Node Similarity Index and Community Identification in Bipartite Networks

Dongqi Wang, Mingshuo Nie, Dongming Chen, Li Wan, Xinyu Huang

Software College, Northeastern University, China
wangdq@swc.neu.edu.cn, neunms@163.com, chendm@mail.neu.edu.cn, 1446445456@qq.com, neuhxy@163.com

Abstract

Bipartite networks or affiliation networks are a particular class of complex networks. It comprises two types of nodes, and only edges between the nodes of different types are allowed. The bipartite network model is a natural representation of the relationships between diverse entities. Most of the traditional complex network research focuses primarily on a single network, so research on bipartite networks is particularly necessary. In this paper, a novel DA similarity is proposed to measure the similarity between nodes, which takes both the influence of nodes and neighborhood structure information of nodes into consideration. Based on the DA similarity index, a community detection algorithm for bipartite networks (CDBNS), is firstly proposed. The experimental results show that DA similarity is superior to traditional similarity indices, and the CDBNS algorithm has an excellent performance in modularity and time-consuming. Furthermore, we employ the CDBNS algorithm in recommendation tasks and propose a recommendation algorithm called RASCS, which calculates the node similarity of each community detected by CDBNS and incorporates user-based collaborative filtering to achieve recommendation. It is also verified by experiments on several real-world datasets that the RASCS algorithm outperforms some baselines, such as RACD, ItemBasedCF, and UserBasedCF algorithms.

Keywords: Bipartite networks, Community detection, Similarity, Recommendation

1 Introduction

Complex networks have become a new and promising research hotspot, which includes a very wide range, many networks daily belong to complex networks, such as interpersonal relationship networks [1]. The theory of complex networks is widely used in the real world. Liu et al. [2] researched the feasibility of career path prediction from social network data. Preoțiuc-Pietro et al. [3] employed a novel dataset with political ideology labels to predict the political ideology of invisible users for determining the differences in both political leaning and engagement. In researches on traditional community detection algorithms, Wang et al. formulated a generalized procedure-oriented framework to evaluate the proposed algorithms for discussing their merits and faults [4]. Azaouzi et al. provided a taxonomy of existing models based on computational nature (either centralized or distributed), and some practical applications for social networks [5]. These researches provide extremely important references and data for scholars.

In order to study the commonness of complex networks and solve the problems in the process of research, scholars have proposed a variety of theoretical methods. At present, many literatures had discovered the unique attributes hidden in complex networks. Among the numerous discoveries, the discoveries of two achievements had a significant and far-reaching impact, which laid a solid theoretical foundation for the follow-up study. The first is the description of the characteristics of the small-world in the reference [6] in 1998, which was put forward by Professor Strogatz. The other is the related introduction of scale-free property in reference [7] in 1999, which was put forward by Professor Barabasi.

With the deepening of research on complex networks, the "community" characteristics of complex networks have also been found, and this characteristic exists in many types of networks. In a social network, for example, acquaintances can form a small group of people who can be considered to exist in a community [8]. People who exist in the same community are more closely related to each other and are relatively alienated from people outside the community. It is of great significance to study community structure, for example, the analysis of community structure on the Internet is helpful in hot spot tracking, information dissemination [9], and recommendation. Zhang et al. [10] proposed an improved music recommendation method based on bipartite graph link prediction with homogeneous nodes similarity. Therefore, the research of the structure of the community and the analysis of the relationship within the community can help scholars understand networks more effectively and accurately.

The bipartite network is an important category of

*Corresponding Author: Dongming Chen; E-mail: chendm@mail.neu.edu.cn
DOI: 10.3966/160792642021052203016

complex networks in real-world systems, where the nodes are divided into two types such that no two nodes of the same type are adjacent [11]. At present, there are two types of community detection algorithms for bipartite networks, one is the mapping method, the other is the direct method. The mapping method regards the bipartite network as two single networks with a type of nodes and uses the method for a single network to conduct research. For example, Melamed, Roger, and others have proposed dual-projection [12] and single-projection methods [13], as well as weighted and unweighted projection. However, this method has some disadvantages, for instance, some information will be lost during the projection process. The direct method is to conduct research directly based on the bipartite networks, that is, an index about community detection can be proposed in advance, and then the index is employed for detecting community. There are also some specific algorithms following this idea, such as the ant colony optimization algorithm [14], optimization algorithm for modular function [15], BSSCD algorithm proposed by Yan et al. in 2016 [16], etc. However, these methods are also characterized by instability and high complexity. Therefore, we propose a similarity index, which can consider the degree of the two types of nodes as well as the common neighbor nodes in the bipartite network and can make up for the disadvantages of traditional similarity indices, which is of great significance. Besides, though the collaborative filtering algorithm has been very successful and widely used, there are still some non-trivial disadvantages that should be solved, such as efficiency and sparseness, etc. Based on our newly proposed similarity index, we also propose a novel algorithm for community detection, and simultaneously, a new recommendation algorithm is designed to overcome these problems.

2 Node Similarity

2.1 Basic Terminologies

A graph is composed of a set of nodes and a set of edges between nodes, usually expressed as $G(V, E)$, where G represents a graph, V is a set of nodes in graph G, and E is a set of edges in graph G. The degree of node V_i refers to the number of edges associated with V_i in the graph. For a directed graph, there are in-degree and out-degree, and the degree of the node in the directed graph is equal to the sum of the in-degree and out-degree of the nodes. For an undirected graph, the degree of the node in the undirected graph is the total number of edges associated with the nodes. In a bipartite network, nodes can be divided into two disjoint sets, and all edges are established between nodes from different sets.

2.2 Node Similarity

Thanks to the progress of complex networks, the researches in the field of network structure are more detailed and in-depth. There are many methods to measure the similarity between nodes so far, such as Jaccard index [17], Sorensen index [18], Salton index [19], Common Neighbors index (CN) [20], Resource Allocation index (RA) [21] and Adamic-Adar index (AA) [22], etc. These calculation formulas are shown in Table 1.

