

Article

Deep Community Detection Method for Social Networks

Ling Wu ^{1,2,3}, Qishan Zhang ^{2,*}, and Chi-Hua Chen ^{1,3,*}

¹ College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350116, China; wuling1985@fzu.edu.cn (L.W.); chihua0826@fzu.edu.cn (C.C.)

² School of Economics and Management, Fuzhou University, Fuzhou 350116, China; wuling1985@fzu.edu.cn (L.W.); zhangqs@fzu.edu.cn (Q.Z.)

³ Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou University, Fuzhou 350116, China; wuling1985@fzu.edu.cn (L.W.); chihua0826@fzu.edu.cn (C.C.)

* Correspondence: zhangqs@fzu.edu.cn; chihua0826@fzu.edu.cn; Tel.: +86-138-5918-3858

Abstract: With the fast development of the mobile Internet, the online platforms of social networks have rapidly been developing for the purpose of making friends, sharing information, etc. In these online platforms, users being related to each other forms social networks. Literature reviews have shown that social networks have community structure. Through the studies of community structure, the characteristics and functions of networks structure and the dynamical evolution mechanism of networks can be used for predicting user behaviours and controlling information dissemination. Therefore, this study proposes a deep community detection method which includes (1) matrix reconstruction method, (2) spatial feature extraction method and (3) community detection method. The original adjacency matrix in social network is reconstructed based on the opinion leader and nearer neighbors for obtaining spatial proximity matrix. The spatial eigenvector of reconstructed adjacency matrix can be extracted by an auto-encoder based on convolution neural network for the improvement of modularity. In experiments, four open datasets of practical social networks were selected to evaluate the proposed method, and the experimental results show that the proposed deep community detection method obtained higher modularity than other methods. Therefore, the proposed deep community detection method can effectively detect high quality communities in social networks.

Keywords: community detection; social network; convolutional neural network; auto-encoder

1. Introduction

Graph is an important data representation of complex networks. Effective community detection provides users a deeper understanding of networks, and it can benefit a lot of useful applications such as node classification, node recommendation, link prediction etc. However, the large scale and high-dimensionality of networks make community detection method suffer from high computation and space cost [1]-[5]. Network embedding is an effective yet efficient way to convert graph data into a low dimensional space, in which the network structural information and network properties are maximally preserved.

For community detection, some studies used auto-encoder (AE) algorithm [6]-[11] to analyze the adjacency matrix in a social network, reduce the dimension of social network, and extract the features of network topology [12]-[15]. The fully-connected strategy is applied for generating the auto-encoder networks, some noises may be obtained in the fully-connected networks when the unrelated nodes of social network exist. For the analyses of network topology, the convolutional neural network (CNN) algorithm is a powerful tool to extract the spatial eigenvector of network topology [16]-[20]. However, applying convolutional neural networks to detect social community has not been investigated.

Therefore, this study proposes a deep community detection method which includes (1) matrix reconstruction method, (2) spatial feature extraction method and (3) community detection method (shown in **Figure 1**). Due to limited cyberspace information in original adjacency matrix to present the structure of network topology, the proposed matrix reconstruction method based on a novel cyberspace structure reconstruction strategy is applied to find the opinion leader in the social network and find the nearer neighbors for reconstructing adjacency matrix with the spatial proximity matrix. Furthermore, this study also proposes a spatial feature extraction method based on auto-encoder and convolutional neural network to extract the spatial features of the reconstructed adjacency matrix. Finally, the extracted spatial features are adopted into the proposed community detection method for clustering the nodes of social network into k groups.

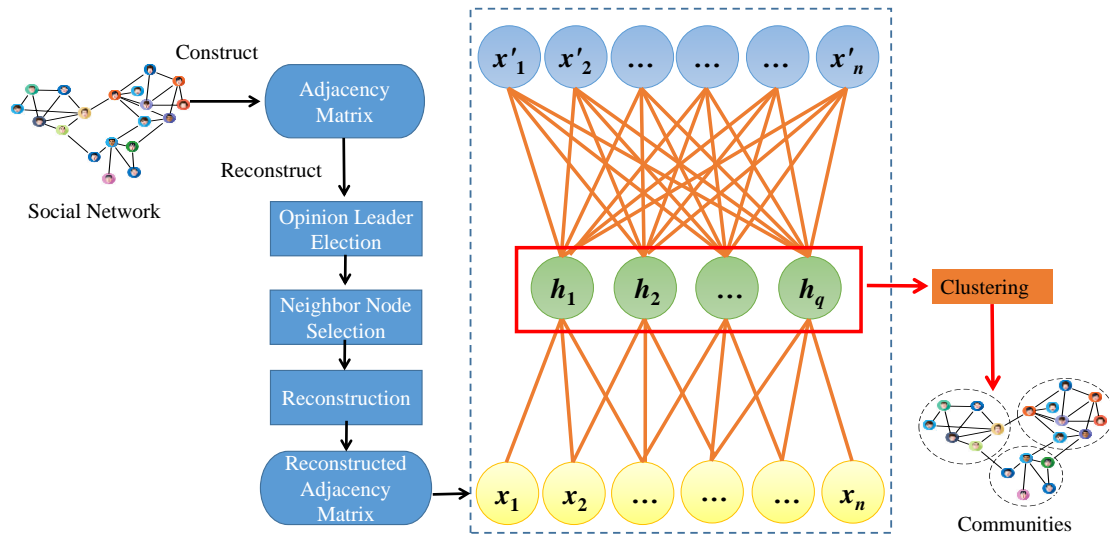


Figure 1. The proposed deep community detection method

The contributions of this study are summarized as follows.

