Artif Intell Rev
DOI 10.1007/s10462-012-9325-3

Detecting network communities using regularized spectral clustering algorithm

Liang Huang · Ruixuan Li · Hong Chen · Xiwu Gu · Kunmei Wen · Yuhua Li

Abstract The progressively scale of online social network leads to the difficulty of traditional algorithms on detecting communities. We introduce an efficient and fast algorithm to detect community structure in social networks. Instead of using the eigenvectors in spectral clustering algorithms, we construct a target function for detecting communities. The whole social network communities will be partitioned by this target function. We also analyze and estimate the generalization error of the algorithm. The performance of the algorithm is compared with the standard spectral clustering algorithm, which is applied to different well-known instances of social networks with a community structure, both computer generated and from the real world. The experimental results demonstrate the effectiveness of the algorithm.

Keywords Community detection · Graph laplacian · Eigenvector · Spectral clustering algorithm · Regularized spectral clustering algorithm

1 Introduction

Recent years, social networks have attracted more and more attention on the internet. A typical social network can be represented as $G = (V, E)$, where V represents the person and E represents the relationship between them. With the prevalence of the web 2.0, online social network service (SNS) websites, such as LiveJounal, delicious, facebook and twitter, are getting more and more popular. Almost all these SNS websites maintain strong community structure according to the people with the same background or interests. Nodes in the same community tend to have close connectivity. Detecting the community structure

L. Huang · R. Li (✉) · X. Gu · K. Wen · Y. Li
Huazhong University of Science and Technology, Wuhan, China
e-mail: rxli@hust.edu.cn

H. Chen
Huazhong Agricultural University, Wuhan, China

Published online: 08 March 2012

in social networks is an issue of considerable practical interest that has received a great deal of attention.

Community detection (Barthélemy and Fortunato 2007) is important for many reasons. Clustering web clients who have similar interests and are geographically near to each other may improve the performance of services provided on the web. We can also find benefits in the case of recommendation system. Through identifying the groups of customers with similar interests in the network, one can set up efficient recommendation systems, and guide the customers with the list of items of the retailer. This technique can enhance the business opportunities in the networks. Similar benefits can be obtained for a lot of applications in social networks and other complex systems.

Additionally, community detection can help us classify the nodes in the networks according to their structural position in the communities (Watts and Strogatz 1998). Nodes with a central position in their communities usually have an important function of control and are stable within the community. Nodes lying at the boundaries between groups play an important role of mediation and maintain the relationships and exchanges between different communities. Such a classification seems to be more meaningful in social and complex networks for further work.

Many algorithms of identifying the communities have been proposed in the past few years (Lee et al. 2008). The traditional method for detecting community structure in social networks is hierarchical clustering. By choosing $\mathbf{N}$ clustering centers, the hierarchical clustering uses the nodes that closely connected the centers each time, and the whole community structure can be found in this way.

Newman (Newman and Girvan 2004) proposed an iterative, divisive method based on the progressive removal of links with the largest betweenness (see Newman and Girvan 2004). The algorithm measures the fraction of all shortest paths passing on a given link, or alternatively, the probability that a random walk on the network runs over that link. By removing links with high betweenness, one progressively splits the whole network into disconnected components, until the network is decomposed into communities consisting of one single vertex. Unluckily, it has an evident disadvantage that it has extraordinarily high computational cost.

Fortunato (Fortunato et al. 2004) developed an algorithm of hierarchical clustering that consists in finding and removing iteratively the edge with the highest information centrality. The algorithm shows that, although it runs to completion in a time $O(n^4)$, it is very effective especially when the communities are very mixed and hardly detectable by other methods.

In addition, Newman and Girvan (Newman 2004) proposed a quantitative method called modularity to identify the network communities. Unlike the other methods, modularity method defines a quantity function called Q. One can identify the communities by optimizing the quantity function Q. It seems to be an effective method to detect communities in networks. However, Fortunato and Barthélemy (Fortunato 2010) claimed that the size of a detected module depends on the size of the whole network, which seriously limits this method. To solve this problem, Li (Chen et al. 2008) developed another quantitative method called the modularity density to quantify the communities in social networks. The modularity density method considers both nodes and edges in the quantity function. However, as we can see, the modularity density is still an NP-hard problem.

An alternative way to tackle the problem is spectral clustering (Chung 1997; Donetti and Munoz 2004; Luxburg 2006). Spectral clustering algorithms include all methods and techniques that partition the set into clusters by using the eigenvectors of matrices, like the adjacency matrices or their derived matrices (especially graph laplacian matrix Aggyriou et al. 1950). Spectral clustering algorithm consists of a transformation of the initial set of

objects into a set of points in space, whose coordinates are elements of eigenvectors. The set of points is then clustered via standard techniques, such as k-means.

In our work, we introduce the regularized spectral clustering algorithm (Belkin et al. 2006, 2004) to handle the large scale online social networks. Clearly, with the explosive increase of network, the computational cost of the spectral clustering is getting more and more unacceptable. It turns out to be a frightening job for spectral clustering algorithm which needs frequently matrix operation. Instead of using vectors as in spectral clustering algorithm, the regularized spectral clustering algorithm chooses sample matrix of the social network to construct a target function which can partition the social network naturally. With the regularized spectral algorithm, the whole network can be clustered with relatively smaller sample matrix to alleviate the whole computational cost. The experiments show the proposed method achieves good results with relatively smaller computational cost compared to the spectral clustering algorithm.

The convergence rate of regularized clustering algorithm has been well understood in Cao and Chen (2011). In our work, we focus on the estimate of generalization error based on a Bernstein-type inequality of U-statistics. Error analysis of this method is directly and simply compared with sample error estimated in Cao and Chen (2011).

The rest of the paper is organized as follows. In Sect. 2, we introduce the graph laplacian and the standard spectral clustering algorithm. Section 3 presents the details of the regularized spectral clustering algorithm and the generalization error of the algorithm. The results of the standard and the regularized spectral clustering algorithms are tested in Sect. 4. Finally, Sect. 5 concludes the paper.

