IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 22, 2018, accepted December 11, 2018, date of publication January 14, 2019, date of current version May 3, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2891968*

# Community Detection in Signed Networks Based on the Signed Stochastic Block Model and Exact ICL

**SHUQIU PING[1], DAYOU LIU[1], BO YANG[1], YUNGANG ZHU[1], HECHANG CHEN[1], AND ZHENG WANG[1,2]**

[1]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, College of Computer Science and Technology, Jilin University, Changchun 130012, China
[2]Engineering Training Center, Changchun Institute of Technology, Changchun 130012, China

Corresponding authors: Dayou Liu (dyliu@jlu.edu.cn) and Bo Yang (ybo@jlu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61502198, Grant 61572226, Grant 61472161, Grant 61876069, Grant 61471133, and Grant 61871475, in part by the Humanities and Social Sciences Research Youth Project of Ministry of Education under Grant 17YJCZH261, in part by the Guangdong Natural Science Foundation under Grant 2018A030313339, in part by the Natural Science Foundation of Jilin Province under Grant 20180101041JC, in part by the Science and Technology Development Program of Jilin Province of China under Grant 20180201067GX and Grant 20180201044GX, in part by the Science Foundation of the Education Department of Jilin Province under Grant 2016186, in part by the Guangdong Science and Technology Project under Grant 2017A070712019, in part by the Guangdong Application-oriented Special Projects under Grant 2017B010126001, and in part by the Special Innovation Project of the Guangdong Education Department under Grant 2017KTSCX094 and Grant 2017KQNCX098.

**ABSTRACT** There has been an increasing interest in detecting community in signed networks because signed networks contain more information (both positive and negative edges) than unsigned networks (only positive edges). Many methods have been proposed to find communities in signed networks; however, most of them can be regarded as the discriminative methods that do not concern with how the signed networks are generated so that they are usually difficult to characterize accurately the intrinsic community structure. The existing method SSL which is based on a generative model can achieve high accuracy in community detection in signed networks. However, SSL needs to estimate the models in the possible space one by one, which requires a large amount of calculation. In view of this, we propose a method to find community in signed networks, in which the exact integrated complete data likelihood (ICLex) for the signed stochastic block model proposed in SSL is derived and a greedy search is employed to optimize the value of the derived ICLex for signed networks to find communities. Our method has a rigorous probabilistic interpretation and does not need to estimate the models one by one from the possible space. The knowledge on hyper-parameters of our model is not necessary. In the experiments, the proposed method is tested on the synthetic and real-world networks and compared with several current methods. The experimental results show that our method can find the communities in signed networks more accurately than these current methods and more efficiently than SSL.

**INDEX TERMS** Complex networks, communities, data mining.

## I. INTRODUCTION

In the real world, a lot of relationships can be represented as networks, where nodes denote actors and edges denote relationships between them. In general, an edge between a pair of nodes in an unsigned network represents there is a type of relationship between them, and the non-existence of an edge means that there is no such type of relationship between them. For example, the internet, in which nodes represent routers and edges represent the physical connections between them [1], food webs, in which an edge represents the existence of the predator-prey interaction between two species [2], and so on [3]–[6]. In contrast to unsigned networks, signed networks have more powerful description capability in that they contain not only positive edges but also negative edges. With the signs of these edges, the types of relationships as well as the existence of the relationships can be represented. For instance, positive edges can represent friendship, trust, like, etc., and negative ones can represent foe, distrust, dislike, etc. Moreover, some problems may appear if a system which contains both positive and negative relationships is represented as an unsigned network. For example, if negative relationships in the system are also


2169-3536 © 2019 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

represented as positive or nonexistent edges in an unsigned network, the original relationships of the system may be distorted.

The networks that represent relationships are often complex and difficult for people to understand. Community detection is one of the most effective tools to reduce the complexity so that the network can be more easily understood [7]. Community detection in signed networks is to identify $K$ antagonistic communities, so that most edges within communities are positive and most edges between communities are negative [8]. In this sense, communities are in line with the clusters defined in balance theory in social science [8], where a strongly (or weakly) balanced network can be divided into two (or more) clusters so that all connections between clusters are negative and all connections within clusters are positive [8]–[10]. Unfortunately, the real-world signed networks are usually unbalanced due to their frustrations, that is, positive links between clusters and negative links within clusters [8]. Therefore, it is a great challenge to design an effective and efficient method for community detection in signed networks.

There have been a number of methods proposed for community detection in unsigned networks, such as the method based on division [11], the method based on modularity [12], clique percolation method (CPM) [13], the method based on evolutionary computation [14], and so forth. These mentioned methods have their own characteristics, but they can not describe the process of network generation. Stochastic block model (SBM) can characterize the network's generation process and mine communities in a network without a priori knowledge about how the communities connect. And SBM has attracted more and more attention and has been extended in different directions since proposing originally by Holland *et al.* [15]. For example, Airoldi *et al.* [16] proposed the mixed membership SBM in which a node can be assigned to more than one community. Karrer and Newman *et al.* [17] proposed the degree-corrected SBM based on Poisson distribution. Yang *et al.* [18] proposed the method DSBM to detect communities and their evolutions in dynamic social networks. In 2016, Newman and Reiner [7] integrated the number of communities into stochastic block model based on Poisson distribution so that community assignments and community number can be found simultaneously. Côme and Latouche [19] proposed the exact integrated complete data likelihood (ICLex) and integrated the standard stochastic block model with ICLex. Wyse *et al.* [20] also derived the ICLex for a bipartite network generative model and achieved satisfactory results in their experiments. However, none of aforementioned methods can be directly applied to signed networks because negative edges are not considered. For example, the generative models [7], [19] can not generate negative edges, thus can not be used to fit to any signed network directly.

There have been many studies for community detection in signed networks. P. Doreian and A. Mrvar proposed a frustration-based method DM that finds the communities by minimizing the sum of the number of positive edges between communities and the number of negative edges within communities [21]. Yang *et al.* [22] proposed a random-walk based method FEC, which is more effective than DM. Traag and Bruggeman [23] proposed a method PSA that finds the communities by optimizing a signed modularity deduced from potts model. Gong *et al.* [24] proposed a multi-objective evolutionary based method to find communities in signed networks. However, all these aforementioned methods are not generative models based methods. They can be regarded as the discriminative methods that do not concern with how the signed networks are generated so that they are usually difficult to characterize accurately the intrinsic community structure and lack rigorous probabilistic interpretations. B. Yang *et al.* [8] proposed a method SSL to analyze signed networks. In SSL, a generative model, signed stochastic block model SSBM, was proposed. Then SSL obtains communities by parameter estimation and model selection relying on a variational Bayes EM and a variation based approximate evidence. SSL has a rigorous probabilistic interpretation and can achieve high accuracy in community detection in signed networks. However, to get the number of communities, in the range of community number from $K_{min}$ to $K_{max}$ that is given in advance, SSL needs to estimate the parameters of all the $K_{max}-K_{min}+1$ models one by one. Then every model is evaluated according to the evidence approximation. The estimated result of the model with the highest score is returned as the output of SSL. The time complexity of SSL is $O(I\times n^2\times(k_{\max}-k_{\min})^2)$ [8], where $I$ is the iteration number, $n$ is the node number. With the increasing of $K_{max}$, the increasing of the computational cost is relatively fast.

In view of this and motivated by [8], [19], and [20], a method SSBMI (signed stochastic block model and exact ICL) for detecting simultaneously the number of communities as well as community assignments in signed networks is proposed in this paper. The ICLex for SSBM is derived and communities are obtained by a greedy search to optimize the value of ICLex for SSBM. Our method has a rigorous probabilistic interpretation and does not need to evaluate and select model one by one from the possible space. The knowledge on hyper-parameters of SSBMI is not necessary.

