

Article

An Influence-Based Label Propagation Algorithm for Overlapping Community Detection

Hao Xu ¹, Yuan Ran ², Junqian Xing ³ and Li Tao ^{1,*}¹ College of Computer and Information Science, Southwest University, Chongqing 400700, China² Hanhong College, Southwest University, Chongqing 400700, China³ Faculty of Engineering, The University of Sydney, Sydney, NSW 2006, Australia

* Correspondence: tli@swu.edu.cn

Abstract: Of the various characteristics of network structure, the community structure has received the most research attention. In social networks, communities are divided into overlapping communities and disjoint communities. The former are closer to the actual situation of real society than the latter, making it necessary to explore a more effective overlapping community detection algorithm. The label propagation algorithm (LPA) has been widely used in large-scale data owing to its low time cost. In the traditional LPA, all of the nodes are regarded as equivalent relationships. In this case, unreliable nodes reduce the accuracy of label propagation. To solve this problem, we propose the influence-based community overlap propagation algorithm (INF-COPRA) for ranking the influence of nodes and labels. To control the propagation process and prevent error propagation, the algorithm only provides influential nodes with labels in the initialization phase, and those labels with high influence are preferred in the propagation process. Lastly, the accuracy of INF-COPRA and existing algorithms is compared on benchmark networks and real networks. The experimental results show that the INF-COPRA algorithm significantly improves the extended modularity (EQ) and normal mutual information (NMI) of the community, indicating that it can outperform state-of-art methods in overlapping community detection tasks.

Keywords: complex network; community detection; overlapping community; label propagation

MSC: 68R10



Citation: Xu, H.; Ran, Y.; Xing, J.; Tao, L. An Influence-Based Label Propagation Algorithm for Overlapping Community Detection. *Mathematics* **2023**, *11*, 2133. <https://doi.org/10.3390/math11092133>

Academic Editor: Carlo Bianca

Received: 28 February 2023

Revised: 12 April 2023

Accepted: 28 April 2023

Published: 2 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Community detection is a basic task in complex network analysis. In complex networks, the community structure is defined as a group in the network [1,2]. The vertices within the same community are closely connected, while those between different communities are sparse. Community detection is widely used in different fields, and plays an important role in network structure analysis. For example, detecting the community structure of social networks [3–5] can help users to locate groups with similar interests. In bioinformatics [6], community detection can reveal the interaction patterns among proteins or genes. Moreover, in infectious disease prevention [7,8] community detection can provide a better understanding of infectious disease mechanisms and scope. Thus, detecting the community structure is pivotal for mining the significance of data generated by interpersonal interactions in social networks, strengthening relationships between individuals, and achieving more accurate data classification.

Researchers around the world have proposed many methods to perform the task of community detection, including optimization-based methods and heuristic algorithms. Optimization-based methods optimize the modularity [9,10] or likelihood function [11] through the design of a specific objective. Currently, search methods based on biological evolution (e.g., ant colony optimization [12], genetic algorithms [13], etc.) are widely utilized to optimize the designed objective. Unlike optimization-based methods, heuristic

methods assume that the formation of community structure is dynamic and can be described by a generation model (e.g., Markov chain [14]), and that in general the community structure can be obtained by inputting the network into generation models.

However, with the growth of network scale, network topology becomes more complex. To detect communities, there are several problems that must be resolved. The majority of traditional community detection algorithms use the number of communities as a known parameter or obtain the number of communities with the assistance of the maximum modularity [15]. However, community detection algorithms based on modularity make it easy to disregard small-scale communities, which can affect the accuracy of the algorithm to an extent. Further, the structure of overlapping communities is complex. Several practices [16] have shown that the interaction between users of a real network is highly complex, which accounts for overlapping communities. For example, in a social context, a particular person may be deeply connected to their family while being related to coworkers and sports club members as well. From the perspective of network science, the organization of nodes into different groups should be considered by community detection methods [17]. The existing overlapping community detection algorithm is usually proposed in a specific environment, and cannot be applied to different network structures [18]. Lastly, the time expenses of the algorithm are large. In the era of big data, the scale of social networks has gradually increased. Community detection in large-scale network calls for high algorithmic efficiency. Traditional community detection algorithms cannot effectively mine large-scale networks, making it urgent to develop low-complexity algorithms.

Label propagation [19] is a method that only uses the network topology structure to discover communities, and often shows better performance compared to other algorithms when processing large-scale data [20,21]. Therefore, the label propagation algorithm (LPA) has gained the attention of many researchers over the past decade. The existing research [22,23] has covered optimization of the LPA to enhance its robustness and stability. Despite this, two problems limit its stability and effectiveness. First, in the label initialization phase the algorithm treats all of the nodes equally and assigns labels to all of the nodes. This practice requires a lot of time, and introduces errors when propagating labels. Second, in the label propagation stage, multiple alternative labels exist and the algorithm randomly selects from them. This leads to instability of the algorithm and difficulty in finding the optimal solution. To solve the instability of label propagation and improve its accuracy, we propose the influence-based community overlap propagation algorithm (INF-COPRA).

INF-COPRA uses weights to describe every node and label of the network to consider its importance. The main idea of INF-COPRA is as follows. In the initialization phase, we construct a node influence function; nodes with an influence greater than a certain threshold are labeled to control the propagation sequence and prevent error propagation. In the label propagation stage, we construct a label influence function and select labels with greater influence according to a certain order, where the function comprehensively considers the importance of the labels and their corresponding nodes. By reducing the randomness of the algorithm, we optimize the performance of the algorithm. Finally, the results on Lancichinetti–Fortunato–Radicchi (LFR) benchmark networks and five real networks show that our method effectively improves the accuracy and stability of community detection, especially in large-scale complex networks.

The rest of this paper is organized as follows. In Section 2, we introduce related works about community detection algorithms, including both disjoint and overlapping community detection algorithms. In light of the shortcomings of current community detection algorithms, we propose our INF-COPRA algorithm in Section 3. In Section 4, we present experiments on validation datasets to prove the accuracy of our algorithm. Lastly, Section 5 provides the conclusions of this paper and the future outlook for additional research.

2. Related Work

In this section, we review the existing research on community detection. Based on different detection targets, community detection algorithms can be divided into disjoint

community detection and overlapping community detection algorithms [24]. Thus, we summarize the existing research from the two aspects of disjoint community detection and overlapping community detection.

2.1. Disjoint Community Detection Algorithm

The purpose of a disjoint community detection algorithm is to divide the nodes in a complex network into multiple independent sets. Any two sets do not intersect, and any node can only belong to one community, which is the initial form of community detection. Common disjoint community detection algorithms can be primarily divided into sub-graph segmentation, hierarchical clustering, the modularity optimization algorithm, the bionic computing method, and heuristic methods. Below, we review several of the classical algorithms.

The graph segmentation method is derived from task allocation in parallel computing. The first method proposed to solve the problem of community mining was the Kernighan–Lin (KL) [25] algorithm, which sets the size of two communities in advance and gradually adjusts the distribution of nodes in the community to make the target function of the algorithm reach the maximum or approximate maximum. However, this method cannot determine the stopping condition of the algorithm when the number of communities is unknown. In the community detection algorithm based on hierarchical clustering [26], a similarity measure is defined to quantify the similarity between nodes. Common measures include cosine similarity and the Jaccard index. Different nodes are grouped into corresponding communities based on their similarity. The modularity optimization algorithm [27] is one of the most widely used methods in community detection. Modularity is used to measure the quality of the division of specific communities in the network. By searching the possible communities in the network, it is possible to locate the partition with the largest modularity. Owing to the high time complexity of the exhaustive search, approximate optimization methods, including greedy algorithms and simulated annealing, are often used in practice. For example, the Louvain algorithm [28], among the most classic modularity optimization algorithms today, iteratively optimizes the local community until it can no longer improve the global modularity.