2 Graph laplacian and spectral clustering algorithm

2.1 The similarity graph

The social networks adopted in this work are restricted to the undirected and unweighted social graph $G = (V, E)$ with a set of nodes $x_1, \ldots x_n$, e.g. the Zachary karate club and the dolphin network. Aimed at turning 'eye inspection' of communities into a more quantitative measure, the explicit introduction of a metric or similarity measure is required. There are many methods can be used to quantify the social network. The most straightforward choice would be the Euclidean distance.

We use the work of Xing et al. (2003) to turn the pairwise similarity, which is represented as the shortest distance, into Euclidean distance. There are several popular ways to construct the similarity graph of the given data with Euclidean distance, e.g, the ε-neighborhood graph, the k-nearest neighbor graph and the fully connected graph. The ε-neighborhood graph only connects the nodes whose pairwise distances are smaller than ε and the k-nearest neighbor graph connect node v_i with node v_j if v_j is among the k-nearest neighbors of v_i. The fully connected graph simply connects all points with each other using a similarity function.

In this paper, the fully connected graph is adopted with the following two similarity functions.

$$s(x_i, x_j) = \exp(-\|x_i - x_j\|^2/(2\sigma^2)) \tag{1}$$

$$s(x_i, x_j) = 1/\|x_i - x_j\|^2 \tag{2}$$

where Formula (1) is the Gaussian similarity function and $\|x_i - x_j\|$ represents the Euclidean path between x_i and x_j. The similarity graph constructed in this way is called the Gaussian

similarity. We also use Formula (2) to construct the similarity graph for the comparison. The clustering results of these two kinds of similarity functions will be compared in the experiments.

2.2 The graph laplacian

Spectral clustering algorithm uses the eigenvectors of the graph laplacian matrices to cluster the social network. We introduce the graph laplacian in this subsection. As mentioned above, the similarity graph is constructed by the Gaussian kernel function or the simple similarity function in this paper. The similarity matrix of the social graph is denoted as $W = (w_{i,j})_{i,j=1}^{n}$, where w_{ij} is the similarity or the weight between the nodes. As G is undirected, $w_{ij} = w_{ji}$. The degree d_i of $v_i \in V$ is defined as $d_i = \sum_{j=1}^{n} w_{ij}$.

The degree matrix D of V can be defined as the diagonal matrix with the degrees $d_1, \ldots, d_n$ on the diagonal. The simplest form of graph laplacian matrix is defined as follows.

$$L = D - W \tag{3}$$

The laplacian matrix in Formula (3) is the unnormalized form. The spectral clustering algorithm uses the eigenvalues and the corresponding eigenvectors to cluster the social network.

There are two other matrices which are called normalized graph laplacians in the literatures. Both matrices are closely related to each other and are defined as follows.

$$L_{sym} := D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2} \tag{4}$$

$$L_{rw} := D^{-1} L = I - D^{-1} W \tag{5}$$

The first matrix is denoted by L_{sym} as it is a symmetric matrix, and the second one by L_{rw} as it is closely related to a random walk. In this paper, both unnormalized and normalized graph laplacian will be included in the experiments for the comparison.

2.3 The spectral clustering algorithm

Algorithm 1 Standard spectral clustering algorithm.

Require:
The social matrix of social graph $G = (V, E)$;
Ensure:
The k clusters of the social network.
1: Compute the similarity matrix of the social matrix;
2: Compute the graph laplacian matrix of the similarity matrix;
3: Compute the eigenvalues and the eigenvectors of the laplacian matrix;
4: Use the eigenvectors of the first k eigenvalues to cluster the network;
5: **return** The clusters of the given social network.

The spectral clustering algorithm uses the laplacian matrix of the given social network data to find the community structure. The eigenvalues of laplacian matrix are either zero or positive (Donetti and Munoz 2004). If G is the given network data, the multiplicity k of the eigenvalue 0 of laplacian matrix equals the number of connected components $A_1, \ldots, A_k$ in the graph. Therefore, if the graph under analysis is connected, there is only one zero eigenvalue corresponding to the constant eigenvector. In contrast, for non-connected graphs, the laplacian matrix is block diagonal. Each block is the Laplacian matrix of a subgraph.

In the real social network, the subgraph or the community is not fully disconnected, which means some links still exists between different communities. This leaves only one

trivial eigenvector with eigenvalue 0 and several 1 approximate linear combinations of the old ones with slightly non-vanishing eigenvalues (Donetti and Munoz 2004). Therefore, the normalized spectral clustering uses the first several eigenvectors to detect the communities of the network data (with the smallest eigenvalues).

Algorithm 1 is the standard spectral clustering algorithm. The details of spectral clustering algorithm can be found in Luxburg (2006). In this paper, the regularized spectral clustering algorithm is compared with this algorithm in two standard benchmark social network data: the Zachary karate club (Zachary 1977) and the dolphin network (Boisseau et al. 2003). In order to makes the experiment more comparative, two similarity functions (the simple and Gaussian similarity functions) are adopted in the experiments for social graph construction, both unnormalized and normalized laplacian matrices are also included in the comparison.

2.4 The modularity function

To verify the regularized spectral algorithm, it is necessary to compare the results of these two algorithms. In this paper, we use the concept of modularity to quantify the results. In Newman and Girvan (2004), Newman defined the modularity as follows. Given a network division, let e_{ii} be the fraction of edges in the network between any two nodes in the subgroup i, and a_i be the total fraction of edges with one node in group i. The modularity Q is then defined as follows.

$$Q = \sum_j (e_{ii} - a_i^2) \tag{6}$$

Formula (6) measures the fraction of edges that fall between communities minus the expected value of same quantity in a random graph with the same community division. In Newman (2004), Newman obtained the community structure of the network through optimizing the modularity. In this paper, the modularity is narrowed to quantify the clustering results of our algorithm.