- (1). A matrix reconstruction method based on a novel cyberspace structure reconstruction strategy is proposed to obtain the spatial proximity matrices of social networks.
- (2). A spatial feature extraction method based on auto-encoder and convolutional neural network is proposed to learn spatial eigenvector and extract the spatial features of social networks.
- (3). The matrix reconstruction method and spatial feature extraction method are applied to analyze the topology of social network for the improvement of social community detection.

The remainder of the paper is organized as follows. Section 2 presents the processes of the proposed deep community detection method to analyze the structure of social networks for extracting features of social networks and detecting social communities. The proposed methods will be evaluated, and practical experimental results will be analyzed and discussed in Section 3. The conclusions and future research directions are summarized in Section 4.

2. Deep Community Detection Method

This section presents the concepts of the proposed matrix reconstruction method, spatial feature extraction method and community detection method.

2.1. Matrix Reconstruction Method

Because the original order of nodes in an adjacency matrix is nonsense, a matrix reconstruction method is proposed to adjust the order of nodes in the adjacency matrix for arranging the neighbor nodes with high correlation. Therefore, some spatial features among nodes may be formatted after reconstructing the adjacency matrix for extracting spatial features and detecting social community.

The proposed matrix reconstruction method consists of three sub-methods which include (1) opinion leader election method, (2) neighbor node selection method and (3) reconstruction method described in the following subsections.

2.1.1. Opinion Leader Election Method

The opinion leader who is an important member in a group (i.e., an important node in an adjacency matrix) may affect the attitudes of other members. In practical social networks, several members may follow and be a friend of the opinion leader in a community. Therefore, the proposed opinion leader election method will analyze the influence of each node to find the most important node (i.e., the most influential opinion leader) in the adjacency matrix as an initial node for matrix reconstruction.

In this study, an adjacency matrix \mathbf{A} is constructed according to the edges between each two nodes. The parameter V (i.e., $\{v_1, v_2, \dots, v_n\}$) denotes as a node set, and the parameter E denotes as an edge set. If the edge between node v_i and node v_j exists, the value of $a_{i,j}$ is 1; otherwise, the value of $a_{i,j}$ is 0 (shown in Equation (1)).

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & L & a_{1,n} \\ a_{2,1} & a_{2,2} & L & a_{2,n} \\ M & M & O & M \\ a_{n,1} & a_{n,2} & L & a_{n,n} \end{bmatrix}, \text{ where } a_{i,j} = \begin{cases} 1, e_{i,j} \in E \\ 0, e_{i,j} \notin E \end{cases} \quad (1)$$

A transition probability matrix \mathbf{C} (shown in Equation (2)) can be constructed according to the adjacency matrix \mathbf{A} . When a node connects to several nodes, the weight of each edge is lower; when a node connects to less nodes, the weight of each edge is higher. For instance, if node v_i only connects to node v_j , node v_j is an important node for node v_i , and the transition probability $c_{i,j}$ is 1.

$$\mathbf{C} = \begin{bmatrix} c_{1,1} & c_{1,2} & L & c_{1,n} \\ c_{2,1} & c_{2,2} & L & c_{2,n} \\ M & M & O & M \\ c_{n,1} & c_{n,2} & L & c_{n,n} \end{bmatrix}, \text{ where } c_{i,j} = \frac{a_{i,j}}{\sum_{k=1}^n a_{i,k}} \quad (2)$$

In initial stage, the weight of each node is 1, and a node weight matrix \mathbf{S} is constructed in Equation (3). The limiting weight matrix \mathbf{S}^* is calculated based on the transition probability matrix \mathbf{C} (shown in Equation (4)). Finally, the node i_leader with the highest weight (i.e., opinion leader) is found according to Equation (5).

$$\mathbf{S} = [s_1 \quad s_2 \quad \dots \quad s_n], \text{ where } s_i = 1 \text{ in initial stage} \quad (3)$$

$$\begin{aligned} \mathbf{S}^* &= \lim_{m \rightarrow \infty} \mathbf{S} \times \mathbf{C}^m \\ &= \lim_{m \rightarrow \infty} [s_1 \quad s_2 \quad L \quad s_n] \times \begin{bmatrix} c_{1,1} & c_{1,2} & L & c_{1,n} \\ c_{2,1} & c_{2,2} & L & c_{2,n} \\ M & M & O & M \\ c_{n,1} & c_{n,2} & L & c_{n,n} \end{bmatrix}^m \\ &= [s_1^* \quad s_2^* \quad L \quad s_n^*] \end{aligned} \quad (4)$$

$$i_leader = \arg \max_{1 \leq i \leq n} s_i^* \quad (5)$$

The pseudo codes of the proposed opinion leader election method are presented in **Algorithm 1**.

Algorithm 1. Opinion leader election method

Inputs: an adjacency matrix $\mathbf{A}_{n \times n}$ with n nodes and the number of iterations I_{\max}

Outputs: the node ID of opinion leader

- (1) Construct the adjacency matrix $\mathbf{A}_{n \times n}$.
 - (2) **for** $i = 1$ to n
 - (3) **for** $j = 1$ to n
 - (4) Determine the weight of edge from node v_i to node v_j based on the amount of connections of node v_i .
 - (5) **end for**
 - (6) **end for**
 - (7) **for** $i = 1$ to n
 - (8) Set the weight of node v_i in the initial stage as 1.
 - (9) **end for**
 - (10) **while**(the number of iterations is less than I_{\max})
 - (11) **for** $i = 1$ to n
 - (12) Set the weight of node v_i in the next iteration as 0.
 - (13) **for** $j = 1$ to n
 - (14) Calculate the weight of node v_j multiplied by the weight edge from node v_j to node v_i as the value of temp.
 - (15) Set the weight of node v_i in the next iteration plus the value of temp.
 - (16) **end for**
 - (17) **end for**
 - (18) **end while**
 - (19) Return the ID of the node with a highest weight.
-

2.1.2. Neighbor Node Selection Method

After finding the opinion leader (i.e., node i_leader), a neighbor node selection method is proposed to find the neighbor node $j_neighbor$ which is highest relevant to node i_leader .

