Physica A 523 (2019) 1263–1275

Contents lists available at ScienceDirect

Physica A

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

Clusters detection based leading eigenvector in signed networks[☆]

Yinghong Ma[a,*], Xiaoyu Zhu[a], Qinglin Yu[b,c]

[a] Business School, Shandong Normal University, Jinan, China
[b] Department of Mathematics and Statistics, Thompson Rivers University, BC, Canada
[c] School of Science, Xian Polytechnic University, Xian, China

HIGHLIGHTS

- An initializing and optimizing stages spectral algorithm is presented.
- Both the positive and negative information of edges are considered.
- The number of clusters is given automatic by the spectral algorithm.
- A fitness function is defined to fine-tune clusters of each node belongs to.
- No parameters involved in the spectral algorithm ensures the robustness of it.

ARTICLE INFO

Article history:
Received 8 November 2018
Received in revised form 16 January 2019
Available online 24 April 2019

Keywords:
Signed network
Spectral algorithm
Leading eigenvector
Clusters
Conflicts

ABSTRACT

The structural balance theory offers a comprehensive way to understand stability and tensions in social systems. However, most of the real social networks are unbalanced in which people are not exclusively divided into groups such that people within a group are friendly to each other but hostile to everyone in other groups. That is, there are conflict edges in a partition regardless of how we divide people in a given social network. The natural question to ask is that how many conflict edges should be changed to make a network balanced. Alternatively, the clustering problem is formulated to optimize minimum conflicts or maximum balanceness. In this paper, utilizing the relationship between balancedness and spectrum space, we propose a spectral algorithm based leading eigenvectors of signed networks to partition clusters and make balancedness maximum. The spectral algorithm is a two stages approach, partition subnetworks corresponding to temporary clusters to increase the objective value and fine-tune partition based on the fitness of nodes. The robustness of the algorithm is completely dependent on the adjacent matrixes of signed networks. And it can measure the balanceness of network in global way with the lowest errors. The experimental results on both real signed networks and synthetic networks demonstrate that the leading eigenvector based method is highly effective and accuracy.

[☆] This work is supported by National Science Foundation of China (No.71471106) and Natural Sciences and Engineering Research Council of Canada.

[*] Corresponding author.
E-mail address: yinghongma71@163.com (Y. Ma).

https://doi.org/10.1016/j.physa.2019.04.061
0378-4371/© 2019 Elsevier B.V. All rights reserved.

1. Introduction

In social networks, relations among members not only exhibit friendship and cooperation, but also hostility and competition. Positive and negative edges were used to describe cooperative (friendly/trustful) and competitive (hostile/distrustful) relationships. Assigning signs to edges were a better way to include additional information to networks than traditional binary or weighted approaches [1,2]. With the increase of popularity of online social networks such as Slashdot, Wikipedia and Epinions et al. signed networks as a possible theoretical model for such networks have attracted more studies, including link prediction [3,4], clustering [5,6], evolution [7,8] and so on.

The theoretical foundation of signed networks is the structural balance theory, introduced by psychologist Heider. It assumes that "the friend of my friend is my friend" and "the enemy (friend) of my friend (enemy) is my enemy" [9]. Such opinion lays the base of structural balance theory which provided a scheme to understand stability and tensions in an social system. Heider defined a balanced network as two clusters in which the nodes are positively connected within each cluster whereas negatively connected between two clusters [9,10]. Davis et al. generalized balancedness of the structural balance theory to weak balancedness. A weakly balanced network is composed of more than two clusters without conflicting edges [10]. This means that weakly balanced networks satisfy "the enemy of my enemy is my enemy".

However, real social networks are usually not structural balanced or weak balanced. Therefore, an interesting topic is raised: What is the minimum conflict edges are required to change their signs such that the resulting network becomes structural (weak) balanced? Up to now, a large number of approximate algorithms on detecting clusters and computing conflicts in signed networks have emerged, despite that the problem has been proved to be NP-hard [11].

The clustering algorithms for signed networks are roughly divided into four categories: (i) The first class is generalizing the traditional algorithms in unsigned networks to signed networks, such as the signed graph Laplacian matrix [12] which is generalized from the traditional unsigned Laplacian matrix [13], SBDSLPA algorithm [14] based on traditional label propagation algorithm [15], and GN-H algorithm [16], FEC algorithm [17], and CRA algorithm [18] which are called two-stage algorithms. In the two-stage algorithms, the initial clusters are detected on positive subnetwork and then modified by the information of negative edges in the first and the second stage respectively. The normalized cut [19] or approximate signed cut [17] have been used to modify initial results at the second stage. However, it is because the initial clusters are detected by the positive subgraphs without negative information, the modify stage have no chance to get optimal clusters basically. (ii) The second class is cluster dynamics to partition signed networks. The main idea of cluster dynamics is that two nodes with a positive link approaching to each other gradually while two nodes with a negative link growing away from each other, such as DEC algorithm [20] and DBAS algorithm [21]. In these methods, a dynamical equation is introduced to evolve the phases of nodes, and when reach to convergence state the nodes have consistent phase belong to a cluster and the nodes with inconsistent phases belong to different clusters. Yet, these methods have two defects [22]: One is the boundaries between clusters are not obvious if the number of clusters is very large. The other is the elements in clusters cannot get consistent phase state if the network deviates from balance to a greater extent. (iii) The third class is taking the evolutionary methods to optimize the objective function which represents the sum of negative weights inside clusters and positive weights across clusters [23]. When the objective function value is optimized, the corresponding node's classification is cluster. The possible optimization methods include genetic algorithm [24], memetic algorithm [25], particle algorithm [26], non-negative matrix factorization [27] and so forth. However, the evolutionary algorithms depend on the rational parameters and effective initialization very much during evolution. And the evolutionary results will vary greatly for the different parameters or initialization, thus leading to the partition results not being unique and lacking robustness. In addition, some algorithms such as non-negative matrix factorization [27] partition the fixed number of clusters. As a result, the value of the objective function is not optimal for the system. (iv) The fourth class is establishing statistical probability models to measure the blockness of the network such as SSBM model [28] etc. The model parameters provide vital clues about the probabilities that each node belongs to different clusters and the centrality of each node in its corresponding cluster. Ref. [29] extended Potts model incorporating negative links and social balance theory, resulting in a method to partition the signed networks. The comparison experiments proves that the method of calculating network connection by mathematical scaler can improve the accuracy of partition.

To tackle some of above shortcomings, we explore the relationship between network balance and spectral characteristics proposing a new algorithm based on the spectral characteristics to partition the signed networks into 2 or more clusters, the positive (negative) edges inside the cluster as much (few) as possible and the positive (negative) edges between clusters as few (more) as possible. The spectral algorithm transforms the balancedness of the network into spectral features of the adjacent matrix for measurement, and calculates the information of positive and negative edges simultaneously. The spectral algorithm does not involve any variable parameters both in initialization and partitioning process. Combining with the hierarchical idea of GN-H [16], the spectral algorithm improves the objective function by repeatedly partitioning clusters. Moreover, we add an optimizing segment to fine-tune the obtained clusters further.

The arrangement of this issue is as follows: The basic definitions are introduced in Section 2. In Section 3, a novel spectral method to partition signed networks into clusters is presented. The feasibility of this method is verified by a benchmark. In Section 4, experiments on both real and synthetic networks are presented to test the robustness of this method, and the superiors are proved by comparing with other five algorithms. Discussions and conclusions are in the final section of this paper.