3 The regularized spectral clustering algorithm

3.1 The algorithm

In our work, the regularized clustering algorithm is introduced to cluster the given social network data. Regularized clustering algorithm uses a target function $\mathbf{f}$ defined on whole dataset instead of the vectors which is adopted in spectral clustering algorithm to cluster the given data. For details, we refer to Cao and Chen (2011), which is shown as follows.

$$\begin{aligned} \alpha^{\mathbf{x}} = \arg\min_{\alpha \in \mathscr{R}^n} & \frac{1}{n^2}\alpha^T \mathbf{K}L\mathbf{K}\alpha + \gamma \alpha^T \mathbf{K}\alpha, \\ s.t. \ & \frac{1}{n^2}\alpha^T \mathbf{K}D\mathbf{K}\alpha = 1, \\ & \alpha^T \mathbf{K}D\mathbf{1} = 0 \end{aligned} \tag{7}$$

$\alpha^{\mathbf{x}} = (\alpha_1^{\mathbf{x}}, \ldots, \alpha_n^{\mathbf{x}}) \in \mathscr{R}^n$ is the coefficient of the target function in Formula (8). $\mathbf{K}$ is the corresponding similarity graph. $\mathbf{1}$ is the vector of all ones. L and D are the corresponding laplacian and degree matrix. The details are introduced in Sect. 2.2. One can work out the target function f by applying the coefficient α in Formula (8).

$$f(x) = \sum_{i=1}^{n} \alpha_i^{\mathbf{x}} K(x_i, x) \tag{8}$$

Let P be the projection onto the subspace of $\mathbb{R}^u$ orthogonal to $\mathbf{K1}$. One obtains the solution for the linear constrained problem in Formula (7). P can be calculated by solving the generalized eigenvalue problem as follows.

$$P(\gamma \mathbf{K} + \mathbf{K}L\mathbf{K})PV = \lambda P\mathbf{K}^2 PV \tag{9}$$

The details of Formula (9) can be found in Belkin et al. (2006). The final solution is given by $\alpha = PV$, where V is the eigenvector corresponding to the eigenvalues. Similar to the standard spectral clustering algorithm, the chosen eigenvectors of the smallest k eigenvalues can be used to compute the k clustering results.

The regularized spectral clustering algorithm is presented in Algorithm 2.

Algorithm 2 Regularized spectral clustering algorithm.

Require:
The social matrix of social graph $G = (V,E)$;
Ensure:
The k clusters of the social network.
1: Choose the sample nodes of the social matrix;
2: Compute the similarity matrix of the sample nodes;
3: Compute the graph laplacian matrix of the similarity matrix;
4: Compute the generalized eigenvectors and eigenvalues;
5: Use the generalized eigenvectors of the first k generalized eigenvalues to compute the corresponding coefficient α;
6: Compute the target function values of the whole social network data correspond to the respective coefficient α;
7: Use the k vectors of the target function values to cluster the social network;
8: **return** The clusters of the given social network.

3.2 Estimate the generalization error of the algorithm

The convergence rate of regularized clustering algorithm has been well studied in Cao and Chen (2011). In this subsection, we focus on the estimate of generalization error based on a Bernstein-type inequality of U-statistics. Our method of error analysis is directly and simply compared with sample error estimated in Cao and Chen (2011).

Let $\mathscr{X}$ be a compact metric space and ρ be a probability measure on $\mathscr{X}$. Denote $p(x) = \int_{\mathscr{X}} K(x, y)d\rho(y)$ for all $x \in \mathscr{X}$.

The generalization error is defined as follows.

$$\mathscr{E}(f) = \frac{1}{2} \int_{\mathscr{X}} \int_{\mathscr{X}} (f(x) - f(y))^2 K(x, y)d\rho(y)d\rho(x) \tag{10}$$

The generalization error measures the quality of clustering by a function.

Given a set $\mathbf{x} = \{x_i\}_{i=1}^n$ of samples independently drawn according to ρ, the empirical error of a function f is denoted by $\mathscr{E}(f)$ as follows.

$$\mathscr{E}_{\mathbf{x}}(f) = \frac{1}{n^2} f^T L_n f = \frac{1}{2n^2} \sum_{i,j=1}^{n} (f(x_i) - f(x_j))^2 K(x_i, x_j) \tag{11}$$

The clustering algorithm we investigated in this paper is based on a Tikhonov regularization scheme associated with a kernel function. The reproducing kernel Hilbert space $\mathscr{H}_K$ associated with the kernel K is defined to be the closure of the linear span of the set of functions $\{K_x := K(x, \cdot), x \in \mathscr{X}\}$ with the inner product $\langle \cdot, \cdot \rangle_K$ given by $\langle K_x, K_{x'} \rangle_K = K(x, x')$. The reproducing property takes the form $fx = \langle f, K_x \rangle_K, \forall x \in \mathscr{X}, f \in \mathscr{H}_K$. The reproducing property with the Schwartz inequality yields that for all $f \in \mathscr{H}_K$.

$$\|f\|_\infty \leq \kappa \|f\|_K, \quad \text{where} \quad \kappa = \sup_{x \in \mathscr{X}} \sqrt{K(x, x)} \tag{12}$$

The regularized clustering algorithm in Formula (7) can be rewritten by an offline regularization scheme in $\mathscr{H}_K$ (see Cao and Chen 2011).

$$f_{\mathbf{x},\gamma} = \arg \min_{f \in \mathscr{H}_K} \{\mathscr{E}_{\mathbf{x}}(f) + \gamma \|f\|_K\} \tag{13}$$

To estimate the generalization error $\mathscr{E}(f_{\mathbf{x},\gamma})$, we investigate the difference between $\mathscr{E}(f_{\mathbf{x},\gamma})$ and $\mathscr{E}_{\mathbf{x}}(f_{\mathbf{x},\gamma})$. Since $f_{\mathbf{x},\gamma}$ depends on $\mathbf{x}$, we need to introduce the covering number as the measure of assumption space capacity.

For $R > 0$, define $B_R = \{f \in \mathscr{H}_K, \|f\|_K \leq R\}$. We denote the covering number of B_R with the metric $\|\cdot\|_\infty$ by $\mathscr{N}(B_R, \varepsilon)$, where ε is the radius.

