Hindawi
Mobile Information Systems
Volume 2022, Article ID 4368829, 10 pages
https://doi.org/10.1155/2022/4368829

Research Article

Unsupervised Community Detection Algorithm Based on Graph Convolution Network and Social Media

Hua Zhou and Yusha Zhang

College of Computer Science and Engineering, Hunan University of Information Technology, Changsha, Hunan 410151, China

Correspondence should be addressed to Yusha Zhang; zhangyusha@hnuit.edu.cn

Received 31 March 2022; Revised 18 May 2022; Accepted 23 May 2022; Published 18 June 2022

Academic Editor: Fazli Wahid

In view of the difficulty and low efficiency of most existing algorithms in detecting large-scale community networks, an unsupervised community detection algorithm based on graph convolution networks and social media is proposed. First, some positive and negative sample nodes are labeled according to the node similarity to complete the graph segmentation. Then, the improved graph convolution network model is used for training to obtain the local community where the given starting node is located. Finally, the local community is optimized by setting the threshold of membership degree, so as to further screen the nodes outside the community and obtain accurate community detection results. The experimental analysis of the proposed algorithm based on Flixster, Douban, and Yelp datasets shows that when the number of community divisions is 12, the modularity values on the three datasets are 0.59, 0.62, and 0.69, respectively, and the standard deviations of F1 are 0.044, 0.048, and 0.040, respectively. Overall, the proposed unsupervised community detection algorithm has better robustness.

1. Introduction

In the real world, the network has the following characteristics: the edge distribution between vertices is uneven, and the vertices have unequal degrees of height. In addition, the distribution of edges is not only globally uneven but also locally uneven. The concentration of edges in special vertex groups is very high, while the concentration between these vertex groups is very low [1–3]. This feature of the network is called community structure. Community, also known as clustering or module, is a group of vertices that have common attributes or play similar roles in the network. There are various possible groups and organizations in a society, and communities can also appear in network systems in other fields such as biology, computer science, engineering, economics, political science, and so on. Community detection can determine the module and the boundary of the module, so as to classify the nodes [4, 5]. The node in the community center may have important control and stability functions in the community. Nodes at the edge of the module play an important role in mediation and guide the relationship and communication between different communities [6]. It can be seen that community detection plays an important role in network analysis [7].

At present, the research on the static network community detection has been relatively mature, and scholars have made many research achievements [8, 9]. The static community detection methods mainly include the methods based on modularity, label propagation, matrix decomposition, spectral clustering, local optimization, and so on [10]. Reference [11] proposed a graph-based label propagation algorithm. By detecting the node similarity and connectivity information included in the label propagation process, a label propagation graph was constructed to obtain the candidate community, and the connectivity component of the label propagation graph was calculated, which effectively realized the community detection. However, the algorithm does not consider the association between the node and the community, and the detection result has room for improvement. Reference [12] used the weighting method based on the social attributes of links and the degree of public adjacent nodes to update the fitness function, and

introduced the concept of stability to control the expansion of local communities, so as to complete accurate community detection, but the method is too simple and has poor universality. In order to improve the quality of community detection methods in practical use, reference [13] compared and studied six community detection methods. The results showed that Walktrap method produced the most communities that met the hypothetical intervention requirements and had good detection performance. Reference [14] proposed a probability matrix and an improved spectral clustering algorithm of probability matrix for community detection. Markov chain was used to calculate the transition probability between nodes and construct a probability matrix to realize community detection. The algorithm has good clustering performance and detection accuracy, but its detection reliability needs to be strengthened in the face of large-scale networks.

In recent years, deep learning has become a hot topic in artificial intelligence and machine learning. Many machine learning tasks, such as target detection, image classification, machine translation, and natural language processing, once relied heavily on manual feature engineering to extract rich feature information, and these have been completely changed by various deep learning models [15]. The success of deep learning is partly due to the effectiveness of extracting potential representations from Euclidean data. Therefore, many scholars also try to extend deep learning to network data. This kind of model is called Graph Neural Network (GNN) [16]. Reference [17] proposed a deep community detection method including matrix reconstruction and spatial feature extraction, which obtained the spatial adjacency matrix through the opinion leader and the original adjacency matrix in the nearest reconstructed social network. This method used the automatic encoder based on convolutional neural network to extract the spatial feature vector of the reconstructed adjacency matrix, which effectively improves the modularity of the algorithm. But due to lack of the analysis of social media, the overall detection accuracy is not high. Reference [18] proposed a method to analyze the performance of community detection algorithm using network visualization. Two algorithms were applied to four real-world networks with various characteristics to prove the usefulness and universality of this method. Reference [19] proposed a scheme to improve the existing community detection algorithm by considering the information topology and content. It can detect a more meaningful community structure in the network with incomplete information, but the discrimination of invalid information is not high, resulting in the unexpected detection effect. Reference [20] realized community mixing in any number of communities based on quantum annealing and gate quantum technology. The new hybrid algorithm produced modular values similar to classical heuristics, which have good detection effect and performance. However, the algorithm complexity is high.

Based on the above analysis, aiming at the problems of low detection accuracy and low robustness of existing algorithms, an unsupervised community detection algorithm based on graph convolution network and social media is proposed. In order to make the graph convolution neural network suitable for the unsupervised community division field, the proposed algorithm uses the fixed same weight to replace the training weight, improves the propagation rules, and optimizes the local community by setting the threshold of membership to further ensure the accuracy of community detection. After a series of processing such as graph segmentation, graph convolution network training, and result optimization, the ideal community detection results can be obtained, and the network complexity is simplified. The F1's standard deviation of the proposed algorithm is 0.044, 0.048, and 0.040 on the three datasets, respectively, which has better robustness.

