Physica A 492 (2018) 980–993

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

A clustering algorithm for determining community structure in complex networks

Hong Jin, Wei Yu, ShiJun Li *

State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China
Computer School, Wuhan University, Wuhan 430072, China

HIGHLIGHTS

- By spectral analysis DENCLUE is able to deal with community detection problem.
- It reduces the number of parameters and Sheather–Jones plug-in can select its value.
- It is able to find community structure with arbitrary size and shape.
- It has good scalability in deal with high dimensional data.

ARTICLE INFO

Article history:
Received 19 January 2017
Received in revised form 26 September 2017
Available online 2 December 2017

Keywords:
Community detection
Density based clustering
Spectral analysis
Parameter estimation

ABSTRACT

Clustering algorithms are attractive for the task of community detection in complex networks. DENCLUE is a representative density based clustering algorithm which has a firm mathematical basis and good clustering properties allowing for arbitrarily shaped clusters in high dimensional datasets. However, this method cannot be directly applied to community discovering due to its inability to deal with network data. Moreover, it requires a careful selection of the density parameter and the noise threshold. To solve these issues, a new community detection method is proposed in this paper. First, we use a spectral analysis technique to map the network data into a low dimensional Euclidean Space which can preserve node structural characteristics. Then, DENCLUE is applied to detect the communities in the network. A mathematical method named Sheather–Jones plug-in is chosen to select the density parameter which can describe the intrinsic clustering structure accurately. Moreover, every node on the network is meaningful so there were no noise nodes as a result the noise threshold can be ignored. We test our algorithm on both benchmark and real-life networks, and the results demonstrate the effectiveness of our algorithm over other popularity density based clustering algorithms adopted to community detection.

1. Introduction

Networks are extensively used to represent complex systems in social science, engineering, biology and so on. A common feature of these networks is community structure [1], forming group structures in networks based on work, friendship, family, and other types of relations [2]. In the World Wide Web communities may correspond to groups of pages dealing with the same or related topics. While in metabolic networks they may be related to functional modules such as cycles and

* Corresponding author at: Computer School, Wuhan University, Wuhan 430072, China.
E-mail address: shjli@whu.edu.cn (S. Li).

https://doi.org/10.1016/j.physa.2017.11.029
0378-4371/

pathways. Communities are also called clusters or modules with high concentrations of edges within special groups and low concentrations between these groups [3]. The nodes in one community probably share common properties or play similar roles within the group. In view of this, community detection can lead to important advances in complex systems analysis.

For this reason, a large variety of community detection algorithms have been proposed such as modularity-based algorithms, random walk-based algorithms, clustering based algorithms and matrix decomposition-based algorithms [4–9]. A more detailed analysis can be found in [10]. To some extent the process of community detection is similar to clustering analysis.

Because communities in networks often have an arbitrary size and shape, finding communities in complex networks is a challenging task [11]. Under these circumstances, new clustering methods have recently been introduced to solve this highly complex problem. Density-based clustering methods look for clusters of arbitrary size and shape and have hence been widely used in community detection. Recently, Huang et al. proposed a novel density-based network clustering method called graph-skeleton-based clustering (gSkeletonClu) [12]. This algorithm can find communities as well as hubs and outliers. However, the main difficulty for the gSkeletonClu algorithm is the relatively complicated parameter selection process. Though it provides a convenient way to automatically set the parameter value, excessive evaluation of some indexes is required to finally determine the appropriate parameter setting.

In order to address the sensitivity issue, a generalized density-based method was proposed, denoted as GDENCLUE for convenience, with the particular goal of community detection in the complex networks. Because all nodes are meaningful comparing with the original density based clustering algorithm, the parameter noise threshold can be eliminated, so that only the parameter the influence factor is needed for GDENCLUE. As a kernel based clustering approach, it introduces the influence function to characterize the density distribution of the dataset. The approximation of the density distribution function is obtained by summing up the influence functions. The process to study local minima of the density distribution function can be viewed as a tool to find the clusters from datasets. Moreover, the selection of the kernel scale parameter determines the number of clusters. In this algorithm named influence factor is determined by a Sheather–Jones plug-in.

This paper focuses on one approach of density based clustering in computer science. Combined with the spectral analysis, the complex network data is first transformed before applying a further clustering procedure [13]. It is able to produce excellent results as various approximate optimization techniques. However, it does not require an impractical large computational effort like other algorithms when optimizing the modularity exhaustively. When converted to Euclidean Space there was no noise present in the data, alleviating the dependency on parameters for density based clustering.

The rest of the paper is organized as follows. Section 2 provides a general introduction to spectral clustering analysis. While in Section 3 the nonparametric estimation approach sheather-Jones plug-in method is described. Then the clustering algorithm proposed for determining community structure is detailed in Section 4. Section 5 provides several experimental evaluations and discussions about our approach. Finally, the conclusions are drawn in Section 6 along with some issues for further work.

2. Common spectral clustering analysis

Given a set of data points $x_1, \ldots, x_n$ and some measurement of similarity $s_{ij} \geq 0$ between all pairs of data points x_i and x_j, common spectral clustering represents the data in form of the similarity graph $G = (V, E)$ [14]. In this graph each vertex v_i represents a data point x_i. If the similarity s_{ij} between the corresponding data points x_i and x_j is positive or larger than a certain threshold, then there exists an edge between these two vertices [15]. The most commonly used similarity measurement is Gaussian function as formula (1) shows:

$$s_{ij} = \exp\left(\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right) \tag{1}$$

In the following graph G is assumed to be weighted, each edge between two vertices v_i and v_j carries a non-negative weight $w_{ij} \geq 0$. Let $W = \left(w_{ij}\right)_{i,j=1,\ldots,n}$ be the weighted adjacency matrix of the graph. Then the problem of clustering can now be formulated using the weighted similarity graph. That is to find a partition of the graph such that the edges between different groups have very low weights and the edges within a group have high weights.

For a given complex network $G = (V, E)$ itself can be seen as a kind of similarity graph, its adjacency matrix A can be regarded as the weight matrix W of the graph. In which element A_{ij} is equal to 1 if node i has directly connected to node j and 0 otherwise. It can be represented as formula (2) shows:

$$A_{ij} = \begin{cases} 1 & \text{if } i, j \text{ are connected} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The main tools for spectral clustering are graph Laplacian matrices that derived from W [16]. There is no unique convention how the different matrices denoted, we choose the normalized symmetric Laplacian as formula (3) shows:

$$L_{sym} = I - D^{-1/2} A D^{-1/2} \tag{3}$$

In which D is the diagonal matrix with elements as formula (4) shows and N is the number of the points.

$$D_{ii} = \sum_{j=1}^{N} A_{ij} \tag{4}$$

In order to implement spectral clustering the first k eigenvectors corresponding to the k smallest eigenvalues of the graph Laplacian matrix should be computed [17]. By stacking the obtained eigenvectors as columns, each row of the matrix together can be viewed as the initial data which to be clustered [18].

On the basis of the normalized symmetric Laplacian L_{sym}, there is another matrix also called normalized graph Laplacian defined as $D^{-1/2}AD^{-1/2}$. Taking into consideration the computation complexity, we choose this normalized matrix to discuss in the following parts.

