Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 5931727, 10 pages
https://doi.org/10.1155/2022/5931727

Research Article

Overlapping Community Detection Based on Strong Tie Detection and Non-Overlapping Link Clustering

Lin Guo[1] **and Miao Zhang**[2]

[1]*School of Economics and Management, Changchun University of Science and Technology, Changchun, Jilin 13022, China*
[2]*College of Electronics and Information Engineering, Tongji University, Shanghai 200082, China*

Correspondence should be addressed to Lin Guo; guolin@cust.edu.cn

Received 18 October 2021; Revised 31 May 2022; Accepted 10 June 2022; Published 22 December 2022

Academic Editor: Jian Lin

Many clustering algorithms are in favour of node-based methods, but a link between nodes has one single feature, so link-based clustering is sometimes easier than node-based methods. Being dependent on the characteristics of links, a detection algorithm for a non-overlapping link community is put forward in this paper. The method proposed also distinguishes the differences between nodes with a high degree of accuracy and detects communities with a minimal number of overlapping nodes. On the basis of three different datasets, experiments were conducted to compare the proposed algorithm with different non-overlapping and overlapping clustering algorithms, and the results show that our algorithm generates the least number of overlapping nodes and achieves a good community partition.

1. Introduction

Community detection [1–3] is used to divide datasets into several communities based on the relationships among users, from which the network structure can be identified and the functional roles of users can be analyzed. Overlapping community detection algorithms unfold as follows [4–7]: The clique percolation method is executed, clique expansion, local fitness maximization, rough set theory, graph clustering, and so on. Most existing clustering techniques have focused on topological structures based on various criteria, including normalized cuts, molecularity, structural density, and stochastic flows or cliques [8, 9]. In general, the kernel of those algorithms mentioned above is the performance of clustering on the basis of the similarities between nodes or topological structures; they classify nodes with common attributes or characteristics into the same clusters. There are many research studies about clustering, but most of them are in favour of node-based methods.

Node-based methods [10–12] envisage that the strong ties are formed between nodes when there is a high probability of forming triads through different types of relationships. Notice that nodes belong to multiple groups, but links are existent for just one dominant reason (e.g., two people linked work together or have common interests), which means that links that occupy unique clusters and nodes naturally account for multiple clusters as a result of their links.

The idea of link communities has been proposed independently by a number of authors in both physics [13] and machine learning literature [14, 15]. It is used to create communities when there are different types of edges in an ego network [16]. Although clustering links is a much more flexible and simpler approach than clustering nodes for edges that occupy a unique feature and nodes that occupy multiple features due to their links, the work on link community construction is obviously lacking.

There are two main difficulties in detecting communities from ego-network [17]: First, the number of communities in a given network is unknown, which is the usual drawback of many algorithms because they do not have valid criteria for measuring the community structure. Two, nodes belong to more than one community, which means overlapping community structures exist in complex networks. The

proposed link clustering method can be used to solve these problems well. Non-overlapping link clustering is used in this paper to achieve overlapping node clustering when all links of a node belong to one cluster. This is followed by classifying the node to this cluster or, otherwise, classifying it under the several corresponding clusters. Problems can be simplified by converting the analysis objects from nodes to links, and link-based approaches are good.

The remainder of this paper is organized as follows: Section 2 proposes the concrete implementation strategy and algorithm description in detail. Section 3 introduces the experimental results. Section 4 states the conclusions of research achievements. Section 5 states the compliance with ethical standards.

2. Non-Overlapping Link Clustering

Assumption 1. Any two reachable nodes that are indirectly linked in a graph have a certain probability of being directly connected under some external promptings.

Although social networks contain vast amounts of data that are produced by users, there is not enough available information about single users or single phenomena. To cope with the data sparseness problem, we need to enrich knowledge reserves and expand the network structure by adding implicit edges according to Assumption 1.

If nodes a and b are linked together and nodes b and c are linked together, but nodes a and c are not linked, then the conclusion that nodes a and c will never be directly associated with each other at any time in the future is incorrect. The non-linked nodes only indicate that the existing environment does not meet the conditions necessary to generate a direct connection, and those nodes may be linked immediately by some driving forces. Therefore, the probabilities of linking the centre node with the reachable and non-adjacent nodes need to be computed, after which valuable information can be mined by finding implicit edges to enrich knowledge about the centre node.

2.1. Strong Tie Detection. There are various types of nodes in heterogeneous networks, and the ways of linking a node with others are also different and embodied by the types of links (directed or undirected). For instance, if the connection is built by following or being followed methods, then the edge is one-way; if the connection is built by common interests, then the edge is bidirectional. When analyzing the characteristics of the complex network for a particular object, this object is set to be the centre node of the social network.

Definition 1 (egoNet): A network that sets the analyzed object as the centre node (denoted as ego) is called egoNet. The presentation format is $<N, E, W>$, where N is the set of nodes representing the members of the egoNet, E is the set of edges denoting the complex relationships, and W is the set of weights signifying the connection strengths between the ego and other nodes.

The nodes in egoNet have certain connection and describe some features of the ego. In addition, the analysis of the ego node does not need to study all nodes in the network. Therefore, the egoNet is used for processing and analyzing a particular problem or phenomenon.

Definition 2 (positive edge): e_0 is the ego in a directed path, and the edge that has the same direction as one of the point-out edges of e_0 is called the positive edge, such as $<e_0 \longrightarrow e_1 \longrightarrow e_2 \longrightarrow \ldots \longrightarrow e_n>$.

Definition 3. e_0 is the ego in a directed path, and the edge that has the same direction as one of the point-in edges of e_0 is called a negative edge, such as $<e_0 \leftarrow e_1 \leftarrow e_2 \leftarrow \ldots \leftarrow e_n>$.