2.2. Overlapping Community Detection Algorithm

Because communities often overlap in real networks, overlapping community discovery [29] is gradually receiving more research attention. The classic research on overlapping communities [30] includes algorithms based on clique percolation, label propagation, local spectral clustering, random walks, and local optimization.

The clique percolation method (CPM) algorithm [31] is an early effective algorithm for detecting overlapping communities; in this method, a clique is a complete sub-graph existing in the network, that is, a set of any two connected nodes. Based on the assumption that multiple cliques overlap to form a community, the community is detected by searching for adjacent cliques. Zhang et al. [32] used non-negative matrix factorization (NMF) to solve the problem of community mining. This algorithm can calculate the membership of nodes with respect to every community. However, the community parameters of this method are based on modularity optimization, and the calculation cost of the algorithm is large. Algorithms based on a random walk [33] focus on the time required for the random walk to escape from the community. Because a random walk should be trapped in a community for a long time, it is possible to find communities by maximizing the random walk within the same community. Algorithms [34,35] based on local optimization seek to optimize the fitness function describing the local topological structure in order to achieve a near-optimal solution. Reducing the problem of community mining in social networks to a local optimization problem represents a convincing solution. This is because the process of forming associations is initiated by the participants in a localized way, making this approach very intuitive. The ClusterONE algorithm [36] continuously selects seeds and uses the yield function to find the local optimal cluster around the seeds. The COPRA

algorithm [19,37] modifies the classic LPA algorithm, ensuring that every node can retain multiple labels to find overlapping community structures. In the SLPA algorithm [38], SLPA provides storage for every node to store the received information, that is, the labels. The probability of receiving labels stored by nodes is interpreted as membership strength. SLPA does not require the number of communities at the beginning of the algorithm, allowing it to be determined based on the label clustering in the network.

In summary, the existing overlapping community detection algorithms are either unstable or not suitable for large-scale networks [39]. By contrast, LPA only uses the network topology to detect the community in the network. It requires neither prior information about the community nor a predefined fitness function [19]. Thus, it has often shown better performance than other algorithms when processing large-scale data [40]. In order to design an overlapping community detection algorithm suitable for large-scale data, our proposed approach integrates the ideas of label propagation and an influence mechanism, resulting in an improved model by controlling the accuracy of label propagation in the network.

2.3. LPA for Overlapping Community Detection

Because this paper focuses on optimizing LPA for overlapping community detection, we introduce the flow of the LPA in this section. The problem of community detection in the network using LPA can be transformed into the problem of finding nodes with the same label in the network. Nodes with the same label are considered to belong to the same community. The procedure of the overlapping community detection algorithm based on label propagation is as follows:

1. Label initialization; given a network $G = (V, E)$ with $|V|$ nodes, for any node $v \in V$ in the network, assign a separate label $c(v)$ to every v , with the membership of node v to label $c(v)$ being 1;
2. Update the label of node v based on the neighbors (Formula (1)). If there are multiple labels to choose from, randomly select γ labels from the alternative labels. The parameter γ is the maximum number of labels that a node can have;
3. If the algorithm reaches the set maximum number of iterations, the algorithm stops; otherwise, continue to step 2;
4. The community is divided based on the node label.

Here, $b_t(c, v)$ represents the membership of label c with respect to node v in the t -th iteration. $b_t = (c, v)$ represents the subordinate coefficient of node v to community c . The calculation formula of $b_t(c, v)$ is as follows:

$$b_t(c, v) = \frac{\sum_{y \in N(v)} b_{t-1}(c, y)}{|N(v)|}, \quad (1)$$

where $\sum_{y \in N(v)} b_{t-1}(c, y)$ indicates the sum of the subordinate coefficients of label c for the neighbors of node v during the $t - 1$ iteration, while $N(v)$ represents the neighbors of node v , and $|N(v)|$ is the number of those neighbors.

Because of the explosive growth of data scale, community detection algorithms based on label propagation have received more research attention lately. To improve the stability and robustness of the LPA, researchers have made various modifications to it. However, such methods rely on random selection, which can limit the accuracy of the detected communities. To address this issue, we propose INF-COPRA in this paper. Our algorithm utilizes node and label sorting techniques to enhance the accuracy of the algorithm while preserving the low computational complexity of label propagation. By incorporating a novel initialization method and a deterministic label updating rule, our algorithm can achieve higher accuracy in identifying overlapping communities compared with existing label propagation-based methods.

3. Influence-Based Community Overlap Propagation Algorithm

In this section, we propose INF-COPRA. We optimize the initialization phase and label propagation phase of the COPRA algorithm. Next, we introduce the process of INF-COPRA in detail.

The present section details the INF-COPRA algorithm, as depicted in Figure 1. By identifying important nodes in the network as community centers and spreading from these central nodes to their neighbors, it is possible to calculate the influence of different communities on every node. When the influence exceeds a certain threshold, it is determined that a particular node belongs to a particular community.

Therefore, the first step involves computing the influence of every node using the prescribed Formula (2), after which the nodes are arranged in descending order of influence and assigned diverse labels. Labels are assigned exclusively to nodes with an influence exceeding the mean influence. In the subsequent label propagation stage, the label of a given node is updated iteratively while considering its neighbors. In cases where there are multiple potential labels, the label with the highest label influence is selected using Formula (3). Upon satisfaction of the termination condition, the algorithm determines the community structure based on the labels assigned to the nodes.

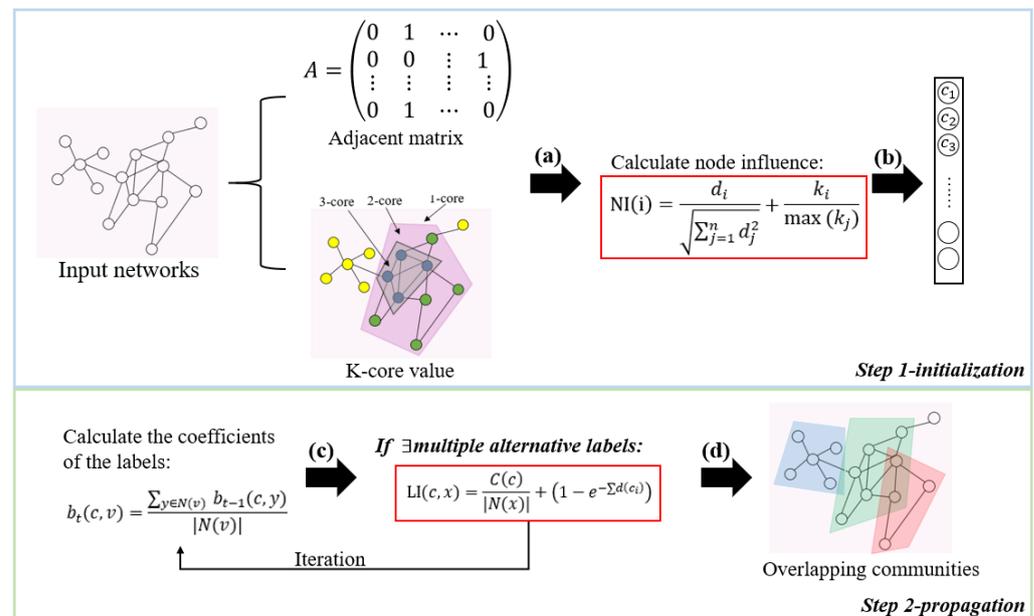


Figure 1. Flowchart of INF-COPRA, which consists of (a) calculate the influence of nodes through local and location information, (b) sort the influence of nodes and assign labels to nodes with high influence, (c) update the labels through the labels of neighbors, (d) after the iteration, the community is divided based on the labels carried by the nodes.

Next, we introduce the initialization and label propagation phases of INF-COPRA.