Table 1. Several common similarity indices. where a and b represent two nodes, the respective neighbor node sets are represented as $S(a)$ and $S(b)$, the degree of node a is represented as $d(a)$ and the degree of node b is represented as $d(b)$

Similarity index	Equation
Jaccard	$S_{(a,b)} = \frac{\lvert S(a) \cap S(b) \rvert}{\lvert S(a) \cup S(b) \rvert}$
Sorenson	$S_{(a,b)} = \frac{2\lvert S(a) \cap S(b) \rvert}{d(a)+d(b)}$
Salton	$S_{(a,b)} = \frac{2\lvert S(a) \cap S(b) \rvert}{\sqrt{d(a)+d(b)}}$
CN	$S_{(a,b)} = \lvert S(a) \cap S(b) \rvert$
RA	$S_{(a,b)} = \sum_{c \in S(a) \cap S(b)} \frac{1}{d(c)}$
AA	$S_{(a,b)} = \sum_{c \in S(a) \cap S(b)} \frac{1}{\log d(c)}$

The Jaccard index is one of the early indices of similarity, and its calculation result can be expressed as the ratio of the union of the common neighbors of the two nodes to the respective neighbor nodes of the two nodes. The Sorenson index represents the ratio of common neighbors of two nodes to the degree sum of the two nodes. In the Salton index, we actually employ a vector method to calculate similarity. Specifically, we regard neighbor nodes of a node in the network as vectors. The CN index is based on an intuitive assumption that the more the number of common neighbors of two nodes, the greater their similarity. The RA index is proposed from the perspective of network resource allocation, which is employed to consider the similarity between nodes in the network. The AA index takes into account the influence of the degree of the intersection of two adjacent nodes, and its idea is that the contribution of nodes with a small degree is greater than that of nodes with a large degree. Therefore, each node is given a weight according to the degree of neighbor nodes, and the weight is $\frac{1}{\log d(c)}$.

The weight value is determined by calculating the information entropy. Specifically, if the information entropy of a certain index is smaller, it indicates that the more information provided, the greater the role it can play in the comprehensive evaluation, and the greater its weight.

Jaccard index measures the similarity of two node sets by the proportion of different nodes in all the nodes. The Sorenson index and Salton index only consider the degree of the two target nodes and the number of their common neighbors, in which, Salton index uses the method based on vectors to calculate similarity, which is called similarity features of local information based on common neighbor nodes. CN index, AA index, and RA index only consider the influence of the degree of the common neighbors of the two target nodes, but not the influence of the degree of the two target nodes, in which, AA index is widely employed in similarity calculation and has been recognized by many scholars. These indices for calculating node similarity of single networks only consider the influence of a type of nodes on themselves and are easy to ignore the similarity between connected nodes. Therefore, they are suitable for single networks or link prediction, but not for bipartite networks or community detection.

In this paper, a similarity index (DA similarity index) for bipartite networks is proposed, which considers the degree of two types of nodes in bipartite networks and the degree of their common neighbor nodes. The index takes into account the characteristics of the AA index and the Salton index, and it makes up for the disadvantage of insufficient consideration of the role of the AA index to the common neighbor nodes. The specific formula of DA is defined as follows:

$$S_{(a,b)} = \frac{\sum_{c \in S(a) \cap S(b)} \frac{1}{\log d(c)}}{\sqrt{d(a) \times d(b)}}, \tag{1}$$

where a and b represent two target nodes, their common neighbor node is represented by c, the neighbor node set of node a is recorded as $S(a)$, the neighbor node set of node b is recorded as $S(b)$, $d(a)$ and $d(b)$ respectively represent the degree of node a and node b, and $d(c)$ represents the degree of node c.

3 CDBNS Algorithm

3.1 Proposal for Algorithm

There are two types of nodes in a bipartite network, and the two nodes of the edges must belong to different types. If we calculate the similarity between the same type of nodes, we can consider the information of different types of nodes from the perspective of structural similarity, that is, the common neighbor nodes, to get the similarity between the same type of nodes. If node a is the target node and node b is the most similar node to node a, node a and node b should belong to the same community. According to this, the community detection of the whole network can be realized by dividing the nodes with high similarity into a node set.

3.2 Definitions

When computing the similarity, there is often more than one node in the network, such as the similarity between the target node a and the nodes b and c is the same after computing, and the nodes b and c are also the most similar nodes of the target node a. Therefore, node a and the nodes b and c should belong to the same community, then forcibly dividing the three nodes into a node set will lead to inaccurate results if nodes b and c belong to different communities. Currently, it is necessary to calculate the membership degree of node a, node b, and node c in their respective communities. Besides, when dealing with isolated edges in the network, if the isolated nodes and edges are discarded and only connectivity subgraphs are considered, the network information will be incomplete. The two nodes contained in the isolated edge do not connect with any other nodes, so it is impossible to calculate the similarity between the two nodes and other nodes. In this algorithm, the two nodes contained in the isolated edge are regarded as a community. Given the above problems, the following definitions are made:

Definition 1: The membership degree of node a to community C is the sum of similarity degree of node a and all nodes in community C. The formula is as follows:

$$S_{a,C} = \sum_{b \in C} S_{(a,b)} = \frac{\sum_{c \in S(a) \cap S(b)} \frac{1}{\log d(c)}}{\sqrt{d(a) \times d(b)}}, \tag{2}$$

where a and b represent two nodes, C represents a community, and node b belongs to community C. Membership degree is a calculation method designed for the community to which the nodes with the most similar nodes belong. For example, in Figure 1, it is assumed that after similarity calculation, there are two most similar nodes of node C, that is, B and D. However, since B and D belong to different communities C_1 and C_2, it is necessary to calculate the membership degree of node C to communities C_1 and C_2.

The algorithm also involves the merging between communities, which is mainly to solve the merging problem between the small community (i.e. a community with two nodes) and other relatively larger communities after computing the similarity between the nodes. Therefore, it is necessary to measure the similarity between communities.

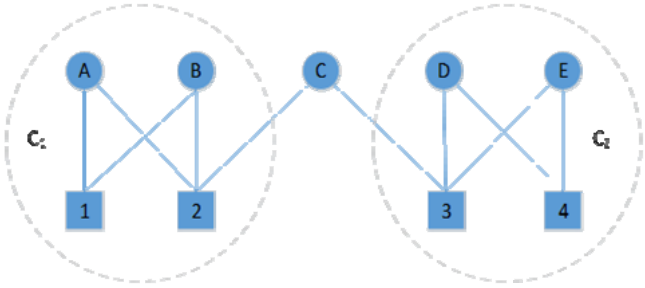


Figure 1. The special nodes and communities

Definition 2: The similarity between community C_1 and community C_2 is the sum of the similarities of all nodes in the two communities. The formula is as follows:

$$S_{C_1,C_2} = \sum_{a\in C_1}\sum_{b\in C_2} S_{(a,b)} = \sum_{a\in C_1}\sum_{b\in C_2} \frac{\sum_{c\in S(a)\cap S(b)} \frac{1}{\log d(c)}}{\sqrt{d(a)\times d(b)}}, \quad (3)$$

where a and b represent two nodes, and C_1 and C_2 represent two communities. Node a belongs to community C_1 and node b belongs to community C_2.

