Personalized Recommendation Algorithm Based on Rating System and User Interest Association Network

Jiaquan Huang, Zhen Jia*

College of Science, Guilin University of Technology, Guilin, China
Email: *jjjzzz0@163.com

Abstract

In most available recommendation algorithms, especially for rating systems, almost all the high rating information is utilized on the recommender system without using any low-rating information, which may include more user information and lead to the accuracy of recommender system being reduced. The paper proposes a algorithm of personalized recommendation (UNP algorithm) for rating system to fully explore the similarity of interests among users in utilizing all the information of rating data. In UNP algorithm, the similarity information of users is used to construct a user interest association network, and a recommendation list is established for the target user with combining the user interest association network information and the idea of collaborative filtering. Finally, the UNP algorithm is compared with several typical recommendation algorithms (CF algorithm, NBI algorithm and GRM algorithm), and the experimental results on Movielens and Netflix datasets show that the UNP algorithm has higher recommendation accuracy.

Keywords

Recommender Systems, Association Network, Similarity, Bipartite Network, Collaborative Filtering

1. Introduction

With the spread of smartphones and the arrival of the information revolution, people can access a large amount of online information more and more conveniently and conduct frequent online social and business activities, such as online shopping, online movie watching, VR viewing houses, and so on. However, in the process of accessing information, people inevitably search for many redun-

dant or unsuitable information, which is known as Information Overload problem [1] [2] [3]. With the explosive growth of data on the Internet, the information overload problem is increasing. Compared with computers, the human brain's ability to process information is extremely limited, so how to filter the huge amount of data and quickly get the information they need has become an urgent need for Web users, and then the recommender systems [4] [5] have emerged. In the recommender systems, a series of technical analysis and computation methods were used to recommend items of interest to users based on information about the response between the user and the item [6] [7]. In daily life, the shadow of recommendation system can be seen everywhere, such as music recommendation in music platform, movie recommendation in movie platform, item recommendation in shopping platform, attraction recommendation in travel platform, and so on.

Since Resnick and Varian proposed recommendation systems in 1997 [8], there has been a frenzy of scientific research on recommender systems. At present, Collaborative Filtering (CF) algorithm, GRM algorithm and NBI algorithm are the typical recommendation algorithms [9]. After a paper on collaborative filtering algorithm was published by Amazon in 2003 [10], the CF algorithm has been known and widely used and become one of the most popular typical recommendation algorithms. In CF algorithm, the information about the behavior of user is used to analyze the correlation between users or items and thus obtain the interest of the target user. The GRM algorithm was a global sorting algorithm whose main idea is to sort items in descending order of popularity and recommend the top-ranked items to the target users. The NBI algorithm was proposed by Zhou [9] in 2007, which is based on the idea of resource allocation to quantitatively calculate the similarity of users in order to predict the items preferred by the target users.

Traditional recommendation algorithms are constrained by problems such as cold start [11] and data sparsity [12], which make them difficult to obtain desirable results in practical applications. In a few years, scholars have investigated around the problem of how to improve the recommendation accuracy, and many methods to improve and extend the classical algorithm have been proposed. One class of methods is the recommendation algorithm proposed based on the network structure perspective. Based on the behavioral influence of e-reading groups, a collaborative filtering recommendation algorithm was proposed by Liu *et al.*, which can effectively alleviate the sparsity problem and improve the recommendation quality of the recommendation system [13]. A recommendation algorithm based on deep neural networks was proposed by Bi *et al.*, which can be a solution to handle the data sparsity issues and cold start issues. Bi *et al.* used deep neural network to build a regression model to predict user ratings and give recommendation lists [14]. Tian *et al.* established a multi-subnet complex complex network (MSCCN) with multi-user relationships and proposed an intelligent recommendation algorithm based on user-item bipartite graph and quality diffusion (MD) algorithm, which effectively improved the accuracy of the rec-

ommendation system [15]. To improve the efficiency and diversity of the recommender system, Cui *et al.* proposed a new multi-objective evolutionary algorithm (PMOEA), and the PMOEA achieves a good balance between accuracy and diversity [16]. To solve the data sparsity problem, Liu *et al.* proposed a algorithm that is based on personal trust and item similarity [17]. Liu projected the bipartite network into a single mode network and proposed an e-commerce recommendation algorithm [18]. Another is the recommendation algorithm proposed by analyzing the characteristics of users and the links among users. Zhang *et al.* presented a recommendation algorithm based on labeling network [19]. They divided users' ratings of items into positive and negative to recommend items of interest to the target users. Li *et al.* combined the ideas of social networks and collaborative filtering to propose a collaborative filtering algorithm based on community detection [20]. Li *et al.* based on a new community detection algorithm to mine the user-item network for inter-user social relationship and select a part of user community as the set of neighbor candidates as users, which effectively improves the efficiency of the recommendation system. Chen *et al.* proposed a recommendation algorithm based on optimized user similarity [21], which is increased to the conventional cosine similarity with a balancing factor to classify the scale differences in item ratings by different users. To solve the problems of data sparsity, cold start, and scalability in recommender systems, Duan *et al.* took into account the varying levels of user trust in their study of the effect of trust relationships on user feature vectors. They then presented a recommendation algorithm that integrates the dual effects of trust relationships and expert users (ETBRec) [22].