The rest of this paper is organized as follows. In section 2, we present our model and method in detail. In Section 3, we describe the experiments and discuss the results. And we give the conclusion in section 4. The proofs of the relevant formulas are given in the appendix.

## II. MODEL AND METHODS

### A. MODEL AND NOTATIONS

Given a signed network with $n$ nodes, we represent it as a graph $G = (V, E)$, where $V$ denotes the set of nodes and $E$ denotes the set of edges. Different from the usual representation of an unsigned network, each element of $E$, $(i, j, +1/-1)$, denotes there is a positive (negative) relationship between node $i$ and node $j$. Let $W$ denote the adjacency matrix of the graph and each element of $W$, $w_{ij} \in$

$\{+1, 0, -1\}$, denotes there is a positive (non-existent, negative) edge between node $i$ and node $j$.

Then we adopt the signed stochastic block model introduced by Yang *et al.* [8] to characterize the generation process of a signed network. We assume that there are $K$ communities in the network and each node belongs to one of the $K$ communities. Let the community assignments of all these nodes be represented by a vector $Z = \{z_1, z_2, \ldots, z_n\}$, where $n$ is the node number, $z_i$ is an integer which denotes the community digital label of node $i$. $z_i = z_j$, when node $i$ and node $j$ are in the same community, otherwise $z_i \neq z_j$. Note that, to avoid maintaining the continuity of the community digital labels in $Z$ during our following optimization process, we do not require that the community digital labels in $Z$ must be continuous from 1 to $K$. For this reason, we introduce two arrays *map*, *mapi* to save the mapping from $\{1, 2, \ldots, K\}$ to the community digital labels contained in $Z$and the inverse mapping respectively. For example, $Z$ can be $\{1, 1, 3, 3\}$ rather than must be $\{1, 1, 2, 2\}$ when $K = 2, n = 4$. And the corresponding $map$=$\{1, 3\}$, Z=$\{map_1, map_1, map_2, map_2\}$.

Then assuming that the value of $z_i$ follows the multinomial distribution with parameter $\Omega$, we can draw the value of $z_i$ from the $K$ communities as follows:

$$z_i \sim M(1, (\omega_1, \omega_2, \ldots, \omega_K)), \quad \sum_k^K \omega_k = 1$$

Given the community assignments $z_i$ and $z_j$, the edge between node $i$ and $j$ is independently generated following the multinomial distribution with parameter $\Pi = (\pi_1, \pi_0, \pi_{-1})$ or with parameter $\Theta = (\theta_1, \theta_0, \theta_{-1})$ where each component in $\Pi$ ($\Theta$) denotes the probability that there is a positive edge, no edge or negative edge between a pair of nodes within the same community (in different communities).

$$w_{ij} \sim M(1, (\pi_1, \pi_0, \pi_{-1})), \quad \sum_h \pi_h = 1, \ z_i = z_j$$
$$w_{ij} \sim M(1, (\theta_1, \theta_0, \theta_{-1})), \quad \sum_h \theta_h = 1, \ z_i \neq z_j$$

Then given a signed network's directed graph, we can write the likelihood of complete data of the model as follows:

$$P(W, Z|\Omega, \Pi, \Theta) = P(W|Z, \Pi, \Theta)P(Z|\Omega) \tag{1}$$

where

$$P(Z|\Omega) = \prod_i \prod_k \omega_k^{\delta(z_i, map_k)}$$
$$P(W|Z, \Pi, \Theta) = \prod_{i\sim j} P(w_{ij}|z_i, z_j, \Pi, \Theta)$$
$$= \prod_{i\sim j} \prod_h (\pi_h^{\delta(w_{ij},h)})^{\delta(z_i,z_j)} (\theta_h^{\delta(w_{ij},h)})^{1-\delta(z_i,z_j)}$$
$$\delta(x, y) = \begin{cases} 1 & x = y \\ 0 & otherwise \end{cases}$$

### B. EXACT INTEGRATED COMPLETE DATA LIKELIHOOD FOR SSBM

So far, the parameters $\Omega$, $\Pi$, and $\Theta$ of the aforementioned model are not known. The variational Bayes approach can be employed to estimate these parameters and the hide variables. However, this approach is an approximation technique by optimizing a lower bound and needs to estimate and select model one by one from the possible model space. To address this, we introduce firstly some prior distributions for the model parameters by using the Bayesian framework [19], [25] that also was used in [8] but for a different purpose. In the following, we will see that this can lead to all these parameters being integrated out, and the derived ICLex for signed networks is only related to the community assignments rather than the posterior probabilities of the community assignments.

Specifically, Dirichlet distribution is selected for the model parameter $\Omega$.

$$P((\omega_1, \omega_2, \ldots, \omega_K)|(\rho_1^0, \rho_2^0, \ldots, \rho_K^0)) = Dir(\Omega; \mathrm{P}^0)$$

To obtain the uniform distribution conjugate prior, we set these hyper-parameters to 1, that is, $\rho_k^0 = 1 = \rho^0, \forall k$.

Since the edge between a pair of nodes is drawn from a multinomial distribution, we select Dirichlet distribution as the prior distributions of $\Pi$ and $\Theta$.

$$P((\pi_1, \pi_0, \pi_{-1})|(\eta_1^0, \eta_0^0, \eta_{-1}^0)) = Dir(\Pi; \mathrm{H}^0)$$
$$P((\theta_1, \theta_0, \theta_{-1})|(\mu_1^0, \mu_0^0, \mu_{-1}^0)) = Dir(\Theta; \mathrm{M}^0)$$

Similarly, to obtain the uniform distribution conjugate priors, we set these hyper-parameters to 1, that is, $\eta_h^0 = 1, \forall h$ and $\mu_h^0 = 1, \forall h$.

Then, with these prior distributions over the model parameters, we can get the analytical forms of the marginal distributions $P(W|Z)$ and $P(Z|K)$. And the exact integrated complete data log likelihood is as follows:

$$\begin{aligned} ICLex(Z, K) &= \log(W, Z|K) \\ &= \log\left\{\frac{\Gamma(\sum_k^K \rho_k^0)\prod_{k=1}^K \Gamma(\rho_k)}{\Gamma(\sum_k^K \rho_k)\prod_{k=1}^K \Gamma(\rho_k^0)}\right\} \\ &\quad + \log\left\{\frac{\Gamma(\sum_h \eta_h^0)\prod_h \Gamma(\eta_h)}{\Gamma(\sum_h \eta_h)\prod_h \Gamma(\eta_h^0)}\right\} \\ &\quad + \log\left\{\frac{\Gamma(\sum_h \mu_h^0)\prod_h \Gamma(\mu_h)}{\Gamma(\sum_h \mu_h)\prod_h \Gamma(\mu_h^0)}\right\} \end{aligned} \tag{2}$$

where

$$\rho_k = \rho_k^0 + \sum_i \delta(z_i, map_k)$$
$$\eta_h = \eta_h^0 + \sum_{i\sim j} \delta(w_{ij}, h)\delta(z_i, z_j)$$
$$\mu_h = \mu_h^0 + \sum_{i\sim j} \delta(w_{ij}, h)(1 - \delta(z_i, z_j))$$

The proof of formula (2) is given in the appendix.

