

Fusion of Internal Similarity to Improve the Accuracy of Recommendation Algorithm

Dr.S.Dhanabal*, Dr.K.Baskar*, R.Premkumar**

*Associate Professor, Department of Computer Science and Engineering, Kongunadu College of Engineering and Technology, Tamil Nadu.

** Assistant Professor, Department of Computer Science and Engineering, Kongunadu College of Engineering and Technology, Tamil Nadu.

Abstract: Collaborative filtering algorithms (CF) and mass diffusion (MD) algorithms have been successfully applied to recommender systems for years and can solve the problem of information overload. However, both algorithms suffer from data sparsity, and both tend to recommend popular products, which have poor diversity and are not suitable for real life. In this paper, we propose a user internal similarity-based recommendation algorithm (UISRC). UISRC first calculates the item-item similarity matrix and calculates the average similarity between items purchased by each user as the user's internal similarity. The internal similarity of users is combined to modify the recommendation score to make score predictions and suggestions. Simulation experiments on RYM and Last.FM datasets, the results show that UISRC can obtain better recommendation accuracy and a variety of recommendations than traditional CF and MD algorithms.

Keywords: Collaborative filtering; mass diffusion; recommendation accuracy; recommendation system; user internal similarity

1 Introduction

This is an era of information overload. Increasing information makes it very difficult to collect useful information [1]. Therefore, recommendation systems were created. Unlike search engines that use keywords to search for necessary information, recommendation systems find a user's potential behavior preferences based on the user's personal historical behavior footprint and provide each user with a personalized recommendation list [2]. Faced with different recommendation scenarios, researchers have developed a variety of recommendation algorithms, such as collaborative filtering recommendation (CF) [3–7] and content-based recommendation [8]. Their main idea is that similar users may purchase the same product or products purchased by the same user may be similar. The CF algorithm gives recommendations by calculating the similarity between user preferences. This recommendation method can be very effective because individuals who purchase many common items are more likely to share the same preferences in the future. These methods have been proven to provide accurate recommendation results. Some algorithms based on physical principles have also been used to design recommendation algorithms, such as mass diffusion (MD) [9–17] and heat conduction (HC) [18–20]. By combining MD and HC, a new fusion algorithm can be obtained. This fusion algorithm is superior to the original algorithm in both recommendation precision and recommendation diversity.

In general, both the CF algorithm and the MD algorithm tend to recommend popular products because these products have more edges, and it is easier to obtain a higher recommendation score. These algorithms have poor novelty and diversity [10,12,14,21]. Researchers in the field of recommendation systems are now mainly concerned with the precision of the recommendation, but in a real environment, the precision of the recommendation alone cannot explain the performance of the recommendation system. However, it is simple to simply improve the diversity of recommendations, but it often sacrifices the precision of recommendations to a large extent. This article introduces a novel method that can improve the novelty and diversity of the original recommendation algorithm and recommendation precision.

2 Related Work

Collaborative filtering (CF) is one of the most widely used and successful technologies in recommender systems [7–8]. Neighborhood-based collaborative filtering algorithms are widely used in

recommendation scenarios based on user-item interactions. Neighborhood-based methods use user-item interaction to calculate the similarity between users or items and then use the neighborhood of active users (in user-based recommendations) or item neighborhood (in item-based recommendations) to generate recommendations, deal with new user problems by including content-based recommendations or combine different methods to reduce the negative impact of data sparsity.

As an application of physics theory to the recommendation system, the mass diffusion algorithm is essentially a resource redistribution process between objects by adjacent users, which achieves high precision but low diversity [4,9,17]. Similar to the MD algorithm, there is also the HC algorithm [18–20]. The HC algorithm is similar to the heat conduction process in the user-item bipartite graph, and the HC algorithm has higher diversity but lower precision.

It has been found that precision and diversity seem to be two opposing aspects: when one side rises, the other side falls. A good recommendation algorithm should have both high precision and high diversity, so researchers are studying how to improve the diversity of recommended products without losing the recommendation precision. Why do we need to study the diversity of recommendations, rather than pure precision? People recognize that a good recommendation system should provide diversified items instead of items that are too similar. When items are too similar, users may not repeat purchases of similar products, while increasing diversity will allow users to choose from a wider range. Currently, researchers are attempting to solve the conflict between recommendation precision and recommendation diversity and find the ideal balance between the two indicators. Smyth and McClave proposed a bounded greedy selection algorithm and proved that it can improve the diversity of recommendations without reducing the precision [22]. Hurley et al. regarded the trade-off between similarity and diversity as a binary optimization problem and defined a controller to explicitly adjust the two metrics to obtain the optimal trade-off [23].

