IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 13, 2020, accepted August 13, 2020, date of publication August 18, 2020, date of current version August 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3017701

Decentralized Collaborative Filtering Algorithms Based on Complex Network Modeling and Degree Centrality

JUN AI[1,2], ZHAN SU[1,2], KAILI WANG[1], CHUNXUE WU[1], AND DUNLU PENG[1]

[1] School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China
[2] Shanghai Key Lab of Modern Optical Systems, University of Shanghai for Science and Technology, Shanghai 200093, China

Corresponding authors: Jun Ai (aijun@outlook.com) and Zhan Su (suzhan@foxmail.com)

This work was supported by the Young Scientists Fund of the National Natural Science Foundation of China under Grant 61803264.

ABSTRACT Given that everyone online is saturated with information, the theoretical significance of recommendation algorithms is evident in the fact that users need help finding products and content they care about. Collaborative filtering predicts a user's rating on an item by finding similar users that rated the item or similar items that were rated by the user, and using the selected similar neighbors to ''collaboratively filter'' the recommendation. In the process, selected neighbors are considered equally important despite their differences in popularity. Here, we explore a method of modeling recommender systems as networks that can be constructed by considering items as nodes and similarity between them as links. Our research shows that item centrality has a negative impact on the accuracy of rating predictions, which needs to be considered for better algorithm performance. Experiments show that collaborative filtering algorithms can be decentralized by our method and provide a better accuracy of rating prediction. Furthermore, the relationship between the prediction target and its neighbors can be further evaluated based on both their similarity and their centrality.

INDEX TERMS Recommender systems, complex networks, decentralized collaborative filtering, degree centrality, network modeling.

I. INTRODUCTION

Recommender Systems (RSs) have received increasing attention in recent years. The systems have critical roles for most websites and e-commerce sites, such as online dating [1], movie recommendations [2]–[6], the evaluation of temporal networks [7], [8], collaborative recommendations [9] and so on. Technology titans like Amazon, Netflix and Taobao are using similar systems and algorithms to find new customers as well as selling more merchandise to old ones.

Consequently, recommendation algorithms have become one of the hot topics and have been studied in the past decade by scientists and engineers from different backgrounds. Various algorithms are being explored and presented by the community. Computer science researchers resolve the problem by using collaborative filtering and content-based filtering. The main ideas of those algorithms are making automatic predictions about users' preferences or items' potential customers. The predictions are made based on the collected information of many other users or other items, respectively [10], [11].

In contrast, physicists attempt to solve the problem with users and items considered as nodes in networks, and to predict the connection and weight of the connection between a user and an item, also known as the link prediction [12], [13].

Recommendation algorithms can be classified into three types based on the main data sources used in predictions [11], [14]–[16].

- Content-based recommendation algorithms
- Collaborative filtering recommendation algorithms
- Hybrid recommendation algorithms

Content-based (CB) recommendation algorithms focus on finding items with similar content or characteristics to items positively rated by target users [17], [18]. Data used in CB prediction is item content, such as the key words of papers, genres of movies and musics and so on. People's preference on items with such content is the foundation of possible recommendations.

The associate editor coordinating the review of this manuscript and approving it for publication was Chaitanya U. Kshirsagar.

Similarly, Collaborative Filtering (CF) methods treat the recommendation problem as making automatic predictions (also known as filtering) about the interests of the target user by collecting preferences or taste information from many other users (also known as collaborating) [19]. The ''collaborative'' refers to the collaboration effort of a group of users to ''filter'' the abundant data and make recommendations to the target user. Data used in CF methods is ratings or likes from users on different items. A group of similar users or items is the foundation to make the prediction.

Furthermore, the hybrid algorithms combine different content-based and CF methods to exploit information of both users and items [20], [21]. Various methods are used in this category to improve the recommendation the accuracy of the recommendations or improve the performance [22], [23]. Or targeting specific problems like cold-start problems [5], [24] or sparsity of data [15], [24], [25].

In addition, CF algorithms can be further classified into two subcategories based on the methods used in prediction [10], [12], [21], [26].

- Memory-based CF
- Model-based CF

The memory-based algorithms perform predictions based on the training data set, which is usually loaded into the memory [15], [23]. Similarity is usually calculated between each pair of users or items in this type of CF. And top-k neighbors with the highest similarity are selected for the target user u [2]. However, some literature uses the k Nearest Neighbor (*kNN*) algorithm to find the most similar neighbors for the target user [10], [23].

The model-based CF algorithms aim to find patterns in the data and construct a model to predict new ratings [27]. Models like Matrix Factorization (MF) methods, slope one algorithms, complex network modeling, Bayesian probabilistic and neural networks are all used in different approaches and backgrounds.

However, complex network modeling and node centrality have not been used together to explore possible improvements in recommender systems in recent works.

Our main contributions are presented based on the previous analysis as follows:

1) **We explore the centrality of nodes in complex network models to improve rating prediction in RS.**
2) **Our research on item-item network modeling in recommender systems indicates popular items is overrated in the prediction process and need a corresponding correction.**
3) **A framework is proposed to improve the accuracy of most memory-based CF methods.**

II. RELATED BACKGROUND

A. COMPLEX NETWORK MODELING AND CENTRALITY

Networks of coupled dynamical systems have been used to model various complex systems [28], such as biological oscillators, excitable media, neural networks, software structures [29]–[31], spatial games, genetic control networks, collaboration networks in science [32], real-world airport network [33], the Internet topology [34] and even recommender systems [4], [35].