The node i_leader is presented as node v_i , and an Euclidean distance $r(i, j)$ is applied to measure the distance between node v_i and node v_j based on Equation (6). The node $j_neighbor$ with the shortest distance (i.e., the nearest neighbor) is found according to Equation (7).

$$r(i, j) = \sqrt{\sum_{k=1}^n d(i, j, k)^2}, \text{ where } d(i, j, k) = a_{i,k} - a_{j,k} \quad (6)$$

$$j_neighbor = \arg \min_{1 \leq j \leq n} r(i, j), \text{ where } j \neq i \quad (7)$$

2.1.3. Reconstruction Method

The order of nodes in the adjacency matrix \mathbf{A} can be determined based on the proposed opinion leader election method and neighbor node selection method. The opinion leader node can be elected as the first node, and the nearest neighbor of the opinion leader node can be elected as the second node; then the nearest neighbor of the second node can also be elected by the neighbor node selection method in accordance with Equations (6) and (7), and so on. The adjacency matrix \mathbf{A} can be reconstructed as matrix \mathbf{X} (shown in Equations (8)) by **Algorithm 2**.

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \text{L} & x_{1,n} \\ x_{2,1} & x_{2,2} & \text{L} & x_{2,n} \\ \text{M} & \text{M} & \text{O} & \text{M} \\ x_{n,1} & x_{n,2} & \text{L} & x_{n,n} \end{bmatrix}, \text{ where } x_{i,j} = \begin{cases} 1, & e_{i,j} \in E \\ 0, & e_{i,j} \notin E \end{cases} \quad (8)$$

Algorithm 2. Opinion leader election method

Inputs: an adjacency matrix $A_{n \times n}$ with n nodes

Outputs: the reconstructed adjacency matrix $X_{n \times n}$

- (1) Construct the adjacency matrix $A_{n \times n}$.
 - (2) Create the list L_n .
 - (3) Create the list O_n .
 - (4) **for** $i = 1$ to n
 - (5) Push node v_i into the list L_n .
 - (6) **end for**
 - (7) Set the node i_leader from the opinion leader election method as node v_i .
 - (8) **while**(the length of the list L_n is larger than 0)
 - (9) Pull node v_i from the list L_n .
 - (10) **for** $j = 1$ to n
 - (11) Calculate the Euclidean distance between node v_i and node v_j .
 - (12) **end for**
 - (13) Find and Set node $j_neighbor$ from the neighbor node selection method as node v_i .
 - (14) Push node v_i into the list O_n .
 - (15) **end while**
 - (16) Construct the adjacency matrix $X_{n \times n}$ according to the list O_n .
 - (17) Return the reconstructed adjacency matrix $X_{n \times n}$.
-

2.2. Spatial Feature Extraction Method

Due to the convolutional neural network as an excellent deep learning model for spatial analyses, this study proposes an auto-encoder based on convolutional neural network to automatically extract the spatial features of the reconstructed adjacency matrix for social networks. For building the structure of the proposed auto-encoder based on convolutional neural network, the number of neurons in the input layer is same with the number of neurons in the output layer. The spatial features in social networks can be extracted by the convolutional layer, so this study adopts the values of neurons after convolutional computation as spatial feature vectors.

2.2.1. The Description of a Simple Case

In this case, both of input layer and output layer have four neurons (i.e., four nodes in a social network), respectively. The size of filter in the convolutional layer (i.e., the first hidden layer) is 1×3 , so two neurons are trained in the hidden layer. The structure of auto-encoder based on convolutional neural network is showed in **Figure 2**.

Each parameter in the neural network is described as follows.

- The weight set of filter in the convolutional layer is $\{\alpha_1, \alpha_2, \alpha_3\}$.
- The adjustment variables of the two neurons in the convolutional layer are $\{b_{1,1}, b_{1,2}\}$.
- The weight set between the convolutional layer and the output layer is $\{\beta_{1,1}, \beta_{1,2}, \beta_{1,3}, \beta_{1,4}, \beta_{2,1}, \beta_{2,2}, \beta_{2,3}, \beta_{2,4}\}$.
- The adjustment variables of the four neurons in the output layer are $\{b_{2,1}, b_{2,2}, b_{2,3}, b_{2,4}\}$.
- The two neurons in the convolutional layer which are $\{h_1, h_2\}$ can be calculated by Equations (9) and (10).
- The four neurons in the output layer which are $\{x'_1, x'_2, x'_3, x'_4\}$ can be calculated by Equations (11), (12), (13) and (14).

- The loss function analyzes the mean squared errors by Equation (15) to optimize the neural network.

