness, numerous scholars have conducted extensive research. In the context of domestic research, Wang Yonggui and Liu Kaiqi (2020) proposed an optimized clustering-based collaborative filtering recommendation algorithm.They preprocessed the original rating matrix based on user rating differences, thus constructing a user category preference matrix. This approach better reflects user interest preferences and alleviates data sparsity issues[2]. Feng Xiang (2020) tackled the challenge from the perspective of domain-based collaborative filtering, introducing a method that incorporates user penalty factors and item penalty factors.This method reduces the similarity between active users and popular items, consequently improving the quality of recommendations.[3]. Bao Kaili et al. (2019) presented a recommendation parallel algorithm that combines naive Bayes and collaborative filtering.They utilized a parallel naive Bayes classification algorithm to build a text sentiment classifier and integrated it with rating values to construct a comprehensive rating model. By optimizing the parallel ALS algorithm, they achieved improvements in recommendation accuracy and system stability [4]. Gu Mingxing et al. (2020) proposed an algorithm that combines K-means++ clustering with

ISSN 2616-5775 Vol. 7, Issue 5: 199-206, DOI: 10.25236/AJCIS.2024.070526

collaborative filtering. This algorithm introduces a time factor on top of traditional rating similarity, and experiments on the MovieLens dataset demonstrated its effectiveness in improving prediction accuracy[5].All of these improved recommendation algorithms can actively address the cold start problem and to some extent alleviate the issue of low precision in collaborative filtering recommendations.

This article tackles current issues and potential optimizations in e-commerce user recommendation methods. Existing algorithms overlook the impact of popular items on user similarity calculations, leading to a bias towards these items in recommendations. To address this, the article suggests introducing penalty factors to adjust similarity weights, enhancing the traditional cosine similarity method. The article validates this approach using MovieLens datasets, demonstrating its feasibility[6].

2. Based on Penalty Factor Modified Item-CF Algorithm

2.1 Algorithm Basic Framework

Collaborative filtering algorithms calculate similarity based on differences in user evaluations of products, ignoring the impact of a product's popularity or necessity on similarity.Some popular and widely favored items may not adequately reflect users' "individuality" and latent preferences, nor indicate strong similarity between different users. In such cases, their influence on user similarity measurement should be reduced.Existing algorithms, when calculating user similarity, do not consider the impact of popular items on similarity[7]. That is, since most users provide feedback on popular items, the algorithm's recommendation results tend to increasingly favor popular products. On the other hand, when calculating item similarity, the algorithm's results are more susceptible to the influence of active users, as active users provide feedback on multiple items.To address these issues, this paper introduces a penalty factor to reduce the similarity weight of popular items and active users. Additionally, it improves the similarity calculation method based on traditional cosine similarity. This modification aims to correct the traditional Item-CF algorithm (Item-CF-Correct) and achieve an enhancement in recommendation coverage[8].

From Figure3,the basic framework of the Item-CF-Correct recommendation algorithm is illustrated in the diagram below.