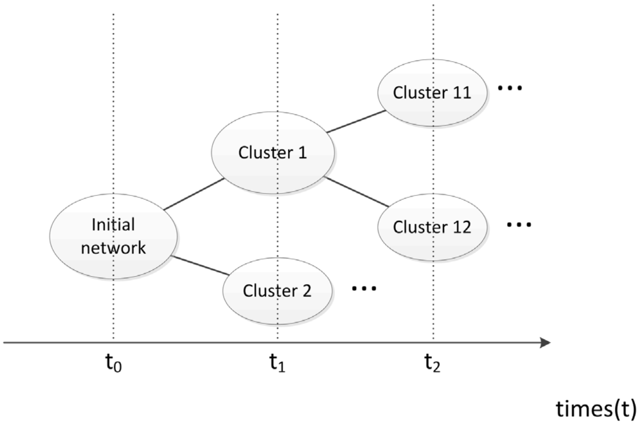


Fig. 1. The partition process. The lateral axis shows the time.

2. Signed networks and its cluster structures

A signed network is denoted by $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes and $E = \{(v_i, v_j) \mid v_i, v_j \in V\}$ is the set of edges, its adjacent weight matrix is denoted by $A = (A_{ij})_{n\times n}$, where $A_{ij} > 0$, < 0, or $= 0$ indicates a positive edge, a negative edge, or no edge between nodes v_i and v_j, respectively, and absolute value $|A_{ij}|$ denotes connection strength on the edge (v_i, v_j). Throughout this paper, we assume that networks are undirected and connected.

In (weak) balanced signed networks, nodes are partitioned into two (or more than two) clusters such that the nodes within the same cluster are connected by positive edges while the nodes between different clusters are connected by negative edges [10]. Yet, for unbalance signed networks, no matter how nodes are partitioned, there always exist conflict edges, that is, either positive edges between two clusters or negative edges within a cluster. The problem of cluster partition is to find some nodal groups, among which the number of conflict edges is minimized [7,23]. Therefore, the objective function is

$$H = \sum_{(v_i,v_j)\in E} A_{ij}\Theta_{c_i c_j}. \tag{1}$$

where c_i is a cluster with v_i belonging to, $\Theta_{c_i c_j} = 1$ if $c_i = c_j$; Otherwise, $\Theta_{c_i c_j} = -1$. We rewrite the objective function (1) in an equivalent form

$$H = \sum_{c_i=c_j} A_{ij} - \sum_{c_i\neq c_j} A_{ij}. \tag{2}$$

Radicchi et al. [30] proved that minimizing conflict edges in unbalance networks is equivalent to a satisfiability problem, which means the optimization of Eq. (1) or (2) is NP-Complete. So we turn to design an approximate algorithm to find nearly optimal solutions for Eq. (1) or (2). To get a good approximation to optimal solutions quickly, we use a heuristic approach which refining a partition further by partitioning a cluster each time to create a hierarchical structure displayed in Fig. 1. In this process, each term in the second sum $\sum_{c_i\neq c_j} A_{ij}$ of (2) remains there as process goes on, because once a pair of nodes v_i, v_j is partitioned into different clusters, they will always belong to different clusters in remaining process. This process offers a robust way to increase the value of objective function (2) and thus reaching to an approximation of optimal value quickly.

3. Detect method

In this section, we propose a method that partitions a signed network into two clusters, which relies on the relationship between eigenvectors of the adjacent matrix A and its topological structure (balanced or unbalanced). Let λ_i be the ith largest eigenvalue of A, and u_i be the eigenvector corresponding to λ_i.

3.1. Relation between eigenvectors and balancedness

A cycle is *balanced* if it contains an even number of negative edges. A signed network can be partitioned into two clusters without any conflicts (i.e., network is balanced) if and only if all its simple cycles are balanced [9,10]. Next, we analyze the relation between eigenvectors and balancedness of simple cycles in signed networks. Since $Au_i = \lambda_i u_i$, the value of jth entry of u_i is determined by the values of its neighbors

$$u_{ij} = \frac{1}{\lambda_i} \sum_{v_k \in N(v_j)} A_{jk} u_{ik}, \tag{3}$$

where $N(v_j)$ denotes the neighbors of the node v_j. The above equation demonstrates that each value u_{ij} equals a weighted sum of those values u_{ik} over all its neighbors v_k with the weight A_{jk}/λ_i. Therefore, for the edge (v_j, v_k), the contribution of u_{ik} to u_{ij} is $u_{ik}A_{jk}/\lambda_i$. For a balanced network G, all simple cycles in G are balanced and u_{ij} is positively (negatively) related to u_{ik} if and only if the nodes v_j, v_k belong to the same (different) cluster, the sign on (v_j, v_k) is positive. Hence,

$$\begin{cases} u_{ij}u_{ik} > 0, & \textit{if } c_j = c_k; \\ u_{ij}u_{ik} < 0, & \textit{if } c_j \neq c_k. \end{cases} \tag{4}$$

An analysis similar to Eq. (4) can be found in [31]. For unbalanced networks, (4) may not hold. However, intuitively we can use the value of $A_{jk}u_{ik}u_{ij}$ as a measurement to balancedness of an edge (v_j, v_k): the larger (smaller) $A_{jk}u_{ik}u_{ij}$ is, the more balanced (unbalanced) of the edge (v_j, v_k) becomes. Thus, the objective function of cluster detection in signed networks can be modified to

$$B_s = \sum_{(j,k)\in E} u_{ij} A_{jk} u_{ik}. \tag{5}$$

Under normalization ($u_i^T u_i = 1$), the quantity (5) has the upper bound λ_1 and the maximum can be attained at u_1, that is, $max \sum_{(j,k)\in E} u_{ij}A_{jk}u_{ik} = \sum_{(j,k)\in E} u_{1j}A_{jk}u_{1k}$. Consequently, the leading eigenvector u_1 can be used as an index to measure balance of signed networks. Therefore, we utilize the leading eigenvector u_1 to partition signed networks into two clusters,

$$\begin{cases} v_i \in Cluster_1, & \textit{if} \quad u_{1i} > 0; \\ v_i \in Cluster_2, & \textit{if} \quad u_{1i} < 0. \end{cases} \tag{6}$$

3.2. Optimization

Recursively, a network (or a subnetwork) is partitioned into two clusters to improve the objective based on the information from the leading eigenvector u_1 discussed in the previous section. For the normalized u_1 (i.e., $|u_1| = 1$), its entries has the bound $-1 < u_{1i} < 1$ ($i = 1, 2, \ldots, n$). If we take 1 as the nucleus of one cluster, then -1 is the nucleus of another cluster. This means that the larger the value $|u_{1i}|$ is, the closer the node v_i is to the center of the cluster. However, due to unbalance, some nodes with tiny entry values in u_1 (i.e., $u_{1i} \approx 0$) may end up in a wrong cluster. To improve the accuracy of partition, we adjust the temporary established clusters. Considering the objective function, we define *fitness* for each node v_i,

$$f_i = \sum_{v_j \in N(v_i)} A_{ij} \Theta_{c_i c_j}. \tag{7}$$

We assess the change of the objective H after moving a node v_i from its current assigned cluster to another cluster. From (7), clearly $\Delta f_i = \Delta H$ (where Δ denotes the net change). Thus, we can improve H if Δf_i is positive. Therefore, we define the *fitness* $f_{i;k}$ of node v_i in relation to a temporary cluster TC_k (i.e., when it belongs to TC_k) as

$$f_{i;k} = \sum_{v_j \in N(v_i)} A_{ij} \Theta_{c_i c_j}. \tag{8}$$

The above equation offers an effective way to improve the partitioned nodes. That is, we relocate the node v_i to a perpetual cluster PC_k if and only if

$$\max_r f_{i;r} = f_{i;k}. \tag{9}$$

3.3. Algorithm

Based on the ideas developed in the previous two sections, we design an algorithm to partition signed networks as follows. For the convenience of description, we use *temporary cluster* (TC) to denote clusters that require to be partitioned further, and *perpetual clusters* (PC) to denote clusters that require no further partition.