Then different from the variational Bayes method which maximizing a lower bound for approximation, we can maximize the above analytical criterion. From above description, we can see that both the modeling process and the derived ICLex have rigorous probabilistic interpretations. And all the hyper-parameters of our model can be set according to the above description. The knowledge on them is not necessary.

### C. OPTIMIZATION

Obviously, it is not practical to get the global optimum of the aforementioned ICLex through exhaustive search. We use an optimization method leveling greedy heuristic in SSBMI, which was used in [12] and [19]. With an upper bound of the number of communities $K_{max}$ which is decided by the user according to the knowledge on the problem at hand, the optimization algorithm initializes the community assignment of each node randomly within the range of 1 to $K_{max}$. Then a sweep is executed under a random node order. Specifically, in every step of the sweep, we select a node according to the random order and test whether changing the community assignment of the node among the current $K$ non-empty communities will lead to a higher ICLex. When such a change exists, the node is assigned to the community with the maximal increase of ICLex. Otherwise, the node's community assignment remains unchanged.

Moving a node from the $p$th community to the $q$th community will cause the $p$th community to become empty or not. When the $p$th community is not empty after moving, the corresponding increment of ICLex can be computed as follows:

$$\begin{aligned} &ICLex^{try} - ICLex \\ &= \log\left\{\frac{\rho_q}{\rho_p - 1}\right\} + \log\left\{\frac{\prod_h \Gamma(\eta_h^{try})}{\Gamma(\sum_h \eta_h{}^{try})}\frac{\Gamma(\sum_h \eta_h)}{\prod_h \Gamma(\eta_h)}\right\} \\ &\quad + \log\left\{\frac{\prod_h \Gamma(\mu_h^{try})}{\Gamma(\sum_h \mu_h{}^{try})}\frac{\Gamma(\sum_h \mu_h)}{\prod_h \Gamma(\mu_h)}\right\} \end{aligned} \tag{3}$$

where $ICLex^{try}$ is the value of ICLex after moving.

When the $p$th community is empty after moving, the empty community will be eliminated. That is, the current community number $K = K - 1$. Correspondingly, the dimension of $P^0$ is reduced by 1. The corresponding increment of ICLex can be computed as follows:

$$\begin{aligned} &ICLex^{try} - ICLex \\ &= \log\left\{\frac{\rho_q}{\rho^0}\frac{\Gamma((K-1)\rho^0)}{\Gamma(K\rho^0)}\frac{\Gamma(K\rho^0 + n)}{\Gamma((K-1)\rho^0 + n)}\right\} \\ &\quad + \log\left\{\frac{\prod_h \Gamma(\eta_h^{try})}{\Gamma(\sum_h \eta_h{}^{try})}\frac{\Gamma(\sum_h \eta_h)}{\prod_h \Gamma(\eta_h)}\right\} \\ &\quad + \log\left\{\frac{\prod_h \Gamma(\mu_h^{try})}{\Gamma(\sum_h \mu_h{}^{try})}\frac{\Gamma(\sum_h \mu_h)}{\prod_h \Gamma(\mu_h)}\right\} \end{aligned} \tag{4}$$

The proofs of formulas (3) and (4) are given in the appendix.

The sweep is repeated until there is no change during a sweep. Then the community assignments of the nodes with the highest ICLex are returned as the optimization result. Obviously, the optimization method is only guaranteed to reach a local optimum. Then we can run the optimization method with multiple initializations or restarts and take the best result.

The algorithm of optimization on one initialization is presented in table 1.

**TABLE 1. Algorithm OPTIMIZATION_ONE_INI.**

Algorithm OPTIMIZATION_ONE_INI

Input: $K_{max}$, $P^0$, $H^0$, $M^0$, $W$, *table*. // $W$ is the adjacency matrix //of the graph corresponding to the given signed //network, *table* is an array for storing the log(Γ()) //values that may be used in the algorithm

Output: *Z, localI*. // *localI* is the value of ICLex with $Z$

1. Set *stop* = 0;
2. Initialize randomly $Z$ in the range 1 to $K_{max}$;
3. Compute $K$; // $K$ is the non-empty community number
4. Compute $\rho$, $\eta$, $\mu$; // $\rho = (\rho_{1,\ldots,}\rho_K)$, $\eta = (\eta_1, \eta_0, \eta_{-1})$, //$\mu = (\mu_1, \mu_0, \mu_{-1})$
5. Initialize arrays *map, mapi*; // *map* keeps the mapping //from{1, 2, ... , $K$} to the community digital labels in //the current $Z$, *mapi* keeps the inverse mapping
6. while *stop* == 0 do
7. *stop* = 1;
8. Generate a random update order of all the nodes, *order*;
9. for $i = 1 : n$
10. $j = order(i)$;
11. Count the number of positive (negative) edges between node $j$ and each non-empty community;
12. for $k$=1 : $K$
13. Compute incrementally $\eta^{try}$, $\mu^{try}$ corresponding to moving node $j$ to the $k$th community;
14. Compute the corresponding ICLex increment and keep the largest increment *maxI* among the $K$ communities;
15. end for
16. if *maxI* > 0
17. stop = 0;
18. update $Z$, $K$, *map*, *mapi*, $\rho$, $\eta$, $\mu$;
19. end if
20. end for
21. end while
22. Compute *localI*;
23. return $Z$, *localI*;

### D. COMPLEXITY

Now we analyze the time complexity of SSBMI. Computing $\rho$ in line 4 takes $O(n)$. Computing the statistics $\eta$, $\mu$ in line

4 takes $O(m)$ where $m$ is the total number of positive and negative edges in the signed network, because of computing the position number within the same community takes $O(K)$, computing the position number between different communities takes $O(1)$ and the number of 0 edges in $\eta(\mu)$ within the same community (between different communities) can be obtained by subtracting the total number of positive and negative edges within the same community (between different communities) from the number of positions within the same community (between different communities). Computing the numbers of positive (negative) edges between node $j$ and the $K$ communities in line 11 takes O($d$) where $d$ is the node degree. Initializing the edge numbers takes O($K$). Computing the maximum increment $maxI$ in line 12-15 for the selected node among $K$ communities takes $O(K)$ since updating the statistics $\eta^{try}$, $\mu^{try}$ takes $O(1)$. A sweep in line 9-20 takes $O(n\times(\bar{d}+\bar{K}))$, where $\bar{d}$ is the average degree of the nodes and $\bar{K}$ is the average community number during the sweep. In the worst case, it takes $O(n\times(\bar{d}+K_{\max}))$. Assuming the iteration number of the loop (sweep number) in line 6-21 is C, the optimization for an initialization takes $O(C\times n\times(\bar{d}+K_{\max}))$ in the worst case. In contrast to SSL, the time complexity of SSBMI increases with $K_{max}$ and does not increase with $(K_{max}-K_{min})^2$. As we will see in the experimental studies that the sweep number in line 6-21 is typically about 10 or so.

## III. RESULTS AND DISCUSSION

In this section, we present our experimental study on both synthetic and real-world networks. We compare SSBMI with 3 methods: the random-walk based FEC [22], the modularity-optimization based PSA [23], the signed stochastic block model and variational techniques based method SSL [8]. We set the hyper-parameters of SSL according to B. Yang's suggestion.