Studies [4] and [5] both use complex networks to model recommender systems as an item-item or a user-user network. Prediction of ratings in those algorithms takes network characteristics into consideration. For example, [5] constructed an item-item network and calculated the spatial distribution of item nodes, and used their distance in-between to replace similarity in predictions. [4] calculated the community structure [36]–[39] of item-item networks to adjust neighbor's weight in predictions. Both methods improve the accuracy of rating prediction, which means that the network modeling in recommender systems can reveal information to better understand of the systems.

Centrality metrics in network science provide methods to quantitatively measure node importance in the networks [36], [40]–[44]. One can use centrality measures to rank critical points in real-world complex systems [30], [45], or measure popularity of items by degree in recommender systems [11].

Since recommender systems can be considered as a complex networks [4], [5] with items or users as nodes and ratings or preferences as connections in-between, we can definitely design a model to construct an item-item network for further analysis of the system. With degree centrality considered as an indicator of item popularity, the relationship between node degree and prediction accuracy can be analyzed.

B. RELATED COLLABORATIVE FILTERING METHODS

He *et al.* in their work [2] proposed a method in 2015 based on similarity and top k similar neighbors, assuming opinion spread among users in recommender systems. The method provides more accurate predictions compared to some existing literature [2], [11]. This approach is considered as a memory-based CF, because the core idea is based on a similarity equation measuring the opinion spread between users. The top k neighbors are chosen for prediction instead of selection by *kNN* algorithms.

If an item-based CF is chosen, the algorithm follows a traditional pattern of memory-based CF, with an additional normalization of ratings to counteract user bias. After the normalization, a) similarity between users is calculated, b) the top k neighbors are selected for the target item i, c) for each user u not rated the target item i, calculate its prediction based on the ratings of k neighbors by user u.

Specifically, [2] used Equation 1 in [2] to normalize ratings from users.

$$e_{ui} = \frac{2 \times (r_{ui} - r_u^{min})}{r_u^{max} - r_u^{min}} - 1 \tag{1}$$

where r_u^{max} and r_u^{min} stands for maximum and minimum rating value from user u such that $e_{ui} \in [-1, 1]$. If r_u^{max} equals to r_u^{min}, e_{ui} is assumed to be zero.

Further, Equation 2 is used to capture the similarity between the items in [2].

$$s_{ij} = \frac{1}{c_i} \sum_{i=1}^{n} \frac{a_{ui} \cdot a_{uj} \cdot e_{ui} \cdot e_{uj}}{c_u} \tag{2}$$

where c_i is the degree of item i and c_u is the degree of user u in a bipartite graph consisting of users and items. a_{ui} and a_{uj} indicate whether or not user u has rated item i and item j, respectively. $a_{ui} = 1$ if user u has rated i, otherwise $a_{ui} = 0$. Therefore, the equation consider the similarity between user i and j is proportional to the sum of the products of the normalized scores of the jointly rated items. a_{ui} and a_{uj} restrict that both items must be rated by the same user. The two normalized ratings e_{ui} and e_{uj} must be sufficiently large enough and have the same signal to make a significant contribution to the similarity results.

Moreover, Equation 2 is tuned slightly to be more precise as follows [3].

$$\tilde{s}_{ij} = \frac{1}{c_i} \sum_{i=1}^{n} \frac{a_{ui} \cdot a_{uj}(1 - |e_{ui} - e_{uj}|)}{c_u} \tag{3}$$

III. PROPOSED METHOD

A. RECOMMENDER SYSTEM DESCRIPTION

Suppose there are t users and m items in a recommender system, the user set and item set can be denoted as $U = \{u_1, u_2, u_3, \ldots, u_t\}$ and $I = \{i_1, i_2, i_3, \ldots, i_m\}$, respectively.

Moreover, rating matrix $R = \{r_{ui}\}_{t\times m} \in R^{t\times m}$ is a t multiply m real matrix, where r_{ui} is the element at the $u - th$ row and $i - th$ column, and r_{ui} indicates the rating of user u on item i. The value of r_{ui} ranges from 0.5 to 5.

Correspondingly, a matrix $A = \{a_{ui}\}_{t\times m} \in R^{t\times m}$ is a t multiply m binary matrix, and a_{ui} stands for the element at the $u - th$ row and the $i - th$ column of the matrix. The value of this element can be defined as follows.

$$a_{ui} = \begin{cases} 1, & \textit{if} \text{ user } u \text{ has rated item } i \\ 0, & \textit{otherwise} \end{cases} \tag{4}$$

where $u = 1, \ldots, t$ and $i = 1, \ldots, m$.

B. MODELING ITEM-ITEM NETWORKS FROM RS

Figure 1 demonstrates how to model a network in a recommender systems. Any type of similarity equations can be used to calculate the weights between each pair of items, such as Equation 2, Equation 3,Pearson correlation and Cosine correlation. It is important to mention that the data used in the similarity calculation can be either the ratings between two targets or content-related data like the genres of movie.

The method is convenient for establishing links between items or users. However, the similarity exists between each pair of targets, and the item-item networks constructed by similarity results is a fully connected network as a result. The networks with n nodes have $\frac{n\cdot(n-1)}{2}$, $n \cdot (n - 1)$ links in undirected and directed networks, respectively. From the perspective of network science, there are too many links in the network to calculate meaningful centrality.

Thus, we need to control the number of links in networks to analyze proper network structure and centrality, by removing links with a weight less than zero or a threshold value. The structure of the topology is becoming more apparent with the smaller weight links gradually removed in Figure 1.

