Common community structure in time-varying networks

Shihua Zhang,* Junfei Zhao, and Xiang-Sun Zhang
National Center for Mathematics and Interdisciplinary Sciences, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China

(Received 13 November 2011; revised manuscript received 18 January 2012; published 10 May 2012)

In this report we introduce the concept of common community structure in time-varying networks. We propose a novel optimization algorithm to rapidly detect common community structure in such networks. Both theoretical and numerical results show that the proposed method not only can resolve detailed common communities, but also can effectively identify the dynamical phenomena in time-varying networks.

DOI: 10.1103/PhysRevE.85.056110 PACS number(s): 89.75.Fb, 89.65.−s, 89.75.Kd

Networks are powerful abstractions of relational data and have become very popular tools in many fields, including sociology, biology, and physics [1]. The characteristic of community structure in networks; i.e., networks are naturally divided into modules or communities, has attracted enormous attention in the past decade. The community analysis can provide insights into the structure and dynamic formation of networks. Many methods for community detection in one network have been developed and studied, including the fuzzy community structure identification problem [2] and the more challenging community detection problem in directed networks [3] (see Ref. [4] for recent comprehensive reviews).

However, previous studies have concentrated on uncovering community structure in a static network, which represents only a summarized picture of all possible relations. One typical example is a protein interaction network in biology, which represents all proteins of an organism and all interactions regardless of the conditions and time under which interactions may take place [5]. In reality most of relationships modeled by networks evolve with time or conditions [6].

Several recent studies have touched on the analysis of dynamic networks, including analyzing changes of global properties, detecting anomalous changes, mining dynamic frequent subnets, and discovering similar evolving regions in evolving networks [7] and even the dynamic communities by combining the information of communities in each network using traditional community detection methods [8]. However, the community structure in two or more slices of a series of time-varying networks has been addressed by only a few studies until recently [9].

In this report we propose the concept of common community structure in two or more networks of a series of time-varying networks. The basic assumption is that essential and common community structure may exist in two or more networks, and local dynamic changes may happen. This is very realistic in time-varying networks of many robust systems.

Suppose that we are given the structure of two or more networks of the same vertices, then we aim to determine whether there exists any common community structure or, for example, similar groups or communities in these networks. More specifically, some communities appearing in both or multiple networks can be combined to form the common community structure in them. As part of this goal, we further attempt to uncover the dynamic characteristics of some vertices. Mathematically, the common community structure and dynamical characteristic are stored in matrices which can be determined by an efficient optimization procedure.

Let us focus initially on the problem in two networks that will be very useful in analyzing time-varying networks. To formulate the problem easily, we consider the common notation of clustering or community structure detection problems. The objective of classical community detection in networks is to partition the vertex set V of the graph $G(V,E)$ with $|V| = N$ into K distinct subsets in a way that densely connected groups of vertices are placed in the same community. In this case a convenient representation of a given partition is the partition matrix $U = [u_{ik}]$ (or $[u_i]$, u_i is a membership vector) with size of $N \times K$ [10]. And $u_{ik} = 1$ if and only if vertex i belongs to the kth subset in the partition; otherwise it is zero. From the definition of the partition, it clearly follows that $\sum_{k=1}^{K} u_{ik} = 1$ for all i. The generalization of the hard partition follows by allowing u_{ik} to attain any real value from the interval $[0,1]$, and the corresponding matrix is also called membership matrix.

In the following, we adopt the popular membership matrix representation to formulate the problem. Nepusz *et al.* [10] have suggested that an edge between vertex v_1 and v_2 implies the similarity of v_1 and v_2, and likewise, the absence of an edge implies dissimilarity, i.e, $a_{ij} \simeq u_i u_j^T$ or $A \simeq UU^T$, where $A = (a_{ij})$ is the adjacency matrix of a network. At the same time, the same vertices in two networks should have similar membership vectors. These considerations can be formulated as

$$\min \sum_{g=1}^{2} \|A_g - H_g H_g^T\|_F^2 + \lambda_1 \sum_{g=1}^{2} \|H_g - H\|_1 + \lambda_2 \|H\|_1 \quad (1)$$

$$\text{subject to} \quad \begin{cases} \sum_{k=1}^{K} (H_g)_{ik} = 1; \quad (H_g)_{ik}, \quad H_{ik} \geqslant 0; \\ g = 1,2, \quad i = 1,\ldots,N, \quad k = 1,\ldots,K, \end{cases}$$

where A_g is the adjacency matrix of network $G(V,E_g)$, H_g is the membership matrix of network $G(V,E_g)$, and $\|\cdot\|_F$ and $\|\cdot\|_1$ are the entrywise matrix norm ($\|\cdot\|_F$ is known as the Frobenius norm). We note that H is the virtual membership matrix, which reflects the membership of nodes determined by the topological information of two networks. Based on the above formulation, we aim to determine an optimal H to capture the common community structure in