Since the eigenvectors of matrices L_{sym} and $D^{-1/2}AD^{-1/2}$ are in total the same, the difference is that the eigenvectors of the latter matrix corresponding to the largest eigenvalues [19]. Consequently, the spectral clustering algorithm discussed here chooses to compute the largest k eigenvalues with the corresponding eigenvectors of the normalized symmetric Laplacian matrix $D^{-1/2}AD^{-1/2}$.

3. Density based clustering algorithm

Based on a well-developed area of statistic and pattern recognition known as kernel density estimation, density-based clustering approach models the overall density of a set of points as the sum of influence functions or kernel functions associated with each point [20]. The resulting overall density function will have local peaks. For each data point, a hill climbing procedure finds the nearest peak [21]. Then the set of data points associated with a particular peak becomes a cluster.

First, we explained the general notion of influence and density function. Given a d-dimensional data space denoted as F^d, the density function at a point $x \in F^d$ is defined as the sum of the influence functions of all data objects at that point [22]. The basic idea of the algorithm is followed [20].

(1) Influence and Density Function

Given a data object$y \in F^d$, its influence function is a function $f_I^y : F^d \to R^+$as formula (5) in which f_I represents a basic influence function [23].

$$f_I^y(x) = f_I(x, y) \tag{5}$$

The density function is defined as the sum of the influence functions of all the data objects at that point. Assumed that we have n data objects $D = \{x_1, x_2, \ldots, x_n\} \in F^d$, the density function is defined as formula (6)

$$f_I^D(x) = \sum_{i=1}^{n} f_I^{x_i}(x) \tag{6}$$

Theoretically, the arbitrary function can be chosen as an influence function [23]. On the specific type of the influence function, it requires a distance function $d : F^d \times F^d \to R^+$ to measure the distance between any two d-dimensional data objects. Since the definitions do not depend on the choice of the distance function, for simplicity in the following we assume a Euclidean distance function. When dealing with the complex network data, we choose Gauss influence function as formula (7) shows.

$$f_{Gauss}(x, y) = e^{-\frac{d(x,y)^2}{2\sigma^2}} \tag{7}$$

The density function which results from a Gauss Influence function is as (8):

$$f_{Gauss}(x) = \sum_{i=1}^{n} e^{-\frac{d(x,x_i)^2}{2\sigma^2}} \tag{8}$$

(2) Definitions of Density-Attractor and Density-Attracted

These definitions require first to introduce the notion of gradient. The gradient of a function $f_I^D(x)$ is defined as formula (9)

$$\nabla f_I^D(x) = \sum_{i=1}^{n} (x_i - x) \cdot f_I^{x_i}(x) \tag{9}$$

In the context of Gauss influence function, the gradient is defined as formula (10):

$$\nabla f_{Gauss}^D(x) = \sum_{i=1}^{n} (x_i - x) \cdot e^{-\frac{d(x,x_i)^2}{2\sigma^2}} \tag{10}$$

Now, we are able to define the notions of density-attractor and density-attracted.

Given a point $x^* \in F^d$, if it is a local maximal of the overall density function f_I^D, then we call it a density-attractor of the known influence function [24].

A point $x \in F^d$ is density-attracted to a density-attractor x^*, if there exists a point set $\{x_0, x_1, \ldots, x_k\}$ satisfying the condition that

$x_0 = x$,

$$x_i = x_{i-1} + \delta \cdot \frac{\nabla f_I^D(x_{i-1})}{\left\| \nabla f_I^D(x_{i-1}) \right\|} \quad \text{and} \quad d\left(x_k, x^*\right) < \varepsilon$$

In which δ represents the step length of hill climbing strategy. ε represents a certain distance threshold between the two points.

On the basis of the above definitions, we are now able to illustrate the cluster formation process under the idea of density-based clustering.

The local peaks known as the density attractors of the resulting overall density function can be used to define clusters in a natural way [25]. In addition, for each data point a hill climbing process finds the nearest density attractor associated with that point [19]. Consequently, the set of data objects associated with a particular peak becomes a natural cluster.

4. Detecting the community structure in the complex networks using density based clustering

Through a common spectral clustering analysis, we can transform complex network data into Euclidean Space Data. Then the community detection in complex network can be viewed as a general clustering problem. In this transition process, when we use density based clustering algorithm to deal with community discovering there exists no noise data. Consequently, compared to the originate density based clustering algorithm the parameter noise threshold can be neglected.

4.1. Parameter selection of the density based clustering for community detection

In view of the above analysis, here we only need to consider one of the parameters for the original density based clustering algorithm. The parameter σ determines the influence of a point in its neighborhood. It amounts to determine the bandwidth h in the density estimates [23].

In order to understand how to assess the quality of a density estimator, we must first give some related background knowledge. Let $\hat{f}$ express an estimator of f under a fixed value h. When h is small it indicates that data points observed near x contribute more on $\hat{f}(x)$, relatively a larger h indicates that distant data contribute nearly equally to observations near x [26]. The integrated squared error (ISE) as formula (11) shows can be used to evaluate the performance of $\hat{f}$ as an estimator of f [27].

$$ISE(h) = \int_{-\infty}^{\infty} \left(\hat{f}(x) - f(x)\right)^2 d_x \tag{11}$$

It can be seen that $ISE(h)$ is a function of the observed data. The performance of $\hat{f}$ is dependent on the observed sample [28]. In order to make the generic properties of an estimator do not rely on particular observed sample, the mean integrated squared error ($MISE$) which further average $ISE(h)$ over all samples seems more sensible. It can be denoted as $MISE(h) = E\{ISE(h)\}$.

Consequently, $MISE(h)$ can be regarded as a global measure of error with respect to the sampling density [29]. Meanwhile, through an interchange of expectation and integration it can be expressed as formula (12) shows

$$MISE(h) = \int MSE_h\left(\hat{f}(x)\right) dx \tag{12}$$

in which

$$MSE_h\left(\hat{f}(x)\right) = E\left\{\left(\hat{f}(x) - f(x)\right)^2\right\} = \text{var}\left\{\hat{f}(x)\right\} + \left(bias\left\{\hat{f}(x)\right\}\right)^2$$

and $bias\left\{\hat{f}(x)\right\} = E\left\{\hat{f}(x)\right\} - f(x)$. From Eq. (12) we can explain $MISE(h)$ as the accumulation of the local mean squared error at every x.

Both $MISE(h)$ and $ISE(h)$ can be used to measure the quality of the estimator $\hat{f}$. To better understand bandwidth selection a further analysis of $MISE(h)$ should be executed. Assuming that K represents a symmetric, continuous probability density function with mean zero and variance $0 < \sigma_K^2 < \infty$. In addition, for a given function g a measure of its roughness denoted as $R(g)$ which is expressed by formula (13)

$$R(g) = \int g^2(z)\, dz \tag{13}$$

with the assumption that $R(K) < \infty$ and f is sufficiently smooth [30], f will have two bounded continuous derivatives that $R\left(f''\right) < \infty$.