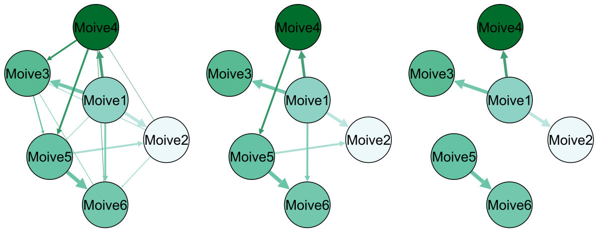

FIGURE 1. **From the left to the right, the link weights in subfigures are assumed to be greater than 0, 0.5, 0.7, respectively. The similarity can be obtained between each pair of movies. We treat the similarities as links and their weights, and construct an item-item network. Due the similarity ranges between one and zero, the weights of the links are different and varied. This figure shows that the structure of topology is becoming clearer with smaller weight links removed gradually (from zero to any links with weights less than 0.7).**

However, some nodes in the network have only small weight links, due to the factor that their style, content, or ratings are peculiar than others. Notably, when all links belonged to such kind of nodes are removed, experiments show that isolated nodes in the modeled network account for a large proportion of observed nodes.

In order to avoid this phenomenon and to keep the topology connected, two parameters q and p are introduced. q is the ratio of links to nodes which we want to keep in the network. p is the minimum number of links that each node must have in the network after link removing. Therefore, q and p can be used to determine how many links are in the network as well as maintaining the minimum number of links for each item.

The following four tasks are performed in the process of modeling item-item networks in RS. (1) determine similarity between each item pair in RS; (2) for each item i, keep the p links with the highest weight; (3) maintain the top $n \cdot q - n \cdot p$ links in the network (typically the q is larger than p); and (4) remove all other links to make the topology clear.

According to the experimental results in [5], $p \in \{1, 2, 3\}$ and $q \in [1, 20]$ are appropriate for typical recommender systems. p larger than three or q larger than twenty causes the network to become entangled with too many links, $p = 0$ causing the subgraphs to become disconnected from one another. The larger the q is, the larger the average degree of the network.

An modeled network example is shown in Figure 2, which confirms the effectiveness of our method for modeling the network from RS.

C. CONSIDER CENTRALITY AS A INDICATOR OF A NEIGHBOR's POPULARITY AND STATUS

The movie 'Toy Story' is a very large node at the bottom of Figure 2, and the high degree of the movie indicates that

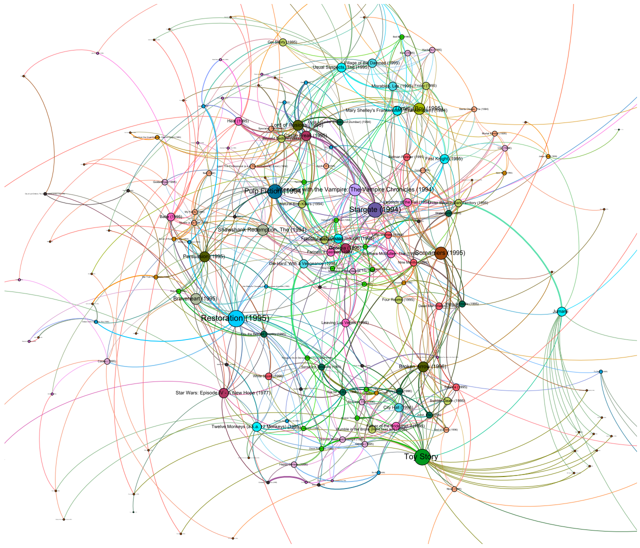

FIGURE 2. **A partial figure showing the modeled network with $q = 3$ and $p = 2$ containing the various item nodes of a real-world recommender system. Node color indicates different communities and node size indicates the degree of nodes. The link weight represents the similarity between two item nodes. The similarity function used here is Equation 3.**

'Toy Story' is rather popular and its status is important in the network. Further research is presented to explore whether such popularity and status affect the accuracy of rating predictions.

In our research, we use a simple centrality method, degree centrality, to test our assumption. The degree of items is normalized due to a possible wide range of node connection number.

For a graph G with n nodes, the normalized degree centrality for node n_i is defined as follows.

$$Degree^*(n_i) = \frac{M_{n_i}}{max_{n \in N}(Degree(n)) + 1} \tag{5}$$

where $M_{n_i} = Degree(n_i)$ is the degree of node n_i representing the number of nodes connected with n_i, and n is the number of nodes in the graph G. $max(Degree(n))$ is the maximum degree of nodes in the network. Thus, $Degree^*(n_i)$ is the normalized degree centrality that is assumed to be less than 1.0.

Therefore, we propose a function to measure the centrality of item node as follows.

$$C(j) \in \mathcal{F} = \{\frac{1}{Degree^*(j) + 1}, 1 - Degree^*(j), Degree^*(j)\} \tag{6}$$

where $Degree^*(j)$ is the normalized degree centrality value of j. $\mathcal{F}$ is a set of functions, in which there are three functions can be used for possible prediction. For simplicity in experiments, we denote $C_a(j) = \frac{1}{Degree^*(j)+1}$, $C_b(j) = 1 - Degree^*(j)$ and $C_c(j) = Degree^*(j)$.

The reason that we design function $C(j)$ is we are uncertain whether the popularity of some item will emphasize or de-emphasize the importance of an item.

