Semi-supervised Stochastic Blockmodel for Structure Analysis of Signed Networks

Xueyan Liu[a,b], Wenzhuo Song[a,b], Katarzyna Musial[c], Xuehua Zhao[d], Wanli Zuo[a,b], Bo Yang[a,b,*]

[a] *School of Computer Science and Technology, Jilin University, Changchun, Jilin, China, 130012*

[b] *Key Laboratory of Symbolic Computation and Knowledge Engineer (Jilin University), Ministry of Education, Changchun, Jilin, China, 130012*

[c] *Advanced Analytics Institute, School of Computer Science, Faculty of Engineering and IT, University of Technology Sydney, NSW, Australia, 2007*

[d] *School of Digital Media, Shenzhen Institute of Information Technology, Shenzhen, China, 518172*

Abstract

Finding hidden structural patterns is a critical problem for all types of networks, including signed networks. Among all of the methods for structural analysis of complex network, stochastic blockmodel (SBM) is an important research tool because it is flexible and can generate networks with many different types of structures. However, most existing SBM learning methods for signed networks are unsupervised, leading to poor performance in terms of finding hidden structural patterns, especially when handling noisy and sparse networks. Learning SBM in a semi-supervised way is a promising avenue for overcoming the above difficulty. In this type of model, a small number of labelled nodes and a large number of unlabelled nodes, coupled with their network structures, are simultaneously used to train SBM. We propose a novel semi-supervised signed stochastic blockmodel and its learning algorithm based on variational Bayesian inference, with the goal of discovering both assortative (the nodes connect more densely in same clusters than

[*] Corresponding author.

Email addresses: Xueyanl17@mails.jlu.edu.cn (Xueyan Liu), songwz17@mails.jlu.edu.cn (Wenzhuo Song), Katarzyna.Musial-Gabrys@uts.edu.au (Katarzyna Musial), lcrlc@sina.com (Xuehua Zhao), zuowl@jlu.edu.cn (Wanli Zuo), ybo@jlu.edu.cn (Bo Yang)

that in different clusters) and disassortative (the nodes link more sparsely in same clusters than that in different clusters) structures from signed networks. The proposed model is validated through a number of experiments wherein it compared with the state-of-the-art methods using both synthetic and real-world data. The carefully designed tests, allowing to account for different scenarios, show our method outperforms other approaches existing in this space. It is especially relevant in the case of noisy and sparse networks as they constitute the majority of the real-world networks.

Keywords: network data mining, signed networks, semi-supervised learning, stochastic blockmodel

1. Introduction

Signed networks, which denote the positive and negative relationships between individuals, exist in various fields, including biology, sociology, information science [1, 2]. For example, the two countries can be allies or not when it comes to international relations. A person may like or dislike others in social activities. In signed networks, positive links represent “cooperative” or “like” and negative links denote “hostile” or “dislike”. Compared with unsigned networks, containing just one type of relationship, signed networks are capable of representing more information.

Community or multipartite structures detection is a particularly important task for signed networks because it aids us to understand the hidden patterns or the rules of the networks. Communities and multipartite structures [3] are both defined as groups, but the definition of a group itself is substantially different in both cases. For communities, the nodes in the same groups link densely and between the different groups the links are sparse. For example, scholars from the same field are more likely to be connected as opposed to scholars coming from different research areas. They can cooperate with each other or, quite opposite, compete with each other. For multipartite structures, if the nodes are from different types or groups, they connect densely and otherwise sparsely. For instance, users make positive or negative comments about the product, while a user rarely connects to other users in user-product rating networks. Detecting these two types of structures is important from the perspective of analysing, understanding, and forecasting the function, the hidden regularity, and the evolution of the signed networks.

In recent years, many methods have appeared for detecting hidden structural patterns in signed networks, including methods based on random walk [4, 5], methods based on social balance theory [6, 7, 8], spectral clustering algorithms [9, 10], and generative methods [11, 12, 13, 14]. However, all of them are unsupervised, i.e., they only focus on topology information, ignoring other potentially available information. Thus, it is hard for them to deal with noisy and sparse signed networks, which commonly exist in the real world. Semi-supervised learning methods that utilize side information, such as partial node labels or pair-wise constraints, are more suitable to handle complicated networks than unsupervised methods [15]. Nowadays, many semi-supervised learning methods have been proposed for network analysis [16, 17, 18, 19, 20]. However, these methods are only designed for unsigned networks. Therefore, this work raises a more complex question: *Can we develop a more flexible model for semi-supervised learning that can detect both community and multipartite structures in signed networks?*

To address the above problem, we propose a novel semi-supervised signed stochastic blockmodel (S^{4}BM) as well as an effective learning algorithm (S^{4}BL) in this paper. We choose to work with the stochastic blockmodel (SBM) as it is a powerful tool to discover and characterize both the communities and multipartite structures in networks (for details, please see Section 3 on Model and Method). Proposed S^{4}BM and S^{4}BL can accurately detect communities and multipartite structures in signed networks using partially available node labels under the semi-supervised learning technique. Specifically, S^{4}BM assumes that few labels of the nodes are known, and it introduces a parameter to describe the relations between the nodes' blocks and the labels. Thus, the known labels will aid to infer the hidden blocks. Besides, it uses two 3-dimension vectors to characterize the probabilities of existing positive, negative, and nonexistent links between two nodes that are from the same blocks and different blocks. Also, these two vectors depict the types of hidden structures. In this way, S^{4}BM combines the label information and the sign information.

In summary, the main contributions of this work are as follows:

(1) **A novel generative semi-supervised signed stochastic blockmodel (S^{4}BM)**. The current semi-supervised methods can handle only unsigned networks. Meanwhile, the methods for analyzing signed networks can only use topology information. Proposed S^{4}BM is capable of utilizing heterogeneous information, i.e., both signed information and label information. Thus, it can deal with a more complex situation, such as noisy and sparse

networks. To the best of our knowledge, S^{4}BM is the first semi-supervised model for signed networks.

(2) Based on variational Bayesian inference, **an efficient semi-supervised learning algorithm (S^{4}BL) is proposed to estimate the parameters and latent variables of the proposed model**.

(3) **Extensive validations and comparisons are performed** on both synthetic and real-world data sets to test the efficacy of the proposed model and algorithm.

The organization of the rest paper is as follows. In section 2, the state-of-the-art methods for signed networks mining as well as semi-supervised learning approaches for complex networks are summarized and analyzed. In section 3, we present the semi-supervised signed stochastic blockmodel and its learning method. Section 4 tests the performance of the proposed model and learning algorithm on synthetic and real-world datasets. Finally, Section 5 summarizes the proposed work.

2. Related Work

2.1. Structural Analysis of Signed Networks

Recently, many methods have been developed for structural analysis of signed networks. Methods based on random walk concept are extended from the stochastic process for unsigned networks. For example, Yang *et al.* assumed that the agents walk only on positive links for finding communities and then used both positive and negative links to compute a cutoff for extracting communities [4]. Zhou *et al.* assumed that the agents walk on positive links with a higher probability than on negative links [5]. Methods based on social balance theory assume that there are more intra-community positive links and inter-community negative links in a signed network. For instance, Traag *et al.* [6] and Anchuri *et al.* [7] combined together the social balance theory with the modularity function for detecting communities in signed networks. Shen and Chung [8] first learned node embeddings by using stacked auto-encoder and a constraint in terms of structural balance theory, and then used K-means to divide embeddings for signed network clustering. Spectral clustering methods constructed signed Laplacian by integrating two Laplacians of the networks containing only the positive relationships or negative relationships in the original signed networks. Then, they calculated and partitioned the eigenvectors, which correspond to the smallest K (i.e., the number of clusters) eigenvalues of signed Laplacian. For example, Chiang *et al.* [9] and Mercado *et al.* [10] combined the Laplacians using arithmetic mean and geometric mean, respectively. The methods discussed above are discriminative ones, which need to predefine the heuristic rules and the objective functions. The performance of such methods is greatly affected by the quality of the predefined rules or functions for detecting clusters, which are hard to designed manually well due to insufficient prior knowledge about structures.

Unlike the discriminative methods, the generative methods can model the generative process of the signed networks with the hidden structures. And the structures can be discovered by fitting the generative model to a given network and estimating the parameters of the generative process. In 2014 and 2015, Jiang and Chen *et al.* proposed signed probabilistic mixture models to detect overlapping communities from signed networks [11, 12]. Yang *et al.* presented signed SBM (SSBM), which can find communities or multipartite structures in signed networks in 2017 [13]. In 2018, Zhao *et al.* improved SSBM for discovering more types of structures, including not only communities or multipartite structures but also outliers, hubs, and hybrid structures, from signed networks [14]. All the above methods only consider the topology information, leading to unsatisfied performance for noisy and sparse signed network clustering. However, in the real world, most of the signed networks have noisy links and are sparse [21, 13]. Thus, it is vital to introduce the semi-supervised technique, which uses some known node labels for network mining tasks.

2.2. Semi-supervised Learning for Complex Networks Clustering

Semi-supervised settings are common in many real scenarios. For example, node labels, which contain ground truth information for the network, are often partially available. Supplementary information such as node labels or pair-wise constraints can improve the accuracy of clustering [15, 17]. According to the type of supplementary information, the existing semi-supervised methods for detecting structural patterns fall into two categories: label-based methods [15, 22, 19], which use specific labels of some nodes, and pair-wise constraints based methods [23, 17, 24], which require information about whether two nodes are in the same cluster or not. It is important to note that these existing semi-supervised learning techniques focus only on the discriminative methods for community detection in unsigned networks.

Some studies have investigated side information-oriented SBMs. For example, Moore *et al.* proposed active learning for SBM based on a Gibbs sampling method [25]. Peel proposed a supervised SBM to use available node labels [26]. Zhang *et al.* studied the phase transitions of sparse networks clustering with a fraction of known labels [16]. Mossel *et al.* explored how side information affects the performance of the belief propagation algorithm on sparse networks [18]. Ganij *et al.* transferred the partial label information to the constraint and then they added it to the existing SBM algorithms [20]. However, the above existing semi-supervised learning methods are designed only for unsigned networks. These methods can not be directly used for signed networks, because they only model whether there is a link or not between two nodes, ignoring the type of relationship.

3. Model and Method

In this section, we propose a semi-supervised signed stochastic blockmodel and a variational Bayes learning algorithm for parameter estimation. To give background information for the developed concepts, we start the section with a brief introduction of what stochastic blockmodel (SBM) is.