If one is just analyzing adjacent nodes, then a partial conclusion will be drawn. Therefore, it is essential to compute the probability of establishing a connection between the ego and its reachable and indirectly linked nodes and finally replenish the structure of the network by placing edges between the nodes of high probability of connection.

Edges can be analyzed on the basis of the distribution of interests or behaviour trajectories to determine the connection strengths among nodes. User-item matrix $\mathbf{A}$ is built by analyzing the level of user's attention to different items, where $\alpha_i = \{\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{ir}\} \in \mathbf{A}$ is the ith row vector of the matrix, and α_{ic} represents the amount of time user i focuses on item c. For the convenience of data comparison among different users, α_{ic} should be normalized by the sum of all the members in α_i. The calculation formula is the following:

$$eic = \frac{\alpha ic}{\sum_{k=1}^{r} \alpha_{ik}}, \tag{1}$$

where $e_{ic} \in (0, 1)$ represents the degree of attention user i pays to item c. Supposing that an edge exists between nodes p and q due to the common interest in item c, the formula for calculating the linkage intensity is the following:

$$\omega_{pq} = 1 - \exp\left(-e_{pc} \times e_{qc}\right). \tag{2}$$

Supposing multiple interrelationships exist between nodes a and b, the probability of constructing a direct connection is influenced by two factors (Node a is the ego): first, the shortest-path between nodes a and b, and second, the strength of the tie between Node a and the nodes in the shortest path from a to b.

2.1.1. The Impact of Factor 1. The more intermediate nodes in the shortest path from a to b there are, the less likely it is that a link will be constructed between them. Therefore, we use $t(1-t)^l$ to express the influence of the length of the shortest path on the probability of linking the nodes at the ends of the path, where parameter t is the declining factor of the establishment of a direct interrelation, and l is the number of nodes in the shortest path.

2.1.2. The Impact of Factor 2. Because Node a is the ego, the analysis of whether a connection will be generated is focused on the ego, that is, it is used to test whether Node a will be linked to other nodes. The algorithm proposed in this paper can handle both undirected and directed graphs. In practice, the situation of a directed graph is more complex than that of an undirected graph. Therefore, it is necessary to pay attention to the direction of each edge in the shortest path.

$$w(x, y) = \sum_{k} \rho^{f(x,y)} \times \omega_{\gamma(\gamma+1)}, \tag{3}$$

where w is the probability of generating a direct interrelation between nodes x and y, ρ is the impact of a negative edge on generating a positive edge from a to b, and $f(x, y)$ judges whether a negative edge exists in the shortest path from nodes x to y. Due to the multiple relations between the nodes in the complex network, many homogeneous or heterogeneous edges among nodes may exist, so it is essential to accumulatively calculate the influence of each edge on the probability. It can be seen that the probability of linking the start node and the end node in the path is the product of the probability of linking each pair of the adjacent nodes because the intermediate nodes have the same degree of influence on the final result. The formula is as follows:

$$L = \prod_{\gamma=\text{start}}^{\text{end}-1} \left(\sum_{k} w(\gamma, (\gamma+1)) \right) = \prod_{\gamma=\text{start}}^{\text{end}-1} \left(\sum_{k} \rho^{f(\gamma,(\gamma+1))} \times \omega_{\gamma,(\gamma+1)} \right). \tag{4}$$

By integrating the above two factors, the final formula of the probability of placing a link between two reachable nodes with multiple relations is the following:

$$P = t(1-t)^{l} \prod_{\gamma=\text{start}}^{\text{end}-1} \left(\sum_{k} \rho^{f(\gamma,(\gamma+1))} \times \omega_{\gamma(\gamma+1)} \right). \tag{5}$$

Therefore, the formula for calculating the probability of establishing direct ties among nodes is as follows:

$$\begin{aligned} f(v, w) &= \begin{cases} 1, & 1_{\{\text{Rdirec}(v,w)=1\}}, \\ 0, & \text{else}, \end{cases} \\ s_{ij} &= \begin{cases} t(1-t)^{l} \prod_{\gamma=i}^{j-1} \left(\sum_{k} \rho^{f(\gamma,(\gamma+1))} \times \omega_{\gamma(\gamma+1)} \right) 1_{\{\text{path}(i,j)=1\}}, \\ 0, \quad 1_{\{\text{path}(i,j)=0\}}. \end{cases} \end{aligned} \tag{6}$$

[1fn1]$\{\cdot\}$ is the Heaviside function that yields 1 if the argument is logically true and 0, otherwise. Rdirec (v, w) tests whether negative edges exist between nodes v and w (the direction from v to w is positive), which returns 1 if it exists and 0, otherwise. Function path (i, j) judges whether there is a pathway between nodes i and j, regardless of the nature of the edges. ρ is the impact factor of negative edges in generating an edge from Node v to Node w. t is a diminishing intensity parameter, while l is the length of the shortest path between nodes i and j. Therefore, the longer l is, the smaller the connection is. k is the number of different relationships between nodes i and j. ω_{pq} is the weight of the edge between nodes p and q. S_{ij} is the probability of establishing direct ties between nodes i and j.

2.2. Linking Probabilities Prediction. Starting from the ego, the egoNet can be traversed through by the directions of edges. For instance, <ego, e_1, e_2, …, e_n> is one of the traversal results generated by a certain direction from the ego. If the strength of the link between e_i $(1 \le i \le n)$ and the ego is bigger than threshold Θ, then the edge can be added when e_i is indirectly linked to the ego. If the strength is smaller than Θ, then the traverse is terminated.