Only the high rating information is utilized by the current recommendation algorithms on rating systems utilize, however, the low rating information is ignored. In a 1 - 5 rating system, most of the high ratings of only 3 - 5 are used as valid information, while information on ratings of 1 - 2 is not considered. This will lose some of the intrinsic information implied by the scoring data and may have an impact on the performance of the recommender system. An algorithm (UNP algorithm) for personalized recommendations is suggested on rating systems in this paper. The UNP algorithm is designed by fully mining user rating information based on the establishment of a user interest association network combined with the idea of collaborative filtering. Due to the fact that every user rating on items has been used, the similarity of interests among users is fully reflected, thus effectively improving the recommendation effect. The experimental results on Movielens and Netflix datasets demonstrate that the accuracy of UNP algorithm is greatly improved compared with CF algorithm, GRM algorithm and NBI algorithm, in which the best accuracy improvement rate reaches 71.83%.

The structure of this paper is as follows. Section 2 briefly introduces the foundation of bipartite networks. In Section 3, the UNP algorithm in this paper is introduced in detail, and its specific algorithmic steps are given. In Section 4, the experimental comparison results of UNP algorithm with CF algorithm, GRM algorithm and NBI algorithm are presented. Finally, conclusions are discussed in

Section 5.

2. Preliminaries

In general, a bipartite network [23] can be used to depict the connection between users and items in recommender systems. For a bipartite network $G(E,V)$, where $V = V_u \cup V_o$, the V_u and V_o are respectively the set of two disjoint nodes in G. E is the set of connected edges between V_u and V_o. Assume that the recommendation system has n users that constitute the V_u, which is denoted as $V_u = u_i (i = 1, 2, \cdots, n)$, and m items that constitute the V_o, denoted as $V_o = o_j (j = 1, 2, \cdots, m)$. Subsequently, the matrix A can express the bipartite network G.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}, \quad (1)$$

where $a_{ij} = 1$ or 0 ($i = 1, 2, \cdots, n$, $j = 1, 2, \cdots, m$). $a_{ij} = 1$, which means u_i has selected o_j, otherwise $a_{ij} = 0$.

In a rating system, it is assumed that the u_i's rating of the o_j is denoted as a_{ij}^r, the ratings of m items by n users can be used as a rating matrix A^r.

$$A^r = \begin{bmatrix} a_{11}^r & a_{12}^r & \cdots & a_{1m}^r \\ a_{21}^r & a_{22}^r & \cdots & a_{2m}^r \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}^r & a_{n2}^r & \cdots & a_{nm}^r \end{bmatrix}, \quad (2)$$

In recommendation algorithms, it is generally necessary to measure the similarity of interests among users, and different recommendation algorithms consider different metrics from various perspectives. In this paper, the Euclidean Distance is used to establish a user interest association network $G_u(E_u, V_u)$, where V_u is the set of users in the bipartite network G_u, and E_u is the set of edges in G_u, which represents the relevance of interests between users. The G_u can be represented by the matrix B_u.

$$B_u = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{bmatrix}, \quad (3)$$

where $b_{ij} = 1$ or 0 ($i, j = 1, 2, \cdots, n$). if there is a strong correlation of interest between the u_i and u_j, then $b_{ij} = 1$, otherwise $b_{ij} = 0$.

There is an example in **Figure 1**. **Figure 1(a)** shows a user-item bipartite network with the values on its edges being the user's ratings of the items, then we can obtain the corresponding rating matrix (A^r) of the network (As shown in **Figure 1(b)**) and the matrix representation of the network (As shown in **Figure 1(c)**).
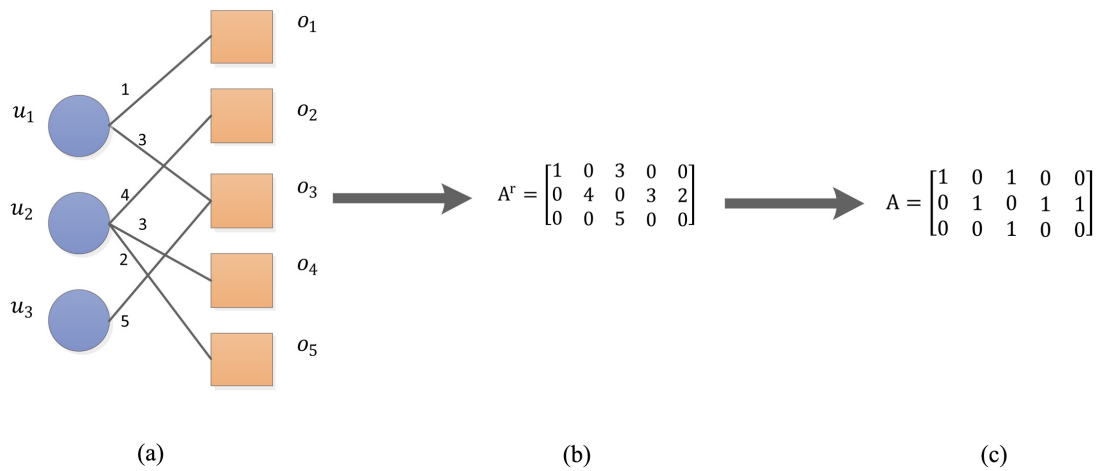
Figure 1. An example of bipartite network G, (a) user-item bipartite network; (b) scoring matrix A^r; (c) the matrix of bipartite network.