For a kernel $K(x, y)$, the integral operator $S_K \colon L^2_\rho \to L^2_\rho$ is defined by

$$S_K f(x) = \int_{\mathscr{X}} K(x, y) f(y) d\rho(y), \quad x \in \mathscr{X} \tag{14}$$

Now we establish an estimate of generalization error $\mathscr{E}(f_{\mathbf{x},\gamma})$ as follows.

Theorem 1 *Assume the second largest eigenvalue λ_2 of S_K is positive and there exists a constant $l > 0$ such that $p(x) \geq l$ for any $x \in \mathscr{X}$. Then for all $\mathscr{E} > 0$, $\delta > 0$, and $n > M_\delta$,*

$$\mathscr{E}(f_{\mathbf{x},\gamma}) \leq \mathscr{E}_{\mathbf{x}}(f_{\mathbf{x},\gamma}) + \varepsilon \tag{15}$$

with confidence at least $1 - 2\mathscr{N}\left(B_1, \frac{\varepsilon}{16\kappa^2 R^2}\right) \exp\left\{-\frac{|n/2|\varepsilon^2}{128\kappa^6 R^4}\right\}$. *Here,*

$M_\delta := \max\{64l^{-2}\kappa^4 \log^2(2/\delta), 64\lambda_2^{-2}\kappa^4 \log^2(2/\delta)\}$ *and* $R := \sqrt{r^{-1} + 4l^{-1}\lambda_2^{-1}}$.

The detail proof of Theorem 1 can be found in Appendix.

In fact, combining the estimates of regularization error and space error, we can derive the learning rate of clustering algorithm.

4 Experiments

In this paper, we use three social networks to verify the regularized spectral algorithm: Zachary karate club, Lusseau's network of bottlenose dolphins and the online social network data of wiki-vote network (refer to, http://snap.stanford.edu/ncp/). The former two networks are small networks consisting of dozens of nodes and edges. The standard and regularized spectral clustering algorithm are compared by the results of these two social networks. The wiki-vote network consists of thousands of nodes and edges. The experiments are carried out on Linux with 4GB main memory and 4*Intel(R) Xeon(R) E5420@2.50GHZ cpu. Matlab R2009a was selected as our environment. Partial time consuming functions were replaced with **C** and **C++**.

4.1 Zachary karate club

In this subsection, we consider the well-known karate club friendship network studied by Zachary who observed 34 members of a karate club over a period of two years. During the course of the study, a disagreement appeared between the administrator of the club and the club's instructor, which ultimately resulted in the instructor's leaving and starting a new club, absorbing about a half of the original club's members with him. Zachary constructed a network of friendships between members of the club, using a variety of measures to estimate the strength of ties between individuals, which has became a commonly used workbench for community-finding algorithm testing. In our work, the whole network instead of the partial nodes of the network is chosen when testing our algorithms.

Figure 1 shows the eigenvalues of the Zachary club. The six diagrams in Fig. 1 are corresponding to the simple weight and the Gaussian weight laplacian matrix. We adopted the simple weight to calculate the eigenvalues of the unnormalized and normalized laplacian matrix. The results can be found in the top three figures. The bottom three figures shows the eigenvalues of the Gaussian weight eigenvalues version. From the figure, we can infer that the eigenvalues of the unnormalized laplacian matrix is much larger than the normalized laplacian matrix, and the difference between the sym and rw laplacian matrix is subtle. Figure 1 shows that the smallest eigenvalues of laplacian matrix is 0. Because of the connectivity of the Zachary karate club, the multiplicity of the zeros in the eigenvalues is 1 which means there is only one component in Zachary karate club. Similar results can be found in dolphin network.

The social graph of Zachary club was divided by the standard and the regularized spectral clustering algorithm both in the experiments. The results are shown in Tables 1 and 2. Table 1 is the modularity of the spectral clustering algorithm with simple similarity function. Apparently, the modularity drops fast when the community number increases.

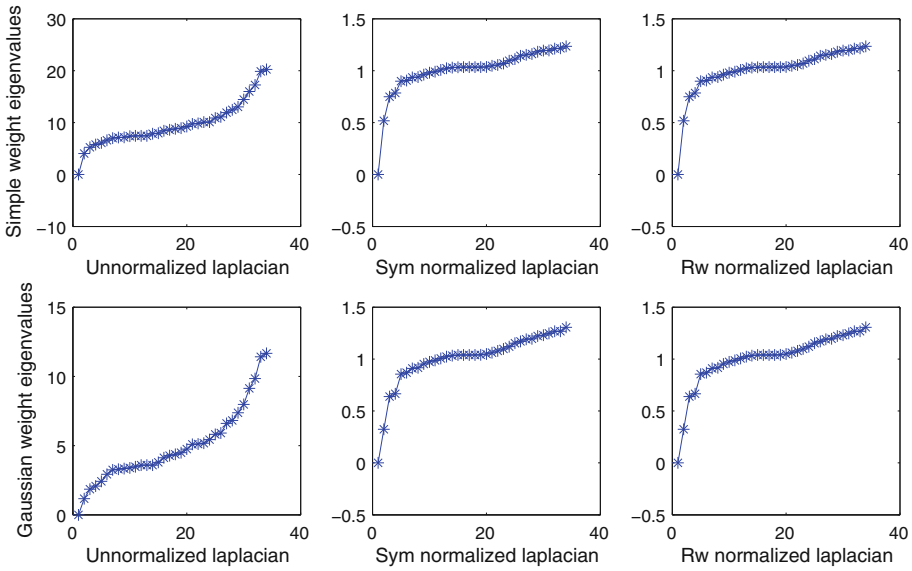

Fig. 1 The eigenvalues of the Zachary karate club. Note that the *top three figures* represent the results of the simple weight laplacian matrix, and the *bottom three figures* represent the results of the Gaussian weight laplacian matrix

Table 1 The modularity quantities of different methods for Zachary club with simple weight