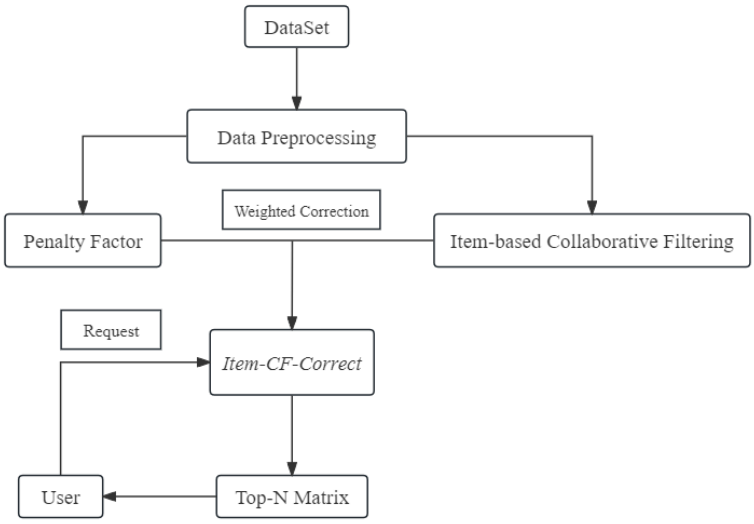

Figure 3. The basic framework of the Item-CF-Correct recommendation algorithm

2.2 Collaborative Filtering Algorithm Processing

Currently, the most widely used collaborative filtering recommendation algorithms mainly include user-based collaborative filtering algorithms and item-based collaborative filtering algorithms. Collaborative filtering recommendation algorithms analyze data to generate the nearest neighbor set for the current user[9]. The top N items most interesting to the user in this set are recommended to the current user, known as Top N recommendations. The collaborative filtering recommendation algorithm

ISSN 2616-5775 Vol. 7, Issue 5: 199-206, DOI: 10.25236/AJCIS.2024.070526

process can be divided into three main steps:

(1) Representing data as a matrix: Establish an m*n user-item rating matrix based on user evaluations of items. Here, m represents the total number of users, n represents the total number of items, and the entry (i, j) in the matrix represents the rating given by the i-th user to the j-th item.

(2) Discovering nearest neighbors: Discovering nearest neighbors involves calculating user or item similarity to establish the nearest neighbor set for the current user. Based on this set, items are sorted by interest to implement recommendations for the current user. This process can be described as follows: for the target user u, establish a neighbor set in order of similarity by computing the similarity between the target user and other users. Let the similarity between the target user and user v be denoted as s(u, v), where similarity decreases in the order of s(u, v1) > s(u, v2) > s(u, v3), and so on. The accuracy of establishing the nearest neighbor set is crucial for the success of collaborative filtering algorithms.

User-based collaborative filtering algorithms calculate user similarity to establish the nearest neighbor set. In this paper, cosine similarity is employed to calculate the similarity between the target user (denoted as u) and the neighbor user (denoted as v). The calculation formula is as follows:

$$w_{uv} = \frac{|N_{(u)} \cap N_{(v)}|}{\sqrt{|N_{(u)}||N_{(v)}|}} \tag{1}$$

In this context, N(u) represents the set of items that the target user u has provided feedback on, and N(v) represents the set of items that the neighbor user v has provided feedback on. The item-based collaborative filtering algorithm establishes the nearest neighbor set by calculating item similarity. In this paper, cosine similarity is used to calculate the similarity between item i and item j, with the following formula:

$$w_{ij} = \frac{|N_{(i)} \cap N_{(j)}|}{\sqrt{|N_{(i)}||N_{(j)}|}} \tag{2}$$

In this context, N(i) represents the set of users providing feedback on item i, and N(j) represents the set of users providing feedback on item j.

(3) Generate recommendation set: After obtaining the nearest neighbor set for the target user, calculate the predicted rating for the unrated item i based on this neighbor set. Select the top N items with the highest predicted ratings as the Top N recommendation set. The calculation formula is as follows:

$$p_{Ti} = \overline{r_{u_T}} + \frac{\sum_{u \in N_{u_T}} sim(u_T,u) \times (r_{u,i} - \overline{r_u})}{\sum_{u \in N_{u_T}} sim(u_T,u)} \tag{3}$$

2.3 Improvement of Item Popularity Penalty Factor Similarity Correction

Collaborative filtering algorithms calculate similarity based on differences in user evaluations of items, overlooking the influence of an item's popularity or necessity on similarity.Some widely popular items may not adequately reflect users' "individuality" and latent preferences, nor indicate strong similarity between different users.In such cases, their impact and contribution to user similarity measurement should be reduced.For instance, two platform users who highly rate the book "Python" may find it challenging to determine that they share the same interests based on this popular item.Conversely, if both users have viewed niche professional books, it can better reflect similar interests and preferences[10].Therefore, less popular items can better reflect similarity between users, and the higher the item's obscurity, the stronger the similarity.