In recent years, recommendation algorithms based on complex networks [24], especially bipartite graphs, have attracted increasing attention from scholars. This type of algorithm draws on the ideas of mass diffusion and heat conduction to abstract the input data of the recommendation system into complex network models. The complexity is lower than that of the classic collaborative filtering algorithm, and it has better scalability. The recommendation algorithm based on the bipartite graph network structure was first proposed by Zhou et al. based on the principle of mass diffusion [21,25,26]. It is assumed that each user initially has certain recommended resources, considers the related items through a resource allocation mechanism, and finally reallocates the resources. According to the reallocated item resource score, the topN items with the highest resource scores that users have not selected are recommended to the user. The essence of the MD algorithm is to use a new user similarity calculation method to achieve the recommended effect. This kind of algorithm is based on the similarity between users, similar to collaborative filtering. Their disadvantage is that they tend to recommend highly popular items to users, and the diversity of algorithms is very low. Starting from the influence of the initial resource allocation algorithm, Liu found that when items with different degrees are given different initial resources, not only can the diversity of recommendations be improved, but recommendation precision can also be retained to a certain extent [27].

Adomavicius et al. established a recommendation system model of the maximum bipartite graph matching problem, which improved the coverage of recommendation results but did not effectively solve the recommendation of long-tail items [28–29]. Li et al. proposed a bipartite graph random walk recommendation algorithm, also known as the bipartite graph projection and ranking (BGPR) algorithm, which improves on the bipartite graph-based recommendation algorithm. Although it increases the diversity to a certain extent, it still tends to recommend popular items [30]. The PersonalRank algorithm proposed by Liu et al. is the most famous algorithm in graph-based random walks [31]. This algorithm alleviates the problems of cold start and sparse scoring when less information is available, and the main consideration is to improve the precision of recommendations. To improve the precision and diversity of recommendation algorithms and solve the problem of missing information, this paper proposes an improved method based on the internal similarity of users. This method balances and repairs the original recommendation score to generate a new recommendation list. Using internal similarity to reduce the

impact of popular items and increase the impact of cold items on recommendation results improves the long-tail mining ability of the recommendation system to a large extent.

3 Proposed Method

The suggested length of a manuscript is 10 pages. Each page in excess of 15 will be charged an extra fee. This section introduces the motivation for our proposed method, then conducts a detailed analysis of our proposed method and finally introduces the evaluation indicators we use.

3.1 Resource Model

We notice that the traditional CF method relies heavily on items purchased by two users. The more of the same items purchased, the more similar the two users are. The similarity of users is also related to their personal preferences. If the average similarity of items purchased by a user is high, it indicates that the user's purchase preferences are relatively concentrated. The more it shows that the user has a specific preference habit, the more the user's preferences can represent a hidden category of a certain type of item. If an item the user bought is purchased by other similar users similar, then the other items the user purchased are more likely to be purchased by others, so we believe that the more concentrated a user's purchases, the greater their contribution to other users' recommendations. The calculation method of user internal similarity (UIS) is shown in Eq. (1).

$$\text{UIS}(u) = \frac{2 * \sum_{i, j \in N(u), i \neq j} \text{sim}(i, j)}{|N(u)| * |N(u) - 1|} \quad (1)$$

In this paper, we propose a recommendation method to balance user similarity and use this method in user-based CF and mass diffusion. First, we convert the user-item bipartite graph into the user-item adjacency matrix form. Then, we calculate the similarity matrix of item-item and further calculate the internal similarity of each user. When the recommendation algorithm recommends for each target user, other recommended users contribute a score to their unpurchased items and multiply this recommendation score by the internal similarity of the recommended user to obtain the final recommendation list for the target user. Finally, the topN item with the highest score is selected according to the recommendation list to the target user, and the process is demonstrated in Fig. 1 below.

3.2 User Internal Similarity Collaborative Filtering

In the collaborative filtering algorithm, similarity calculation is a key step. There are many different methods for calculating the similarity. Cosine similarity is a similarity index that is used more frequently and has a better effect. Current researchers often use cosine similarity in CF. The similarity index used in this article is cosine similarity. The calculation of cosine similarity is shown in Eq. (2).