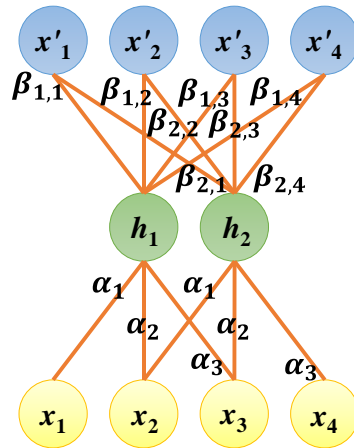


Figure 2. A simple case of auto-encoder based on convolutional neural network

$$h_1 = \alpha_1 \times x_1 + \alpha_2 \times x_2 + \alpha_3 \times x_3 + b_{1,1} \quad (9)$$

$$h_2 = \alpha_1 \times x_2 + \alpha_2 \times x_3 + \alpha_3 \times x_4 + b_{1,2} \quad (10)$$

$$x'_1 = \sum_{i=1}^2 \beta_{i,1} \times h_i + b_{2,1} \quad (11)$$

$$x'_2 = \sum_{i=1}^2 \beta_{i,2} \times h_i + b_{2,2} \quad (12)$$

$$x'_3 = \sum_{i=1}^2 \beta_{i,3} \times h_i + b_{2,3} \quad (13)$$

$$x'_4 = \sum_{i=1}^2 \beta_{i,4} \times h_i + b_{2,4} \quad (14)$$

$$E = \sum_{i=1}^4 \frac{1}{2} (x'_i - x_i)^2 = \sum_{i=1}^4 \frac{1}{2} \sigma_i^2 \quad (15)$$

The gradient descent method is adopted for the optimization of auto-encoder based on convolutional neural network. The weights in the neural network are updated by Equations (16), (17), (18), (19), (20) and (21). When the training process is complete, the reconstructed adjacency matrix will be inputted into the trained neural network to retrieve the values of neurons (i.e., $\{h_1, h_2\}$) in the convolutional layer for spatial feature extraction.

$$\begin{aligned} \frac{\partial E}{\partial \beta_{j,i}} &= \frac{\partial E}{\partial \sigma_i} \frac{\partial \sigma_i}{\partial x'_i} \frac{\partial x'_i}{\partial \beta_{j,i}} \\ &= \sigma_i \times 1 \times h_j \\ &= \sigma_i \times h_j \end{aligned} \quad (16)$$

$$\begin{aligned}
\frac{\partial E}{\partial b_{2,i}} &= \frac{\partial E}{\partial \sigma_i} \frac{\partial \sigma_i}{\partial x'_i} \frac{\partial x'_i}{\partial b_{2,i}} \\
&= \sigma_i \times 1 \times 1 \\
&= \sigma_i
\end{aligned} \tag{17}$$

$$\begin{aligned}
\frac{\partial E}{\partial \alpha_1} &= \sum_{i=1}^4 \frac{\partial E}{\partial \sigma_i} \frac{\partial \sigma_i}{\partial x'_i} \sum_{j=1}^2 \frac{\partial x'_i}{\partial h_j} \frac{\partial h_j}{\partial \alpha_1} \\
&= \sum_{i=1}^4 \sigma_i \times 1 \times \sum_{j=1}^2 \beta_{j,i} \times \frac{\partial h_j}{\partial \alpha_1} \\
&= \sigma_i
\end{aligned} \tag{18}$$

$$\begin{aligned}
\frac{\partial E}{\partial \alpha_2} &= \sum_{i=1}^4 \frac{\partial E}{\partial \sigma_i} \frac{\partial \sigma_i}{\partial x'_i} \sum_{j=1}^2 \frac{\partial x'_i}{\partial h_j} \frac{\partial h_j}{\partial \alpha_2} \\
&= \sum_{i=1}^4 \sigma_i \times 1 \times \sum_{j=1}^2 \beta_{j,i} \times \frac{\partial h_j}{\partial \alpha_2} \\
&= \sum_{i=1}^4 \sigma_i \times (\beta_{1,i} \times x_2 + \beta_{2,i} \times x_3)
\end{aligned} \tag{19}$$

$$\begin{aligned}
\frac{\partial E}{\partial \alpha_3} &= \sum_{i=1}^4 \frac{\partial E}{\partial \sigma_i} \frac{\partial \sigma_i}{\partial x'_i} \sum_{j=1}^2 \frac{\partial x'_i}{\partial h_j} \frac{\partial h_j}{\partial \alpha_3} \\
&= \sum_{i=1}^4 \sigma_i \times 1 \times \sum_{j=1}^2 \beta_{j,i} \times \frac{\partial h_j}{\partial \alpha_3} \\
&= \sum_{i=1}^4 \sigma_i \times (\beta_{1,i} \times x_3 + \beta_{2,i} \times x_4)
\end{aligned} \tag{20}$$

$$\begin{aligned}
\frac{\partial E}{\partial b_{1,j}} &= \sum_{i=1}^4 \frac{\partial E}{\partial \sigma_i} \frac{\partial \sigma_i}{\partial x'_i} \frac{\partial x'_i}{\partial h_j} \frac{\partial h_j}{\partial b_{1,j}} \\
&= \sum_{i=1}^4 \sigma_i \times 1 \times \beta_{j,i} \times 1 \\
&= \sum_{i=1}^4 \sigma_i \times \beta_{j,i}
\end{aligned} \tag{21}$$

151 2.2.2. The Description of a General Case

152 In this case, the reconstructed adjacency matrix can be split n records, and the dimension of each
 153 record is expressed as $1 \times n$. Both of input layer and output layer have n neurons (i.e., n nodes in a
 154 social network), respectively. There are q neurons in the convolutional layer (i.e., the first hidden
 155 layer). The structure of auto-encoder based on convolutional neural network for the general case is
 156 showed in **Figure 3**. The mean squared errors are considered by loss function, and the gradient
 157 descent method is adopted to minimize these errors. In testing and performing stage, the spatial
 158 features can be extracted according to the values of neurons (i.e., \mathbf{H} shown in Equation (22)) in the
 159 hidden layer.