3. UNP Algorithm

In general, the connection between users and items is regarded as a bipartite network in the recommendation system. In this paper, a user interest correlation network is added to the bipartite network of users and items, and then the information of these two networks is combined to explore the information of items of interest to the target users, so as to realize personalized recommendation for the users. The UNP algorithm mainly targets the rating system to perform personalized recommendation. In the UNP algorithm, the Euclidean Distance between users is employed to analyze the similarity of users. By setting a filtering condition, the strong correlation information between users' interests is obtained, for which a user interest association network is constructed. Based on the information of the user interest association network and the idea of combining collaborative filtering, the behavior records of users with the same interest as the target user are analyzed in the network to predict the items that are most probably to be enjoyed by the user, so as to achieve personalized recommendation. Figure 2 displays the UNP algorithm's flowchart, and the UNP algorithm's precise steps are as follows.

Step 1: Calculate Euclidean Distance

In the rating system, the rating information of users on items is relatively easy to obtain. Under the assumption that users are rational, the rating information can accurately reflect the users' liking of the selected items, and therefore the rating information is used very frequently. The Euclidean distance between users will be calculated by the rating information of users on items, with a larger distance indicating a weaker interest correlation between users and vice versa. The Euclidean distance s_{ij}^{u} between user u_i and user u_j is calculated as shown below.

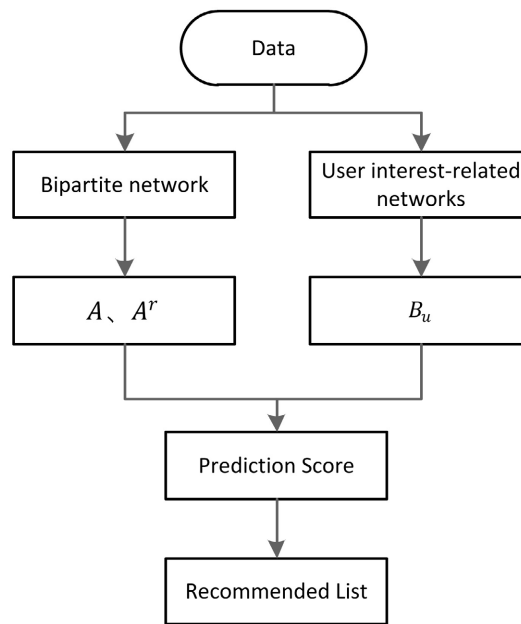$$s_{ij}^{u} = \sqrt{\sum_{l=1}^{m}\left(a_{il}^{r} - a_{jl}^{r}\right)^2},\tag{4}$$

Figure 2. The flow chart of UNP algorithm.

where m is the number of items in the recommender system, a_{il}^r is an element of A^r in Equation (2), and obviously $s_{ii}^u = 0$.

Step 2: Build user interest association network

By analyzing the connections between users and items, we can analyze the items that target users are interested in more precisely. Therefore, we would like to incorporate the user-to-user association information to improve the accuracy of the recommender system. In this section, the strength of the connection between users is measured by the Euclidean distance, and from this, a user interest association network is built to express the strength of interest relevance between users.

In Equation (4), a larger s_{ij}^u implies a wider interest variability between u_i and u_j and vice versa. The algorithm sets a threshold (α) to discern the strength of interest similarity between two users, which plays a filtering role on the interest relevance of two users, and thus establishes a user interest association network. Suppose the user interest association network matrix B_u, where the elements b_{ij} in matrix B_u are shown as follows.

$$b_{ij} = \begin{cases} 1, s_{ij}^u \le \alpha \\ 0, \text{others} \end{cases}, \tag{5}$$

where the Euclidean Distance $s_{ij}^u \le \alpha$ between user u_i and user u_j is regarded as a strong correlation between the two user interests, corresponding to $b_{ij} = 1$, and conversely $b_{ij} = 0$ indicates that the interests between users are weakly correlated (filtered out in the calculation).

Step 3: Calculating prediction scores

In the user interest association network, the neighbor nodes of each user are the strongly related users with matching interests. Combining the information of

the bipartite network, the forecast scores of the users for the unselected items are calculated as follows.

$$w_{ij}^u = \frac{\sum_{l=1,l\neq i}^n b_{li} a_{lj}^r}{\sum_{l=1,l\neq i}^n b_{li}},$$ (6)

where w_{ij}^u is the predicted rating of user u_i for the unselected item (o_j), a_{lj}^r is the element of the rating matrix A^r in Equation (2), and b_{li} is the element of the matrix B_u in Equation (3).