Zachary	Number of communities				
Simple weight	1	2	3	4	5
Unnormalized laplacian	1	0.7858	0.5546	0.1624	Negative
Sym normalized laplacian	1	0.7858	0.61275	0.1986	Negative
Rw normalized laplacian	1	0.7858	0.64109	0.52859	Negative

Table 2 The modularity quantities of different methods for Zachary club with Gaussian weight ($\sigma = 1$)

Zachary	Number of communities				
Gaussian weight	1	2	3	4	5
Unnormalized laplacian	1	0.7858	0.28065	0.24018	Negative
Sym normalized laplacian	1	0.84213	0.63603	Negative	Negative
Rw normalized laplacian	1	0.7858	0.55878	0.53112	Negative
Regularized clustering	1	0.89172	0.57361	0.41783	Negative

The modularity of regularized spectral clustering algorithm is listed in row 4 ($\sigma = 0.5$, $\gamma = 0.01$)

The maximum modularity is 1 when there is only one group in the network which means all edges are within the group and no edges are between the groups. The modularity quantities with the Gaussian weight of standard spectral clustering algorithm can be found in the first three rows in Table 2. Figure 3 shows the best split of Zachary karate club. In this figure, Group 1 and group 2 owns 16 nodes and 19 nodes respectively. The modularity is 0.84213 with Gaussian weight and sym laplacian matrix. In the experiment, the σ parameter is set empirically with 1 which can clearly distinct the weight from different distance. The final results show that the Gaussian weight function is more effective than the simple weight function. The results of regularized clustering algorithm are listed in row 4 of Table 2. The Gaussian kernel function is used to simplify the comparison. We set $\sigma = 0.5$, $\gamma = 0.01$ in the experiment. The result shows that the clustering effect of the regularized clustering algorithm is similar to the standard clustering algorithm which indicates the reliability of the regularized spectral clustering algorithm.

4.2 Dolphin network

In this subsection, our algorithms are applied to the famous dolphin network. The network is compiled by Lusseau (Boisseau et al. 2003) from seven years of field studies of the dolphins, with ties between dolphin pairs being established by observation of statistically significant frequent association. There are 62 dolphins in Lusseau's dolphin network which is still tested by the standard and regularized spectral clustering algorithm.

Figure 3 is the eigenvalues of simple and Gaussian weight function of dolphin networks. The distribution of the eigenvalues is similar to the Zachary club social network. Table 3 and Table 4 show the modularity of the Gaussian and simple weight when using spectral clustering algorithm. In this experiment, the parameter σ is set to 1.5 based on the scale of the dolphin network. The clustering results show that the Gaussian similarity is more effective than the simple similarity. The modularity with normalized laplacian is larger than that with the unnormalized laplacian when the number of communities is small. The modularity

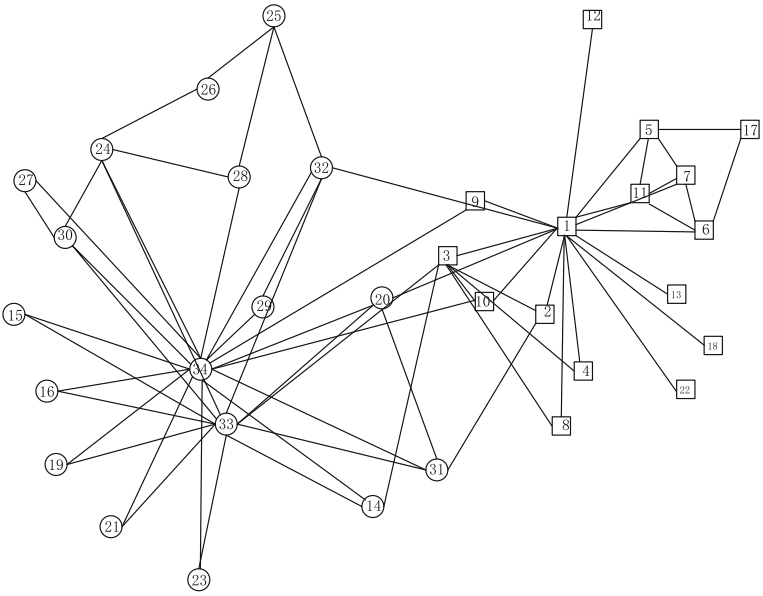

Fig. 2 The best split of Zachary karate club. Note that the *circle* and *square* nodes represent the two groups of the Zachary karate club

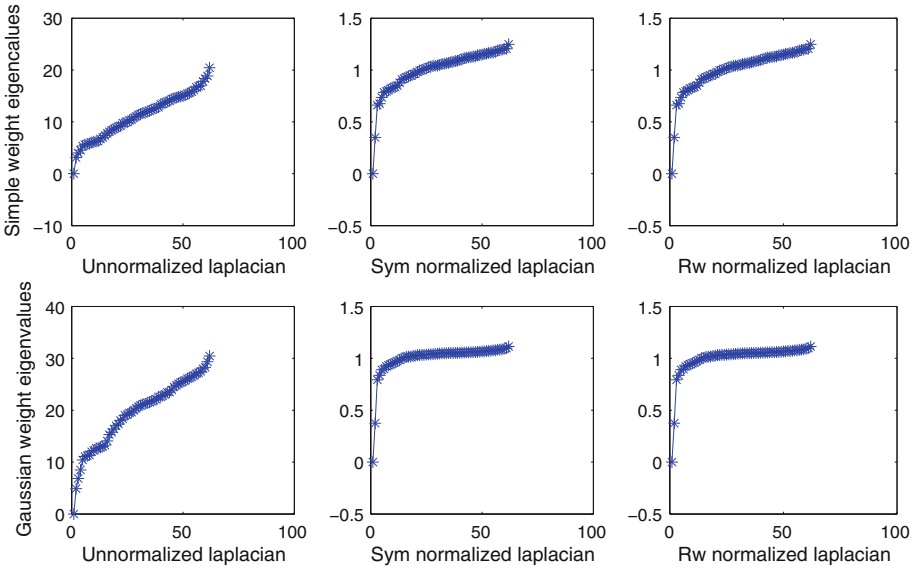

Fig. 3 The eigenvalues of the dolphin network. Note that the *top three figures* represent the results of the simple weight laplacian matrix, and the *bottom three figures* represent the results of the Gaussian weight laplacian matrix

decreases sharply with the normalized laplacian matrix. It is easy to infer that the dolphin network is ideal for splitting into two groups. Therefore, the decrease of the modularity cannot expose the actual results. Figure 4 shows the best splitting of dolphin network into two groups. There are 21 nodes in group 1 and 41 nodes in group 2.