$$\mathbf{H} = \begin{bmatrix} h_1 & h_2 & L & h_q \end{bmatrix} \tag{22}$$

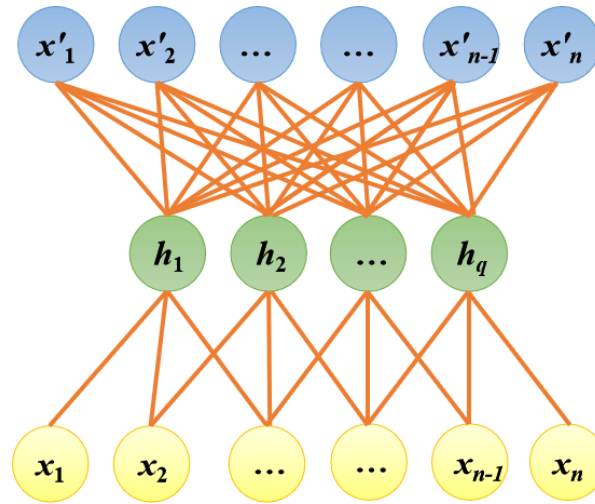


Figure 3. A general case of auto-encoder based on convolutional neural network

2.3. Community Detection Method

The n records from the reconstructed adjacency matrix can be clustered into k groups by the K-means algorithm for detecting communities in a social network. Each record is adopted into the trained neural network, and the spatial features of each record is extracted by the proposed spatial feature extraction method. Therefore, the dimension of each record is expressed as $1 \times q$ (e.g., the vectors of the j -th record shown in Equation (23)) after extracting spatial features and adopted into the K-means algorithm.

$$\mathbf{H}_j = \begin{bmatrix} h_{j,1} & h_{j,2} & \dots & h_{j,q} \end{bmatrix} \quad (23)$$

The proposed community detection method includes three steps as follows.

Step (1). The k records are randomly elected as k cluster centers from the n records. The vectors of the i -th cluster center are expressed as Equation (24).

$$\mathbf{Z}_i = \begin{bmatrix} z_{i,1} & z_{i,2} & \dots & z_{i,q} \end{bmatrix} \quad (24)$$

Step (2). The Euclidean distance $\phi(i, j)$ is applied to measure the distance between the j -th record and the i -th cluster center based on Equation (25). The n records are distributed into the k groups according to the distance between the record and the cluster center, and the center of each cluster is recalculated based on the records in the cluster.

$$\phi(i, j) = \sqrt{\sum_{k=1}^q d(i, j, k)^2}, \text{ where } d(i, j, k) = h_{j,k} - z_{i,k} \quad (25)$$

Step (3). If each cluster center has no changes, the clustering process is finished. Otherwise, Steps (1) and (2) are performed repeatedly.

3. Experimental Results and Discussions

In experiments, four open datasets of practical social networks were adopted to evaluate the performance of the proposed deep community detection method.

3.1. Experimental Environments

This subsection presents datasets, an evaluation factor and experimental designs.

3.1.1. Datasets

Four open datasets of practical social networks which include (1) network19, (2) karate, (3) dolphins, and (4) football were collected for the evaluation of the proposed method. **Table 1** shows the description of datasets and the value of k for K-means algorithm.

Table 1. Datasets of practical social networks

Dataset name	The number of nodes	The number of edges	The value of k	Dataset description
network19	19	37	3	The dataset 'network19' presents a simple network; the network includes 19 nodes which can be clustered into 3 groups.
karate	34	78	3	The dataset 'karate' presents the friendships among 34 members of a karate club at an university.
dolphins	62	159	3	The dataset 'dolphins' presents the social network of 62 dolphins in Doubtful Sound in New Zealand.
football	115	613	12	The dataset 'football' presents the American football games of 115 football teams.

3.1.2. Evaluation Factor

The modularity (shown in Equation (26)) is a popular factor for evaluating the performance of community detection method.

$$Q = \frac{1}{2m} \sum_{ij} (A_{ij} - \frac{k_i k_j}{2m}) \delta(C_i, C_j) \quad (26)$$

Each parameter in Equation (26) is defined and described as follows.

- The parameter A denotes an adjacency matrix.
- The parameter m denotes the number of edges.
- The parameter k_i denotes the degrees of the i -th node.
- If the i -th node and the j -th node are clustered into the same group, the value of $\delta(C_i, C_j)$ is 1.
- The range of Q (i.e., modularity) is between 1 and -1. If the value of Q equals to 1, the performance of community detection method is higher.

3.1.3. Experimental Design

For the evaluation of the proposed method, four cases were designed to compare the modularity of different method combinations as follows.

Case (1): The auto-encoder method was adopted to extract the features of original adjacency matrix in a social network for community detection. The label of Case (1) was expressed as 'AE'.

Case (2): The auto-encoder method was adopted to extract the features of reconstructed adjacency matrix in a social network for community detection. The label of Case (2) was expressed as 'RM+AE'.

Case (3): The auto-encoder method based on convolutional neural network was adopted to extract the features of original adjacency matrix in a social network for community detection. The label of Case (3) was expressed as 'AE+CNN'.

Case (4): The auto-encoder method based on convolutional neural network was adopted to extract the features of reconstructed adjacency matrix in a social network for community detection. The label of Case (4) was expressed as 'RM+AE+CNN'.

In experiments, four cases and four datasets were considered to evaluate the performance of the proposed deep community detection method. The structure of neural network for each dataset is showed in **Table 2**.

Table 2. The structure of neural network

Dataset name	The number of nodes in social network	The structure of neural network
network19	19	19-40-80-160-80-40-19
karate	34	34-40-80-160-80-40-34
dolphins	62	62-40-80-160-80-40-62
football	115	115-40-80-160-80-40-115

3.2. Experimental Results of Matrix Reconstruction Method

Three steps in the proposed matrix reconstruction method include (1) finding the opinion leader in a social network, (2) finding neighbor nodes, and (3) reconstructing the adjacency matrix of the social network. The reconstructed adjacency matrices of practical social networks in open datasets are showed in Figures 4, 5, 6 and 7.