Step 4: Give the recommendation list

The predicted ratings of all the unselected items of u_i are listed with descending order, and the top L items will be constituted into the recommendation list of u_i and recommended to u_i. The concrete implementation of the UNP algorithm is shown in **Algorithm 1**.

4. Experimental Results and Analysis

For the purpose of measuring the effectiveness of the UNP algorithm, two benchmark datasets Movielens and Netflix were used for testing experiments. The CF algorithm, GRM algorithm and NBI algorithm are also applied to compare the effects.

4.1. Experimental Dataset

There are 100,000 ratings (the score of 1 - 5), 943 users and 1682 movies on the MovieLens dataset. The Netflix dataset contains 197,248 ratings (the score of 1 - 5), 3000 users and 2779 movies. In general, the dataset is randomly divided into two components (training set and test set) with a ratio of 9:1. More detailed information about the datasets is given in **Table 1**.

1: **Input**: A^r
2: **Output**: Recommendation list L
3: $A \Leftarrow FunctionGetA(A^r)$
4: $ED \Leftarrow Get_EuclideanDistance(A^r)$
5: **for** i = 1 to n **do**
6:     **for** i = 1 to m **do**
7:         $b_{ij} \Leftarrow FunctionGetb(ED)$
8:     **end for**
9: **end for**
10: $B_u \Leftarrow \{b_{ij}\}$
11: **for** i = 1 to n **do**
12:     **for** i = 1 to m **do**
13:         $w_{ij}^u \Leftarrow FunctionGetw(ED, B_u)$
14:     **end for**
15: **end for**
16: $W \Leftarrow \{w_{ij}^u\}$
17: $SortW \Leftarrow FunctionSort(W)$
18: **for** i = 1 to n **do**
19:     $L(u_i) \Leftarrow FunctionGetL(SortW)$
20: **end for**

Algorithm 1. UNP algorithm.

Table 1. The detailed information of datasets.

Dataset	n	m	Rating	Sparsity
Movielens	943	1682	100,000	0.9370
Netflix	3000	2779	197,248	0.9763

In **Table 1**, n, m, Rating and Sparsity represent the number of users, the number of movies, the number of ratings and sparsity respectively.

4.2. Evaluation Metrics

In general, the actual effectiveness of an algorithm needs to be measured by some evaluation metrics. The commonly used evaluation metrics for the accuracy of recommendation algorithms are ranking accuracy, hit rate and recall rate.

1) Sorting accuracy

$$r_{ij} = \frac{l_{ij}}{L}, \tag{7}$$

where r_i is the sorting accuracy of o_j in the test set, l_i is the ranking of o_j in the recommendation list, and L is the recommendation list's length. The $\bar{r}$ is defined as the average ranking accuracy of all users in the recommender system.

$$\bar{r} = \frac{\sum_{i=1}^{M} r_i}{M}, \tag{8}$$

2) Hit rate

$$p(u_i) = \frac{\Gamma(u_i) \cap T(u_i)}{L}, \tag{9}$$

where $p(u_i)$ is the hit rate of u_i, $\Gamma(u_i)$ is the set of items in the recommendation list of u_i, $T(u_i)$ is the set of items in the test set of u_i, and L is the length of the recommendation list. The $\bar{p}$ is defined as the average hit rates.

$$\bar{p} = \frac{\sum_{i=1}^{n} p(u_i)}{n}. \tag{10}$$

3) Recall rate

$$Recall(u_i) = \frac{|\Gamma(u_i)| \cap |T(u_i)|}{|T(u_i)|}, \tag{11}$$

where $Recall(u_i)$ is the recall of u_i and $|T(u_i)|$ denotes the set of items in the test set. The ($\overline{Recall}$) is defined as the average recall rates.

$$\overline{Recall} = \frac{\sum_{i=1}^{n} Recall(u_i)}{n}. \tag{12}$$

4.3. Determining the α

In this section, the α (the threshold value of user interest relevance) will be addressed how to get and the results of the experiments are displayed in **Figure 3** and **Figure 4**.