2. System Model

The goal of local community detection is to find the community structure where the given starting node is located. Formally, the original graph G is divided into two parts: one part is the local community C containing the given node, and the other part is the remaining nodes in the network, that is, G–C. The nodes in the local community C are closely connected, while the nodes in the local community are sparsely connected with the nodes outside the local community [21]. The proposed community detection algorithm is different from the previous algorithms. Most of the traditional algorithms are based on greedy and intelligent evolution, and the proposed algorithm uses the semi-supervised learning of graph convolution network and social media to detect the community. The overall structure of the algorithm is shown in Figure 1.

The algorithm can be roughly divided into three stages: the first stage is graph segmentation and node labeling. The second stage is the training stage, that is, use the graph convolution network for training to obtain the local community. The third stage is result optimization, that is, deal with the obtained community to make the result better.

3. Graph Segmentation and Node Labeling

In many large complex networks, such as Taobao shopping network, there are a large number of communities, but people often pay attention not to the community structure of the whole network, but to the community where a node in the network is located. Therefore, except for some information around the node, other information in the network is irrelevant. Consequently, a graph segmentation method is proposed to remove the useless information in the network while keeping the concerned content unchanged [22].

According to the general definition of community, the nodes within the community are closely connected, while the connections between communities are relatively sparse. Then, due to the close connection between the nodes in the community, the distances between the nodes in the same community are very small. Based on this idea, a graph segmentation method is designed: cutting the K-order neighbor subgraph of a given starting node v_i in the input network. The specific description is shown in Algorithm 1.

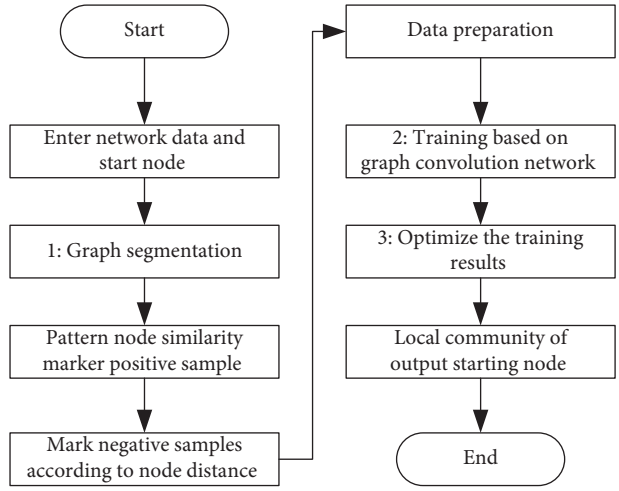

Figure 1: The overall structure of the proposed community detection algorithm.

Input: Network G; the given starting node v_i, the parameter K;
Output: Subgraph G_Ω.
Begin
(1) Node set: $G_\Omega = \{v_i\}$.
2. $\varphi = 1$; $N = \{v_i\}$
(2) **While** $\varphi < K$ **do**
Neigh = $\varnothing$
(3) **For** each $v \in N$
Neigh← add the neighbor nodes of v into neighbor set *Neigh*
End for
(4) $G_\Omega = G_\Omega \cup$ Neigh
(5) $N \leftarrow$ Neigh
(6) $\varphi = \varphi + 1$
(7) **End While**
(8) Construct a subgraph from nodes in set G_Ω.
End

Algorithm 1: Pseudo code of graph segmentation algorithm.

In Algorithm 1, the inputs are network G and the given starting node v_i, and the output is the segmented subgraph G_Ω. The algorithm first adds the given starting node v_i to the set G_Ω (line 1). Next, find all neighbor nodes of the node v_i, add them to the set G_Ω, and update the neighbor set N (line 2–6). Repeat the above process until the order reaches the given threshold K. The final output is the subgraph corresponding to the set G_Ω. It should be noted that in the later process, the degree of nodes in the segmented subgraph should also be maintained as the corresponding degree in the original graph G. The example of graph segmentation is shown in Figure 2, in which it is assumed that the given starting node is node 3 and the threshold K is 2.

After segmenting the original graph G, a smaller subgraph G_Ω is obtained. Before the semisupervised training of graph convolution network, it is also necessary to know the community labels of some nodes, so as to calculate the loss value and update parameters after the training of graph convolution network [23]. Therefore, the next main task is to label some positive sample nodes (in the community) and negative sample nodes (not in the community). For the labeling of positive samples, the local modularity function ϕ and the similarity between nodes and communities are used for guidance, and the positive samples do not need many, so the stop condition is set as $\phi < 1$, that is, the number of internal edges in the community is just greater than the number of external edges. For the selection of positive samples, select the node that can increase the local modularity ϕ and has the greatest similarity with the community $C(v_i)$, where the given starting node v_i is located as the positive sample. The specific measurement method is as follows:

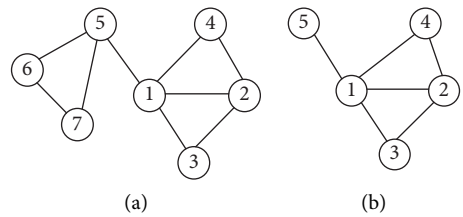

Figure 2: Graph segmentation example: (a) Before segmentation; (b) After segmentation.