Silverman proposed an elementary approach: by means of replacing f with a normal density and its variance set to match the sample variance [31]. Then it estimates $R\left(f''\right)$ by $R\left(\varphi''\right)/\hat{\sigma}^5$, in which φ is the standard normal density function. Therefore, Silverman's rule of thumb can be represented by formula (14) [31]

$$h = \left(\frac{4}{3n}\right)^{1/5} \hat{\sigma} \tag{14}$$

when f is multimodal the ratio of $R\left(f''\right)$ to $\hat{\sigma}$ may be larger than that for normally distributed data, it could lead to over-smoothing. By considering the interquartile range (IQR) it can overcome this difficulty. Which is a more robust measure of spread than is $\hat{\sigma}$ [32], then Silverman suggests to replace $\hat{\sigma}$ in (14) by formula (15)

$$\widetilde{\sigma} = \min\left\{\hat{\sigma}, IQR/\left(\Phi^{-1}(0.75) - \Phi^{-1}(0.25)\right)\right\} \approx \min\left\{\hat{\sigma}, IQR/1.35\right\} \tag{15}$$

In which Φ is the standard normal cumulative distribution function. Due to its strong tendency to over smooth, it is not recommended to use in general case. However, Silverman's rule of thumb is still valuable to produce approximate bandwidth for pilot estimation used in sophisticated plug-in methods [33].

Alternatively, choose the empirical estimation of $R\left(f''\right)$ instead of Silverman's rule of thumb as a better choice. Then the kernel-based estimator is as formula (16) shows:

$$\left(\hat{f}\right)''(x) = \frac{d^2}{d_{x^2}}\left\{\frac{1}{n\sigma_0}\sum_{i=1}^{n} L\left(\frac{x - X_i}{\sigma_0}\right)\right\} = \frac{1}{n\sigma_0^3}\sum_{i=1}^{n} L''\left(\frac{x - X_i}{\sigma_0}\right) \tag{16}$$

In which σ_0 represents the bandwidth and L represents a sufficiently differentiable kernel used to estimate f''.

As previously mentioned, here we introduce a representative plug-in method to find the bandwidth. By a pilot bandwidth, plug-in methods estimate one or more important features of the density function f. In comparison to other plug-in methods, Sheather-Jones as a typical plug-in method is easier to operate and perform better [26]. It is a two-stage process, explained as follows [27]:

Firstly, use a simple rule of thumb to calculate the bandwidth σ_0. Then by means of this bandwidth to estimate $R\left(f''\right)$, it is the only unknown element in the expression of the optimal bandwidth.

At the second stage, compute the bandwidth σ and produce the final kernel density estimate.

4.2. Algorithm

When it is individualized in community detection, for a known complex network it can be described as follows.

(1) Formalize the input data to density based clustering.

(a) Define A to be the adjacency matrix such that $A_{ij} = 1$ if node i and j are connected by an edge and $A_{ij} = 0$ otherwise.

(b) Construct the matrix D, it is a diagonal matrix whose (i, i)-element is the sum of A's ith row.

(c) Compute a normalized matrix of A, defining this Laplacian:

$$L = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$$

(d) Find the k eigenvectors $\{e_1, e_2, \ldots, e_k\}$ of the normal Laplacian matrix L associated to the largest k nontrivial eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_k\}$.

(e) Form the matrix $E = [e_1, e_2, \ldots, e_k]$ by stacking the eigenvectors in columns.

(f) Compute the matrix H from E by normalizing each of E's rows to norm 1.

$$H_{ij} = E_{ij} \Big/ \left(\sum_j E_{ij}^2\right)^{1/2}$$

(g) Treating each row of H as a point y_i, $i = 1, 2, \ldots, n$ in R^k.

(2) Cluster points y_i in R^k into clusters via density based clustering

(a) Derive a density function for the space occupied by the formed input dataset.

(b) Identify the points that are local maxima.

(c) Associate each point with a density attractor by hill climbing process.

(d) Define clusters consisting of points associated with a particular density attractor.

By this way, we can uncover the community structure with high quality. Meanwhile, it should be noted that in this method the number of the clusters is dependent on the parameter of density based clustering. It is different from the group of community detection algorithms represented by modularity. For which, the number of communities depends on the modularity optimization.

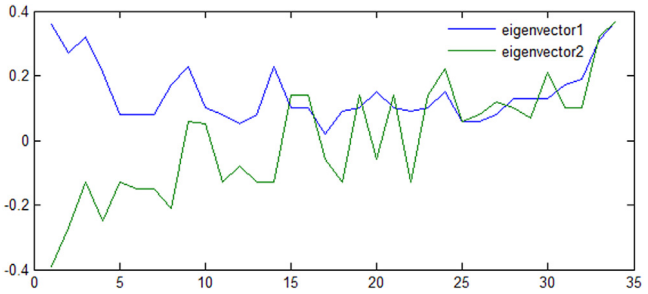

Fig. 1. The eigenvectors of the normalized Laplacian matrix corresponding to the Zachary club.

4.3. Time complexity analysis

Suppose m and n representing the number of edges and the number of nodes respectively of the investigated complex network. Obviously, the time complexity of the proposed algorithm consists of two parts. The spectral analysis part has time complexity $o\left(mkt + nk^2t + k^3t\right)$, where k is the number of computed eigenvectors needed in the experimental process, t is the number of iterations required until convergence. The most expensive step of spectral analysis is the computation of the eigenvalues/eigenvectors of Laplacian matrix. This process uses the Implicitly Restarted Lanczos Method. Specifically, for sparse network where $k \prec n$, the time complexity can be simplified as $o\left(mkt + nk^2t\right)$. Moreover, the Implicitly Restarted Lanczos Method to some extent takes near linear time with respect to the number of nodes n. While the density based clustering algorithm part has time complexity $o\left(n^2\right)$. The complexity of our algorithm is hence $o\left(mkt + nk^2t + n^2\right)$.

5. Experiments and results

In order to validate the efficiency of our algorithm, experiments on both real world and benchmark networks were carried out. There were three representative real world networks such as Karate club network, Dolphin network and American College Football network. In addition, a benchmark network with predetermined community structures was given. Here we choose gSkeletonClu as the comparison algorithm. gSkeletonClu is one of the competitive density-based clustering methods used in community detection problem. The results show that the communities discovered by our algorithm are more close to the actual situation. Additionally, the parameter involved can be computed relatively accurate by the Sheather-Jones Plug-in method.

5.1. Real world and benchmark networks testing

5.1.1. Karate Club network

In this subsection, we took the well-known karate club network as an example. During the course of the Karate club study, a disagreement happened between the administrator and the club's instructor [34]. Which led to the instructor's leaving and constructing a new club, taking about a half members of the original club with him. Here we applied our algorithm to this network as an attempt to identify the factions involved in the split of club.

First, by spectral analysis we transformed the karate club network data into Euclidean space data. Fig. 1 showed the two eigenvectors associated with the first two largest eigenvalues of the corresponding normalized Laplacian matrix of Karate club network.

The plug-in selector for the two-dimensional Euclidean space data obtained from the eigenvector 1 and eigenvector 2 displayed on the left panel of Fig. 2. On the right was the acquired original Euclidean space data of the karate club network data. Moreover, on the left in Fig. 2 it displayed a contour plot with the upper 25%, 50% and 75% contours of the sample highest density regions. The optimal bandwidth produced in Fig. 2 was $\sigma = 0.0688$.

