Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 6523363, 12 pages
https://doi.org/10.1155/2022/6523363

Research Article

Label Propagation Community Detection Algorithm Based on Density Peak Optimization

Ma Yan[1] **and Chen Guoqiang**[2]

[1]*Department of Software Engineering, School of Computer and Information Engineering, Henan University, Henan Province, China*
[2]*Information Security Department, School of Computer and Information Engineering, Henan University, Henan Province, China*

Correspondence should be addressed to Chen Guoqiang; chengq08@163.com

Received 4 January 2022; Revised 2 March 2022; Accepted 10 March 2022; Published 28 April 2022

Academic Editor: Xingsi Xue

Community structure detection in a complex network structure and function is used to understand network relations and find its evolution rule; monitoring and forecasting its evolution behavior have an important theoretical significance; in the epidemic monitoring, network public opinion analysis, recommendation, advertising push and combat terrorism, and safeguard national security, it has wide application prospect. A label propagation algorithm is one of the popular algorithms for community detection in recent years; the community detection algorithm based on tags that spread the biggest advantage is the simple algorithm logic, relative to the module of optimization algorithm, convergence speed is very fast, the clustering process without any optimization function, and the initialization before do not need to specify the number of complex network community. However, the algorithm has some problems such as unstable partitioning results and strong randomness. To solve this problem, this paper proposes an unsupervised label propagation community detection algorithm based on density peak. The proposed algorithm first introduces the density peak to find the clustering center, first determines the prototype of the community, and then fixes the number of communities and the clustering center of the complex network, and then uses the label propagation algorithm to detect the community, which improves the accuracy and robustness of community discovery, reduces the number of iterations, and accelerates the formation of the community. Finally, experiments on synthetic network and real network data sets are carried out with the proposed algorithm, and the results show that the proposed method has better performance.

1. Introduction

Community structure is a very important attribute in complex networks. Therefore, community structure plays a crucial role not only in the analysis of the social relations in human society [1] but also in the analysis of the functional relations between biological network organizations and organs [2], as well as the analysis of the citation relations between collaborative networks among scientists [3]. Therefore, the discovery of community structures from complex networks has been extensively studied in the past decade [4–8].

In 2002, Girvan and Newman achieved pioneering work pointing out that community structure is common in complex networks and proposed modularity Q to measure the stability of communities in networks [1]. Although the definition of community structure has not been unanimously determined by clear relevant studies, it is generally considered that a community is a group of nodes, which can also be called a community or a group of modules. These nodes are characterized by tight internal connections and sparse external connections [9].

As one of the hot spots of current research, community discovery algorithm based on label propagation has been widely used in community detection. This algorithm is a graph-based semisupervised learning method [10]. The advantage of semisupervised learning is that it can determine a lot of unlabeled samples by a small number of marked samples, thus improving the effectiveness of learning process [11, 12]. The basic idea of label propagation is to predict the label information of unlabeled nodes by using the topological relations between nodes from the label

information of labeled nodes and finally complete the division of the graph to form a clustering structure. Although this algorithm has the advantages of simple implementation, clear logic, no need to know the number of communities in advance, time complexity is close to linearity, etc., the unstable partition results and strong randomness are the defects of this algorithm. In each iteration of the label propagation algorithm, which community a node belongs to depends on the label with the largest cumulative weight of its neighbor nodes. Therefore, when more than one of the largest neighbor labels appears on a node, one of them will be randomly selected as its own label. This kind of randomness will bring avalanche effect, that is, a small clustering result error at the beginning will be continuously amplified. In addition, the updating order of node labels will also have a great impact. Obviously, the earlier the updating of the most important node will accelerate the process of convergence. In the label propagation algorithm, the closer the initial label is set to the core point, the more accurate the clustering effect is. However, in specific applications, it is often not feasible to know the number of communities in advance, and it is very inefficient to determine the number of communities (K) by searching all possible candidate communities. Therefore, we are inspired by the density peak algorithm (DP) [13] and propose a label propagation algorithm based on density peak (DPLPA) for solving complex networks. The central idea of DP algorithm is that the core nodes are surrounded by other nodes in the same class, and there is no possibility for the core nodes to be closely connected. In other words, the core nodes have higher density, so this algorithm is feasible to calculate the core number. But unfortunately, DP algorithms cannot be directly used in a complex network, so DP algorithm is improved, and it can be applied to a complex network, can be reasonably come to the core number, applied to the label propagation algorithm, and according to the topology of the network that similarity matrix and priority to update nodes, reduce the randomness and the number of iterations.

2. Background

2.1. Label Propagation Algorithm. Raghavan et al. proposed the label propagation algorithm (LPA) [14], which used the label values of a few preset nodes to divide the community structure on a large-scale complex network. However, the accuracy of LPA is low because of the randomness of propagation, which leads to a large error in clustering results. The reason is that when the neighbor node label frequency appears with multiple highest values, the algorithm is fair to each label. We randomly select a label as the label of the update node. Therefore, the algorithm will appear small and fragmented communities or large communities which are not in line with the actual situation when the community is divided. Figure 1 is a situation where an error occurs in the label propagation process. The d-label finally appears in two communities, which is not in line with the actual situation.