Input: Network G_Ω; the given starting node v_i, Node distance τ;
Output: Negative sample set *NS*.
Begin
(1) Negative sample set: $NS = \varnothing$.
(2) Distance = 1; $N = \{v_i\}$
(3) **While**Distance < τ**do**
Neigh = $\varnothing$
(4) **For** each $v \in N$
Neigh← add the neighbor nodes of v into neighbor set *Neigh*
End for
(5) $N \leftarrow$ Neigh
(6) Distance = Distance + 1
(7) **End While**
(8) *NS*←Neigh.
End

Algorithm 2:Pseudo code of negative sample node labeling process.

$$\Delta\phi = \frac{e_{in} + x}{e_{out} - x + y} - \frac{e_{in}}{e_{out}}, \tag{1}$$

$$\mu_\phi(v) = \begin{cases} \max\limits_{v_j \in NC} \dfrac{\left|N(v) \cap N\left(v_j\right)\right| + 1}{\left|N\left(v_j\right)\right|}, & \Delta\phi \geq 0, \\ 0, & \Delta\phi < 0. \end{cases} \tag{2}$$

(1) represents the change of local modularity ϕ after a node is added to the local community $C(v_i)$, where e_{in} represents the number of internal edges of the local community $C(v_i)$, e_{out} represents the number of edges connected between the local community $C(v_i)$ and external nodes; x represents the number of neighbors of node v_i in the local community $C(v_i)$, and y represents the number of neighbors of node v_i not in the local community $C(v_i)$. According to the definition of x and y, the sum of x and y is just the degree of the node v_i. Equation (2) represents the membership degree of a node v outside the local community belonging to the local community, where $N(v)$ represents the neighbor node set of the node v; NC represents a collection of nodes that are neighbors of the node v and are in a local community. In this process, the node with the largest $\mu_\phi(v)$ value and satisfying $\Delta\phi \geq 0$ is labeled as a positive sample, that is, the node in the local community.

After labeling the positive samples, it is also necessary to label some negative sample nodes (not in the local community). The negative sample labeling process is shown in Algorithm 2. The strategy of negative sample selection makes use of the node difference. It is believed that when the two nodes are far away, the corresponding node difference is large. Therefore, a node far enough from a given starting node v_i can be selected as a negative sample. A distance parameter τ ($\tau < K$) is set here, and then the node with a distance of τ from the given starting node v_i is labeled as a negative sample.

After Algorithm 2, the node set NS with the distance τ from the given starting node v_i is obtained. The nodes in the set NS are labeled as negative samples. So far, the strategies of graph segmentation and node labeling have been given, and some labeled nodes $M_L = PS \cup NS$ can be obtained, where *PS* is the positive sample set.

4. Data Preparation and Training

4.1. Data Preparation. For the subgraph G_Ω obtained after graph segmentation, it is assumed that the number of nodes in the subgraph is N, the adjacency matrix is expressed as A, the degree matrix is expressed as **D** (the degree in the original graph **G** rather than the degree in the subgraph), and the feature matrix is expressed as F. Finally, a $N \times Q$ feature matrix $\widehat{F}$ is wanted to obtain, which represents the feature representation of each node. Q is the dimension of the features of each node. In the proposed algorithm, $Q = 2$. Then, for a neural network with L layers, the output of layer l

can be expressed as follows: $\mathbf{H}^{l+1} = f(\mathbf{H}^l, \mathbf{A})$, where $\mathbf{H}^0$ is the input feature matrix $\mathbf{F}$, $\mathbf{H}^L$ is the final output $\widehat{F}$.

In the traditional convolution, its operation can be regarded as adding the information of neighbor nodes to each node. Then, for the network, the same operation can also be completed using (3):

$$f\left(\mathbf{H}^l, \mathbf{A}\right) = \sigma\left(\mathbf{A}\mathbf{H}^l\omega^l\right), \tag{3}$$

where ω represents the weight. In (3), multiplying by the adjacency matrix $\mathbf{A}$ of the network is equivalent to adding the features of neighbor nodes to each node, but not adding the features of the node itself, unless the node has a self-ring. In order to solve this problem, each node is forcibly added with a self-ring. The value of the diagonal position of the adjacency matrix is set to 1, that is $\widetilde{A} = \mathbf{A} + \mathbf{I}$, where $\mathbf{I}$ represents the identity matrix, the diagonal elements of the matrix are all 1, and the values of other positions are 0. In addition, the changed matrix $\widetilde{A}$ is not regularized, so the matrix needs to be regularized so that the sum of each row in the matrix is 1. The regularization calculation is as follows:

$$\widehat{A} = \mathbf{D}^{-1/2}\widetilde{A}\mathbf{D}^{-1/2}. \tag{4}$$

After regularizing the matrix, equation (3) will be rewritten into the form of (5), where D represents the degree matrix, $\widetilde{A} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix -after adding the self-ring:

$$f\left(\mathbf{H}^l, \mathbf{A}\right) = \sigma\left(\mathbf{D}^{-1/2}\widetilde{A}\mathbf{D}^{-1/2}\mathbf{H}^l\omega^l\right). \tag{5}$$

However, there is another problem to be solved, that is, the $\mathbf{H}^l$ in equation (5) (feature matrix $\mathbf{F}$ in the input layer). In the dataset used, nodes do not have their own features, so it is necessary to construct the feature matrix of the network. The simplest way is to replace the feature matrix with the identity matrix, that is, $\mathbf{F} = \mathbf{I}$. Any matrix multiplied by the identity matrix remains unchanged, but the identity matrix cannot distinguish the features of each node, because the diagonal elements in the identity matrix are all 1. The proposed method adopts another construction method of feature matrix. The specific steps are as follows: the matrix $\widehat{A}$ in equation (4) is Eigen decomposed, $\widehat{A} = \mathbf{U}\Lambda\mathbf{U}^T$, where $\mathbf{U}$ is matrix composed of eigenvectors, Λ is the diagonal matrix composed of eigenvalues. Then, the diagonal elements of the diagonal matrix Λ composed of eigenvalues are deformed according to equation (6) to form the feature matrix $\mathbf{F}$.

$$f_i = e^{-\lambda_i}, \tag{6}$$

where f_i is the diagonal element of the feature matrix $\mathbf{F}$; λ_i is the eigenvalue of the feature matrix $\mathbf{F}$.