Based on the formalized input data, we can obtain the community detection result of Karate club network by our algorithm. As Fig. 3 shown, it was divided into 4 communities represented by different colors. Compared with the actual situation, node 1 had been separated from one of the standard communities which includes nodes 1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13, 14, 17, 18, 20, 22. Similarly, nodes 33 together with 34 were identified as individual community separated out of the other standard community which includes all the rest nodes. Through above analysis, for our algorithm it has given the main community structure.

For comparison, in Fig. 4 it showed the community detection result of Karate club network by gSkeletonClu algorithm. Especially, in gSkeletonClu algorithm it defined two types of nodes such as hubs and outliers. In which nodes 10 and 20 represented hubs while nodes 12, 15, 16, 19, 21 and 23 represented outliers. Compared to the standard partition, exclusive the two special types of nodes the remaining nodes were divided into 4 communities. Each two of them composed one of the standard communities, it can be seen from the figure. For example, nodes 25, 26, 32 and 29 constructed one community

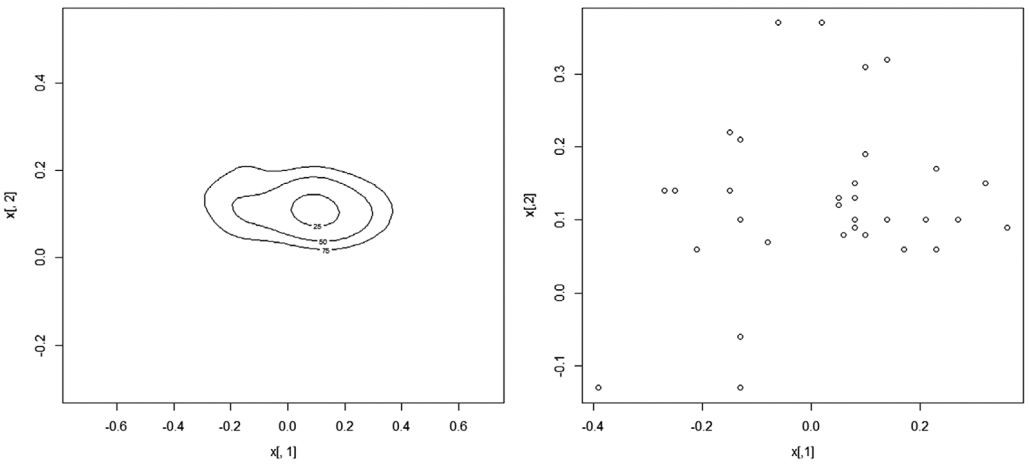

Fig. 2. The optimal bandwidth obtained by Sheather-Jones method for Zachary club network.

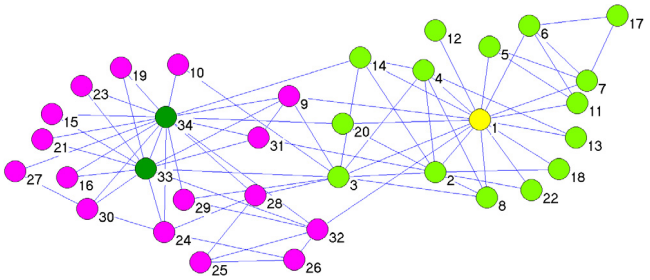

Fig. 3. Karate club network divided by our method.

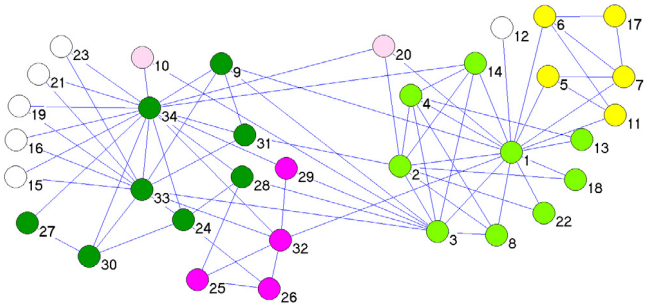

Fig. 4. Karate club network divided by gSkeletonClu.

while nodes 9, 24, 27, 28, 30, 31, 33 and 34 constructed another community. Merging these two communities, then it formed the main part of one of the two standard communities. Consequently, the community structures discovered by gSkeletonClu algorithm were much more different from the actual situation.

5.1.2. Dolphin network

The dolphin network was constructed by the observations of 62 bottlenose dolphins over a period of 7 years [35]. In which nodes represent dolphins and edges represent that the nodes with edges have more interactions than those without edges. Previous study showed that in this network it contains 2 communities and one of them further breaks down into 4 sub-communities [36].

First, by means of spectral analysis we converted the dolphin network data into Euclidean space data. Fig. 5 showed this change of representation, in which the two eigenvectors corresponding to the two largest eigenvalues of the dolphin network's normalized graph Laplacian matrix were displayed. In the left panel of Fig. 6 it gave the optimal bandwidth value

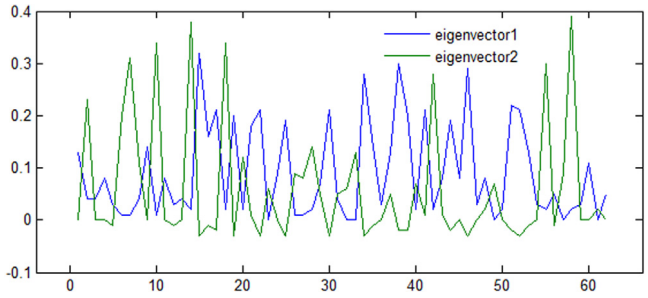

Fig. 5. The eigenvectors corresponding to the two largest eigenvalues of the Dolphin network's normalized graph Laplacian matrix.

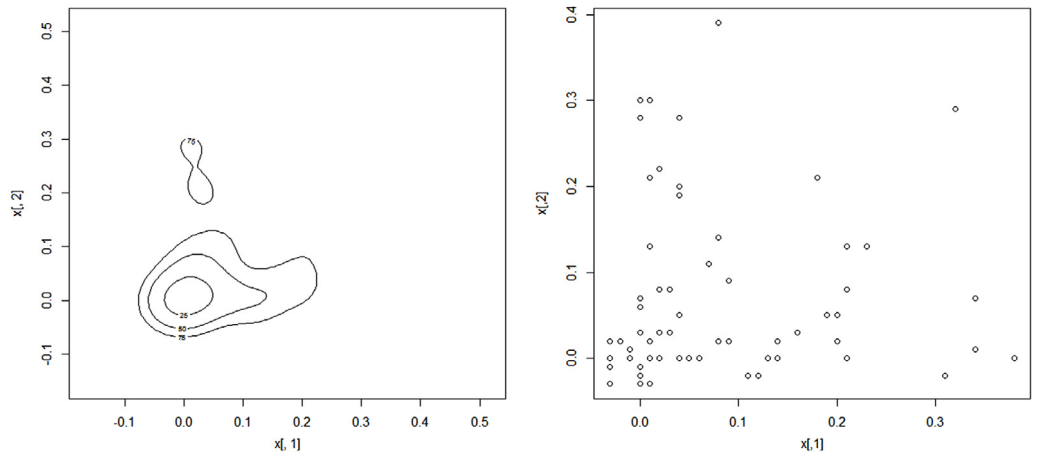

Fig. 6. The derived optimal bandwidth by Sheather-Jones plug in selector for the two dimensional Euclidean data points related to the Dolphin network.

of the plug-in selector for the above formalized data constructed by eigenvector 1 and eigenvector 2. Similarly, in the right panel of Fig. 6 was the obtained two dimensional Euclidean space data of dolphin network. Moreover, from the left panel of Fig. 2 it can be seen that there were two parts with upper 75% contours of the sample highest density regions. The optimal bandwidth produced in Fig. 5 was $\sigma = 0.0355$.

