The Tenth International Symposium on Operations Research and Its Applications (ISORA 2011)
Dunhuang, China, August 28–31, 2011

A SVC-based Method to Detect Multi-scalar Noisy Communities in Complex Networks*

Jun-Fei Zhao[1] Jiguang Wang[2,†] Xiang-Sun Zhang[1,‡]

[1]Academy of Mathematics and Systems Science,
Chinese Academy of Sciences, Beijing 100190, China.
[2]Beijing Institute of Genomics,
Chinese Academy of Sciences, Beijing 100029, China.

Abstract Due to the prevalence of community phenomenon in social and biological networks, community detection is becoming one of the most important problems in complex network research. Although many methods have been developed to partition networks, most of them are based on heuristic algorithms, which do not ensure the stable solutions. In this paper, we design an accurate method based on support vector clustering, which can detect the multi-scalar community in noisy networks. Since our method formulates the primal problem as a quadratic programming, we can get accurate solution in large scale networks. More importantly, the new method can efficiently exclude neutral nodes or noisy nodes, i.e. nodes do not belong to any communities.

Keywords Support Vector Clustering; Community Detection; Neutral Nodes

1 Introduction

In recent years, we have observed a great devotion to the study of complex networks for the reason that many interesting systems can be represented as networks composed of vertices and edges. Example ranges from the internet, social networks to biological networks. The prolific progress in the study of complex networks has revealed many interesting topological properties such as scale-free [2], small-world [13, 11] and network transitivity [1]. Here, we are interested in another property, which is also common to many complex networks, community or modular structure [14, 7], i.e., networks consisting of specific and relatively separate dense subnetworks. The detection of modules and the analysis of community structure can help us understand the design principle of complex system and the functionality of every component. For example,the communities of biological networks are often sets of components which have similar functions.

There have been many methods which can be used to find communities in the network. Newman firstly propose one quality function known as "modularity", which measures the quality of the possible divisions of a network [12]. Based on this measure, there are many heuristic methods optimizing "modularity" to obtain the optimal division. However, Santo Fortunato has proved that modularity optimization has resolution limit and

*The paper is supported by No. 60873205 Research Grant supported by NSFC.
†Corresponding to wangjg@big.ac.cn
‡Corresponding to zxs@amt.ac.cn

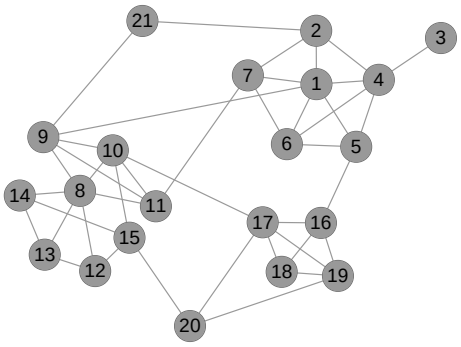


Figure 1: The network with “noise nodes”

may fail to identify modules smaller than a scale which depends on the global property of the network, even when the modules are well-defined [6]. Apart from modularity-based methods, another series of methods is clustering-based. Through a wide variety of similarity measure, one can use classical clustering methods to detect the communities in a network, such as K-means and hierarchical clustering [17]. There are also some methods based on information theory [18] and flow simulation [4]. These methods have dramatically improved the efficiency and accuracy of module detection, but like the heuristic methods, they cannot guarantee the optimality and uniqueness of the result [5].

In many networks, there may be some nodes whose degree of membership to any community is very low. These nodes are “noisy nodes” of the networks and should be excluded in community detection. However, until now, there is still no such method which can deal with this kind of networks.Figure 1 is one example with “noisy nodes”. Obviously, this network comprises of three “core communities”, while vertex 3, 20 and 21 do not belong to any community.

In this paper, based on support vector clustering [3], we propose one novel method for community detection and apply it to some artificial and real-world networks. Comparing with previous methods in terms of stability and scalability, it has obvious advantages. Our method can detect network community stably, which means that our method has unique solution because it formulates the original problem to a quadratic programming. Therefore, it is scalable to the large-scale problem. Additionally, through tuning the parameter, our method can detect multi-scalar network communities [9]. Owing to the property of support vector clustering, our method is suitable to analyze the networks with “noisy nodes” and detect the “core community” of a network.