*Corresponding author: zsh@amss.ac.cn

both networks. Specifically, the value of the element H_{ik} in H is non-negative, and it represents the intensity of vertex i in two or more networks of a time-varying system belonging to one of their common community k. To solve the problem easily, we remove the constraints $\sum_{k=1}^{K}(H_g)_{ik} = 1$ ($g = 1,2$; $i = 1,\ldots,N$). Then the magnitude of $(H_g)_{ik}$ reflects the intensity of vertex i belonging to community k in the network $G(V,E_g)$. This formulation allows us to map the communities of two networks as well as their common communities.

Here we assume that the two networks have the same number N of nodes and the same number K of communities. But this model can be potentially applied to the cases where one of the two networks loses some nodes which enable the two networks do not have exactly the same number of nodes and communities. For example, one of these two networks is of size N_1 with $N_1 < N$ and has only $K-1$ communities, then we can still store them in an $N \times N$ adjacency matrix and assume it has K communities. To solve the model, we can add small corresponding values to the denominator of related terms in the updating rule, and the membership matrix has K columns with small membership values in one of them.

The nonconvexity and the nonsmoothness of the objective function of Eq. (1) make it a challenging mathematical programming problem. To practically solve the problem Eq. (1), we employ a decomposition technique. We can easily find that, given the common communities matrix H, the technique leads to two symmetrical non-negative factorization matrix (SNMF) problems [11] coupled with a penalty term as follows

$$\min \sum_{g=1}^{2} \left\| A_g - H_g H_g^T \right\|_F^2 + \lambda_1 \sum_{g=1}^{2} \| H_g - H \|_1. \qquad (2)$$

Fortunately, it can be divided into two independent subproblems, which can be solved in a symmetric NMF manner with the following updating rule:

$$(H_g)_{ik} \leftarrow (\widetilde{H}_g)_{ik}\left[1 - \beta + \beta \frac{(A_g \widetilde{H}_g)_{ik}}{(\widetilde{H}_g \widetilde{H}_g{}^T \widetilde{H}_g)_{ik}}\right], \qquad (3)$$

where $\widetilde{H}_g = H_g + \Delta(H_g - H)$, and $0 < \beta \leqslant 1$ ($\beta = 1/2$ has been used empirically). The columns of H_1 and H_2 determine the community structure in two networks, respectively. According to Eq. (3), we update H_1 and H_2 in each iterative step separately, and their orders are independent. Then the columns of H_1 and H_2 may correspond to unrelated communities. To avoid the inconsistency, we reorder their columns by maximizing correlations of corresponding columns to facilitate the optimization procedure. We should note that this strategy will not affect the optimal property, since each entry of $H_g H_g^T$ is independent with the column order of H_g ($g = 1,2$).

While given the community matrix H_g of each network, the model (1) leads to the following problem:

$$\min \lambda_1 \sum_{g=1}^{2} \| H_g - H \|_1 + \lambda_2 \| H \|_1. \qquad (4)$$

This formulation with positive combination of L_1 norm of variables can be transformed into a large-scale linear programming problem through a well-known procedure. More interestingly, it can be solved efficiently by a further decomposition technique [12]. We should note, owing to the L_1 norm, that generally the optimal solution has an excellent property; i.e., there are as many zeros for $\| H_g - H \|_1$ and $\| H \|_1$ as possible. This point exactly serves the final goal, i.e., consistency and sparseness of the membership of each vertex.