3.1. Stochastic Blockmodel Concept

The stochastic blockmodel (SBM) is a powerful tool to discover and characterize both the communities and multipartite structures in networks. SBM assumes that a network consists of several groups (also known as blocks), and the nodes in the same blocks have similar linkage patterns. Mathematically, SBM [27] is defined as a tuple $X = (K, \Omega, \Pi)$. K determines the number of blocks in the network. Ω is a K-dimension vector, and each element denotes the probability of a node belonging to a specific block. Π is a $K \times K$ matrix, and its element π_{kl} refers to the link probability of arbitrary two nodes from block k and block l, respectively. Different values of Π depict different structures contained in the networks. For example, we can describe a network containing several communities by a specific Π in which the main diagonal entries are higher than the off-diagonal entries. Similarly, a network containing multipartite structures can be characterized as that the main diagonal entries of Π are lower than off-diagonal entries. Based on SBM, we can generate a network if we know its parameters by the following steps: a) assign each node to one of K blocks according to the assignment probability Ω; b) generate links between two nodes according to their blocks and link probability Π. Also, we can assume that an observed network is generated by an

SBM and then fit the SBM to it to infer the parameters which characterize its generate process.

3.2. Semi-supervised Signed Stochastic Blockmodel (S^{4}BM)

A signed network N containing n nodes, which are partially labelled by integers ranged from 1 to M, can be denoted by $n \times n$ adjacency matrix A and $n \times M$ label matrix C. For the adjacency matrix A, $a_{ij} = 1$ or -1 if there is a positive or negative link between node i and node j; otherwise, $a_{ij} = 0$. For the label matrix C, each row of which is a one-of-M vector, $c_{ic} = 1$ if the label of node i is c; otherwise $c_{ic} = 0$. A is the observed data. If the label of node i is known, then C_i is the observed data; otherwise, it is a latent variable.

We propose S^{4}BM to model a partially labelled signed networks represented as:

$$< N, C >= (K, \Pi, \Theta, \Omega, \alpha). \tag{1}$$

This model assumes that there are K latent blocks in network N, and the nodes in the same blocks have similar linkage patterns. $\Omega = (\omega_1, \omega_2, ..., \omega_K)$ denotes the proportion of nodes in each block or the probability that a node is assigned to one of K blocks. In addition, let an $n \times K$ matrix Z be a latent variable that indicates the assignment relationship between nodes and blocks. Furthermore, z_i is a one-of-K vector. If node i belongs to block k, $z_{ik} = 1$; otherwise, $z_{ik} = 0$. $\Pi = (\pi_1, \pi_{-1}, \pi_0)$ denotes the probability that there is a positive, negative, or nonexistent link between two nodes in the same block. Similarly, $\Theta = (\theta_1, \theta_{-1}, \theta_0)$ denotes the probability that there is a positive, negative, or nonexistent link between two nodes belonging to different blocks. Let an $M \times K$ matrix α denote the mapping relations of labels and blocks, where α_{ck} is higher if label c is more relevant to block k.

In the S^{4}BM model, z_i, a_{ij} and C_i follow multinomial distributions as shown in Equations (2), (3), and (4), respectively. From Equations (2) and (3), we know that $p(z_{ik} = 1) = \omega_k$, $p(a_{ij} = h) = \pi_h$ for $h \in \{1, -1, 0\}$, if node i and node j are in the same block, otherwise, $p(a_{ij} = h) = \theta_h$.

$$z_i \sim Mul(1, \Omega = \{\omega_1, \omega_2, ..., \omega_K\}), \tag{2}$$

$$\begin{cases} a_{ij} \sim Mul(1, \Pi = \{\pi_1, \pi_{-1}, \pi_0\}) & \text{if } z_{ik}z_{jl}\text{=}1 \text{ and } k = l, \\ a_{ij} \sim Mul(1, \Theta = \{\theta_1, \theta_{-1}, \theta_0\}) & \text{if } z_{ik}z_{jl}\text{=}1 \text{ and } k \neq l, \end{cases} \tag{3}$$

$$C_i \sim Mul(1, p_C = \{p(1|z_i, \alpha), p(2|z_i, \alpha), ..., p(M|z_i, \alpha)\}), \tag{4}$$

For Equation (4), we define $p(c|z_i, \alpha)$ by the following softmax function:

$$p(c|z_i, \alpha) = \exp(\alpha_c z_i^T) / \sum_{c=1}^{M} \exp(\alpha_c z_i^T).$$

$\alpha_c z_i^T$ denotes the level of consistency between the block of node i and the label c. If the value of $\alpha_c z_i^T$ is larger, block and label of node i are more consistent and the label of node i is more likely to be c. In that case, the nodes within the same blocks are more likely to have the same labels. Otherwise, they tend to have different labels.

According to the S^{4}BM, a signed network with labels can be generated according to the following steps:

1. Assign nodes to blocks according to Ω;
2. Generate the label of node i according to p_C;
3. Generate positive and negative links between nodes within the same blocks according to Π;
4. Generate positive and negative links between nodes belonging to different blocks according to Θ.

Accordingly, the log-likelihood of complete data is as follows (please see Appendix A.1 for the detailed derivation):

$$\begin{aligned}\log p(N, C, Z|\alpha, \Pi, \Theta, \Omega) = &\sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik} \log \omega_k + \sum_{i<j}\Big(\sum_{k} z_{ik} z_{jk} \log M(a_{ij}; \Pi) \\ &+ \sum_{l \neq k} z_{ik} z_{jl} \log M(a_{ij}; \Theta)\Big) + \sum_{i=1}^{N}\sum_{c=1}^{M} C_{ic} \log p(c|Z_i, \alpha),\end{aligned} \tag{5}$$

where $M(a_{ij}; \Pi) = \prod_h \pi_h^{\delta(a_{ij}, h)}$, $M(a_{ij}; \Theta) = \prod_h \theta_h^{\delta(a_{ij}, h)}$, and $h \in \{1, -1, 0\}$. $\delta(a_{ij}, h) = 1$ if $a_{ij} = h$; otherwise $\delta(a_{ij}, h) = 0$.

The S^{4}BM can be described in a Bayesian framework. Let the Dirichlet distribution be the prior distribution of the parameters Ω, Π, and Θ, as follows:

$$p(\Omega|\boldsymbol{\rho}^0 = \{\rho_1^0, ..., \rho_K^0\}) = \mathrm{Dir}(\Omega; \boldsymbol{\rho}^0),$$

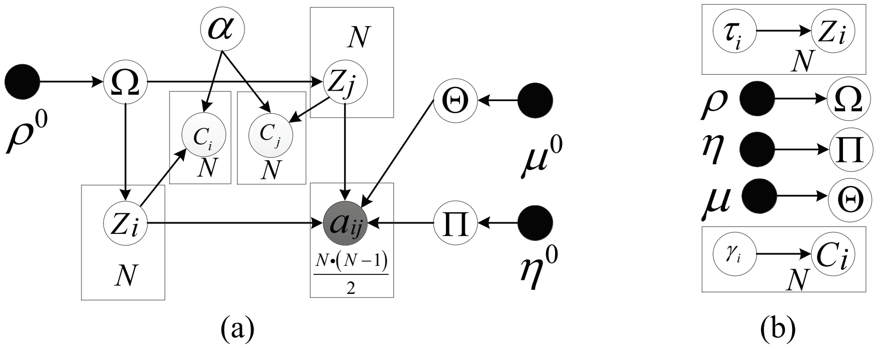


Figure 1: The graphical model of S^{4}BM (a) and corresponding variational distribution (b).

$$p(\Pi|\boldsymbol{\eta}^0 = \{\eta_1^0, \eta_{-1}^0, \eta_0^0\}) = \mathrm{Dir}(\Pi; \boldsymbol{\eta}^0),$$
$$p(\Theta|\boldsymbol{\mu}^0 = \{\mu_1^0, \mu_{-1}^0, \mu_0^0\}) = \mathrm{Dir}(\Theta; \boldsymbol{\mu}^0),$$

where $\forall k \in [1, K]$: ρ_k^0, $\forall h \in \{1, -1, 0\}$: η_h^0, and $\forall h \in \{1, -1, 0\}$: μ_h^0 are super-parameters, which can be interpreted as effective prior pseudo-occupations of respective blocks, prior pseudo-observations of three types of links (positive, negative, and nonexistent) within or between blocks, respectively. In other words, Ω, Π and Θ are regarded as random variables. The graphical model of S^{4}BM is shown in Figure 1 (a).

3.3. S^{4}BM Learning Algorithm (S^{4}BL)

This section presents the S^{4}BM learning algorithm (S^{4}BL). Because the network structure is generated by S^{4}BM according to the defined parameters and the hidden variables Z and C, we must fit the model to the observed network in order to estimate the distributions of parameters and variables. The expectation maximization (EM) method can be used to solve the problem. While the summation in the likelihood involves $(KM)^N$ terms, we adopt the variational Bayes approximate inference to learn the approximate distributions of parameters and hidden variables by introducing $q(Z, C, \Pi, \Theta, \Omega|\tau, \boldsymbol{\eta}, \boldsymbol{\mu}, \boldsymbol{\rho})$. $q(Z, C, \Pi, \Theta, \Omega|\tau, \boldsymbol{\eta}, \boldsymbol{\mu}, \boldsymbol{\rho})$ is a decomposable variational distribution to approximating to the intractable posteriori distribution $p(Z, C, \Pi, \Theta, \Omega|N, \tau, \boldsymbol{\eta}, \boldsymbol{\mu}, \boldsymbol{\rho})$ [28].

Based on Equation (5) and Figure 1 (a), the log-likelihood of N can be written as follows:

$$\log p(N|\alpha,\boldsymbol{\eta}^0,\boldsymbol{\mu}^0,\boldsymbol{\rho}^0) = \log \sum_Z \sum_C \int\int\int p(N,C,Z,\Pi,\Theta,\Omega|\alpha,\boldsymbol{\eta}^0,\boldsymbol{\mu}^0,\boldsymbol{\rho}^0) d_\Pi d_\Theta d_\Omega, \tag{6}$$

where the parameters Ω, Π and Θ are regarded as variables with prior distributions.