This paper only deals with the probability of establishing a direct tie between the ego and its indirectly connected nodes and complements edges with high probability, by which the optimal result can be achieved with the minimal computational complexity by just focusing on the ego.

Figure 1(a) presents the strengths of the links between different nodes (note: direct and indirect edges in the graph represent different kinds of connection). Figure 1(b) is achieved by analyzing Figure 1(a), and the weights of nodes are calculated on the basis of the strengths of links. Figure 1(b) unifies different characteristics of links. The probability of linking is important, for which reason the edges can be simplified to be undirected, un-weighted, and non-overlapping ones. The weight of each ordinary node is computed by the distance between itself and the ego. Therefore, the smaller the distance is, the greater the weight is. Take Node 13 as an example. The process of calculating the weight based on formula (4) is the following: $0.2 \times 0.8^3 \times$

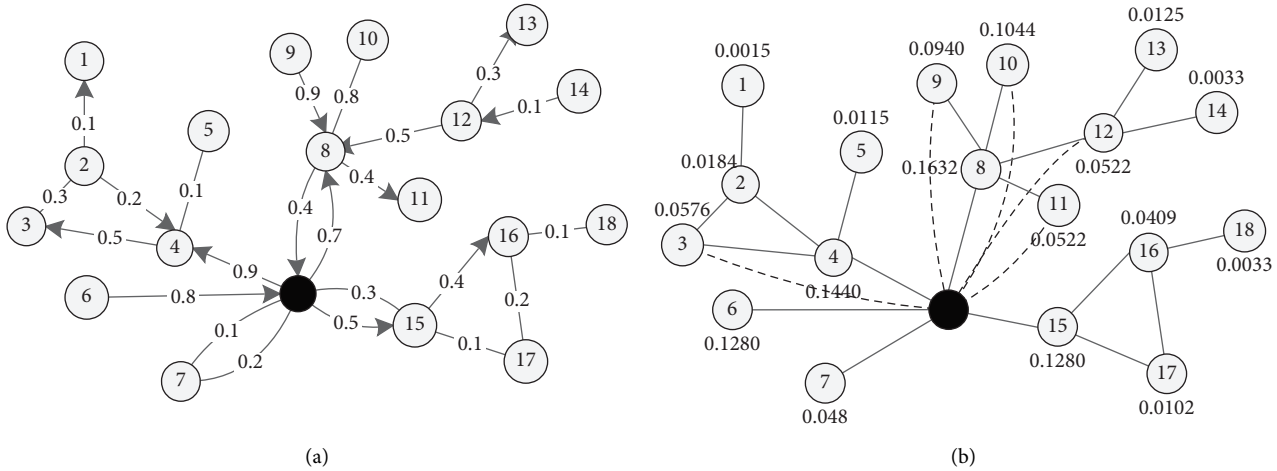


FIGURE 1: egoNet. The black node denotes the ego. $c = 0.2$ and $\rho = 0.8$. (a) Diverse associations between nodes. The numbers denote the strengths of links. (b) Linking probabilities. The numbers denote the probability of linking the ego to the others.

$(0.7 + 0.4 \times 0.8) \times 0.5 \times (0.8 \times 0.3) = 0.0125$. From Figure 1(b), it can be found out that nodes 4, 6, 8, and 15 are relatively close to the ego, and Node 8 is the nearest one. If the threshold φ is set to 0.05, then the link between the ego and Node x will be added when $s_{\text{ego},x} \geq 0.05$, so nodes 3, 9, 10, 11, and 12 are directly linked to the ego.

2.3. Overlapping Node Clustering. When all links of a node belong to one community, then the node can be classified into this corresponding cluster; otherwise it is an overlapping node.

Definition 4 (the node attribute): the attribute of Node c is $Attri_c = <n_1, n_2, n_3, \ldots, n_m>$, where n_x is an adjacent node of c, $x \in [1, m]$.

Supposing that the only information available is network topology, we can use the node's neighbours to illustrate the most fundamental characteristic. Therefore, the analysis of the adjacent node set is equivalent to that of characteristics of the ego.

Definition 5 (the edge attribute): E_{xy} denotes the edge between nodes x and y, and the attribute of E_{xy} is $Attri_{Exy} = Attri_x \cup Attri_y$.

Because the edge attribute is embodied by the two linked nodes, the union of their attribute lists can be used to express the edge attribute, based on which, non-overlapping link clustering can be performed. Each cluster denotes a social dimension, which is usually formed by different locations, religions, and interests. Social dimensions in the complex network are generated in different ways that make their forms and properties different while the analysis of the features of edges is the most concise way of achieving the overlapping partitions of nodes.

Because there is no possibility that a pair of disjoint links (that do not share a node) are similar, the link clustering is limited in connected pairs of links that share a node that is simple and efficient. The link clustering algorithm proposed in this paper is precisely described in Algorithm 1.

The time complexity for judging whether there is an association between nodes is $O(e)$, where e is the number of edges. The time complexity of judging whether the nodes in C and C' meet the merging conditions is $O(n^{2\text{fn}2})$. Hence, the total time is $O(e + n^2)$.

Overlap(B, D) computes the overlapping degree of clusters B and D according to the Jaccard index, which has the following function: $\text{overlap}(B, D) = B \cap D / B \cup D$. Cover($B$, D) computes the degree of the smaller cluster covered by the bigger cluster between B and D, the function of which is $\text{cover}(B, D) = B \cap D / \min(B, D)$. When merging the link clusters, the algorithm not only combines the unnecessary clusters but also controls the number of overlapping nodes and reduces the overlapping areas among clusters. Length(c_i) computes the number of nodes in c_i and prevents the generation of excessive overlapping nodes.