4.2. Training Based on Graph Convolution Network. In the training based on graph convolution network, the applicability of the propagation rules of graph convolution network is improved to make it suitable for unsupervised community division. At present, the graph convolution network uses completely random weight ω to participate operation in the supervised learning task. The weight value is adjusted through training to obtain the optimal model and predict the division results. The proposed community detection algorithm processes the data without original labels, which belongs to unsupervised learning. Each node has the same attributes by default [24, 25]. This means that each node has the same contribution to the graph structure data and focuses more on the topology of the graph data than other algorithms. Therefore, the fixed same weight is used to replace the training weight to ensure that the effect of community division is only related to the topology of the graph network. Since the weight value does not affect the result of community division in essence, the weight value is 1 for convenience of calculation. In fact, for models that do not need to train optimization functions or have the same nature of input data, the fixed weight is reasonable and can also reduce the computational complexity.

Based on this, the propagation rule of graph convolution network is modified as follows:

$$f\left(\mathrm{H}^l, \mathrm{A}\right) = \sigma\left(\mathrm{D}^{-1/2}\widetilde{A}\mathbf{D}^{-1/2}\mathbf{H}^l\right). \tag{7}$$

Because the data without label cannot simulate the input signal through its own information. Therefore, it is necessary to label some specific nodes in the graph to simulate the input signal Γ on the graph. In the algorithm, the input signal of the graph can be expressed as follows: $\Gamma = N \times \mathbb{C}$, where N is the number of nodes, and $\mathbb{C}$ is the number of clusters. Second, after the initial node of the simulate signal is selected by the method given in the algorithm, the parameters are shared by using the improved graph convolution network propagation rules.

Let v_i be the initial signal node, whose signal value is Γ_i, and v_j be the first-order neighbor node of v_i. The aggregation $\overline{\Gamma}$ defined here is the average of the signal value at the node i and its first-order neighbor node j, that is,

$$\text{Aggregate}\left(\overline{\Gamma}\right) = \sum_{j \in \text{The first-order neighbor of node } i} \frac{\mathbf{A}_{ij}\Gamma_i}{j+1}. \tag{8}$$

Based on the above aggregation definition, the parameter sharing process of graph convolution network propagation rule is explained as follows:

Rewrite the propagation rule (7) into the form of aggregation at node i of signal value and graph convolution network classifier:

$$\begin{aligned}
\text{Aggregate}\left(f, \overline{\Gamma}\right)_i &= \mathbf{D}^{1/2} * \widetilde{A} * \mathbf{D}^{1/2} * \Gamma_i \\
&= \sum_{k=1}^{N} \mathbf{D}_{i,k}^{-1/2} \sum_{j=1}^{N} \widetilde{A}_{ij}\Gamma_i \sum_{l=1}^{N} \mathbf{D}_{j,l}^{-1/2} \\
&= \sum_{i=1}^{N} \frac{1}{D_{i,i}^{1/2}} \widetilde{\mathbf{A}}_{ij} \frac{1}{\mathbf{D}_{j,j}^{1/2}} \Gamma_i \\
&= \sum_{i=1}^{N} \frac{\widetilde{A}_{ij}}{\sqrt{\mathbf{D}_{i,i}\mathbf{D}_{j,j}}} \Gamma_i.
\end{aligned} \tag{9}$$

It can be seen from equation (9) that the signal simulated by manual label realizes parameter sharing through the modified graph convolution network propagation rules. In fact, the essence of parameter sharing can be understood as a kind of weighted average. Because in unsupervised learning tasks, the input data do not have the original label. Therefore, the weighted average parameter sharing method between the manual label of the node and the neighbor node without the label can be regarded as a label transfer process. The propagated effective information value is the weighted average of artificial labels, and its weight is related to the degree (topology information) of nodes and their first-order neighbor nodes. The weighted average sharing process of effective information is shown in Figure 3.

For the case that training is not required and the input data attributes are the same, the fixed weight can ensure that the information transmission is only related to the topology of the graph. Therefore, the accuracy of information transmission can be guaranteed when the input adjacency matrix remains unchanged [26]. The artificial label of simulated signal nodes diffuses in this way. Each time it propagates to the first-order neighbor node, it can also be understood as the Laplace smoothing of the input signal in the graph. And because the purpose of the community detection algorithm is to find communities with strong internal correlation in the graph structure data, this division method based on the information transmission between nodes and their neighbor topology will naturally have better results. Moreover, this way of transmitting labels according to the topology of nodes is very similar to the way of information dissemination in sociology, so the community structure divided according to this method should also have a more real social nature.