Using Jensen's inequality, the log-likelihood in Eq. (6) can be rewritten as follows:

$$\begin{aligned}
&\log p(N|\alpha,\boldsymbol{\eta}^0,\boldsymbol{\mu}^0,\boldsymbol{\rho}^0)\\
&= \log \sum_Z \sum_C \int\int\int q(Z,C,\Pi,\Theta,\Omega|\tau,\boldsymbol{\eta},\boldsymbol{\mu},\boldsymbol{\rho})\\
&\quad \times \left\{ p(N,C,Z,\Pi,\Theta,\Omega|\alpha,\boldsymbol{\eta}^0,\boldsymbol{\mu}^0,\boldsymbol{\rho}^0)/q(Z,C,\Pi,\Theta,\Omega|\tau,\boldsymbol{\eta},\boldsymbol{\mu},\boldsymbol{\rho}) \right\} d_\Pi d_\Theta d_\Omega\\
&\quad \geq \sum_Z \sum_C \int\int\int q(Z,C,\Pi,\Theta,\Omega|\tau,\boldsymbol{\eta},\boldsymbol{\mu},\boldsymbol{\rho})\\
&\quad \times \log \left\{ p(N,C,Z,\Pi,\Theta,\Omega|\alpha,\boldsymbol{\eta}^0,\boldsymbol{\mu}^0,\boldsymbol{\rho}^0)/q(Z,C,\Pi,\Theta,\Omega|\tau,\boldsymbol{\eta},\boldsymbol{\mu},\boldsymbol{\rho}) \right\} d_\Pi d_\Theta d_\Omega\\
&= E_q[\log p(N,C,Z,\Pi,\Theta,\Omega|\alpha,\boldsymbol{\eta}^0,\boldsymbol{\mu}^0,\boldsymbol{\rho}^0)] - E_q[\log q(Z,C,\Pi,\Theta,\Omega|\tau,\boldsymbol{\eta},\boldsymbol{\mu},\boldsymbol{\rho})]\\
&= \boldsymbol{\mathcal{L}}(q(\cdot)),
\end{aligned} \tag{7}$$

where $\boldsymbol{\mathcal{L}}(q(\cdot))$ is the evidence lower bound (ELBO), which can be maximized to learn parameters.

According to the mean-field theory, the variational distribution $q(Z,C,\Pi,\Theta,\Omega|\tau,\gamma,\boldsymbol{\eta},\boldsymbol{\mu},\boldsymbol{\rho})$ can be approximately factorized as follows:

$$q(Z,C,\Pi,\Theta,\Omega|\tau,\gamma,\boldsymbol{\eta},\boldsymbol{\mu},\boldsymbol{\rho}) = q(\Pi|\boldsymbol{\eta})q(\Theta|\boldsymbol{\mu})q(\Omega|\boldsymbol{\rho})\prod_{i=1}^{n} q(z_i|\tau_i)\prod_{i=1}^{n} q(C_i|\gamma_i), \tag{8}$$

where $q(\Pi)$, $q(\Theta)$, and $q(\Omega)$ are Dirichlet distributions with parameters $\boldsymbol{\eta}$, $\boldsymbol{\mu}$, and $\boldsymbol{\rho}$, respectively; $q(z_i)$ and $q(C_i)$ are multinomial distributions with parameters τ and γ, respectively; τ_{ik} denotes the probability of node i belonging to block k; and γ_{ic} is the probability of node i being labelled with c. The variational model of S^{4}BM is shown in Figure 1 (b).

Then, the ELBO can be rewritten as follows by substituting the facterized distribution in Eq. (8) to Eq. (7):

$$\begin{aligned}
&\mathcal{L}(q(\cdot)) \\
&= E_q[\log p(N, C, Z, \Pi, \Theta, \Omega|\alpha, \boldsymbol{\eta}^0, \boldsymbol{\mu}^0, \boldsymbol{\rho}^0)] \\
&\quad - E_q[\log q(Z, C, \Pi, \Theta, \Omega|\tau, \boldsymbol{\eta}, \boldsymbol{\mu}, \boldsymbol{\rho})] \\
&= E_q[\log p(N|Z, \Pi, \Theta)] + E_q[\log p(C|Z, \alpha)] + E_q[\log p(Z|\Omega)] \\
&\quad + E_q[\log p(\Pi|\boldsymbol{\eta}^0)] + E_q[\log p(\Theta|\boldsymbol{\mu}^0)] + E_q[\log p(\Omega|\boldsymbol{\rho}^0)] \\
&\quad - E_q[\log q(Z|\tau)] - E_q[\log q(C|\gamma)] - E_q[\log q(\Pi|\boldsymbol{\eta})] \\
&\quad - E_q[\log q(\Theta|\boldsymbol{\mu})] - E_q[\log q(\Omega|\boldsymbol{\rho})].
\end{aligned} \tag{9}$$

In order to maximize the ELBO, we can optimize $q(\Pi)$, $q(\Theta)$, $q(\Omega)$, $q(Z)$ and $q(C)$ by coordinate descent and α by gradient descent because $\partial\mathcal{L}_{[\alpha_{ck}]}/\partial\alpha_{ck} = 0$ does not produce a closed form solution. In other words, one of the parameters is updated in each iteration while the others are fixed (for details of derivation please see the Appendix A.2).

Updating the parameters of the variational distribution of latent variables Z and C by

$$\begin{aligned}
\tau_{ik} \propto\, & e^{\psi(\rho_k) - \psi(\sum\limits_k \rho_k) + \sum_{c=1}^M \gamma_{ic}(\alpha_{ck} - (\beta(\tau_i^{old})^T)^{-1}\beta_k)} \\
& \times \prod_{j\neq i}^{n} \Big(e^{\tau_{jk} \sum\limits_h \delta(a_{ij}, h)(\psi(\eta_h) - \psi(\sum\limits_h \eta_h))} \\
& \times \prod_{l\neq k} e^{\tau_{jl} \sum\limits_h \delta(a_{ij}, h)(\psi(\mu_h) - \psi(\sum\limits_h \mu_h))} \Big),
\end{aligned} \tag{10}$$

and

$$\gamma_{ic} = e^{\sum_{k=1}^K \alpha_{ck}\tau_{ik} - 1} / \sum_{c=1}^{M} \sum_{k=1}^{K} \tau_{ik} e^{\alpha_{ck}}, \tag{11}$$

where $\beta = [\beta_1, \beta_2, ..., \beta_K], \beta_k = \sum_{c=1}^M \exp(\alpha_{ck})$, and $\psi(\cdot)$ is the digamma function.

We update the hyperparameters of the variational distribution $q(\Omega)$, $q(\Pi)$, and $q(\Theta)$ by

$$\rho_k = \rho_k^0 + \sum_{i=1}^{n} \tau_{ik}, \tag{12}$$

$$\eta_h = \eta_h^0 + \sum_{i<j}^{n} \sum_{k=1}^{K} \tau_{ik} \tau_{jk} \delta(a_{ij}, h), \tag{13}$$

and

$$\mu_h = \mu_h^0 + \sum_{i<j}^{n} \sum_{k \neq l}^{K} \tau_{iq} \tau_{jl} \delta(a_{ij}, h), \tag{14}$$

Finally, α is optimized by gradient descent with the following derivative:

$$\partial \mathcal{L}_{[\alpha_{ck}]} / \partial \alpha_{ck} = \sum_{i=1}^{N} \gamma_{ic} (\tau_{ik} - \tau_{ik} e^{\alpha_{ck}} / \sum_{c=1}^{M} \sum_{k=1}^{K} \tau_{ik} e^{\alpha_{ck}}). \tag{15}$$

The semi-supervised learning algorithm S^{4}BL is presented in Algorithm 1. In S^{4}BL, τ, γ, and α are initialized by sampling from a uniform distribution. After that, τ_i and γ_i are aligned to the given labels if available. $\boldsymbol{\rho^0}$ is initialized to be K dimension vector, wherein each element is 1 and we set $\boldsymbol{\rho} = \boldsymbol{\rho^0}$. The 3-dimension vectors $\boldsymbol{\eta^0}$, $\boldsymbol{\eta}$, $\boldsymbol{\mu^0}$ and $\boldsymbol{\mu}$ are all initialized according to the percentage of positive, negative, and nonexistent links in a network. Similar to our previous work [13], we assume that there are two types of networks when considering the social balance theory: (i) in one type of the network, most of the links follow the theory ($\eta_1^0 > \mu_1^0$ and $\eta_{-1}^0 < \mu_{-1}^0$) and (ii) in another type of the network, most of the links violate the theory ($\eta_1^0 < \mu_1^0$ and $\eta_{-1}^0 > \mu_{-1}^0$). Thus, we set $\boldsymbol{\eta^0} = \boldsymbol{\eta} = (0.6 \times rp, 0.4 \times rn, 0.5 \times ro)$, $\boldsymbol{\mu^0} = \boldsymbol{\mu} = (0.4 \times rp, 0.6 \times rn, 0.5 \times ro)$ for the first type and $\boldsymbol{\eta^0} = \boldsymbol{\eta} = (0.4 \times rp, 0.6 \times rn, 0.5 \times ro)$, $\boldsymbol{\mu^0} = \boldsymbol{\mu} = (0.6 \times rp, 0.4 \times rn, 0.5 \times ro)$ for the second type, where rp, rn, and ro denote the ratios of positive, negative and nonexistent links to the total links in the network. Because we do not know the type of the most of the networks, we run S^{4}BL twice using theses two initialization settings, respectively, and then choose the result with the higher ELBO.

3.4. Time Complexity Analysis

Updating the posteriors of Z and C takes $O(K^2n^2)$ and $O(KM^2n)$ respectively by the **for** loop in line 04-11 from Algorithm 1. The time complexities of updating the posterior of Ω and calculating parameter α by the **for** loop in line 12-17 are $O(Kn)$ and $O(K^2M^2n \cdot t)$ respectively, where t is the number of iterations of gradient descent. It takes $O(Kn^2)$ to update the posteriors of Π and Θ by the **for** loop in line 18-21. In total, the time complexity of

Algorithm 1 S^{4}BL

```
Input: $N, K, C$;
Output: $Z$;
initialize $\tau, \gamma, \mu, \eta, \rho, \alpha$;
Let $S$={$x$|the label of node $x$ is unknown};
repeat
    for node $i \in S$ do
        for $k = 1$ to $K$ do
            update $\tau_{ik}$ according to Equation (10);
        end for
        for $c = 1$ to $M$ do
            update $\gamma_{ic}$ according to Equation (11);
        end for
    end for
    for $k = 1$ to $K$ do
        update $\rho_k$ according to Equation (12);
        for $c = 1$ to $M$ do
            update $\alpha_{ck}$ by gradient descent using Equation (15);
        end for
    end for
    for $h \in \{1, -1, 0\}$ do
        update $\eta_h$ according to Equation (13);
        update $\mu_h$ according to Equation (14);
    end for
until convergence
calculate $Z$ according to $\tau$;
```

S^{4}BL is $O(K^2 n(n + M^2 \cdot t) \cdot T)$, where $K \ll n$, $M \ll n$ and T is the number of iterations of the **repeat** loop until convergence.

4. Experiments

In this section, we first introduce the experimental settings. Then, we evaluate the performance of the S^{4}BL algorithm on synthetic networks. We compare S^{4}BL with signed network clustering approaches by using different numbers of labels to show the effectiveness of S^{4}BL. Also, we test S^{4}BL on networks with varying levels of noise and sparsity to demonstrate the capabilities of S^{4}BL to deal with noisy and sparse networks. After that, we compare S^{4}BL with unsupervised signed network clustering methods and unsigned network clustering methods that can use both link and label information for fairness on real-world networks, since the existing methods for signed network mining can use only link information. Finally, we test the quality of parameter estimation of S^{4}BL.