In the algorithm, complexity of the initialization is $O(1)$. The Step 1 consists four parts: complexity of the Step 1.1 is $O(n^2)$; according to the Power Iteration method, complexity of the Step 1.2 is $O(n^2)$; complexity of the Step 1.3 is $O(n)$; complexity of the Step 1.4 is $O(1)$. Combining with the number of clusters, k, thus the total complexity of the Step 1 is $k[O(n^2) + O(n^2) + O(n) + O(1)] \approx O(kn^2)$. Complexity of the Step 2 is $O(nkd)$, where d is the average degree. Therefor, the complexity of the spectral algorithm is $O(n^2k)$ (See Table 1).

Table 1
Spectral algorithm (SA) based leading eigenvector.

```
Input: Adjacent weighted matrix $A$.
Output: $PC_1, PC_2, \ldots, PC_k$
Initialization: Set $TC_1 = \{v_1, v_2, \ldots, v_n\}$.
Step 1. The initial partition, forming temporary clusters.
    For each temporary cluster $TC_i$, do following:
    Step 1.1. Extract subnetwork $A^i$ corresponding to $TC_i$;
    Step 1.2. Compute the leading eigenvector $u_1$ of $A^i$;
    Step 1.3. Partition $TC_i$ into two temporary clusters $TC_{i1}$ and $TC_{i2}$ according to Eq. (6).
    Step 1.4 If $TC_{i1} = TC_i$ and $TC_{i2} = \emptyset$, then $TC_i$ cannot be further partitioned.
    Until there is no temporary cluster can be partitioned further more.
Step 2. Optimization phase, fine-tuning partition based on fitness of nodes.
    Step 2.1 For each node $v_i$, computing its fitness $f_{i;k}$ as it belongs to $TC_k$ by Eq. (8).
    Step 2.2 Relocating $v_i$ according to Eq. (9) if necessary.
    Step 2.3 Until all the nodes no longer need to be relocated stop Step 2.
```

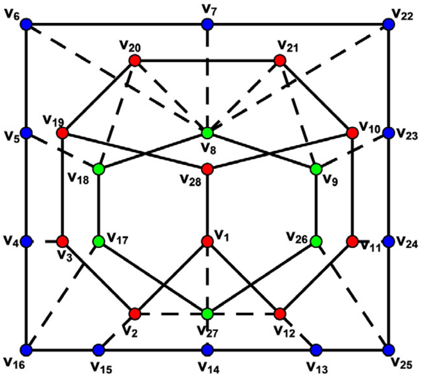


Fig. 2. The network $S1$, the solid lines are positive edges and the dotted lines are negative.

3.4. A benchmark

To compare with the existing algorithms, in this section, we use a synthetic network, denoted by $S1$ and shown in Fig. 2 which was used widely to test the validity of partition algorithms [17,20,21,26].

At first stage, we calculate the eigenvalues of the network and the subnetworks formed by clusters. The eigenvectors of the leading eigenvalue u^{S1} of network S1 is shown in Table 2). $S1$ is partitioned into two temporary clusters $TC_1 = \{v_8, v_9, v_{17}, v_{18}, v_{26}, v_{27}\}$ and $TC_2 = V - TC_1$ according Eq. (6).

For simplicity, we also denote the subnetworks by same notations as the clusters in this section. We continue to compute the leading eigenvalue u^{TC_1} and u^{TC_2} of the subnetworks TC_1 and TC_2 respectively, shown lines u^{TC_1} and u^{TC_2} in Table 2. All the values of u^{TC_1} are greater than 0, so TC_1 do not partitioned by Eq. (6). And TC_2 is need to partition into two temporary clusters TC_{21} and TC_{22} by Eq. (6). We go on computing the eigenvectors of the leading eigenvalues of subnetworks TC_{21} and TC_{22}, shown as Table 2. All the values in the two lines $u^{TC_{21}}$ and $u^{TC_{22}}$ are positive. Then output temporary clusters $TC_1 = \{v_8, v_9, v_{17}, v_{18}, v_{26}, v_{27}\}$, $TC_{21} = \{v_1, v_2, v_3, v_{10}, v_{11}, v_{12}, v_{19}, v_{20}, v_{21}, v_{28}\}$ and $TC_{22} = \{v_4, v_5, v_6, v_7, v_{13}, v_{14}, v_{15}, v_{16}, v_{22}, v_{23}, v_{24}, v_{25}\}$ by Eq. (6).

The second stage fine-tuning is to optimal objective Eq. (1). The fitness of all nodes respect to the temporary clusters are shown in Table 3. It is found that there is no any node need to be repartitioned according to Eq. (9). Therefore, output the perpetual clusters: $PC_1 = TC_1$, $PC_2 = TC_{21}$ and $PC_3 = TC_{22}$. The result coincides the previous research [17,20,21,26].

4. Experiments

Experiments of cases studies on both real networks and synthetic networks verify the effectiveness, and experimental comparison with five other algorithms proves the superiority of the method in this issue.

4.1. Data

Two classes data set are used to test performance of the spectral algorithm. One is real data set and the other is synthetic data set.

The real data are 9 real networks: Benchmark S1 [17], 4 real signed social networks and 4 subgraphs from Epinions. 4 real signed networks includes U.S. supreme court justices network (SCJ) [32], Slovene parliamentary party network

Table 2
The leading eigenvectors of $S1$ and the subnetworks induced by clusters.

Node	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
u^{S1}	−0.0924	−0.0678	−0.0379	−0.0407	−0.1504	−0.2432	−0.2625	0.5193	0.2923	−0.1179
u^{TC_1}	–	–	–	–	–	–	–	0.4082	0.4082	–
u^{TC_2}	0.3363	0.3028	0.2755	−0.1828	−0.0809	−0.0399	−0.0290	–	–	0.2730
$u^{TC_{21}}$	0.3769	0.2489	0.2489	–	–	–	–	–	–	0.3769
$u^{TC_{22}}$	–	–	–	−0.2887	−0.2887	−0.2887	−0.2887	–	–	–
Node	v_{11}	v_{12}	v_{13}	v_{14}	v_{15}	v_{16}	v_{17}	v_{18}	v_{19}	v_{20}
u^{S1}	−0.0379	−0.0678	−0.0031	−0.0362	−0.0031	−0.0433	0.1222	0.2923	−0.1179	−0.3282
u^{TC_1}	–	–	–	–	–	–	00.4082	0.4082	–	–
u^{TC_2}	0.2755	0.3028	−0.2219	−0.1617	−0.2219	−0.1470	–	–	0.2730	0.1557
$u^{TC_{21}}$	0.2489	0.2489	–	–	–	–	–	–	0.3769	0.2489
$u^{TC_{22}}$	–	–	−0.2887	−0.2887	−0.2887	−0.2887	–	–	–	–
Node	v_{21}	v_{22}	v_{23}	v_{24}	v_{25}	v_{26}	v_{27}	v_{28}		
u^{S1}	−0.3282	−0.2432	−0.1504	−0.0407	−0.0433	0.1222	0.1328	−0.0856		
u^{TC_1}	–	–	–	–	–	0.4082	0.4082	–		
u^{TC_2}	0.1557	−0.0399	−0.0809	−0.1828	−0.1470	–	–	0.3204		
$u^{TC_{21}}$	0.2489	–	–	–	–	–	–	0.4497		
$u^{TC_{22}}$	–	−0.2887	−0.2887	−0.2887	−0.2887	–	–	–		

The lines of u^{S1}, u^{TC_i}, $u^{TC_{ij}}$ are eigenvectors respect to leading eigenvalues of networks S1, TC_i, TC_{ij}
"–" in the table shows the node is not included respect to the subnetwork.

Table 3
The fitness of $S1$.

Nodes	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}	v_{13}	v_{14}
$f_{i;1}$	−1	−1	0	0	−1	−1	−1	2	2	0	0	−1	0	−1
$f_{i;21}$	3	2	2	−1	0	0	0	−2	−1	3	2	2	−1	0
$f_{i;22}$	0	−1	−1	2	2	2	2	−3	−1	0	−1	−1	2	2
Nodes	v_{15}	v_{16}	v_{17}	v_{18}	v_{19}	v_{20}	v_{21}	v_{22}	v_{23}	v_{24}	v_{25}	v_{26}	v_{27}	v_{28}
$f_{i;1}$	0	−1	2	2	0	−2	−2	−1	−1	0	−1	2	2	0
$f_{i;21}$	−1	0	0	−1	3	2	2	0	0	−1	0	0	−3	3
$f_{i;22}$	2	2	−1	−1	0	0	0	2	2	2	2	−1	−1	0

Table 4
The statistical properties of real social networks.