The clustering results of links are several sets of nodes, and the temporary clusters of nodes are recognized as the indicative node sets. If two nodes in the indicative node set have been classified into different clusters, then the connection times of the two clusters should be added. If one of the nodes in the indicative node sets has been classified into one cluster, while the other node has yet to be classified, then the undetected node will be added to this cluster. When the node sets no longer change throughout the iterative process, the operation is terminated. At that moment, further combination is executed based on the connection intension of clusters that is computed according to the indicative node sets.

Threshold Θ_1 is set according to the characteristics of actual data and realistic requests. Threshold Θ_2 can affect the accuracy of community partition and the number of clusters. If Θ_2 is too small, then that will lead to the combination of clusters having many different attributes. On the contrary, if Θ_2 is too big, then that will lead to the generation of many clusters that contain few nodes. The setting method of Θ_2 is described below in detail.

Input: The complex network G
Output: The overlapping node clusters
(1) **FOREACH** (ego, x) **IN** G
(2) **IF** path(ego, x) = = 1 **THEN**
//judging whether a pathway exists between the ego //and Node x
(3) **IF** $S_{\text{ego}x} > \varphi$ && directLink(ego, x) = = 0 **THEN**
//judging whether the ego and Node x are adjacent
(4) link(ego, x)
(5) **ELSE** continue
(6) **END FOREACH**
(7) **FOREACH** path(t, k) = = 1 **IN** G
(8) $Attri_{Link\_tk} = Attri_t \cup Attri_k$
//computing the edge attribute
(9) **END FOREACH**
(10) **FOREACH** ($Link_{xe}$, $Link_{xd}$) IN *edgelist*
(11) **IF** overlap($Attri_{Link\_xe}$, $Attri_{Link\_xd}$) $\geq \Theta 1$ **FOREACH**
(12) put $Link_{xe}$ and $Link_{xd}$ into the same cluster
(13) **ELSE**
(14) divide $Link_{xe}$ and $E = Link_{xd}$ separately
(15) **END FOREACH**
(16) sort C in descending order by the size of node sets
//merging from the core collection
(17) **UNTIL** C' is unchanged **DO**
//C' is the combined clustering sets
(18) combine temporary clusters of high tie strength by analyzing the indicative clusters
(19) **END UNTIL**
(20) delete clusters in which all nodes are covered by other clusters in C'
(21) judge whether c_i in C and c_j' in C' meet the combination conditions:
(a) **IF** C' is null **THEN** add c_i into C'
(b) **IF**cover(c_i, c_j') $\geq \Theta_2$ && length(c_i) > 2 **THEN** merge c_i and c_j'
(c) **IF** cover(c_i, c_j') $< \Theta_2$ && length(c_i) > 2 **THEN** add c_i into C' as a separate clusterAlgorithm 1: BCluster.

Figure 2 presents two special cases of link clustering. Figure 2(a) describes the situation of a node with one edge, based on which nodes like x should be put into the nearest cluster and not be classified separately. Node y in the structure $x - y$ in Figure 2(a) is covered by Cluster A. Thus, the coverage ratio of structure $x - y$ is 1/2. To classify the node with one edge to the nearest cluster, which is the minimum requirement for link clustering, $\Theta_2 = 1/2$ is the lowest limit of Θ_2. Figure 2(b) describes two communities that contain a large number of common overlapping node, but clusters like those should be combined. Because structure A in Figure 2(b) has two nodes covered by Cluster B, the coverage ratio of structure A is 2/3, then $\Theta_2 = 2/3$ is the highest requirements. Given the above, $\Theta_2 \in [1/2, 2/3]$.

An example is given to illustrate the process of link clustering.

On the basis of the algorithm proposed in this paper, the network illustrated in Figure 3 can be analyzed to detect overlapping communities. Table 1 describes the attributes of each node according to the adjacent nodes. The edge attributes can be obtained by computing the union of the attribute sets of the two linked nodes shown in Table 2. Take Edges 1–3 as an example. The process of computing the attribute lists is $Attri_{Link\_1,3} = Attri_1 \cup Attri_3 = \{3, 4, 5, 6, 7\} \cup \{1, 2, 4\} = \{1, 2, 3, 4, 5, 6, 7\}$.

Supposing that $\Theta_1 = 1$, the edges with the same attributes are placed into one cluster. For instance, because $Attri_{Link\_1,3} = Attri_{Link\_1,4} = \{1, 2, 3, 4, 5, 6, 7\}$, the links between 1–3 and 1–4 are classified into one cluster. The results of link clustering are shown in Table 3.

Because the results of temporary link clustering contain redundant information, cluster merging is needed based on the overlapping ratio of clusters. If the merging condition is set as $\Theta_2 = 2/3$, then the final clustering results are {1, 2, 3, 4}, {1, 5, 6, 7, 8, 9}, and {9, 10}. Similarly, if $\Theta_2 = 1/2$, the final clustering results are {1, 2, 3, 4, 5, 6, 7} and {1, 5, 6, 7, 8, 9, 10}.

Through the above analysis, the final result is shown in Figure 4.

3. Experiments

The parameters used in the experiment are shown in Table 4.

The datasets used throughout the experiments are as follows: MovieLens (http://www.datatang.com/datares/detail.aspx?id=44295), Zachary's Karate Club (http://www-personal.umich.edu/~mejn/netdata/), and Dolphin's Associations (http://www-personal.umich.edu/~mejn/netdata/):

(1) The dataset of MovieLens is a synthesized recommendation system and virtual community, which is commonly used for social computing.