4.1. Experimental Settings

4.1.1. Compared algorithms

To the best of our knowledge, our method is the first semi-supervised method for both community and multipartite structure mining in signed networks; therefore, we can conduct a comparative analysis with six unsupervised algorithms: SSL (SSBM learning algorithm) [13], PSA (Potts model algorithm) [6], and FEC (finding and extracting communities) [4], VBS (variational Bayes approach for improved SSBM) [14], DNE-SBP (deep network embedding with structural balance preservation) [8], and GM (geometric mean Laplacian) [10]. In addition, we also include a baseline method that only uses labels in the experiments (see Table 1).

It is worth noting that we can extend S^{4}BM to unsigned networks by setting $\pi_{-1} = 0$ and $\theta_{-1} = 0$. Thus, we also compare S^{4}BM with methods which can use label information on unsigned real-world networks to further show the performance of S^{4}BM. In this experiments, we select four methods for unsigned network mining: BLOS (unsupervised block-wise SBM) [29], LP (semi-supervised community detection) [30], SMMB (supervised mixed membership blockmodel) [26], and a baseline method.

Table 1: Compared algorithms

Methods	Rationale
PSA [6]	modularity optimization
FEC [4]	random walk model
SSL [13]	variational Bayes inference
VBS [14]	variational Bayes inference for improved SSBM
DNE-SBP [8]	signed network embedding using stacked auto-encoder and pairwise constraints about structural balance theory
GM [10]	extended spectral clustering using geometric mean Laplacian
Baseline	labeled nodes are assigned to correct blocks and unlabeled nodes are randomly assigned

4.1.2. Evaluation metric

To evaluate the performance of the algorithms, we adopt the normalized mutual information (NMI) [31], which measures the agreement between two partitions. The definition of NMI is as follows:

$$NMI(A,B) = -2\sum_{i=1}^{c_A}\sum_{j=1}^{c_B} m_{ij}\log(\frac{m_{ij}n}{m_{i.}m_{.j}})/\big(\sum_{i=1}^{c_A} m_{i.}\log(\frac{m_{i.}}{n}) + \sum_{j=1}^{c_B} m_{.j}\log(\frac{m_{.j}}{n})\big)$$

where A and B are the real and detected partitions, respectively; c_A and c_B are the numbers of real blocks and detected blocks, respectively; m_{ij} is the number of nodes in both real block i and detected block j; and $m_{i.}$ and $m_{.j}$ denote the numbers of nodes in real block i and detected block j, respectively.

4.2. Validation of Synthetic Networks

In this study, we generate synthetic signed networks using two models: SSBM (signed SBM), which was proposed in our previous work [32, 13], and SLFR (signed LFR), which is an extension of LFR (Lancichinetti, Fortunato and Radicchi) [33]. In the networks generated by SSBM, the distributions of both node degrees and block sizes are uniform; in contrast, in the networks generated by SLFR, these distributions are both power law.

SSBM is described as follows:

$$\begin{aligned} X &= SSBM(K, n, \Pi, \Theta, \Omega) \\ &= SSBM\big(K, n, (\pi_1, \pi_{-1}, \pi_0), (\theta_1, \theta_{-1}, \theta_0), \Omega\big) \end{aligned} \tag{16}$$

where K, Π, Θ and Ω are equally defined in Eq. (1) in Section 3, and n is the number of nodes in a network.

The description of SLFR is as follows:

$$X = SLFR(n, k_{avg}, k_{max}, \lambda_1, \lambda_2, s_{min}, s_{max}, \upsilon, p_{+1}, p_{-1}) \tag{17}$$

where n is the number of nodes; k_{avg} and k_{max} are the average and the maximum degree respectively, of each node; λ_1 and λ_2 are the exponents of the power law distributions of node degrees and block sizes, respectively; s_{min} and s_{max} are the minimum and maximum block size, respectively; υ is the proportion of inter-block links of each node; p_{+1} and p_{-1} are the proportions of intra-block positive links and inter-block negative links, respectively.

Note that both SSBM and SLFR can generate networks with community and multipartite structures by regulating π_0 and θ_0 simultaneously, or υ, respectively. For example, if we set $\pi_0 < \theta_0$ or $\upsilon < 0.5$, we can generate networks with communities, otherwise, we will generate networks with multipartite structures.

4.2.1. Test S^4BL on networks with different numbers of labels

In this part, we generate random signed networks *with high levels of noise* according to the following model settings:

- **SSBM-I** :
  $X = SSBM\big(4, 128, (0.4, 0.4, 0.2), (0.1, 0.1, 0.8), (1/4, 1/4, 1/4, 1/4)\big)$.
- **SSBM-II**:
  $X = SSBM\big(4, 128, (0.1, 0.1, 0.8), (0.4, 0.4, 0.2), (1/4, 1/4, 1/4, 1/4)\big)$.
- **SLFR-I**:
  $X = SLFR(128, 16, 20, 2, 1, 20, 40, 0.3, 0.5, 0.5)$.
- **SLFR-II**:
  $X = SLFR(128, 16, 20, 2, 1, 20, 40, 0.9, 0.5, 0.5)$.

SSBM-I and SLFR-I generate networks with community structures, in which nodes are densely linked in the same block and sparsely linked among different blocks (see Figure 2 (a) and (c)). SSBM-II and SLFR-II generate networks with multipartite structures, in which nodes are sparsely linked in the same block and densely linked among different blocks (see Figure 2 (b)

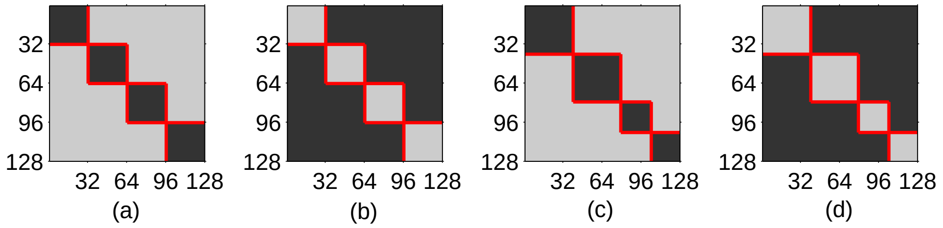


Figure 2: The adjacency matrix of generated networks, where dark and light grey denote, respectively, dense and sparse link distributions. (a) and (b) are networks with uniform distributions, while (c) and (d) are networks with power law distributions. (a) and (c) show community structures, while (b) and (d) show multipartite structures.

and (d)). For the SSBM-I and SSBM-II networks, we set $\pi_1 = \pi_{-1}$ and $\theta_1 = \theta_{-1}$; for the SLFR-I and SLFR-II networks, we set $p_{+1} = p_{-1}$. In other words, we generate as many noisy links as normal links, which makes clustering much more challenging.

For the networks generated by SSBM, in which block size and node degree are uniformly distributed (we denote this kind of network as homogeneous signed networks), we randomly selected the same number of nodes from each block as labelled nodes. The labels of the selected nodes were fed into the S^{4}BL algorithm as priors. In the experiment, nodes were selected according to the following schemes:

I1. Nodes are randomly selected out of two blocks.

I2. Nodes are randomly selected out of four blocks.

Figures 3 (a) and (b) show the performance of the algorithms on the SSBM-I networks with communities. The results indicate that VBS performs best among the six unsupervised algorithms, and achieving 0.71 of the NMI. Fig. 3 (a) shows that S^{4}BL achieves 0.93 of the NMI when 6 labels for each of two blocks are available. This value approaches 1 when 12 labels are available for each block. In other words, S^{4}BL can assign almost all nodes to their correct blocks when 24 out of a total of 128 nodes are labelled as priors. As shown in Figure 3 (b), S^{4}BL can find all blocks when 4 labels are available for each of four blocks. These two figures demonstrate that, compared with unsupervised methods, S^{4}BL can use no more than 20% additional labels to upgrade at almost 0.3 of the NMI.

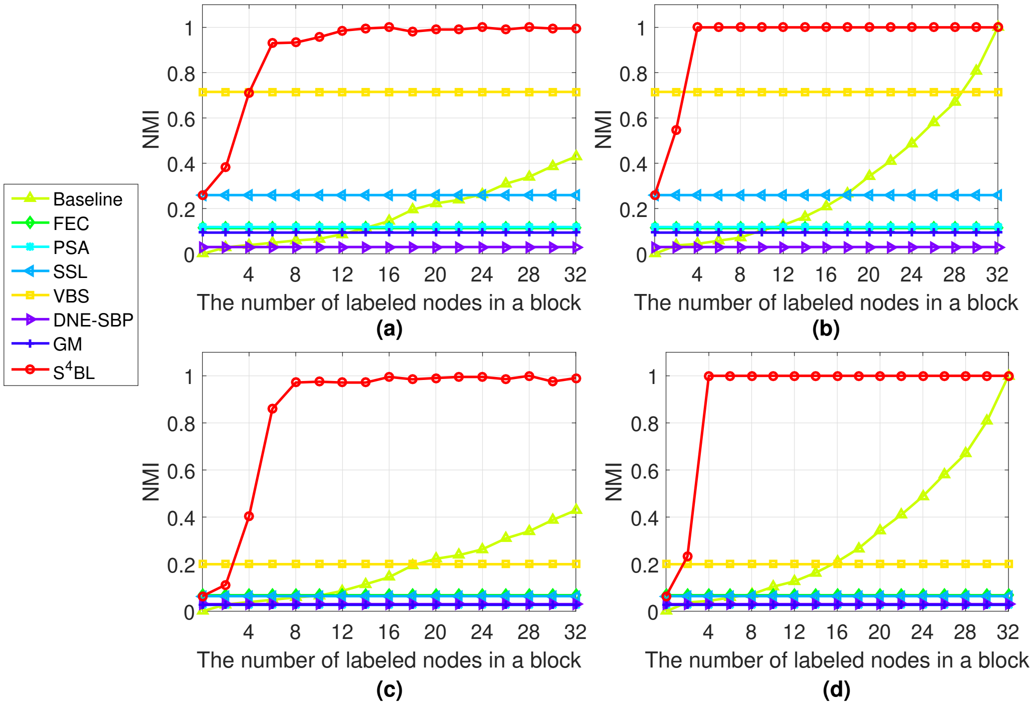


Figure 3: NMI by Baseline, six unsupervised methods, and S^{4}BL (our method) on homogeneous signed networks. (a) and (b) show the results on SSBM-I signed networks with settings I1 and I2, respectively. (c) and (d) show the results on SSBM-II signed networks with settings I1 and I2, respectively. I1 denotes the labeled node are from two of four blocks. I2 denotes the labeled nodes are from each of four blocks. S^{4}BL can improve the NMI a lot with a small amount of labeled nodes.