	S1	SCJ	SPP	GGS	SM	Net.123	Net.1667	Net.2411	Net.55568
n	28	9	10	16	18	3360	18	1039	166
# of edges	37	36	45	58	78	88769	22	30154	1024
% of negative edges	45.9%	47%	60%	50%	47.4%	21.4%	27.3%	14.8%	6.6%

Table 5
The synthetic networks with different p and s, where $n = 450$.

p, s	$p = 0.3, s = \|E\|/20$	$p = 0.4, s = \|E\|/20$	$p = 0.4, s = \|E\|/10$	$p = 0.2, s = \|E\|/20$
$+/-$edges	11338/19193	15197/25403	15703/24915	7430/12560
Conflict edges	1526	2030	4062	1000

(SPP) [33], GahukuGama subtribes network (GGS) [34], and Sampson monastery network (SM) [35]. The other 4 real networks are subgraphs from Epinions. Epinions is a large-scale signed network which is collected by the website https://snap.stanford.edu/data/signnets. It consists of 131,828 nodes and 841,372 edges of which the ratio of positive edges is 86.0%. In order to obtain tested data directly, we extract the 4 connected subnetworks by the following method: Select a node together with its first and second order neighbors as the node set of the subnetwork, and the edge set includes the edges and their signs. The scalars of the 4 extracted subnetworks are shown in Table 4, where Net.x denotes the subnetwork composed of node x and its two-ordered neighbors. By Table 4, Net.123 and Net.2411 have much larger node size and edge size than Net.1667 and Net.55568. Net.123 and Net.2411 are looked as the dense networks and Net.1667 and Net.55568 are the sparse.

The synthetic networks are a set of unbalanced synthetic signed networks generated by the following three steps: At first step, a balanced network with given n nodes and given number of clusters is generated. For example, let the nodes $n = 450$ and clusters number three, sizes of the three clusters be 100, 150 and 200 respectively. At second step, random generate edges with probability p, and set the positive sign on intra-cluster edges and negative sign on inter-cluster edges. And at third step, random pick up s edges in the network and reverse their signs. Such signed networks are unbalanced with s conflicts. The parameters p and s of the synthetic unbalanced signed networks are in Table 5.

Table 6
The leading eigenvalue and the fitness of SCJ network.

Nodes	1Stevens	2Ginsburg	3Souter	4Breyer	5Kennedy	6Alito	7Roberts	8Scalia	9Thomas
u^{SCJ}	0.3857	0.3453	0.2714	0.3162	−0.1417	−0.3037	−0.3241	−0.4073	−0.4185
$f_{i;1}$	67	87	73	79	11	−37	−35	−81	−95
$f_{i;2}$	−99	−59	−33	−46	70	103	110	97	88

Table 7
The leading eigenvalues of SM network.

Node	1John Bosco	2Gregory	3Basil	4Peter	5Bonaventur	6Berthold	7Mark	8Victor	9Ambrose
u^{SM}	0.1538	0.3034	0.2869	−0.4584	−0.1519	−0.3070	0.1990	−0.3952	−0.2139
u^{TC_2}	−0.0160	0.4875	−0.3565	–	–	–	0.2340	–	–
Node	10Romuald	11Louis	12Winfrid	13Amand	14Hugh	15Boniface	16Albert	17Elias	18Simplicius
u^{SM}	−0.1420	−0.2380	0.0707	0.0139	0.1475	0.0924	0.1239	0.2301	0.2291
u^{TC_2}	–	–	0.2514	−0.3978	0.1217	0.4518	0.2945	−0.1436	−0.1800

"–" in the table shows it is not included respect to the eigenvalue.

Table 8
The fitness of SM network.

Node	1John Bosco	2Gregory	3Basil	4Peter	5Bonaventur	6Berthold	7Mark	8Victor	9Ambrose
$f_{i;21}$	4	11	−5	−13	0	−3	7	−6	2
$f_{i;22}$	3	−2	9	−7	2	−8	−2	−7	−6
$f_{i;1}$	−4	−10	−13	14	8	8	−6	11	7
Node	10Romuald	11Louis	12Winfrid	13Amand	14Hugh	15Boniface	16Albert	17Elias	18Simplicius
$f_{i;21}$	0	−2	6	−6	3	9	6	−2	1
$f_{i;22}$	−1	−3	2	5	2	−5	−3	5	8
$f_{i;1}$	5	7	2	2	−4	−1	−3	−8	−7

4.2. Case study: real social networks

To test performance of the spectral algorithm, we apply it to real social networks, such as SCJ, SPP, GGS, SM. We give the detailed process of applying the spectral algorithm to SCJ and SM only. The computations for partitioning SPP and GGS are very similar.

First, we compute the leading eigenvector u^{SCJ} of the SCJ network, shown in Table 6. In the initial step, SCJ is partitioned into two temporary clusters, $TC_1 = \{$*1Stevens, 2Ginsburg, 3Souter, 4Breyer*$\}$ and $TC_2 = \{$*5Kennedy, 6Alito, 7Roberts, 8Scalia, 9Thomas*$\}$. Checking the fitness Table (Table 6), there is not any node to be repartitioned in the optimization segment. Then, TC_1 and TC_2 are perpetual clusters, i.e., $TC_1 = PC_1$ and $TC_2 = PC_2$. However, both $f_{5;1}$ and $f_{5;2}$ are larger than zero, and $f_{5;1} = 11 < f_{5;2} = 70$ indicates that the positive weights between v_5 and TC_1 is unbalanced and conflictive. The result is the same as the previous researches [6,14,17,20,21,25–29,32].

The original *SM* network is directed, so we convert it into an undirected version as suggested in [36]: If $A_{ij}A_{ji} = 0$ and $A_{ij} + A_{ji} \neq 0$, then $A_{ij} = A_{ji} = A_{ij} + A_{ji}$; If $A_{ij}A_{ji} < 0$, then $A_{ij} = A_{ji} = A_{ij} + A_{ji}$; If $A_{ij}A_{ji} > 0$ and $A_{ij} + A_{ji} > 0$, then $A_{ij} = A_{ji} = \max\{A_{ij}, A_{ji}\}$; If $A_{ij}A_{ji} > 0$ and $A_{ij} + A_{ji} < 0$, then $A_{ij} = A_{ji} = \min\{A_{ij}, A_{ji}\}$.

For the *SM* network, its leading eigenvector u^{SM} is shown in Table 7. So the network is partitioned into two temporary clusters $TC_1 = \{$*4Peter, 5Bonaventur, 6Berthold, 8Victor, 9Ambrose, 10Romuald, 11Louis*$\}$ and $TC_2 = \{$*1JohnBosco, 2Gregory, 3Basil, 7Mark, 12Winfrid, 13Amand, 14Hugh, 15Boniface, 16Albert, 17Elias, 18Simplicius*$\}$. For the subnetwork corresponding to TC_2, its leading eigenvector u^{TC_2} is shown in Table 7. So TC_2 is partitioned into two clusters $TC_{21} = \{$*2Gregory, 7Mark, 12Winfrid, 14Hugh, 15Boniface, 16Albert*$\}$, $TC_{22} = \{$*1JohnBosco, 3Basil, 13Amand, 17Elias, 18Simplicius*$\}$. The eigenvectors respect to the leading eigenvalues of TC_1 and TC_{21} and TC_{22} are all positive.