3.3 Algorithm Description

The algorithm does not need to preset parameters (such as the number of communities), nor does it need to extract maximal connectivity subgraphs from the network. It can directly start from any node in the network and can get the community detection results of the whole bipartite network.

The steps of the CDBNS (Community Detection Algorithm for Bipartite Networks Based on Node Similarity) algorithm are as follows:

Algorithm. CDBNS

Input: Bipartite Networks.

Output: Community detection Results.

Step 1: Obtain the neighbor nodes of the neighbor nodes of the target node.

1-1 Select any node *n* from the network as the initial node, get the neighbor nodes of node *n*, add these to the list *neibor_node1*, and create the list *Same type*. The selection of initial nodes is completely random;

1-2 Select any node *m* from that list *neibor_node1*, obtain one of its neighbor nodes, store it in the list *neibor_node2*, and perform step 1-3. Node *n* is the initial node that has been determined, so it does not need to be added to *neibor_node2*;

1-3 Move all nodes from *neibor_node2* to the list *Same type* and perform step 1-4;

1-4 Repeat 1-2 to 1-4 to make sure all the original nodes in the list *neibor_node1* have been traversed, and perform step 2.

Step 2: Calculate the similarity between the initial node and the node obtained by the above steps.

2-1 If the number of the same type neighbor nodes of node *n* (i.e., *Same type*) is 0, put the node and its neighbor nodes into the list *cluster_list* and regard them as a community, then perform step 1. If not, calculate the similarity of node *n* to each node in the list *Same type*, and perform step 2-2;

2-2 According to the similarity, we save the node with more than one most similar neighbor node to list *special_node_tag*. The nodes stored in this list are equivalent to being labeled and return to step 1. Otherwise, the neighbor nodes in the same type with the highest similarity need to be selected to form a list with node *n*, and perform step 2-3;

2-3 If all nodes of the same type as node *n* in the network have been traversed, then several lists are generated, each list represents a community, and step 3 is performed. Otherwise, choose any node from the same type as the initial node and repeat step 1.

Step 3: Merge communities of the same type.

3-1 Merge several lists obtained in step 2 to form a new community, according to the transitivity of similarity between nodes and perform step 3-2;

3-2 Calculate the membership degree between the nodes and the communities according to Equation (2) for the nodes in the list *special_node_tag*. Put the node into the community with the highest degree of membership, and perform step 3-3;

3-3 Determine whether all nodes in the list *special_node_tag* have been traversed. If so, the final partition result *final_cluster_A* of the node is obtained, and perform step 4. If not, perform 3-4;

3-4 Repeat step 1 to step 3-3 for another type of nodes to obtain the partition result *final_cluster_B* of the other type of nodes.

Step 4: Merge communities of different types. The communities in *final_cluster_A* and *final_cluster_B* are merged according to the number of edges. After merging communities, the final partition result of bipartite network communities is obtained.

When the nodes of the bipartite network are not totally traversed, we need to randomly select one of the other types of node as input according to step 1-1, and repeat step 1-2 to step 3-3 to obtain the partition result of the other type of node, that is, *final_cluster_B*.

Figure 2 is a flow of the execution process of the CDBNS algorithm. It shows that the CDBNS algorithm does not need to set any parameters in advance, nor does it need to process the network to obtain the maximum connective subgraph in advance. It randomly selects an initial node from the network, and obtains the neighbor node sets of the node firstly, and then obtains the neighbor nodes of the same type of the initial node through the neighbors of the neighbor nodes. According to the number of the same type of neighbor nodes of the initial node, it is decided whether the initial node and its neighbor nodes are divided into an isolated community. For those nodes with the same type of neighbor nodes, the similarity between the initial node and its neighbor nodes should

be calculated according to the similarity definition, and the neighbor nodes with the highest similarity degree should be selected to form a community with the initial node. If the initial node has more than one neighbor nodes with the highest similarity in a type, the initial node is stored in a special node set. When all nodes have obtained the most similar neighbor nodes of the same type, the merging of communities of the same type is carried out.

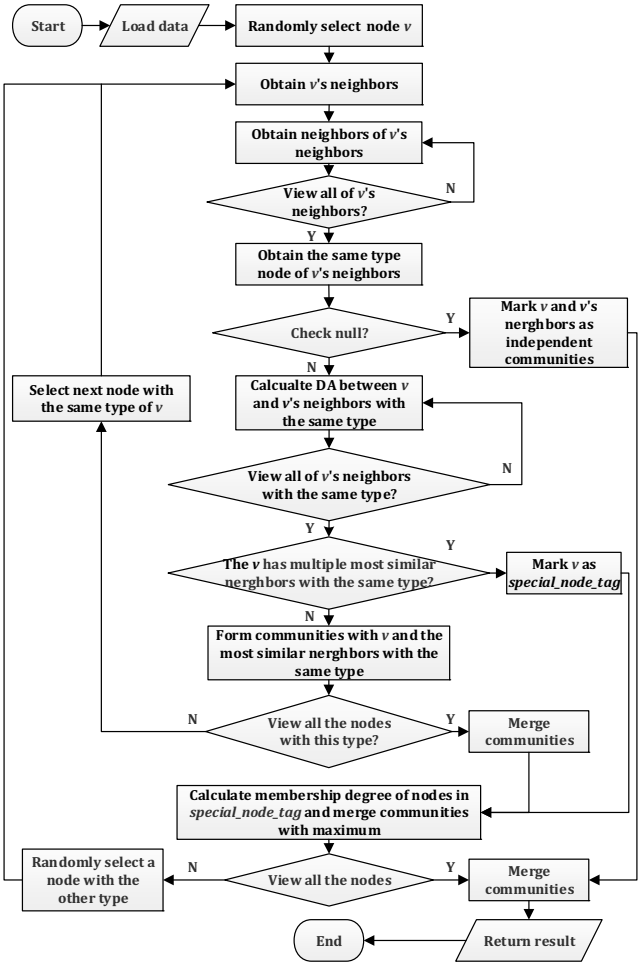


Figure 2. The specific flow chart of CDBNS algorithm

When merging communities of the same type, we merge the primary communities first and then calculate the membership degree between the nodes in the special node set and the merged communities according to the definition of membership degree. Select the community with the highest membership to add nodes from a special node set to the community. For another type of node in the network, the same steps are performed to obtain the merging results of another type of community. Finally, the merging results of the two types of communities are combined according to the principle of maximizing the number of edges, and the isolated communities are added to the merged communities to obtain the community results of the whole bipartite network.