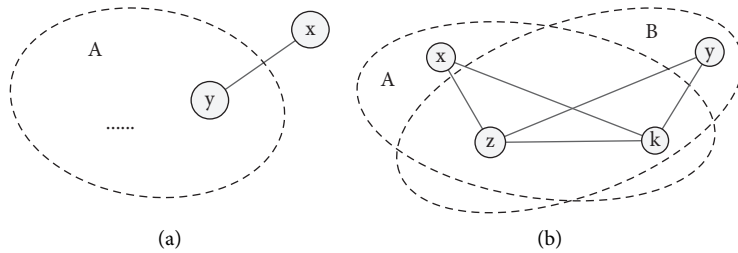


Figure 2: Extreme cases of link-partitions. (a) The lowest degree and (b) the highest degree.

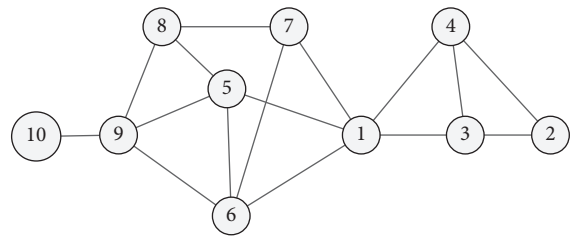


Figure 3: The network topology.

Table 1: The list of nodes.

①	{3, 4, 5, 6, 7}
②	{3, 4}
③	{1, 2, 4}
④	{1, 2, 3}
⑤	{1, 6, 8, 9}
⑥	{1, 5, 7, 9}
⑦	{1, 6, 8}
⑧	{5, 7, 9}
⑨	{5, 6, 8, 10}
⑩	{9}

Table 2: The attribute list of edges.

①③	{1, 2, 3, 4, 5, 6, 7}
①④	{1, 2, 3, 4, 5, 6, 7}
①⑤	{1, 3, 4, 5, 6, 7, 8, 9}
①⑥	{1, 3, 4, 5, 6, 7, 9}
①⑦	{1, 3, 4, 5, 6, 7, 8}
②③	{1, 2, 3, 4}
②④	{1, 2, 3, 4}
③④	{1, 2, 3, 4}
⑤⑥	{1, 5, 6, 7, 8, 9}
⑤⑧	{1, 5, 6, 7, 8, 9}
⑤⑨	{1, 5, 6, 8, 9, 10}
⑥⑦	{1, 5, 6, 7, 8, 9}
⑥⑨	{1, 5, 6, 7, 8, 9, 10}
⑦⑧	{1, 5, 6, 7, 8, 9}
⑧⑨	{5, 6, 7, 8, 9, 10}
⑨⑩	{5, 6, 8, 9, 10}

(2) The dataset of Zachary's Karate Club is a social network of friendships between 34 members, so edges in the graph describe the higher frequency of interactions between members.

(3) The dataset of Dolphin's Associations is an undirected social network of frequent associations between 62 dolphins, which has 62 nodes and 159 edges.

3.1. Comparison Methods

(1) INB [11] is a method based on the analysis of induced dependencies to build an inference network.

(2) RNM [18] is a local expansion method based on rough neighbourhood.

(3) NOVER-based [19] is a greedy algorithm, which iteratively removes the edges of a network in the increasing order of their neighborhood overlap and calculates the modularity score of the resulting network components after the removal of each edge.

In Figure 5, different colours of nodes represent different clusters. The clustering result of the dataset of Zachary's Karate Club is shown in Figure 5(a) The overlapping nodes ascertained by LFM, RNM, and BCluster are <3, 9, 10, 14, 31>, <3, 9, 10, 31>, and <3, 14>. It can be discovered that nodes 9, 10, and 31 have every reason to be assigned to a unique cluster rather than an overlapping node, for which reason, BCluster has a strong ability of community partition and generates the minimum quantity of overlapping nodes. Therefore, BCluster can give a precise description of a node. Node 9 is similar to Node 14 (both of them have many connections with other clusters), while BCluster defines Node 14 as an overlapping node, and Node 9 is assigned to one cluster, the result of which is that Node 9 has a weak association with other clusters because it is only linked by some overlapping nodes, and Node 14 is linked directly to Node 34 (the centre of the community), which makes Node 14 closely connected with the two clusters. As a result, Node 14 is difficult to distinguish.

Table 3: Temporary link clustering results, $\Theta_1 = 1$.

	$c-1$	$c-2$	$c-3$	$c-4$	$c-5$	$c-6$	$c-7$	$c-8$	$c-9$	$c-10$
Edge	{①③} {①④}	{①⑤}	{①⑥}	{①⑦}	{②③} {②④} {③④}	{⑤⑥} {⑤⑧} {⑥⑦} {⑦⑧}	{⑤⑨}	{⑥⑨}	{⑧⑨}	{⑨⑩}
Sets of nodes	①③④	①⑤	①⑥	①⑦	②③④	⑤⑥⑦⑧	⑤⑨	⑥⑨	⑧⑨	⑨⑩

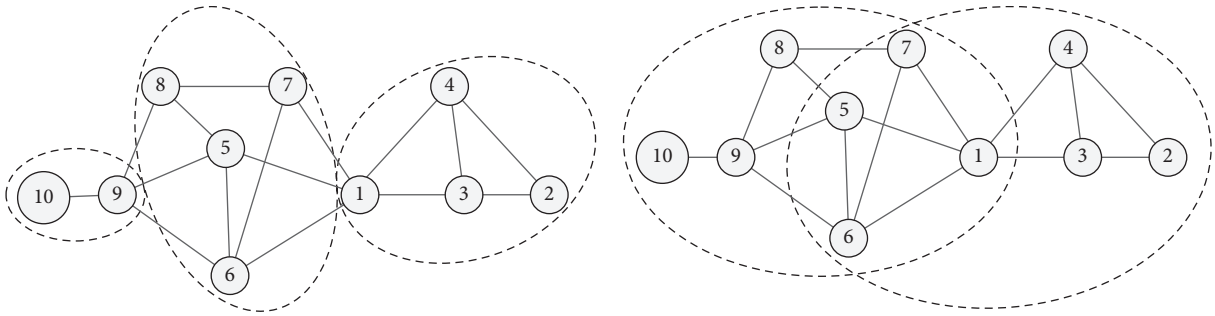


Figure 4: Results of link clustering. (a) $\Theta_1 = 1$, $\Theta_2 = 2/3$ and (b) $\Theta_1 = 1$, $\Theta_2 = 1/2$.