Therefore, we have the following algorithm for discovering common communities in two undirected networks. We first set the parameters λ_1, λ_2, β, and K, initialize the membership matrices $H_1$and H_2, and set $H = \frac{H_1+H_2}{2}$. For the subproblem Eq. (2), we use the update rule Eq. (3) to update H_1 and H_2, respectively. Then using the new H_1 and H_2 we solve the subproblem Eq. (4) to obtain the new H, by subdividing it into $N \times K$ one-dimensional optimization subproblem. We iteratively solve the subproblem Eqs. (2) and (4) until H does not change too much (e.g., $\frac{\| H_{\text{new}} - H_{\text{old}} \|_F^2}{\| H_{\text{old}} \|_F^2} < 10^{-5}$, where H_{new} and H_{old} are the H in current step and last step, respectively). The final H, H_1, and H_2 store the common communities and dynamical information. The H (H_1 and H_2) can be considered as a fuzzy partition of the network(s) directly [13]. It can also be employed to determine a hard partition by assigning a node into a single community according to the maximum value in each row of H (H_1 and H_2) [14].

The time complexity of the proposed algorithm is $O(TKN^2)$, where T is the number of iterations. The efficiency of the method can also be seen in its application to networks with size of 10 000 (see Appendix). Note that the method can be applied onto a single network by minimizing the criterion: $\| A_g - H_g H_g^T \|_F^2$, and it shows competitive performance with two popular algorithms (see Appendix).

The formulation for two networks can be easily extended to more than two networks as follows:

$$\min \sum_{g=1}^{G} \left\| A_g - H_g H_g^T \right\|_F^2 + \lambda_1 \sum_{g=1}^{G} \| H_g - H \|_1 + \lambda_2 \| H \|_1, \qquad (5)$$

where all the H_g and H are non-negative matrices. The algorithm can also be easily extended.

The key issue in community detection is the proper choice of K. Here we employ the stochastic nature of the proposed algorithm to achieve this. We should note that a similar strategy has been used to determine the number of clusters in gene expression studies [14]. The differences and similarities of multiple realizations are employed to evaluate the robustness of a partition of a specific K. Specifically, for each run, the vertices assignment can be defined by a connectivity matrix C of size $N \times N$, with entry $c_{ij} = 1$ if vertices i and j belong to the same communities, and $c_{ij} = 0$ if they belong to different clusters. We can then compute the consensus matrix, $\overline{C}$, defined as the average connectivity matrix over many runs. The entries of $\overline{C}$ range from 0 to 1 and reflect the probability that vertices i and j belong to one community.

We adopt the entropy as a measure of the stability of the common community structure. We assume that the c_{ij} are independent of each other and define the average common community entropy score as

$$E = -\frac{2}{N(N-1)} \sum_{(i;j)} [c_{ij} \log_2 c_{ij} + (1 - c_{ij}) \log_2 (1 - c_{ij})],$$

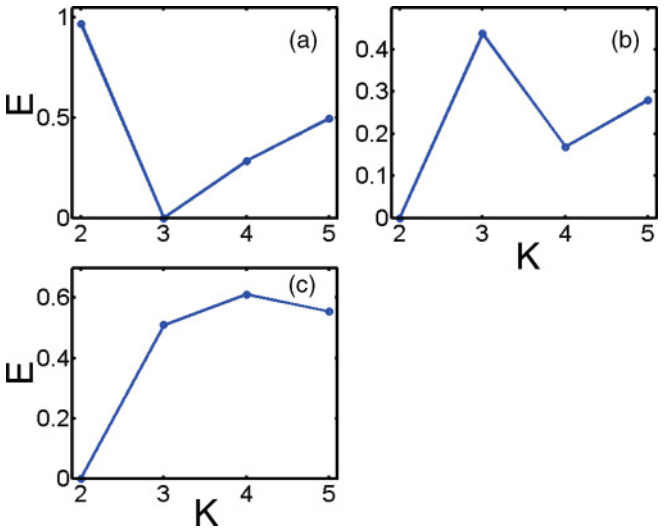

FIG. 1. (Color online) The common community entropy for each testing network system in the following analysis: (a) the simulated networks, (b) the karate club networks, (c) the US Senate networks.

where the sum is taken over all edges and m is the total number of edges in the network. If the network is totally unstable (i.e., in the most extreme case $c_{ij} = 0.5$ for all pairs), $E = 1$, while if the edges are perfectly stable under noise ($c_{ij} = 0$ or 1), $E = 0$. We have demonstrated that the E score can help to select the number of communities in the time-varying networks (Fig. 1). For example, the E score for the simulated networks corresponds to a very small value for $K = 3$, which indicates that the system has three distinct communities.