3.4 Algorithm Complexity

In the algorithm proposed in this paper, the main execution steps include searching the most similar nodes for the target nodes, merging the primary communities, calculating the membership of the marked nodes and communities, and merging heterogeneous communities, etc. Assuming that the number of nodes in a bipartite network is n and the average degree of nodes is m. The time complexity of searching the most similar nodes for all target nodes is $O(mn)$. Assuming that all nodes in the network are marked, the time complexity of calculating the membership of the marked nodes and the communities is $O(n)$. The time complexity of merging communities is $O(1)$. Therefore, the time complexity of CDBNS algorithm is $O(mn)$, which is close to the time complexity $O(n)$ of the most efficient Label Propagation Algorithm (LPA) [23] and is better than the time complexity $O(nlog(n))$ of Louvain algorithm [24].

The space complexity of storing n nodes is $O(n)$, and the average degree of each node is m. The space complexity of storing all nodes and their neighbor nodes is $O(mn)$. The space complexity of storing primary community is $O(2n)$ in a bipartite network. And the space complexity of storing collections of nodes with multiple most similar nodes is $O(n)$. Therefore, the space complexity of CDBNS algorithm is $O(n+mn+2n+n) = O(mn)$.

4 RASCS Algorithm

With the development of information technology, there's an increasing tendency in the sizes of real-world datasets. Therefore, it is a very challenging problem to find an efficient and accurate method to obtain valuable information in a limited time. The idea of combining community detection with a recommendation algorithm is proposed, which can improve recommendation efficiency. Therefore, in this paper, we combine CDBNS algorithm with user-based collaborative filtering algorithm [25], a new recommendation algorithm, RASCS (Recommendation Based on Attribute Similarity and Community Structure) is proposed. This algorithm preprocesses the data and obtains the community information of each user based on the CDBNS algorithm. At the same time, we take the user attributes into account to define the similarity of user attributes, and select the similar users in the target user community according to the similarity of user attributes, and then completes the recommendation.

For the rationality of recommendation, it is well understood that users in a community will have similar hobbies and lifestyles, so the interest in commodities may be similar to a large extent.

4.1 Proposal for Algorithm

At present, the mainstream recommendation algorithms are facing many problems, among which the sparsity of data is an urgent problem to be solved. Take Taobao for example, the number of registered users and online goods of Taobao has reached the order of hundreds of millions, while the number of visitors to Taobao has reached tens of millions or even hundreds of millions every day, and tens of thousands of goods are sold every minute. If the collaborative filtering algorithm is employed for the recommendation, the size of the matrix will be very large. Therefore, it will be very difficult to calculate on this order of magnitude. In fact, there are only dozens of goods that users really buy or are interested in. In this case, the intersection of goods purchased by any two users is very small, so the effect will be very bad. Although the application of community detection in recommendation process can solve the problem of data sparsity to a certain extent, when faced with a large network, there are still a large number of users in a community, which will lead to some unnecessary calculations. In addition, in real life, attributes between users have a very large impact on user interest. For example, two users with the same gender but at different ages may not like the same things. The age group is the same, but two users of different genders may not like the same things. The similarity between two users calculated by the similarity formula is merely a kind of structural similarity, so we cannot just take structural similarity as the only index to measure whether two users are similar, but also consider the influence of attribute factors.

4.2 Definitions

The idea of RASCS algorithm is to divide users with high similarity into a community by community detection strategy, and then select the top K users as the most similar user set according to the order of attribute similarity and the influence of attribute factors on similarity, and then make relevant recommendations to the target users. Relevant definitions are as follows:

Definition 3: For attributes that can be quantified, taking age for example, the value of a user u on attribute i is D_u^i, and the value of another user v on attribute i is D_v^i, the similarity between users u and v on attribute i is defined as follows:

$$S^i(u,v)=\begin{cases}1, & D_u^i=D_v^i\\ \dfrac{1}{\left|D_u^i-D_v^i\right|}, & D_u^i\neq D_v^i\end{cases}, \tag{4}$$

Definition 4: For attributes of enumeration types, if the value of a user u on attribute i is D_u^i, and the value of another user v on attribute i is D_v^i, the similarity between users u and v on attribute i is defined as follows:

$$S^i(u,v)=\begin{cases}1, & D_u^i=D_v^i\\ 0, & D_u^i\neq D_v^i\end{cases}, \tag{5}$$

Definition 5: Assume that the set of attributes of user u is N_u and the set of attributes of user v is N_v, the attribute similarity between the two users is defined as follows:

$$S(u,v)=\frac{1}{\left|N_u\bigcap N_v\right|}\sum_{i\in N_u\cap N_v}S^i(u,v), \tag{6}$$

4.3 Algorithm Process

The top K users and the top N recommended items that are most similar to the target users are pre-setted with empirical values, and they will be probably altered in computation under specific conditions.

The cold start problem in the process of the algorithm mainly appears when new users register. When a new user registers, the algorithm searches for a community that matches the new user according to the corresponding personal information provided by the user when registering, and then recommends products to the new user in terms of the users in the community.The RASCS algorithm steps are as follows:

Algorithm. RASCS

Input: User-Item scoring information.

Output: recommended list.

```
Executing CDBNS algorithm and return the result G of the bipartite network.
Establishing a two-dimensional matrix A_{m×n} according to the scoring records of user-item.
// Calculating the attribute similarity between users (only including age and gender).
for nei in neighbors:
// When age is the same.
  if communities[nei][1]=communities[userId][1]:
      age ← 1.0
  else
      age ← 1.0 / abs(communities[nei][1] - communities[userId][1])
  end if
// When gender is the same.
```

12. **if** communities[nei][2]=communities[userId][2]:
13. gender ← 1.0
14. **else**
15. gender ← 0
16. **end if**
17. neighbors_dist[nei] ← (age_si + gender_si) / num_att
18. **end for**
19. Selecting the top *K* users as the most similar user set of target users.
20. Sorting according to the recommended value, and selecting the top *N* recommendations to the target user.
21. **return** recommended_list

To visually describe the process in more detail, a flowchart is drawn as shown in Figure 3.