3.1. Initialization in INF-COPRA

In the initialization phase of the COPRA algorithm, it is customary to consider all of the unlabeled samples as equivalent and later transmit label information to all of the neighbors without considering its reliability [29]. However, this approach may result in the propagation of inaccurate labels, mainly owing to unreliable samples, consequently leading to a considerable reduction in the clustering accuracy. To address this issue, we propose the integration of an influence function that calculates the influence of every node, and set a threshold accordingly. Specifically, we label only the influential nodes during the initialization stage, thereby controlling the propagation sequence and preventing the

spread of erroneous labels. This approach enhances the accuracy of the COPRA algorithm and effectively mitigates the impact of unreliable samples on the clustering process.

The influence of nodes is mainly contingent upon local, global, and positional information. The computation of global information has high time complexity and is unsuitable for large networks, as previously noted in the literature [41]. Conversely, Kitsak et al. [42] posited that while local information is computationally efficient, it is not a definitive factor affecting nodes' propagation ability. Hence, in order to better elucidate the propagation performance of nodes this paper employs an influence calculation technique based on the K-core value and degree. This method is more suitable for assessing nodes' propagation performance.

K-core decomposition is a widely employed method in network analysis. It involves the iterative elimination of nodes with a degree of 1 and their corresponding edges in the network until all of the nodes in the network possess a degree greater than 1 and the K-core value of the removed nodes is 1. Progressively eliminating all of the nodes in the network culminates in the K-core values of all of the nodes in the network. While the K-core value can provide a rough approximation of the propagation ability of a network, it may not be comprehensive enough. Thus, this paper adopts an influence calculation technique that comprehensively integrates the "degree" and "K-core" of nodes, which respectively capture their local information and positional information. Together, these two metrics serve as the standard for accurately assessing node influence and enable precise community division. The influence calculation function for node i is as defined in Formula (2):

$$NI(i) = \frac{d_i}{\sqrt{\sum_{j=1}^n d_j^2}} + \frac{k_i}{\max(k_j)}, \tag{2}$$

where d_i is the degree of node i , d_j represents the degree of the neighbors of node i , n represents the number of neighbors of node i , k_i represents the K-core value of node i , and $\max(k_j)$ represents the maximum K-core value of the neighbor of node j . In our formula, a node is more important when it has a greater degree than its neighbor. When a node is at the core location of the network and its neighbor node is at the edge, its influence is greater.

During the initialization phase, we initially calculate every node's influence by applying Formula (2), sort the nodes based on their influence, and allocate distinct labels to nodes with influence exceeding the average value. Unlike the conventional LPA practice of assigning labels to all of the nodes, our proposed technique has the advantage of mitigating time complexity during the initialization phase and preventing the propagation of erroneous labels through unreliable nodes. The algorithm in the initialization phase is presented below as Algorithm 1.

Algorithm 1 Initialization

Input: network $G = (V, E)$, V is the node set in the network, E is the edge set in the network

Output: node influence sequence dc

- 1: **if** node i is the neighbor of node j **then**
 - 2: $A_{ij}=1$
 - 3: **else**
 - 4: $A_{ij}=0$
 - 5: **for** each node v_i **do**
 - 6: $NI(i) = \frac{d_i}{\sqrt{\sum_{j=1}^n d_j^2}} + \frac{k_i}{\max(k_j)}$
 - 7: dx =order node by $NI(i)$
 - 8: $dc = dx(NI(i) > \text{avg}(\sum NI(i)))$
 - 9: **return** dc
-

3.2. Label Propagation in INF-COPRA

INF-COPRA adopts a synchronous update strategy; that is, a node selects a label based on the node's membership to the label and retains labels with membership exceeding a given threshold. Because the label represents a community in the network, a node's membership with respect to a label represents its membership in the community, and is collectively referred to as the node's membership. During COPRA, nodes only preserve labels with a membership above a certain threshold. However, if multiple alternative labels are present then the COPRA algorithm randomly selects one, leading to poor stability and difficulty in identifying optimal labels. To address these issues, we introduce the concept of the comprehensive label influence. The factors determining comprehensive label influence include the label subordinate coefficient and the sum of node degrees. By considering the order of influence, we select labels with greater influence to reduce algorithmic randomness. We propose Formula (3) for computing label influence, which comprehensively considers the impact on the label selection process considering label influence factors:

$$LI(c, v) = \frac{|N_c(v)|}{|N(v)|} + \left(1 - e^{-\sum_{i \in N_c(v)} d(i)}\right), \quad (3)$$

where $LI(c, v)$ represents the influence of label c on node v , $|N(v)|$ represents the number of neighbors of node v , and $|N_c(v)|$ represents the number of neighbors that carry the label c . Furthermore, $\sum_{i \in N_c(v)} d(i)$ denotes the sum of node degree that have been assigned the label c .

The label influence comprehensively considers the importance of the label and its location. In this paper, the frequency of label appearance is regarded as the importance of a label; the more frequently a label appears, the more important it is. In addition, the labels located at different nodes have different probabilities of being propagated to other nodes; thus, in this paper we take the degree of the node where the label is located as the location importance of the node. The label propagation stage is described in Algorithm 2.

Algorithm 2 Propagation

Input: network $G = (V, E)$, threshold ρ

Output: label list ls

```

1: for each node  $v$  do
2:    $LI(c, v) = \frac{|N_c(v)|}{|N(v)|} + \left(1 - e^{-\sum_{i \in N_c(v)} d(i)}\right)$ 
3:   if  $LI(c, v) < \rho$  then
4:      $ls.delete(c)$ 
5:   else
6:      $ls.add(c)$ 
7: return  $ls$ 

```

4. Experiments

4.1. Datasets

To verify the effectiveness of the algorithm, we selected five real networks with different scales and LFR artificial networks with different characteristics to serve as experimental data. The five real networks were the Karate Club [43], Dolphins [44], American University Football Game [24], Facebook User, and Enron E-Mail Communication Network [45] databases. The statistical properties of the networks are shown in Table 1.

To generate synthetic networks, we utilized the LFR benchmark network model [46]. The parameter settings for the LFR benchmark network are provided in Table 2. By manipulating the mixing parameters and overlapping density, we generated a diverse set of artificial networks while keeping the other parameters at their default values. Note that the LFR benchmark network assumes that the degree of nodes and the size of communities in the network follow power-law distributions with power-law distribution indices of t_1 and t_2 . The mixing parameter, denoted as μ , ranges from 0 to 1, and represents the proportion of

connections between different community structures within the same network, while C_{min} represents the number of nodes in the smallest community, C_{max} represents the number of nodes in the largest community, O_n represents the proportion of overlapping nodes in the total number of nodes in the network, and O_m represents the number of communities to which overlapping nodes belong. Through the adjustment of these parameters, we could generate networks with distinct and varying characteristics.

Table 1. Statistical characteristics of real network datasets.

	Node	Edge
karate	34	78
dolphins	62	159
football	115	613
Facebook	4039	88,234
Email	36,692	183,831

Table 2. Parameter settings of LFR.

Network	N	μ	O_n
LFR- μ	1000–8000	0.1	-
		0.2	-
		0.3	-
		0.4	-
		0.5	-
		0.6	-
LFR- O_n	1000–8000	-	0.1
		-	0.2
		-	0.4
		-	0.6
		-	0.8
		-	-

4.2. Evaluation Metrics

In the context of real networks, standard community divisions are frequently unavailable. Therefore, modularity was utilized as the primary evaluation index to assess the performance of the algorithm. Conversely, for synthetic networks that possess standard community divisions, normalized mutual information (NMI) was utilized to evaluate the similarity between the communities detected by the algorithm and the actual communities. In the subsequent sections we provide detailed explanations of these two evaluation metrics.