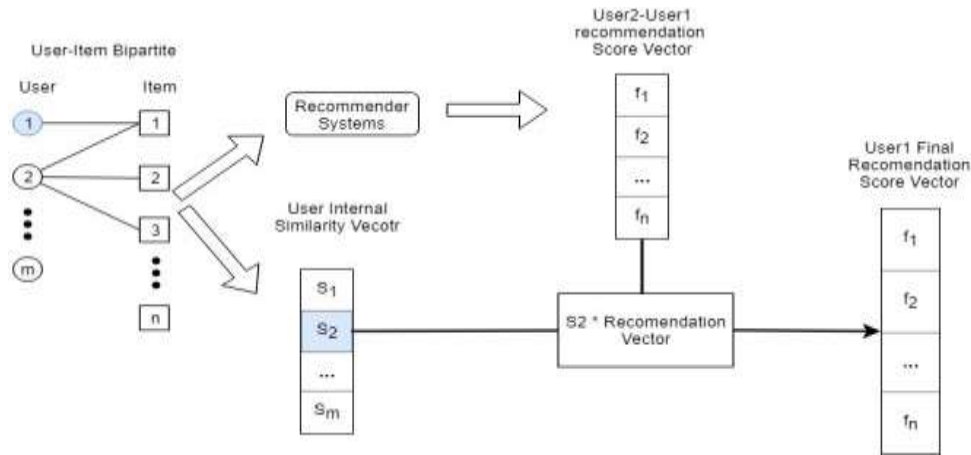


Figure 1: Flow chart of the user internal similarity recommendation

$$\text{Sim}(u) = \frac{|N(u) \cap N(v)|}{\sqrt{|N(u)| |N(v)|}} \quad (2)$$

$N(u)$ and $N(v)$ represent the collection of items that users u and v have interacted with.

In this paper, user-based CF is used to generate recommendation scores for items that the target user i has not interacted with. First, we determine all users j who have a common purchase behavior with the target user i and calculate the similarity between these users and user i . Finally, for user i , we can obtain the score of all the items he has not scored, as shown in Eq. (3).

$$f_i^a = \sum_{u=1}^N S_{iu} \alpha_{va} \quad (3)$$

α_{va} indicates the rating of item a by user v because it is implicit feedback. If v has bought item a , $\alpha_{va} = 1$; otherwise, $\alpha_{va} = 0$.

3.3 User Internal Similarity Mass Diffusion

The task of the recommendation system is to generate an ordered list of items that have not been purchased for the target user. Many recommended algorithms are inspired by the diffusion-like process. Mass diffusion initializes the resource f_a for the object a , where all the resources on the object constitute the resource vector f . Then, the resources between objects are reallocated according to formula $f' = \omega f$, where ω is called the resource transfer matrix.

The original recommendation algorithm that simulates the quality diffusion process is called the MD algorithm, also known as network-based reasoning (NBI) and ProbS. The recommendation process of the MD algorithm is as follows: for the target user u_i , the initial resource vector of the object is defined as $f_\alpha = a_{i\alpha}$, where if user u_i collects object α , $a_{i\alpha} = 1$; otherwise, $a_{i\alpha} = 0$. The transfer matrix ω is shown in Eq. (4).

$$\omega_{\alpha\beta} = \frac{1}{k_{o_\beta}} \sum_{l=1}^m \frac{a_{l\alpha} a_{l\beta}}{k_{u_l}} \quad (4)$$

where k_{o_β} is the degree of item o_β and k_{u_l} is the degree of user u_l .

For the MD algorithm, inspired by K-nearest neighbor CF, we proposed improvements to the MD algorithm. The original MD algorithm is divided into three diffusion processes: the first is the initial resource allocation, and the second is that the item side transfers resources to the corresponding user and then performs the third mass diffusion. Before the third mass diffusion, we first select K users with the highest scores at the user side and only use the resources of the K users with the highest scores for the third mass diffusion. Then, after the third mass diffusion, the final item quality is obtained. Experiments show that our improved MD algorithm has better performance.

Then, we further combine the internal similarity of users to improve the MD algorithm. In the specific improvement method, for each user, the mass diffusion process is performed separately, and we obtain the final item resource score f_l' of user u_l , multiply it by the internal similarity of user u_l to obtain a revised recommendation score f_l'' , and then present topN recommendations to the user. For user u_l , the resource score balanced by the user's internal similarity is shown in Eq. (5).

$$f_l'' = f_l' * UIS(l) \quad (5)$$

2 Evaluation Index

The performance of the recommendation system can be determined according to the evaluation index, and the specific evaluation index used should be selected according to the type of recommendation algorithm. In topN recommendation, precision and recall are the most commonly used evaluation indicators by researchers. Therefore, to evaluate the proposed improved model, these two indicators are used as evaluation indicators to measure the prediction precision. Although the precision of the recommendation is very important, the novelty, diversity and long-tail mining ability of the recommendation are also very important. This paper uses five indicators: precision, recall, coverage, novelty and diversity.

Precision is the proportion of recommended items that users actually similar to in the test set occupy the recommended list. The higher the index, the better the recommended performance. The calculation of precision is shown in Eq. (6).

$$Precision = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|} \quad (6)$$

Recall refers to the proportion of items that users actually like in the test set that occupies the recommendation list. The higher the index, the better the recommended performance. The calculation of recall is shown in Eq. (7).

$$Recall = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|} \quad (7)$$

In addition to the two accuracy indicators precision and recall, the nonaccuracy indicators coverage, novelty and diversity are also very commonly used in the recommendation system.