2 Methods

2.1 Diffusion Kernel

Before applying support vector clustering to the problem of network community detection, we should define vertex distance on the network. There have been many such measures [5]. For example, measures based on the concept of structural equivalence. Another class of measures is based on properties of random walks on networks. In this paper,

we use diffusion kernel to define distance between every two vertexes [8].

In 2002, Kondor *et al.* proposed the concept of diffusion kernel which is based on the matrix exponentiation idea and can capture the long-range relationships between data points induced by the local structure of the graph.

An undirected,unwigheted graph G comprises two essential elements: vertex set V and edge set E which is the set of unordered pairs $\{v_1, v_2\}$, where $\{v_1, v_2\} \in V$ whenever the vertices v_1 and v_2 are joined by an edge(denoted $v_1 \sim v_2$). Given $G = \{V, E\}$, we can define its (negative) Laplacian matrix

$$H_{ij} = \begin{cases} 1 & \text{for } i \sim j \\ -d_i & \text{for } i = j \\ 0 & \text{otherwise} \end{cases}$$

where d_i is the degree of vertex i (number of edges emanating from vertex i).

The diffusion kernel of G is defined as the exponentiation of its Laplacian matrix

$$DK_\beta = e^{\beta H} = \lim_{n\to\infty}(1 + \frac{\beta H}{n})^n$$

where the limit always exists and is equivalent to

$$e^{\beta H} = I + H + \frac{1}{2!}H^2 + \frac{1}{3!}H^3 + \cdots.$$

Obviously, K_β is symmetric and positive semidefinite, hence it is a candidate for a kernel.

In the normalized diffusion kernel,DK_{ij} mean the similarity between data point i and j. So we use $1 - DK_{ij}$ to represent the distance between data point i and j.

2.2 Support Vector Clustering

Support vector clustering (SVC) is a kernel-based unsupervised learning clustering method and was proposed by Asa Ben-Hur *et al.* In this algorithm, data points are mapped by means of a Gaussian kernel to a high dimensional feature space, where we search for the minimal enclosing sphere. This sphere, when mapped back to data space, can separate into several contours enclosing data points. The points enclosed by each contour are associated with the same cluster.

Let $x_j \subseteq \chi$ be a data set with N points, with $\chi \subseteq \mathbb{R}^d$, the data space. Φ denotes the nonlinear transformation from χ to some high dimensional feature-space. SVC can be formulated as such a problem:

$$min_{R,a,\xi}\ R^2 + C\sum \xi_j$$

subject to

$$\|\Phi(x_j) - a\|^2 \leq R^2 + \xi_j,\ \xi_j \geq 0.$$

where C is a constant, ξ_j a slack variable.

Turning this problem to its dual form, we get such a quadratic programming of the variables β_j:

$$min_\beta\ \sum_{i,j}\beta_i\beta_j K(x_i, x_j) - \sum_j K(x_j, x_j)\beta_j$$

subject to

$$0 \leq \beta_j \leq C, \sum_j \beta_j = 1.$$

Paper[3] proposed that we should use the Gaussian kernel in SVC:

$$K(x_i, x_j) = e^{-q\|x_i - x_j\|^2}.$$

where q is called the width of Gaussian kernel. Because we use diffusion kernel to define distance measure between two vertexes, we compute Gaussian kernel in this paper as:

$$K(x_i, x_j) = e^{-q(1-DK_{ij})^2}.$$

After solving this quadratic programming, we can classify the nodes by the values of β_j and ξ_j. The node X_j with $0 < \beta_j < C$ is referred to as *support vector* or SV because it is mapped to the surface of the feature space sphere. A point X_j with $\xi_j > 0$ and $\beta_j = C$ lies outside the feature-space sphere. It will be called a *bounded support vector* or BSV.