NMI was introduced by Danon et al. [47], and is commonly used to evaluate the accuracy of community detection algorithms when analyzing networks with a known community structure. Mutual information measures the similarity between the communities discovered by the algorithm and the actual community structure. The resulting value is confined to the range of [0–1], where a value closer to 1 indicates a higher degree of similarity between the two community structures, indicating more effective algorithmic performance. The calculation of NMI is presented as follows:

$$NMI(P, P') = \frac{-2 \sum_{i=1}^{|P|} \sum_{j=1}^{|P'|} n_{ij} \log \left(\frac{n_{ij} \cdot n}{n_i^P \cdot n_j^{P'}} \right)}{\sum_{i=1}^{|P|} n_i^P \log \left(\frac{n_i^P}{n} \right) + \sum_{j=1}^{|P'|} n_j^{P'} \log \left(\frac{n_j^{P'}}{n} \right)}, \tag{4}$$

where P represents the community structure detected by the algorithm, P' represents the real community structure of the network, $|P|$ represents the number of detected communities, $|P'|$ represents the number of real communities, n_{ij} represents the number of common nodes in P_i and P'_j , and n is the number of nodes.

Modularity is a method proposed by Mark Newman to assess the robustness of a network community structure [9]. Because modularity does not consider overlapping nodes, it can only be used as an evaluation metric for disjoint community detection. To evaluate the quality of division in overlapping communities, Shen Wei-hua et al. [48] introduced extended modularity (EQ), which considers the overlapping degree of nodes based on the Q function. The computation method is as follows:

$$EQ = \frac{1}{2m} \sum_c \sum_{i,j \in C_k} \frac{1}{O_i O_j} \left[A_{ij} - \frac{k_i k_j}{2m} \right], \quad (5)$$

where m represents the number of edges in the network, A_{ij} is the adjacency matrix of the network, k_i and k_j represent the degrees of node i and node j , respectively, and O_i represents the number of communities that node i belongs to at the same time. If EQ is 1 or close to 1, the community structure is obvious; however, in actual networks the value of EQ is usually in the range of [0.3–0.7].

5. Results and Discussion

In this section, we present the results of a community detection conducted on LFR benchmark networks and real networks. Considering the availability of a known community structure in LFR benchmark networks, we utilized NMI as the evaluation metric. By contrast, modularity was employed as the evaluation metric for real networks, which frequently lack a standard community division. To provide a comprehensive analysis of the algorithm's effectiveness, we present the experimental results for synthetic networks and real networks separately.

5.1. Results on Synthetic Networks

The mixing parameter μ in LFR benchmark networks represents the proportion of connections between different community structures within the same network, while O_n represents the proportion of overlapping nodes. Higher values of μ and O_n indicate a greater degree of blurring at the boundary between different communities and a correspondingly complex network structure. By controlling these two parameters, it is possible to identify the benchmark networks with different attributes. Therefore, we constructed four types of synthetic networks with different sizes containing 1000, 2000, 5000, and 8000 nodes. We adjusted the μ and O_n values to modulate the complexity of the network, enabling a comprehensive multidimensional evaluation of our algorithm's effectiveness.

By manipulating μ within the range of [0.1–0.6] while keeping the other parameters constant, we constructed networks containing 1000, 2000, 5000, and 8000 nodes. On this basis, we compared the performance of COPRA [19], CPM [31], SLPA [38], LFM [49], DANMF [50], EMOFM-SC [51], and our proposed algorithm. The experimental results are presented in Table 3. Both the COPRA and SLPA algorithms show improvements upon label propagation. The COPRA algorithm maintains multiple labels for every node and calculates the membership of every label to achieve the purpose of detecting overlapping communities. By contrast, SLPA records the historical label sequence of each node in the iteration process, counts the frequency of label occurrence in the sequence, and filters out the low-frequency labels according to the set threshold. The CPM algorithm detects communities by looking for maximal cliques in the network. The LFM algorithm expands from an arbitrary seed node to form a community through local optimization and stops expanding when the fitness value of the community stops increasing. The DANMF algorithm combines the advantages of a deep autoencoder network (DAN) and non-negative matrix factorization (NMF), and can perform effective dimensionality reduction and clustering on high-dimensional network data, thereby achieving efficient community detection. Finally, the EMOFM-SC algorithm is an evolutionary multi-objective optimization-based fuzzy method for overlapping community detection.

In Table 3, N denotes the number of nodes in the synthetic networks, which were set at 1000, 2000, 5000, and 8000. Note that the maximum number of nodes was limited to

8000 owing to the inability of certain algorithms to detect community structures in larger networks. Every column of the table represents the NMI of the communities identified by every algorithm in the network with varying structures. A higher NMI value represents greater accuracy in community detection. We varied the mixing parameter μ from 0.1 to 0.6, corresponding to the six rows of the table, while keeping the number of nodes constant. All of the other LFR parameters were set to their default values.

For mixing parameters falling within the range of [0.1–0.3], the proportion of connections between nodes in different communities is low, indicating a clear boundary and an obvious community structure within the network. Irrespective of node number, our proposed algorithm INF-COPRA consistently exhibits higher accuracy in this scenario. At μ values close to 0.1, the NMI value of our proposed algorithm approaches 1, demonstrating that the communities identified by our algorithm are similar to the standard community when the community structure of the network is clear. In addition, despite its low time complexity, the CPM algorithm produces communities of unsatisfactory quality. Even when μ is small, the communities identified by the CPM algorithm deviate significantly from the standard community.

Table 3. The NMI values of every algorithm under different mixing parameters.

N	μ	INF-COPRA	COPRA	CPM	SLPA	LFM	DANMF	EMOFM-SC
1000	0.1	0.97	0.9	0.224	0.68	0.8	0.94	0.963
	0.2	0.92	0.74	0.15	0.59	0.62	0.81	0.835
	0.3	0.73	0.58	0.071	0.49	0.47	0.78	0.802
	0.4	0.45	0.13	0.043	0.32	0.24	0.55	0.611
	0.5	0.29	0.05	0.026	0.07	0.25	0.5	0.478
	0.6	0.2	0.006	0.0009	0.02	0.19	0.34	0.38
2000	0.1	0.98	0.9	0.23	0.6	0.917	0.95	0.947
	0.2	0.97	0.76	0.17	0.59	0.803	0.84	0.909
	0.3	0.83	0.6	0.08	0.54	0.526	0.78	0.853
	0.4	0.55	0.06	0.05	0.32	0.291	0.48	0.732
	0.5	0.29	0.01	0.02	0.07	0.25	0.44	0.205
	0.6	0.2	0.009	0.007	0.02	0.177	0.36	0.082
5000	0.1	0.98	0.6	0.3	0.88	0.78	0.9	0.927
	0.2	0.95	0.5	0.28	0.79	0.7	0.88	0.884
	0.3	0.9	0.24	0.27	0.68	0.61	0.86	0.851
	0.4	0.64	0.009	0.07	0.22	0.4	0.58	0.673
	0.5	0.28	0.007	0.05	0.07	0.23	0.34	0.258
	0.6	0.08	0.006	0.015	0.002	0.14	0.2	0.104
8000	0.1	0.96	0.74	0.67	0.7	0.81	0.9	0.916
	0.2	0.94	0.68	0.59	0.6	0.75	0.87	0.805
	0.3	0.8	0.47	0.42	0.48	0.54	0.52	0.649
	0.4	0.6	0.29	0.27	0.3	0.36	0.27	0.531
	0.5	0.32	0.04	0.02	0.1	0.18	0.005	0.152
	0.6	0.2	0.006	0.0009	0.02	0.09	0.0004	0.093

As the μ value gradually increases and the network structure becomes more complex, the accuracy of every algorithm gradually decreases, and COPRA, SLPA, and CPM even approach zero. In certain networks with high μ , our algorithm is not as accurate as DANMF and EMOFM-SC, however it outperforms other algorithms by a considerable margin. This means that there is room for improvement in our proposed algorithm when faced with complex community structures. We speculate that when the boundaries between communities become blurred, the interference of label propagation in the network gradually increases, which affects the final experimental results. It is remarkable that our algorithm can show good performance as the number of nodes reaches 8000, when other algorithms gradually lose the ability to mine community structure but our algorithm is able to maintain good results.

Subsequently, we held the other parameters constant while varying O_n (representing the overlapping density) in the range of [10–80%]. The experimental results are presented in Table 4, where N denotes the number of nodes in the synthetic network, which was set to 1000, 2000, 5000, and 8000.