Table 3 The modularity quantities of different methods for dolphin network with simple weight

Dolphin	Number of communities				
Simple weight	1	2	3	4	5
Unnormalized laplacian	1	0.94244	0.93027	0.8356	0.7636
Sym normalized laplacian	1	0.952	0.52983	0.386	0.11462
Rw normalized laplacian	1	0.952	0.65196	0.3332	Negative

Table 4 The modularity quantities of different methods for dolphin network with Gaussian weight ($\sigma = 1.5$)

Dolphin	Number of communities				
Gaussian weight	1	2	3	4	5
Unnormalized laplacian	1	0.94244	0.93027	0.89583	0.83467
Sym normalized laplacian	1	0.952	0.67004	0.34671	0.0037
Rw normalized laplacian	1	0.952	0.72707	0.11129	Negative
Regularized clustering	1	0.98583	0.49251	Negative	Negative

We list the modularity of regularized spectral clustering algorithm in row 4 ($\sigma = 0.7$, $\gamma = 0.04$)

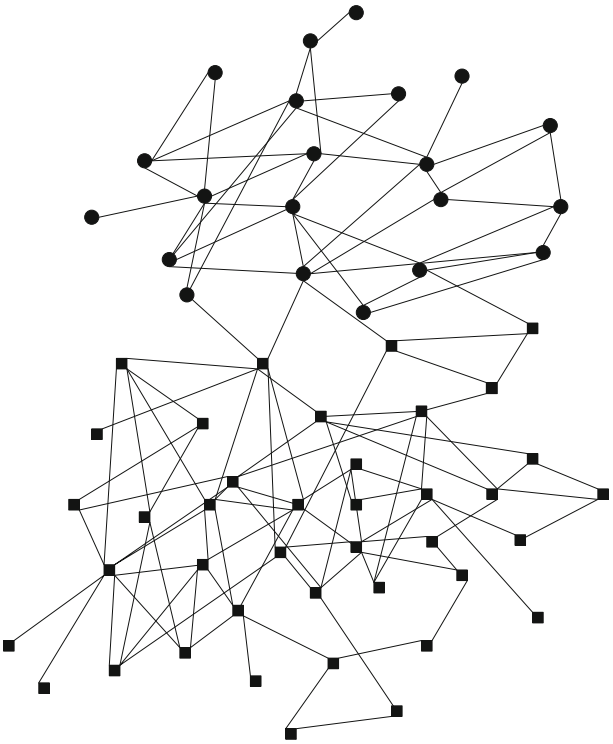

Fig. 4 The best split of dolphin network. Note that the *circle* and *square* nodes represent the two groups of the dolphin network which was split by Gaussian weight sym normalized laplacian method

Similar to the Zachry club, the regularized clustering algorithm is only tested with Gaussian weight and the unnormalized laplacian matrix methods. The modularity quantity is shown in the last row of Table 4, where $\sigma = 0.7$, $\gamma = 0.04$. The difference between the two algorithms is subtle.

4.3 Wiki-vote network

We apply our algorithm in wiki-vote network in this subsection. Wikipedia is a free encyclopedia written collaboratively by volunteers around the world. A small part of Wikipedia contributors are administrators, who are the users with access to additional technical features that aid in maintenance. For a user, in order to become an administrator, a request for adminship is issued, and the Wikipedia community via a public discussion or a vote decides who can access to adminship. There are 103,663 total votes and 7,066 users participating in the elections (either casting a vote or being voted on).

The network contains all the users and discussions from the inception of Wikipedia till January 2008, which includes 8,297 nodes and 103,663 edges in the wiki-vote social network. This network is an unweighted but directed one. It is hard to apply our algorithm directly in this network. Therefore, we transform the network into an undirected social graph simply neglecting the direction of the vote relationship. In this way, we get the undirected wiki-vote social network with 8,297 nodes and 201,524 edges that can be used in our algorithm. The regularized clustering algorithm is only adopted due to the scale of this network. One can find that the difference of clustering results between normalized and unnormalized laplacian is subtle from the first two experiments. Therefore, the experiment only adopts the unnormalized laplacian matrix and the Gaussian similarity to alleviate the computational overhead.

In the experiment, we considered sample sets of sizes $\in \{500, 1000, 2000\}$. The clustering results can be found in Fig. 5. The average elapsed time for one experiment is about 22 min

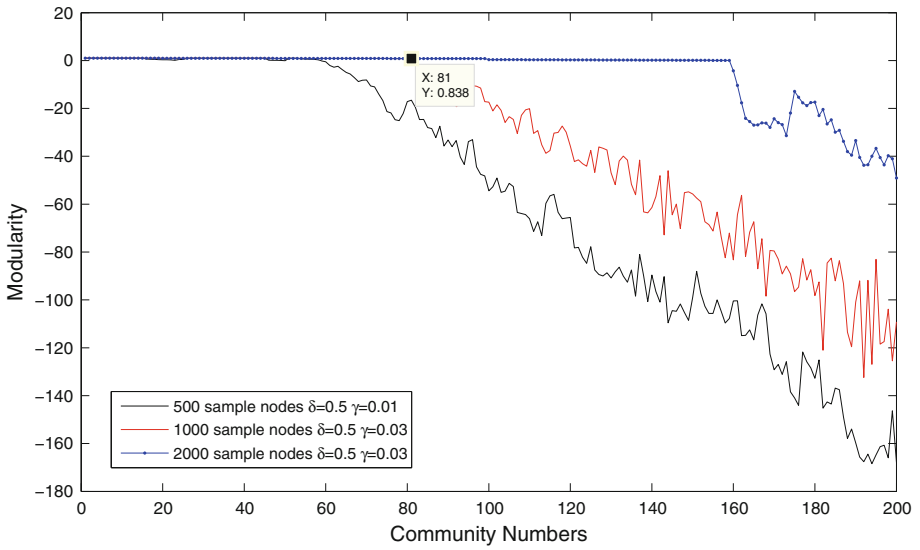

Fig. 5 The *three modularity curves* of the regularized spectral clustering algorithm when applying in wiki-vote social network