We should note that the parameters λ_1, λ_2, and β can also be evaluated with the E score by running the method with many trials. Parameter selection is a challenging problem for many problems now. It is also difficult to design an exact selection model for the proposed method, which heavily relies on the structure of networks. However, similar to the selection of K, we can compare the E score of different parameter settings by running the method with many trials. As the new method can be run efficiently for relatively large-scale networks, together with current efficient computing hardware systems, we think that it is feasible to run the method many trials for parameter selection. Moreover, domain knowledge about real networks may provide valuable information for parameter selection.

The membership matrix H_g for each network represents the community structure of each network, and the features of H can be employed to describe the dynamic structure of these networks. For each run, we can define the following index S for vertex i as the ratio between the second maximal value and the maximal value of row i of H. The ratio is a positive value less than one. In reality there is no rigid threshold for significant S score due to the diversity of networks, but we can select top ones based on the popular Z score (i.e., $Z = \frac{S-\mu(S)}{\sigma(S)}$, where $\mu(S)$ is the mean of S and $\sigma(S)$ is the standard deviation of S). By removing the active dynamic vertices according to this index, we can define the stable common communities of these networks.

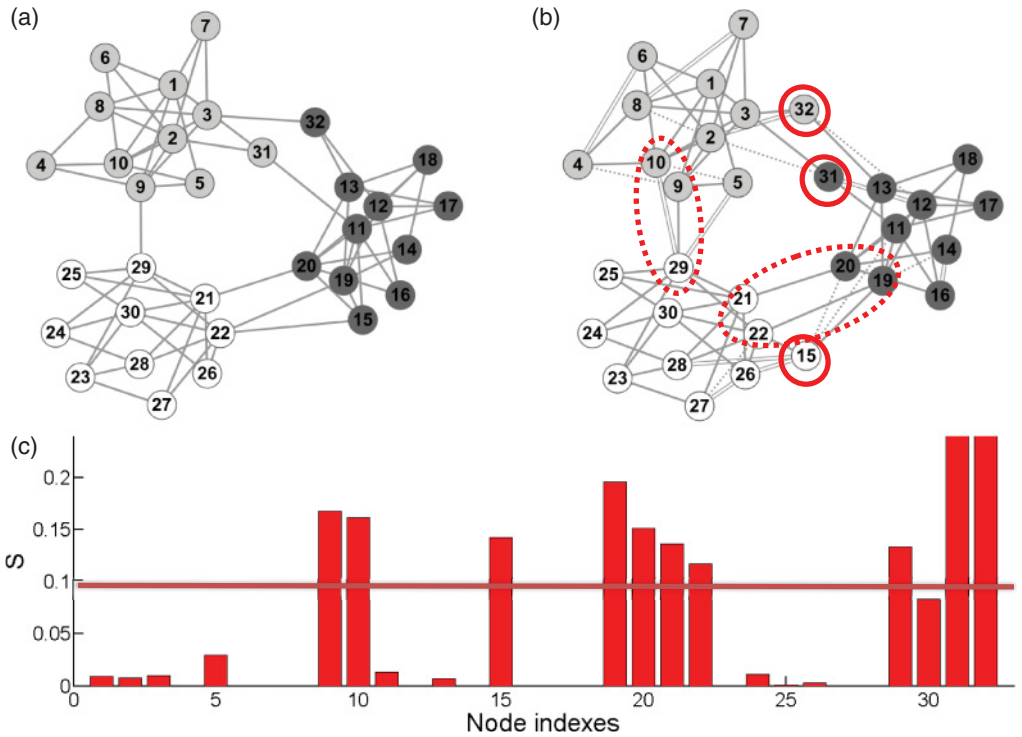

FIG. 2. (Color online) Illustration of a toy example to show the major idea. (a) The system under the first condition where the links are marked with solid lines. (b) The system under the second condition with links of some vertices changing, where dotted lines mean links exist in a previous condition, but disappear in current condition, while double lines mean new links. (c) The dynamic index shows the dynamic properties of vertices. Some vertices with high values affect the community structure. The horizontal line was drawn to indicate several distinct S values, whose corresponding nodes have been marked in (b). A similar line has been drawn in Fig. 3(c).