For the analyses of the dataset ‘network19’, **Figure 4(a)** shows the original adjacency matrix, and **Figure 4(b)** shows the reconstructed adjacency matrix. The 7-th node in the original adjacency matrix is elected as the opinion leader. The boundaries of some communities in the original adjacency matrix are overlapped. After performing the proposed matrix reconstruction method, the nodes in the same community are closer, and the boundaries of communities are easier to be detected in the reconstructed adjacency matrix. Therefore, the reconstructed adjacency matrix is suitable for spatial feature extraction and community detection.

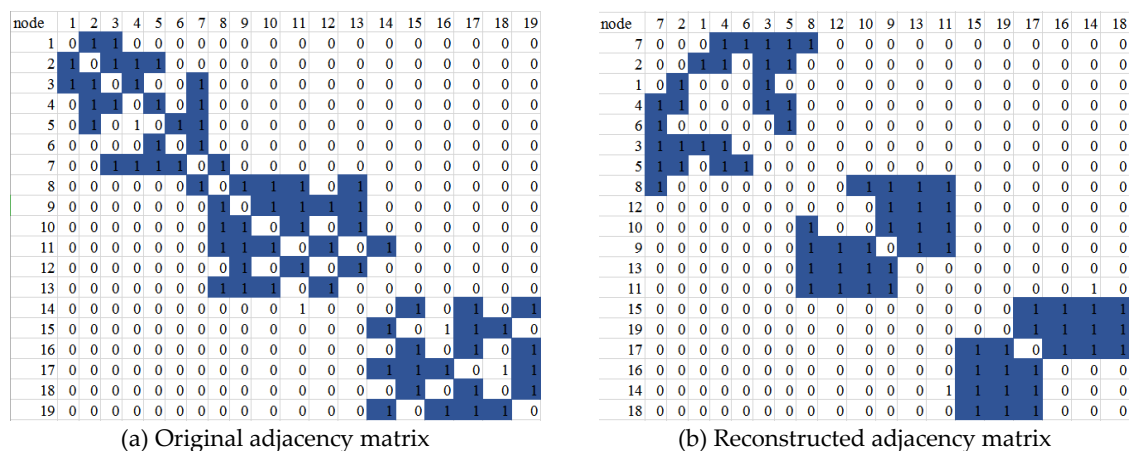


Figure 4. The matrix reconstruction method for ‘network19’

For the analyses of the dataset ‘karate’, **Figure 5(a)** shows the original adjacency matrix, and **Figure 5(b)** shows the reconstructed adjacency matrix. The 34-th node in the original adjacency matrix is elected as the opinion leader. The boundaries of communities in the original adjacency matrix are difficult to be detected.

Although the reconstructed adjacency matrix in this case may have a little of improvement for community detection, the number of isolated points in the reconstructed adjacency matrix is less.

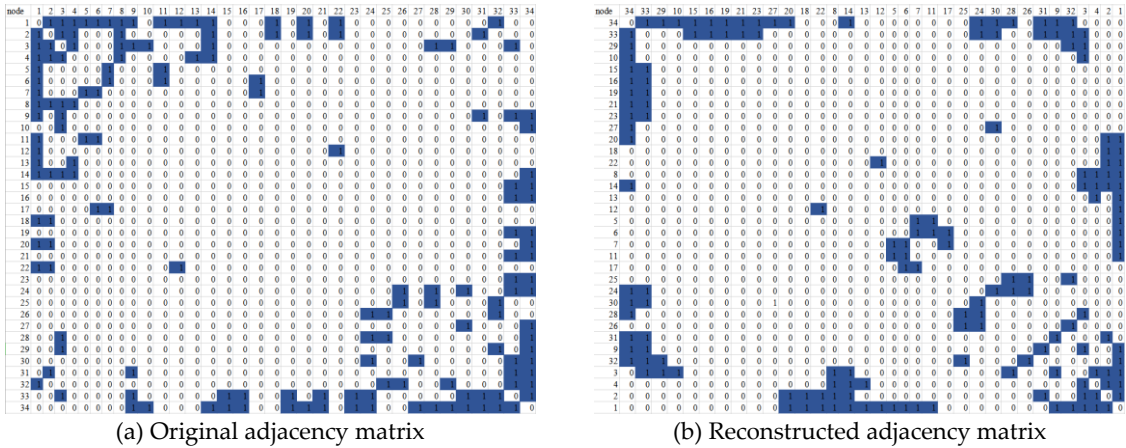


Figure 5. The matrix reconstruction method for ‘karate’

For the analyses of the dataset ‘dolphins’, **Figure 6(a)** shows the original adjacency matrix, and **Figure 6(b)** shows the reconstructed adjacency matrix. The 15-th node in the original adjacency matrix is elected as the opinion leader. The edges in the original adjacency matrix are scattered, and it is difficult to detect social communities. After performing the proposed matrix reconstruction method, the boundaries of communities are clearer for detection, and the number of isolated points is less. The cyberspace structure of communities is clearer for spatial feature extraction.