Considering that user similarity is influenced by popular items, a popularity penalty factor is introduced as a weighting coefficient to suppress the impact of popular items.The introduction of a popularity penalty factor as a weighting coefficient improves the cosine similarity calculation method.The more frequently an item appears, the more mainstream it is, and the less it contributes to the similarity of user interests.The cosine similarity calculation method, modified with the introduction of the penalty factor, is expressed in the following formula:

$$W_{uv} = \frac{\sum_{i \in N_{(u)} \cap N_{(v)}} \log \frac{n}{n_i}}{\sqrt{|N_{(u)}||N_{(v)}|}} \tag{4}$$

In the context of equation (4), the penalty factor represents the total number of items in the dataset,

ISSN 2616-5775 Vol. 7, Issue 5: 199-206, DOI: 10.25236/AJCIS.2024.070526

while represents the number of items with positive feedback from users.Given a fixed dataset where the total number of items remains constant, a smaller value indicates that the user has purchased fewer items.This implies that the user is less active (with fewer positive feedback instances).Consequently, their influence on the similarity between items is greater.Conversely, a larger value suggests a more active user (with more positive feedback instances), resulting in a smaller impact on the similarity between items. The modified algorithm proposed in this paper is referred to as Item-CF-Correct[11].

3. Experimental Results and Analysis

3.1 Experimental Dataset

To validate the effectiveness of the proposed improvement methods in this paper, the algorithms presented in this chapter are tested on the following datasets. The datasets used in the experiments are introduced below[12].

The datasets selected for this paper are the open-source MovieLens 100k and MovieLens 1m datasets (abbreviated as ml 100k and ml 1m datasets, respectively). The ml 100k and ml 1m datasets in this dataset are widely used in academic research. The ml 100k dataset contains nearly 100,000 reviews from 943 users for 1682 movies. The ml 1m dataset includes rating information from 6040 users for 3952 movies. Prior to the experiments, the dataset is preprocessed, and it is divided into training and testing sets, with a ratio of 4:1 between the training set and the testing set.

The information statistics for the two datasets used in this experiment are shown in Table 1.

Table 1. Dataset Statistics Results

Dataset Name	Number of Items	Number of Ratings
ml 100k	1682	10000
ml 1m	3952	1000209

3.2 Evaluation Metrics

In the design of this paper, Top-N recommendations are employed. Therefore, precision is chosen as the metric to analyze accuracy. The precision of the recommendation results is defined by Formula (5), where $R(u)$ represents the user's actual movie-watching set, and $T(u)$ is the recommended list output by the system.

$$Precision = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|} \tag{5}$$

To enhance the system's ability to explore users' latent interests and ensure that the recommendation results can focus on more niche items, some scholars have defined coverage to describe the long-tail item mining capability of recommendation systems. Coverage is defined as the proportion of recommended items to the total item set. The simplest formula for calculating coverage is as follows:

$$Coverage = \frac{\left|\cup_{u \in U} R(u)\right|}{|I|} \tag{6}$$

In Formula (6), U represents the set of users,I represents the number of items in the system, and $R(u)$ represents the recommendation set for each user.

3.3 Performance Verification of Item-CF-Correct Algorithm

This experiment is based on the Windows operating system and utilizes Python 3.7 for algorithm reproduction.