In view of the problems of LPA, domestic and foreign scholars have proposed many improvement measures. Tibély and Kertész [15] proved that the LPA will produce different community structures for the same network, and the algorithm still has a lot of randomness. Leung et al. [16] discovered the possibility of LPA application on tens of millions of networks and found the potential of large-scale data application of the algorithm. Barber and Clark [17] proposed the LPAm to solve the problem that the LPA cannot integrate different clustering results well by adding some restrictive conditions. Liu and Murata [18] solved the problem that LPAm was easy to fall into local optimal solution by optimizing the modularity. Zhuoxiang et al. [19] calculated the K value by calculating the potential influence of nodes. When the K value is less than the actual number of communities, the algorithm will not get the correct partition result. Xie and Szymanski [20] combined the label propagation algorithm with the Markov clustering algorithm (MCL) and proposed a new label propagation algorithm LabelRank. The biggest feature of the LabelRank algorithm is that a node can have multiple neighbor labels during the propagation process. Lin et al. [21] sorted the node weights and then updated the node labels in order. Zhang et al. [22] proposed a labeling algorithm based on edge clustering coefficient. Kipf and Welling [23] extended the graph-based label propagation algorithm and used graph convolution neural network for label propagation. The algorithm realized the propagation of label information through the aggregation of adjacent nodes. In addition, PageRank is used to quantify the importance of nodes, and LPAp algorithm [24] based on the importance of nodes is proposed. An improved community discovery algorithm based on feedback control [25], objective function [17], circle [26], and other methods for label propagation is proposed. The above algorithm is to optimize and improve the problem of node label in the propagation process, which can improve the stability and accuracy of LPA to a certain extent, but most of them bring more or less increased computational overhead, and do not achieve very ideal results.

However, Zhu et al. proposed another label propagation algorithm (LP) in reference [27]. They described the clustering problem as a form of propagation on the graph, in which the label of one node propagates to the neighboring nodes according to the similarity between them. In this process, LP fixes a small number of tags on the known label data. Then, the tagged data, like a signal source, pushes the label through the unlabeled data. Therefore, an accurate number of known tags will play an important role in the propagation process of LP algorithm, greatly improving the accuracy of clustering results.

The algorithm based on label propagation can be described as follows:

(1) Propagation label: $F = P \times F$

(2) Reset the label of the core point in F: $FL = YL$

(3) Repeat steps (1) and (2) until F converges

Where step (1) multiplies the probability transition matrix P and the label matrix F to propagate the label of each node to other nodes with the probability of P. If the

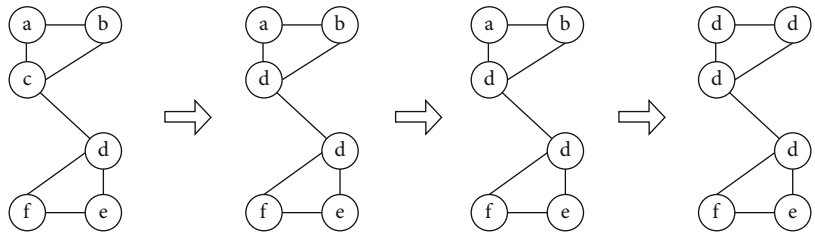

Figure 1: Example of the propagation of an error.

similarity between two nodes is very high, the easier it is for each other's label to be replaced by its own. Step (2) the most important thing is the known label, which cannot be changed, so every time it is propagated, it must return to the original label. As the label data point continues to propagate its label, the final class boundary passes through the high-density area and stays in the low-density interval. It is equivalent to the label node of each different category to divide the sphere of influence.

However, it is still an open question to determine the number of known labels. Traditional community detection algorithm can obtain the number of communities by optimizing the objective function or evaluation index. However, these methods are easily affected by many factors such as initial matrix and optimization objective function, so it is difficult to accurately determine the number of communities. In order to solve the above problem, we use an improved density peak clustering to obtain the kernel number as the input parameter of LP.

2.2. Density Peak Algorithm. In 2014, Rodriguez and Laio [13] proposed a density-based clustering method in Science, which can recognize clusters of various shapes, and the parameters are easy to determine. This method overcame the disadvantages of DBSCAN algorithm [28], which had large density differences among different classes and was difficult to determine the neighborhood range and had strong robustness. The core idea of the density peak algorithm (DP) is based on the assumption: for the center point of each cluster, the density of the cluster center point is greater than the density of surrounding neighbor points and the distance between the cluster center point and the higher density point is relatively large. Therefore, the DP algorithm has two quantities to calculate: the local density of the node and the distance from the high-density node. Usually, ρ_i is used to represent the local density of node i, and δ_i is used to represent the distance between node i and the high-density node.

There are two ways to define local density ρ_i, one of which is

$$\rho_i = \sum_j \chi(d(i,j) - d_c), \tag{1}$$

where

$$\chi(x) = \begin{cases} 1, x \le 0, \\ 0, \text{otherwise}. \end{cases} \tag{2}$$

Here, $d(i,j)$ represents the distance from node i to node j , and d_c is the cut-off distance, that is, the number of nodes whose distance to node i is less than d_c.

The second method is the Gaussian kernel function:

$$p_i = \sum_j \exp\left(-\frac{d(i,j)^2}{d_c^2}\right). \tag{3}$$

The minimum distance δ_i between node i and other higher local density nodes is denoted by the formula defined as

$$\delta_i = \begin{cases} \max_j(d(i,j)), \text{otherwise}, \\ \min_{j:\rho_j>\rho_i}(d(i,j)), \text{if } \rho_j > \rho_i. \end{cases} \tag{4}$$

When all the nodes have calculated ρ_i and δ_i, only the nodes with higher ρ_i and δ_i can become the center points of the cluster, and the points with larger local density distance δ_i but smaller local density ρ_i are abnormal points. The remaining nodes are assigned to the point with the highest local density among the neighbors.

Because the DP algorithm is a density-based clustering algorithm, it has the advantage of detecting clusters of arbitrary shape without the need to set the center point (K value) in advance. Moreover, when selecting the center point, the selection process of the center point can be visually seen through the decision graph. However, DP algorithm still has some defects. Firstly, the value of cut-off distance d_c needs to be set artificially, and improper setting will have a great impact on clustering results. Secondly, the central point needs to be selected artificially, so human subjective factors will affect the clustering results.