Figure 3 and **Figure 4** show the Euclidean Distance plots between users in Netflix and movie, respectively. In **Figure 3**, the black curve is the actual Euclidean Distance, and the red straight line is the average Euclidean Distance of all
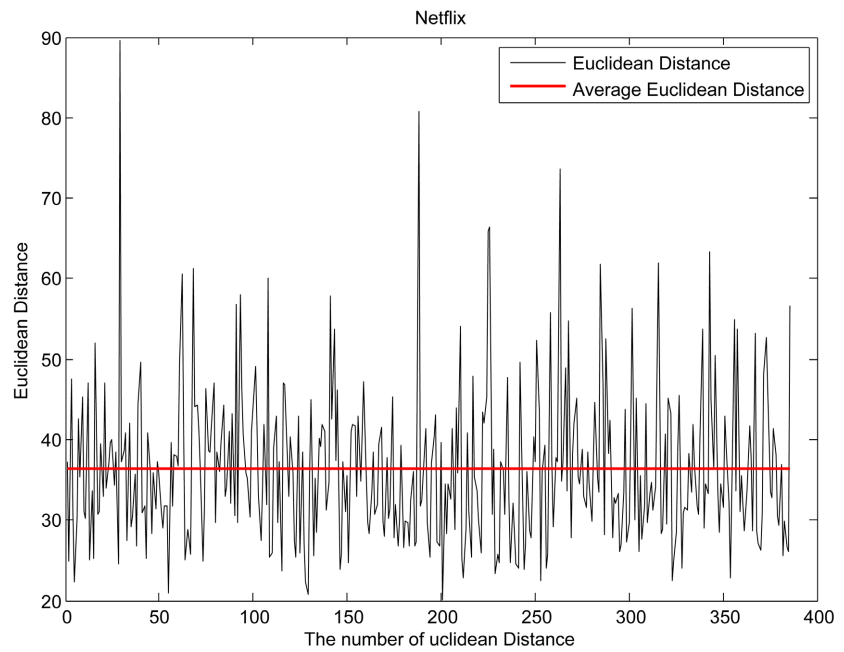

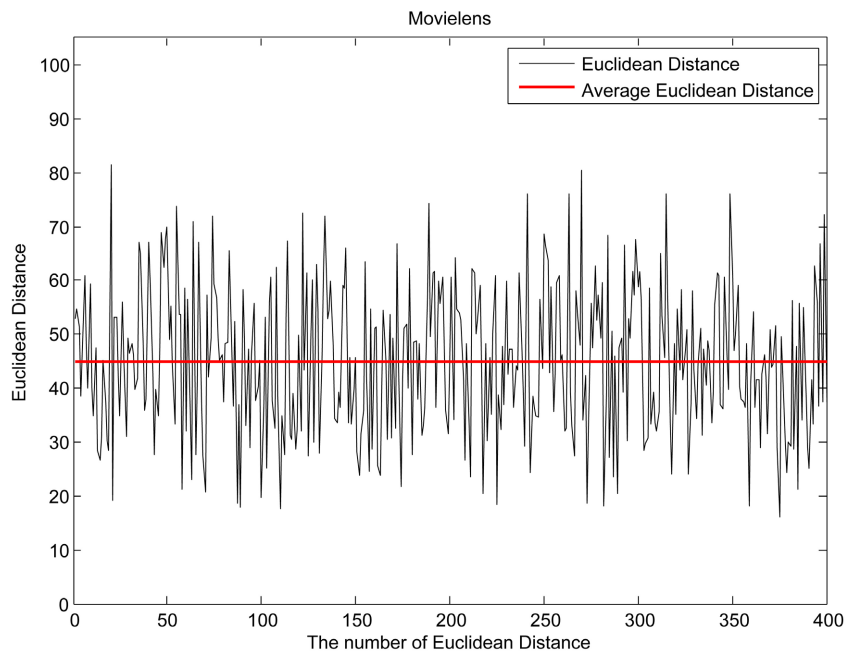**Figure 3.** Euclidean distance of users on the Netflix.

Figure 4. Euclidean distance of users on the Movielens.

data in Netflix. From **Figure 3**, we can see that the red straight line is in the middle of the black curve, and the ratio of the black curve image below the red straight line to the whole black curve image is 0.5859, which is greater than 0.5 after several random independent repetitions of the experiment, so the threshold α in the Netflix dataset is 35.9513.

Similar to **Figure 3**, the corresponding threshold α in the Movielens dataset is 44.7746. Meanwhile, the analysis from **Figure 3** and **Figure 4** show that in constructing the user interest association network it is reasonable to select the threshold as its corresponding average Euclidean Distance.

4.4. Compare the Accuracy of the Algorithms

In this part, the hit rate, recall and ranking accuracy of UNP algorithm, CF algorithm, GRM algorithm and NBI algorithm are calculated in the experiment and the results are compared.

1) Hit rate

In recommendation algorithm, the higher hit rate means the higher recommendation performance and the final experimental results are shown in **Figure 5**.

Figure 5(a) shows the graphs of the average hit rate of the four algorithms (UNP, CF, GRM, and NBI algorithms) on the Netflix dataset, and **Figure 5(b)** shows the graphs of the average hit rate on the Movielens dataset, where the horizontal coordinates of the two graphs are the length of L. From **Figure 5(a)** and **Figure 5(b)**, we can see that the NBI algorithm has the highest hit rate on both datasets, and the UNP algorithm has a higher hit rate than the NBI algorithm. The hit rate of GRM algorithm is the highest on both data sets, and the hit rate of UNP algorithm is higher than that of GRM algorithm.

2) Recall rate

The same with Hit rate, a higher recall also means a higher recommendation precision and the final experimental results obtained are shown in **Figure 6**.