Before the experiment, to assess the impact of the parameter K (neighborhood size) on the performance of the Item-CF algorithm, experiments were conducted with different values of K on datasets ml 100k and ml 1m. The experimental results are shown in Figure 7 below:

ISSN 2616-5775 Vol. 7, Issue 5: 199-206, DOI: 10.25236/AJCIS.2024.070526

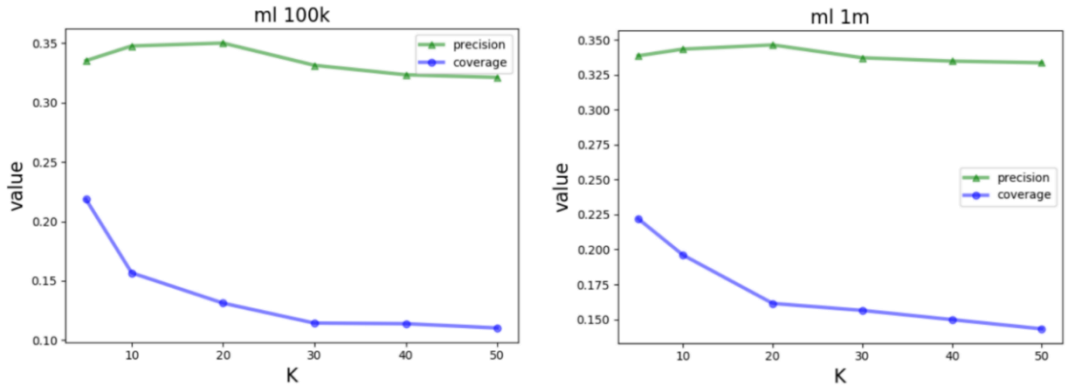

Figure 7. The Impact of Different K Values on the Performance of the Item-CF Algorithm: (A) Results for ml 100k, (B) Results for ml 1m.

From Figure 7, it can be observed that when the value of K is 20, the algorithm achieves the highest precision. When K is too large or too small, it affects the accuracy of the Item-CF algorithm. Similarly, we can conclude that there is a negative correlation between the value of K and the coverage of the Item-CF algorithm. For the subsequent experiments, we choose K=20 based on the above observations.

The test results of the Item-CF-Correct algorithm after penalty factor correction on ml 100k and ml 1m are shown in Figure 8. The vertical axis represents the precision and coverage of the recommendation system, while the horizontal axis represents the penalty factor. The selected experimental results in the figure illustrate the most significant trends in data variation.

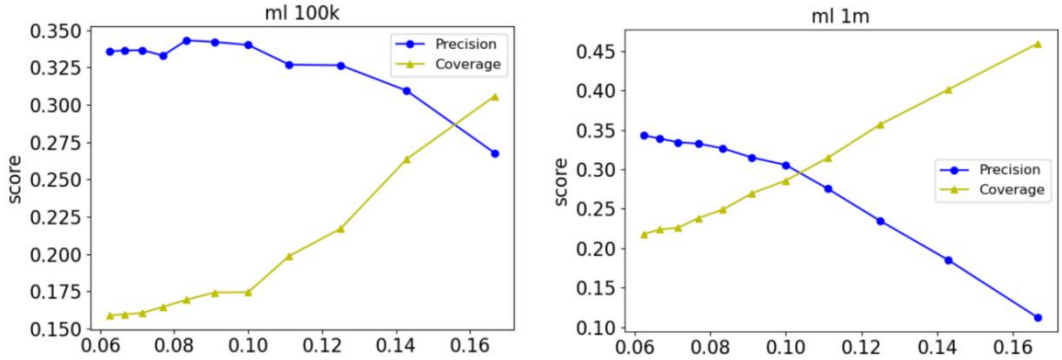

Figure 8. Performance Variation Chart of the Item-CF-Correct Algorithm: (A) Results for ml 100k, (B) Results for ml 1m.

From Figure 8, in plot (A), it is evident that the accuracy of the algorithm shows a noticeable trend of initially rising and then declining. This suggests that penalizing popular items to some extent has a positive impact on the algorithm's accuracy, which is consistent with the characteristics observed in the previous test results on the ml 100k dataset. When the penalty factor is greater than 0.17, the algorithm's accuracy reaches its peak before decreasing. It is also noticeable that the rate of decline in accuracy is quite rapid, indicating that increasing the penalty factor has a significant impact on the algorithm. This observation aligns with the performance on the first two datasets. The coverage rate of the algorithm consistently increases, with a change rate that starts slow and then accelerates, indicating that a larger penalty factor has a more substantial impact on the algorithm.