In the optimizing segment, the fitness matrix is computed (Table 8), from which the node 1*JohnBosco* is relocated to TC_{21} since $f_{i;21} > f_{i;22} > f_{i;1}$, where $f_{i;21}, f_{i;22}, f_{i;1}$ correspond to TC_{21}, TC_{22}, TC_1, respectively. Hence TC_1, TC_{21}, TC_{22} are all stable and become perpetual clusters. Thus, we obtain the ground truth partition consistent with the previous research [17,35,37].

With spectral algorithm, we omit the details and give the clusters of networks SPP and GGS respectively. Two clusters in SPP are $PC_1 = \{$*1SKD, 3SDSS, 6ZS, 8SLS, 9SPS-SNS*$\}$, $PC_2 = \{$*2ZLSD, 4LDS, 5ZS-ESS, 7DS, 10SNS* $\}$, and three clusters in GGS are $PC_1 = \{$*1GAVEN, 2KOTUN, 15NAGAD, 16GAMA*$\}$, $PC_2 = \{$*3OVE, 4ALIKA, 6GAHUK, 7MASIL, 8UKUDZ, 11GEHAM, 12ASARO*$\}$, $PC_3 = \{$*5NAGAM, 9NOTOH, 10KOHIK, 13UHETO, 14SEUVE*$\}$ respectively. The results are consistent with the previous researches [6,14,17,20,21,25–29,32].

The results of clusters in the four real social networks are exactly the same as those of the previous studies, which indicate the effectiveness of the spectral algorithm.

4.3. Case study: Synthetic networks

We use an unbalanced network with parameters $p = 0.3$ and $s = |E|/20$ as an example to illustrate performance of the spectral algorithm. The example network (denoted by $S2$) contains 11338 positive edges and 19193 negative edges, of which there are 1526 conflict edges. Applying the spectral algorithm, initially all the nodes are in one cluster, the leading eigenvector u_1 of the initial network is computed and shown in red in Fig. 3(a). Based on the algorithm, $S2$ is partitioned into two clusters $TC_1 = \{v_1, v_2, \dots, v_{250}\}$ and $TC_2 = \{v_{251}, v_{252}, \dots, v_{450}\}$. In the next iteration, the subnetwork corresponding to TC_1 is considered and its leading eigenvector u'_1 is computed and shown in blue in Fig. 3(a). So the subnetwork is partitioned into two clusters $TC_{11} = \{v_1, v_2, \dots, v_{100}\}$ and $TC_{12} = \{v_{101}, v_{102}, \dots, v_{250}\}$. For the three temporary clusters TC_{11}, TC_{12} and TC_2, we compute its fitness values which are displayed in blue, black and red in Fig. 3(b). According to Eq. (9), there is no further fine-tuning required. Thus, the experiment result is identical with the design. In addition, from the experimental process, the initial partition results are already very correct, which indicates the robustness of the spectral algorithm. In Fig. 3(a), the entries of the leading eigenvector u_1 of $S2$ are in red, in which the first 250 values are negative and the rest is positive. According to the algorithm, $S2$ is divided into two clusters TC_1 and TC_2 in this iteration; the entries of the leading eigenvector u'_1 of the subnetwork corresponding to TC_1 are in blue, in which the first 100 values are positive and the rest is negative. In Fig. 3(b), the blue line $f(:;1)$ denotes the fitness values of the nodes belong to TC_{11}; the black line $f(:;2)$ denotes the fitness values of the nodes belong to TC_{12}; the red line $f(:;3)$ denotes the fitness values of the nodes belong to TC_2.

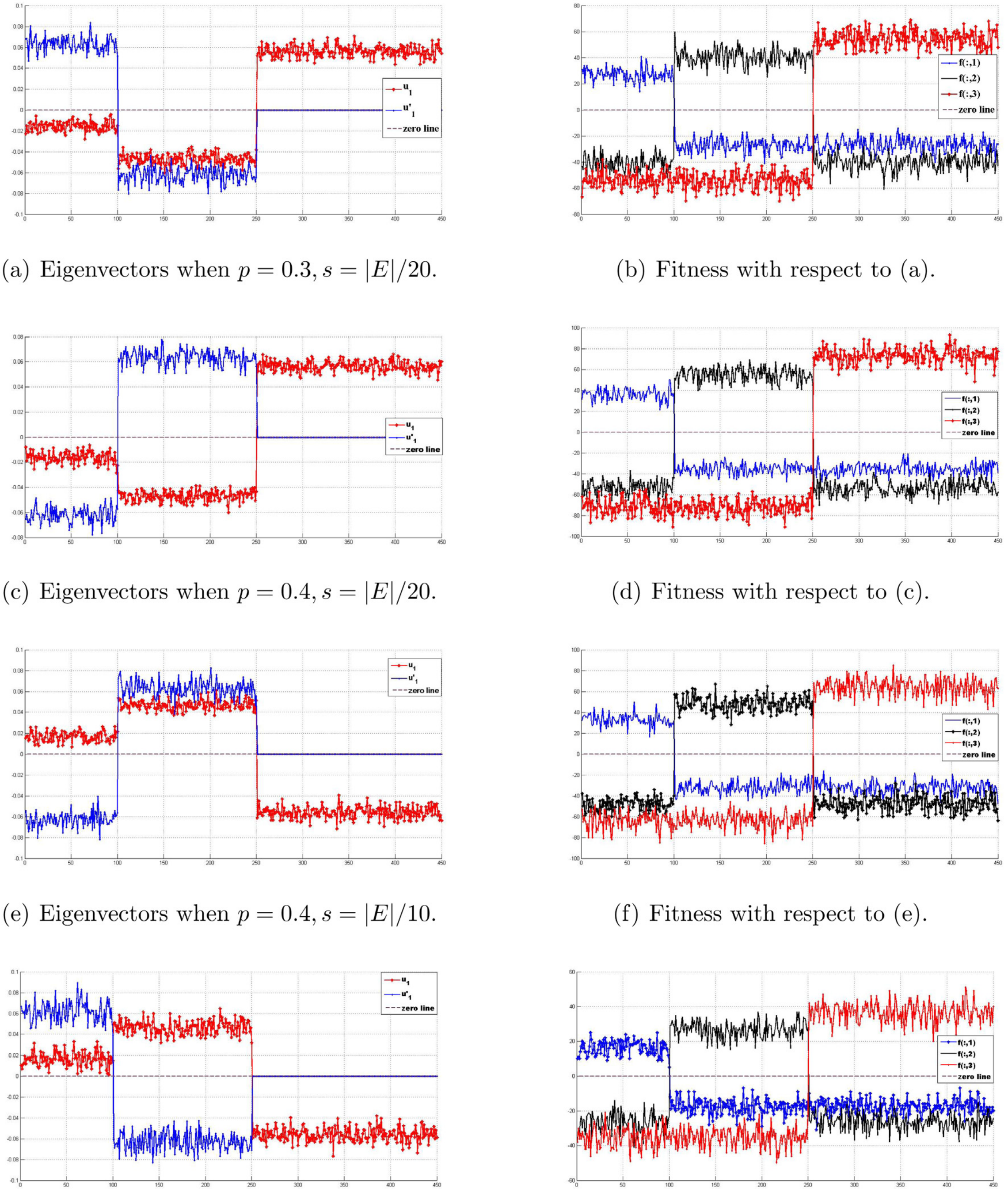
The other three networks have similar characters with $S2$, shown in Figs. 3(c) to 3(h). In Fig. 3, we noticed that nodes are grouped correctly as in their future clusters by the leading eigenvalue in the first iteration, which hints empirically that u_1 encodes the cluster structures of networks. However, the gap between eigenvectors and zeros are reduced as the ratio of the conflicts increasing. This implies that the cluster structures are ambiguous in general, and the small clusters are more likely to vanish.