Figures 3 (c) and (d) depict the performance of the algorithms on the SSBM-II networks with multipartite structures. Multipartite structure detection is more difficult than community detection and most of the unsupervised algorithms like SSL, FEC, PSA, DNE-SBP, and GM perform only slightly better than random assignment. The NMI of VBS, which is the best performing unsupervised method, is only at the level of 0.2. As indicated in Figure 3 (c), however, S^{4}BL can assign almost all nodes to the correct blocks when 8 labels are available for each of two blocks. Figure 3 (d) shows that S^{4}BL can detect all correct communities when 4 labels are provided for each of four blocks. In this case, compared with unsupervised methods, S^{4}BL upgrade more than 0.8 of the NMI with an increase in labels of no more than 15%.

Comparing settings I1 and I2 for S^{4}BL, the most results on setting I2 are slightly better than those on setting I1 when the number of labelled nodes in the network are equal for this two settings, i.e., the number of labelled nodes in a block as shown in Figure 3 (a) (or (c)) is twice of that in Figure 3 (b) (or (d)). It suggests that S^{4}BL performs better when the node labels directly impact each block.

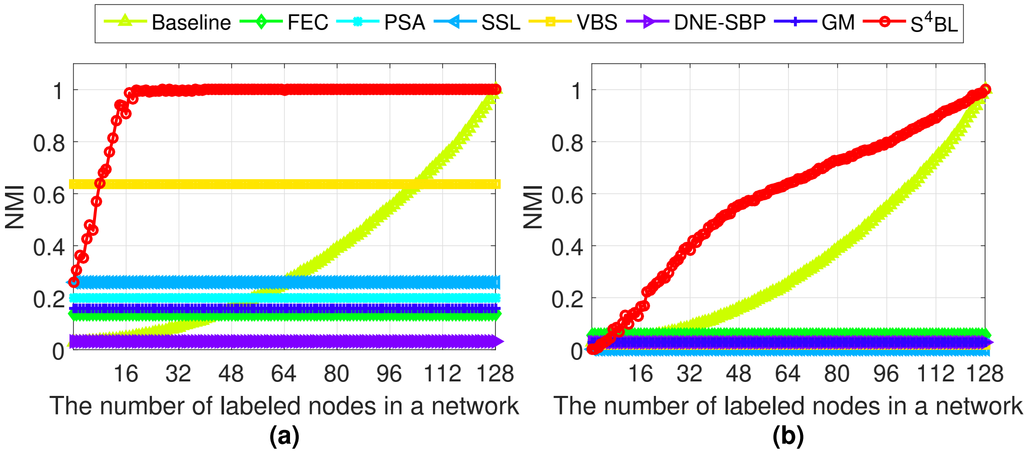


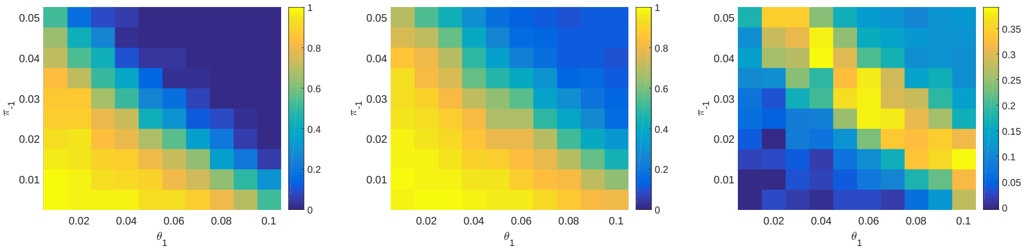
Figure 4: NMI by Baseline, six unsupervised methods and S^{4}BL (our method) on the heterogeneous signed networks. (a) shows the results on networks with communities generated by SLFR-I and (b) shows the results on networks with multipartite structures generated by SLFR-II.

For the networks generated by SLFR, in which block size and node degree are both heterogeneous with power law distributions, we selected nodes with high degrees for labelling. Figure 4 shows the experimental results. Among all unsupervised methods, the NMI of VBS is the largest, with 0.65, when detecting communities from SLFR-I networks as shown in Figure 4 (a). However, S^{4}BM can achieve 1 in terms of NMI by using less than 15% additional labels. From Figure 4 (b), we know that all unsupervised methods perform poorly on SLFR-II networks, for which NMIs are less than 0.06. However, the NMI of S^{4}BL is more than 0.25 if we label 15.6% nodes.

Figures 3 and 4 indicate that the performance of SBM can be significantly improved with the aid of semi-supervised learning mechanism, in which both labelled and unlabelled nodes are simultaneously used, for learning the network structures. In our comparison, S^{4}BL outperformed the unsupervised methods and the baseline, which only use either topological information of unlabelled nodes or prior information of labelled nodes for learning, but not both.

4.2.2. Test S^{4}BL on networks with different levels of noise

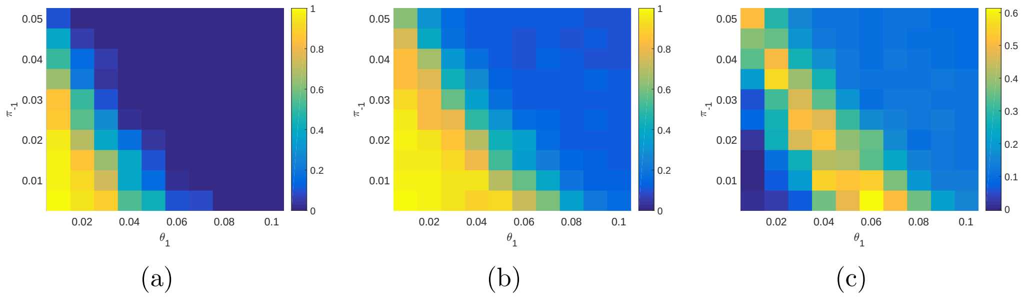
We test S^{4}BL on networks with different levels of noise and fixed sparsity, which means networks with different proportions of negative links in blocks and positive links between different blocks. We use SSBM as shown in Eq. (16) to generate networks by fixing the link sparsity and varying the noise. For generating networks with communities, we set $\pi_0 = 0.8$ and $\theta_0 = 0.9$ and then increasing π_{-1} from 0.01 to 0.1 by 0.01 step and increasing θ_1 from 0.005 to 0.05 by 0.005 step, respectively. For generating networks with multipartite structures, we set $\pi_0 = 0.9$ and $\theta_0 = 0.8$ and then increasing π_{-1} from 0.005 to 0.05 by 0.005 step and increasing θ_1 from 0.01 to 0.1 by 0.01 step, respectively. With π_{-1} and θ_{+1} increasing, the noisy links increase until the noisy links are as many as normal links, and detecting hidden structural in networks becomes more challenging. For each setting, we generated 30 instances of network having characteristics as defined in a given setting.



(a) (b) (c)

Figure 5: The results of SSL (a), i.e., NMI_{SSL}, S4BL (b), i.e., NMI_{S^4BL} and the increased NMI (c), i.e., $NMI_{inc} = NMI_{S^4BL} - NMI_{SSL}$, on the networks with communities under different levels of noise. θ_1 (or π_{-1}) represents the probability of noisy links inter-block (or intra-block).

We test the unsupervised algorithm SSL (which can be regarded as the unsupervised version of S^{4}BL) and our proposed semi-supervised algorithm S^{4}BL with 4 known labels in each group by using above generated networks. First, we apply SSL and S^{4}BL to the generated networks and calculate the NMI of each algorithm. Then, we calculate and report the average NMI of SSL (NMI_{SSL}) and S^{4}BL (NMI_{S^4BL}) for each setting. Finally, we report the increased NMI, i.e., $NMI_{inc} = NMI_{S^4BL} - NMI_{SSL}$, to show the improvement of our method on signed networks with various levels of noise. The larger NMI_{inc}, the more improvement of S^{4}BL.



(a) (b) (c)

Figure 6: The results of SSL (a), i.e., NMI_{SSL}, S4BL (b), i.e., NMI_{S^4BL} and the increased NMI (c), i.e., $NMI_{inc} = NMI_{S^4BL} - NMI_{SSL}$, on the networks with multipartite structures under different levels of noise. θ_1 (or π_{-1}) represents the probability of noisy links inter-block (or intra-block).

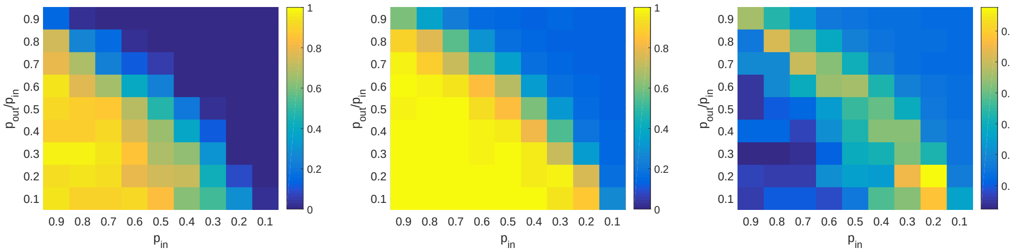
Figures 5 (a) and (b) show the NMIs of SSL and S^{4}BM on networks with communities, respectively, and (c) shows their differences in terms of NMIs. Similarly, Figures 6 (a), (b), and (c) show the NMIs of SSL and S^{4}BM, and their differences on networks with multipartite structures, respectively. θ_1 and π_{-1} are the positive inter-block and negative intra-block link probability respectively.

When the noisy levels are low, i.e., $1/2\theta_1 + \pi_{-1} < 0.045$ in Figure 5 (c) or $\theta_1 + \pi_{-1} < 0.045$ in Figure 6 (c) as shown in the the bottom left of each panel, NMI_{inc} is less than 0.1 , which denotes that there is little room for improvement. This is because that the most of NMIs of SSL are lager than 0.8 as shown in Figure 5 (a) and Figure 6 (a). Thus the unsupervised methods is capable to deal with networks with low level of noise. As the noisy links increase, the NMIs of SSL rapidly decrease to around 0.2 and then almost to 0. However, with the aid of known labels, S4BL performs still well when $1/2\theta_1 + \pi_{-1} < 0.75$ in Figure 5 (b) and $\theta_1 + \pi_{-1} < 0.8$ in Figure 6 (b). In theses cases, $NMI_{inc} > 0.2$ in Figures 5 (c) and 6 (c).

From Figures 5 and 6, we can conclude that S^{4}BL can handle signed networks with high noise better than unsupervised methods in terms of detecting hidden structural patterns.