We first test the proposed method using a pair of simulated toy networks of a time-varying system under two time points with 16 links' difference [Figs. 2(a) and 2(b)]. In the system, there are three clear communities, however, in the two conditions, the links of some vertices have changed due to some perturbations. We aim to identify these communities, and uncover those link dynamics that can affect the community structure. The link dynamics exist in two types: link changes within a community and link changes between two communities. The dynamics happening within a community do not affect the community structure, while those between communities do. For example, the absence of links (15,11) and (15,20) and the emerging links (15,28) and (15,26) makes the vertex 15 move to another community. Our method can not only identify the community structure well, but also can accurately distinguish the link dynamics that affect the community structure [Fig. 2(c)].

We next apply our method to the karate club network and its variants with 12 links' difference compared with the original one. The original karate club network was constructed based on the observed social interactions between members of a karate club, in which a dispute arose and the club split into two clubs. We assumed there are changes upon the members' relationship as shown in Fig. 3(b). Our method can identify well the core communities which correspond to the two real subclubs [Figs. 3(a) and 3(b)]. At the same time, we can uncover the vertices whose link dynamics can affect the community structure. For example, the links of vertices 10 and vertices 20 have great difference, and the two vertices are located at the boundary of two communities. These two nodes have evolved into opposite communities, which can be reflected well by the index S [Fig. 3(c)].

We further apply our method to the set of time-varying networks consisting of 100 vertices (senators) and eight time points (i.e., eight time-varying networks) corresponding to three-month epochs starting on 1 January 2005 and ending on 31 December 2006. The network data were created using the method developed by Kolar *et al.* [15] based on the United States 109th Congress voting records and analyzed in Ho *et al.* [16]. An edge between two senators in such a network indicates that their votes were mostly similar during that particular epoch. We observed that two successional networks have relatively small changes. As an example, we show the networks ($t = 1$ and 5) and identify the common community among them [Figs. 4(a) and 4(b)]. Our method can well identify the two common communities which perfectly capture party affiliations: Republican senators are almost always in community 1, while Democratic senators are almost always in community 2. More interestingly, we can also identify the dynamic changing of some vertices which reflect the changes of political opinions of some senators [Fig. 4(c)]. For example, the votes of Democrat Nelson were unaligned with either Democrats or Republicans at $t = 1$, while his votes were gradually shifting toward Republicans, which can be found by the index S.

In this report, we investigate the common community structure in time-varying networks. Rather than treating each slice of a series of time-varying networks independently, we consider them simultaneously by defining a common community structure among them. We have proposed a new framework for recovering the common community structure and exploring the dynamic changes in these networks by solving an elaborate mathematical programming problem via existing decomposition techniques. We have applied the method to both real and simulated networks, demonstrating that it is able to recover known common community structure and reveal dynamic changes among them. The nondeterministic characteristic of the method allows it for the selection of