3. Methodology

In this section, the proposed label propagation based on density peak optimization clustering algorithm (DPLPA) is introduced. The core idea of DPLPA is to regard the high-density nodes surrounded by nodes of low-density neighbors as the community center points, and the distance between the community center points should be far away. In other words, a node with a higher density is more closely connected to its neighbors and is more likely to be the core point of the community. A community network is a complex network with connections between nodes, which usually reflects the network structure based on the connections between nodes. However, DP algorithm is a density-based clustering

algorithm that handles any shaped data set by calculating the distance between nodes to use high-density areas as a basis for judgment. But this way of calculating distance directly based on coordinates is not applicable to community networks. If the distance between nodes in community network is calculated, the similarity between nodes will become meaningless because the distance between nodes is more uniform or even the same. Therefore, DP algorithm cannot be directly used to detect community networks. In order to solve this problem, this paper uses the improved DP algorithm [29] to obtain the number of communities in a complex network as the input parameter of the label propagation algorithm.

3.1. Predictive Fetch of Label Matrix. Let $G = (V, E)$ be a complex network with no direction and no weight. The node set V contains n nodes, the edge set E contains m edges, and the adjacency matrix of the graph G is A. If node i and node j have an edge connected, then $a_{ij} = 1$ in the adjacency matrix A; otherwise, $a_{ij} = 0$. Therefore, the node similarity formula of node i and node j is obtained, which is expressed by Salton index [30], also known as cosine similarity:

$$S(i, j) = \frac{|N(i) \cap N(j)|}{\sqrt{||N(i)| \times |N(j)||}}, \tag{5}$$

where $N(i)$ and $N(j)$ represent the neighbor nodes of node i and node j, respectively, $|N(i)|$ represents the number of neighbor nodes of node i, so the molecular formula $|N(i) \cap N(j)|$ represents the number of neighbors shared by node i and node j, while denominator formula $\sqrt{|N(i)| \times |N(j)|}$ represents the number of neighbors expected to be shared by node i and node j. The value of $S$is between 0 and 1. When S is closer to 1, the similarity between the two nodes is very high. The formula for the distance between node i and node j is as follows:

$$d_{i,j} = \begin{cases} \dfrac{1}{S(i, j) + \sigma}, i \neq j, \\ 0, i = j. \end{cases} \tag{6}$$

Among them σ is a small positive number, in order to avoid the denominator being 0.

Next, we have two methods to calculate the local density of the node, one is to use the Gaussian kernel function, and the formula is as follows:

$$\rho_i = \sum_j \exp\left(-\frac{d_{i,j}^2}{d_c^2}\right), \tag{7}$$

where ρ_i represents the local density of node i, $d_{i,j}$ represents the distance between node i and node j, d_c represents the cut-off distance, and the size of $d_c$is selected according to [13]. Then, ρ_i normalizes the value:

$$\rho^* = \frac{\rho_i}{\max_j \rho_j}. \tag{8}$$

Then, we start to define the distance formula between nodes:

$$\delta_i = \begin{cases} \max_j\left(d_{ij}\right), \text{if } \max \rho_i, \\ \min_{j:\rho_j > \rho_i}\left(d_{ij}\right), \text{otherwise.} \end{cases} \tag{9}$$

Among them, when the local density of node i is the largest, its distance is the maximum value of the distance between node i and other nodes. When the local density of node i is not the maximum, its distance is the distance between the node whose local density is slightly larger than that of node i and node i.

Then, δ_i is standardized:

$$\delta_i^* = \exp\left(-\left(\frac{d_a^2}{\delta_i^2}\right)\right). \tag{10}$$

The threshold d_a is selected from the list of δ, which is about 80% of the list of δ from small to large [13].

Finally, take ρ^* as the X-axis and δ^* as the Y-axis to generate a decision graph. Then, we calculate each node $\gamma = \rho^* \times \delta^*$, select a value greater than the sum of the average value of γ and the standard deviation of γ to enter the list, and then arrange them in order, and finally select the appropriate cut-off value as the core number (as the known label K) and apply it to the label propagation algorithm.

3.2. Label Propagation Algorithm Based on Density Peak. LP is a graph-based clustering algorithm, so need to construct a graph first. The nodes of the graph are the data points. This paper uses the Gaussian kernel method to construct the weight between the two nodes:

$$w_{ij} = \exp\left(-\frac{d_{ij}^2}{\beta^2}\right). \tag{11}$$

Among them, d_{ij} is the distance between node i and node j, and β is the hyperparameter, and the similarity matrix composed of weight w is obtained.

Next, the known label is propagated through the edges between nodes. The greater the weight of the edge, the more similar the two nodes, and the easier the label is to spread. We define the probability transition matrix:

$$P_{ij} = P(i \longrightarrow j) = \frac{w_{ij}}{\sum_{k=1}^{n} w_{ik}}, \tag{12}$$

where P_{ij} represents the probability of propagating the label of node i to node j. Since there are known K core points with known labels, a $K \times K$ label matrix YL is defined. The i th row represents the label indication vector of node i, that is, if the label of the ith node is i, then the ith element is 1, and the rest are 0. It also defines an unlabeled matrix YU

DPLPA
Input: $G = (V, E)$
Output: Label matrix F
1 Construct adjacency matrix A from complex network$G = (V, E)$.
2 Calculate node similarity S by Equation (5).
3 Calculate the distance matrix d between nodes by Equation (6).
4 Calculate the local density of the node ρ^* by Equations (7) and (8).
5 Calculate δ^* by Equations (9) and (10).
6 Calculate $\gamma = \rho^* \times \delta^*$ get K core points.
7 Get probability transition matrix P by Equations (11) and (12).
8 Build label matrix F by Equation (13).
9 while F convergence criteria not reached do
10: $F = PF$
11: $FL = YL$
12: end while
13:/∗Iteratively update F until convergence, and the label change of the node has been very small. ∗/

Algorithm 1: Gives the pseudocode of DPLPA.