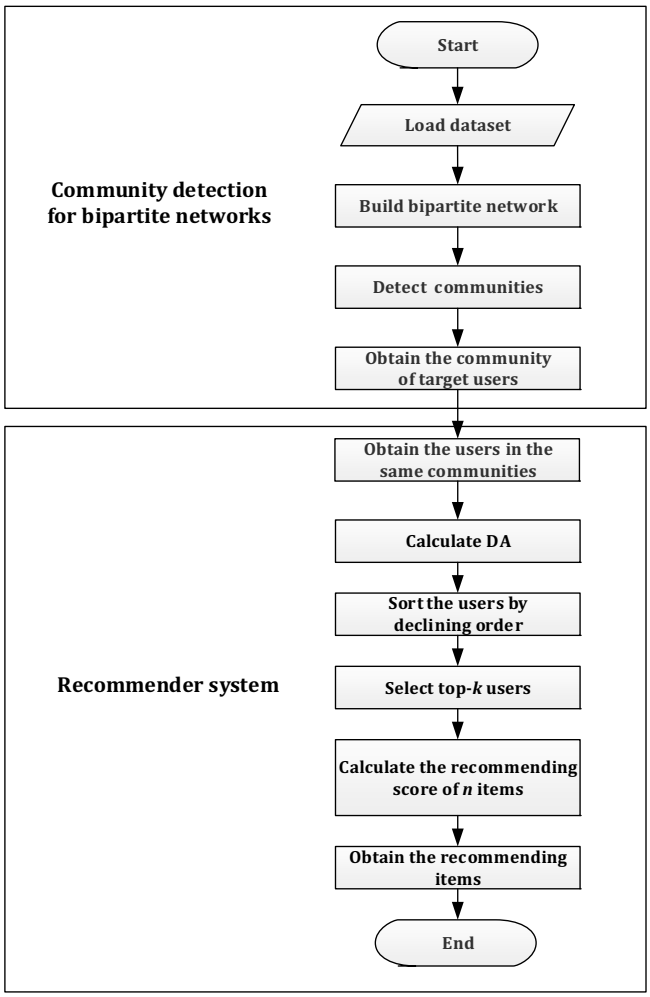


Figure 3. The flow chart of RASCS algorithm

5 Experimental Analysis

This chapter is divided into three parts: DA similarity index experiment, CDBNS algorithm experiment, and RASCS algorithm experiment.

The lab environment is configured with Intel(R) Core (TM) i5-6600 CPU @ 3.30 GHz. The system memory is 8.00 GB. The operating system is Windows 7 64-bit flagship version. The programing language is Python 2.7 with NetworkX 1.11.

5.1 DA Similarity Index Experiment

In this experiment, the MovieLens dataset [26] is one of the most commonly employed datasets in the recommendation mission based on complex networks, such as Kart et al. proposed a supervised machine learning-based link prediction model for weighted and bipartite social networks, which employs MovieLens dataset [27], and Son et al. proposed a novel CBF method that employs a multiattribute network to effectively reflect several attributes when calculating correlations for recommending items to users [28], which is published by GroupLens. As a user-movie-scoring dataset, the dataset contains more than 100,000 scoring records for 1,682 movies by 943 users. The dataset is a typical bipartite network, including two types of nodes, that is, movie and user, each record is an edge of the bipartite network. In user-based collaborative filtering algorithm, the similarity index is also employed to calculate the similarity between users to find a similar user set of target users. Therefore, this experiment combines the DA similarity index with user-based collaborative filtering algorithm.

In this experiment, three evaluation criteria, Precision, Recall and F-measure, are employed to evaluate the performance of this experiment. Precision represents the ratio of all "correctly retrieved results" to all "actually retrieved". Recall represents the ratio of all "correctly retrieved results" to "all results that should be retrieved". The contradiction between Precision and Recall sometimes occurs, so they need to be considered comprehensively. F-measure combines Precision and Recall as a harmonic mean. The greater the Precision, Recall and F-measure, the better the result is.

In the collaborative filtering algorithm, if $K(u)$ is employed to recommend K items to the target user u, and $G(u)$ is employed to represent the items of real interest to the target user u in the test set, the corresponding calculation formulas of the three evaluation criteria are as follows:

$$Precision = \frac{\sum_u |K(u) \cap G(u)|}{\sum_u |K(u)|}, \quad \textbf{(7)}$$

$$Recall = \frac{\sum_u |K(u) \cap G(u)|}{\sum_u |G(u)|}, \tag{8}$$

$$F\text{-}measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{9}$$

In this paper, the experimental results are compared with RA, CN and AA. These three indices are selected because they have better performance in link prediction.

The number of nearest neighbor nodes are set to 5, 10, 20, 40, 80, and 160, resulting in experimental data for Precision, Recall, and F-measure as shown in Figure 4, Figure 5, and Figure 6.