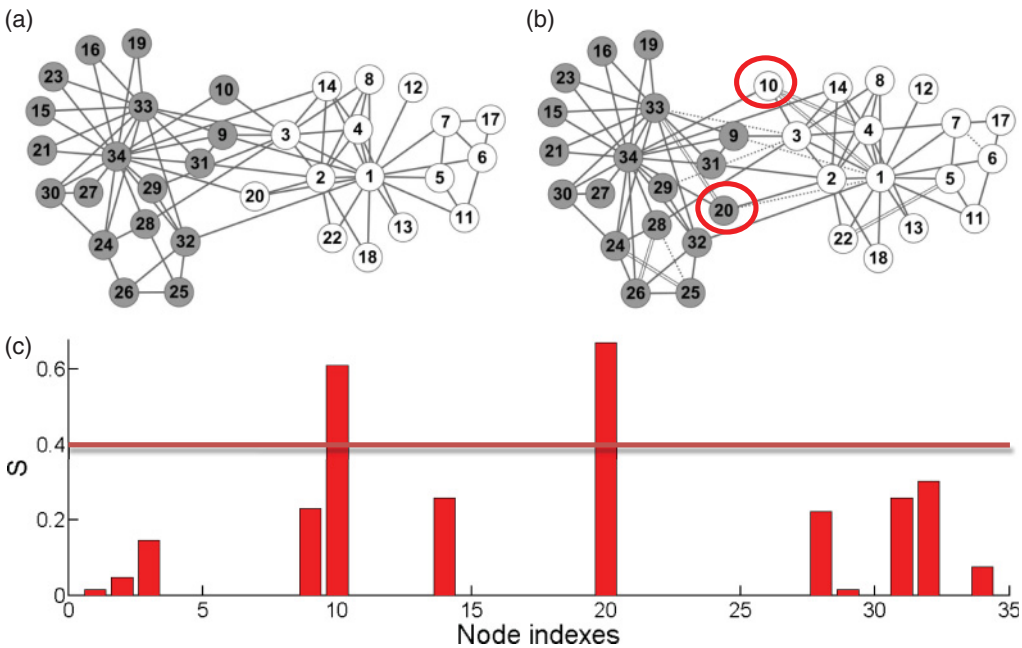

FIG. 3. (Color online) (a) The original karate club network. (b) The artificial evolving network with 12 links' difference compared with the network in (A). (c) The dynamic index shows the dynamic properties of vertices.

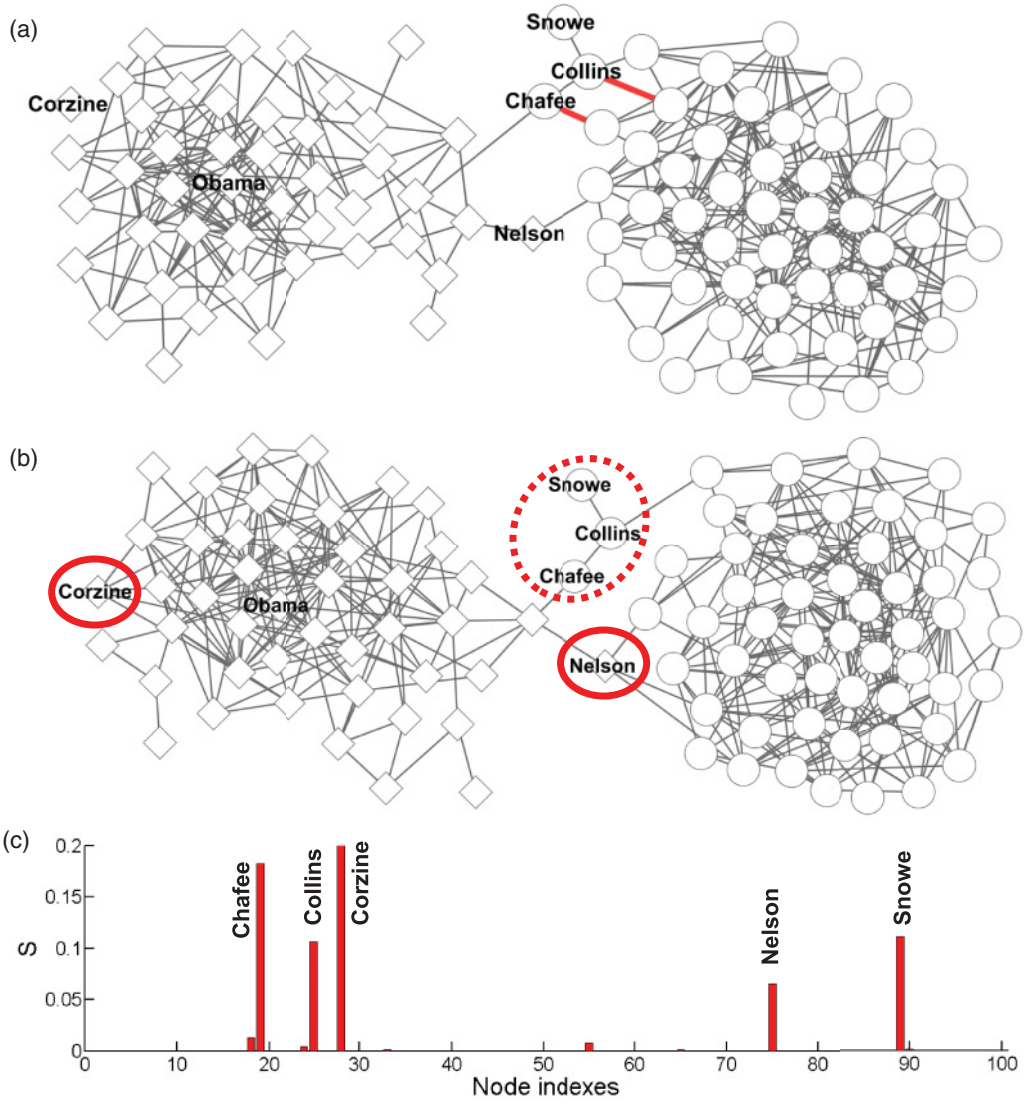

FIG. 4. (Color online) (a) The US Senate networks at different time points: (a) $t = 1$, (b) $t = 5$, and five vertices show distinct dynamic characteristics. (c) The dynamic indexes show the dynamic properties of vertices. Vertex shape show the two political paries: square means Democrat, and circle means Republican.