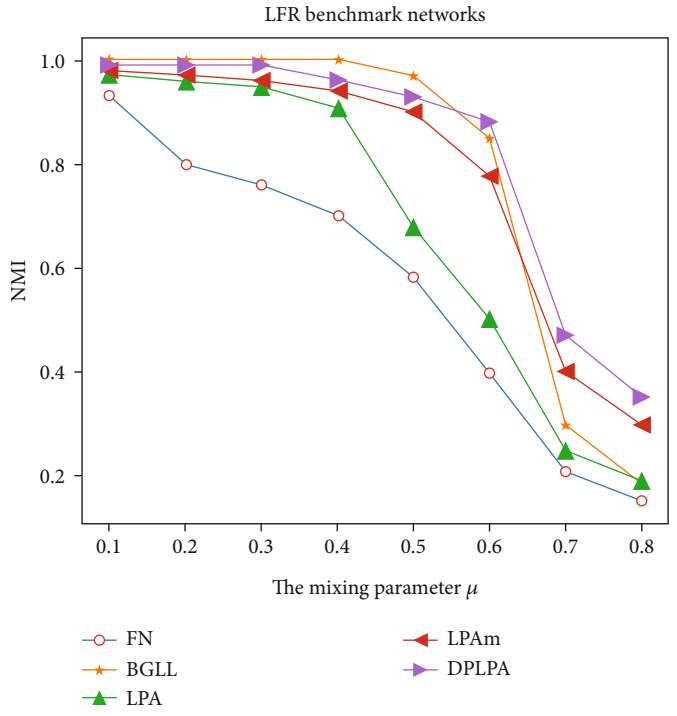

Figure 2: Experimental results on benchmark data set.

Table 1: Concrete description of real network.

Network	n	m	k	Descriptions
Karate	34	78	2	Zachary's karate club
Dolphins	62	159	2	Dolphin social network
Polbook	105	441	3	Books about US politics
Football	115	616	12	American college football

of unlabeled nodes. We combine to get the label matrix of all nodes:

$$F = [YL, YU]. \tag{13}$$

Table 2: Q value comparison of different algorithms in real network.

Networks	FN	BGLL	LPA	LPAm	DPLPA
Karate	0.3807	0.4188	0.3450	0.3496	0.3714
Dolphins	0.4955	0.5188	0.4788	0.4913	0.3789
Polbook	0.5020	0.4986	0.4953	0.4888	0.5063
Football	0.5497	0.6046	0.5445	0.5780	0.5539

Then, the label matrix F is propagated according to the similarity between nodes in the probability matrix P; the formula is as follows:

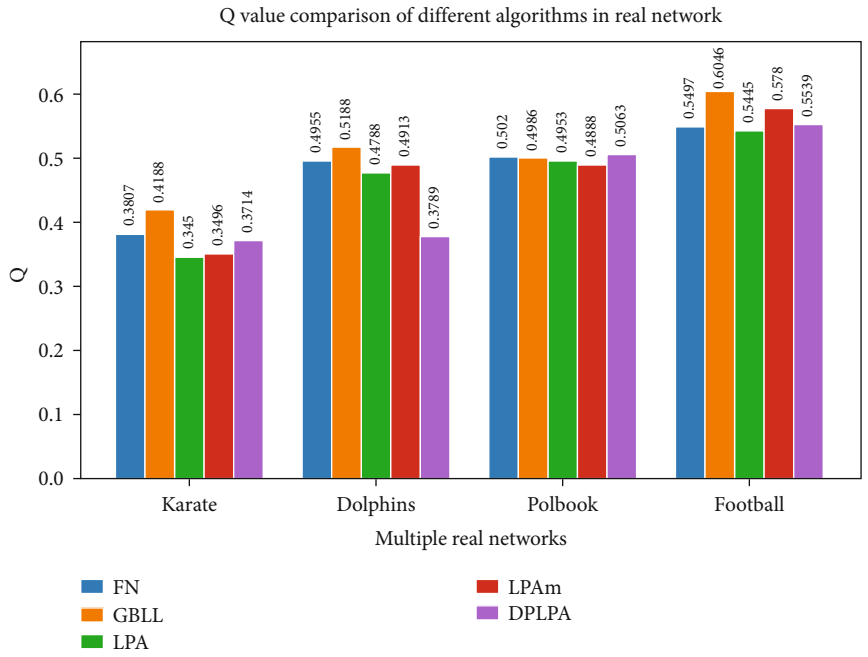

Figure 3: Modularity Q comparison of multiple algorithms on multiple data sets.

$$F = PF. \tag{14}$$

After the propagation, the YL in the known label matrix F changes during the propagation process, but YL is the label value we took for core nodes before, which is accurate and the label should not be changed. Therefore, need to reset the label matrix F, and the formula is as follows:

$$FL = YL. \tag{15}$$

Then, the label matrix F is propagated through the probability matrix P again, and the YL part in the propagated matrix F is reset. We iterate this process until the label change difference of YU in matrix F reaches a critical point. At this moment, DPLPA completes the label partition. Algorithm 1 shows the algorithm flow of DPLPA.

After obtaining the clustered label matrix F, the algorithm will gather the nodes with the value of 1 in the same dimension from F together to form a community. All nodes are divided according to the dimension. The clustering algorithm is finished, and the complex network is also divided.

4. Experimental Study

In order to assess our algorithm, we use a variety of real and synthetic data sets to test, and some classic methods to compare at the same time, including DPLPA in this paper, Newman fast greedy algorithm (*FN*) [31], Louvain algorithm (*BGLL*) [32], LPA [14], and improved label propagation algorithm (*LPAm*) [17]. The hardware environment of the experiment is as follows: Inter (R) Core (TM)i7-7700M CPU, 3.60 GHz, and 8 GB memory. The DPLPA is implemented in Python3.7 64-bit.

Table 3: Network actual grouping of football data sets.