Then by the proposed density based clustering algorithm the community structures of dolphin network can be discovered as Fig. 7 shown. It can be seen that the dolphin network was divided into 6 communities. It was a little different from the actual situation as previous study indicated. The community represented by green was inconsistent with the previous study argued. Apart from the community painted in green the rest nodes agglomerated together were actually broken down into 4-sub communities. While Fig. 7 showed this part was divided into 5-sub communities. The distinct difference between the actual partition and our algorithm was the community painted in white. It formed a community of its own rather than merged into the community shown in pink.

For the comparative algorithm gSkeletonClu the community detection result of dolphin network was shown in Fig. 8. It may be caused by some latent structures in the dolphin network. The comparative algorithm exhibited relative low performance Many nodes were identified as hubs and outliers. To some extent, it was not in accordance with actual situation. In Fig. 8 the nodes identified as hubs were classified as one community including *Feather*, *MN23*, *DN63*, *Kringel*, *Double*, *TR88*, *SN100* and *Fish*. Similarly, the other type of nodes outliers were viewed as another community constructed by *Zig*, *Quasi*, *Ripplefluke*, *DN16*, *Knit*, *Wave*, *TR82*, *SN89*, *Haecksel*, *Vau*, *Five*, *Cross*, *Zap*, *Thumper*, *PL*, *TR77*, *Zipfel*, *SN96*, *CCL*, *Stripes*, *TSN83*, *TR120*, *SMN5*, *SN63*, *Fork*, *Beak*, *TSN103*, *Whitetip and Bumper*. Then the rest nodes formed 3 different communities. Two of them together was inconsistent with the main part of the actual community structures as the previous study argued. Additionally, most of the nodes in community painted in yellow were the same as the community painted in pink of our algorithm.

5.1.3. American college football network

The American College Football Network was the last example of real world networks. It described the schedule of Division games for the 2000 season [12]. In which nodes represent teams and edges represent regular season games between the two teams they connect. Previous study indicated that it can be divided into 11 communities each containing around 8–12

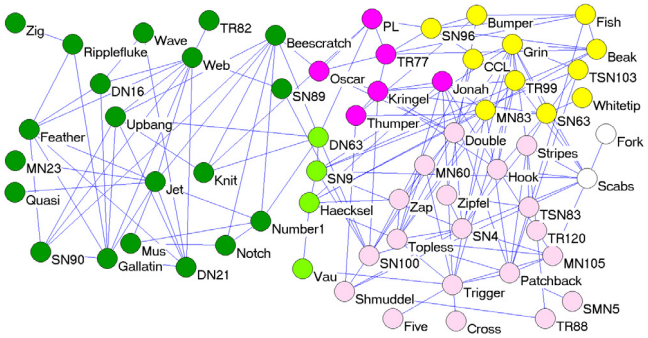

Fig. 7. Dolphin network divided by the proposed algorithm. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

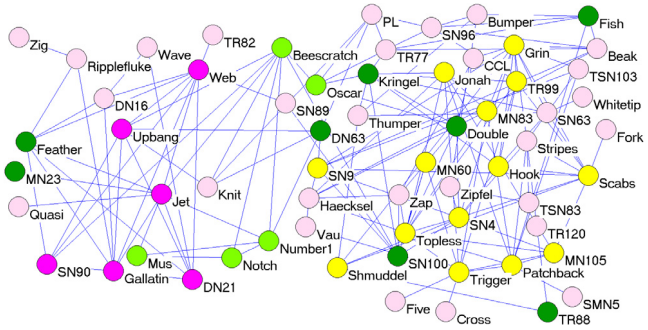

Fig. 8. Dolphin network divided by the comparative algorithm gSkeletonClu. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

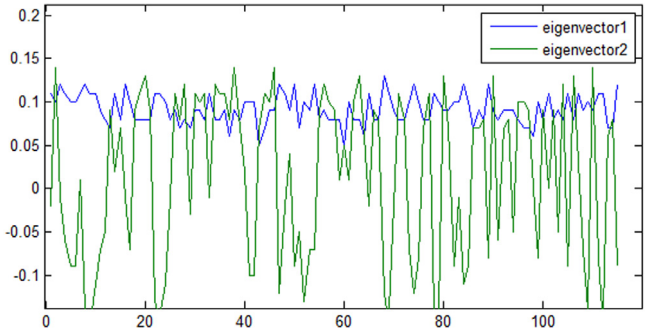

Fig. 9. The eigenvectors corresponding to the two largest eigenvalues of the American College Football network's normalized graph Laplacian matrix.

teams [11]. Teams in the same community play more frequently than those in different communities. Note that there exist four independent teams which do not belong to any community.

The same as before, we should first transform the network data of American college football network into Euclidean space data. Fig. 9 showed the two eigenvectors corresponding to the two largest eigenvalues of the American college football network's normalized graph Laplacian matrix. For the obtained two-dimensional Euclidean space data represented by eigenvector 1 and eigenvector 2, the optimal bandwidth selected by the plug-in selector was shown in the left panel of Fig. 10. While in the right panel of Fig. 10 showed the original two-dimensional data points related to the American college football network. Moreover, in the left panel of Fig. 10 there were more than two parts with upper 50% and 75% contours of the sample highest density regions. The derived bandwidth value was $\sigma = 0.0194$.

With the optimal bandwidth setting, the American college football network was divided into 11 communities. As shown in Fig. 11, the communities discovered by our algorithm were basically in accordance with the actual background. To look deep into the structure there were several points should be noted. According to the background, nodes *FloridaState*, *Maryland* which in the community represented by red were actually belong to another community colored by rosepink. While nodes

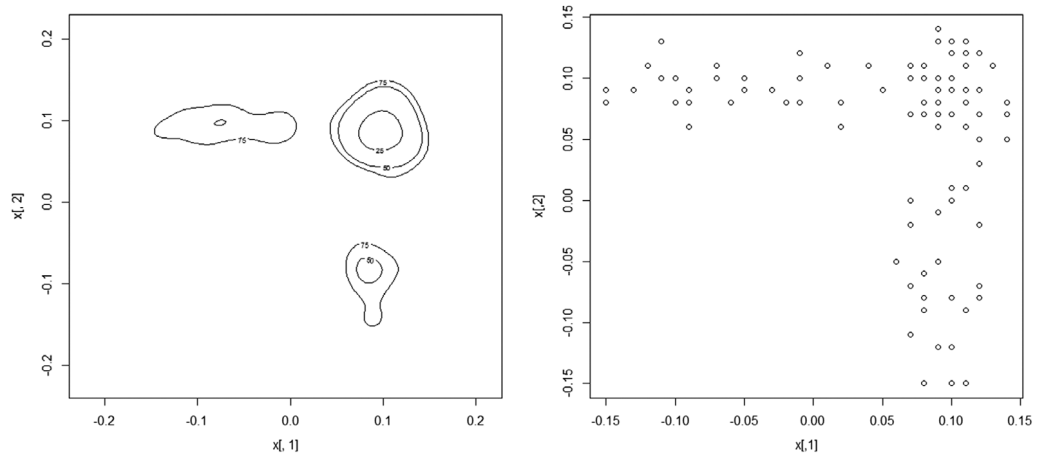

Fig. 10. The derived optimal bandwidth by Sheather-Jones plug in selector for the two dimensional Euclidean data points related to the American College Football network.