2.3 From Clustering to Community Detection

We try to apply SVC to the problem of network community detection because it has some good properties. For example, decreasing the width of Gaussian kernel increases the number of contours, which corresponds to clusters in multiple scale. This property can enable us to find multi-scale communities in the network when we apply this method to the community detection problem, because many real-world networks tend to have a hierarchical community structure[15].

What's more exciting is that by using a soft margin it can deal with outliers(BSV). We can exploit this property to deal with the problem which section 1.1 proposes. In a network, after applying SVC to it, the "noisy nodes" just correspond to those BSV. In section 3.1, we show this effect of our method by applying it to the network figure 1 shows.

2.4 Community Assignment

In original SVC, for each point X, the distance of its image in feature space from the center of the sphere can be computed as:

$$R^2(x) = K(x,x) - 2\sum_j \beta_j K(x_j, x) + \sum_{i,j} \beta_i \beta_j K(x_i, x_j).$$

The radius of the sphere is:

$$R = \{R(x_i) \mid x_i \text{ is a support vector}\}.$$

The method of differentiating between points that belong to different clusters is based on such an observation: given a pair of data points that belongs to different components (clusters), any path that connects them must exit from the sphere in feature space. In other words, there must be one point y such that $R(y) > R$. However, in a network, there is not such a real path between two vertexes as in a Euclidean space. In our method, we overcome this obstacle by defining a "virtual path" between every two vertexes.

For arbitrary two vertexes x, y, $K(x,:), K(y,:)$ denote the distance vectors, which comprise their distance from all the nodes of the network(include themselves). We define a node Z on the "virtual path" between x and y, by the linear combination of $K(x,:)$ and $K(y,:)$

$$K(z,:) = \lambda K(x,:) + (1-\lambda)K(y,:),\ 0 \leq \lambda \leq 1.$$

In addition,

$$K(z,z) = 0.$$

Then, we can define adjacency matrix $A_{i,j}$ between pairs of points x_i and x_j whose images lie in or on the sphere in feature space:

$$A_{ij} = \begin{cases} 1 & \text{if } R(z) \leq R, \text{when } \lambda \text{ run from 0 to 1} \\ 0 & \text{otherwise.} \end{cases}$$

In numerical experiments, we use twenty decile points of $[0,1]$ to compute A_{ij}. Communities are now defined as the connected components of the graph induced by A.

3 Results

We test our method by applying it on four kinds of synthetic network and to four real-world networks. The results show our method are more efficient than previous ones.

3.1 Synthetic Networks

3.1.1 The Network with "Noise Nodes"

Firstly, we show that our method can deal with "noise nodes" and detect the "core communities" by applying our method to the network in figure 1. As figure 2 shows, our result is consistent with intuition: vertex 3, 20 and 21 are detected as "noise nodes"; the remaining nodes are divided into three "core communities". This result is obtained when $C = 0.1$ and $q = 2.9$.

3.1.2 Rings of Cliques and Y-shaped Network

We address that our method can be used to overcome resolution limit, by using two synthetic networks proposed by Fortunato and Barthl̀elemy.

Figure 3 shows two Benchmarking of the procedure to address resolution limitations. The first network is called rings of cliques which is made out of identical cliques connected by single links. It has been proved that If the number of cliques is larger than $\sqrt{L}$, modularity optimization would lead to a partition where the cliques are combined into groups of two or more (represented by a dotted line).

The other is called Y-shaped network comprising of four pairwise identical cliques (complete graphs with m and p < m nodes, respectively); if m is large enough with respect to p (e.g.m = 20, p = 5), modularity optimization merges the two smallest modules into one (shown with a dotted line).

Our method can yield the accurate number of communities. The computational results of our method are shown in figure 4.