Coverage is the most commonly used nonaccuracy indicator. Coverage describes the ability of a recommendation system to mine the long tail of items. There are many definitions of coverage, and the

simplest and most commonly used is the ratio of items recommended by the recommendation system to all items. The calculation of coverage is shown in Eq. (8).

$$Coverage = \frac{|\bigcup_{u \in U} R(u)|}{|I|} \quad (8)$$

The novelty index refers to the average popularity in the recommendation list given by the recommendation system. If the user has not heard of most of the items in the recommendation list, the novelty is lower. The calculation of novelty is shown in Eq. (9).

$$Novelty = \frac{1}{N} \sum_{a \in O_R^K} \frac{d_a}{K} \quad (9)$$

where O_R^K represents the set of the first K recommended items of user u , and d_a represents the degree of item a . The larger the index is, the more likely the recommendation algorithm is to recommend the more popular products. In contrast, the more likely it is to recommend the less popular products, the more likely it is to reflect the user's interest.

Diversity describes the dissimilarity between two items in the recommendation list. The calculation of diversity is shown in Eq. (10).

$$Diversity = 1 - \frac{\sum_{i, j \in R(u), i \neq j} sim(i, j)}{\frac{1}{2} |R(u)| |R(u) - 1|} \quad (10)$$

4 Experiments

4.1 Dataset

In this paper, we use two datasets, Last.FM and RYM (rate your music). In our experiment, we ignore the score, and the interaction between each user and the movie is 1 and that without interaction is 0. The Last.FM data collection collects the social network, tagging and music art listening information of 2000 users. FM website. Rate your music (RYM) dataset is also a very common dataset for researchers, which comes from the music scoring website <http://rateyourmusic.com>. The distribution of these datasets is shown in the following table. To verify the performance of the improved algorithm, each dataset is randomly divided into two parts: 90% of the edges are training sets, and 10% of the edges are test sets. The training set is regarded as known information for recommendation, and the test set is used to evaluate algorithm performance.

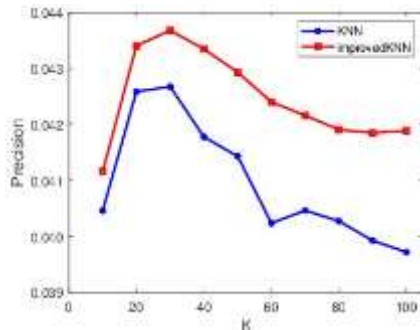
This article uses two datasets: Last.FM and rate your music (RYM). In the experiment, we ignore the user's rating of the items. The user's interaction with the item is 1, and the noninteraction is 0. The Last.FM dataset collects information on social networks, tagging and music artist listening of 2,000 users on the Last.FM website. The rate your music (RYM) dataset is also a very common dataset for researchers, from the music rating website <http://rateyourmusic.com>. The distribution of these datasets is shown in Tab. 1. To verify the performance of the improved algorithm, each dataset is randomly divided into two parts: 90% of the connected edges are the training set, and 10% of the connected edges are the test set. The training set is regarded as known information for recommendation, and the test set is used to evaluate the performance of the algorithm.

Table 1: Dataset description

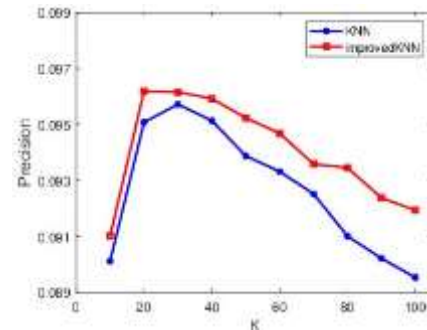
Dataset	Users	Items	Links	Sparsity
Last.FM	1892	17632	92834	2.78×10^{-3}
RYM	3378	4489	66408	4.38×10^{-3}

4.2 Result

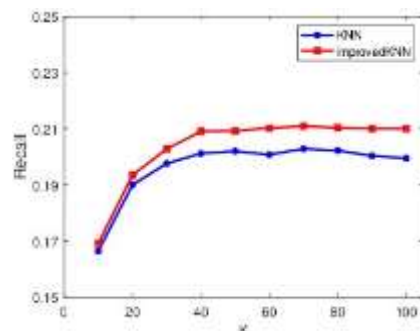
To test the effect of our improved algorithm on the CF algorithm and the MD algorithm, we performed experiments on the original CF algorithm and original MD algorithm and compared our improved algorithm. Both the CF algorithm and our improved MD algorithm are based on KNN, where K is the core parameter in the KNN algorithm. In the experiment, we set different values of K and observed the performance comparison between our method and the original algorithm under the conditions of different K values.