In the community detection task, the node label should follow the invariable property of the equivalent replacement of the community, that is, the divided community is not affected by the specific meaning of the specific label. Through Softmax, the output of the improved graph convolution network obtains the prediction probability $\widehat{p}_i$ of the community to which each node belongs. $\mathbb{C}$ represents the label set of all communities and p_i represents the real probability of the community to which each node belongs. Therefore, the loss function is defined as

$$\text{Loss} = \underset{\pi \in S_{\mathbb{C}}}{INF} - \sum_{i \in V} \pi(p_i) \log(\widehat{p}_i), \tag{10}$$

where $S_{\mathbb{C}}$ is the permutation and combination set of all communities $\mathbb{C}$, π is one of the permutations in $S_{\mathbb{C}}$. Hypothesis $F1$: $\mathbf{F} \longrightarrow \mathbb{C}$ is the function mapping from the initial feature matrix $\mathbf{F}$ of the node to the real community $\mathbb{C}$. Hypothesis $F2$: $\mathbf{F} \longrightarrow \widehat{\mathbb{C}}$ is the function mapping from the initial feature matrix $\mathbf{F}$ of the node to the model prediction community $\widehat{\mathbb{C}}$, and $\mathbf{f}_i$ represents the eigenvector of each node in the matrix $\mathbf{F}$. The final loss function is calculated as follows:

$$\text{Loss} = \underset{\pi \in S_{\mathbb{C}}}{INF} - \sum_{i \in V} F_1(\pi(\mathbf{f}_i)) \log(F_2(\pi(\mathbf{f}_i))). \tag{11}$$

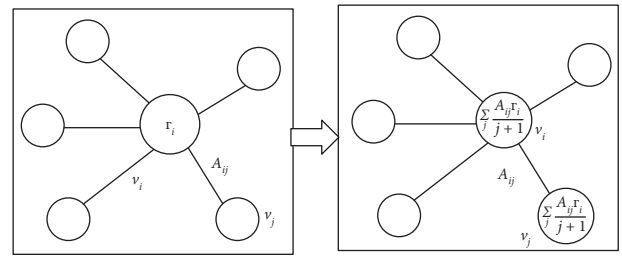

Figure 3: Parameter sharing process of one time weighted average.

The ultimate goal of the loss function is to take the minimum cross-entropy loss on all possible permutations of $\mathbb{C}$, so as to update the parameters in the network model.

5. Optimization of Training Results

After training, a size of $N \times 2$ classification matrix $\mathbf{F}$ is obtained. According to the matrix $\mathbf{F}$, whether each node belongs to a local community can be judged. The specific method is as follows: for the two-dimensional feature of each node, if the value of the first dimensional feature is greater than the value of the second dimensional feature, the node is in the local community; On the contrary, the node does not belong to the local community. Specifically, the mathematical expression for judging whether a node n belongs to a local community is as follows:

$$\Psi(n) = \begin{cases} \mathbf{F}_{n1} > \mathbf{F}_{n2}, & n \in C(v_i), \\ \mathbf{F}_{n1} \leq \mathbf{F}_{n2}, & n \notin C(v_i). \end{cases} \tag{12}$$

The above equation can classify each node and finally get the desired local community, but there is a problem when classifying. When the membership degree of a node n belonging to a local community is much greater than that not belonging to a local community, that is $\mathbf{F}_{n1} \gg \mathbf{F}_{n2}$, it is no problem to judge the node n as in a local community. However, when the two membership degrees are very close, that is $\mathbf{F}_{n1} \approx \mathbf{F}_{n2}$, although the category of the node can still be judged from the size, it is unreasonable to divide it into local communities [27]. Therefore, the threshold control method is adopted here. Only when the membership degree of the node belonging to the local community is greater than the threshold, it can be added to the local community, as follows:

$$\Psi(n) = \begin{cases} \mathbf{F}_{n1} > \mathbf{F}_0, & n \in C(v_i), \\ \mathbf{F}_{n1} \leq \mathbf{F}_0, & n \notin C(v_i). \end{cases} \tag{13}$$

For the setting of threshold $\mathbf{F}_0$, according to the division result of (12), add all the membership degrees belonging to local communities and calculate the average value as the threshold. The specific calculation is as follows:

$$\mathbf{F}_0 = \sum_{n \in C(v_i)} \frac{\mathbf{F}_{n1}}{|C(v_i)|}. \tag{14}$$

So far, the local community can be redivided according to $\mathbf{F}_0$, and the result of redivision is the local community

where the final given starting node v_i is located. First, the membership degrees of nodes whose first dimensional feature (membership degree belonging to local community) is greater than 0.5 are summed, and the threshold $\mathbf{F}_0$ is calculated. Then, according to the threshold, all nodes with membership degrees greater than the threshold are added to the local community. Finally, the set Ψ of nodes in the local community is output.

6. Experiment and Analysis

The hardware configuration of the experiment is: 8G memory, i5-4200H dual core processor, and the operating system is Windows10. In the experiment, the effectiveness of the proposed algorithm is verified on three real datasets: Flixster, Douban, and Yelp.

(1) Flixster dataset: Flixster is a platform that provides users with movie reviews and movie recommendations. Users can also add others to their friends list to form a friend relationship network. The dataset contains the user's rating of the movie and the friend relationship information of each user. Among them, the user's rating value of the movie is between 0.5 and 5, and the rating interval is 0.5.

(2) Douban dataset: Douban is a famous Chinese social networking site in China. It provides users with a platform to freely evaluate and recommend movies, books, and music. Users express their preference for the object by scoring the object (1–5 points). At the same time, users can establish friends with others, so as to know the behavior of friends on Douban. In order to evaluate the recommendation quality of the model, experiments are carried out on the dataset of Douban film, which has both the scoring data of users on the film and the social relationship between users.