Next we construct another unbalanced network (denoted by $S3$) with clusters of the same size. The network has 400 nodes, and the four clusters have nodes $v_1, v_2, \dots, v_{100}$, nodes $v_{101}, v_{102}, \dots, v_{200}$, nodes $v_{201}, v_{102}, \dots, v_{300}$ and nodes $v_{301}, v_{102}, \dots, v_{400}$, respectively. With the given 400 nodes, we generate a random network with the parameters $p = 0.2$ and $s = |E|/10$, and $S3$ contains 4713 positive edges and 11300 negative edges, of which there are 1601 conflict edges. Comparing with $S2$, the ratio of the positive edges decreases from 37.14% to 29.43%. It is clear that for signed networks, lesser the positive edges, more detrimental to the cluster structure. Moreover, the proportion of the conflict edges increases from 5% to 10%, and thus leads to a greater difficulty of partitioning. Initially, all the nodes are in one cluster, the leading eigenvector u_1 of the initial network is computed and shown in blue in Fig. 4. Based on the algorithm, $S3$ is partitioned into two clusters $TC_1 = \{v_1, v_2, \dots, v_{14}, v_{16}, \dots, v_{79}, v_{81}, \dots, v_{100}, v_{301}, \dots, v_{400}\}$ and $TC_2 = \{v_{15}, v_{80}, v_{101}, v_{102}, \dots, v_{300}\}$. In the next iteration, the two subnetworks corresponding to TC_1 and TC_2 are considered and their leading eigenvectors $u_1^{(1)}$ and $u_1^{(2)}$ are computed and shown in red and black in Fig. 4. According to the algorithm, the two subnetworks are partitioned into four clusters, $TC_{11} = \{v_1, v_2, \dots, v_{14}, v_{16}, \dots, v_{79}, v_{81}, \dots, v_{100}\}$, $TC_{12} = \{v_{301}, v_{302}, \dots, v_{400}\}$, $TC_{21} = \{v_{15}, v_{80}, v_{101}, v_{102}, \dots, v_{200}\}$ and $TC_{22} = \{v_{201}, v_{202}, \dots, v_{300}\}$. For the four temporary clusters, only TC_{21} can be further partitioned to improve the objective function, its corresponding leading eigenvector $u^{(3)}$ is computed and shown in green in Fig. 4. So TC_{21} can be partitioned into two clusters $TC_{211} = \{v_{101}, v_{102}, \dots, v_{200}\}$ and $TC_{212} = \{v_{15}, v_{80}\}$.

For the five temporary clusters, Fig. 5 indicates that nearly all the nodes are partitioned correctly by the initial partition except the nodes in TC_{212}. For these nodes of TC_{212}, their fitness associated to TC_{212} are lower than the ones associated to other clusters, therefore the nodes of $TC_{212} = \{v_{15}, v_{80}\}$ were reallocated to TC_{11}. Thus, all clusters are identified.

4.4. Compare experiments

For further test the superiors and effectiveness of the spectral algorithm (SA), we compare it with the existing algorithms, SCA [12], FEC [17], SBDSLPA [14], DEC [20] and NEA [38]. The following is the description of these algorithms. (i) SCA is a spectral clustering algorithm based on signed Laplacian matrix [12]. The top-k eigenvectors of the signed Laplacian matrix associated with the signed network are calculated, then k-means algorithm is used to partition all nodes. The similarity between SCA and SA is that both of them are based on spectral features, and the difference is that SCA is based on signed Laplace matrix while SA is based on adjacency matrix. (ii) FEC is a two-stage algorithm that detect clusters in signed networks. The first stage is to find cluster by random walks. And the second stage is to extract clusters by approximate signed cut. (iii) SBDSLPA is a signed network label propagation algorithm with structural balance degree [14], where the structural balance degree is represented by the number of balanced triangles of a node incident with. (iv) DEC is a dynamic clustering algorithm for signed networks [20]. At each time step, the phase of each node is closer to the average of its positive neighbors' phases. When reach to convergence state the nodes have consistent phase belong to a cluster and the nodes with inconsistent phases belong to different clusters. (v) NEA is to partition signed networks by evolutionary equation [38].



(a) Eigenvectors when $p = 0.3, s = |E|/20$.

(b) Fitness with respect to (a).

(c) Eigenvectors when $p = 0.4, s = |E|/20$.

(d) Fitness with respect to (c).

(e) Eigenvectors when $p = 0.4, s = |E|/10$.

(f) Fitness with respect to (e).

(g) Eigenvectors when $p = 0.2, s = |E|/20$.

(h) Fitness with respect to (g).

Fig. 3. Four pairs of graphs demonstrate algorithm process for unbalanced networks with different p and s.

4.4.1. On real social networks

We first make the comparison experiments on the 9 real social signed networks, shown in Table 4, SCJ, SPP, GGS, SM, S1 and 4 other subnetworks, Net.123, Net.2411, Net.1667 and Net.55568. The results of the experiments are shown in Table 9, where H_m and H_a are the maximum and average of the objective value of Eqs. (2).

For the results obtained by six algorithms on 9 networks, 15 of the 18 scalars of SA is superior to or equivalent to five other algorithms. Moreover, in all experimental data, H_m equals to H_a for the 9 networks which show that SA possess

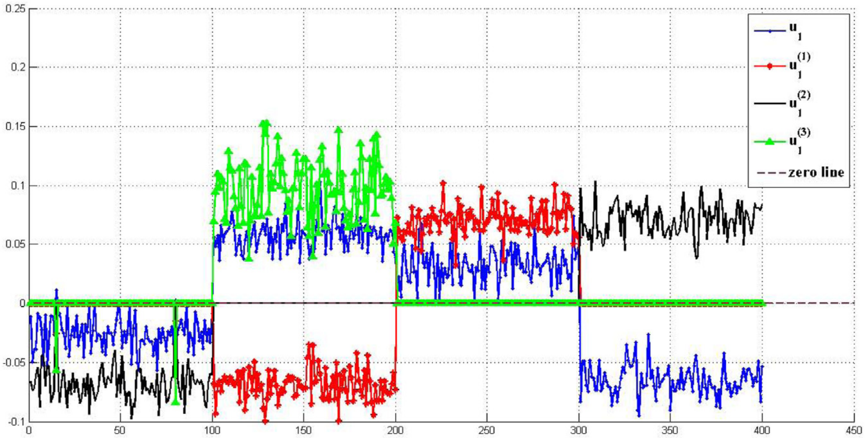


Fig. 4. The entries of the leading eigenvector u_1 of $S3$ are plotted in blue; the leading eigenvectors $u_1^{(1)}$ and $u_1^{(2)}$ corresponding to TC_1 and TC_2 are in red and black; the leading eigenvector $u_1^{(3)}$ of the subnetwork corresponding to TC_{21} is in green.

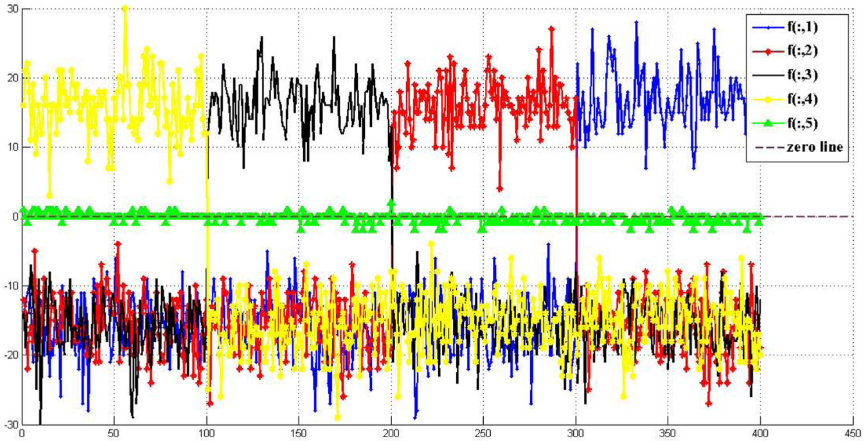


Fig. 5. The blue line $f(:; 1)$ denotes the fitness values of the nodes belong to TC_{12}; the red line $f(:; 2)$ denotes the fitness values of the nodes belong to TC_{22}; the black line $f(:; 3)$ denotes the fitness values of the nodes belong to TC_{211}; the yellow line $f(:; 4)$ denotes the fitness values of the nodes belong to $Cluster_{11}$; the green line $f(:; 5)$ denotes the fitness values of the nodes belong to TC_{212}.