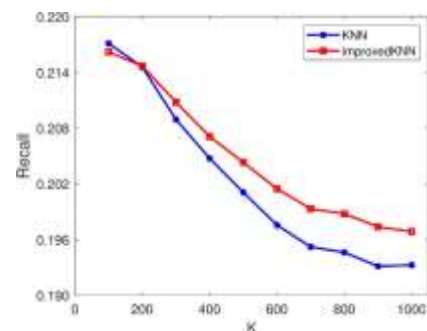
(a) The precision of the CF algorithm on the RYM dataset



(b) The precision of the CF algorithm on the Last.FM dataset



(c) The precision of the CF algorithm on the RYM dataset



(d) The precision of the CF algorithm on the Last.FM dataset

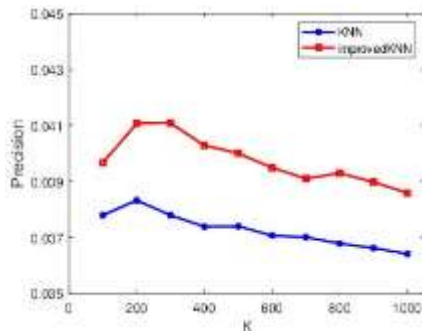
Figure 2: Comparison between the original CF algorithm and the improved CF algorithm in the precision and recall evaluation indexes on RYM and Last.FM datasets

K represents the number of nearest neighbors selected, and different K values lead to different recommendation accuracies. The CF algorithm is sensitive to the value of K , and we chose K in [10–100]. For the MD algorithm, the value of K was not very sensitive, and we chose K in [100–1000]. For the original CF algorithm and the improved algorithm, the performance of different datasets on the evaluation index of precision and recall is shown in Fig. 2.

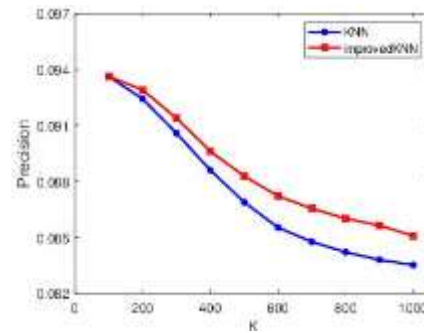
We can see in Fig. 2 that regardless of the value of K in the RYM dataset, our improved algorithm is superior to the original CF algorithm. With the increase in K , the precision of the algorithm first increases and then decreases and reaches the maximum value when $K = 30$. With the increase in the K value for Last.FM, our algorithm became better than the original algorithm.

We also implemented the improved method on the MD algorithm. Because the MD algorithm is very similar to the CF algorithm, as shown in Fig. 3, the MD algorithm obtained precision and recall similar to the CF algorithm. Our algorithm had an absolute advantage on the RYM dataset. Consistent with the CF algorithm, our algorithm had no obvious advantage when the K value was small in the Last.FM dataset. From the description of the algorithm, the smaller the value of K we used, the lower the number of nearest

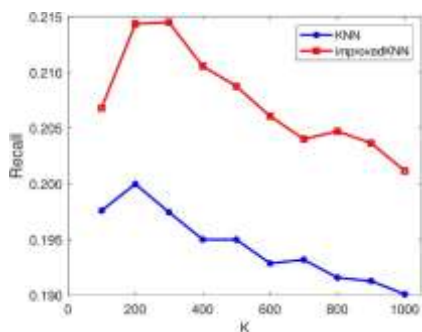
neighbor users selected, which led to fewer opportunities to use our improved method. From this point, it was proven that our algorithm can effectively improve the accuracy of the recommendation algorithm.



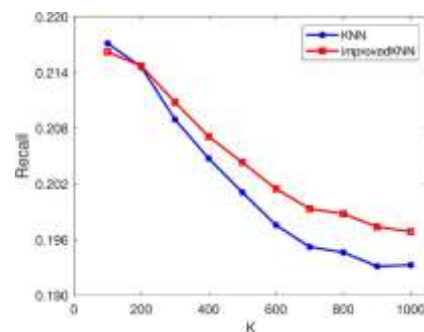
(a) The precision of the MD algorithm on the RYM dataset



(b) The precision of the MD algorithm on the Last.FM dataset



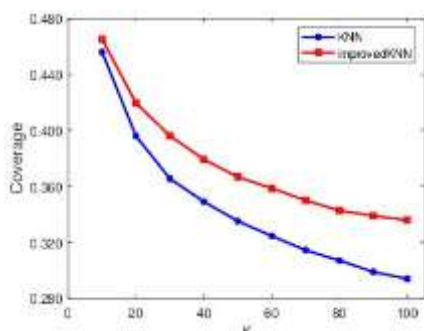
(c) The precision of the MD algorithm on the RYM dataset



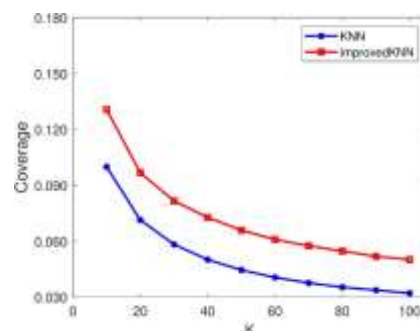
(d) The precision of the MD algorithm on the Last.FM dataset

Figure 3: Comparison between the original MD algorithm and the improved MD algorithm in the precision and recall evaluation indexes on RYM and Last.FM datasets