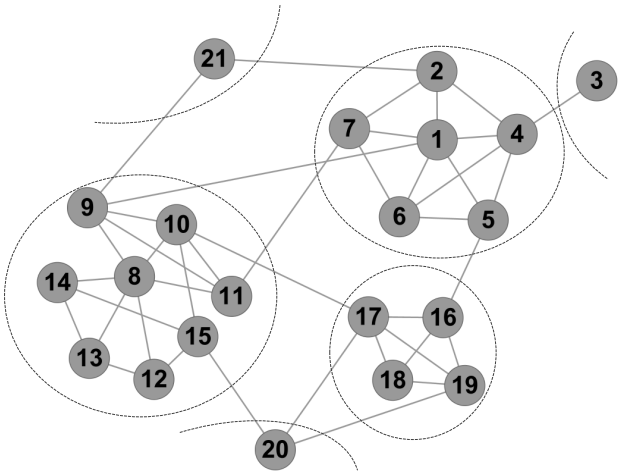


Figure 2: The result of our method by applying it to the network of Figure 1.

3.1.3 Computer-Generated Graphs

Computer-Generated Graphs are another type of widely used exemplar networks for testing the effectiveness of community detection algorithms. The experiment is designed by Girvan and Newman[7]. Each graph was constructed with 128 vertices divided into four communities of 32 vertices each. Edges were placed between vertex pairs independently at random. The probabilities were chosen so as to keep the average degree of a vertex equal to 16 and the average number of edges of each node connecting to nodes of other communities to be k_{out}.

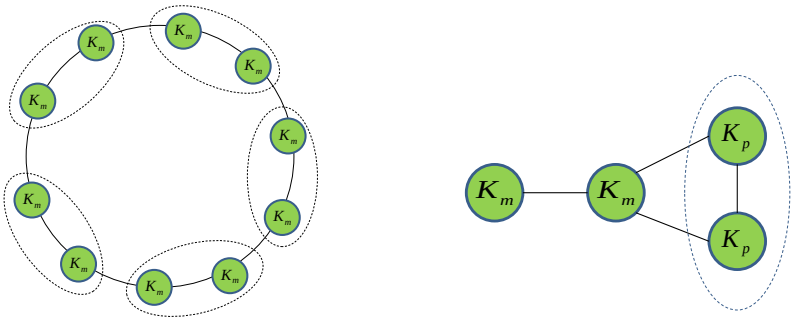
Figure 5 shows the fraction of nodes which are classified into correct community by our method and two other methods. Our method has better performance than that proposed by Girvan and Newman. When $k_{out} < 5.5$, the present method performances as well as D-value method[10].

3.2 Real-world Networks

3.2.1 The Karate Club Network

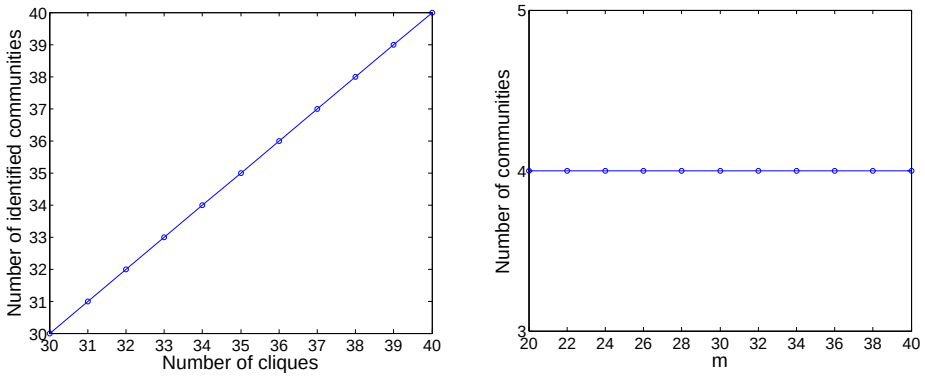
Now, we turn to apply our method on real-world networks. The first example is the famous karate club network analyzed by Zachary[16]. The network consists of 34 members of a karate club as nodes and 78 edges representing friendship between members of the club which was observed over a period of two years. Due to a dispute between the club's administrator and its principal karate teacher, the club eventually split into two smaller clubs, centered around the administrator and the teacher.

Figure 6 shows the result of our method. As the scale parameter of the Gaussian kernel, q , is increased, the network splits into three smaller network. Compared with Figure 4, our result is reasonable:the module on the left is exactly the smaller club centered



(a) Rings-of-cliques (b) Y-shaped Network

Figure 3: Benchmarking of the procedure to address resolution limitations.