4.2.3. Test S^{4}BL on networks with different levels of sparsity

In this part, we test S^{4}BL on networks with different levels of sparsity and fixed noise. Similarly, we use SSBM to generate networks by fixing the noise and varying the sparsity. For convenience, we denote $p_{in} = \pi_1 + \pi_{-1}$ and $p_{out} = \theta_1 + \theta_{-1}$. To fix the noise, we set $\pi_{-1} = 0.4 \times p_{in}$ and $\theta_1 = 0.4 \times p_{out}$, in other words, the noisy links account for 40% of all links. The link probability in the blocks is higher than between different blocks in networks with communities, and it is smaller in networks with multipartite structures. Thus, we set $p_{in} > p_{out}$ for generating networks with communities and $p_{in} < p_{out}$ for generating networks with multipartite structures. Specifically, we increase p_{in} from 0.1 to 0.9 by 0.1 step and vary p_{out} from $0.1 \times p_{in}$ to $0.9 \times p_{in}$ by $0.1 \times p_{in}$ step for each p_{in} to generate networks with communities. In the same way, we increase p_{out} from 0.1 to 0.9 by 0.1 step and vary p_{in} from $0.1 \times p_{out}$ to $0.9 \times p_{out}$ by $0.1 \times p_{out}$ step for each p_{out} to generate networks with multipartite structures. For the same ratio of p_{out} to p_{in} or p_{in} to p_{out}, the networks become sparser and sparser and the mining task becomes more and more challenging as p_{in} or p_{out} decreases. We generate 30 networks for each setting.

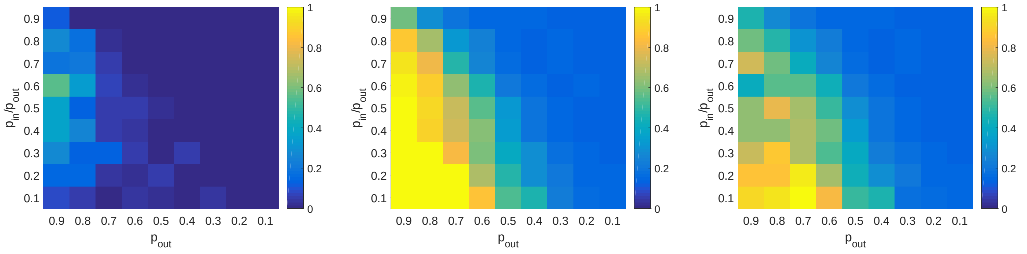


(a) (b) (c)

Figure 7: The results of SSL (a), i.e., NMI_{SSL}, S4BL (b), i.e., NMI_{S^4BL} and the increased NMI (c), i.e., $NMI_{inc} = NMI_{S^4BL} - NMI_{SSL}$, on the networks with communities under different levels of sparsity. $p_{in} = \pi_1 + \pi_{-1}$ and $p_{out} = \theta_1 + \theta_{-1}$ denote the link probability intra-block and inter-block, respectively.

Similarly to Section 4.2.2, We test SSL and S^{4}BL with 4 known labels in each block on the above generated networks. Figures 7 and 8 show the NMIs of SSL (NMI_{SSL}), the NMIs of S^{4}BL (NMI_{s^4BL}), the increased NMIs (NMI_{inc}) on networks with, respectively, communities and multipartite structures. p_{out}/p_{in} in Figure 7 and p_{in}/p_{out} in Figure 8 both determine whether the hidden structure is clear or not in networks. p_{out}/p_{in} or p_{in}/p_{out} is lower, the structure is clearer.

From each row of Figures 7 (a)-(b) and Figures 8 (a)-(b), we know that both SSL and S^{4}BL perform worse and worse with the network becoming



(a) (b) (c)

Figure 8: The results of SSL (a), i.e., NMI_{SSL}, S4BL (b), i.e., NMI_{S^4BL} and the increased NMI (c), i.e., $NMI_{inc} = NMI_{S^4BL} - NMI_{SSL}$, on the networks with multipartite structures under different levels of sparsity. $p_{in} = \pi_1 + \pi_{-1}$ and $p_{out} = \theta_1 + \theta_{-1}$ denote the link probability intra-block and inter-block, respectively.

sparser and sparser. However, the performance of S^{4}BL is superior to SSL in the same network. For example, when the network is sparse ($p_{in} = 0.2$), NMI of SSL is less than 0.3 even if the community is really clear ($p_{out}/p_{in} = 0.1$) in Figure 7 (a). In this case, the NMI of S^{4}BL, however, is larger than 0.85 in Figure 7 (b).

Figure 7 one can note that when links are dense in networks with communities, i.e., $p_{in} \geq 0.7$, the NMIs of SSL are lager than 0.8 if the communities are clear ($p_{out}/p_{in} < 0.5$). When the links are sparse, i.e., $p_{in} < 0.4$, almost NMIs of SSL become less than 0.5 no matter how clear the structures are. However, the NMIs of S^{4}BL are larger than 0.75 when $0.1 < p_{in} < 0.4$ and $p_{out}/p_{in} < 0.3$. The improvement of NMIs between SSL and S^{4}BL, i.e., NMI_{inc}, are larger than 0.4 in this area. From Figure 8, SSL almost can not deal with networks with multipartite structures when the noise links occupy 40% of the total links. However, the NMIs of S^{4}BL are larger than 0.8 when the networks are dense ($p_{out} > 0.6$) and the multipartite structures are clear ($p_{in}/p_{out} < 0.4$).

From Figures 7 and 8, we can conclude that S^{4}BL can deal with networks with sparse links better than unsupervised methods in terms of mining structural patterns in signed networks.

4.3. Validation on the Real-World Networks

4.3.1. Test S^{4}BL on signed networks

We test S^{4}BL using one of the Monastery network [34] and WikiEditor network [35], which both have the ground-truth and are shown to be chal-

Table 2: Statistics for the real-world networks used in the experiments.

Networks	n	e	K	Block Structure
Monastery [34]	18	78	3	community
WikiEditor [35]	20,198	347,218	2	community and bipartite

langing from the perspective of unsupervised methods. Monastery network, which is much more vague in terms of network structure, is the affect relationship among 18 monks at time $T2$. WikiEditor network, which has both community and bipartite structures, denotes the relationships among $20,198$ users who edited the Wikipedia items. Table 2 exhibits the statistics for these two networks, where n, e, K are the numbers of nodes, edges and blocks. For the real-world network testing, we select nodes with high degrees for labelling.

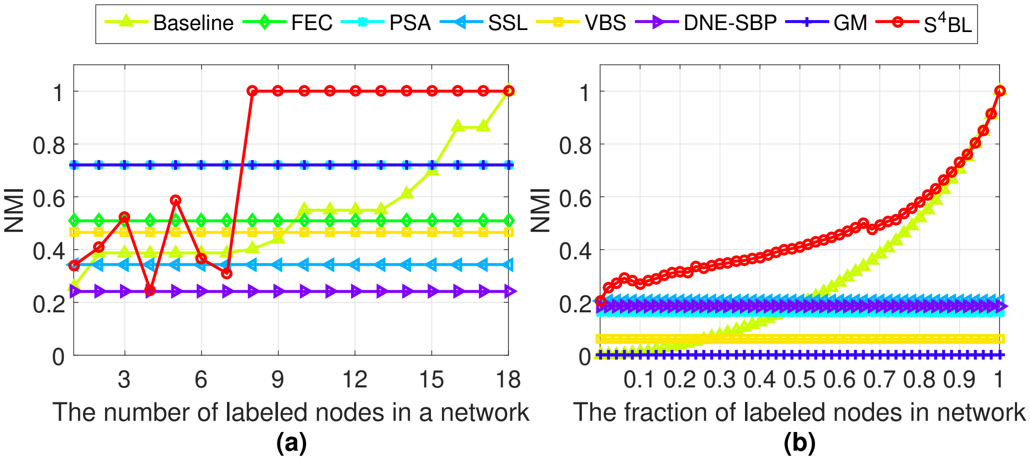


Figure 9: NMI by Baseline, six unsupervised methods, and S^{4}BL (our method) on two real-world signed networks: (a) Monastery; (b) WikiEditor.

The experimental results are shown in Figure 9. For the Monastery network as shown in Figure 9 (a), when the number of labels is kept small, the performance of S^{4}BL fluctuates around DNE-SBP and PSA; however, S^{4}BL achieves better NMI and can assign all nodes to correct communities when the number of labelled nodes exceeds 7. From Figure 9 (b), the NMI of the best performed unsupervised method, SSL, could barely achieve 0.2 because

Table 3: Statistics for real-world unsigned networks.

Networks	n	e	K	Block Structure
Dolphins [36]	62	159	2	community
Football [37]	115	613	12	community
Karate [38]	34	78	2	community
Polbooks [39]	105	441	3	community
Polblogs [40]	1224	16715	2	community

WikiEditor network is sparse and its hidden structure is intricate. But if we labelled 2% nodes, the NMI of S^{4}BM will be 0.255. The results in Figure 9 (b) shows that our method can deal with large-scale networks.

4.3.2. Test S^{4}BL on unsigned networks

For fairness, we use five common real-world unsigned networks, dolphins network [36], football network [37], karate network [38], polbooks network[39] and polblogs [40], to test the performance of the S^{4}BL algorithm when compared with unsigned networks clustering methods that can use additional label information. We do it as there are no semi-supervised methods for signed network detecting hidden structural patterns. Table 3 shows the statistics for these networks, where n, e and K are the numbers of nodes, edges, and communities, respectively. Like in the real-world signed networks discussed above, the nodes with high degrees are selected for labelling with high priorities.

S^{4}BL, SMMB and LP perform well on the dolphins network, karate network and polblogs network, but the performances of S^{4}BL and LP are more stable than SMMB, as shown in Fig. 10 (a), (c) and (e). S^{4}BL performs better than the other algorithms on the football network and polbooks network, as shown in Fig. 10 (b) and (d). From the results, we can conclude that (1) S^{4}BM achieves high NMI with fewer labels compared with other methods; (2) the NMIs of S^{4}BL steadily increase with using more and more labels; (3) the results of S^{4}BM outperform algorithms that can use additional label information on unsigned networks.