Table 9
Comparison results on real social signed networks.

	SCJ	SPP	GGS	SM	S1	Net.123	Net.1667	Net.2411	Net.55568
	H_m/H_a	H_m/H_a	H_m/H_a	H_m/H_a	H_m/H_a	H_m/H_a	H_m/H_a	H_m/H_a	H_m/H_a
SCA	624/624	6408/6408	54/54	100/100	41/41	63673/63673	20/20	24778/24778	946/946
FEC	490/346	6408/6408	54/52	116/72	41/35	50813/36680	10/10	20950/20664	138/131
SBDSLPA	624/624	6408/6408	54/52	128/128	39/31	**68529**/68487	14/14	24948/24938	860/854
DEC	484/484	6408/6408	54/54	118/110	49/49	51381/51329	18/16	21252/20406	888/888
NEA	624/624	6408/6408	54/54	128/121	49/43	68469/68447	20/18	**25008/25004**	946/860
SA	**624/624**	**6408/6408**	**54/54**	**128/128**	**49/49**	68511/**68511**	**20/20**	24990/24990	**946/946**

Each experiment has been repeated 20 times independently.

the higher robustness than other algorithms. Only three scalars of SA that are inferior to SBDSLPA or NEA algorithms. In the experimental results of Net.123, the optimal value of SBDSLPA is superior to SA, while the average result of SA is superior to SBDSLPA. In the experiments on the Net.2411, the two scalars of SA are slightly inferior to NEA. To detect the deep reason of the failure of SA, we discuss the size of clusters on Net.123 and Net.2411, where the clusters of the two networks are obtained by SA, NEA and SBDSLPA. The distribution of the cluster size of these two networks are shown in Table 10.

There are large diversity of the size of clusters and the total number of clusters obtained by the three algorithms. The number of clusters association with SA are much smaller than the other two algorithms. The cluster sizes association with

Table 10
The distribution of clusters in Net.123 and Net.2411.

Net.123																
SBDSLPA	Cluster size	1	2	3	4	6	8	9	10	11	12	15	37	65	2537	
	Cluster' number	462	33	13	3	6	3	2	2	1	1	1	1	1	1	Tnb = 530
NEA	Cluster size	1	2	3	4	5	6	7	8	10	11	25	135	153	2409	
	Cluster' number	448	33	14	2	4	1	1	1	2	1	1	1	1	1	Tnb = 511
SA	Cluster size	1	2	3	4	6	8	9	10	33	38	108	159	468	2550	
	Cluster' number	0	0	0	1	0	0	0	0	1	1	1	1	1	1	Tnb = 7
Net.2411																
SBDSLPA	Cluster size	1	2	3	4	5	7	9	12	830						
	Cluster' number	125	9	5	4	1	1	1	1	1						Tnb = 148
NEA	Cluster size	1	2	3	4	6	8	14	16	841						
	Cluster' number	129	9	1	1	1	1	1	1	1						Tnb = 145
SA	Cluster size	1	2	3	4	7	12	14	174	858						
	Cluster' number	0	0	0	0	1	0	0	1	1						Tnb = 3

Tnb represents the total number of clusters respect to the algorithm.

Table 11
Comparison results on synthetic signed networks.

s	p=0.2				p=0.1				p=0.05			
	0	5%	10%	20%	0	5%	10%	20%	0	5%	10%	20%
SCA	19899	17852	15310	11259	9668	8257	7286	4770	4478	3973	3065	1955
FEC	20121	12735	10515	4243	10192	5621	4162	1702	5122	2613	4085	420
DEC	16801	−5372	−4718	−3419	8501	−2495	−2368	−1572	3783	−1357	−1249	−861
SBDSLPA	20121	18040	15968	12177	10192	9045	7956	5829	3435	2357	2178	1755
NEA	20121	18040	15968	12177	10192	9045	8080	5862	5114	4171	3233	1259
SA	**20121**	**18040**	**15968**	**12177**	**10192**	**9045**	**8080**	**6020**	**5122**	**4661**	**4025**	**2953**

SA are uniformly distributed while the other two methods are not. 87% and 88% of the clusters in Net.123 are single-node clusters association with SBDSLPA and NEA respectively. And, these data in Table 11 are 84% and 89% respectively. By the definition of cluster, those clusters obtained by SBDSLPA and NEA in Net.123 and Net.2411 are not truly clusters, which prove that SBDSLPA and NEA lost the efficiency and reasonable. Overall the above analysis of the six algorithms on the 9 networks, we find SA has superior efficiency and scientific.

4.4.2. On synthetic networks

We compare experimental results on the synthetic networks in Table 11. The generated networks contain three clusters, with sizes 100,150 and 200, respectively. The generated parameters are $p = 0.2 \sim 0.05$ and $s = 0 \sim |E|/5$. The results of Table 11 show that SA is superior to or equivalent to five other algorithms. It demonstrates that SA performs best among the six methods.

Comparing the experimental results of SA with FEC, SA takes account of both positive and negative edges information, rather than the transition probability method in FEC. In the optimization stage of SA, fitness of all nodes are checked which confirms the lowest errors. It is displayed by the difference results of FEC and SA when $s \neq 0$, FEC is far less than SA.

Comparing the experimental results of SA with SBDSLPA and NEA: SA fully considers the balancedness of the whole network rather than the triad balancedness. Hence, when the network is relatively dense, the partition results of SBDSLPA, NEA and SA are basically consistent, such as the experimental data in Table 11 as $p = 0.2$ and 0.1. However, when the network is relatively sparse, as the degree of unbalance increases, SA has an advantage over SBDSLPA and NEA, such as the experimental data as $p = 0.05$ in Table 11. In this case SA had stronger anti-noise (unbalancedness) capability than SBDSLPA and NEA.

Comparing the experimental results of SA with DEC, when measuring the structure balancedness of the network, SA is based on spectral features which can be seen as a global feature, while DEC is take the similarity of pair of nodes as measurement, which is a local feature. Moreover, SA does not involve any parameters in the calculation, while DEC relies on two uncertain parameters and random initialization during the calculation, which results in extremely low objective values.

Compared the results of SA with SCA, they are two global optimization algorithms based on the characteristics of adjacency matrix spectrum. Yet, SA not only outperforms SCA in accuracy, it runs without any prior knowledge, such as the number of clusters. All in all, SA algorithm is proved superior wherever on real data set and synthetic networks.

5. Discussion and conclusion

In this paper, we investigate relationships between balancedness of signed networks and its eigenvectors, and propose a spectral algorithm to partition signed networks to achieve maximum balancedness. For signed networks, the leading eigenvector corresponding to the largest eigenvalue reflects the maximum balancedness to a certain extent when networks are partitioned into two clusters. So we take advantage of this feature, and partition signed networks into two clusters to improve the objective value according to the leading eigenvector. Based on the weak balance theory, we repeatedly partition a subnetwork corresponding to a cluster to increase the objective value until it cannot be improved anymore. In addition, to optimize clusters, we also take a fine-tuning which relocate nodes based on their fitness. Fitness of a node can be seen as a sub-objective of Eq. (1) which shares a common goal as the original objective H. So the spectral algorithm involves two procedures of temporary partition and optimization. In the partition stage, each node is limited to choose one of two clusters, and thus reduces the chance of errors. To test the performance of the spectral algorithm, we apply the algorithm to real social networks and synthetic networks. The experimental results indicate that the spectral algorithm possesses high effectiveness, robustness and validity.