D. DECENTRALIZATION OF COLLABORATIVE FILTERING METHODS

First, a prediction method is given by Equation 7 [2].

$$r'_{ui} = \overline{r_i} + \frac{\sum_{j=1}^{n}[(r_{uj} - \overline{r_j}) \cdot s_{ji}]}{\sum_{j=1}^{n} s_{ji}} \tag{7}$$

where $\overline{r_i}$ and $\overline{r_j}$ are the average item ratings in the training data set, n is the number of selected neighbors of item i. Algorithms select top k neighbors of item i with the highest similarity s_{ji} between item j and item i, which could be calculated by Equation 2 or Equation 3.

Equation 8 [4] improved the prediction equation by identifying communities of nodes in the item-item network of RS.

$$r'_{ui} = \overline{r_i} + \frac{\sum_{j=1}^{n}[(r_{uj} - \overline{r_j}) \cdot s_{ji} \cdot \varphi]}{\sum_{j=1}^{n} s_{ji}} \tag{8}$$

where φ is a function that increases or decreases the weight of similarity between item i and item j based on whether or not they are in the same community.

$$\varphi = \begin{cases} \varphi_1 \in [1, 2], & m_i = m_j, \\ \varphi_2 \in (0, 1], & m_i \neq m_j, \end{cases}$$

which is used to adjust the similarity between each pair of items. m_i and m_j donates item i's and item j's community numbers, respectively.

If the degree centrality did affect the accuracy of rating prediction, we could use an algorithm to neutralize the possible adverse impact in the prediction.

As a result, the decentralized equations can be improved as follows to achieve the goal.

$$r'_{ui} = \overline{r_i} + \frac{\sum_{j=1}^{n}[(r_{uj} - \overline{r_j}) \cdot s_{ji} \cdot C(j)]}{\sum_{j=1}^{n} s_{ji}} \tag{9}$$

or with communities considered,

$$r'_{ui} = \overline{r_i} + \frac{\sum_{j=1}^{n}[(r_{uj} - \overline{r_j}) \cdot s_{ji} \cdot \varphi \cdot C(j)]}{\sum_{j=1}^{n} s_{ji}} \tag{10}$$

where $C(j)$ is a weight function. The value of $C(j)$ is determined by the centrality of item j. We call the method as **Decentralized Collaborative Filtering (DCF)** which corresponds to the classical Collaborative Filtering (CF) algorithms.

Theoretically, the method can be applied to any Collaborative Filtering (CF) with similarity calculations. In a corresponding DCF to the original CF method, the following four tasks are executed: (1) determine similarity between each pair of items or users in RS; (2) model an item-item or user-user network based on similarities and remove unnecessary links; (3) calculate the degree of nodes in the network; and (4) predict ratings based on Equation 9 or similar equations.

IV. EXPERIMENT RESULTS

A. DATASET

To test the performance of the presented algorithm, both MovieLens and Netflix are chosen as benchmark data sets.

The MovieLens had 671 users and 9125 movies, and 10^5 ratings distributed from 0.5 to 5. The Netflix data set consisted of 124600 ratings, 1246 users and 3100 movies, which is a random sample from the original Netflix data set.

To test presented algorithm appropriately, a 10-fold cross-validation is applied to the data set, and the training set is randomly divided into 10 clusters. Each data cluster is used for testing and the remaining nine clusters are used for training. Any algorithm in the test predicts each cluster based on the information provided by the other 9 clusters, in which the algorithm picks the top k neighbors ($k \in \{1, 2, 3, 4, 5, \ldots, 19, 20, 25, 30, \ldots, 150\}$) for rating predictions. Multiple experiments are averaged for final results.

B. COMPARISON ALGORITHMS

To validate and test our methods and compare with some state-of-the-art algorithms, **CF-Pearson, CF-Cosine, CF-OS, CF-OS and CF-MIOS** are used as the abbreviations to represent several recent or standard methods using Pearson correlation collaborative filtering, Cosine correlation collaborative filtering, Opinion Spreading [2], and Modularized Improved Opinion Spreading [4], respectively. The latter two algorithms were proposed in 2015, and 2019.

Because the proposed method balanced the items or users with its own degree centrality, it is named as decentralized collaborative filtering algorithms based on complex network modeling and degree centrality, which is abbreviated as **DCF**. Combining with the target algorithm like CF-OS and CF-MIOS, it is labeled as **DCF-IOS** (based on Equation 9) and **DCF-MIOS** (based on Equation 10) in the experiments.

Other algorithms such as **Resource Allocation Collaborative Filtering (RA) [46], Multi-Level Collaborative Filtering (MLCF) [47], Item Global Profile Expansion (IGPE) [48], Slope 1 [49], Non-negative Matrix Factorization (NMF) [50] and Bayesian Probabilistic Matrix Factorization (BPMF) [51]** are also used to compare the optimal accuracy of prediction, which provide a broad picture of the algorithms' performance.

We used a machine with Intel(R) Core (TM) i9-9900K CPU @ 4.8GHz with 32GB main memory. The algorithms are implemented in FSharp language running on a Windows 10 x64 operating system. All the parameters affecting the computation time were set the same for all algorithms whenever possible.

C. PERFORMANCE EVALUATION

Two metrics are chosen to measure the prediction accuracy, namely, Normalized Mean Absolute Errors (NMAE) and Normalized Root Mean Squared Errors (NRMSE). Using NMAE and NRMSE instead of MAE and RMSE, we can comfortably compare the optimal accuracy of the proposed method with some existing work without unnecessary programming [11].