Table 4: Parameter description.

Θ_1	The threshold of function overlap()
Θ_2	The threshold of cover()
MP	Multiplicity precision
MR	Multiplicity recall
FB	Synthesis of P and R

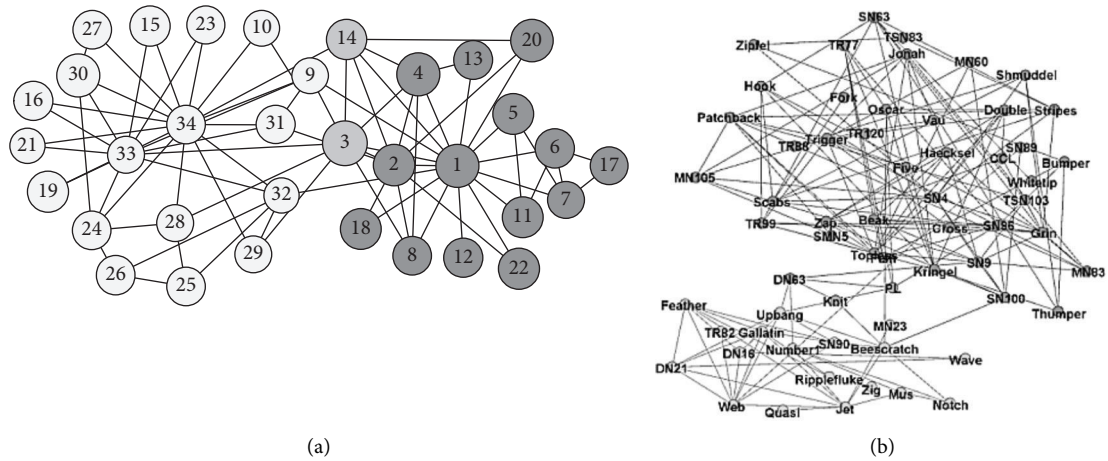


Figure 5: The execution results of BCluster on different datasets. (a) Zachary's Karate Club. $\Theta_1 = 0.85$, $\Theta_2 = 0.5$. (b) Dolphin's associations. $\Theta_1 = 0.85$, $\Theta_2 = 0.5$.

The performance of BCluster on the dataset of Dolphin's Associations is shown in Figure 5(b), and the overlapping nodes' IDs are 8, 20, and 31, while LFM finds that nodes 8, 20, 29, 31, and 40 overlap. By analyzing the edges of each overlapping node that is generated, BCluster is correct, generates the lowest number of overlapping nodes, and is used to achieve the optimal community partition.

For the purpose of analyzing the experimental results, the following measurement parameters are used: multiplicity precision calculated by [20]: $MP = \mathrm{Min}(|C(e) \cap C(e')|, |L(e) \cap L(e')|)/|C(e) \cap C(e')|$ and multiplicity recall by $MR = \mathrm{Min}(|C(e) \cap C(e')|, |L(e) \cap L(e')|)/|L(e) \cap L(e')|$. FB is a comprehensive measure of MP and MR, and the algorithm is

Table 5: The performance comparisons of different algorithms.

Dataset	INB			NOVER-based			RNM			BCluster		
	MR	MP	FB	MR	MP	FB	MR	MP	FB	MR	MP	FB
Karate Club	0.70	0.92	0.79	0.83	0.91	0.87	0.84	**1.00**	0.91	**1.00**	**1.00**	**1.00**
Dolphin	0.37	0.90	0.53	0.73	**0.90**	0.81	0.46	0.97	0.62	0.73	**0.98**	0.83
MovieLens	0.33	0.83	0.47	**0.84**	0.80	0.82	0.56	0.86	0.68	0.82	**0.89**	**0.85**

Table 6: The comparisons of BCluster in different datasets.

	$K-1$	$K-2$	$D-1$	$D-2$	$M-1$	$M-2$	$M-3$
Degree distribution	3.571	3.889	4.222	5.220	11.61	25.924	24.372
Diameter	3	3	5	5	5	5	5
Density	0.275	0.229	0.248	0.130	0.002	0.005	0.005
Modularity	0.275	0.240	0.217	0.383	0.503	0.475	0.491
Average clustering coefficient	0.736	0.690	0.470	0.279	0	0	0
Average path length	1.813	1.900	2.190	2.487	2.518	2.527	2.514
Ratio (%)	47.06	64.71	37.70	78.69	5.99	47.06	49.94