Groups	Numbers
1	2 26 34 38 46 90 104 106 110
2	20 30 36 56 80 95 102
3	3 7 14 16 33 40 48 61 65 101 107
4	4 6 11 41 53 73 75 82 85 99 103 108
5	45 49 58 67 76 87 92 93 111 113
6	37 43 81 83 91
7	13 15 19 27 32 35 39 44 55 62 72 86 100
8	1 5 10 17 24 42 94 105
9	8 9 22 23 52 69 78 79 109 112
10	18 21 28 57 63 66 71 77 88 96 114
11	12 25 51 60 64 70 98
12	29 47 50 54 59 68 74 84 89 115

4.1. Evaluation Metrics. In this article, in order to verify the accuracy of the algorithm, we use the community discovery modularity function Q [31] proposed by Newman as the evaluation index of the experiment. Modularity is defined as

$$Q = \frac{1}{2E}\sum_{ij}\left[A_{ij} - \frac{k_i k_j}{2E}\right]\theta(c_i, c_j), \tag{16}$$

where E represents the total number of edges of the community network, Arepresents the adjacency matrix, k_i represents the degree of node i, and c_i represents the community allocated by node i. $\theta(c_i, c_j)$ is defined as follows:

$$\theta(c_i, c_j) = \begin{cases} 1, (i,j) \in c, \\ 0, \text{otherwise}. \end{cases} \tag{17}$$

Figure 4: Partition of football data set by DPLPA.

Among them, when node i and node j are in the same community, $\theta(c_i, c_j)$ is 1; otherwise, it is0.

At the same time, we still use standardized mutual information (*NMI*) [33] to measure the similarity of two clustering results. It is an important measure of community discovery. It can basically objectively evaluate the comparison between a community division and a real division. For accuracy, the value range of *NMI* is $[0, 1]$, and the higher the value, the closer the divided community is to the real community result. *NMI* is defined as

$$\mathrm{NMI}(A, B) = \frac{-2\sum_{i=1}^{CA}\sum_{i=1}^{CB} C_{ij} \log\left(C_{ij}N/C_{i.}C_{.j}\right)}{\sum_{i=1}^{CA} C_{i.} \log\left(C_{i.}/N\right) + \sum_{j=1}^{CB} C_{.j} \log\left(C_{.j}/N\right)}. \tag{18}$$

Among them, $A(B)$ represents the community discovery algorithm $A(B)$, C is the confusion matrix, C_{ij} represents the number of nodes shared in the method $A(B)$ partition, CA (CB) represents the number of communities in the community discovery method $A(B)$, and $C_{i.}(C_{.j})$ represents the ith row (column j) in C and N represents the number of nodes. If the clustering results of methods A and B are the same, then $\mathrm{NMI}(A, B) = 1$.

4.2. Performance on Synthetic Networks. The use of artificially synthesized networks to evaluate the effectiveness of the algorithm has become an effective means to test the pros and cons of the algorithm. Among them, the most used benchmark test network for community detection, LFR benchmark, was proposed by Lancichinetti et al. [34]. The LFR reference network is an extension of the GN reference network [1] and has high practical value. The LFR benchmark network reflects the heterogeneity of community distribution and the power-law distribution of node degrees. Some of the important parameters are described as follows: n represents the number of nodes, k represents the average degree of nodes, max k represents the maximum degree of nodes, and min c represents the minimum community size, max c represents the maximum community size, $\tau 1$ and $\tau 2$ represent the negative exponents of the power-law distribution of node degree and community size, respectively, and μ is equal to the ratio of the number of connected edges between communities in the network to the total number of edges, to express the obvious degree of the community in the network; the smaller the μ value, the more obvious the structure of the community. Figure 2 shows the comparison of the algorithm's *NMI* experiment results on the LFR benchmark data set.

Table 4: K value comparison of different algorithms in real network.

Networks	FN	BGLL	LPA	DPLPA	True K
Karate	3	4	2	2	2
Dolphins	4	5	4	2	2
Polbook	4	3	4	3	3
Football	6	10	10	12	12

The parameters set in this LFR experiment are $n = 1000$, $k = 15$, max $k = 40$, min $c = 20$, max $c = 50$, $\tau 1 = 2$, $\tau 2 = 1$, and the range of μ is from 0.1 to 0.8. It can be seen from Figure 2 that when μ is small, that is, the community

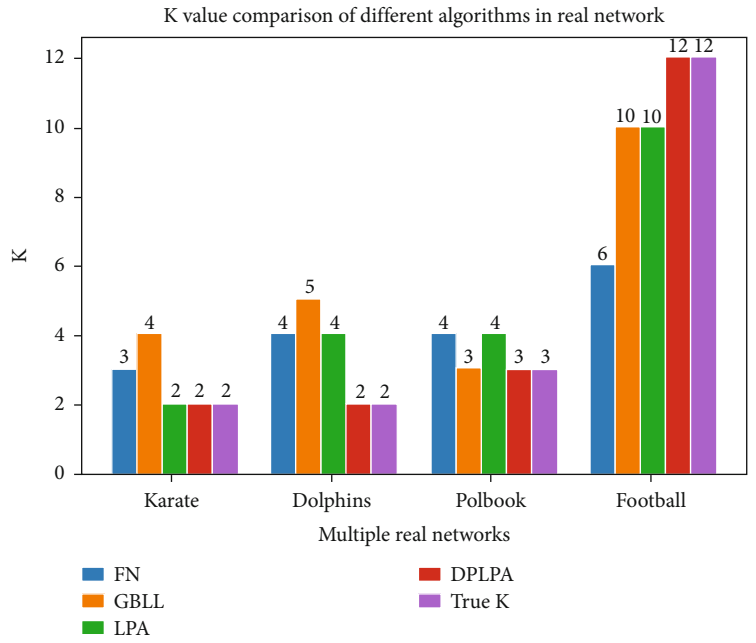

Figure 5: K value comparison of multiple algorithms on multiple data sets.