The NMI [26] between the communities obtained by a method and the ground truth is adopted to evaluate the community detection result. NMI is defined as:

$$NMI=\frac{2\sum_{k=1}^{K_A}\sum_{l=1}^{K_B}m_{kl}\log(\frac{m_{kl}n}{M_{k.}M_{.l}})}{\sum_{k=1}^{K_A}M_{k.}\log(\frac{M_{k.}}{n})+\sum_{l=1}^{K_B}M_{.l}\log(\frac{M_{.l}}{n})}$$

where $M$ denotes a confusion matrix of partition $A$ and $B$. Each entry $m_{kl}$ is the number of nodes that occur simultaneously in the real community $k$ and the detected community $l$, $K_A(K_B)$ is the number of communities of partition $A$ ($B$), $M_{k.}$ denotes the sum over the $k$th row of $M$, and $M_{.l}$ denotes the sum over the $l$th column, $n$ is the number of nodes. A large NMI means a good partition. The maximum value of NMI is 1 and the minimum value of NMI is 0.

### A. EXPERIMENTS ON SYNTHETIC NETWORKS

We choose the model $SG(c, n, d, Pin, P-, P+)$ proposed by Yang *et al.* [22] to produce synthetic signed networks, where

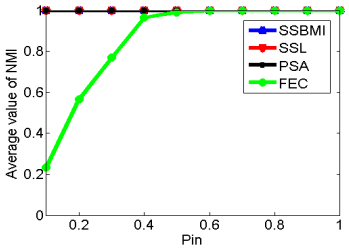



**FIGURE 1.** The test result on the balanced networks.

$c$ denotes the number of communities, $n$ denotes the number of nodes in each community, $d$ is the average degree, $Pin$ is the probability of an edge lying in a community, $P-$ is the probability that an edge within a community is negative, $P+$ is the probability that an edge between communities is positive. All the numbers of nodes in these synthetic networks are set to 1000. Because the connection capability is limited in general, the degree of nodes should not increase with the increasing of the number of nodes. We set the degree to 30 for all the synthetic signed networks. These networks are undirected and suitable for all the algorithms for comparison in our experiments. For SSL, $K_{min}$, $K_{max}$ and the restart number are set to 2, 10 and 3, respectively. For SSBMI, we set $K_{max}$ and the restart number to 20 and 500, respectively. We verify SSBMI on two types of synthetic signed networks: balanced networks and unbalanced networks.

For balanced networks, the test strategy of SSL is adopted. We use the model $SG(4, 250, 30, Pin, 0, 0)$ to produce the networks, where $Pin$ increases from 0.1 to 1 stepping by 0.1. Fig. 1 shows the performances of the four algorithms on the balanced networks. The y axis is the average NMI obtained over 10 synthetic networks. As we can see, SSL, PSA, and SSBMI find the same as ground truth for all $Pin$, and their performances do not vary with $Pin$.

To produce the unbalanced networks, the model $SG(4, 250, 30, 0.8, P-, P+)$ is used firstly, where $P-$ and $P+$ gradually increase from 0.1 to 1 stepping by 0.1, respectively. We generate 10 networks for each configuration specified by a set of specific parameters of $SG$. This test is more comprehensive than the test only using networks generated by $SG(., ., ., 0.8, P-, 0.2)$ and $SG(., ., ., 0.8, 0.2, P+)$ in [8]. Fig. 2 shows the performances of the four algorithms in this test. Each point over the 3D surfaces is the average NMI obtained for a configuration. As we can see, SSL perform better. Most of the average NMIs obtained by SSL are 1 or close to 1 in the area about $P+ < 1-P-$. Near the outside of the line $P+ = 1-P-$, SSL also get the satisfactory average NMIs. SSBMI performs the best and is the most robust to heavy frustrations among these methods. PSA has a very good performance when $P- < 0.2$. With the increasing of $P-$, the average NMIs obtained by PSA decrease and become very small when $P- >= 0.6$. On the whole, FEC does not perform better than PSA in this test.

For further validating the performance of the proposed method, We use the model $SG(4, 250, 30, 0.2, P-, P+)$

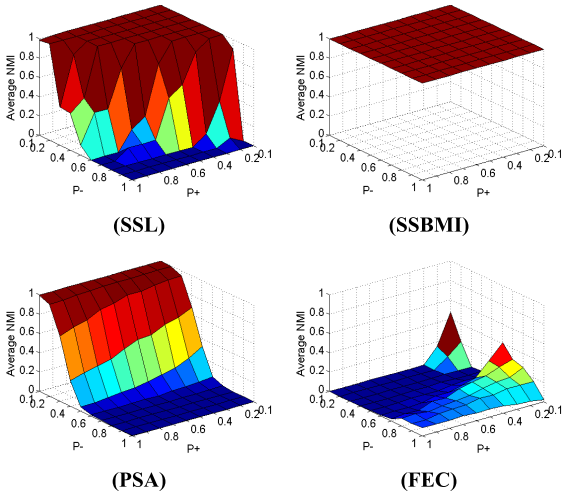



**FIGURE 2.** The test result from *SG* (4, 250, 0.8, *P*−, *P*+).

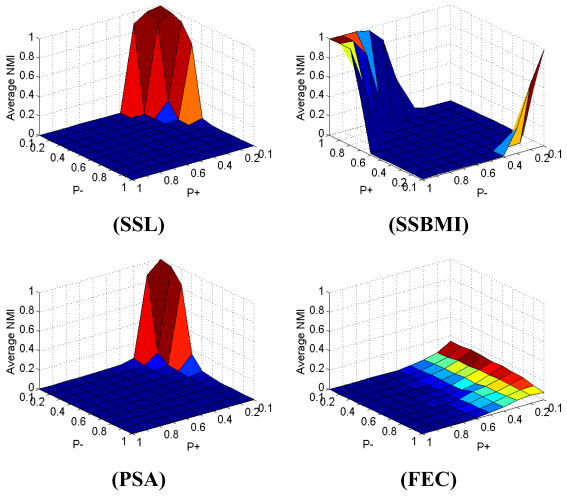



**FIGURE 3.** The test result from *SG* (4, 250, 0.2, *P*−, *P*+).

to produce the unbalanced networks, where $P-$ and $P+$ gradually increase from 0.1 to 1 stepping by 0.1, respectively. These networks are more difficult to detect. We also generate 10 networks for each configuration. Fig. 3 shows the performances of the four algorithms in this test. Each point over the 3D surfaces is also the average NMI obtained for a configuration. In order to facilitate observation, Fig. 3(b) is rotated 90 degrees. As we can see, all the four methods perform poor. But SSL and SSBMI perform better than the other two again. On the whole, FEC still does not perform better than PSA in this test.

Note that, it is possible to detect the communities from the networks generated by model $SG(.)$ used in the tests for the unbalanced networks when $P+ > 1 - P-$. In fact, if we turn the signs of the positive (negative) edges of the networks outside the line $P+ = 1-P-$, such as the network generated by $SG(4, 250, 30, 0.8, 0.9, 0.9)$, to negative (positive), then these networks will become very similar to the corresponding networks inside the line such as the network generated by $SG(4, 250, 30, 0.8, 0.1, 0.1)$. And if a method does not assume

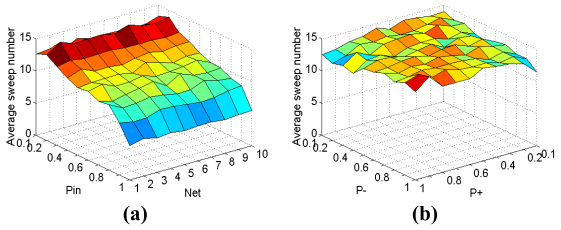



**FIGURE 4.** The sweep number of SSBMI.

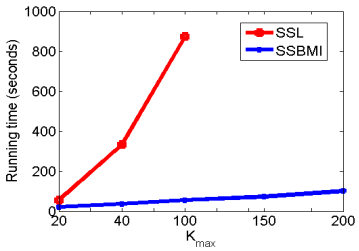



**FIGURE 5.** The run times of SSL and SSBMI.

any biased connection mode in advance, such as assuming that there are more positive edges within the communities than between the communities, then the network outside the line $P+ = 1 - P-$ and the corresponding one inside the line are not essentially different for this method.