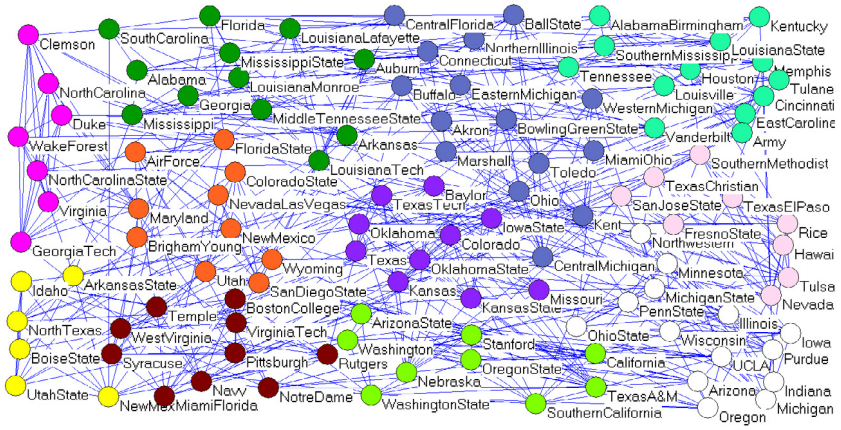

Fig. 11. American college football network divided by proposed algorithm. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Nebraska, TexasA&M in green should be divided into the community represented by blue. Also, the nodes *UCLA, Oregon* in white actually were part of the community represented by green. One last point remained to be noted was that nodes *Kentucky, Vanderbil, Tennessee, LouisianaState* colored by lightgreen should be separated into the community represented by dark green. Although the community detection result of our algorithm was a little different from the actual situation, it remained the relatively complete community structure of the actual situation.

For comparative, Fig. 12 showed the community detection result discovered by gSkeletonClu, which also retrieved 11 communities. In which the nodes identified as hubs and outliers were respectively regarded as independent communities. Nodes *Wyoming, CentralFlorida, Connecticut, LouisianaTech, LouisianaMonroe, MiddleTen-nesseeState, Navy,* and *NotreDame* viewed as hubs were colored by white. Specifically, for this network the comparative algorithm found no outliers. In accordance with the background situation, four of the nodes in hubs were independent teams. While for gSkeletonClu the partition of the rest nodes matched perfectly with the original background knowledge. There existed only a small quantity of misclassified nodes. It can be seen that for this dataset the comparison algorithm performs well.

5.1.4. Benchmark network

At last, we use LFR-benchmark generator to produce a network with power law degree distribution and implanted communities within the network as an example. LFR have been proposed by Lancichinetti et al. [37], in which the distributions of node degree and community size are both power laws with tunable exponents. This model accounts for important features of real networks, like the fat-tailed distributions of node degree and community size.

Based on the LFR-benchmark model, the following parameters such as the number of nodes N, the average degree $\langle k \rangle$, the mixing parameter μ (which means that each node shares a fraction $1 - \mu$ of its links with the nodes of its own community and a fraction μ with the nodes of other communities), the maximum degree $k_{\max}$, exponent for the degree distribution γ,

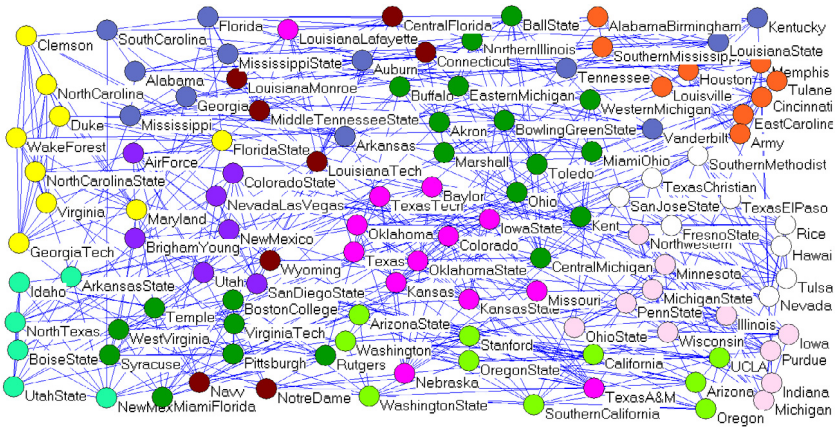

Fig. 12. American college football network divided by the comparative algorithm gSkeletonClu. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

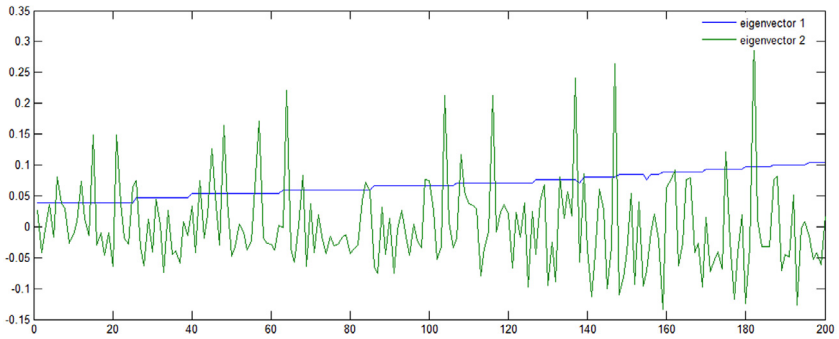

Fig. 13. The eigenvectors corresponding to the two largest eigenvalues of the benchmark network's normalized graph Laplacian matrix.

the exponent for the community size distribution β, the maximum for the community sizes $s_{\max}$, and the minimum for the community sizes $s_{\min}$.

We tested the proposed algorithm on benchmark network with parameter settings as $N = 200$, $\langle k \rangle = 7$, $\mu = 0.3$, $k_{\max} = 20$, $\gamma = 0.6$, $\beta = 0.4$, $s_{\max} = 20$, $s_{\min} = 10$. Then the partitioning results of the proposed algorithm and the comparative algorithm both compared to the predefined communities of the graph.

First, by spectral analysis we transformed the benchmark network data into the Euclidean space data. Fig. 13 showed the eigenvectors corresponding to the two largest eigenvalues of the benchmark network's normalized graph Laplacian matrix. In the left panel of Fig. 14 it displayed the result of the plug-in selector for the two dimensional Euclidean space data represented by eigenvector 1 and eigenvector 2. While in the right panel of Fig. 14 showed the corresponding two dimensional data points of the benchmark network. From the left panel of Fig. 14 it described a contour plot with four parts upper 75% contours of the sample highest density regions. The derived optimal bandwidth was $\sigma = 0.0113$.