(3) Yelp dataset: Yelp is the largest comment website in the United States, which includes merchants in various fields such as restaurants, hotels, and scenic spots. Yelp users can score merchants on the website, submit comments, share experiences with friends, among others. A total of 4506 users with more than 15 score records and 4483 subjects were randomly selected for the experiment.

The details of the datasets used in the experiment are shown in Table 1. The number of users and items in each dataset, the sparsity of scores, and the number of relationships among all users are counted in the table.

6.1. Evaluation Index. For unsupervised community detection algorithm, recall, precision, and F1 are used to evaluate its performance. The specific calculation method is as follows:

Table 1: Dataset statistics.

Dataset	Flixster	Douban	Yelp
User	8594	2408	4554
Item	11023	36289	4438
Rating	154077	750130	42185
Sparsity	1.8%	9.1%	2.3%
User_links	89022	19921	244639

$$\begin{aligned} \text{recall} &= \frac{\left|T_{\text{Found}} \cap T_{\text{True}}\right|}{\left|T_{\text{True}}\right|} \\ \text{precision} &= \frac{\left|T_{\text{Found}} \cap T_{\text{True}}\right|}{\left|T_{\text{Found}}\right|} \\ \text{F1} &= \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}, \end{aligned} \tag{15}$$

where T_{Found} represents the community detected by the algorithm and T_{True} represents the real community where the given node is located. As can be seen from the above equations, recall is the proportion of the correct nodes detected by the algorithm in the real community, precision is the proportion of the correct nodes detected by the algorithm in all the nodes detected in the community, and F1 is the combination of recall and precision. The value range of the three indexes is 0–1, and the larger the value, the better the detection result.

6.2. Comparison Chart of Community Division Results. Modularity can be used to measure the quality of a community division. The greater the modularity, the better the effect of community division. For the three datasets, the effect of different community division numbers compared with other types of community division algorithms is shown in Figure 4. In the experiment, the community division result of graph convolution network is to run the propagation rules of five-layer graph convolution network under the same division number, calculate the modularity, and take the optimal value as the result comparison.

As can be seen from Figure 4, for Flixster, Douban, and Yelp datasets, the proposed algorithm has the largest modularity compared with other algorithms, and its modularity is the most stable with the change of the number of community divisions. Taking the number of community divisions as 12 as an example, the modularity values of the proposed algorithm on the three datasets are 0.59, 0.62, and 0.69, respectively. Based on graph segmentation and node labeling, the proposed algorithm uses the improved graph convolution network to obtain local communities and optimize the detection results, which ensures the detection effect to a great extent. At the same time, the detection effect on Yelp dataset is generally better. That is because the dataset has a high degree of user association and low sparsity of scoring information, which is conducive to better