An intriguing result can be seen for the networks consisting of 8000 nodes. Specifically, when the overlapping density reaches 80% all of the algorithms except for those based on label propagation fail to converge, and are unable to effectively identify communities within the network. Although the SLPA and COPRA algorithms are able to identify communities, there is a significant disparity between their outcomes and the standard community division. By contrast, our proposed algorithm demonstrates a high degree of similarity with the standard community division, thereby validating the efficacy of INF-COPRA. These findings underscore the accuracy of our algorithm, particularly when dealing with networks characterized by ambiguous community boundaries.

Table 4. The NMI values of every algorithm under different overlapping densities.

N	O_n	INF-COPRA	COPRA	CPM	SLPA	LFM	DANMF	EMOFM-SC
1000	0.1	0.95	0.85	0.48	0.88	0.78	0.89	0.923
	0.2	0.91	0.74	0.38	0.79	0.7	0.82	0.905
	0.4	0.63	0.2	0.16	0.38	0.42	0.65	0.647
	0.6	0.35	0.009	0.07	0.02	0.16	0.39	0.478
	0.8	0.2	0.002	0.05	0.007	0.07	0.26	0.091
2000	0.1	0.98	0.6	0.32	0.77	0.77	0.89	0.875
	0.2	0.96	0.56	0.3	0.79	0.68	0.88	0.893
	0.4	0.69	0.1	0.12	0.38	0.46	0.65	0.649
	0.6	0.33	0.004	0.06	0.02	0.25	0.37	0.531
	0.8	0.09	0.002	0.03	0.007	0.15	0.2	0.192
5000	0.1	0.95	0.73	0.65	0.71	0.82	0.89	0.798
	0.2	0.91	0.67	0.59	0.7	0.74	0.86	0.634
	0.4	0.8	0.47	0.42	0.28	0.55	0.51	0.571
	0.6	0.6	0.29	0.27	0.02	0.37	0.28	0.208
	0.8	0.32	0.03	0.017	0.007	0.2	0.005	0.081
8000	0.1	0.95	0.755	0.265	0.71	0.482	0.489	0.693
	0.2	0.928	0.738	0.159	0.6	0.374	0.486	0.592
	0.4	0.809	0.492	0.106	0.28	0.192	0.32	0.502
	0.6	0.563	0.011	0.065	0.1	0.162	0.28	0.291
	0.8	0.32	0.008	-	0.01	-	-	-

5.2. Results on Real Networks

In addition to constructing the LFR benchmark network, we assessed the effectiveness of every algorithm on five real datasets. Considering the absence of a standard community division for these datasets, we utilized EQ as the evaluation metric. The EQ values for every algorithm are presented in Table 5.

Table 5. Comparison of EQ values of different algorithms in real networks.

	INF-COPRA	COPRA	CPM	SLPA	LFM	DANMF	EMOFM-SC
karate	0.413	0.387	0.116	0.214	0.328	0.39	0.257
dolphins	0.392	0.379	0.287	0.381	0.341	0.375	0.272
football	0.591	0.359	0.331	0.353	0.513	0.603	0.301
Facebook	0.558	0.456	0.263	0.505	0.386	0.386	0.409
Email	0.392	0.224	0.013	0.317	0.348	0.35	0.28

In the Karate and Dolphins datasets the number of nodes was less than 100, and the INF-COPRA algorithm ranks first. This indicates that our algorithm can effectively detect network communities even in small-scale networks. While the EQ value of INF-COPRA

on the Football dataset is slightly lower than that of DANMF, the execution time of INF-COPRA is far lower than that of the community mining algorithms based on NMF, and the time complexity is not on the same level either. Furthermore, INF-COPRA ranks second after DANMF, with both being far more effective than other algorithms. On the Facebook and Enron datasets, INF-COPRA achieves the best results, with an EQ value 10% higher than the second-best algorithm. This shows from another dimension that the algorithm proposed in this paper has significant advantages in large-scale networks.

To further assess the efficacy of the INF-COPRA algorithm, we present visualizations of the community detection results obtained utilizing INF-COPRA on real networks. Figure 2 illustrates the community division produced by INF-COPRA on the Karate network (EQ = 0.413), which is partitioned into two communities, namely, purple and green. Notably, nodes 3 and 20 are overlapping nodes that belong to the two communities. The visualization results obtained using the Karate dataset underscore the practical significance of INF-COPRA. This is because this dataset depicts social relationships within a karate club in which disagreements between two managers have led members to align themselves into separate communities that reflect the internal dynamics of the club.

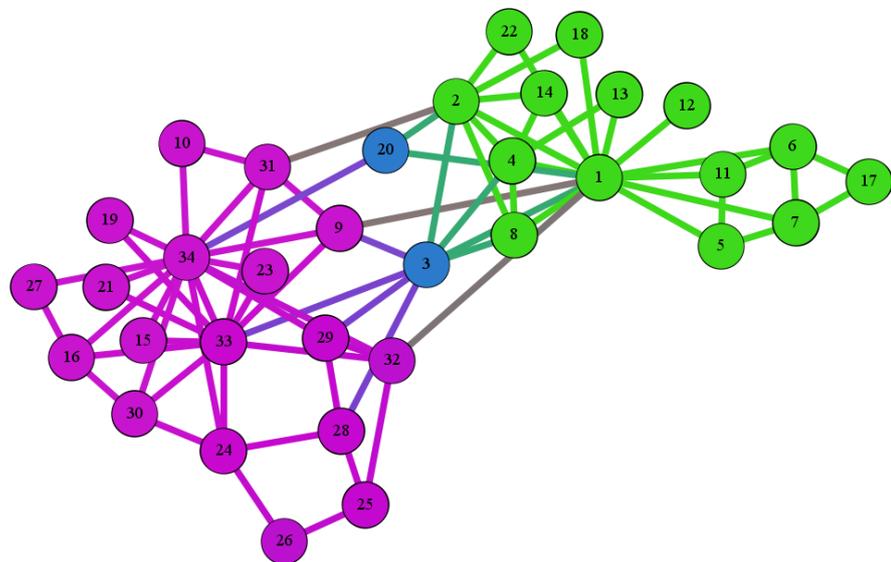


Figure 2. Community division of INF-COPRA on the Karate network.

Figure 3 visualizes the community division in the Dolphins network (EQ = 0.392), which is divided into four different communities, with nodes 20, 39, and 40 being overlapping nodes. This indicates that dolphins are not only active within a single group, and have frequent connections with other dolphin communities.

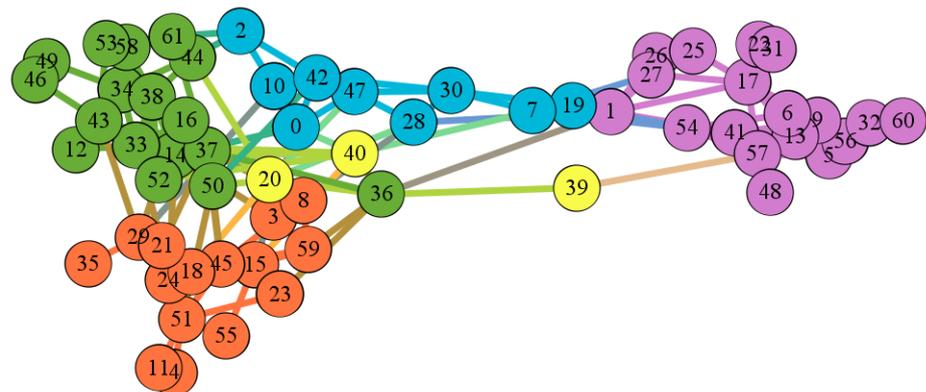


Figure 3. Community division of INF-COPRA on the Dolphins network.