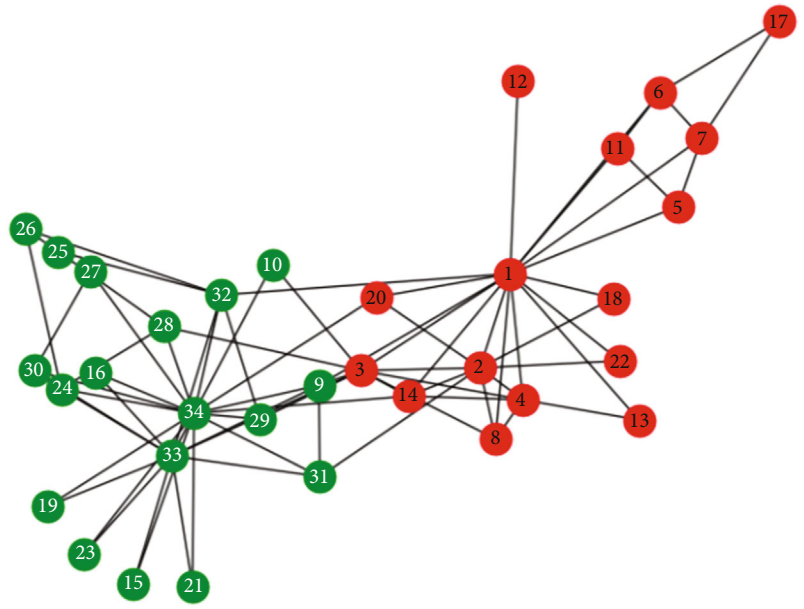

Figure 6: Partition of Karate data set by DPLPA.

structure of the complex network is obvious, the *NMI* values of the other algorithm results are high except for the FN algorithm. The *NMI* value of the FN algorithm and the LPA both began to decrease significantly. The remaining algorithms all began to decrease when the μ value was 0.6, but the DPLAP decreased relatively slowly compared with the BGLL and LPAm, and finally, the *NMI* value is higher, so this can indicate that the DPLPA has a higher accuracy rate in community exploration and has better stability in high-complexity community exploration.

4.3. Real-World Networks. In order to further compare the pros and cons of the algorithms, this paper also tested the algorithm in a few real social networks. These networks are of different sizes but are representative and involve various fields. See Table 1 for details, where n represents the node, m represents the number of edges, and k represents the number of communities that have been identified.

Among them, Karate [35] is a data set of member relations of a university karate club in the United States, which is constructed based on the interactions between club

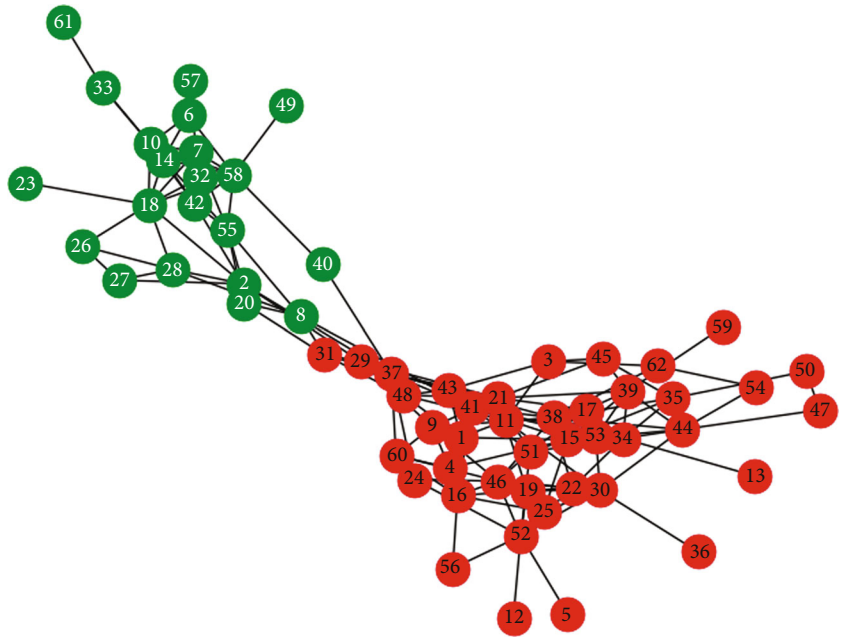

Figure 7: Partition of Dolphins data set by DPLPA.