In plot (B) of Figure 8, it can be observed that the performance of the algorithm is not significantly influenced by the penalty factor for coverage rates until the penalty factor is less than 0.15. In terms of accuracy, a slight upward trend can be observed, although the change is not very pronounced. Nevertheless, an improvement in accuracy is evident, consistent with the earlier experimental results. Regarding accuracy, a noticeable decline occurs when the penalty factor exceeds 0.2. As for coverage rate, a significant increase is observed when the penalty factor exceeds 0.13.

From Figure 9, we can observe that the accuracy of the Item-CF-Correct algorithm is slightly lower than that of the Item-CF algorithm on both datasets. However, its coverage rate is significantly better than the algorithm before correction. The improvement in algorithm coverage better addresses the long-tail issue of items in the recommendation system.

ISSN 2616-5775 Vol. 7, Issue 5: 199-206, DOI: 10.25236/AJCIS.2024.070526

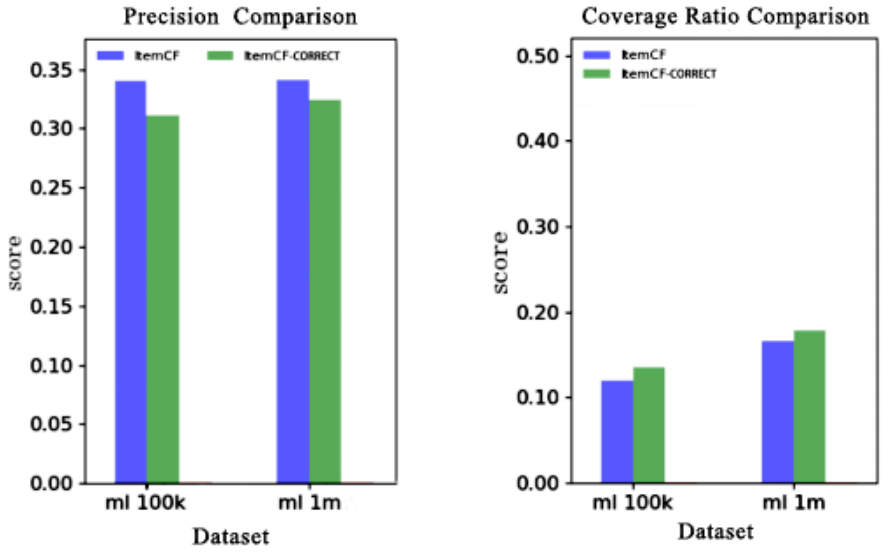

Figure 9. Comparison Chart of Algorithm Performance Before and After Correction in Different Datasets

Table 2. Algorithm Test Results

Dataset Name	Algorithm	Precision	Coverage
ml 100k	Item-CF	0.3379	0.1272
ml 1m	Item-CF	0.3417	0.1721
ml 100k	Item-CF-Correct	0.3162	0.1684
ml 1m	Item-CF-Correct	0.3247	0.1871

From Table 2,simultaneously comparing the results across different datasets reveals that, with an increase in the number of data samples, the coverage rate of the Item-CF algorithm significantly improves. This indicates that the Item-CF algorithm is notably influenced by the quantity of data samples, and algorithm performance tends to improve with a larger dataset. It can also be inferred that insufficient data samples negatively impact algorithm performance.Thus, the proposed Item-CF algorithm based on penalty factor correction, as presented in this paper, can enhance the coverage rate of the algorithm while maintaining accuracy. This optimization contributes to improving the recommendation effectiveness of the algorithm.