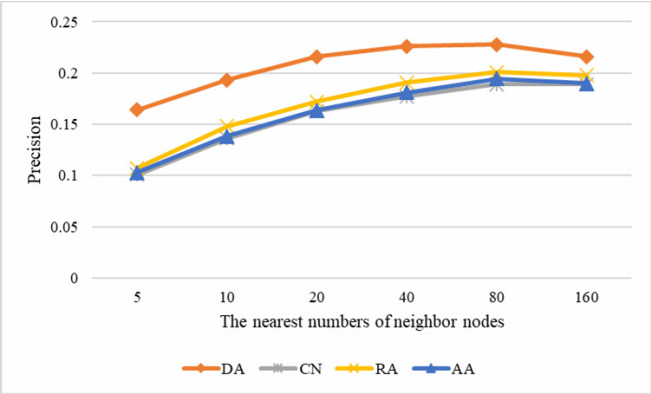


Figure 4. Comparison of Precision experiment results

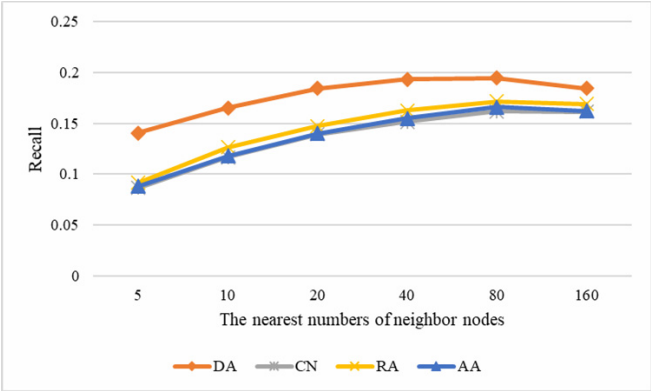


Figure 5. Comparison of Recall experiment results

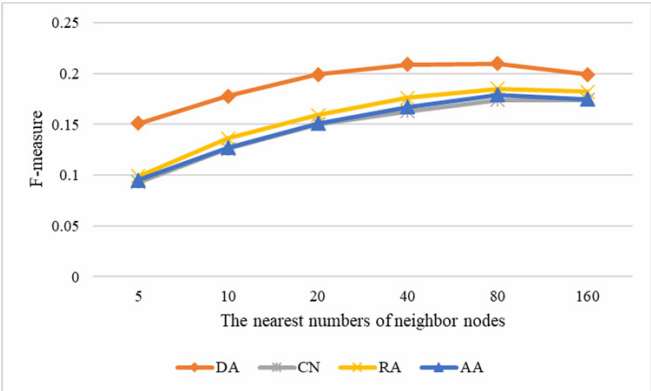


Figure 6. Comparison of F-measure experiment results

Combined with Figure 4 to Figure 6, following the principle that the larger the value is, the better the results will be, we can see that the proposed DA similarity index performs better on Precision, Recall, and F-measure than other traditional measurement indices.

According to Figure 4, the similarity index in Precision is higher than other indices in the number of nearest neighbors. The maximum value of Precision is 0.2276, while the maximum values of CN, RA, and AA are 0.1892, 0.2009, and 0.1944 respectively.

As is shown in Figure 5, for the Recall value, the proposed similarity index is higher than other indices in the number of nearest neighbors. The maximum Recall is 0.1945, while the maximum CN, RA, and AA are 0.1617, 0.1717, and 0.1661 respectively.

In Figure 6, for the F-measure value, the proposed index has obvious advantages in the case of a different number of nearest neighbor nodes. The maximum F-measure of the proposed method is 0.210, while the maximum CN, RA, and AA are 0.174, 0.185, and 0.179 respectively.

5.2 CDBNS Algorithm Experiment

Three datasets are employed to measure the performance of CDBNS algorithm, and the experimental results are compared with that of Louvain [29], Givern-Newman (GN) [30] and Fast-Newman (FN) [31] in modularity and time-consuming.

(1) Pedgett Florentine Families [32]. The multiplex social network consists of 2 layers (marriage alliances and business relationships) describing florentine families in the Renaissance.

(2) Zachary's Karate Club [33]. This is a social network of friendships between 34 members of a karate club at a US university in the 1970s. Each node represents a member of the club, and each edge represents the connection between the two members of the club.

(3) Dolphin social network [34]. This is an undirected social network of frequent associations between 62 dolphins in a community living off Doubtful Sound.

Louvain algorithm performs hierarchical community detection in approximately linear time complexity. It is a modularity optimization algorithm widely employed at present. The quality of community detection is measured by modularity. The higher the modularity, the higher the quality of community detection. GN algorithm is a split-type hierarchical community detection algorithm, which can segment the network by continuously removing the edge with the largest betweenness from the network. FN algorithm is a fast community detection algorithm based on modularity. These three algorithms are widely employed and then representative in community detection, therefore, we employ them as baselines to prove the superiority of the proposed algorithm. The results are shown in Table 2.

Table 2. Modularity comparison

Dataset	Algorithm	Modularity	Time(s)
Pedgett Florentine Families	CDBNS	0.508	0.005
	Louvain	0.419	0.079
	GN	0.394	0.079
	FN	0.397	0.139
Zachary's Karate Club	CDBNS	0.466	0.010
	Louvain	0.389	0.011
	GN	0.401	0.221
	FN	0.381	0.086
Dolphin Social Network	CDBNS	0.515	0.029
	Louvain	0.510	0.010
	GN	0.519	0.759
	FN	0.492	0.058

The proposed algorithm performs well on Pedgett Florentine Families with 0.508 in modularity, its modularity on Zachary's Karate Club is superior to the baselines, and the modularity on Dolphin social network is 0.04 smaller than that on GN algorithm. Besides, the proposed algorithm is superior to GN and FN and is close to Louvain algorithm in time-consuming.

5.3 RASCS Algorithm Experiment

The dataset employed in this experiment is the MovieLens dataset published by GroupLens. When calculating attributes similarity, gender and age are under consideration, and the dataset is divided into three different sizes: 5000, 20000 and 100000.

RACD algorithm [16], Item-Based Collaborative Filtering Recommendation Algorithms (ItemBasedCF) and User-based Collaborative Filtering Recommendation Algorithm (UserBasedCF) are used as the baselines in the experiment.

The idea of RACD algorithm is to combine a bipartite network community detection algorithm with user-based collaborative filtering algorithm. The core idea of UserBasedCF is that if a product is recommended to user a, the first step is to find a group that is similar to user a, and the second step is to use a weighting strategy to recommend a product that user a has not seen or selected.

5.3.1 Time-consuming Comparison of Recommendation to a Single User

Taking the target node "7" as an example, the time taken to finally complete the recommendation for the user is shown in Table 3.

Table 3. Time-consuming for single-user recommendations

Algorithm	Time(s)
ItemBasedCF	0.065
UserBasedCF	0.050
RACD	0.023
RASCS	0.021

As can be seen from Table 3, this algorithm consumes 0.021s which is superior to 0.023s of RACD, 0.050s of UserBasedCF and 0.065s of ItemBasedCF when it is recommended to a single user.

5.3.2 Time-consuming Comparison of Recommendation to Multiple Users

When the target user is uncertain, all users in the network need to be recommended. We have compared the time-consuming on different sizes of datasets, here 5,000, 20,000, and 100,000 records of data are respectively employed.

In this experiment, the running time is obtained following the method in reference [16], that is, the process of community detection is conducted offline, and the results of community detection are saved in the file. The contents of the file are employed as the given data when performing recommendations, so the total time is taken to read the community results and then make the recommendation. As shown in Table 4.

Table 4. Time-consuming for multi-user recommendations

Size	Algorithm	Time(s)
5000	ItemBasedCF	6.975
	UserBasedCF	6.368
	RACD	2.252
	RASCS	1.492
20000	ItemBasedCF	35.176
	UserBasedCF	46.938
	RACD	47.322
	RASCS	7.553
100000	ItemBasedCF	72.503
	UserBasedCF	146.473
	RACD	185.204
	RASCS	41.421

As can be seen from Table 4, when the size of dataset is 5000, the time-consuming of the proposed algorithm is 1.492s, which is 5.483s faster than ItemBasedCF, 4.876s faster than UserBasedCF and 0.760s faster than RACD. When the size of dataset is 20000, the time-consuming of the proposed algorithm

is 7.553s, which is 27.623s faster than ItemBasedCF, 39.385s faster than UserBasedCF and 39.769s faster than RACD. Moreover, when the size of dataset is 100000, the time-consuming of the proposed algorithm is 41.421s, which is 31.082s faster than ItemBasedCF, 105.052s faster than UserBasedCF and 143.783s faster than RACD.