To make the comparison more clearly, Fig. 15 showed the predefined community structures of the benchmark network. Based on the formalized input data for the proposed density based clustering algorithm and the derived optimal bandwidth value, the community structures discovered by our algorithm was shown in Fig. 16. The produced result was essentially consistent with the predefined community structures. There existed only a little difference. For example, node 1 had not been rightly included in the community painted in black of the predefined community structure. The situations for node 2 and 25 were the same as node 1. The community represented by gray of the defined community structure was covered by the community colored yellow of our algorithm. While nodes 12, 25 were also contained in the community painted in yellow. Similarly, the community represented by dark green in both of our algorithm and the predefined community structure was basically consistent while for our algorithm it also contained nodes 17, 19. Besides, nodes 10, 11, 23, 106 were not correctly identified.

For comparison, Fig. 17 showed the community detection result of the benchmark network by gSkeletonClu. Especially, in gSkeletonClu algorithm it defined two types of nodes such as hubs and outliers. Fig. 17 showed that nodes 2, 3, 4, 11, 12, 17, 18, 25, 27, 28, 30, 44, 53, 56, 58, 59, 62, 78, 84, 85, 111, 118, 124 and 134 represented hubs while nodes 7, 8, 22 and 49 represented outliers. Compared to the predefined community structure, exclusive the two special types of nodes the

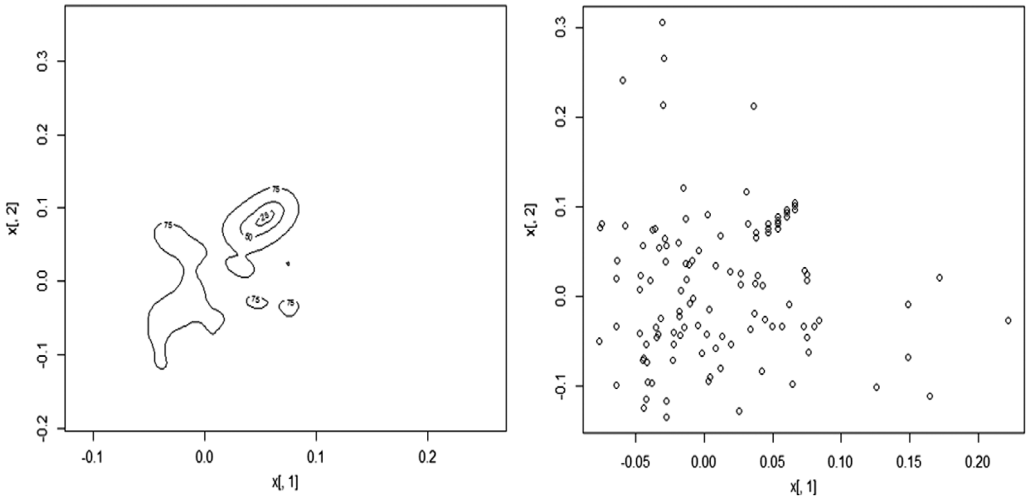

Fig. 14. The derived optimal bandwidth by Sheather-Jones plug in selector for the two dimensional Euclidean data points related to the benchmark network.

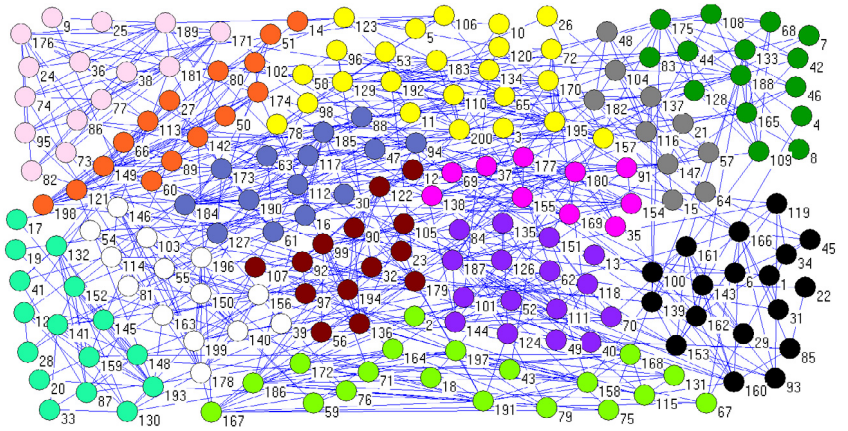

Fig. 15. The predefined communities of the benchmark network.

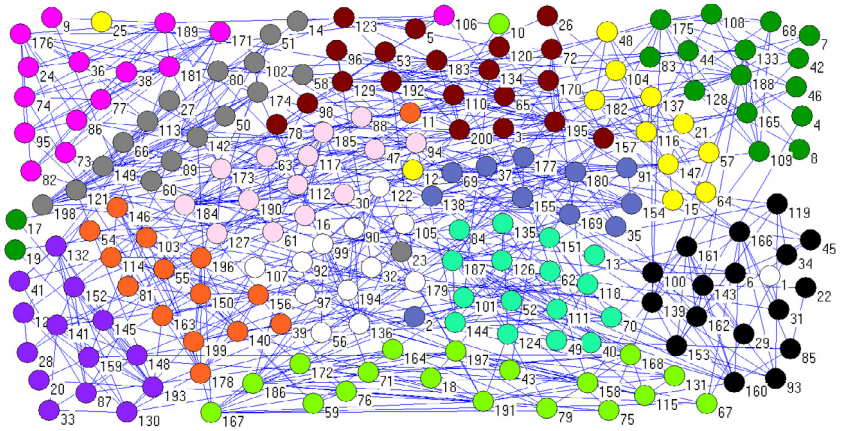

Fig. 16. Benchmark network divided by the proposed algorithm. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

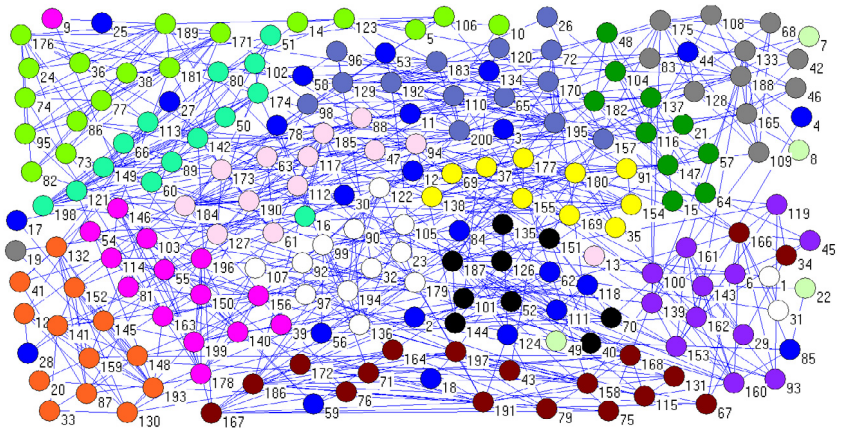

Fig. 17. Benchmark network divided by the comparative algorithm gSkeletonClu. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