Next, we give the experimental results on the sweep number of SSBMI. The experimental result of the balanced networks produced by the aforementioned model is given in Fig. 4(a). Each point over the 3D surface is the average sweep numbers over the 500 restarts of a network. The axis label Net means that each index of this axis denotes a network with a given *Pin*. As we can see, the average sweep numbers are typically about 10 or so. For unbalanced networks, we present the experimental result of the networks produced by $SG(4, 250, 30, 0.2, P-, P+)$ in Fig. 4(b). For each configuration, we select randomly a network from those produced according to the configuration. And each point over the 3D surface in Fig. 4(b) is the average sweep numbers over the 500 restarts of a selected network. As shown in the figure, the average sweep numbers are slightly larger than 10. The average sweep numbers of the networks produced by $SG(4, 250, 30, 0.8, P-, P+)$ are similar to that of those produced by $SG(4, 250, 30, 0.2, P-, P+)$.

In Fig. 5, we show the run times of SSL and SSBMI on the same network generated by $SG(4, 250, 30, 0.8, 0.2, 0.2)$. As shown, with the increasing of $K_{max}$, the advantage of SSBMI over SSL becomes more obvious.

### B. EXPERIMENTS ON REAL-WORLD NETWORKS

We validate the proposed method on five real-wold networks: Slovene parliamentary party network (SPPN), Gahuku-Gama subtribes network (GGSN), and Monastery networks corresponding to three periods (MNT2, MNT3, MNT4). SPPN is a relation network of 10 parties of the Slovene Parliamentary in 1994 [27]. Positive or negative links between two parties denote that they are similar or dissimilar, respectively.

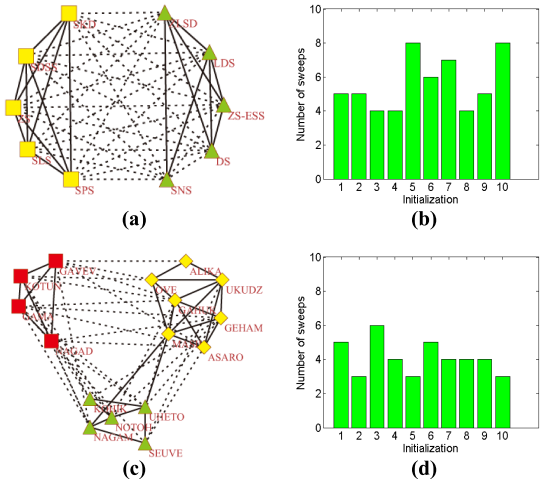



**FIGURE 6.** The communities found by SSBMI from SSPN, GGSN.

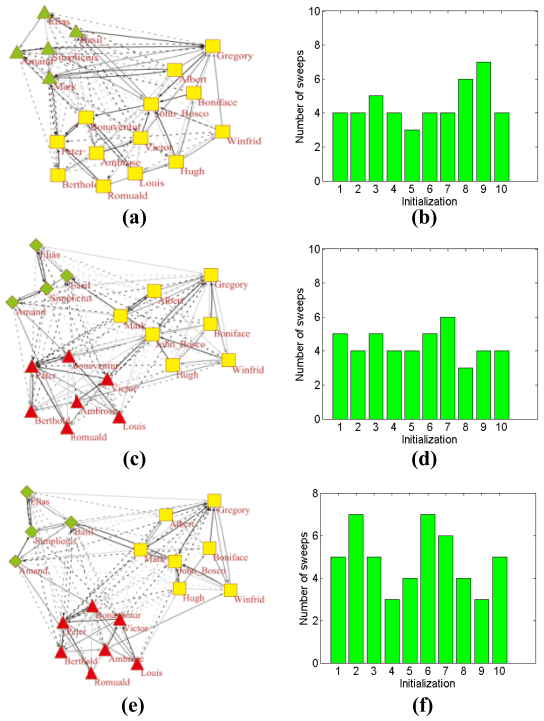



**FIGURE 7.** The communities found by SSBMI from MN.

The 10 parties fall into two communities. GGSN is a political relation network of 16 Gahuku-Gama subtribes in 1954 [28]. Positive or negative links between two subtribes denote that they are allies or enmities, respectively. There are 3 communities in this network. Monastery networks are the social relation networks of 18 young monks at a New England monastery for different periods [8], [22]. Positive or negative links between two monks denote like, praise; or dislike, blame, respectively. As reported, the 18 monks were finally split into 3 groups. These real-world networks include ground truth community structures and are usually regarded as the benchmarks for community detection in signed networks [8], [22]. For all these networks we perform our search with 10 restarts. The values of $K_{max}$ are set to their node numbers, respectively.

Fig. 6(a)(c) show the results of community detection from SPPN, GGSN by SSBMI, respectively. The nodes with the same shape of each network fall in the same community. As we can see, the results are consistent with the ground truths. Fig. 6(b)(d) show the histograms of the sweep numbers over all restarts for SSPN, GGSN, respectively. As we see, for SPPN and GGSN, all the sweep numbers are no more than 10.

Fig. 7(a)(c)(e) show the results of community detection from MNT2, MNT3, MNT4 by SSBMI, respectively. The nodes with the same shape of each network belong to the same community. As we can see, SSBMI decomposes MNT2 into 2 communities. The small community includes some special people. For example, Basil was considered to have a personality problem and Elias and Simplicius were identified as immature, childish, and so on [21]. The result is in line with the human relationships at the beginning of human communication. That is, in the early days of communication, people who have personal problems are easily separated from the remain firstly and form their own community. The results of MNT3 and MNT4 are consistent with the partitions in [21]. Fig. 7(b)(d)(f) show the histograms of the sweep numbers for MNT2, MNT3, and MNT4, respectively. As we see, for all the three networks, all the sweep numbers are less than 10 too.

## IV. CONCLUSION

In this paper, we have proposed a signed networks community detecting method SSBMI which detects the number of communities as well as community assignments in signed networks simultaneously instead of checking every possible model one by one. The ICLex for SSBM is derived and communities are obtained by a greedy search to optimize the value of ICLex for SSBM. Our method has a rigorous probabilistic interpretation and the knowledge on hyper-parameters of our model is not necessary. Extensive experimental studies demonstrate that the proposed method can find the communities in signed networks more accurately than several current methods and more efficiently than the existing method SSL.

The proposed method still needs users to set the value of $K_{max}$ in advance. In the future, we will try to design a new method that does not need users to set the value of $K_{max}$. And the proposed method is designed for detecting communities with hard assignments. That is, each node is belong to one community. In the future, detecting overlapping communities in signed networks will also be our trying direction.