By comparing the actual ratings given by users to the predicted ratings given by algorithms, the smaller the NMAE and NRMSE are, the more accurate the algorithms are. NRMSE squares the prediction error before the summation and thus makes the large errors more obvious.

$$NMAE = \frac{\sum_{(u,i)\in T_{set}} |r_{ui} - \hat{r}_{ui}|}{|T_{set}| \cdot (R_{max} - R_{min})} \tag{11}$$

$$NRMSE = \frac{\sqrt{\sum_{(u,i)\in T_{set}} (r_{ui} - \hat{r}_{ui})^2}}{|T_{set}| \cdot (R_{max} - R_{min})} \tag{12}$$

where T_{set} is the test set, r'_{ui} is the predicted rating and $\hat{r}_{ui}$ is the real rating in the test data set. R_{max} and R_{min} are the maximum rating and the minimum rating in the system, respectively.

D. PERFORMANCE OF DIFFERENT DECENTRALIZED FUNCTIONS

The prediction results shown in Fig.3 and Fig.4 are all item-based, and the results of NMAE and NRMSE show the error level of the target methods on the MovieLens dataset. DCF-MIOS with $C_a(j)$ and DCF-MIOS with $C_b(j)$ are far superior than the original MIOS, suggesting popularity of the items is overstated in their importance. As such, their status should be de-emphasized in the rating prediction to improve accuracy.

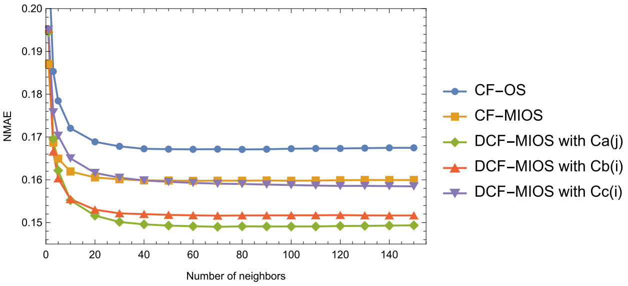

FIGURE 3. The NMAE results indicate CF-OS, CF-MIOS, and the proposed DCF-MIOS approaches with $C_a(i)$, $C_b(i)$ and $C_c(i)$, respectively. The proposed methods improve prediction accuracy compared to the original CF-MIOS. It is also shown that $C_a(i)$ has the best performance among all decentralization functions. All algorithms are item-based.

Approaches with $C_a(i)$, $C_b(i)$ are preferable fixes for decentralizing node popularity in RS. According to our experimental results, $C_a(i)$ has the best performance among the three functions. Therefore, we use $C_a(i)$ in the following sections without further indication.

E. DECENTRALIZED COLLABORATIVE FILTERING IMPROVES THE ACCURACY OF THE CORRESPONDING METHODS

In Fig. 5 and Fig. 6, we introduce more algorithms into testing. Results from the NMAE and NRMSE indicate the accuracy of the methodology in the MovieLens data set. All

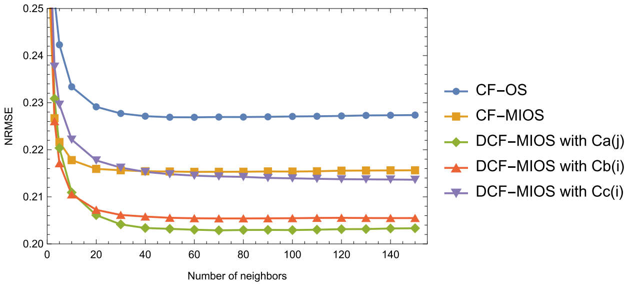

FIGURE 4. The NRMSE results indicate the CF-OS, CF-MIOS, and the proposed DCF-MIOS approaches with $C_a(i)$, $C_b(i)$ and $C_c(i)$, respectively. The proposed methods improve prediction accuracy compared to the original CF-MIOS. It is also shown the $C_a(i)$ has the best performance among all decentralization functions. All algorithms are **item-based**.

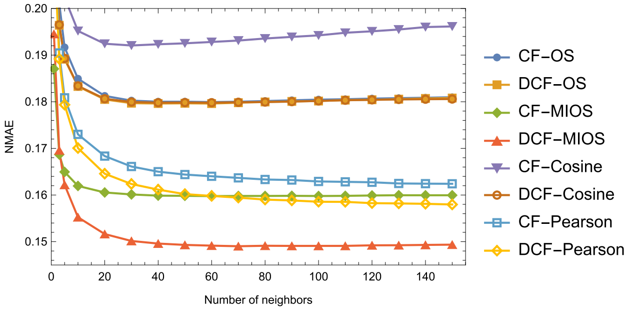

FIGURE 5. The NMAE results of CF-Cosine, CF-Pearson, CF-OS, CF-MIOS and their counterparts DCF-Cosine, DCF-Pearson, DCF-OS, DCF-MIOS. The accuracy of rating predictions is improved in all four methods by introducing network modeling and node decentralization.

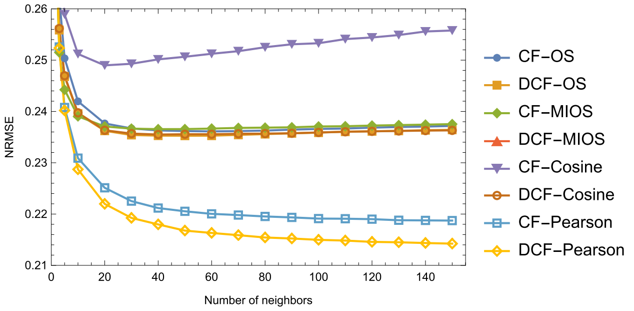

FIGURE 6. The NRMSE results of CF-Cosine, CF-Pearson, CF-OS, CF-MIOS and their counterparts DCF-Cosine, DCF-Pearson, DCF-OS, DCF-MIOS. The accuracy of rating predictions is improved in all four methods by introducing network modeling and node decentralization.

four methods are improved by using network modeling and node decentralization.