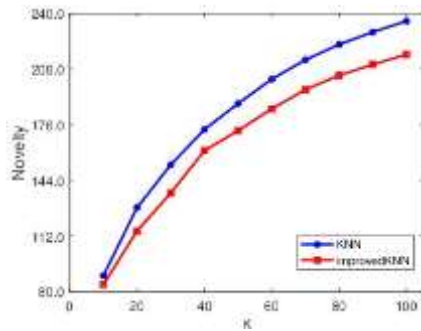
Next, we further studied the performance of our algorithm on coverage, novelty and diversity. We conducted experiments on the two algorithms separately. Fig. 4 shows the performance of the improved CF compared with the original CF algorithm on coverage, novelty and diversity. It can be seen that in addition to the novelty indicator, our other improved algorithms are superior to the original CF algorithm.



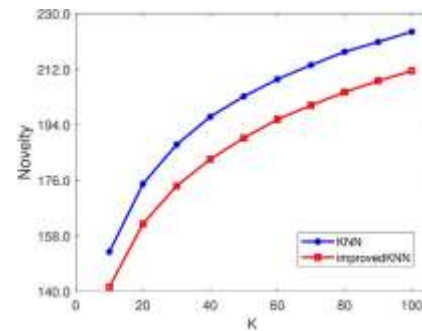
(a) The coverage of the CF algorithm on the RYM dataset



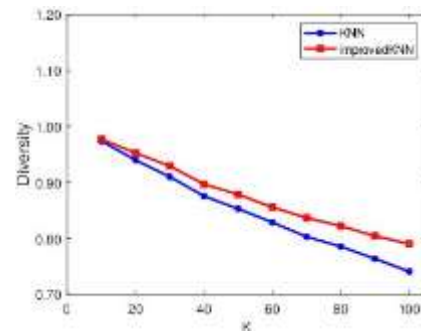
(b) The coverage of the CF algorithm on the Last.FM dataset



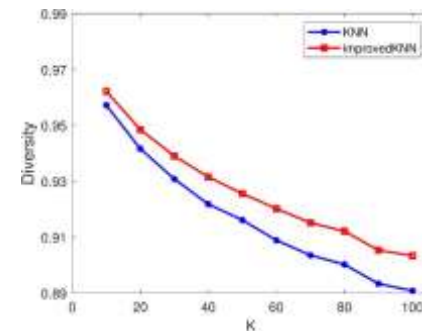
(c) The novelty of the CF algorithm on the RYM dataset



(d) The novelty of the CF algorithm on the Last.FM dataset



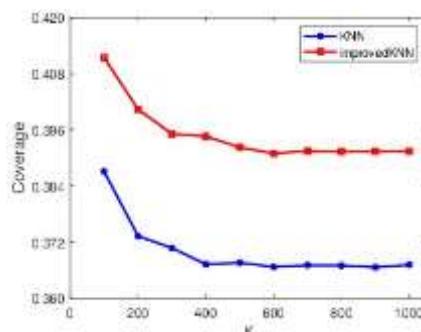
(e) The diversity of the CF algorithm on the RYM dataset



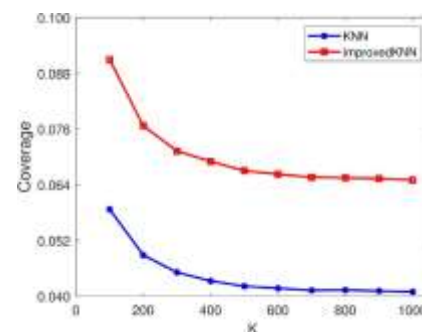
(f) The diversity of the CF algorithm on the Last.FM dataset

Figure 4: Comparison between the original CF algorithm and the improved CF algorithm in the coverage, novelty, and diversity evaluation indexes on the RYM and Last.FM datasets

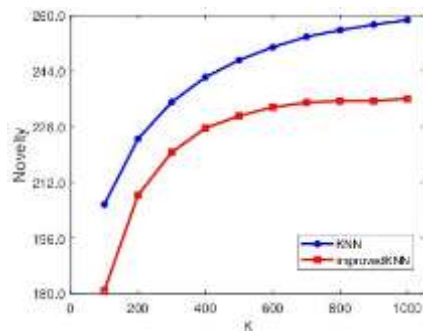
Next, we conducted experiments on MD and evaluated coverage, novelty and diversity. Compared with the CF algorithm, the performance of MD was more obvious, and its trend was basically the same as that of CF. As seen in Fig. 5, the indicators, except for novelty, of our improved MD algorithm were superior to those of the original MD algorithm.