## APPENDIX

### A. INTEGRATED COMPLETE DATA LOG LIKELIHOOD OF SSBMI

$$\begin{aligned} ICLex(Z,K) &= \log P(W,Z|K) \\ &= \log\left\{\frac{\Gamma(\sum_k^K \rho_k^0)\prod_{k=1}^K \Gamma(\rho_k)}{\Gamma(\sum_k^K \rho_k)\prod_{k=1}^K \Gamma(\rho_k^0)}\right\} \\ &\quad + \log\left\{\frac{\Gamma(\sum_h \eta_h^0)\prod_h \Gamma(\eta_h)}{\Gamma(\sum_h \eta_h)\prod_h \Gamma(\eta_h^0)}\right\} \end{aligned}$$

$$+\log\left\{\frac{\Gamma(\sum_h \mu_h^0)\prod_h \Gamma(\mu_h)}{\Gamma(\sum_h \mu_h)\prod_h \Gamma(\mu_h^0)}\right\}$$

where

$$\rho_k = \rho_k^0 + \sum_i \delta(z_i, map_k)$$
$$\eta_h = \eta_h^0 + \sum_{i\sim j} \delta(w_{ij}, h)\delta(z_i, z_j)$$
$$\mu_h = \mu_h^0 + \sum_{i\sim j} \delta(w_{ij}, h)(1-\delta(z_i, z_j))$$

*Proof:*

$$\begin{aligned}\log P(W,Z|K) &= \log\int P(W,Z,\Pi,\Theta,\Omega|K)d\Pi d\Theta d\Omega\\ &= \log\int P(W|Z,\Pi,\Theta)P(\Pi)P(\Theta)d\Pi d\Theta\\ &\quad\times\int P(Z|\Omega)P(\Omega|K)d\Omega\\ &= \log P(W|Z)+\log P(Z|K)\end{aligned}$$

where

$$\begin{aligned}&P(W|Z)\\ &= \int\prod_{i\sim j}\prod_h (\pi_h^{\delta(w_{ij},h)})^{\delta(z_i,z_j)}\frac{\Gamma(\sum_h\eta_h^0)}{\prod_h\Gamma(\eta_h^0)}\prod_h\pi_h^{\eta_h^0-1}d\Pi\\ &\quad\times\int\prod_{i\sim j}\prod_h(\theta_h^{\delta(w_{ij},h)})^{1-\delta(z_i,z_j)}\frac{\Gamma(\sum_h\mu_h^0)}{\prod_h\Gamma(\mu_h^0)}\prod_h\theta_h^{\mu_h^0-1}d\Theta\\ &= \int\frac{\Gamma(\sum_h\eta_h^0)}{\prod_h\Gamma(\eta_h^0)}\prod_h\pi_h^{\sum_{i\sim j}\delta(w_{ij},h)\delta(z_i,z_j)+\eta_h^0-1}d\Pi\\ &\quad\times\int\frac{\Gamma(\sum_h\mu_h^0)}{\prod_h\Gamma(\mu_h^0)}\prod_h\theta_h^{\sum_{i\sim j}\delta(w_{ij},h)(1-\delta(z_i,z_j))+\mu_h^0-1}d\Theta\\ &= \frac{\Gamma(\sum_h\eta_h^0)}{\prod_h\Gamma(\eta_h^0)}\frac{\prod_h\Gamma(\sum_{i\sim j}\delta(w_{ij},h)\delta(z_i,z_j)+\eta_h^0)}{\Gamma(\sum_h\sum_{i\sim j}\delta(w_{ij},h)\delta(z_i,z_j)+\eta_h^0)}\\ &\quad\times\frac{\Gamma(\sum_h\mu_h^0)}{\prod_h\Gamma(\mu_h^0)}\frac{\prod_h\Gamma(\sum_{i\sim j}\delta(w_{ij},h)(1-\delta(z_i,z_j))+\mu_h^0)}{\Gamma(\sum_h\sum_{i\sim j}\delta(w_{ij},h)(1-\delta(z_i,z_j))+\mu_h^0)}\end{aligned}$$

and

$$\begin{aligned}P(Z|K) &= \int\prod_i\prod_k\omega_k^{\delta(z_i,map_k)}\frac{\Gamma(\sum_k\rho_k^0)}{\prod_k\Gamma(\rho_k^0)}\prod_k\omega_k^{\rho_k^0-1}d\Omega\\ &= \int\frac{\Gamma(\sum_k\rho_k^0)}{\prod_k\Gamma(\rho_k^0)}\prod_k\omega_k^{\sum_i\delta(z_i,map_k)+\rho_k^0-1}d\Omega\\ &= \frac{\Gamma(\sum_k\rho_k^0)}{\prod_k\Gamma(\rho_k^0)}\frac{\prod_k\Gamma(\sum_i\delta(z_i,map_k)+\rho_k^0)}{\Gamma(\sum_k(\sum_i\delta(z_i,map_k)+\rho_k^0))}\end{aligned}$$

Let

$$\rho_k = \rho_k^0 + \sum_i \delta(z_i, map_k)$$
$$\eta_h = \eta_h^0 + \sum_{i\sim j} \delta(w_{ij}, h)\delta(z_i, z_j)$$
$$\mu_h = \mu_h^0 + \sum_{i\sim j} \delta(w_{ij}, h)(1-\delta(z_i, z_j))$$

Finally, after taking the logarithms of the above $P(Z|K)$ and $P(W|Z)$ and summing them up, the formula holds.

$$\begin{aligned}ICLex(Z,K) &= \log\left\{\frac{\Gamma(\sum_k^K\rho_k^0)\prod_{k=1}^K\Gamma(\rho_k)}{\Gamma(\sum_k^K\rho_k)\prod_{k=1}^K\Gamma(\rho_k^0)}\right\}\\ &\quad+\log\left\{\frac{\Gamma(\sum_h\eta_h^0)\prod_h\Gamma(\eta_h)}{\Gamma(\sum_h\eta_h)\prod_h\Gamma(\eta_h^0)}\right\}\\ &\quad+\log\left\{\frac{\Gamma(\sum_h\mu_h^0)\prod_h\Gamma(\mu_h)}{\Gamma(\sum_h\mu_h)\prod_h\Gamma(\mu_h^0)}\right\}\end{aligned}$$

### B. MOVING NODE j FROM THE pth COMMUNITY TO THE qth COMMUNITY

1. In the case that the $p$th community is not empty after moving.

$$\begin{aligned}&ICLex^{try}-ICLex\\ &= \log\left\{\frac{\rho_q}{\rho_p-1}\right\}+\log\left\{\frac{\prod_h\Gamma(\eta_h^{try})}{\Gamma(\sum_h\eta_h{}^{try})}\frac{\Gamma(\sum_h\eta_h)}{\prod_h\Gamma(\eta_h)}\right\}\\ &\quad+\log\left\{\frac{\prod_h\Gamma(\mu_h^{try})}{\Gamma(\sum_h\mu_h{}^{try})}\frac{\Gamma(\sum_h\mu_h)}{\prod_h\Gamma(\mu_h)}\right\}\end{aligned}$$