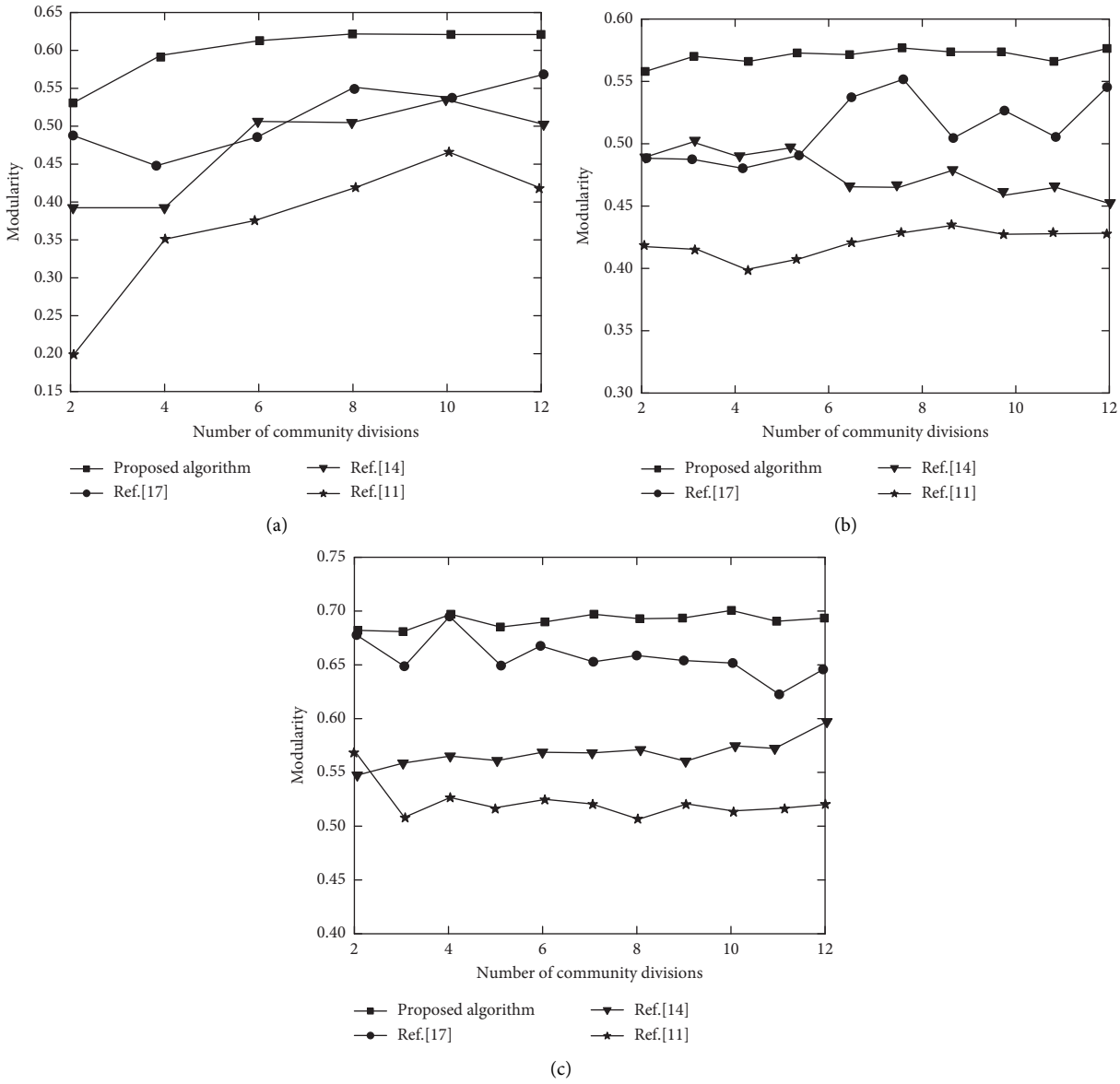

Figure 4: Comparison of community division results of different datasets: (a) Comparison of community division results on Flixster dataset. (b) Comparison of community division results on Douban dataset. (c) Comparison of community division results on Yelp dataset.

unsupervised community detection. Reference [11, 14], respectively, adopt the label propagation method of graph and the improved spectral clustering method. Both of them are traditional methods, and the effects of community detection are poor. Taking Flixster dataset as an example, their modularity are mostly below 0.50. Reference [17] realizes community detection through convolutional neural network, and its detection effect is improved compared with reference [11, 14], but social media is not considered. Therefore, the detection effect in complex communities is not ideal. Taking the number of community divisions as 7 under Douban dataset as an example, its modularity is reduced by about 0.3 compared with the proposed algorithm. In general, the proposed algorithm has good applicability in most cases and has a relatively wider application space.

6.3. Comparison of Algorithm Robustness. The adaptability of the algorithm is verified by calculating the standard deviation of the results of each algorithm in different datasets. The results are shown in Table 2.

It can be found from Table 2 that compared with other algorithms, the standard deviations of recall, precision, and F1 of the proposed algorithm are the smallest. Taking the F1's standard deviation of Flixster, Douban, and Yelp datasets as an example, which is 0.044, 0.048, and 0.040, respectively, so the applicability of the proposed algorithm is stronger. The algorithms in references [11, 14] are more traditional, and the results are volatile, so the standard deviations are large. Taking Flixster dataset as an example, the standard deviation of recall is close to 15–20 times that of the proposed algorithm. Reference [15] uses deep learning

Table 2: Comparison of algorithm robustness.

Dataset	Standard deviation	Ref. [11]	Ref. [14]	Ref. [17]	Proposed algorithm
Flixster	recall	0.212	0.169	0.095	0.037
	precision	0.128	0.106	0.081	0.066
	F1	0.187	0.134	0.072	0.044
Douban	recall	0.236	0.179	0.101	0.042
	precision	0.115	0.097	0.078	0.067
	F1	0.224	0.142	0.069	0.048
Yelp	recall	0.141	0.126	0.067	0.039
	precision	0.103	0.101	0.087	0.063
	F1	0.102	0.094	0.065	0.040

network to realize community detection, and the effect is good, but in the face of large-scale complex network, the detection performance is not ideal. Because the large complex network contains a lot of useless information, on the basis of graph segmentation, the proposed algorithm uses the local modularity function and the similarity between nodes and communities to identify the positive sample nodes, and uses the differences of nodes to distinguish the negative sample nodes, which simplifies the network and improves the adaptability of the algorithm while ensuring that the concerned content remains unchanged. The proposed algorithm has better detection accuracy and robustness.

7. Conclusion

Nowadays, the real world is composed of various networks, so using the community in the network to find valuable information has become a hot research issue. However, most of the existing community detection algorithms use the global information of the network or need a priori information, resulting in poor detection effect. Therefore, an unsupervised community detection algorithm based on graph convolution network and social media is proposed. On the basis of graph segmentation and node labeling, the improved graph convolution network model is trained to obtain the local community, and the threshold of membership degree is used to judge whether there are nodes outside the community in the local community, so as to further optimize the detection results. The experimental results based on Flixster, Douban, and Yelp datasets show that the improved graph convolution network with fixed weight and one-time weighted average parameter sharing can better divide communities. In the next work, the focus is on finding a selection method of better initial node, solving the possible abnormal division of some nodes in the division results, and considering the information contained in the edge in the process of community division, so as to further improve the precision and speed of community detection results.

Data Availability

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

Conflicts of Interest

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

Acknowledgments

This work was supported by the Natural Science Foundation Project of Hunan Province (No. 2020JJ5397), and the Educational Science Research of Hunan Province (No. 20B409).

References

[1] R. George, K. Shujaee, M. Kerwat, Z. Felfli, D. Gelenbe, and K. Ukuwu, "A comparative evaluation of community detection algorithms in social networks," *Procedia Computer ence*, vol. 171, no. 3, pp. 1157–1165, 2020.

[2] M. Zhang, Z Yang, and S Ali, W Ding, Web page information extraction service based on graph convolutional neural network and multimodal data fusion," in *Proceedings of the 2021 IEEE International Conference on Web Services (ICWS)*, pp. 681–687, Chicago, IL, USA, September 2021.

[3] X. Xu, N. Hu, M. Trovati, J Ray, F Palmieri, and H. M Pandey, "DLCD-CCE: a local community detection algorithm for complex IoT networks," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4607–4615, 2020.