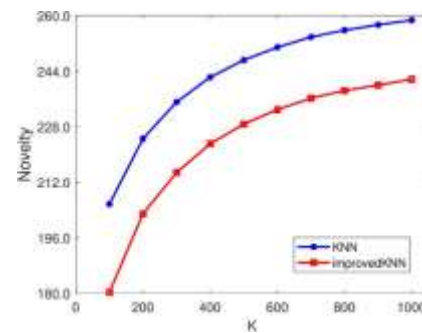
(a) The coverage of the MD algorithm on the RYM dataset



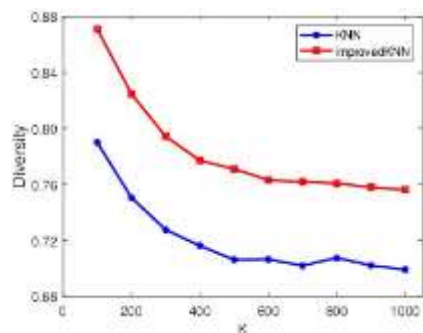
(b) The coverage of the MD algorithm on the Last.FM dataset



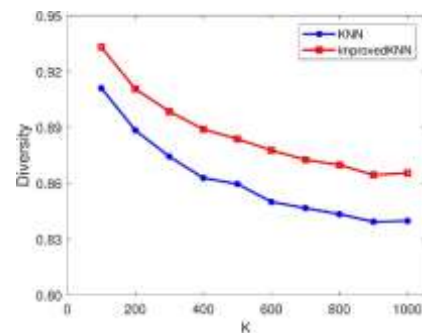
(c) The novelty of the MD algorithm on the RYM dataset



(d) The novelty of the MD algorithm on the Last.FM dataset



(e) The diversity of the MD algorithm on the RYM dataset



(f) The diversity of the MD algorithm on the Last.FM dataset

Figure 5: Comparison between the original MD algorithm and the improved MD algorithm in the coverage, novelty, and diversity evaluation indexes on the RYM and Last.FM datasets

5 Conclusion

In this paper, we proposed an algorithm framework that balances recommendation scores based on user internal similarity and applied our proposed method to both CF and MD recommendation algorithms. The experimental results we provided indicate that our hypothesis is correct, and the original recommendation scores given by recommendation systems do not fully utilize the information of a user's individual behavior patterns. We reduced the weight of the recommendation effect for users who are active but have scattered interests and increased the weight of the recommendation effect for users who have concentrated interests, especially those who are inactive but with concentrated interests. Their information cannot be used well in the original recommendation algorithm, and we proposed a method of simply multiplying the recommendation score and the user's internal similarity to mine as much of the relationship between users and the relationship between users and items as possible.

Researchers have been focusing on improving the accuracy of the recommendation system, and their proposed methods often lead to reduced coverage, novelty and diversity. The improved method we proposed not only improves the accuracy of the recommendation but also significantly improves the long-tail mining ability and the coverage and diversity of the recommendation system. We believe that our findings have a certain value for researchers to study recommendation system improvement. When faced with different recommendation system requirements, we can adjust the parameter K to obtain the desired effect.

We intend to further study how to extract the information of user behavior patterns and anticipate that the method proposed in this paper can provide some inspiration for researchers to research the recommendation system.