*Proof:*

$$\begin{aligned}&ICLex^{try}-ICLex\\ &= \log P(W,Z^{try}|K^{try})-\log P(W,Z|K)\\ &= \log\left\{\frac{\Gamma(\sum_k^K\rho_k^0)\prod_{k=1}^K\Gamma(\rho_k^{try})}{\Gamma(\sum_k^K\rho_k^{try})\prod_{k=1}^K\Gamma(\rho_k^0)}\frac{\Gamma(\sum_k^K\rho_k)\prod_{k=1}^K\Gamma(\rho_k^0)}{\Gamma(\sum_k^K\rho_k^0)\prod_{k=1}^K\Gamma(\rho_k)}\right\}\\ &\quad+\log\left\{\frac{\Gamma(\sum_h\eta_h^0)\prod_h\Gamma(\eta_h^{try})}{\Gamma(\sum_h\eta_h{}^{try})\prod_h\Gamma(\eta_h^0)}\frac{\Gamma(\sum_h\eta_h)\prod_h\Gamma(\eta_h^0)}{\Gamma(\sum_h\eta_h^0)\prod_h\Gamma(\eta_h)}\right\}\\ &\quad+\log\left\{\frac{\Gamma(\sum_h\mu_h^0)\prod_h\Gamma(\mu_h^{try})}{\Gamma(\sum_h\mu_h{}^{try})\prod_h\Gamma(\mu_h^0)}\frac{\Gamma(\sum_h\mu_h)\prod_h\Gamma(\mu_h^0)}{\Gamma(\sum_h\mu_h^0)\prod_h\Gamma(\mu_h)}\right\}\\ &= \log\left\{\frac{\prod_{k=1}^K\Gamma(\rho_k^{try})}{\prod_{k=1}^K\Gamma(\rho_k)}\right\}+\log\left\{\frac{\prod_h\Gamma(\eta_h^{try})}{\Gamma(\sum_h\eta_h{}^{try})}\frac{\Gamma(\sum_h\eta_h)}{\prod_h\Gamma(\eta_h)}\right\}\\ &\quad+\log\left\{\frac{\prod_h\Gamma(\mu_h^{try})}{\Gamma(\sum_h\mu_h{}^{try})}\frac{\Gamma(\sum_h\mu_h)}{\prod_h\Gamma(\mu_h)}\right\}\\ &= \log\left\{\frac{\Gamma(\rho_p-1)\Gamma(\rho_q+1)}{\Gamma(\rho_p)\Gamma(\rho_q)}\right\}\\ &\quad+\log\left\{\frac{\prod_h\Gamma(\eta_h^{try})}{\Gamma(\sum_h\eta_h{}^{try})}\frac{\Gamma(\sum_h\eta_h)}{\prod_h\Gamma(\eta_h)}\right\}\\ &\quad+\log\left\{\frac{\prod_h\Gamma(\mu_h^{try})}{\Gamma(\sum_h\mu_h{}^{try})}\frac{\Gamma(\sum_h\mu_h)}{\prod_h\Gamma(\mu_h)}\right\}\\ &= \log\left\{\frac{\rho_q}{\rho_p-1}\right\}+\log\left\{\frac{\prod_h\Gamma(\eta_h^{try})}{\Gamma(\sum_h\eta_h{}^{try})}\frac{\Gamma(\sum_h\eta_h)}{\prod_h\Gamma(\eta_h)}\right\}\end{aligned}$$

$$+\log\left\{\frac{\prod_h \Gamma(\mu_h^{try})}{\Gamma(\sum_h \mu_h{}^{try})}\frac{\Gamma(\sum_h \mu_h)}{\prod_h \Gamma(\mu_h)}\right\}$$

2. In the case that the $p$th community is empty after moving.

$$\begin{aligned}
&ICLex^{try} - ICLex \\
&\quad = \log\left\{\frac{\rho_q}{\rho^0}\frac{\Gamma((K-1)\rho^0)}{\Gamma(K\rho^0)}\frac{\Gamma(K\rho^0+n)}{\Gamma((K-1)\rho^0+n)}\right\} \\
&\qquad +\log\left\{\frac{\prod_h \Gamma(\eta_h^{try})}{\Gamma(\sum_h \eta_h{}^{try})}\frac{\Gamma(\sum_h \eta_h)}{\prod_h \Gamma(\eta_h)}\right\} \\
&\qquad +\log\left\{\frac{\prod_h \Gamma(\mu_h^{try})}{\Gamma(\sum_h \mu_h{}^{try})}\frac{\Gamma(\sum_h \mu_h)}{\prod_h \Gamma(\mu_h)}\right\}
\end{aligned}$$

*Proof:*

$$\begin{aligned}
&ICLex^{try} - ICLex \\
&= \log P(W, Z^{try}|K^{try}) - \log P(W, Z|K) \\
&= \log\left\{\frac{\Gamma(\sum_{k\neq p}^K \rho_k^0)\prod_{k\neq p}^K \Gamma(\rho_k^{try})}{\Gamma(\sum_{k\neq p}^K \rho_k^{try})\prod_{k\neq p}^K \Gamma(\rho_k^0)}\frac{\Gamma(\sum_k^K \rho_k)\prod_{k=1}^K \Gamma(\rho_k^0)}{\Gamma(\sum_k^K \rho_k^0)\prod_{k=1}^K \Gamma(\rho_k)}\right\} \\
&\quad +\log\left\{\frac{\prod_h \Gamma(\eta_h^{try})}{\Gamma(\sum_h \eta_h{}^{try})}\frac{\Gamma(\sum_h \eta_h)}{\prod_h \Gamma(\eta_h)}\right\} \\
&\quad +\log\left\{\frac{\prod_h \Gamma(\mu_h^{try})}{\Gamma(\sum_h \mu_h{}^{try})}\frac{\Gamma(\sum_h \mu_h)}{\prod_h \Gamma(\mu_h)}\right\} \\
&= \log\left\{\frac{\Gamma((K-1)\rho^0)\Gamma(\rho_q+1)}{\Gamma((K-1)\rho^0+n)}\frac{\Gamma(K\rho^0+n)\Gamma(\rho_p^0)}{\Gamma(K\rho^0)\Gamma(\rho_p^0+1)\Gamma(\rho_q)}\right\} \\
&\quad +\log\left\{\frac{\prod_h \Gamma(\eta_h^{try})}{\Gamma(\sum_h \eta_h{}^{try})}\frac{\Gamma(\sum_h \eta_h)}{\prod_h \Gamma(\eta_h)}\right\} \\
&\quad +\log\left\{\frac{\prod_h \Gamma(\mu_h^{try})}{\Gamma(\sum_h \mu_h{}^{try})}\frac{\Gamma(\sum_h \mu_h)}{\prod_h \Gamma(\mu_h)}\right\} \\
&= \log\left\{\frac{\rho_q}{\rho^0}\frac{\Gamma((K-1)\rho^0)}{\Gamma(K\rho^0)}\frac{\Gamma(K\rho^0+n)}{\Gamma((K-1)\rho^0+n)}\right\} \\
&\quad +\log\left\{\frac{\prod_h \Gamma(\eta_h^{try})}{\Gamma(\sum_h \eta_h{}^{try})}\frac{\Gamma(\sum_h \eta_h)}{\prod_h \Gamma(\eta_h)}\right\} \\
&\quad +\log\left\{\frac{\prod_h \Gamma(\mu_h^{try})}{\Gamma(\sum_h \mu_h{}^{try})}\frac{\Gamma(\sum_h \mu_h)}{\prod_h \Gamma(\mu_h)}\right\}
\end{aligned}$$

## REFERENCES


[1] R. Pastor-Satorras and A. Vespignani, "Measuring the global Internet," in *Evolution and Structure of the Internet: A Statistical Physics Approach*. New York, NY, USA: CUP, 2007, pp. 19–34.

[2] M. Pascual and J. A. Dunne, "From small to large ecological networks in a dynamic world," in *Ecological Networks: Linking Structure to Dynamics in Food Webs*. New York, NY, USA: OUP, 2006, pp. 3–27.

[3] R. Guimerà, S. Mossa, A. Turtschi, and L. A. N. Amaral, "The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles," *Proc. Nat. Acad. Sci. USA*, vol. 102, no. 22, pp. 7794–7799, 2005.

[4] M. Rubinov and O. Sporns, "Complex network measures of brain connectivity: Uses and interpretations," *NeuroImage*, vol. 52, no. 3, pp. 1059–1069, Sep. 2010.

[5] J. Scott and P. J. Carrington, "Social network analysis: An introduction," in *The SAGE Handbook of Social Network Analysis*. London, U.K.: Sage, 2011, pp. 11–25.