(a) Rings-result (b) Y-result

Figure 4: The results of Benchmarking of the procedure to address resolution limitations.

around the administrator;the two smaller modules on the right is also reasonable from the topology of the network.

3.2.2 Football Team Network

The second real-world network we consider is the college football network of the United States [7]. The nodes in the network represent the 115 teams, while the edges represent 613 games played in the course of the year. This network has natural community structure. The teams are divided into conferences of 8-12 teams each and generally games are more frequent between members of the same conference than between teams of different conferences. Interconference play is not uniformly distributed; teams that are geographically close to one another but belong to different conferences are more likely to play one another than teams separated by large geographic distances.

Applying our method to this network,we find that it can detect the natural commu-

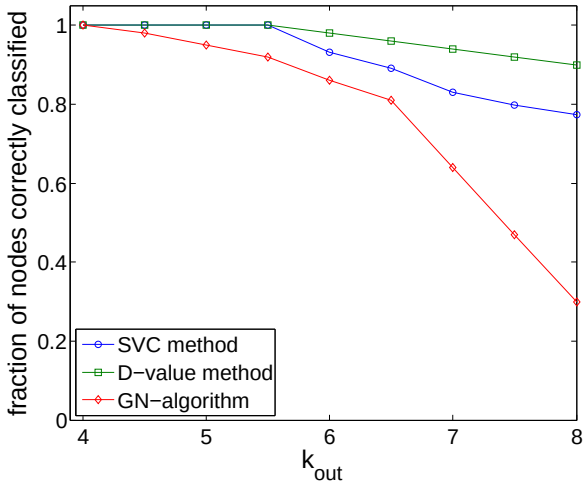


Figure 5: Test of various methods on computer-generated networks with known community structures. It is a plot of the fraction of nodes correctly classified with respect to k_{out}.

nity structure with a high degree of success. Figure 7 shows the result calculated by our method. Due to the fact that there are more games with the teams in the classified communities than the teams in their own conferences, node 29,43,91,111 are incorrectly classified. Our method partitions conference 10(12,25,51,60,64,70,98) into two parts. These two parts are two cliques with only one edge between them, so this partition is also reasonable. In addition, because node 37,59,60,64,98 form a clique, they were detected as a new community.

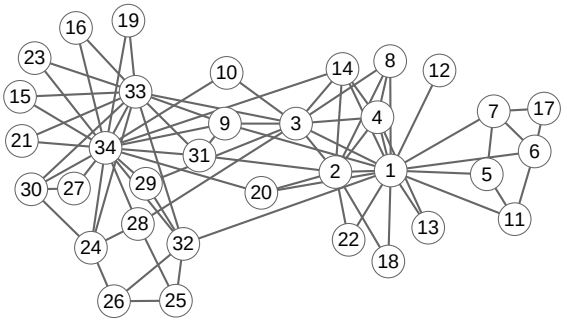
4 Conclusion and Discussion

In this paper, we present a method based on support vector clustering to detect the community structure of a network. The result shows, it can detect the multi-scalar community of a network, which enables us to visualize the community structure of the network more exactly. In addition, our method can deal with "noisy nodes" and detect the "core community". These two kinds of nodes, normal and noisy, may correspond to different functions in social and biological networks, then future implications remain to be explored. As a new clustering method, SVC has not yet been applied broadly. Our work is only a simple step. There are still many applications worth trying.

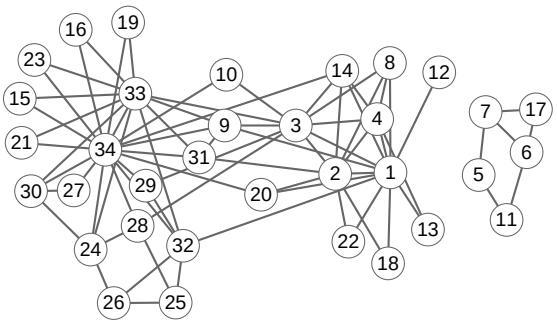
References

[1] A.L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509, 1999.