including step 1 to step 7 in Algorithm 2, and larger sample nodes experiment took relative more time than smaller sample nodes one. 58 communities (modularity ≥ 0) are detected in wiki-vote social network corresponding to 500 sample nodes and 81 communities for 1,000 sample nodes and 159 communities for 2,000 sample nodes. In Fig. 5, the clustering result with 2,000 sample nodes is apparently better than the former two results with relatively fewer sample nodes.

5 Conclusion

We propose a new algorithm aiming at detecting community structure in complex networks in an efficient and systematic way. The algorithm has a considerable speed advantage over previous algorithms by choosing the sample nodes to reduce the network scale and the computational complexity. This allows us to study much larger systems which were too difficult to process previously.

The algorithm has been applied to several social networks including a network of wiki-vote among 8,297 voters. The resulting community structures correspond closely to the standard clustering algorithm with relative smaller computational burden and complexity.

The method can not only extend the community structure analysis to some of the very large networks, but also provides a useful tool for visualizing and understanding the structure of these networks. We hope that this algorithm will be employed successfully in the search and study of communities in social networks, and will help to uncover new interesting properties in this area.

However, the social networks adopted in our paper are relative smaller that consist of thousands nodes in the networks. The bottleneck of our algorithm is the computation capability of the computer. In the future work, we plan to apply our algorithm in distributed computing environments such as MapReduce. We believe that better performance can be achieved in the extremely large online social networks like LiveJounal and Delicious social networks with our algorithm.

Additionally, we use the shortest distance of the social graph to construct the similarity graph. It is highly time consuming that must be improved. There are some other ways to construct the similarity graph that consider the neighboring and the ingoing edge weights like ℓ^1-graph (Cheng et al. 2010). The future work will also consider to adopt different graph construction algorithms to improve the performance of the proposed algorithm.

Acknowledgments This work is supported by National Natural Science Foundation of China under grants 61173170 and 60873225 and 11001092, National High Technology Research and Development Program of China under grant 2007AA01Z403, Innovation Fund of Huazhong University of Science and Technology under grants 2011TS135 and 2010MS068, and the Fundamental Research Funds for the Central Universities under grant 2011PY130.

Appendix Proof of Theorem

We recall some basic facts about U-statics (Peña and Giné 1999; Lugesi et al. 2008). Consider the i.i.d random variables $\{x_i\}_{i=1}^n$ and denote by

$$U_n = \frac{1}{n(n-1)} \sum_{i \neq j} q(x_i, x_j).$$

a U-statistic of order 2, where $q : \mathscr{X} \times \mathscr{X} \to R$ is a symmetric real-valued function. In Cucker and Zhou (2007), a Bemstain-type inequality for U-statistics is established.

$$P\{|U_n - EU_n| > t\} \le 2\exp\Big\{-\frac{\lfloor n/2\rfloor t^2}{2\sigma^2 + 2t/3}\Big\}, \quad \forall t > 0.$$

where $\sigma^2 = Var(q(x, x'))$.

Let $q(x, x') = (f(x) - f(x'))^2 K(x, x')$. We have

$$\sigma^2 = Var(q(x, x')) \le E(q(x, x'))^2 = E[(f(x) - f(x'))^2 K(x, x')]^2 \le 16\kappa^2 \|f\|_\infty^4.$$

Thus, we can derive the following concentration inequality.

Lemma 1 *For all* $\varepsilon > 0$.

$$P\{|\mathscr{E}_{\mathbf{x}}(f) - \mathscr{E}(f)| > \varepsilon\} \le 2\exp\Big\{-\frac{\lfloor n/2\rfloor \varepsilon^2}{32\kappa^2\|f\|_\infty^4}\Big\}.$$

We use the procedure in Hoeffding (1963) to deal with the $\mathscr{E}(f_{\mathbf{x},\gamma}) - \mathscr{E}_{\mathbf{x}}(f_{\mathbf{x},\gamma})$.

Lemma 2 *If* $f_1, f_2 \in B_R$, *we have*

$$P\{|\mathscr{E}(f_1) - \mathscr{E}_{\mathbf{x}}(f_1) - (\mathscr{E}(f_2) - \mathscr{E}_{\mathbf{x}}(f_2))| \le 8\kappa^2 R\|f_1 - f_2\|_\infty.$$

Proof Note that

$$\begin{aligned}
&|\ \mathscr{E}(f_1) - \mathscr{E}_{\mathbf{x}}(f_1) - (\mathscr{E}(f_2) - \mathscr{E}_{\mathbf{x}}(f_2))\ | \\
&= \Big|\int_{\mathscr{X}}\int_{\mathscr{X}} [(f_1(x) - f_1(x'))^2 - (f_2(x) - f_2(x'))^2]K(x, x')d\rho(x)d\rho(x')\Big| \\
&\le \Big|\int_{\mathscr{X}}\int_{\mathscr{X}} (f_1(x) - f_1(x') + f_2(x) - f_2(x'))(f_1(x) - f_2(x) + f_2(x') \\
&\quad - f_1(x'))d\rho(x)d\rho(x')\Big| \\
&\le \|f\|_\infty\|f_1 - f_2\|_\infty.
\end{aligned}$$

Then the desired result follows from the inequality $\|f\|_\infty \le \kappa\|f\|_K$. □

Now we introduce the bound of $f_{\mathbf{x},\gamma}$ in Cao and Chen (2011).