In this section, the parameters $p = 1$ and $q = 1.5$ are used. They are determined by a group of experiments. For the sake of clarity, the experiments are omitted in this article.

This experiment confirms that our method can be used in different memory-based CF methods to improve the prediction accuracy.

F. RESULTS IN ANOTHER DATASET

Similarly, Fig.7 illustrates the NMAE results in Netflix dataset. The improvements of DCF-Cosine are very obvious, but the improvements of DCF-IOS are relatively small. We concluded after analysis that this was due to the lack of a clear community structure in the sample. Nevertheless, the optimal accuracy of DCF-IOS is still a little better than that of CF-MIOS.

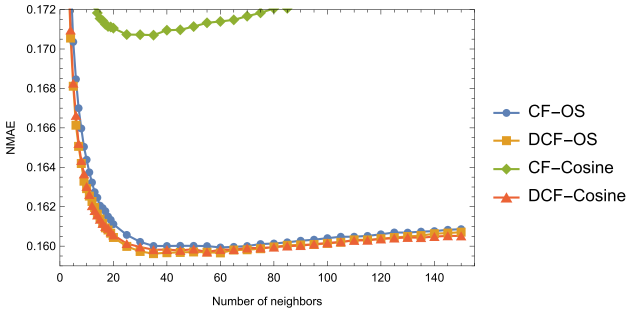

FIGURE 7. NMAE results from the CF-Cosine, IOS and the proposed decentralized approaches of DCF-Cosine, DCF-IOS. Netflix data sets are used for the experiments. All algorithms are **user-based**. These experiments confirm that the proposed method works nicely on different data sets.

G. THE OPTIMAL ACCURACY COMPARISON

The most important results from the experiments are the optimal ones with the smallest errors in rating prediction. Thus, Table 1 shows the result comparison of the optimal accuracy with the top k neighbors among most of the aforementioned algorithms (k may vary by different methods).

TABLE 1. The optimal prediction accuracy (NMAE and NRMSE) of the proposed algorithms **DCF-OS, DCF-MIOS** and other **memory-based** comparison methods in MovieLens data set. The best result is shown in boldface.

	DCF-OS	**DCF-MIOS**	RA	MLCF	IGPE
NMAE	0.1796	**0.1490**	0.1716	0.1812	0.1889
NRMSE	0.2353	**0.2029**	0.2295	0.2383	0.2466

TABLE 2. The optimal prediction accuracy (NMAE and NRMSE) of the proposed algorithms **DCF-OS, DCF-MIOS** and other **model-based** comparison methods in MovieLens data set. The best result is shown in boldface.

	DCF-OS	**DCF-MIOS**	Slope 1	NMF	BPMF
NMAE	0.1796	**0.1490**	0.1697	0.1588	0.1531
NRMSE	0.2353	0.2029	0.2278	0.1992	**0.1942**

Table 1 and Table 2 confirm the effectiveness of the proposed methods that can be used to improve the various memory-based CF approaches. DCF-MIOS has the best NMAE, and NRMSE has improved 14% compared to the original CF-MIOS, which is only 4% percent worse than BPMF in model-based category. The number of neighbors used in the experiments in the table is different.

V. CONCLUSION

Traditional CF methods only consider them based on similarities and cause accuracy losses. Our improvement in prediction accuracy is based on the fact that items in the recommender system differ in popularity, which can be revealed by their centrality in network models.

Furthermore, the additional time complexity of our methods mainly comes from removing irrelevant links and calculating the degree of nodes in the modeled network. The time complexity of sorting link importance and removing links is $O(n \cdot log(n))$, and the time complexity of calculating node degree is $O(n)$ in the worst case scenario, which is acceptable even in large systems.

To compare various nodes differently according to their centrality in the networks, we designed a model to predict user ratings for items, which is capable of improving the predictive accuracy of various memory-based CF methods. The algorithm uses the information from the ratings to calculate the similarity of the targets, and constructs a network based on the similarity results. As a result, centrality metrics can be used in the prediction process, which intriguingly increases the prediction accuracy for every algorithm tested in our experiments.

In particular, the research demonstrates that the inequality of items is nontrivial in recommender systems. Popular item nodes are overrated in prediction without considering their centrality.

Theoretically, our model can be applied to many recommendation algorithms. With similarity results as link weights and the modeling of complex networks, centrality metrics can be taken into consideration to provide better prediction accuracy.

ACKNOWLEDGMENTS

(Jun Ai and Zhan Su contributed equally to this work.) Zhan Su and Jun Ai would like to express love and gratitude to Lingyi Ai, daughter of Zhan Su, for giving us courage to carry on and fight.

REFERENCES

[1] E. Bütün, M. Kaya, and R. Alhajj, "Extension of neighbor-based link prediction methods for directed, weighted and temporal social networks," *Inf. Sci.*, vols. 463–464, pp. 152–165, Oct. 2018.

[2] X.-S. He, M.-Y. Zhou, Z. Zhuo, Z.-Q. Fu, and J.-G. Liu, "Predicting online ratings based on the opinion spreading process," *Phys. A, Stat. Mech. Appl.*, vol. 436, pp. 658–664, Oct. 2015.

[3] J. Ai, L. Li, Z. Su, and C. Wu, "Online-rating prediction based on an improved opinion spreading approach," in *Proc. 29th Chin. Control Decis. Conf. (CCDC)*, May 2017, pp. 1457–1460.