Comparing with the existing algorithms SCA, FEC, DEC, SBDSLPA and NEA, the spectral algorithm has the optimum effect both on real social and synthetic signed networks. The advantages of SA can be summarized as the following: (i)Both positive and negative edge information are considered by SA algorithm, rather than separated them. (ii) In the clustering process, SA does not involve any variable parameters, which makes its robustness very high. (iii) SA algorithm measures the balanceness of networks in global way which consistent the property of cluster and decreases the errors.

The spectral algorithm works well rely on completely by the adjacent matrix no any other prior knowledge or constraints. However, for better application in large-scale networks, the complexity of the spectral algorithm still needs to be improved. The lower efficiency is partly due to the fact that only two clusters are established in each iterative partition. For the future work, we will explore the possibility of obtaining a partition by doing matrix decomposition only once rather than repeatedly computing the leading eigenvectors. Apart from the leading one, other eigenvectors could also have exclusive relationships with cluster structures when there are more than two clusters in the network.

Acknowledgments

We would like to thank the anonymous reviewers for the constructive comments and suggestions, which undoubtedly improved the presentation of this paper. We show our great appreciation to all the authors who collected and shared the data, such as SCJ, SPP, GGS, SM and Epinions be benchmark networks. Finally, we would like to thank the National Science Foundation of China (No. 71471106) and Natural Sciences and Engineering Research Council of Canada that supports this research.

References

[1] D.J. Watts, S.H. Strogatz, Collective dynamics of small-world networks, Nature 393 (1998) 440–442.

[2] A.L. Barabasi, R. Albert, Emergence of scaling in random networks, Science 286 (1999) 509–512.

[3] S. Ranveer, A. Bibhas, Measuring the balance of signed networks and its application to sign prediction, J. Stat. Mech. (2017) 063302.

[4] J. Tang, C. Aggarwal, H. and Liu, Recommendations in signed social networks, in: Int. Conf. on WWW, 2016, pp. 31–40.

[5] P. Esmailian, M. Jalili, Community detection in signed networks: the role of negative ties in different scales, Sci. Rep. 5 (2015) 14339.

[6] P. Doreian, P. Lloyd, A. Mrvar, Partitioning large signed two-mode networks: Problems and prospects, Soc. Netw. 35 (2013) 178–203.

[7] S. Tan, J. Lu, An evolutionary game approach for determination of the structural conflicts in signed networks, Sci. Rep. 6 (2016) 22022.

[8] V.A. Traag, P.V. Dooren, P.D. Leenheer, Dynamical models explaining social balance and evolution of cooperation, PLoS One 84 (2012) 60063.

[9] D. Cartwright, F. Harary, Structural balance: a generalization of Heider's theory, Psychol. Rev. 63 (1956) 277–293.

[10] J.A. Davis, Clustering and structural balance in graphs, Hum. Relat. 20 (1967) 181–187.

[11] B. Dasgupta, G.A. Enciso, E. Sontag, Y. Zhang, Algorithmic and complexity results for decompositions of biological networks into monotone subsystems, Biosystems 90 (2007) 161–178.

[12] J. Kunegis, S. Schmidt, A. Lommatzsch, J. Lerner, E.W. De Luca, S. Albayrak, Spectral analysis of signed graphs for clustering, prediction and visualization, in: Siam. Int. Conf. Data Min., 2010, pp. 559–570.

[13] A. Azran, Z. Ghahramani, Spectral methods for automatic multiscale data clustering, in: IEEE Conf. Comput. Vis. Pattern Recog., 2006, pp. 190–197.

[14] L. Fang, Q. Yang, J. Wang, W. Lei, Signed network label propagation algorithm with structural balance degree for community detection, in: 14th Int. Conf. on Smart Homes and Health Telematics, 2016, pp. 427–435.

[15] S. Gregory, Finding overlapping communities in networks by label propagation, New J. Phys. 12 (2009) 2011–2024.

[16] X. Li, H. Chen, S. Li, Exploiting emotions in social interactions to detect online social communities, PACIS 2010 (2010) 1426–1437.

[17] B. Yang, W. Cheung, J. Liu, Community mining from signed social networks, IEEE Trans. Knowl. Data Eng. (2007) 1333–1348.

[18] T. Sharma, A. Charls, P.K. Singh, Community mining in signed social networks-an automated approach, in: Int. Conf. on Comput. Eng. Appl., 2009, pp. 152–157.

[19] K.Y. Chiang, J.J. Whang, I.S. Dhillon, Scalable clustering of signed networks using balance normalized cut, in: ACM Int. Conf. Inf. Knowl. Manag., 2012, pp. 615–624.

[20] J. Chen, H. Wang, L. Wang, W. Liu, A dynamic evolutionary clustering perspective: Community detection in signed networks by reconstructing neighbor sets, Physica A 447 (2016) 482–492.

[21] J. Wu, L. Zhang, Y. Li, Y. Jiao, Partition signed social networks via clustering dynamics, Physica A 443 (2016) 568–582.

[22] J. Chen, L. Zhang, W. Liu, Z. Yan, Community detection in signed networks based on discrete-time model, Chin. Phys. B 26 (2017) 574–583.

[23] P. Anchuri, M. Magdonismail, Communities and balance in signed networks: A spectral approach, in: IEEE/ACM Int. Conf. Adv. Soc. Networks Anal. Min., 2013, pp. 235-242.

[24] A. Amelio, C. Pizzuti, Community mining in signed networks: a multiobjective approach, in: IEEE/ACM Int. Conf. Adv. Soc. Networks Anal. Min., 2013, pp. 95–99.

[25] Y. Li, J. Liu, C. Liu, A comparative analysis of evolutionary and memetic algorithms for community detection from signed social networks, Soft Comput. 18 (2014) 329–348.

[26] Q. Cai, M. Gong, B. Shen, L. Ma, L. Jiao, Discrete particle swarm optimization for identifying community structures in signed social network, Neural Netw. 58 (2014) 4–13.

[27] Z. Li, J. Chen, Y. Fu, L. Zhang, Community detection based on regularized semi-nonnegative matrix tri-factorization in signed networks, Mob. Netw. Appl. 2 (2017) 1–9.

[28] J.Q. Jiang, Stochastic block model and exploratory analysis in signed networks, Phys. Rev. E 91 (2015) 062805.

[29] V.A. Traag, J. Bruggeman, Community detection in networks with positive and negative links, Phys. Rev. E 80 (2009) 036115.

[30] F. Radicchi, D. Vilone, S. Yoon, H. Meyer-Ortmanns, Social balance as a satisfiability problem of computer science, Phys. Rev. E 75 (2007) 026106.

[31] P. Bonacich, P. Lloyd, Calculating status with negative relations, Soc. Netw. 26 (2004) 331–338.

[32] D. Patrick, A. Mrvar, Partitioning signed social networks, Soc. Netw. 31 (2009) 1–11.

[33] S. Kropivnik, A. Mrvar, An analysis of the Slovene parliamentary parties network, in: A. Ferligoj, A. Kramberger (Eds.), Developments in Data Analysis, Metod. zv. 12 (1996) 209–216.

[34] K.E. Read, Cultures of the central highlands new guinea, Southwestern J. Anthropol. 10 (1954) 1–43.

[35] F.S. Sampson, A Novitiate in a Period of Change: An Experimental and Case Study of Social Relationships, a Novitiate in a Period of Change: An Experimental and Case Study of Social Relationships (Ph.D. thesis), Cornell University, 1968.

[36] E. Terzi, M. Winkler, A spectral algorithm for computing social balance, in: A. Frieze, P. Horn, P. Pralat (Eds.), Algorithms and Models for the Web Graph, WAW 2011, in: Lecture Notes in Computer Science, vol. 6732, Springer, Berlin, Heidelberg, 2011, pp. 1–13.

[37] Y. Su, B. Wang, F. Cheng, L. Pan, An algorithm based on positive and negative links for community detection in signed networks, Sci. Rep. 7 (2017) 10874.

[38] X. Zhu, Y. Ma, Z. Liu, A novel evolutionary algorithm on communities detection in signed networks, Physica A 503 (2018) 938–946.