[6] D. Borsboom and A. O. J. Cramer, "Network analysis: An integrative approach to the structure of psychopathology," *Annu. Rev. Clin. Psychol.*, vol. 9, pp. 91–121, Mar. 2013.

[7] M. E. J. Newman and G. Reinert, "Estimating the number of communities in a network," *Phys. Rev. Lett.*, vol. 117, no. 7 p. 078301, Aug. 2016.

[8] B. Yang, X. Zhao, and X. Liu, "Bayesian approach to modeling and detecting communities in signed network," in *Proc. AAAI*, Austin, TX, USA, 2015, pp. 1952–1958.

[9] D. Cartwright and F. Harary, "Structural balance: A generalization of Heider's theory," *Psychol. Rev.*, vol. 63, no. 5, pp. 277–293, 1956.

[10] J. A. Davis, "Clustering and structural balance in graphs," *Hum. Relations*, vol. 20, no. 2, pp. 181–187, 1967.

[11] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 12, pp. 7821–7826, Apr. 2002.

[12] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 6, p. 066133, Jun. 2004.

[13] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, Jun. 2005.

[14] C. Pizzuti, "GA-Net: A genetic algorithm for community detection in social networks," in *Parallel Problem Solving From Nature*, Dortmund, Germany, 2008, pp. 1081–1090.

[15] P. W. Holland and S. Leinhardt, "An exponential family of probability distributions for directed graphs," *J. Amer. Stat. Assoc.*, vol. 76, no. 373, pp. 33–50, Mar. 1981.

[16] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," *J. Mach. Learn. Res.*, vol. 9, pp. 1981–2014, Sep. 2008.

[17] B. Karrer and M. E. J. Newman, "Stochastic blockmodels and community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 83, p. 016107, Jan. 2011.

[18] T. Yang, Y. Chi, S. Zhu, Y. Gong, and R. Jin, "Detecting communities and their evolutions in dynamic social networks—A Bayesian approach," *Mach. Learn.*, vol. 82, no. 2, pp. 157–189, Feb. 2011.

[19] E. Côme and P. Latouche, "Model selection and clustering in stochastic block models based on the exact integrated complete data likelihood," *Stat. Model.*, vol. 15, no. 6, pp. 564–589, Mar. 2015.

[20] J. Wyse, N. Friel, and P. Latouche, "Inferring structure in bipartite networks using the latent blockmodel and exact ICL," *Netw. Sci.*, vol. 5, no. 1, pp. 45–69, Feb. 2017.

[21] P. Doreian and A. Mrvar, "A partitioning approach to structural balance," *Social Netw.*, vol. 18, no. 2, pp. 149–168, Apr. 1996.

[22] B. Yang, W. Cheung, and J. Liu, "Community mining from signed social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 10, pp. 1333–1348, Oct. 2007.

[23] V. A. Traag and J. Bruggeman, "Community detection in networks with positive and negative links," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 80, no. 3, p. 036115, Sep. 2009.

[24] M. Gong, Q. Cai, X. Chen, and L. Ma, "Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 82–97, Feb. 2014.

[25] K. Nowicki and T. A. B. Snijders, "Estimation and prediction for stochastic blockstructures," *J. Amer. Stat. Assoc.*, vol. 96, no. 455, pp. 1077–1087, Dec. 2011.

[26] L. I. Kuncheva and S. T. Hadjitodorov, "Using diversity in cluster ensembles," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, The Hague, The Netherlands, 2004, pp. 1214–1219.

[27] S. Kropivnik and A. Mrvar, "An analysis of the slovene parliamentary parties network," *Develop. Data Anal.*, pp. 209–216, Jan. 1996.

[28] K. E. Read, "Cultures of the central highlands, New Guinea," *Southwestern J. Anthropol.*, vol. 10, no. 1, pp. 1–43, 1954.




**SHUQIU PING** received the M.S. degree from the College of Computer Science and Technology, Jilin University, Changchun, China, in 2009, where he is currently pursuing the Ph.D. degree in computer science and technology. His main research interests include machine learning, data mining, and knowledge science and engineering.



**DAYOU LIU** has been a Professor with the College of Computer Science and Technology, Jilin University, China, since 1990. From 1993 to 2004, he was the Dean of the College of Computer Science and Technology, Jilin University. From 2006 to 2011, he was the Dean of the Faulty of Informatics, Jilin University. He has authored eight books and published more than 300 articles which have more than 4000 citations by other colleagues. His research interests include artificial intelligence, knowledge engineering, statistical learning, data mining, and expert systems.

Prof. Liu is the Distinguished Member of the China Computer Federation (CCF), the Honorary Chair of the Computer Federation of Jilin province of China, and the Honorary Committee Member of the CCF Technical Committee on artificial intelligence and pattern recognition. He received the State Scientific and Technological Progress Award of China, in 2006.



**BO YANG** received the B.Sc., M.S., and Ph.D. degrees in computer science from Jilin University, in 1997, 2000, and 2003, respectively, where he is currently a Professor with the College of Computer Science and Technology. He is currently the Director of the Key Laboratory of Symbolic Computation and Knowledge Engineer, Ministry of Education, China. He ever visited several universities and labs as a Research Fellow or Visiting Scholar, including the Computer Science Department with Hong Kong Baptist University, from 2005 to 2006, with Warwick University, from 2006 to 2007, with George Mason University, from 2014 to 2015, and so on. His current research interests include the areas of data mining, complex/social network modeling and analysis, and multi-agent systems. He is currently working on the topics of discovering structural and dynamical patterns from large-scale and time-evolving networks with applications to Web intelligence, recommender systems, and early detection/control of infectious events. He has published over 70 articles on international journals, including the IEEE TKDE, the IEEE TPAMI, the IEEE TCYB, ACM TWEB, DKE, JAAMAS, KBS, and international conferences, including IJCAI, AAAI, ICDM, WI, PAKDD, and ASONAM. He has served as the associated editors and peer reviewers for international journals including Web intelligence and served as the SPC or PC members for international conferences including IJCAI.



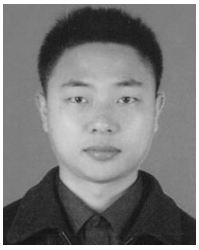
**YUNGANG ZHU** received the Ph.D. degree in computer science from Jilin University, China, in 2012, where he is currently an Associate Professor with the College of Computer Science and Technology.

He had been a visiting Research Fellow or Postdoctoral Fellow with the Vienna University of Technology, Austria, the Dresden University of Technology, Germany, and the University of Trento, Italy. His current research interests include probabilistic graphical models, information fusion, statistical machine learning, and data mining, with applications to knowledge engineering. In recent years, he has published more than ten articles on international journals or conferences.

Dr. Zhu serves as an Associate Editor for the IEEE CANADIAN JOURNAL OF ELECTRICAL AND COMPUTER ENGINEERING. He is the Committee Member of the CCF Computer Applications Technical Committee and the CAAI Intelligent Service Technical Committee. He served with the program committee for several IEEE international conferences.



**HECHANG CHEN** received the M.S. degree from the College of Computer Science and Technology, Jilin University, in 2014, where he is currently pursuing the Ph.D. degree. He was enrolled with the University of Illinois at Chicago as a joint training Ph.D. student, from 2015 to 2016. His current research interests include heterogenous data mining and complex network modeling with applications to computational epidemiology.



**ZHENG WANG** received the M.S. degree from Northeast Electric Power University, Jilin, China, in 2008. He is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Jilin University, Changchun, China. His main research interests include data mining and knowledge science and engineering.