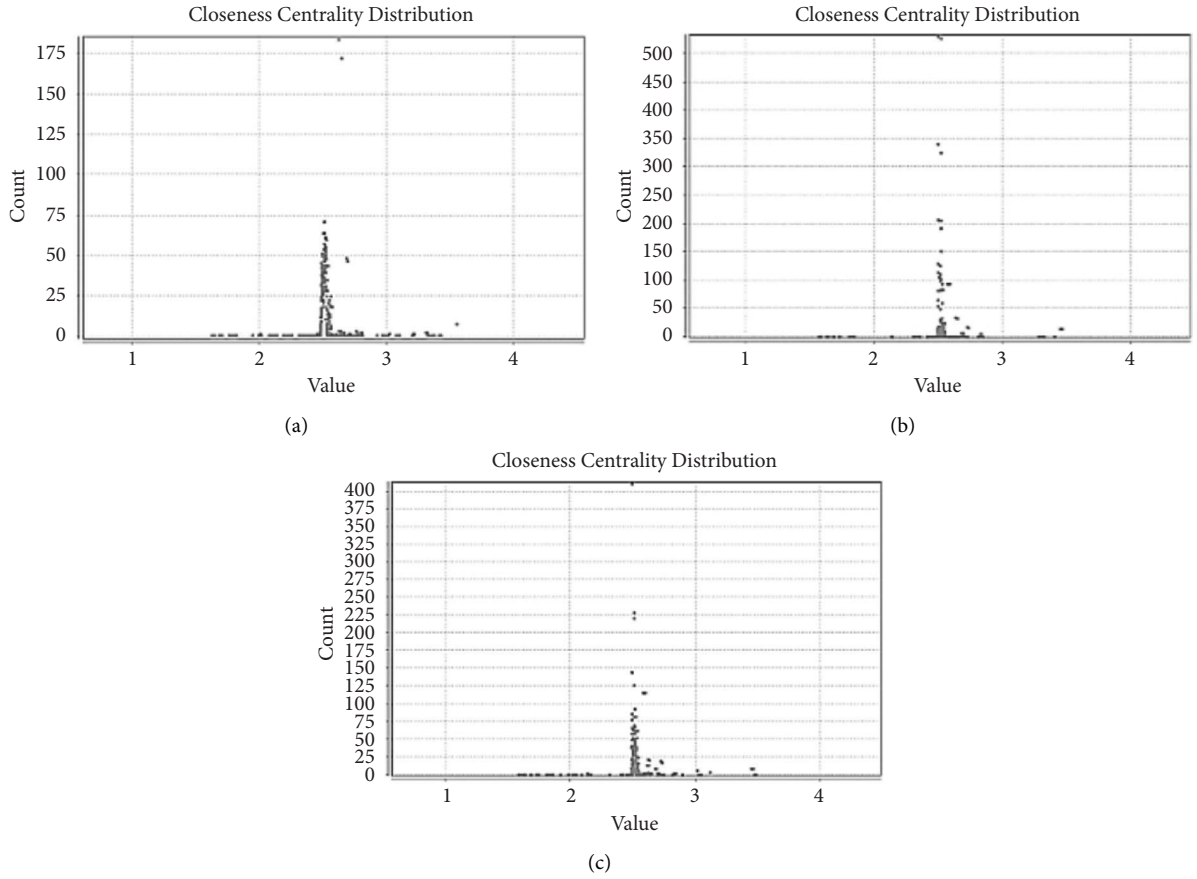
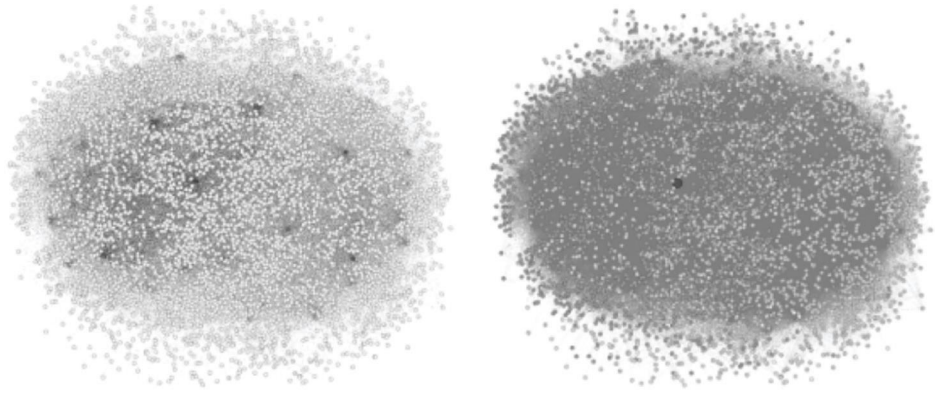


Figure 6: Closeness centrality distribution of (a) $M-1$, (b) $M-2$, and (c) $M-3$.

$FB = MP \times MR \times 2/(MP + MR)$. Table 5 shows a comparison of the performances of different clustering algorithms on different datasets. We manually labeled 20,000 rows of node information that was randomly selected from MovieLens because the dataset has no tags and unable to perform a comparison of correctness.

The two clusters of Karate Club separated by BCluster are denoted as $K-1$ and $K-2$. The two separated clusters of



(a) (b)

Figure 7: Feature analysis of nodes in $M-2$. (a) The distribution of the centre and the overlapping nodes. (b) The heat map of the center node.

Dolphin are $D-1$ and $D-2$. The three separated clusters of MovieLens are $M-1$, $M-2$, and $M-3$. On the basis of different measurements, the comparisons of BCluster in different datasets are shown in Table 6.

On the basis of three different datasets, experiments are conducted to compare BCluster with INB, IOP, and RNM. The results show that BCluster generates the least number of overlapping nodes and achieves a good community partition.

The quality of the communities can be measured. The results of it are shown in Figure 6, which indicates that the distances between nodes are reasonable and comparatively short among most of the nodes in the same cluster. Figure 6 illustrates that the clustering effect of our overlapping clustering algorithm is good.