Lemma 3 *Assume* $\lambda_2 > 0$, *Then for* $0 < \delta < 1$, *we have for* $n > M_\delta$, *with confidence at least* $1 - 2\delta$,

$$\|f_{\mathbf{x},\gamma}\|_K \le \sqrt{\gamma^{-1} + 4l^{-1}\lambda_2^{-1}}.$$

Proof of Theorem 1 Let $m = \mathscr{N}(B_R, \frac{\varepsilon}{16\kappa^2 R})$ and consider $\{f_i\}_{i=1}^m$ such that the disks O_i centered at f_i and with radius $\frac{\varepsilon}{16\kappa^2 R}$ cover B_R.

By Lemma 2, for all $x \in \mathscr{X}^n$ and $f \in O_i$,

$$|\ \mathscr{E}(f) - \mathscr{E}_{\mathbf{x}}(f) - (\mathscr{E}(f_i) - \mathscr{E}_{\mathbf{x}}(f_i))\ | \le 8\kappa^2 R\|f - f_i\|_\infty \le \varepsilon/2.$$

Thus, for all $x \in \mathscr{X}^n$ and $f \in O_i, i = 1 \ldots m$,

$$\sup_{f \in O_i} \mid \mathscr{E}_{\mathbf{x}}(f) - \mathscr{E}(f) \mid \geq \varepsilon \Rightarrow \mid \mathscr{E}_{\mathbf{x}}(f_i) - \mathscr{E}(f_i) \mid \geq \varepsilon.$$

By Lemma 1, we have

$$P\left\{\sup_{f \in O_i} \mid \mathscr{E}_{\mathbf{x}}(f) - \mathscr{E}(f) \mid \geq \varepsilon\right\} \leq P\{\mid \mathscr{E}_{\mathbf{x}}(f_i) - \mathscr{E}(f_i) \mid \geq \varepsilon/2\}$$
$$\leq 2\exp\left\{-\frac{\lfloor n/2 \rfloor \varepsilon^2}{128\kappa^6 R^4 + 4\varepsilon/3}\right\}.$$

Moreover,

$$P\left\{\sup_{f \in B_R} \mid \mathscr{E}_{\mathbf{x}}(f) - \mathscr{E}(f) \mid \geq \varepsilon\right\} \leq \sum_{i=1}^{m} P\left\{\sup_{f \in O_i} \mid \mathscr{E}_{\mathbf{x}}(f) - \mathscr{E}(f) \mid \geq \varepsilon\right\}$$
$$\leq 2\mathscr{N}\left(B_R, \frac{\varepsilon}{16\kappa^2 R}\right)\exp\left\{-\frac{\lfloor n/2 \rfloor \lfloor \varepsilon^2}{128\kappa^6 R^4 + 4\varepsilon/3}\right\}.$$

By Lemma 3, we know $f_{\mathbf{x},\gamma} \in B_R$ for $R = \sqrt{\gamma^{-1} + 4l^{-1}\lambda_2^{-1}}$.

For an $\frac{1}{R}$-centering of B_1 yields and 1-centering of B_R and vice versa, we complete the proof. □

References

Aggyriou A, Herbster M, Pontil M (1950) Combing graph laplacians for semi-supervised learning. Trans Am Math Soc 68:337–404

Barthélemy M, Fortunato S (2007) Resolution limit in community detection. Proc Natl Acad Sci 104:36–41

Belkin M, Niyogi P, Sindhwani V (2006) Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. J Mach Learn Res 7:2399–2434

Belkin M, Matveeva I, Niyogi P (2004) Regularization and semi-supervised learning on large graphs. In: Shawe-Taylor J, Singer Y (eds). Proceedings of the 17th annual conference on learning theory, pp. 624–638

Boisseau OJ, Dawson SM, Haase P et al (2003) The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. Behav Ecol Sociobiol 54:396–405

Cao Y, Chen DR (2011) Consistency of regularized spectral clustering. Appl Comput Harmon Anal 30:319–336

Chen H, Li LQ, Peng JT (2009) Error bounds of muti-graph regularized semi-supervised classfication. Inf Sci 179:1960–1969

Chen LN, Li ZP, Zhang SH et al (2008) Quantitative function for community detection. Phys Rev E 77:036109

Cheng B, Huang TS, Yang JC, Yan SC (2010) Learning With ℓ^1-graph for image analysis. IEEE Trans Image Process 19:858–866

Chung FRK (1997) Spectral graph theory. AMS Press, Providence, R.I

Cucker F, Zhou DX (2007) Learning theory: an approximation theory viewpoint. Cambridge University Press, Cambridge

De la Peña KVH, Giné E (1999) Decoupling: from dependence to independence. Springer, New York

Donetti L, Munoz MA (2004) Detecting network communities: a new systematic and efficient algorithm. J Stat Mech: Theor Exp, P10012

Fortunato S, Latora V, Marchiori M (2004) Method to find community structures based on information centrality. Phys Rev E 70:056104

Fortunato S (2010) Community detection in graphs. Phys Rep 486:75–174

Hoeffding W (1963) Probability inequalities for sums of bounded random variables. J Am Stat Assoc 58:13–30

Lee CH, Zaïane OR, Park HH et al (2008) Clustering high dimensional data: a graph-based relaxed optimization approach. Inf Sci 178:4501–4511

Lugesi G, Stéphan S, Voyatis N (2008) Ranking and empirical minimization of U-statistics. Ann Stat 36: 844–874
Luxburg UV (2006) A tutorial on spectral clustering. Stat Comput 17(4):395–416
Newman MEJ, Girvan M (2004) Finding and evaluating community structure in networks. Phys Rev E 69:026113
Newman MEJ (2004) Fast algorithm for detecting community structure in networks. Phys Rev E 69:066133
Watts DJ, Strogatz SH (1998) Collective dynamics of small-world networks. Nature 393:440–442
Xing EP, Ng AY, Jordan MI (2003) Distance metric learning, with application to clustering with side-information. Adv Neural Inf Process Syst 15
Zachary WW (1977) An information flow model for conflict and fission in small groups. J Anthropol Res 33:452–473