5.3.3 Comparison of Recommendation Effect

In order to measure more accurate recommendation results, the whole data set is divided into test set and train set in the ratio of 2:8. Because of the randomness of the experiment, we carry out the experiment for 20 times on RASCS and 20 results are obtained.

The final result for RASCS is the average of these 20 results, which is shown in Table 5. UserBasedCF and ItemBasedCF are employed as baselines for comparison.

Table 5. Results of experiments

Algorithm	Precision	Recall	F-measure
RASCS	0.2037	0.0960	0.1305
UserBasedCF	0.1943	0.0824	0.1157
ItemBasedCF	0.1905	0.0816	0.1142

As shown in Table 5, the proposed algorithm outperforms the baselines in Precision, Recall, and F-measure. The reason is that we employ the user interests in the same community as the target users to provide recommendations for the target users, therefore, it makes the recommendation more targeted, achieves higher accuracy and outperforms the baselines.

5.4 Analysis of Experimental Process and Results

In the DA similarity index experiment, we take MovieLens dataset as experimental dataset to measure the rationality, reliability and effectiveness of DA similarity index. The experimental results show that the DA similarity index is superior to RA, CN and AA in Precision, Recall and F-measure under the same experimental mission. The advantage of the DA index is that it combines the degree of the two sets of nodes of bipartite networks and the degree of nodes' common neighbor. The DA index provided more information about the target nodes and their neighbor nodes than the compared algorithms.

In the CDBNS algorithm experiment, we employ the Pedgett Florentine Families, Zachary's Karate Club, and Dolphin Social Network as experimental datasets and three classical modularity-based algorithms as baselines to measure the performance of the proposed algorithm. The experiment results prove that the CDBNS algorithm is superior to the state-of-the-art algorithms in modularity and has advantages in time-consuming.

Besides, we divide the MovieLens dataset into different sizes to measure the performance of the proposed algorithm on datasets with different sizes in the RASCS algorithm experiment. The experiment consists of two parts of recommendations, that is single-user and multi-user. We can conclude that the sizes of the dataset significantly affect the running time of the algorithm and has a trivial effect on the accuracy of the final result. The results also indicate that the RASCS algorithm outperforms the baselines in time-consuming, Precision, Recall, and F-measure.

6 Conclusions

This paper proposes a measure index of node similarity (DA similarity index) for bipartite networks, which incorporates the influence of the degree of two types of nodes as well as the influence of the common neighbors of two types of nodes on the similarity calculation. Furthermore, a community detection algorithm based on the DA similarity index for bipartite networks (CDBNS algorithm) is proposed, which is parameter-free and does not extract the maximal connectivity subgraph of bipartite networks. It only measures the similarity between nodes to identify the communities, and merges the nodes with the maximum similarity into the same community. Sequentially, combining the CDBNS algorithm and user-based collaborative filtering algorithm, the method of calculating user attribute similarity is designed, and a new recommendation algorithm (RASCS algorithm) is proposed. Finally, the DA similarity index, CDBNS algorithm, and RASCS algorithm are verified by experiments and compared with typical traditional methods. The experimental results show that the DA similarity index is reasonable and effective, CDBNS algorithm and RASCS algorithm outperform classical algorithms.

Acknowledgments

This paper was funded by Liaoning Natural Science Foundation under Grant No.20170540320, the Doctoral Scientific Research Foundation of Liaoning Province under Grant No.20170520358 and the Fundamental Research Funds for the Central Universities under Grant No. N172415005-2, No. N2017010.

References

[1] S. Qiao, N. Han, K. Zhang, L. Zou, H. Wang, L. A. Gutierrez, Algorithm for Detecting Overlapping Communities from Complex Network Big Data, *Journal of Software*, Vol. 28, No. 3, pp. 631-647, March, 2017.

[2] Y. Liu, L. Zhang, L. Nie, Y. Yan, D. S. Rosenblum, Fortune Teller: Predicting Your Career Path, *2016 Thirtieth AAAI Conference on Artificial Intelligence*, Phoenix, Arizona, USA,

2016, pp. 201-207.

[3] D. Preoțiuc-Pietro, Y. Liu, D. Hopkins, L. Ungar, Beyond Binary Labels: Political Ideology Prediction of Twitter Users, *Meeting of the Association for Computational Linguistics*, Vancouver, Canada, 2017, pp. 729-740.

[4] M. Wang, C. Wang, J. X. Yu, J. Zhang, Community Detection in Social Networks: An In-Depth Benchmarking Study with A Procedure-Oriented Framework, *Proceedings of the VLDB Endowment*, Vol. 8, No. 10, pp. 998–1009, June, 2015.

[5] M. Azaouzi, D. Rhouma, L. B. Romdhane, Community Detection in Large-Scale Social Networks: State-of-the-Art and Future Directions, *Social Network Analysis and Mining*, Vol. 9, No. 1, pp. 23, December, 2019.

[6] D. J. Watts, S. H. Strogatz, Collective Dynamics of 'small-world' Networks, *Nature*, Vol. 393, No. 6684, pp. 440-442, June, 1998.

[7] A.-L. Barabási, R. Albert, Emergence of Scaling in Random Networks, *Science*, Vol. 286, No. 5439, pp. 509-512, October, 1999.

[8] X. Huang, D. Chen, T. Ren, D. Wang, A Survey of Community Detection Methods in Multilayer Networks, *Data Mining and Knowledge Discovery*, Vol. 35, No. 1, pp. 1-45, January, 2021.

[9] D. Chen, P. Du, B. Fang, D. Wang, X. Huang, A Node Embedding-Based Influential Spreaders Identification Approach, *Mathematics*, Vol. 8, No. 9, pp. 1554, September, 2020.

[10] L. Zhang, M. Zhao, D. Zhao, Bipartite Graph Link Prediction Method with Homogeneous Nodes Similarity for Music Recommendation, *Multimedia Tools and Applications*, Vol. 79, No. 19-20, pp. 13197-13215, May, 2020.