Figure 4 visualizes the community divisions in the Football network ($EQ = 0.591$), which is divided into nine different communities. The size of every community in the Football network is similar because the number of players on each football team is similarly limited. The results indicate that the algorithm can accurately identify the different communities in the network while considering the special attributes of every community.

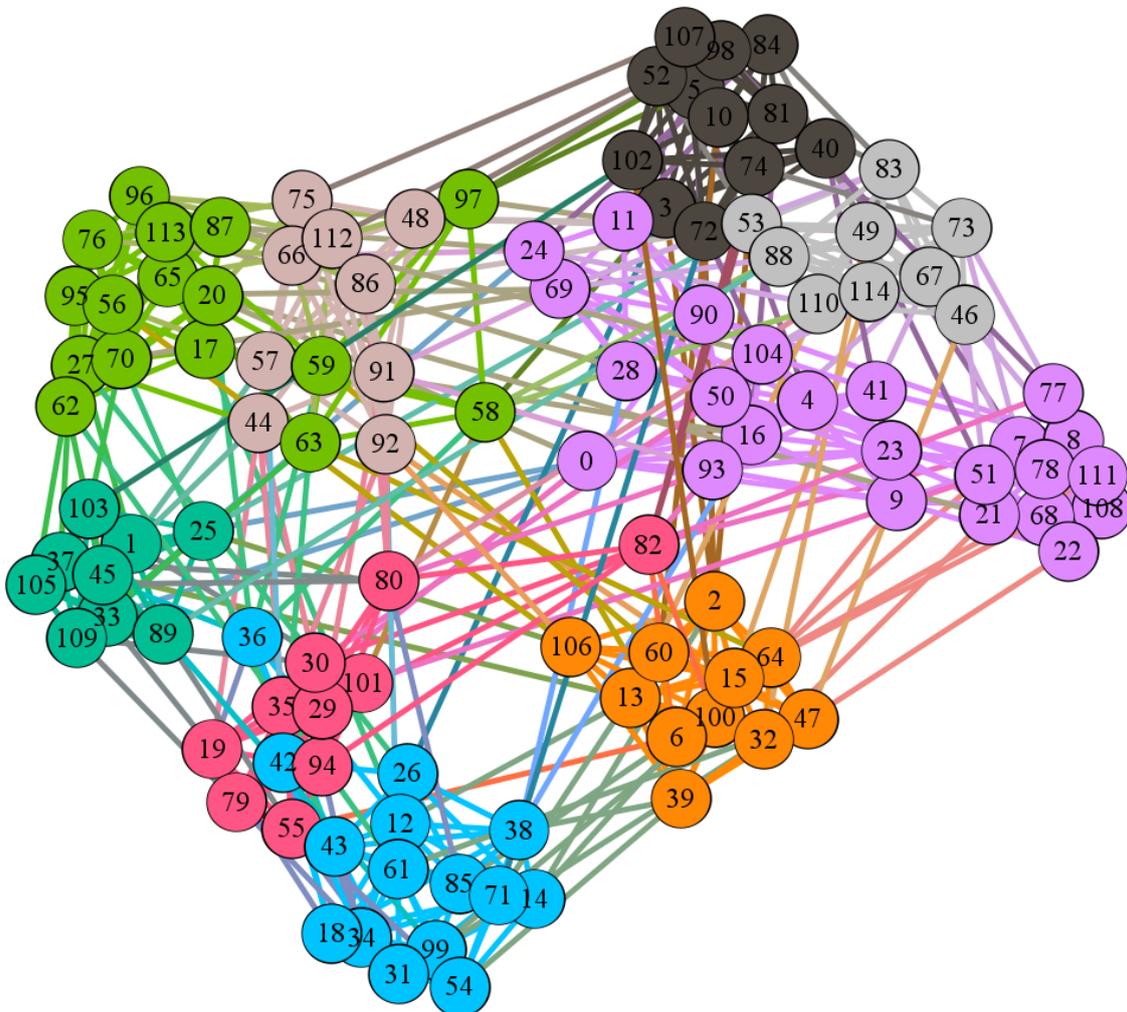


Figure 4. Community division of INF-COPRA on the Football network.

Figure 5 visualizes the community divisions in the Facebook network (where only 10% of the nodes in the network are visualized, $EQ = 0.558$), which is divided into four different communities. There are two “super communities” in the Facebook network in which most nodes are located; at the same time, there are frequent interactions between the two communities. This shows that users with the same interests tend to be frequently connected in social networks.

Figure 6 visualizes the community division of the Enron network (where only 1% of the nodes in the network are visualized, $EQ = 0.392$). The community structure of the Enron network is relatively complex, and it has no clear boundaries between communities. Nodes located in different communities have frequent communication.

5.3. Discussion

In our comparative algorithm, we primarily include local community detection methods. Local community detection methods aim to identify communities based on the local structure of the network, including the nodes’ direct neighbors. These methods typically start with a seed node and iteratively expand the community by adding or removing nodes

according to local criteria. Label propagation methods, by contrast, aim to partition the network according to the nodes' labels or attributes. In these methods, every node is assigned labels, and the labels are propagated through the network until a stable partition is reached.

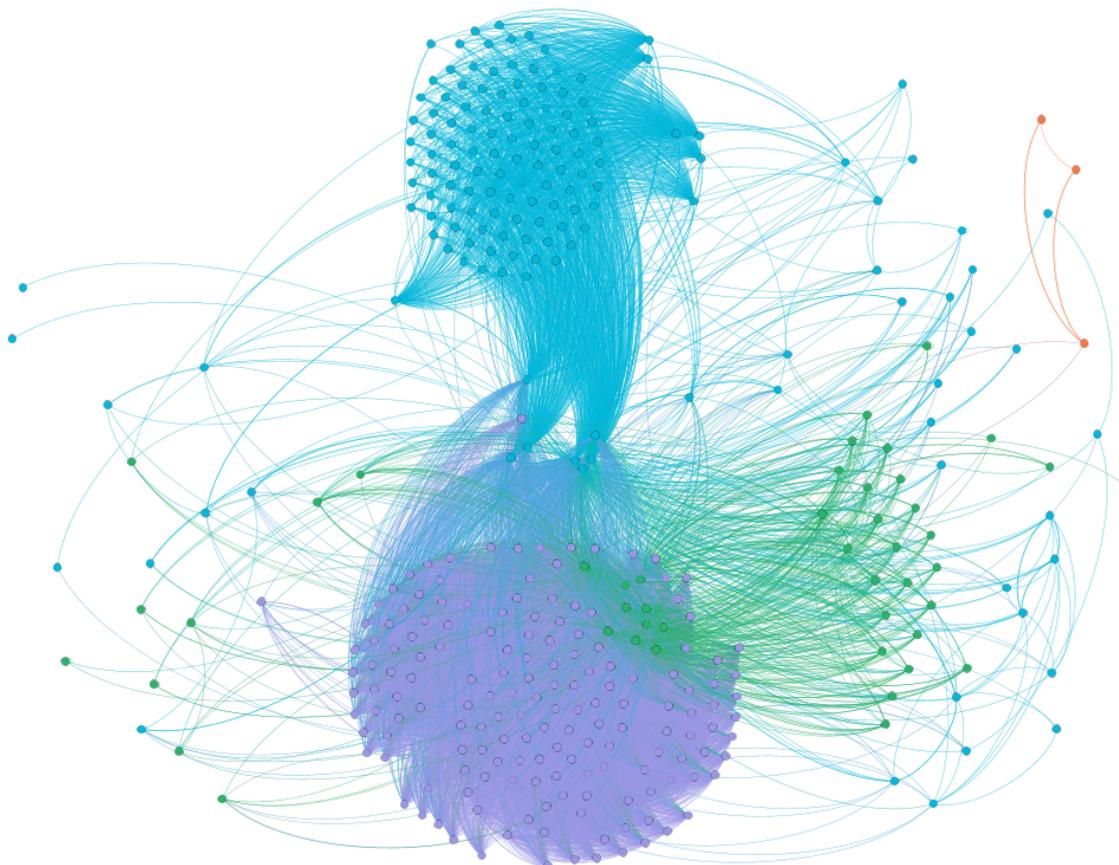


Figure 5. Community division of INF-COPRA on the Facebook network.