Figure 6(a) presents the graphs of the average recall rate of the four algorithms (UNP, CF, GRM, and NBI algorithms) on Netflix, and **Figure 6(b)** presents the graphs of the average recall rate of the four algorithms on the Movielens, where the horizontal coordinates of the two graphs are the length of the recommendation list L. From **Figure 6(a)** and **Figure 6(b)**, we can see that the recall of UNP algorithm is higher than GRM algorithm on Netflix dataset.

3) Sorting accuracy

The sorting accuracy is calculated by Equations (7) and (8), and the results obtained are shown in **Figure 7**, and the average sorting accuracy is shown in **Table 2**.

Figure 7(a) plots the sorting accuracy of four algorithms (UNP algorithm, CF algorithm, GRM algorithm, and NBI algorithm) on the Netflix, and **Figure 7(b)** plots the sorting accuracy of four algorithms on the Movielens. From **Figure 7** and **Table 2**, we can see that the UNP algorithm is the best among these algorithms.
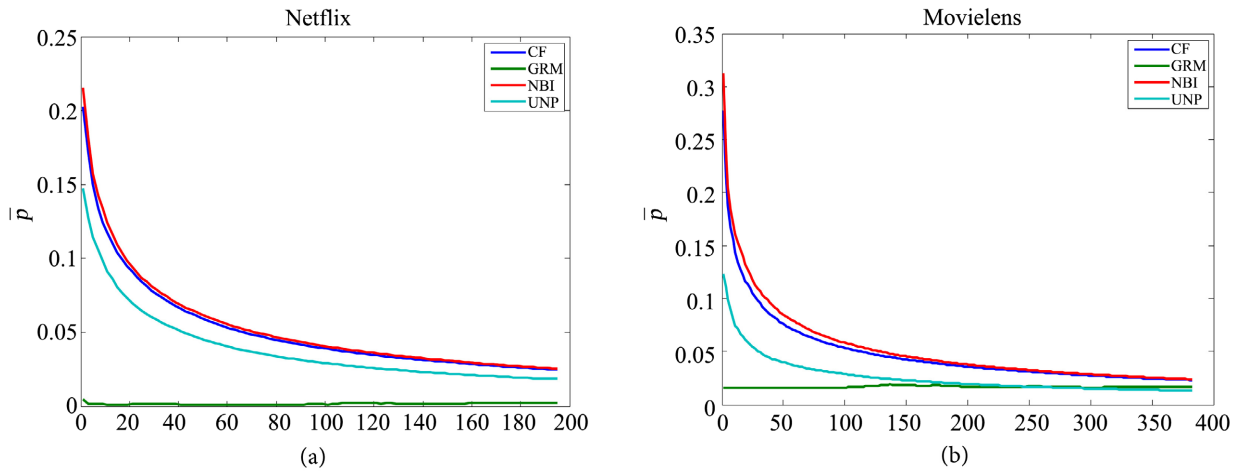
Figure 5. The hit rate of UNP algorithm, CF algorithm, GRM algorithm and NBI algorithm. (a) The hit rate of the Netflix; (b) The hit rate of the Movielens.
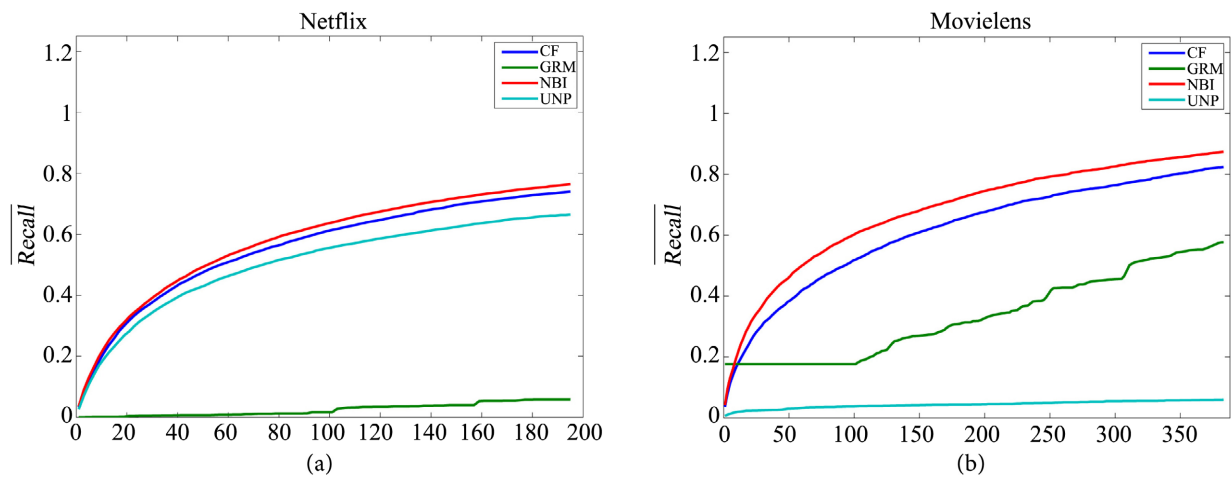