[4] D. Murphy, G. Wittemyer, M. D. Henley, and H. S Mumby, "Detecting community structure in wild populations: a simulation study based on male elephant, Loxodonta africana, data," *Animal Behaviour*, vol. 174, no. 3, pp. 127–148, 2021.

[5] X. Liu, Y. Du, M. Jiang, and X Zeng, "Multiobjective particle swarm optimization based on network embedding for complex network community detection," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 2, pp. 437–449, 2020.

[6] T. Ji, C. Luo, Y. Guo, Q. Wang, L. Yu, and P. Li, "Community detection in online social networks: a differentially private and parsimonious approach," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 1, pp. 151–163, 2020.

[7] A. Jz, A. Yl, W. B. Hao et al., "A no self-edge stochastic block model and a heuristic algorithm for balanced anti-community detection in networks," *Information Sciences*, vol. 518, no. 4, pp. 95–112, 2020.

[8] Z. Peng, M. Rastgari, Y. D. Navaei et al., "TCDABCF: a trust-based community detection using artificial bee colony by feature fusion," *Mathematical Problems in Engineering*, vol. 2021, no. 4, pp. 1–19, Article ID 6675759, 2021.

[9] K. Asmi, D. Lotfi, and A. Abarda, "The greedy coupled-seeds expansion method for the overlapping community detection in social networks," *Computing*, vol. 4, no. 4, pp. 1–19, 2021.

[10] C. Durán, A. Muscoloni, and C. V. Cannistraci, "Geometrical inspired pre-weighting enhances Markov clustering community detection in complex networks," *Applied Network Science*, vol. 6, no. 1, pp. 1–16, 2021.

[11] G. Yang, W. Zheng, C. Che, and W Wenjian, "Graph-based label propagation algorithm for community detection," *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 6, pp. 1319–1329, 2020.
[12] G. Xu, X. Wu, and J. Liu, "A community detection method based on local optimization in social networks," *IEEE Network*, vol. 34, no. 4, pp. 42–48, 2020.
[13] P. N. Zivich, N. R. Smith, L. M. Frerichs, M James, and E. A Allison, "A guide for choosing community detection algorithms in social network studies: the question alignment approach," *American Journal of Preventive Medicine*, vol. 59, no. 4, pp. 597–605, 2020.
[14] S. Ren, S. Zhang, and T. Wu, "An improved spectral clustering community detection algorithm based on probability matrix," *Discrete Dynamics in Nature and Society*, vol. 2020, no. 8, pp. 1–6, Article ID 4540302, 2020.
[15] S. Shalileh and B. Mirkin, "Summable and nonsummable data-driven models for community detection in feature-rich networks," *Social Network Analysis and Mining*, vol. 11, no. 1, pp. 1–23, 2021.
[16] X. You, Y. Ma, and Z. Liu, "A three-stage algorithm on community detection in social networks," *Knowledge-Based Systems*, vol. 187, no. 6, Article ID 104822.1, 2020.
[17] L. Wu, Q. Zhang, C. H. Chen, K Guo, and D Wang, "Deep learning techniques for community detection in social networks," *IEEE Access*, vol. 8, no. 99, Article ID 96016, 2020.
[18] C. Linhares, J. R. Ponciano, F. Pereira, E. C. R Luis, G. D. S. P José, and A. N. T Bruno, "Visual analysis for evaluation of community detection algorithms," *Multimedia Tools and Applications*, vol. 79, no. 25, pp. 17645–17667, 2020.
[19] A. Bhih, P. Johnson, and M. Randles, "An optimisation tool for robust community detection algorithms using content and topology information," *The Journal of Supercomputing*, vol. 76, no. 1, pp. 226–254, 2020.
[20] F. G. Gemeinhardt, R. Wille, and M. Wimmer, "Quantum k-community detection: algorithm proposals and cross-architectural evaluation," *Quantum Information Processing*, vol. 20, no. 9, pp. 1–21, 2021.
[21] D. Yla, W. B. Qi, W. C. Xiao et al., "Community enhanced graph convolutional networks—ScienceDirect," *Pattern Recognition Letters*, vol. 138, no. 3, pp. 462–468, 2020.
[22] R. Agrawal, M. Arquam, and A. Singh, "Community detection in networks using graph embedding," *Procedia Computer Science*, vol. 173, no. 8, pp. 372–381, 2020.
[23] F. Lei, X. Liu, Z. Li, D Qingyun, and W Senhong, "Multihop neighbor information fusion graph convolutional network for text classification," *Mathematical Problems in Engineering*, vol. 4, no. 1, pp. 1–9, Article ID 6665588, 2021.
[24] Y. Yoon, J. Yu, and M. Jeon, "Predictively encoded graph convolutional network for noise-robust skeleton-based action recognition," *Applied Intelligence*, vol. 52, no. 8, pp. 1–15, 2021.
[25] L. Leng, J. Li, H. Shi, and Y Zhu, "Graph convolutional network-based reinforcement learning for tasks offloading in multi-access edge computing," *Multimedia Tools and Applications*, vol. 80, no. 19, Article ID 29163, 2021.
[26] S. Chang, C. Zhao, Y. Li, Z Min, F Chuan, and Q Honglin, "Multi-channel graph convolutional network based end-point element composition prediction of converter steelmaking," *IFAC-PapersOnLine*, vol. 54, no. 3, pp. 152–157, 2021.
[27] J. Zhang, Q. He, and Y. Zhang, "Syntax grounded graph convolutional network for joint entity and event extraction," *Neurocomputing*, vol. 422, no. 4, pp. 118–128, 2021.