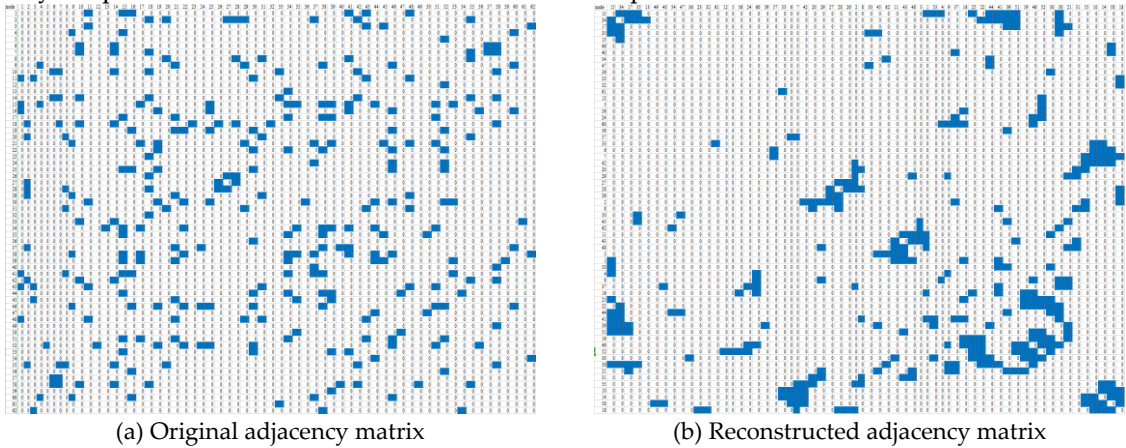


Figure 6. The matrix reconstruction method for ‘dolphins’

For the analyses of the dataset ‘football’, **Figure 7(a)** shows the original adjacency matrix, and **Figure 7(b)** shows the reconstructed adjacency matrix. The 2-nd node in the original adjacency matrix is elected as the opinion leader. The cyberspace structure of communities in the original adjacency is difficult to be detected owing to the scattered distribution. After performing the proposed matrix reconstruction method, 12 communities in the reconstructed adjacency matrix are easier to be detected, so the proposed method can support to extract the spatial features of community in this case.

In accordance with the experimental results in Figures 4, 5, 6 and 7, the nodes in the same community are closer, and the boundaries of communities are clearer. Therefore, the reconstructed adjacency matrices are suitable for spatial feature extraction and community detection.

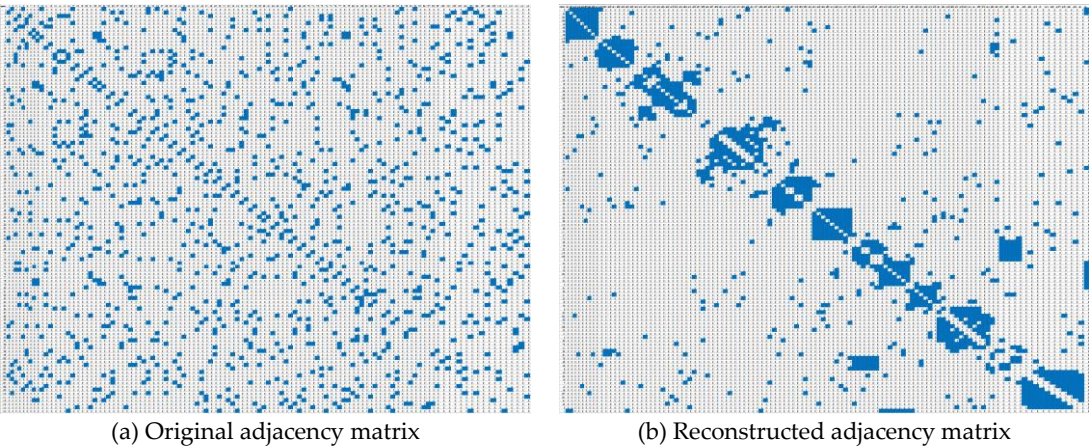


Figure 7. The matrix reconstruction method for ‘football’

3.3. Experimental Results of Spatial Feature Extraction Method and Community Detection Method

This subsection uses the modularity to evaluate the proposed spatial feature extraction and community detection methods, and experimental results of each case in Subsection 3.1.3 are illustrated in Table 3.

Table 3. The modularity of each case

Dataset name	The value of k for K-means	Case (1): AE	Case (2): RM+AE	Case (3): AE+CNN	Case (4): RM+AE+CNN
network19	3	0.667	0.667	0.667	0.667
karate	3	0.051	0.217	0.216	0.370
dophlins	3	0.258	0.307	0.362	0.393
football	12	0.540	0.540	0.586	0.589

For the evaluation of matrix reconstruction method, the modularity in Case (2)(i.e., RM+AE) is higher than the modularity in Case (1)(i.e., AE) for each dataset, and the modularity in Case (4)(RM+AE+CNN) is higher than the modularity in Case (3)(AE+CNN) for each dataset. Therefore, the proposed matrix reconstruction method can support to reconstruct the cyberspace structure of communities for community detection.

For the evaluation of spatial feature extraction method, the modularity in Case (3)(i.e., AE+CNN) is higher than the modularity in Case (1)(i.e., AE) for each dataset, and the modularity in Case (4)(RM+AE+CNN) is higher than the modularity in Case (2)(RM+AE) for each dataset. Therefore, the proposed spatial feature extraction method based on convolutional neural network can extract the spatial features in social networks for the improvement of community detection.

4. Conclusions and Future Work

A deep community detection method for social networks is proposed in this paper. A matrix reconstruction method based on a novel cyberspace structure reconstruction strategy is proposed to acquire spatial proximity matrices and generate reconstructed adjacency matrices. The matrix adds spatial proximity to traditional adjacency matrix and makes convolutional operations easy to compute. Meanwhile, the model of AE based on CNN is constructed to learn spatial eigenvector that automatically extract the spatial features of graph. The spatial eigenvector provides the basis and support for the analysis and application on the network graph. Applying the model of AE based on CNN into the community detection, and experimental results show that the proposed deep

community detection method can effectively detect high quality communities for each practical dataset.