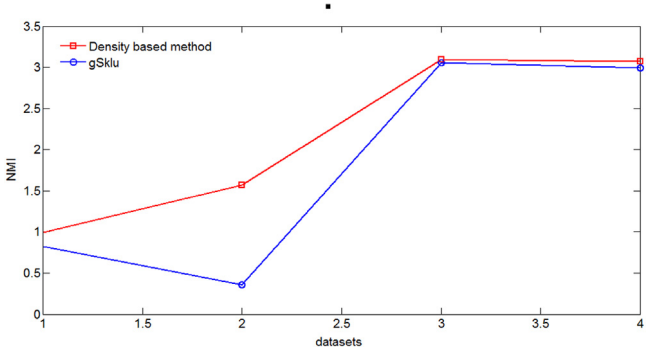

Fig. 18. The NMI value of different networks for our method and gSkeletonClu.

remaining nodes were divided into 13 communities. With the assumption that the two special type of nodes were included in the nearby communities, it also can be seen that most of the communities discovered by the comparative algorithm was different from the predefined partitions. Moreover, the result is less consistent than our algorithm. For example, nodes 1, 31, 34 and 166 had not been rightly included in the community painted in black of the predefined community structure. While nodes 5, 10, 106 and 123 had not divided in the right community of the predefined community painted by yellow. Similar situations can be easily seen from Fig. 17. Therefore, the community structures discovered by gSkeletonClu was much more different from the predefined situation.

5.2. Criterion for accuracy evaluation

In our experiments, an information-theoretic-based measurement named normalized mutual information (NMI) [6] was introduced to evaluate the quality of clusters generated by different methods. It is currently widely applied in measuring the performance of clustering algorithms. The formal definition of the measurement metric NMI is as formula (17) shown

$$NMI = \frac{-2\sum_{i,j} N_{ij} \log\left(\frac{N_{i,j}N}{N_{i.}N_{.j}}\right)}{\sum_i N_{i.} \log\left(\frac{N_{i.}}{N}\right) + \sum_j N_{.j} \log\left(\frac{N_{.j}}{N}\right)} \tag{17}$$

In which N represents the confusion matrix, N_{ij} is the number of vertices in the intersection of cluster X_i and Y_j. While $N_{i.}$ is the sum over row i of N. Similarly, $N_{.j}$ is the sum over column j of N.

Specific to the evaluation of community detection methods, NMI measures the similarity between the revealed structure and the ground truth. The higher the NMI value is, the more similar the detected structure to the true partition is. For the above three real world networks and one benchmark network, the NMI values by the proposed and the comparison algorithms were computed. Obviously, in Fig. 18 it can be seen that our method could achieve a comparable result. More detailed, it showed that the proposed algorithm could achieve a higher NMI value on karate club network, dolphin network and the benchmark network. While for American college football network, the NMI value was close to each other.

6. Conclusions and future work

In this paper, we presented a generalized density based clustering method to detect community structure. When individualized in the community discovering problem, it is normal that the number of related parameters are reduced. Moreover, for the required parameter we found a competitive method named plug-in selector to estimate the optimal parameter value. The experiments demonstrate that the generalized density based clustering method could receive a relatively reasonable community structure without detailed consideration on parameters. In the future, we will do further research in its scalability. Which shows up in two aspects. One is that with the specific properties of density based clustering it is able to find arbitrary size communities. The other is that it is well matched the advantages of spectral analysis and density based clustering algorithm when dealing with the high dimensional dataset.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61272109, 61502350).

References

[1] M. Girvan, M.E.J. Newman, Natl. Acad. Sci. 99 (2002) 7821–7826.
[2] M.E.J. Newman, Proc. Natl. Acad. Sci. 98 (2001) 404–409.
[3] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, D. Parisi, Proc. Natl. Acad. Sci. 101 (2004) 2658–2663.
[4] M.E.J. Newman, M. Girvan, Rev. E 69 (2004) 026113.
[5] A. Clauset, M.E.J. Newman, C. Moore, Phys. Rev. E 70 (2004) 066111.
[6] L. Danon, A. Diaz-Guilera, J. Duch, A. Arenas, J. Stat. Mech. Theory Exp. 09 (2005) P09008.
[7] A. Lancichinetti, S. Fortunato, F. Radicchi, Phys. Rev. E 78 (2008) 046110.
[8] A. Medus, G. Acuna, C.O. Dorso, Physica A 358 (2005) 593–604.
[9] M. Rosvall, C.T. Bergstrom, Proc. Natl. Acad. Sci. USA 105 (2008) 1118–1123.
[10] S. Fortunato, Phys. Rep. 486 (2010) 75–174.
[11] X. Xu, N. Yuruk, Z. Feng, T. Schweiger, Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2007, pp. 824–833.
[12] J. Huang, H. Sun, Q. Song, H. Deng, J. Han, IEEE Trans. Knowl. Data Eng. 25 (2013) 1876–1889.
[13] U. Von Luxburg, Stat. Comput. 17 (2007) 395–416.
[14] C. Alzate, J.A.K. Suykens, IEEE Trans. Pattern Anal. Mach. Intell 32 (2010) 335–347.
[15] A.Y. Ng, M.I. Jordan, Y. Weiss, NIPS (2001) 849–856.
[16] R.R. Nadakuditi, M.E.J. Newman, Phys. Rev. Lett. 108 (2012) 188701.
[17] J. Huang, Technical report Max Planck Institute for Biological Cybernetics, 2005.
[18] Y. Cao, D.R. Chen, Appl. Comput. Harmon. Anal. 30 (2011) 319–336.
[19] S. White, P. Smyth, Proc. SIAM Int'l Conf. Data Mining, 22005.
[20] B.W. Silverman, Chapman and Hall London, 1986.
[21] J. Sander, M. Ester, H. Kriegel, X. Xu, Data Min. Knowl. Discov. 2 (1998) 169–194.
[22] H.P. Kriegel, P. Krer, J. Sander, A. Zimek, Data Min. Knowl. Discov. 1 (2011) 231–240.
[23] M. Ester, H. Kriegel, J. Sander, X. Xu, Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, 1996, pp. 226–231.
[24] T. Takada, Econom. Bull. 3 (2001b) 1–10.
[25] J.N. Hwang, S.R. Lay, A. Lippman, IEEE Trans. Signal Process. 42 (1994) 2795–2810.
[26] S.J. Sheather, M.C. Jones, J. Royal Stat. Soc. Ser. B 53 (1991) 683–690.
[27] J.E. Gentle, Springer New York, 2001.
[28] G.H. Givens, J. Cetacean Res. Manage. 5 (2003) 39–44.
[29] W. Hardle, J.S. Marron, J. Multivariate Anal. 20 (1986) 91–113.
[30] P.J. Davis, P. Rabinowitz, New York, 1984.
[31] B.W. Silverman, Chapman&Hall London, 1986.
[32] W. Hardle, D. Scott, Comput. Statist. 7 (1992) 97–128.
[33] T. Duong, M.L. Hazelton, J. Nonparametr. Stat. 15 (2003) 17–30.
[34] M. Gong, J. Liu, L. Ma et al, Physica A 403 (2014) 71–84.
[35] D. Lusseau, Proc. R. Soc. London Ser. B: Biol. Sci 270 (2003) S186–S188.
[36] D. Lusseau, M.E.J. Newman, Proc. R. Soc. London Ser. B: Biol. Sci. 271 (2004) S477–S481.
[37] A. Lancichinetti, S. Fortunato, F. Radicchi, Phys. Rev. E 78 (2008) 046110.