In terms of performance, local community detection methods tend to be faster than label propagation methods; however, they may not produce the most accurate results. Label propagation methods, by contrast, may be more accurate; however, they may require additional computational resources. As can be seen in the results, the different parameters of the LFR network affect the results as well. In small-scale networks, as the parameters (both μ and O_n) increase gradually and the boundary between different communities becomes blurred, and the accuracy of INF-COPRA falls below that of DANMF. Based on our analysis, we think that this is because the label propagation method only uses the topology of the network. Label propagation methods can be considered a hybrid between local and global community detection methods. Although they rely on the local similarity between nodes, they can propagate information globally through the network, resulting in a more global division. However, the final division remains influenced by the local structure of the network. While this idea improves the efficiency of the algorithm, it may cause the loss of higher-order information in the network. Concurrently, the algorithm can propagate information globally through the network, making more complex network structures less conducive to the propagation of labels. When the number of nodes in the network increases gradually, INF-COPRA can maintain high accuracy, which confirms the conclusion that it is applicable to large-scale networks.

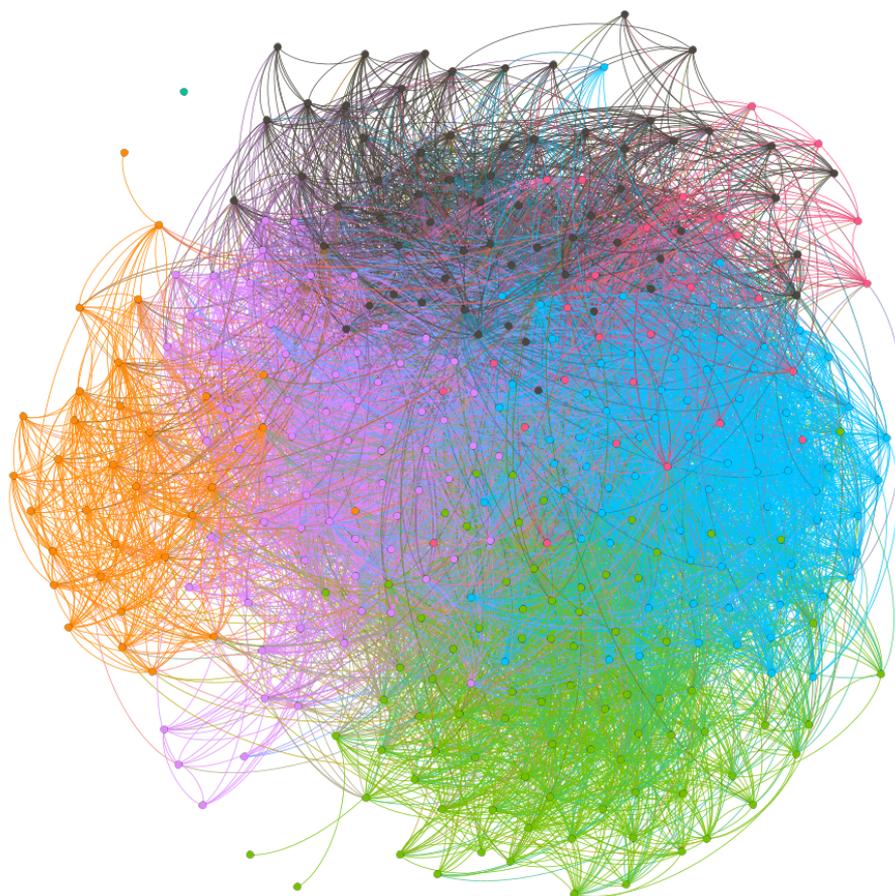


Figure 6. Community division of INF-COPRA on the Enron network.

6. Conclusions

In this paper, we have proposed INF-COPRA to detect overlapping communities. INF-COPRA utilizes a new initialization and label propagation method that removes unnecessary labels, thereby improving the accuracy and robustness of community detection. In particular, we developed two functions to control the process of label propagation. In the first, we sort the nodes by importance and only assign labels to the important ones, reducing the time cost and avoiding label error propagation. Second, when facing multiple alternative nodes, we prioritize those with the highest influence, ensuring that correct messages are transmitted in every iteration. Experimental results on various synthetic networks and real networks demonstrates that INF-COPRA outperforms other state-of-the-art methods on community discovery tasks. On synthetic networks, our algorithm was able to identify communities that were highly similar to the standard community. On real networks, our method achieved better modularity. Nevertheless, on certain small-scale networks our algorithm may not achieve the best results owing to subjective influence and label settings. In addition, network generation factors may affect this approach. Moreover, INF-COPRA has a number of limitations, including that it is applicable only to undirected and unweighted networks. Future research will explore applying the INF-COPRA model to directed weighted networks and extending it to dynamic networks.

Author Contributions: Conceptualization, H.X. and L.T.; methodology, H.X. and L.T.; software, H.X.; validation, H.X., Y.R., J.X. and L.T.; writing—original draft preparation, H.X.; writing—review and editing, Y.R., J.X. and L.T.; supervision, L.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the International Science and Technology Cooperation Projects of China of grant number 2022YFE0112300; National Natural Science Foundation of China of grant number 62271411, 61976181; Natural Science Basic Research Plan in Shaanxi Province of China of grant number 2022JM-325.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yang, B.; Liu, D.; Liu, J. *Handbook of Social Network Technologies and Applications*; Springer: Boston, MA, USA, 2010; pp. 331–346.
2. Fortunato, S.; Newman, M.E. 20 years of network community detection. *Nat. Phys.* **2022**, *18*, 848–850. [[CrossRef](#)]
3. Chen, Y.; Chuang, C.; Chiu, Y. Community detection based on social interactions in a social network. *J. Assoc. Inf. Sci.* **2014**, *539*–550. [[CrossRef](#)]
4. Cai, B.; Wang, Y.; Zeng, L. Edge classification based on Convolutional Neural Networks for community detection in complex network. *Physica A* **2020**, *556*, 124826. [[CrossRef](#)]
5. Li, G.; Guo, K.; Chen, Y.Z. A dynamic community detection algorithm based on Parallel Incremental Related Vertices. In Proceedings of the IEEE International Conference on Big Data Analysis, Beijing, China, 10–12 March 2017; pp. 10–12.
6. Hu, D.; Sarder, P.; Ronhovde, P. Automatic segmentation of fluorescence lifetime microscopy images of cells using multiresolution community detection—A first study. *Microscopy* **2013**, *1*, 54–64. [[CrossRef](#)] [[PubMed](#)]
7. Li, H.; Liu, Z.P.; Chen, L. Identification of overlapping communities in protein interaction networks using multi-scale local information expansion. In Proceedings of the 10th World Congress on Intelligent Control and Automation, Beijing, China, 6–8 July 2012; pp. 6–8.
8. Tian, B.; Li, W. Community Detection Method Based on Mixed-norm Sparse Subspace Clustering. *Neurocomputing* **2018**, *275*, 2150–2161. [[CrossRef](#)]
9. Newman, M. Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA* **2006**, *103*, 8577–8582. [[CrossRef](#)]
10. Cai, Q.; Ma, L.; Gong, M.; Tian, D. A survey on network community detection based on evolutionary computation. *Int. J. Bio-Inspir. Com.* **2016**, *8*, 84–98. [[CrossRef](#)]
11. Prokhorenkova, L.; Tikhonov, A. Community detection through likelihood optimization: In search of a sound model. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 1498–1508.
12. Romdhane, L.B.; Chaabani, Y.; Zardi, H. A robust ant colony optimization-based algorithm for community mining in large scale oriented social graphs. *Expert. Syst. Appl.* **2013**, *40*, 5709–5718. [[CrossRef](#)]
13. Žalik, K.R.; Žalik, B. Multi-objective evolutionary algorithm using problem-specific genetic operators for community detection in networks. *Neural. Comput. Appl.* **2018**, *30*, 2907–2920. [[CrossRef](#)]
14. Satuluri, V.; Parthasarathy, S. Scalable graph clustering using stochastic flows: Applications to community discovery. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data mining, Paris, France, 28 June–1 July 2009; pp. 737–746.
15. Lynn, C.W.; Bassett, D.S. Quantifying the compressibility of complex networks. *Proc. Natl. Acad. Sci. USA* **2021**, *118*, 32. [[CrossRef](#)]
16. Patel, N.V.; Sutaria, N. A survey on community detection in social network using genetic algorithm. *Proc. SPIE Int. Soc. Opt. Eng.* **2015**, *3*, 16–19.
17. Vieira, V.D.F.; Xavier, C.R.; Evsukoff, A.G. A comparative study of overlapping community detection methods from the perspective of the structural properties. *Appl. Net. Sci.* **2020**, *5*, 1–42. [[CrossRef](#)]
18. Mittal, R.; Bhatia, M.P.S. Classification and comparative evaluation of community detection algorithms. *Arch. Comput. Method E* **2021**, *28*, 1417–1428. [[CrossRef](#)]
19. Gregory, S. Finding overlapping communities in networks by label propagation. *NJP* **2010**, *12*, 10. [[CrossRef](#)]
20. Chen, N.; Liu, Y.; Chen, H. Detecting communities in social networks using label propagation with information entropy. *Physica A* **2017**, *471*, 788–798. [[CrossRef](#)]
21. Jia, H.C.; Ratnavelu, K. Detecting Community Structure by Using a Constrained Label Propagation Algorithm. *PLoS ONE* **2016**, *11*, e0155320.
22. Xing, Y.; Meng, F.; Zhou, Y.; Zhu, M.; Shi, M.; Sun, G. A Node Influence Based Label Propagation Algorithm for Community Detection in Networks. *Sci. World J.* **2014**, *2014*, 627581. [[CrossRef](#)]
23. Xie, J.; Szymanski, B.K. LabelRank: A Stabilized Label Propagation Algorithm for Community Detection in Networks. In Proceedings of the 2013 IEEE 2nd Network Science Workshop (NSW), New York, NY, USA, 29 April–1 May 2013; pp. 138–143.
24. Girvan, M.; Newman, M.J. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* **2002**, *12*, 99. [[CrossRef](#)]
25. Kernighan, B.W.; Lin, S. An efficient heuristic procedure for partitioning graphs. *Bell Labs Tech. J.* **2014**, *2*, 291–307. [[CrossRef](#)]