Although the model of AE based on CNN is a meaningful exploration of graph embedding model, the number of neurons in input and output layers is changeless after the deep learning model is constructed. For dynamic community detection, the relationships among nodes may be dynamically changing, so the time-series models are required to extract the spatio-temporal features for arbitrary social networks.

Author Contributions: Conceptualization, Ling Wu and Chi-Hua Chen; Data curation, Ling Wu; Formal analysis, Chi-Hua Chen; Investigation, Ling Wu and Chi-Hua Chen; Methodology, Ling Wu and Chi-Hua Chen; Project administration, Qishan Zhang; Resources, Qishan Zhang; Supervision, Qishan Zhang and Chi-Hua Chen; Validation, Ling Wu and Chi-Hua Chen; Visualization, Chi-Hua Chen; Writing – original draft, Ling Wu and Chi-Hua Chen; Writing – review & editing, Ling Wu, Qishan Zhang and Chi-Hua Chen.

Funding: This work was partially supported by the National Natural Science Foundation of China (Nos. 61906043, 61877010, 11501114 and 11901100), Fujian Natural Science Funds (No. 2019J01243), and Fuzhou University (Nos. 510730/XRC-18075, 510809/GXRC-19037, 510649/XRC-18049 and 510650/XRC-18050).

References

1. D. Yang, X. Liao, H. Shen, X. Cheng, G. Chen, "Modeling the reemergence of information diffusion in social network", *Physica A: Statistical Mechanics and its Applications*, vol. 490, pp. 1493-1500, 2018.
2. D. Yang, X. Liao, H. Shen, X. Cheng, G. Chen, "Dynamic node immunization for restraint of harmful information diffusion in social networks", *Physica A: Statistical Mechanics and its Applications*, vol. 503, pp. 640-649, 2018.
3. D. Yang, X. Liao, J. Wei, X. Cheng, "Modeling Information Diffusion with the External Environment in Social Networks", *Journal of Internet Technology*, vol. 20, no. 2, pp. 369-377, 2019.
4. Y. Yang, X. Zheng, X. Liu, S. Zhong, V. Chang, "Cross-domain dynamic anonymous authenticated group key management with symptom-matching for e-health social system", *Future Generation Computer Systems*, vol. 84, pp. 160-176, 2018.
5. Z. Yu, D. Zhang, Z. Wang, B. Guo, I. Roussaki, K. Doolin, E. Claffey, "Toward context-aware mobile social networks", *IEEE Communications Magazine*, vol. 55, no. 10, pp. 168-175, 2017.
6. J. Zhu, Y. Song, D. Jiang, H. Song, "A new deep-Q-learning-based transmission scheduling mechanism for the cognitive Internet of things", *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2375-2385, 2018.
7. Z. Wang, Z. Peng, Y. Guan, L. Wu, "Manifold regularization graph structure auto-encoder to detect loop closure for visual SLAM", *IEEE Access*, vol. 7, pp. 59524-59538, 2019.
8. J. Zhou, Q. Chen, H. Jiang, S. Cai, G. Shao, H. Kikuchi, "A specific and selective neural response representation with decorrelating auto-encoder", *IEEE Access*, vol. 7, pp. 70011-70020, 2019.
9. S. Mohamed, J. Dong, A.R. Junejo, D.C. Zuo, "Model-based: end-to-end molecular communication system through deep reinforcement learning auto encoder", *IEEE Access*, vol. 7, pp. 70279-70286, 2019.
10. M. Shin, S. Mun, D.K. Han, H. Ko, "New generalized sidelobe canceller with denoising auto-encoder for improved speech enhancement", *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E100A, no. 12, pp. 3038-3040, 2017.
11. Y. Ma, Z. Ye, Y. Li, "Millimeter-wave radar target recognition algorithm based on collaborative auto-encoder", *IEICE Transactions on Information and Systems*, vol. E102D, no. 1, pp. 202-205, 2019.
12. V. Bhatia, R. Rani, "A distributed overlapping community detection model for large graphs using autoencoder", *Future Generation Computer Systems*, vol. 94, pp. 16-26, 2019.
13. M. Gnouma, A. Ladjailia, R. Ejali, M. Zaied, "Stacked sparse autoencoder and history of binary motion image for human activity recognition", *Multimedia Tools and Applications*, vol. 78, no. 2, pp. 2157-2179, 2019.
14. W. Rong, Y. Nie, Y. Ouyang, B. Peng, Z. Xiong, "Auto-encoder based bagging architecture for sentiment analysis", *Journal of Visual Languages & Computing*, vol. 25, no. 6, pp. 840-849, 2014.
15. C. Xu, Y. Dai, R. Lin, S. Wang, "Stacked autoencoder based weak supervision for social image understanding", *IEEE Access*, vol. 7, pp. 21777-21786, 2019.

16. Y. Xu, Y. Chi, Y. Tian, "Deep convolutional neural networks for feature extraction of images generated from complex networks topologies", *Wireless Personal Communications*, vol. 103, no. 1, pp. 327–338, 2018.
17. R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, D. Cohen-Or, "MeshCNN: a network with an edge", *ACM Transactions on Graphics*, vol. 38, no. 4, Article No. 90, 2019.
18. J. Xu, M. Li, J. Jiang, B. Ge, M. Cai, "Early prediction of scientific impact based on multi-bibliographic features and convolutional neural network", *IEEE Access*, vol. 7, pp. 92248–92258, 2019.
19. W. Gao, P. Zhou, "Customized high performance and energy efficient communication networks for AI chips", *IEEE Access*, vol. 7, pp. 69434–69446, 2019.
20. I. Sosnovik, I. Oseledets, "Neural networks for topology optimization", *Russian Journal of Numerical Analysis and Mathematical Modelling*, vol. 34, no. 4, pp. 215–223, 2019.