References

- [1] M. I. Hwang and J. W. Lin, "Information dimension, information overload and decision quality," *Journal of Information Science*, vol. 25, no. 3, pp. 213–218, 1999.
- [2] D. Das, L. Sahoo and S. Datta, "A survey on recommendation system," *International Journal of Computer Applications*, vol. 160, no. 7, 2017.
- [3] J. Lu, D. Wu, M. Mao, W. Wang and G. Zhang, "Recommender system application developments: A survey," *Decision Support Systems*, vol. 74, pp. 12–32, 2015.
- [4] G. Chen, T. Gao, X. Zhu, H. Tian and Z. Yang, "Personalized recommendation based on preferential bidirectional mass diffusion," *Physica A: Statistical Mechanics and its Applications*, vol. 469, pp. 397–404, 2017.
- [5] G. Xu, Z. Wu, Y. Zhang and J. Cao, "Social networking meets recommender systems: Survey," *International Journal of Social Network Mining*, vol. 2, no. 1, pp. 64–100, 2015.
- [6] F. Ricci, L. Rokach and B. Shapira, "Recommender systems: Introduction and challenges," in *Recommender Systems Handbook*. New York: Springer, pp. 1–34, 2015.
- [7] T. Bogers and V. Antal, "Introduction to recommender systems handbook," in *Recommender Systems Handbook*. New York: Springer, pp. 1–35, 2011.
- [8] J. Bobadilla, F. Ortega, A. Hernando and Á. Arroyo, "A balanced memory-based collaborative filtering similarity measure," *International Journal of Intelligent Systems*, pp. 939–946, 2012.
- [9] L. Lü, M. Medo, C. H. Yeung, Y. C. Zhang, Z. K. Zhang *et al.*, "Recommender systems," *Physics Reports*, vol. 519, no. 1, pp. 1–49, 2012.
- [10] T. Zhou, L. L. Jiang, R. Q. Su and Y. C. Zhang, "Effect of initial configuration on network-based recommendation," *EPL (Europhysics Letters)*, vol. 81, no. 5, 58004, 2008.
- [11] S. Pei and H. A. Makse, "Spreading dynamics in complex networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2013, no. 12, 2013.
- [12] D. C. Nie, Y. Fu, J. L. Zhou, Z. Liu, Z. K. Zhang *et al.*, "A personalized recommendation algorithm via biased random walk," in *2014 11th Int. Joint Conf. on Computer Science and Software Engineering (JCSSE)*. Thailand: IEEE, pp. 292–296, 2014.
- [13] D. C. Nie, Y. H. An, Q. Dong, Y. Fu and T. Zhou, "Information filtering via balanced diffusion on bipartite networks," *Physica A: Statistical Mechanics and its Applications*, vol. 421, pp. 44–53, 2015.
- [14] Y. H. An, Q. Dong, C. J. Sun, D. C. Nie and Y. Fu, "Diffusion-like recommendation with enhanced similarity of objects," *Physica A: Statistical Mechanics and its Applications*, vol. 461, pp. 708–715, 2016.
- [15] A. Zeng, A. Vidmer, M. Medo and Y. C. Zhang, "Information filtering by similarity-preferential diffusion processes," *EPL (Europhysics Letters)*, vol. 105, no. 5, 58002, 2014.
- [16] B. Zhang, S. Tang, Z. Ying, Y. Cai, G. Xu *et al.*, "A novel recommendation algorithm based on heterogeneous information network similarity and preference diffusion," in *Int. Conf. on WebAge Information Management*. Cham: Springer, pp. 53–64, 2015.
- [17] L. Lü and W. Liu, "Information filtering via preferential diffusion," *Physical Review E*, vol. 83, no. 6, 066119, 2011.
- [18] T. Zhou, Z. Kuscsik, J. G. Liu, M. Medo, J. R. Wakeling *et al.*, "Solving the apparent diversity-accuracy dilemma of recommender systems," *Proceedings of the National Academy of Sciences*, vol. 107, no. 10, pp. 4511–4515, 2010.
- [19] J. G. Liu, T. Zhou and Q. Guo, "Information filtering via biased heat conduction," *Physical Review E*, vol. 84, no. 3, 037101, 2011.
- [20] W. Ma, X. Feng, S. Wang and M. Gong, "Personalized recommendation based on heat bidirectional transfer," *Physica A: Statistical Mechanics and its Applications*, vol. 444, pp. 713–721, 2016.

- [21] T. Zhou, J. Ren, M. Medo and Y. C. Zhang, “Bipartite network projection and personal recommendation,” *Physical Review E*, vol. 76, no. 4, 046115, 2007.
- [22] B. Smyth and P. McClave, “Similarity vs. diversity,” in *Int. Conf. on Case-Based Reasoning*. Berlin: Springer, pp. 347–361, 2001.
- [23] N. Hurley and M. Zhang, “Novelty and diversity in top-n recommendation—analysis and evaluation,” *ACM Transactions on Internet Technology*, vol. 10, no. 4, pp. 1–30, 2011.
- [24] J. Liu, J. Li, Y. Chen, X. Chen, Z. Zhou *et al.*, “Modeling complex networks with accelerating growth and aging effect,” *Physics Letters A*, vol. 383, no. 13, pp. 1396–1400, 2019.
- [25] Y. C. Zhang, M. Blattner and Y. K. Yu, “Heat conduction process on community networks as a recommendation model,” *Physical Review Letters*, vol. 99, no. 15, 154301, 2007.
- [26] J. G. Liu, B. H. Wang and Q. Guo, “Improved collaborative filtering algorithm via information transformation,” *International Journal of Modern Physics C*, vol. 20, no. 2, pp. 285–293, 2009.
- [27] C. Liu and W. X. Zhou, “Heterogeneity in initial resource configurations improves a network-based hybrid recommendation algorithm,” *Physica A: Statistical Mechanics and its Applications*, vol. 391, no. 22, pp. 5704–5711, 2012.
- [28] G. Adomavicius and Y. Kwon, “Maximizing aggregate recommendation diversity: A graph-theoretic approach,” in *Proc. of the 1st Int. Workshop on Novelty and Diversity in Recommender System*. Citeseer, pp. 3–10, 2011.
- [29] G. Adomavicius and Y. Kwon, “Optimization-based approaches for maximizing aggregate recommendation diversity,” *INFORMS Journal on Computing*, vol. 26, no. 2, pp. 351–369, 2014.
- [30] Q. Liu and E. H. Chen, “Collaborative filtering through combining bipartite graph projection and ranking,” *Journal of Chinese Computer Systems*, vol. 5, 2010.
- [31] C. Yang, T. Liu, L. Liu and X. Chen, “A nearest neighbor based personal rank algorithm for collaborator recommendation,” in *2018 15th Int. Conf. on Service Systems and Service Management*. IEEE, pp. 1–5, 2018.