26. Kigerl, A. Behind the Scenes of the Underworld: Hierarchical Clustering of Two Leaked Carding Forum Databases. *Soc. Sci. Comput. Rev.* **2022**, *3*, 618–640. [CrossRef]
27. Newman, M.E. Fast algorithm for detecting community structure in networks. *Phys. Rev. E* **2004**, *6*, 69. [CrossRef] [PubMed]
28. Blondel, V.D.; Guillaume, J.L.; Lambiotte, R.; Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, *10*, P10008. [CrossRef]
29. Xie, J.; Kelley, S.; Szymanski, B.K. Overlapping Community Detection in Networks: The State of the Art and Comparative Study. *ACM Comput. Surv.* **2013**, *4*, 1–35. [CrossRef]
30. Guimera, R.; Danon, L.; Diaz-Guilera, A.; Giralt, F.; Arenas, A. The real communication network behind the formal chart: Community structure in organizations. *J. Econ. Behav. Organ.* **2006**, *61*, 653–667. [CrossRef]
31. Palla, G.; Deranyi, I.; Farkas, I.; Vicsek, T. Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **2005**, *435*, 814–818. [CrossRef]
32. Zhang, H.; Qiu, B.; Giles, C.L.; Foley, H.C.; Yen, J. An LDA-based community structure discovery approach for large-scale social networks. In Proceedings of the 2007 IEEE Intelligence and Security Informatics, New Brunswick, NJ, USA, 23–24 May 2007; pp. 200–207.
33. Fouss, F.; Piroette, A.; Renders, J.M.; Saerens, M. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Trans. Knowl. Data Eng.* **2007**, *19*, 355–369. [CrossRef]
34. Baumes, J.; Goldberg, M.K.; Krishnamoorthy, M.S.; Magdon-Ismael, M.; Preston, N. Finding communities by clustering a graph into overlapping subgraphs. *IADIS AC* **2005**, *5*, 97–104.
35. Bandyopadhyay, S.; Chowdhary, G.; Sengupta, D. FOCS: Fast Overlapped Community Search. *IEEE Trans. Knowl. Data Eng.* **2015**, *27*, 2974–2985. [CrossRef]
36. Nepusz, T.; Yu, H.; Paccanaro, A. Detecting overlapping protein complexes in protein-protein interaction networks. *Nat. Methods* **2012**, *9*, 471–472. [CrossRef]
37. Raghavan, U.N.; Albert, R.; Kumara, S. Near Linear Time Algorithm to Detect Community Structures in Large-Scale Networks. *Phys. Rev. E* **2007**, *76*, 036106. [CrossRef]
38. Xie, J.; Szymanski, B.K. Community Detection Using A Neighborhood Strength Driven Label Propagation Algorithm. In Proceedings of the 2011 IEEE Network Science Workshop, Washington, DC, USA, 22–24 June 2011; pp. 188–195.
39. Huang, L.; Wang, G.; Wang, Y. LPANNI: Overlapping community detection using label propagation in large-scale complex networks. *IEEE Trans. Knowl. Data Eng.* **2019**, *31*, 1736–1749.
40. Fortunato, S.; Darko, H. Community detection in networks: A user guide. *Phys. Rep.* **2016**, *659*, 1–44. [CrossRef]
41. Centola, D. The Spread of Behavior in an Online Social Network Experiment. *Science* **2010**, *329*, 1194–1197. [CrossRef]
42. Kitsak, M.; Gallos, L.K.; Havlin, S.; Liljeros, F.; Muchnik, L.; Stanley, H.E.; Makse, H.A. Identification of influential spreaders in complex networks. *Nat. Phys.* **2010**, *6*, 888–893. [CrossRef]
43. Zachary, W.W. An Information Flow Model for Conflict and Fission in Small Groups. *J. Anthropol. Res.* **1976**, *33*, 452–473. [CrossRef]
44. Alamsyah, A.; Rahardjo, B. Community Detection Methods in Social Network Analysis. *J. Comput. Theor. Nanosci.* **2014**, *20*, 250–253. [CrossRef]
45. Stanford Large Network Dataset Collection. Available online: <http://snap.stanford.edu/data/> (accessed on 1 June 2014).
46. Lancichinetti, A.; Fortunato, S.; Radicchi, F. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **2008**, *78*, 046110. [CrossRef]
47. Danon, L.; Diaz-Guilera, A.; Duch, J.; Arenas, A. Comparing community structure identification. *J. Stat. Mech. Theory Exp.* **2005**, *9*, P09008. [CrossRef]
48. Shen, H.; Cheng, X.; Cai, K.; Hu, M.B. Detect overlapping and hierarchical community structure in networks. *Physica A* **2009**, *388*, 1706–1712. [CrossRef]
49. Lancichinetti, A.; Fortunato, S.; Kertész, J. Detecting the overlapping and hierarchical community structure in complex networks. *New J. Phys.* **2009**, *11*, 033015. [CrossRef]
50. Ye, F.; Chen, C.; Zheng, Z. Deep autoencoder like nonnegative matrix factorization for community detection. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, 22–26 October 2018; pp. 1393–1402.
51. Ye, T.; Yang, S.; Zhang, X. An evolutionary multiobjective optimization based fuzzy method for overlapping community detection. *IEEE Trans. Fuzzy Syst.* **2019**, *28*, 2841–2855.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.