number of communities and quantification of the stability of the community structure. We should note that our framework can shed lights on the situation that dramatic changes appear in time-varying networks. Specifically, by applying our method on each network respectively, we can detect the community structure of the two networks. By calculating the consistency of the two community structure with a measure like normalized mutual information index, we can see how similar the community structure are in the two networks.

In summary, the main purpose of this report is to propose the new concept and theoretical framework to analyze the common community structure of multiple slices of a series of time-varying networks, which shed light on the network's dynamics and stability. We expect it to become a promising method for time-varying network analysis. We need to point out that the adjacency matrix A used in this framework can be replaced by some *similarity* matrix based on the connectivity like kernel matrix.

This work was partially supported by the National Natural Science Foundation of China, No. 11001256, 11131009, the Special Presidential Prize–Scientific Research Foundation of the CAS, the Special Foundation of President of AMSS at CAS for "Chen Jing-Run" Future Star Program, and the Foundation for Members of Youth Innovation Promotion Association, CAS (to S.Z.). The authors thank Prof. Eric P. Xing for providing the US Senate network data.

APPENDIX

We have applied the reduced formulation onto simulated networks with multiple trials. The networks have been simulated based on the principle suggested in Lancichinetti *et al.* [17]. We found that our method can obtain reasonable results for many different simulation settings assessed with normalized mutual information index (Fig. 5). We also compared it with other typical community methods, which

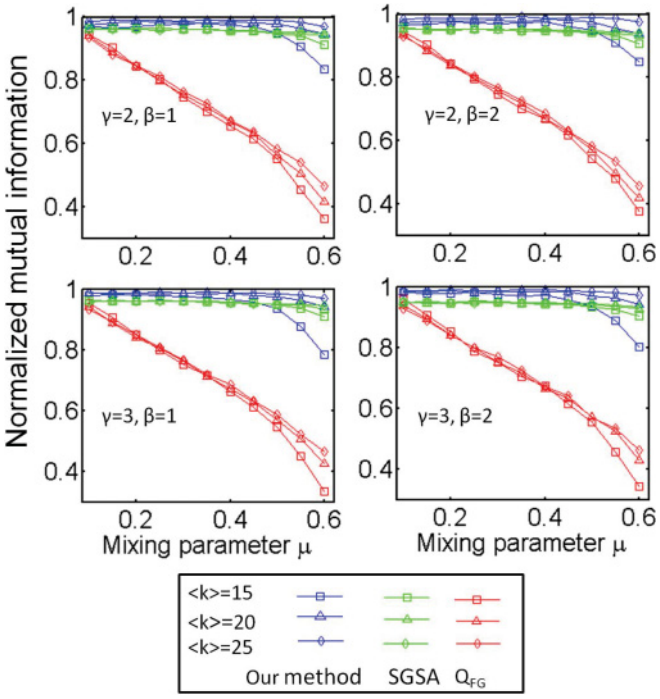

FIG. 5. (Color online) Tests of our method on a single network using the benchmark suggested in Ref. [17]. We also compared it with two modularity optimization algorithms: the fast greedy modularity optimization method (Q_{FG}) [18] and the spin-glass model and simulated annealing method (SGSA) [19]. Each point corresponds to an average over 25 network realizations. Detailed parameter settings of the simulated networks can be seen in Ref. [17].