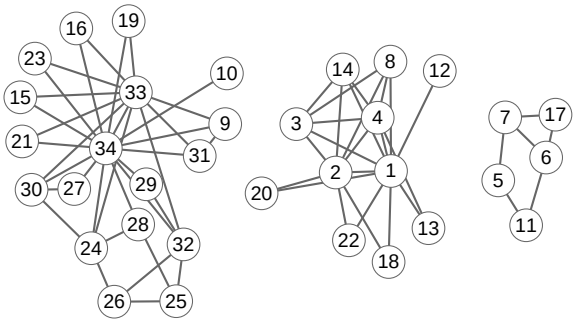
[2] A.L. Barabási and E. Bonabeau. Scale-free networks. *Scientific American*, 288(5):50–9, 2003.



(a) β =0.5,q=3.0



(b) β =0.5,q=3.2



(c) β =0.5,q=3.5

Figure 6: Community detection of the karate club network using SVC with C=2.

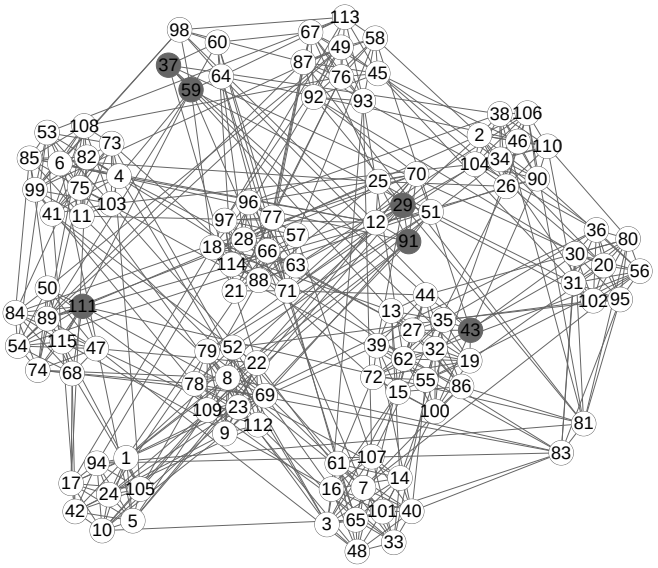


Figure 7: The result calculated by our method,dark color means the nodes are “wrongly” divided.

[3] A. Ben-Hur, D. Horn, H.T. Siegelmann, and V. Vapnik. Support vector clustering. *The Journal of Machine Learning Research*, 2:125–137, 2002.

[4] S.V. Dongen. Graph clustering by flow simulation. *Computer Science Review*, 1(1):27–64, 2000.

[5] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, 2010.

[6] S. Fortunato and M. Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36, 2007.

[7] M. Girvan and M.E.J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821, 2002.

[8] R.I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pages 315–322. Citeseer, 2002.

[9] R. Lambiotte. Multi-scale modularity in complex networks. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2010 Proceedings of the 8th International Symposium on*, pages 546–553. IEEE, 2010.

[10] Z. Li, S. Zhang, R.S. Wang, X.S. Zhang, and L. Chen. Quantitative function for community detection. *Physical Review E*, 77(3):036109, 2008.

[11] S. Milgram. The small world problem. *Psychology today*, 2(1):60–67, 1967.

[12] M.E.J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577, 2006.

[13] I.S. Pool and M. Kochen. Contacts and influence. *Social networks*, 1(1):5–51, 1978.

[14] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9):2658, 2004.

[15] E. Ravasz, A.L. Somera, D.A. Mongru, Z.N. Oltvai, and A.L. Barabási. Hierarchical organization of modularity in metabolic networks. *Science*, 297(5586):1551, 2002.

[16] W.W. Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473, 1977.

[17] S. Zhang, X.M. Ning, and X.S. Zhang. Graph kernels, hierarchical clustering, and network community structure: experiments and comparative analysis. *The European Physical Journal B-Condensed Matter and Complex Systems*, 57(1):67–74, 2007.

[18] E. Ziv, M. Middendorf, and C.H. Wiggins. Information-theoretic approach to network modularity. *Physical Review E*, 71(4):046117, 2005.