[4] J. Ai, Y. Liu, Z. Su, H. Zhang, and F. Zhao, "Link prediction in recommender systems based on multi-factor network modeling and community detection," *EPL (Europhys. Lett.)*, vol. 126, no. 3, p. 38003, Jun. 2019.

[5] J. Ai, Z. Su, Y. Li, and C. Wu, "Link prediction based on a spatial distribution model with fuzzy link importance," *Phys. A, Stat. Mech. Appl.*, vol. 527, Aug. 2019, Art. no. 121155. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378437119306995

[6] C. Wu, J. Wu, C. Luo, Q. Wu, C. Liu, Y. Wu, and F. Yang, "Recommendation algorithm based on user score probability and project type," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, p. 80, Dec. 2019.

[7] T. Wang, X.-S. He, M.-Y. Zhou, and Z.-Q. Fu, "Link prediction in evolving networks based on popularity of nodes," *Sci. Rep.*, vol. 7, no. 1, p. 7147, Dec. 2017.

[8] W. Yu, W. Cheng, C. C. Aggarwal, H. Chen, and W. Wang, "Link prediction with spatial and temporal consistency in dynamic networks," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 3343–3349.

[9] L. Huang, W. Tan, and Y. Sun, "Collaborative recommendation algorithm based on probabilistic matrix factorization in probabilistic latent semantic analysis," *Multimedia Tools Appl.*, vol. 78, pp. 8711–8722, Jun. 2018.

[10] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowl.-Based Syst.*, vol. 46, pp. 109–132, Jul. 2013.

[11] M. Jalili, S. Ahmadian, M. Izadi, P. Moradi, and M. Salehi, "Evaluating collaborative filtering recommender algorithms: A survey," *IEEE Access*, vol. 6, pp. 74003–74024, 2018.

[12] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Phys. A, Stat. Mech. Appl.*, vol. 390, no. 6, pp. 1150–1170, Mar. 2011.

[13] V. Martínez, F. Berzal, and J.-C. Cubero, "A survey of link prediction in complex networks," *ACM Comput. Surv. (CSUR)*, vol. 49, no. 4, p. 69, 2016.

[14] L. Candillier, F. Meyer, and M. Boullé, "Comparing state-of-the-art collaborative filtering systems," in *Proc. Int. Workshop Mach. Learn. Data Mining Pattern Recognit.* Berlin, Germany: Springer, 2007, pp. 548–562.

[15] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.

[16] J. B. Schafer *et al.*, "Collaborative filtering recommender systems," in *The Adaptive Web*. Berlin, Germany: Springer, 2007, pp. 291–324.

[17] R. Van Meteren and M. Van Someren, "Using content-based filtering for recommendation," in *Proc. Mach. Learn. New Inf. Age, MLnet/ECML Workshop*, 2000, pp. 47–56.

[18] D. Wang, Y. Liang, D. Xu, X. Feng, and R. Guan, "A content-based recommender system for computer science publications," *Knowl.-Based Syst.*, vol. 157, pp. 1–9, Oct. 2018.

[19] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Commun. ACM*, vol. 35, no. 12, pp. 61–70, Dec. 1992, doi: 10.1145/138859.138867.

[20] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Model. User-Adapted Interact.*, vol. 12, no. 4, pp. 331–370, 2002, doi: 10.1023/A:1021240730564.

[21] R. Xiong, J. Wang, N. Zhang, and Y. Ma, "Deep hybrid collaborative filtering for Web service recommendation," *Expert Syst. Appl.*, vol. 110, pp. 191–205, Nov. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417418303385

[22] D. Billsus and M. J. Pazzani, "User modeling for adaptive news access," *User Model. User-Adapted Interact.*, vol. 10, nos. 2–3, pp. 147–180, 2000, doi: 10.1023/A:1026501525781.

[23] Z. Su, X. Zheng, J. Ai, L. Shang, and Y. Shen, "Link prediction in recommender systems with confidence measures," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 29, no. 8, Aug. 2019, Art. no. 083133, doi: 10.1063/1.5099565.

[24] M. Balabanović and Y. Shoham, "Fab: Content-based, collaborative recommendation," *Commun. ACM*, vol. 40, no. 3, pp. 66–72, Mar. 1997.

[25] M. M. Azadjalal, P. Moradi, A. Abdollahpouri, and M. Jalili, "A trust-aware recommendation method based on Pareto dominance and confidence concepts," *Knowl.-Based Syst.*, vol. 116, pp. 130–143, Jan. 2017.

[26] C. Wu, J. Wang, J. Liu, and W. Liu, "Recurrent neural network based recommendation for time heterogeneous feedback," *Knowl.-Based Syst.*, vol. 109, pp. 90–103, Oct. 2016.

[27] M. Ranjbar, P. Moradi, M. Azami, and M. Jalili, "An imputation-based matrix factorization method for improving accuracy of collaborative filtering systems," *Eng. Appl. Artif. Intell.*, vol. 46, pp. 58–66, Nov. 2015.

[28] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, p. 440, 1998.

[29] H. Li, X. Yang, Y. Li, L.-Y. Hao, and T.-L. Zhang, "Evolutionary extreme learning machine with sparse cost matrix for imbalanced learning," *ISA Trans.*, vol. 100, pp. 198–209, May 2020.

[30] S. Guo, R. Chen, H. Li, T. Zhang, and Y. Liu, "Identify severity bug report with distribution imbalance by CR-SMOTE and ELM," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 29, no. 2, pp. 139–175, Feb. 2019.