4. Conclusion

The accuracy of collaborative filtering recommendation algorithms is a crucial metric for evaluating the quality of recommendation systems. However, mining long-tail information in the data, recommending more niche items to users, and flattening the long-tail curve are essential for both user choice and item benefits[13]. In this paper, we introduced the weights of popular items and a penalty factor into cosine similarity calculation based on the traditional Item-CF algorithm. The feasibility of the algorithm was validated on two MovieLens datasets. The experimental results demonstrated that the proposed corrected algorithm not only ensures algorithm accuracy but also enhances algorithm coverage. It effectively mines long-tail information in the data, increases the weight of niche items, and improves both the recall and coverage of the recommended product list provided to users. This better reflects the diversity of personalized recommendation algorithms.In the future, we will address data sparsity and cold-start problems by introducing more parameters influencing similarity. This will further mitigate the impact of the penalty factor on algorithm accuracy[14].

Acknowledgement

Research on social recommendation system based on ternary closure -- taking Drone shopping mall as an example, supported by the Zhejiang Yuexiu University Undergraduate Innovation and Entrepreneurship Training Program (Project Number: 202312792016).

ISSN 2616-5775 Vol. 7, Issue 5: 199-206, DOI: 10.25236/AJCIS.2024.070526

References

[1] Bian, L.(2020). Research on User Recommendation Algorithms Based on E-commerce Platforms. Nanjing University .
[2] Shi, X.(2020). Research on Personalized Recommendation Algorithms in E-commerce Considering Price Factors. Tianjin University.
[3] Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. Communications of the ACM, 35(12), 61-70.
[4] Wu, Y., Shen, J., Gu, T. Z., Chen, X. H., Li, H., & Zhang, S. (2007). Algorithm for sparse problem in collaborative filtering. Jisuanji Yingyong Yanjiu/ Application Research of Computers, 24(6), 94-97.
[5] Kuo, R. J., Liao, J. L., & Tu, C. (2005). Integration of ART2 neural network and genetic K-means algorithm for analyzing Web browsing paths in electronic commerce. Decision Support Systems, 40(2), 355-374.
[6] Chen, P., Chen C., Hong Y.(2016).A Collaborative Filtering Recommendation Algorithm Combining Association Rules. Small Microcomput Syst 37(2) : 287-292.
[7] Deng, A., Zhu, Y., Shi, B.(2023).Collaborative Filtering Recommendation Algorithm Based on Predicting Item Ratings.Journal of Software, 14(9): 1621-1628.
[8] Zhang, Y., Dai, J., Xiong, Z., et al.(2013)Collaborative Filtering Algorithm with Stepwise Padding to Alleviate Data Sparsity.Computer Application Research, 30(9): 2602-2605.
[9] Deng, A., Zuo, Z., Zhu,Y.(2004). Collaborative Filtering Recommendation Algorithm Based on Item Clustering.Small Microcomput Syst, 25(9): 1665-1670.
[10] Liu, X., Ge, J., Chen,D.(2010). A Combined Recommendation Algorithm Based on Clustering and Collaborative Filtering.Computer Engineering and Science, 32(12): 125-127.
[11] Wang, Y., Liu K.(2020).An Optimization Clustering Collaborative Filtering Recommendation Algorithm.Computer Engineering and Applications, 56(15): 66-73.
[12] Xu, F.(2020). Implementation of a Collaborative Filtering Algorithm with Improved Similarity. Electronic Technology, 33(2): 54-59.
[13] Bao, K., Liu, Q.(2019).Research on Parallel Algorithm for Combining Naive Bayes and Collaborative Filtering in Takeout Recommendation.Computer Application and Software, 36(11): 250-255, 285.
[14] Gu, M., Huang, W., Huang, Y., et al.(2020)Combining User Clustering and Improved User Similarity in Collaborative Filtering Recommendation.Computer Engineering and Applications, 56(2): 185-190.