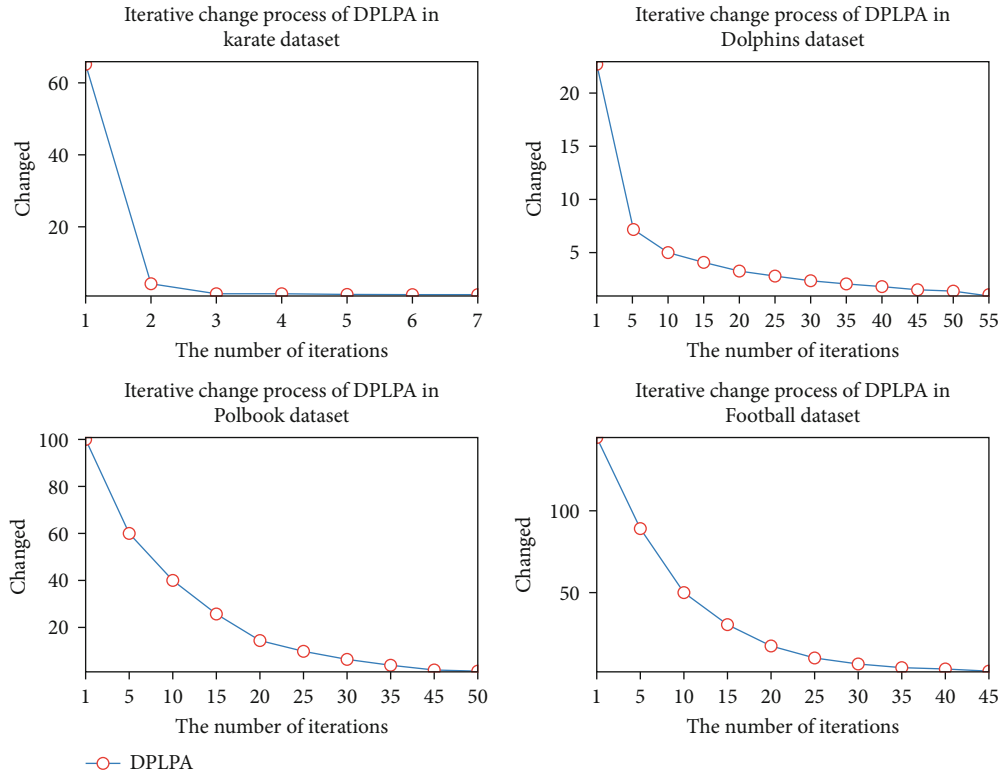

Figure 8: The changes of node labels in DPLPA algorithm during iteration.

members and is often used in the analysis of social networks. Dolphins [36] is a member network constructed from the living habits of 62 wide-mouthed dolphins, and the dolphins that are often together correspond to an edge between nodes. Polbook [37] is a community network constructed through political books sold on Amazon in the United States. Each node represents a book. If two books are purchased by the same customer, there is an edge between them on the corresponding node. Football [1] is a network constructed by the American college football schedule. The nodes represent the participating teams. If there is a match between them, there will be an edge between the nodes. The calculation results of different algorithms on different networks are shown in Table 2 and Figure 3.

In order to better compare the clustering effect of DPLPA on the data set, this paper takes the Football data set to make a detailed explanation. The actual grouping of Football data set is shown in Table 3, and the clustering effect of DPLPA is shown in Figure 4.

It can be seen from Figure 3 that although the value of our method is not the best in some data sets, the division result of the DPLPA is the same as the actual community distribution, which can be seen from Table 3 and Figure 4. This is mainly because in the process of label dissemination, the probability transition matrix well suppresses the randomness of the dissemination process, so that each update of the node is updated to the label of the same community node as much as possible, making the result of community division more stable and closer to the real community situation. Comparison of K values of different algorithms on different networks is shown in Table 4 and Figure 5.

In addition, from Figure 5, we find that the DPLPA can detect the true number of communities, which is completely consistent with the actual K value. This is mainly because the DPLPA begins to calculate the local density and distance of nodes through the topology of the network at the very beginning and selects the number of K values through a decision graph. Therefore, we do not need to provide the K value, and the DPLPA has the advantage of detecting the K value.

In order to better show the experimental results, we use the Karate network and the Dolphins network as case studies to visualize the detected communities. Nodes in the same community are divided by the same color. Figure 6 is the visualization result of DPLPA division of the Karate network. Figure 7 is the visualization result of the DPLPA division of the Dolphins network.

It can be seen from Figure 6 that the local density of node 1 and node 34 is the highest, and it can be seen from Figure 7 that the local density of node 15 and node 18 is the highest, and these nodes have higher node distance, so it is very reasonable for the DPLPA to select these nodes as K, and the result of the division is completely consistent with the result of the actual community division. Therefore, we believe that the DPLPA is an algorithm that can perform high-quality community detection in real communities. In order to observe the convergence of DPLPA, this paper makes a comparison in multiple data sets, as shown in Figure 8.

Where the X axis is the number of iterations, and Y axis is the number of changes during node label iteration, as can be seen from Figure 8, in the process of Karate and Doplhins data clustering, the DPLPA has completed the division of most node labels after the first few iterations and then completed the division of a few nodes. In the process of Polbook and Football data clustering, the labels of most nodes have been partitioned until the 30th iteration. After that, the change curve of node labels becomes flat, indicating that all nodes have completed the label division and the algorithm has converged.

5. Concluding Remarks

In this paper, we propose a DPLPA for complex network community detection. It combines the characteristics of density peak algorithm and can predict the number of communities without a prior condition. It avoids the defects of random label algorithm, such as unstable division and strong randomness, and effectively improves the accuracy of community mining and the stability of the algorithm. In addition, the probability transition matrix is constructed to reduce the number of iterations of label propagation, so that the algorithm has efficient operation time, and finally can quickly find the network community structure. In the test results of the benchmark network and the classical real network, it is found that the proposed algorithm has better stability and accuracy than other advanced algorithms, and the number of communities found is always consistent with the actual number of communities in terms of the predicted K value. However, there is still room for improvement of the algorithm. In future research, we will face large-scale network data and further improve the time complexity of the algorithm. At the same time, dynamic network and overlapping network are also taken as research objects.

Data Availability

The data used in "Label propagation community detection algorithm based on density peak optimization" is a commonly used data set to study complex networks, which can be queried on multiple data websites, for example: https://snap.stanford.edu/data/. http://konect.cc/http://konect.cc/. https://networkrepository.com/index.php. That is where I read the data I used in my experiments. New data availability url http://www-personal.umich.edu/~mejn/netdata/.

Disclosure

Ma Yan and Chen Guoqiang current address is School of Computer and Information Engineering, Henan University, Henan Province, China. This work was outlined at the 2021 17th International Conference on Computational Intelligence and Security (CIS) [38].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Key Science and Technology Program of Henan Province, China (Grant No. 162102210168). Group Name - on behalf of Key Science and Technology Program of Henan Province, China NO:162102210168 Affiliation - Belongs to Henan University, School of Computer and Information Engineering. Email Address - hnkjgg@163.com.

References

[1] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, 2002.

[2] L. Hongchao, Z. Xiaopeng, L. Haifeng et al., "The interactome as a tree–an attempt to visualize the protein-protein interaction network in yeast," *Nucleic acids research*, vol. 32, no. 16, pp. 4804–4811, 2004.