[11] M. Gao, L. Chen, A Projection Based Algorithm for Link Prediction in Bipartite Network, *Meeting of International Conference on Information System and Artificial Intelligence (ISAI)*, Hong Kong, China, 2016, pp. 56-61.

[12] D. Melamed, Community Structures in Bipartite Networks: A Dual-Projection Approach, *PloS One*, Vol. 9, No. 5, pp. e97823, May, 2014.

[13] Y. Cui, X. Wang, Detecting One-Mode Communities in Bipartite Networks by Bipartite Clustering Triangular, *Physica A: Statistical Mechanics and its Applications*, Vol. 457, No. 1, pp. 307-315, September, 2016.

[14] B.-L. Chen, Y. Yuan, Y.-J. Zhang, F.-F. Li, Q. Yu, Measurement and Algorithm for Overlapping Community Partitioning in Bipartite Networks, *CISIS 2018: Complex, Intelligent, and Software Intensive Systems*, Matsue, Japan, 2018, pp. 431-439.

[15] S. J. Beckett, Improved Community Detection in Weighted Bipartite Networks, *Royal Society open science*, Vol. 3, No. 1, pp. 140536, January, 2016.

[16] D. Chen, Y. Yan, D. Wang, X. Huang, Community Detection Algorithm Based on Structural Similarity for Bipartite Networks, *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China, 2016, pp. 98-102.

[17] R. Real, J. M. Vargas, The Probabilistic Basis of Jaccard's Index of Similarity, *Systematic biology*, Vol. 45, No. 3, pp. 380-385, September, 1996.

[18] T. A. Sørensen, A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species content and Its Application to Analyses of the Vegetation on Danish Commons, *Biologiske Skrifter/Kongelige Danske Videnskabernes Selskab*, Vol. 5, No. 4, pp. 1-34, January, 1948.

[19] G. Salton, M. J. McGill, Introduction to Modern Information Retrieval, *New York: Mc Graw Hill*, 1983.

[20] T. Alzahrani, K. Horadam, Finding Maximal Bicliques in Bipartite Networks using Node Similarity, *Applied Network Science*, Vol. 4, No. 1, pp. 21, May, 2019.

[21] T. Zhou, L. Lü, Y.-C. Zhang, Predicting Missing Links via Local Information, *The European Physical Journal B*, Vol. 71, No. 4, pp. 623-630, October, 2009.

[22] L. A. Gundala, F. Spezzano, Estimating Node Indirect Interaction Duration to Enhance Link Prediction, *Social Network Analysis and Mining*, Vol. 9, No. 1, pp. 17, December, 2019.

[23] K. Berahmand, A. Bouyer, A Link-Based Similarity for Improving Community Detection Based on Label Propagation Algorithm, *Journal of Systems Science and Complexity*, Vol. 32, No. 3, pp. 737-758, June, 2019.

[24] M. Plantié, M. Crampes, Survey on Social Community Detection, in: N. Ramzan, R. V. Zwol, J. S. Lee, K. Clüver, X. S. Hua (eds.), *Social Media Retrieval Computer Communications and Networks*, Spring, London, 2013, pp. 65-85.

[25] F. Ortega, D. Rojo, P. Valdiviezo-Diaz, L. Raya, Hybrid Collaborative Filtering Based on Users Rating Behavior, *IEEE Access*, Vol. 6, pp. 69582-69591, November, 2018.

[26] F. M. Harper, J. A. Konstan, The Movielens Datasets: History and Context, *ACM Transactions on Interactive Intelligent Systems*, Vol. 5, No. 4, pp. 19, January, 2016.

[27] O. Kart, O. Ulucay, B. Bingol, Z. Isik, A Machine Learning-based Recommendation Model for Bipartite Networks, *Physica A: Statistical Mechanics and its Applications*, Vol. 553, No. C, pp. 124287, September, 2020.

[28] J. Son, S. B. Kim, Content-based Filtering for Recommendation Systems using Multiattribute Networks, *Expert Systems with Applications*, Vol. 89, No. C, pp. 404-412, December, 2017.

[29] V. D. Blondel, J. L. Guillaume, R. Lambiotte, E. Lefebvre, Fast Unfolding of Communities in Large Networks, *Journal of Statistical Mechanics: Theory & Experiment*, Vol. 2008, No. 10, pp. 10008, October, 2008.

[30] M. Newman, M. Girvan, Finding and Evaluating Community Structure in Networks, *Physical Review E*, Vol. 69, No.2, pp. 026113, February, 2004.

[31] M. Newman, Fast Algorithm for Detecting Community Structure In Networks, *Physical Review E*, Vol. 69, No. 6, pp. 066133, June, 2004.

[32] J. F. Padgett, C. K. Ansell, Robust Action and the Rise of the Medici, 1400-1434, *The American journal of sociology*, Vol.

98, No. 6, pp. 1259-1319, May, 1993.

[33] W. W. Zachary, An Information Flow Model for Conflict and Fission in Small Groups, *Journal of Anthropological Research*, Vol. 33, No. 4, pp. 452-473, Winter, 1977.

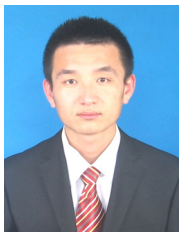
[34] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, S. M. Dawson, The Bottlenose Dolphin Community of Doubtful Sound Features a Large Proportion of Long-Lasting Associations, *Behavioral Ecology and Sociobiology*, Vol. 54, No. 4, pp. 396-405, September, 2003.

Biographies

Dongqi Wang is a lecturer in Software College, Northeastern University, China. He was awarded Ph.D. degree in 2011. His research areas include network security and social network analysis. He has published more than 10 papers indexed by SCI, EI and ISTP. He is also a member of CCF and ACM.

Mingshuo Nie is a Master candidate in Software College, Northeastern University, China. His research interests include complex networks, link prediction, social network analysis and deep reinforcement learning on graph.

Dongming Chen is a professor of Software College, Northeastern University, China. He is a member of ACM, IEEE Computer Society, senior member of China Computer Federation (CCF), and Senior member of China Institute of Communications (CIC). His research interests include complex networks, social network analysis and information security.



Li Wan graduated from Software College, Northeastern University, China. He was awarded a Master's degree in 2019. His research interests include recommender system, and community detection.

Xinyu Huang graduated from Software College, Northeastern University, China. He was awarded a Ph.D. degree in 2020. His research interests include complex networks, multilayer network, community detection and social network analysis.