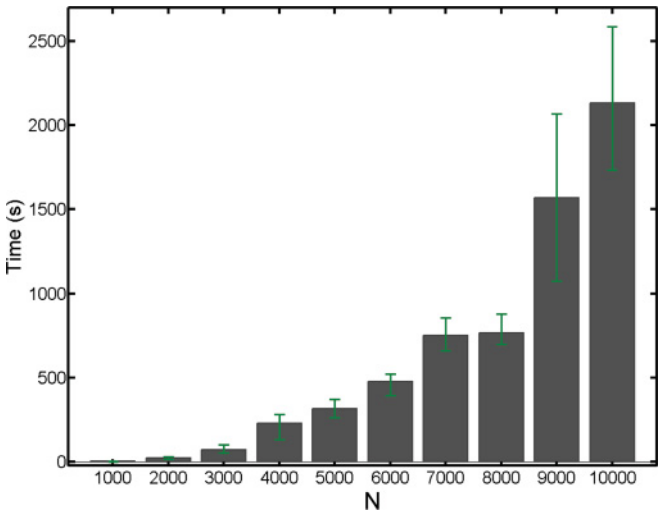

FIG. 6. (Color online) The computation time (in seconds) with network size of $n = 1000$ to 10 000. Each bar corresponds to an average over 25 network realizations.

have shown our method has competitive performance with them. These analyses partially show that our criterion for multiple networks is reasonable.

The computational efficiency of the proposed method can also be seen in the simulation study, where we have applied the reduced formulation onto a single network with 10 000 nodes. Both the theoretic and experimental analyses have shown that our method can scale well (Fig. 6).

[1] L. C. Freeman, Am. J. Sociol. **98**, 152 (1992); K. A. Eriksen, I. Simonsen, S. Maslov, and K. Sneppen, Phys. Rev. Lett. **90**, 148701 (2003); S. Zhang, G. Jin, X.-S. Zhang, and L. Chen, Proteomics **7**, 2856 (2007).

[2] J. Reichardt and S. Bornholdt, Phys. Rev. Lett. **93**, 218701 (2004); G. Palla, I. Derényi, I. Farkas, and T. Vicsek, Nature (London) **435**, 814 (2005); S. Zhang, R. S. Wang, and X. S. Zhang, Physica A **374**, 483 (2007).

[3] E. A. Leicht and M. E. J. Newman, Phys. Rev. Lett. **100**, 118703 (2008).

[4] M. E. J. Newman, Eur. Phys. J. B **38**, 321 (2004); L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, J. Stat. Mech.: Theory Exp. (2005) P09008; S. Fortunato, Phys. Rep. **486**, 75 (2010).

[5] J. Rachlin, D. D. Cohen, C. Cantor, and S. Kasif, Mol. Syst. Biol. **2**, 66 (2006).

[6] J. Leskovec, J. Kleinberg, and C. Faloutsos, in *Proc. 11th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD)*, pp. 177–187, 2005; G. Palla, A. Barabasi, and T. Vicsek, Nature (London) **446**, 664 (2007).

[7] J. Chan, J. Bailey, and C. Leckie, Knowl. Inf. Syst. **16**, 53 (2008).

[8] M. Rosvall and C. T. Bergstrom, PLoS ONE **5**, e8694 (2010).

[9] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J. P. Onnela, Science **328**, 876 (2010).

[10] T. Nepusz, A. Petóoczi, L. Négyessy, and F. Bazsó, Phys. Rev. E **77**, 016107 (2008).

[11] C. Ding, X. He, and H. D. Simon, Proc. SIAM Int'l Conf. Data Mining (SDM'05), 606 (2005).

[12] Note that each element of H in Eq. (4) is an independent variable, so this problem can be decomposed into $N \times K$ one dimensional subproblem, which can be expressed as minimization of $\lambda_1 \sum_{g=1}^{2} |(H_g)_{ij} - (H)_{ij}| + \lambda_2 |(H)_{ij}|$. The optimal solution of this one-dimensional subsection function subproblem can be easily obtained by considering the value of each interval.

[13] S. Zhang, R. S. Wang, and X. S. Zhang, Phys. Rev. E **76**, 046103 (2007).

[14] J. P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov, Proc. Natl. Acad. Sci. USA **101**, 4164 (2004).

[15] M. Kolar, L. Song, A. Ahmed, and E. P. Xing, Ann. Appl. Stat. **4**, 94 (2010).

[16] Q. Ho, L. Song, and E. P. Xing, Journal of Machine Learning Research - Proceedings Track **15**, 342 (2011).

[17] A. Lancichinetti, S. Fortunato, and F. Radicchi, Phys. Rev. E **78**, 046110 (2008).

[18] A. Clauset, M. E. J. Newman, and C. Moore, Phys. Rev. E **70**, 066111 (2004).

[19] J. Reichardt and S. Bornholdt, Phys. Rev. E **74**, 016110 (2006).