Figure 7 displays the reasonability of the centre nodes and overlapping nodes in $M-2$. The black nodes denote the centre in Figure 7(a). Because BCluster is performed through a series of computations on the collections of nodes, there may be more than one centre node per cluster, and the centre nodes are not only associated with one another strongly but also closely linked to many ordinary nodes in the corresponding cluster. Therefore, the centre nodes make the nodes in the cluster highly correlated. Figure 7(b) is a heat map of the nodes in $M-2$, the black node is the centre node, and the colours from dark to light represent the descending degrees of connection between the centre and other nodes. Therefore, it can be concluded that the centre is strongly positively correlated to many nodes in the cluster, based on which, several centre nodes make the nodes in the cluster gathered closely.

4. Conclusions

To solve the data sparseness problem of a single object, the authors of this paper propose an algorithm that calculates the probability of linking any two reachable and indirectly linked nodes and adds edges of high probability into the original graph to increase the quantity of knowledge. An overlapping clustering algorithm that determines the attributes of edges based on the two nodes that are linked and classifies edges into non-overlapping clusters is also presented. Compared to other clustering algorithms, BCluster can accurately distinguish the differences between nodes and detect communities with minimal numbers of overlapping nodes. In future studies, we will focus on how to improve the accuracy of community division, how to subdivide users' multiple identities, how to reduce the complexity of massive node processing algorithm, and other research areas.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Youth Program of The National Social Science Fund of China (Project name: Research on Online Behavior Pattern of Customers and Multidimensional Customer Insight Method under Big Data; Grant No. 19CGL024).

References

[1] J. Cao, S. Wang, and H. Wang, "Detecting communities on topic of transportation with sparse crowd annotations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 1017–1022, 2017.

[2] T. Chakraborty, S. Kumar, N. Ganguly, A. Mukherjee, and S. Bhowmick, "GenPerm: a unified method for detecting non-overlapping and overlapping communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 8, pp. 2101–2114, 2016.

[3] F. Dabaghi Zarandi and M. Kuchaki Rafsanjani, "Community detection in complex networks using structural similarity,"

Physica A: Statistical Mechanics and Its Applications, vol. 503, pp. 882–891, 2018.

[4] R. Fei, S. Li, Q. Xu, B. Hu, and Y. Tang, "The multi-dimensional information fusion community discovery based on topological potential," *IEEE Access*, vol. 8, pp. 3224–3239, 2020.

[5] W. Luo, D. Zhang, H. Jiang, L. Ni, and Y. Hu, "Local community detection with the dynamic membership function," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 5, pp. 3136–3150, 2018.

[6] J. Xiang, Z. Z. Wang, H. J. Li et al., "Community detection based on significance optimization in complex networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 5, Article ID 053213, 2017.

[7] Y. Yuan, D. W. Soh, H. H. Yang, and T. Q. S. Quek, "Learning overlapping community-based networks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 4, pp. 684–697, 2019.

[8] A. C. Gabardo, R. Berretta, and N. J. D. Vries, "Where does my brand end? An overlapping community approach," *Intelligent and Evolutionary Systems*, vol. 8, pp. 133–148, 2017.

[9] T. H. T. Nguyen, D. T. Dinh, S. Sriboonchitta, and V. N. Huynh, "A method for K-means-like clustering of categorical data," *Journal of Ambient Intelligence and Humanized Computing*, vol. 3, 2019.

[10] L. Guo, W. Zuo, T. Peng, and B. K. Adhikari, "Attribute-based edge bundling for visualizing social networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 438, pp. 48–55, 2015.

[11] L. Guo, W. Zuo, and T. Peng, "Inference network building and movements prediction based on analysis of induced dependencies," *IET Software*, vol. 11, pp. 12–17, 2017.

[12] P. Wagenseller, F. Wang, and W. Wu, "Size matters: a comparative analysis of community detection algorithms," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 4, pp. 951–960, 2018.

[13] R. Interdonato, A. Tagarelli, D. Ienco, A. Sallaberry, and P. Poncelet, "Local community detection in multilayer networks," *Data Mining and Knowledge Discovery*, vol. 31, no. 5, pp. 1444–1479, 2017.

[14] K. Berahmand, M. Mohammadi, A. Faroughi, and R. P. Mohammadiani, "A novel method of spectral clustering in attributed networks by constructing parameter-free affinity matrix," *Cluster Computing*, vol. 25, no. 2, pp. 869–888, 2022.

[15] R. Fathi, A. R. Molla, and G. Pandurangan, "Efficient distributed community detection in the stochastic block model," in *Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, IEEE, Dallas, TX, USA, July 2019.

[16] K. Pawan and D. Ravins, "A neighborhood proximity based algorithm for overlapping community structure detection in weighted networks," *Frontiers of Computer Science*, vol. 13, no. 6, pp. 2078–2086, 2019.

[17] B. Ball, B. Karrer, and M. E. J. Newman, "Efficient and principled method for detecting communities in networks," *Physical Review A*, vol. 84, no. 3, Article ID 036103, 2011.

[18] Z. H. Zhang, D. Q. Miao, and J. Qian, "Detecting overlapping communities with heuristic expansion method based on rough neighborhood," *Chinese Journal of Computers*, vol. 36, no. 10, pp. 2078–2086, 2014.

[19] P. Kim and S. Kim, "Detecting community structure in complex networks using an interaction optimization process," *Physica A: Statistical Mechanics and Its Applications*, vol. 465, pp. 525–542, 2017.

[20] N. Meghanathan, "A greedy algorithm for neighborhood overlap-based community detection," *Algorithms*, vol. 9, no. 1, p. 8, 2016.