Figure 6. The recall rate of UNP algorithm, CF algorithm, GRM algorithm and NBI algorithm. (a) The recall rate of the Netflix; (b) The recall rate of the Movielens.
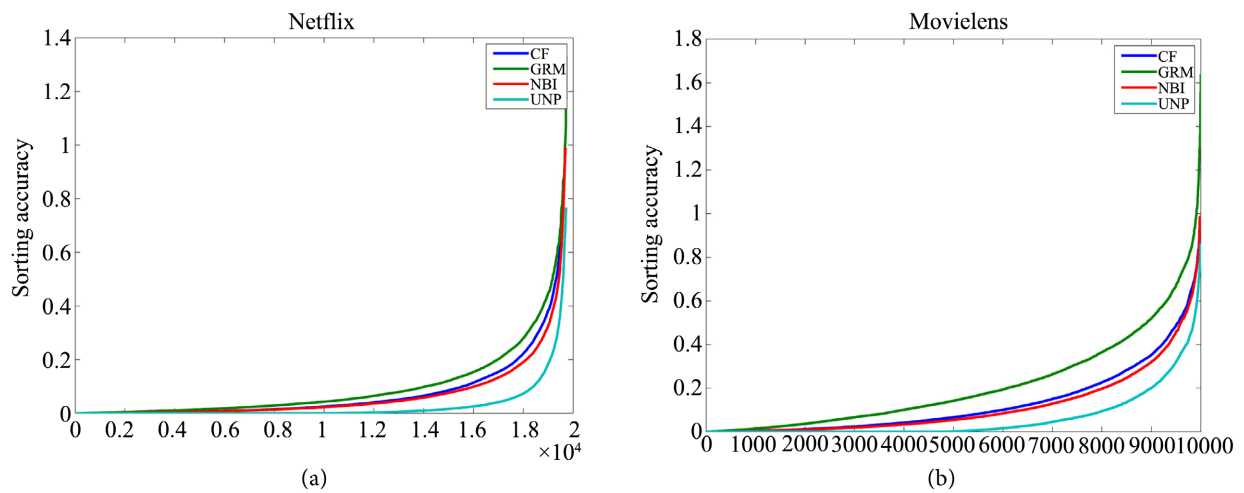

Figure 7. The sorting accuracy of UNP algorithm, CF algorithm, GRM algorithm and NBI algorithm. (a) The sorting accuracy of the Netflix; (b) The sorting accuracy of the Movielens.

Table 2. The average sorting accuracy.

Dataset	Movielens	Netflix
UNP	0.0606	0.0258
CF	0.1315 (↑ 53.9%)	0.0739 (↑ 65.1%)
GRM	0.2151 (↑ 71.8%)	0.0739 (↑ 65.1%)
NBI	0.1174 (↑ 48.4%)	0.0651 (↑ 60.4%)

5. Conclusion

In this paper, we propose the UNP algorithm with higher accuracy for rating systems. Compared with previous related algorithms, UNP algorithm has two advantages. On the one hand, the UNP algorithm makes use of all the rating data, and the information implied by the rating data is more fully explored. On the other hand, the UNP algorithm is a correlation network of users' interests based on the rating distance between users, and the similarity of users' interests can be more objectively reflected by the obtained network, thus better recommendation results can be obtained. The experimental results on the Netflix and Movielens show that the UNP algorithm has the best recommendation effect in terms of the algorithm's ranking accuracy, and its recommendation effect is up to 73.43% higher than the GRM algorithm. Therefore, compared with the CF algorithm, GRM algorithm, and NBI algorithm, the UNP algorithm achieved better recommendation precision. In future research work, a correlation network of items' similarity based on the rating distance will be considered, and other methods will be used to calculate the similarity of users and the similarity of items.

Acknowledgements

Conflicts of Interest

The authors declare there are no conflicts of interest.

References

[1] Peng, W. and Xin, B. (2019) A Social Trust and Preference Segmentation-Based Matrix Factorization Recommendation Algorithm. *EURASIP Journal on Wireless Communications and Networking*, **2019**, Article No. 272. https://doi.org/10.1186/s13638-019-1600-4

[2] Polatidis, N. and Georgiadis, C.K. (2016) A Multi-Level Collaborative Filtering Method That Improves Recommendations. *Expert Systems with Applications*, **48**, 100-110. https://doi.org/10.1016/j.eswa.2015.11.023

[3] Cai, B., Yang, X., Huang, Y., Li, H. and Sang, Q. (2018) A Triangular Personalized Recommendation Algorithm for Improving Diversity. *Discrete Dynamics in Nature and Society*, **2018**, Article ID: 3162068. https://doi.org/10.1155/2018/3162068

[4]   Zhuang, F., Zheng, J., Chen, J., Zhang, X., Shi, C. and He, Q. (2018) Transfer Collaborative Filtering from Multiple Sources via Consensus Regularization. *Neural Networks*, **108**, 287-295. https://doi.org/10.1016/j.neunet.2018.08.022

[5]   Liao, C.L. and Lee, S.J. (2016) A Clustering Based Approach to Improving the Efficiency of Collaborative Filtering Recommendation. *Electronic Commerce Research and Applications*, **18**, 1-9. https://doi.org/10.1016/j.elerap.2016.05.001