4.4. Test the Quality of the Parameter Estimation

In this section, we test the quality of the parameter estimation of S^{4}BL. To do this, we first derive the parameters of generative models based on the

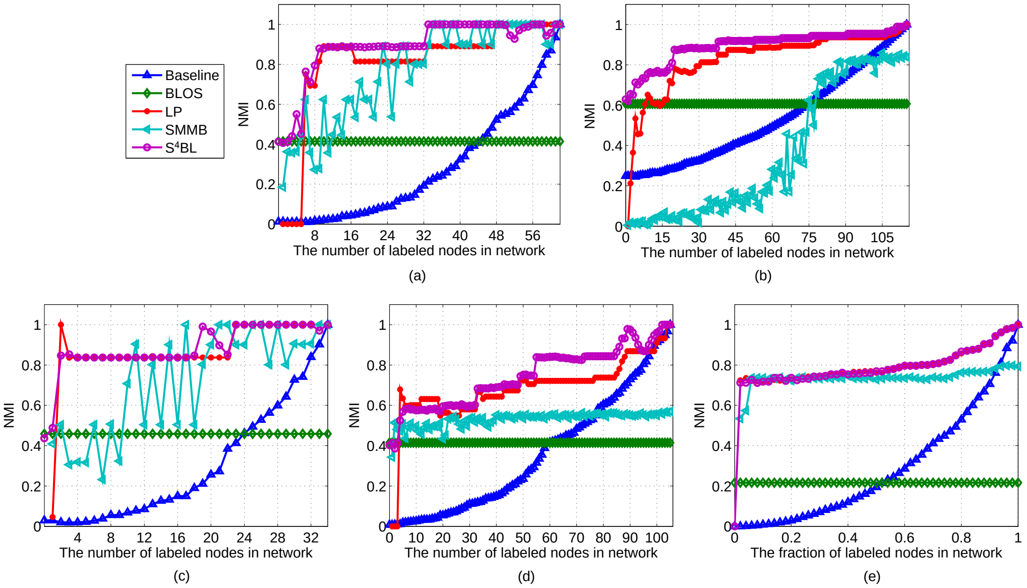


Figure 10: Results on unsigned networks: (a) dolphins; (b) football; (c) karate; (d) polbooks; and (e) polblogs.

estimated hyper-parameters as follows:

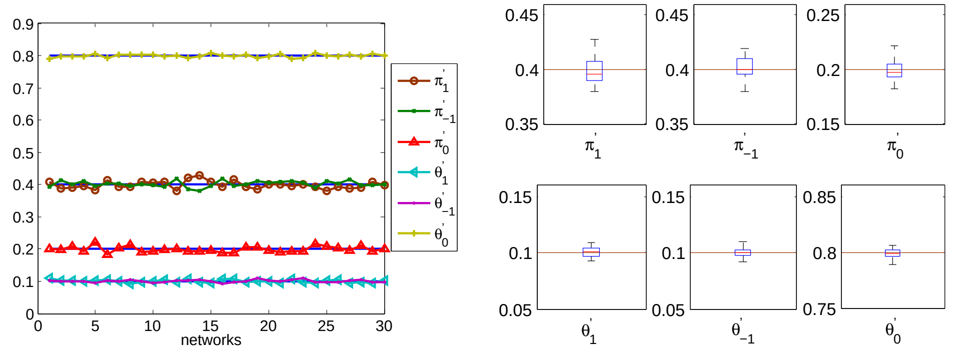
$$\pi_h^{'} = (\eta_h^0 + \sum_k m_k^h) / \sum_{h=1,-1,0} (\eta_h^0 + \sum_k m_k^h),$$

$$\theta_h^{'} = (\mu_h^0 + \sum_{k \neq l} m_{kl}^h) / \sum_{h=1,-1,0} (\mu_h^0 + \sum_{k \neq l} m_{kl}^h),$$

$$\omega_k^{'} = (\rho_k^0 + n_k) / \sum_{k=1}^{K} (\rho_k^0 + n_k),$$

where $h \in \{1, -1, 0\}$; η_h^0 and μ_h^0 are hyper-parameters as defined in paper; m_k^h and m_{kl}^h are, respectively, the number of $h-$links in block k and between blocks k and l; and n_k is the number of nodes in block k.

We first use the model $X = SSBM\big(4, 128, (0.4, 0.4, 0.2), (0.1, 0.1, 0.8), (1/4, 1/4, 1/4, 1/4)\big)$ to generate networks; then, we apply S^{4}BM to the networks to estimate the hyper-parameters and thereafter calculate the model parameters according to the above formulas; finally, we compare the estimated model parameters to the ground truth $\Pi = \{0.4, 0.4, 0.1\}$, $\Theta = \{0.1, 0.1, 0.8\}$ and $\Omega = (1/4, 1/4, 1/4, 1/4)$.



(a) (b)

Figure 11: Testing the quality of parameter estimation. (a) The landscapes of the estimated model parameters obtained on 30 networks. (b) The boxplot of the statistics of the estimated model parameters.

We performed 30 trials by generating 30 networks. The results are shown in Figure 11. For all cases, the estimated Ω' are identical with the ground truth Ω. The landscapes of π_h' and θ_h' are shown in Figure 11(a). As we see, these estimated model parameters are very close to their respective ground truth. In addition, Figure 11(b) shows the detailed statistics of these estimated values.

5. Conclusions

Unsupervised stochastic blockmodels perform very poorly on noisy and sparse signed networks because they only use the networks' topological information. In order to solve this problem, we proposed a semi-supervised signed stochastic blockmodel, or S^{4}BM, to utilize side information such as available labels. We then proposed a variational Bayes learning method, called S^{4}BL to estimate hyper-parameters and latent variables of the S^{4}BM. S^{4}BM is a flexible stochastic model which can detect both community and multipartite structures. To the best of our knowledge, this is the first method using additional information for signed network detecting hidden structural patterns. Extensive experiments were performed to validate S^{4}BL and compare its effectiveness to existing methods on both synthetic and real-world networks. The results indicate that S^{4}BL can significantly improve the performance of SSL through its proposed semi-supervised learning mechanism, and the new model outperforms state-of-the-art methods.

Appendix A. Detailed Derivation of the Log-Likelihood of Complete Data and Parameter Estimation

In this section, we will show the detailed derivation of the log-likelihood of complete data (Eq. (5)) and parameter estimation (Eqs. (10)-(15)).

Appendix A.1. The Derivation of the Log-likelihood of Complete Data

According to multiplication law of probability, we know

$$\log p(N, C, Z|\alpha, \Pi, \Theta, \Omega) = \log p(N|Z, \Pi, \Theta) + \log p(C|Z, \alpha) + \log p(Z|\Omega). \tag{A.1}$$

According to the generative process of links, the logarithmic probability of network N conditioned on the node assignment and the block-block link probability is

$$\begin{aligned} \log p(N|Z, \Pi, \Theta) &= \sum_{i<j} \log p(a_{ij}|z_i, z_j, \Pi, \Theta) \\ &= \sum_{i<j} (\sum_{k} z_{ik} z_{jk} \log M(a_{ij}; \Pi) + \sum_{k \neq l} z_{ik} z_{jl} \log M(a_{ij}; \Theta)), \end{aligned} \tag{A.2}$$

the node assignment logarithmic probability is

$$\log p(Z|\Omega) = \sum_{i=1}^{n} \log M(z_i; \Omega) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \log \omega_k, \tag{A.3}$$

and the logarithmic probability of node labels conditioned on node assignment and the relations between labels and blocks is

$$\log p(C|Z, \alpha) = \sum_{i=1}^{n} \sum_{c=1}^{M} C_{ic} \log p(c|Z_i, \alpha). \tag{A.4}$$

Substitute Eqs. (A.2)-(A.4) into Eq. (A.1), we can obtain Eq. (5).

Appendix A.2. The Derivation of Parameter Estimation

After derive each term in Eq. (9) and substituting them to (9), we can obtain:

$$
\begin{aligned}
\mathcal{L} =& \sum_{h}\Big(\eta_h^0 - \eta_h + \sum_{i<j}^{n}\sum_{q=1}^{K}\tau_{iq}\tau_{jq}\delta(a_{ij},h)\Big)\Big(\psi(\eta_h) - \psi(\sum_{h}\eta_h)\Big) \\
&+ \sum_{h}\Big(\mu_h^0 - \mu_h + \sum_{i<j}^{n}\sum_{q\neq l}^{K}\tau_{iq}\tau_{jl}\delta(a_{ij},h)\Big)\Big(\psi(\mu_h) - \psi(\sum_{h}\mu_h)\Big) \\
&+ \sum_{q=1}^{K}\Big(\big(\rho_q^0 - \rho_q + \sum_{i=1}^{n}\tau_{iq}\big)\big(\psi(\rho_q) - \psi(\sum_{q}\rho_q)\big)\Big) \\
&+ \sum_{i=1}^{N}\sum_{c=1}^{M}\gamma_{ic}(\sum_{k=1}^{K}\alpha_{ck}\tau_{ik} - \log\sum_{c=1}^{M}\sum_{k=1}^{K}\tau_{ik}\exp(\alpha_{ck}) - \log\gamma_{ic}) \\
&- \sum_{i=1}^{n}\sum_{q=1}^{K}\tau_{iq}\log\tau_{iq} + \log\Big\{\big(\Gamma(\sum_{q=1}^{K}\rho_q^0)\prod_{q=1}^{K}\Gamma(\rho_q)\big)/\big(\Gamma(\sum_{q=1}^{K}\rho_q)\prod_{q=1}^{K}\Gamma(\rho_q^0)\big)\Big\} \\
&+ \log\Big\{\big(\Gamma(\sum_{h}\eta_h^0)\prod_{h}\Gamma(\eta_h)\big)/\big(\Gamma(\sum_{h}\eta_h)\prod_{h}\Gamma(\eta_h^0)\big)\Big\} \\
&+ \log\Big\{\big(\Gamma(\sum_{h}\mu_h^0)\prod_{h}\Gamma(\mu_h)\big)/\big(\Gamma(\sum_{h}\mu_h)\prod_{h}\Gamma(\mu_h^0)\big)\Big\},
\end{aligned}
\tag{A.5}
$$

where $\Gamma(\cdot)$ is a gamma function and $\psi(\cdot)$ is the digamma function. Then we estimate the parameters by maximizing Eq. (A.5).

Appendix A.2.1. Estimating $q(z_{ik})$

In Eq. (A.5), the calculation about the term $\log\sum_{c=1}^{M}\sum_{k=1}^{K}\tau_{ik}\exp(\alpha_{ck})$ is hard. Here, we define $\beta\tau_i^T = \sum_{c=1}^{M}\sum_{k=1}^{K}\tau_{ik}\exp(\alpha_{ck})$, where $\beta = [\beta_1, \beta_2, ..., \beta_K]$, $\beta_k = \sum_{c=1}^{M}\exp(\alpha_{ck})$. For $\log(x)$, we know that $\log(x) \leq \zeta^{-1}x + \log(\zeta) - 1, \forall x > 0, \zeta > 0$, and the equality holds iff $x = \zeta$. We set $x = \beta\tau_i^T$, and

$\zeta = \beta(\tau_i^{old})^T$. So $\mathcal{L}'_{[\tau_{ik}]}$ denotes the lower bound of $\mathcal{L}_{[\tau_{ik}]}$:

$$\begin{aligned}\mathcal{L}'_{[\tau_{ik}]} =& \sum_{j\neq i}(\tau_{ik}\tau_{jk}\sum_h \delta(a_{ij},h)(\psi(\eta_h) - \psi(\sum_h \eta_h)) \\ &+ \sum_{k\neq l}\tau_{ik}\tau_{jl}\sum_h \delta(a_{ij},h)(\psi(\mu_h) - \psi(\sum_h \mu_h))) \\ &+ \tau_{ik}(\psi(\rho_q) - \psi(\sum_{k=1}^K \rho_k)) - \tau_{ik}\log\tau_{ik} \\ &+ \sum_{c=1}^M \gamma_{ic}(\alpha_{ck}\tau_{ik} - (\beta(\tau_i^{old})^T)^{-1}\beta_k\tau_{ik} - \log(\beta(\tau_i^{old})^T) + 1).\end{aligned}$$

We set the derivative of $\mathcal{L}'_{[\tau_{ik}]}$ with respect to τ_{ik} to 0, that is,

$$\begin{aligned}\partial\mathcal{L}'_{[\tau_{ik}]}/\partial\tau_{ik} =& \sum_{j\neq i}(\sum_{k=1}^K \tau_{jk}\sum_h \delta(a_{ij},h)(\psi(\eta_h) - \psi(\sum_h \eta_h)) \\ &+ \sum_{k\neq l}\tau_{jl}\sum_h \delta(a_{ij},h)(\psi(\mu_h) - \psi(\sum_h \mu_h))) \\ &+ (\psi(\rho_k) - \psi(\sum_{k=1}^K \rho_k)) - \log\tau_{ik} - 1 \\ &+ \sum_{c=1}^M \gamma_{ic}(\alpha_{ck} - (\beta(\tau_i^{old})^T)^{-1}\beta_k) = 0,\end{aligned}$$

and we can obtain Eq. (10).