[3] M. E. J. Newman, "The structure of scientific collaboration networks," *Proceedings of the National Academy of Sciences*, vol. 98, no. 2, pp. 404–409, 2001.

[4] D. Li, S. Zhang, and X. Ma, "Dynamic module detection in temporal attributed networks of cancers," in *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2021.

[5] Z. Huang, Y. Wang, and X. Ma, "Clustering of cancer attributed networks by dynamically and jointly factorizing multilayer graphs," in *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2021.

[6] X. Ma, P. Sun, and M. Gong, "An integrative framework of heterogeneous genomic data for cancer dynamic modules based on matrix decomposition," *IEEE/ACM Transactions on Computational Biology and Bioinformatics.*, vol. 19, no. 1, pp. 305–316, 2022.

[7] W. Wenming, Z. Liu, and X. Ma, "jSRC: a flexible and accurate joint learning algorithm for clustering of single-cell RNA-sequencing data," *Briefings in Bioinformatics*, vol. 22, no. 5, pp. 1–15, 2021.

[8] W. Wenming and X. Ma, "Joint learning dimension reduction and clustering of single-cell RNA-sequencing data," *Bioinformatics*, vol. 36, no. 12, pp. 3825–3832, 2020.

[9] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, p. 75, 2010.

[10] W. Liu, J. Wang, and S. F. Chang, "Robust and scalable graph-based semi-supervised learning," *Proceedings of the IEEE*, vol. 100, no. 9, 2012.

[11] Y. Chong, Y. Ding, Q. Yan, and S. Pan, "Graph-based semi-supervised learning: a review," *Neurocomputing*, vol. 408, pp. 216–230, 2020.

[12] F. Nie, W. Zhu, and X. Li, "Structured graph optimization for unsupervised feature selection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 12, p. 1, 2019.

[13] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[14] R. U. Nandini, A. Réka, and K. Soundar, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*, vol. 76, no. 3, 2007.

[15] G. Tibély and J. Kertész, "On the equivalence of the label propagation method of community detection and a Potts model approach," *Physica A: Statistical Mechanics and its Applications*, vol. 387, no. 19-20, pp. 4982–4984, 2008.

[16] X. Y. Leung Ian, H. Pan, L. Pietro, and C. Jon, "Towards real-time community detection in large networks," *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*, vol. 79, no. 6, 2009.

[17] J. Barber Michael and J. W. Clark, "Detecting network communities by propagating labels under constraints," *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*, vol. 80, no. 2, 2009.

[18] X. Liu and T. Murata, "Advanced modularity-specialized label propagation algorithm for detecting communities in networks," *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 7, pp. 1493–1500, 2010.

[19] Z. Zhuoxiang, W. Yitong, T. Jiatang, and Z. Zexu, "A novel algorithm for community discovery in social networks based on label propagation," *Journal of Computer Research and Development*, vol. 48, 2011.

[20] J. Xie and B. K. Szymanski, "LabelRank: a stabilized label propagation algorithm for community detection in networks," in *2013 IEEE 2nd Network Science Workshop (NSW)*, West Point, NY, USA, 2013.

[21] Z. Lin, X. Zheng, N. Xin, and D. Chen, "CK-LPA: efficient community detection algorithm based on label propagation with community kernel," *Physica A: Statistical Mechanics and its Applications*, vol. 416, pp. 386–399, 2014.

[22] X.-K. Zhang, X. Tian, Y.-N. Li, and C. Song, "Label propagation algorithm based on edge clustering coefficient for community detection in complex networks," *International Journal of Modern Physics B*, vol. 28, no. 30, p. 1450216, 2014.

[23] F. T. N. Kip and M. Welling, *Semi-supervised classification with graph convolutional networks*, 2017, https://arxiv.org/abs/1609.02907.

[24] L. Page, S. Brin, R. Motwani, and T. Winograd, *The page rank citation ranking: bringing order to the web*, Stanford Digital Libraries Working Paper, 1999.

[25] Y. Li, H. Wang, J. Li, and H. Gao, "Efficient community detection with additive constrains on large networks," *Knowledge-Based Systems*, vol. 52, pp. 268–278, 2013.

[26] M. Qianli and Z. Junhao, "A local strengthened multi-label propagation algorithm for community detection," *Computer Engineering*, vol. 40, no. 6, pp. 171–174, 2014.

[27] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003)*, Washington, DC, USA, 2003.

[28] H. Bäcklund, A. Hedblom, and N. Neijman, *DBSCAN: a density-based spatial clustering of application with noise*, Linköpings Universitet-ITN, Data Mining TNM033, 2011.

[29] H. Lu, Q. Zhao, X. Sang, and J. Lu, "Community detection in complex networks using nonnegative matrix factorization and density-based clustering algorithm," *Neural Processing Letters*, vol. 51, no. 2, 2020.

[30] D. Martin, *Introduction to Modern Information Retrieval*, G. Salton and M. McGill, Eds., McGraw-Hill, New York, 1983.

[31] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*, vol. 69, no. 6, 2004.

[32] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.

[33] A. Estévez Pablo, T. Michel, A. Perez Claudio, and J. M. Zurada, "Normalized mutual information feature selection," *IEEE transactions on neural networks*, vol. 20, no. 2, pp. 189–201, 2009.

[34] L. Andrea, F. Santo, and R. Filippo, "Benchmark graphs for testing community detection algorithms," *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*, vol. 78, no. 4, article 046110, 2008.

[35] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of anthropological research*, vol. 33, no. 4, pp. 452–473, 1977.

[36] L. David, "The emergent properties of a dolphin social network," *Proceedings of the Royal Society B: Biological Sciences*, vol. 270, suppl_2, 2003.

[37] M. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, no. 23, pp. 8577–8582, 2006.

[38] Y. Ma and G. Chen, "Label propagation community detection algorithm based on density peak optimization," in *17th International Conference on Computational Intelligence and Security*, pp. 80–84, 2021.