[31] H. Li, Y. Qu, S. Guo, G. Gao, R. Chen, and G. Chen, "Surprise bug report prediction utilizing optimized integration with imbalanced learning strategy," *Complexity*, vol. 2020, pp. 1–14, Feb. 2020.

[32] P.-P. Zhang, K. Chen, Y. He, T. Zhou, B.-B. Su, Y. Jin, H. Chang, Y.-P. Zhou, L.-C. Sun, B.-H. Wang, and D.-R. He, "Model and empirical study on some collaboration networks," *Phys. A, Stat. Mech. Appl.*, vol. 360, no. 2, pp. 599–616, Feb. 2006.

[33] G. Bagler, "Analysis of the airport network of India as a complex weighted network," *Phys. A, Stat. Mech. Appl.*, vol. 387, no. 12, pp. 2972–2980, May 2008.

[34] J. Ai, H. Zhao, Z. Su, H. Li, and K. M. Carley, "Evolution of IPv6 Internet topology with unusual sudden changes," *Chin. Phys. B*, vol. 22, no. 7, Jul. 2013, Art. no. 078902.

[35] C.-H. Lai, S.-J. Lee, and H.-L. Huang, "A social recommendation method based on the integration of social relationship and product popularity," *Int. J. Hum.-Comput. Stud.*, vol. 121, pp. 42–57, Jan. 2019.

[36] S. Fortunato, V. Latora, and M. Marchiori, "Method to find community structures based on information centrality," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 70, no. 5, Nov. 2004, Art. no. 056104.

[37] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 2, Feb. 2004, Art. no. 026113.

[38] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, Jun. 2005.

[39] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol. 466, no. 7307, pp. 761–764, 2010.

[40] C. Correa, T. Crnovrsanin, and K.-L. Ma, "Visual reasoning about social networks using centrality sensitivity," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 1, pp. 106–120, Jan. 2012.

[41] H. Kim and R. Anderson, "Temporal node centrality in complex networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 85, no. 2, Feb. 2012, Art. no. 026107.

[42] K.-I. Goh, E. Oh, B. Kahng, and D. Kim, "Betweenness centrality correlation in social networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 67, no. 1, Jan. 2003, Art. no. 017101.

[43] J. Ai, H. Zhao, K. M. Carley, Z. Su, and H. Li, "Neighbor vector centrality of complex networks based on neighbors degree distribution," *Eur. Phys. J. B*, vol. 86, no. 4, p. 163, Apr. 2013.

[44] K.-L. Wang, C.-X. Wu, J. Ai, and Z. Su, "Complex network centrality method based on multi-order K-shell vector," *Acta Phys. Sinica*, vol. 68, no. 19, 2019, Art. no. 196402.

[45] H. Li, G. Gao, R. Chen, X. Ge, S. Guo, and L.-Y. Hao, "The influence ranking for testers in bug tracking systems," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 29, no. 1, pp. 93–113, Jan. 2019.

[46] A. Javari, J. Gharibshah, and M. Jalili, "Recommender systems based on collaborative filtering and resource allocation," *Social Netw. Anal. Mining*, vol. 4, no. 1, p. 234, Dec. 2014.

[47] N. Polatidis and C. K. Georgiadis, "A multi-level collaborative filtering method that improves recommendations," *Expert Syst. Appl.*, vol. 48, pp. 100–110, Apr. 2016.

[48] V. Formoso, D. Fernández, F. Cacheda, and V. Carneiro, "Using profile expansion techniques to alleviate the new user problem," *Inf. Process. Manage.*, vol. 49, no. 3, pp. 659–672, May 2013.

[49] D. Lemire and A. Maclachlan, "Slope one predictors for online rating-based collaborative filtering," in *Proc. SIAM Int. Conf. Data Mining*. Philadelphia, PA, USA: SIAM, Apr. 2005, pp. 471–475.

[50] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 556–562.

[51] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 880–887.

JUN AI was born in Raohe County, Heilongjiang, China, in 1980. He received the bachelor's degree in automation from the Shenyang University of Technology, in 2004, and the master's degree in computer application technology and the Ph.D. degree in computer science from Northeastern University, in 2008 and 2013, respectively. After graduation, he has been a Lecturer with the University of Shanghai for Science and Technology, since 2013. His research interests include complex network theory and recommender systems.

ZHAN SU was born in Tieling, Liaoning, China, in 1983. She received the bachelor's degree in information and computing science and the Ph.D. degree in control theory and engineering from Northeastern University, Shenyang, China, in 2005 and 2012, respectively. Her research interests include intelligent control, complex network theory, and recommender systems. She has been a Lecturer with the University of Shanghai for Science and Technology, since 2013.

KAILI WANG was born in Weifang, Shandong, China, in 1993. She received the master's degree in computer science and technology from the University of Shanghai for Science and Technology, in 2020. She was engaged in the complex network theory and recommender systems under the supervision of Wu Chunxue and Jun Ai.

CHUNXUE WU received the Ph.D. degree in control theory and control engineering from the China University of Mining and Technology, Beijing, China, in 2006. He is currently a Professor with the Computer Science and Engineering and Software Engineering Division, School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, China. His research interests include wireless sensor networks, distributed and embedded systems, wireless and mobile systems, and networked control systems.

DUNLU PENG received the Ph.D. degree from Fudan University, in 2006. He is currently a Professor with the University of Shanghai for Science. He has published more than 100 conference or journal article in computer science. His research interests include data analytics, natural language processing, machine learning, recommendation systems, and information fusion.

• • •