[6]   Snchez-Moreno, D., Gil Gonzlez, A.B., Muoz Vicente, M.D., Lpez Batista, V.F. and Moreno Garca, M.N. (2016) A Collaborative Filtering Method for Music Recommendation Using Playing Coefficients for Artists and Users. *Expert Systems with Applications*, **66**, 234-244. https://doi.org/10.1016/j.eswa.2016.09.019

[7]   Koohi, H. and Kiani, K. (2016) User Based Collaborative Filtering Using Fuzzy C-Means. *Measurement*, **91**, 134-139.
https://doi.org/10.1016/j.measurement.2016.05.058

[8]   Resnick, P. and Varian, H.R. (1997) Recommender Systems. C*ommunications of the ACM*, **40**, 56-58. https://doi.org/10.1145/245108.245121

[9]   Zhou, T., Ren, J., Medo, M. and Zhang, Y.C. (2007) Bipartite Network Projection and Personal Recommendation. *Physical Review E*: *Covering Statistical, Nonlinear, Biological, and Soft Matter Physics*, **76**, Article No. 046115.
https://doi.org/10.1103/PhysRevE.76.046115

[10] Linden, G., Smith, B. and York, J. (2003) Amazon.com Recommendation: Item-to-Item Collaborative Filtering. *IEEE Internet Computing*, **7**, 76-80.
https://doi.org/10.1109/MIC.2003.1167344

[11] Bobadilla, J., Ortega, F., Hernando, A. and Bernal, J. (2012) A Collaborative Filtering Approach to Mitigate the New User Cold Start Problem. *Knowledge-Based Systems*, **26**, 225-238. https://doi.org/10.1016/j.knosys.2011.07.021

[12] Salah, A., Rogovschi, N. and Nadif, M. (2016) A Dynamic Collaborative Filtering System via a Weighted Clustering Approach. *Neurocomputing*, **175**, 206-215.
https://doi.org/10.1016/j.neucom.2015.10.050

[13] Liu, X. (2019) A Collaborative Filtering Recommendation Algorithm Based on the Influence Sets of E-Learning Group's Behavior. *Cluster Computing*, **22**, 2823-2833.
https://doi.org/10.1007/s10586-017-1560-6

[14] Bi, J.W., Liu, Y. and Fan, Z.P. (2019) A Deep Neural Networks Based Recommendation Algorithm Using User and Item Basic Data. *International Journal of Machine Learning and Cybernetics*, **11**, 763-777. https://doi.org/10.1007/s13042-019-00981-y

[15] Tian, G., Zhou, S., Sun, G., Chen, C.-C. and Chen, C.-H. (2020) A Novel Intelligent Recommendation Algorithm Based on Mass Diffusion. *Discrete Dynamics in Nature and Society*, **2020**, Article ID: 4568171. https://doi.org/10.1155/2020/4568171

[16] Cui, L., Ou, P., Fu, X., Wen, Z. and Lu, N. (2017) A Novel Multi-Objective Evolutionary Algorithm for Recommendation Systems. *Journal of Parallel and Distributed Computing*, **103**, 53-63. https://doi.org/10.1016/j.jpdc.2016.10.014

[17] Liu, T. and He, Z. (2021) A Novel Personalized Recommendation Algorithm by Exploiting Individual Trust and Item's Similarities. *Applied Intelligence*, **52**, 6007-6021.
https://doi.org/10.1007/s10489-021-02655-1

[18] Liu, G. (2022) An Ecommerce Recommendation Algorithm Based on Link Prediction. *Alexandria Engineering Journal*, **61**, 905-910.
https://doi.org/10.1016/j.aej.2021.04.081

[19] Zhang, P., Song, X., Xue, L. and Gu, K. (2019) A New Recommender Algorithm on Signed Networks. *Physica A*: *Statistical Mechanics and Its Applications*, **520**, 317-321.

https://doi.org/10.1016/j.physa.2019.01.054

[20] Li, X. and Li, D. (2019) An Improved Collaborative Filtering Recommendation Algorithm and Recommendation Strategy. *Mobile Information Systems*, **2019**, Article ID: 3560968. https://doi.org/10.1155/2019/3560968

[21] Chen, H., Li, Z. and Hu, W. (2015) An Improved Collaborative Recommendation Algorithm Based on Optimized User Similarity. *The Journal of Supercomputing*, **72**, 2565-2578. https://doi.org/10.1007/s11227-015-1518-5

[22] Duan, Z., Xu, W., Chen, Y. and Ding, L. (2021) ETBRec: A Novel Recommendation Algorithm Combining the Double Influence of Trust Relationship and Expert Users. *Applied Intelligence*, **52**, 282-294. https://doi.org/10.4236/jmf.2021.111005

[23] Tong, H., Jia, Z., Zhang, M. and Qi, J. (2021) Analysis of Stock-Shareholder Associated Network Based on Complex Network. *Journal of Mathematical Finance*, **11**, 107-122. https://doi.org/10.4236/jmf.2021.111005