Appendix A.2.2. Estimating $q(C_{ic})$

The items with respect to γ_{ic} in Eq. (A.5) are

$$\mathcal{L}_{[\gamma_{ic}]} = \gamma_{ic}(\sum_{k=1}^K \alpha_{ck}\tau_{ik} - \log\sum_{c=1}^M\sum_{k=1}^K \tau_{ik}e^{\alpha_{ck}} - \log\gamma_{ic}).$$

Set the derivative of $\mathcal{L}_{[\gamma_{ic}]}$ with respect to γ_{ic} to 0, that is,

$$\partial\mathcal{L}_{[\gamma_{ic}]}/\partial\gamma_{ic} = \sum_{k=1}^K \alpha_{ck}\tau_{ik} - \log\sum_{c=1}^M\sum_{k=1}^K \tau_{ik}\exp(\alpha_{ck}) - \log\gamma_{ic} - 1 = 0,$$

and we can obtain Eq. (11).

Appendix A.2.3. Estimating $q(\omega_k)$

The items with respect to ρ_k in Eq. (A.5) are

$$\begin{aligned}\mathcal{L}_{[\rho_k]} =& \sum_{i=1}^{N} \tau_{ik}(\psi(\rho_k) - \psi(\sum_{k=1}^{K} \rho_k)) + (\rho_k^0 - 1)(\psi(\rho_k) - \psi(\sum_{k=1}^{K} \rho_k)) \\ &- (\log \Gamma(\sum_{k=1}^{K} \rho_k) - \log \Gamma(\rho_k) + (\rho_k - 1)(\psi(\rho_k) - \psi(\sum_{k=1}^{K} \rho_k))).\end{aligned}$$

Set the derivative of $\mathcal{L}_{[\rho_k]}$ with respect to ρ_k to 0, that is,

$$\begin{aligned}\partial \mathcal{L}_{[\rho_k]}/\partial \rho_k =& \sum_{i=1}^{N} \tau_{ik}(\psi^{'}(\rho_k) - \psi^{'}(\sum_{k=1}^{K} \rho_k)) + (\rho_k^0 - 1)(\psi^{'}(\rho_k) - \psi^{'}(\sum_{k=1}^{K} \rho_k)) \\ &- (\psi(\sum_{k=1}^{K} \rho_k) - \psi(\rho_k) + \psi(\rho_k) - \psi(\sum_{k=1}^{K} \rho_k) + (\rho_k - 1)(\psi^{'}(\rho_k) - \psi^{'}(\sum_{k=1}^{K} \rho_k))) \\ =& (\sum_{i=1}^{N} \tau_{ik} + \rho_k^0 - \rho_k)(\psi^{'}(\rho_k) - \psi^{'}(\sum_{k=1}^{K} \rho_k)) = 0,\end{aligned}$$

and we can obtain Eq. (12).

Appendix A.2.4. Estimating $q(\pi_h)$

The items with respect to η_h in Eq. (A.5) are

$$\begin{aligned}\mathcal{L}_{[\eta_h]} =& \sum_{i<j} (\sum_{k=1}^{K} \tau_{ik}\tau_{jk}(\psi(\eta_h) - \psi(\sum_{h} \eta_h))) + (\eta_h^0 - 1)(\psi(\eta_h) - \psi(\sum_{h} \eta_h)) \\ &- (\log \Gamma(\sum_{h} \eta_h) - \log \Gamma(\eta_h) + (\eta_h - 1)(\psi(\eta_h) - \psi(\sum_{h} \eta_h))).\end{aligned}$$

Set the derivative of $\mathcal{L}_{[\eta_h]}$ with respect to η_h to 0, that is,

$$
\begin{aligned}
\partial\mathcal{L}_{[\eta_h]}/\partial\eta_h = &\sum_{i<j}(\sum_{k=1}^{K}\tau_{ik}\tau_{jk}(\psi^{'}(\eta_h) - \psi^{'}(\sum_h \eta_h))) + (\eta_h^0 - 1)(\psi^{'}(\eta_h) - \psi^{'}(\sum_h \eta_h)) \\
&- (\psi(\sum_h \eta_h) - \psi(\eta_h) + (\psi(\eta_h) - \psi(\sum_h \eta_h)) + (\eta_h - 1)(\psi^{'}(\eta_h) - \psi^{'}(\sum_h \eta_h))) \\
= &(\sum_{i<j}\sum_{k=1}^{K}\tau_{ik}\tau_{jk} + \eta_h^0 - \eta_h)(\psi^{'}(\eta_h) - \psi^{'}(\sum_h \eta_h)) = 0,
\end{aligned}
$$

and we can obtain Eq. (13).

Appendix A.2.5. Estimating $q(\theta_h)$

The items with respect to μ_h in Eq. (A.5) are

$$
\begin{aligned}
\mathcal{L}_{[\mu_h]} = &\sum_{i<j}(\sum_{k\neq l}^{K}\tau_{ik}\tau_{jl}(\psi(\mu_h) - \psi(\sum_h \mu_h))) + (\mu_h^0 - 1)(\psi(\mu_h) - \psi(\sum_h \mu_h)) \\
&- (\log\Gamma(\sum_h \mu_h) - \log\Gamma(\mu_h) + (\mu_h - 1)(\psi(\mu_h) - \psi(\sum_h \mu_h))).
\end{aligned}
$$

Set the derivative of $\mathcal{L}_{[\mu_h]}$ with respect to μ_h to 0, that is,

$$
\begin{aligned}
\partial\mathcal{L}_{[\mu_h]}/\partial\mu_h = &\sum_{i<j}\sum_{k\neq l}^{K}\tau_{ik}\tau_{jl}(\psi^{'}(\mu_h) - \psi^{'}(\sum_h \mu_h)) + (\mu_h^0 - 1)(\psi^{'}(\mu_h) - \psi^{'}(\sum_h \mu_h)) \\
&- (\psi(\sum_h \mu_h) - \psi(\mu_h) + (\psi(\mu_h) - \psi(\sum_h \mu_h)) + (\mu_h - 1)(\psi^{'}(\mu_h) - \psi^{'}(\sum_h \mu_h))) \\
= &(\sum_{i<j}\sum_{k\neq l}^{K}\tau_{ik}\tau_{jl} + \mu_h^0 - \mu_h)(\psi^{'}(\mu_h) - \psi^{'}(\sum_h \mu_h)) = 0,
\end{aligned}
$$

and we can obtain Eq. (14).

Appendix A.2.6. Estimating α_{ck}

Finally, the items with respect to α_{ck} is

$$\mathcal{L}_{[\alpha_{ck}]} = \sum_{i=1}^{N} \gamma_{ic}(\alpha_{ck}\tau_{ik} - \log \sum_{c=1}^{M} \sum_{k=1}^{K} \tau_{ik} \exp(\alpha_{ck})).$$

Calculating the derivative of $\mathcal{L}_{[\alpha_{ck}]}$ with respect to α_{ck}, we obtain Eq. (15).

Acknowledgements

This work was supported by the National Natural Science Foundation of China [grant numbers 61876069, 61572226]; Jilin Province Natural Science Foundation [grant number 20150101052JC]; Jilin Province Key Scientific and Technological Research and Development project [grant numbers 20180201067GX, 20180201044GX]; the China Scholarship Council [grant numbers 201906170205, 201906170208]; Australian Research Council [grant number DP190101087]; Guangdong Natural Science Foundation [grant number 2018A030313339]; Characteristic Innovation Projects of Universities in Guangdong [grant number 2017GKTSCX063]; MOE (Ministry of Education in China) Youth Fund Project of Humanities and Social Sciences [grant number 17YJCZH261]; and Scientific Research Team Project of Shenzhen Institute of Information Technology [grant number SZIIT2019KJ022].

References

[1] G. Facchetti, G. Iacono, C. Altafini, Exploring the low-energy landscape of large-scale signed social networks, Physical Review E 86 (3) (2012) 036116.

[2] W. Song, S. Wang, B. Yang, Y. Lu, X. Zhao, X. Liu, Learning node and edge embeddings for signed networks, Neurocomputing 319 (2018) 42–54.

[3] B. Yang, J. Liu, D. Liu, Characterizing and extracting multiplex patterns in complex networks, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 42 (2) (2012) 469–481. doi:10.1109/TSMCB.2011.2167751.

[4] B. Yang, W. Cheung, J. Liu, Community mining from signed social networks, IEEE Trans. on Knowl. and Data Eng. 19 (10) (2007) 1333–1348.

[5] J. Zhou, L. Li, A. Zeng, Y. Fan, Z. Di, Random walk on signed networks, Physica A: Statistical Mechanics and its Applications 508 (2018) 558 – 566.

[6] V. A. Traag, J. Bruggeman, Community detection in networks with positive and negative links, Physical Review E 80 (3) (2009) 036115.

[7] P. Anchuri, M. Magdon-Ismail, Communities and balance in signed networks: A spectral approach, in: Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012), IEEE Computer Society, 2012, pp. 235–242.

[8] X. Shen, F.-L. Chung, Deep network embedding for graph representation learning in signed networks, IEEE transactions on cybernetics.

[9] K.-Y. Chiang, J. J. Whang, I. S. Dhillon, Scalable clustering of signed networks using balance normalized cut, in: Proceedings of the 21st ACM international conference on Information and knowledge management, 2012, pp. 615–624.

[10] P. Mercado, F. Tudisco, M. Hein, Clustering signed networks with the geometric mean of laplacians, in: Advances in Neural Information Processing Systems, 2016, pp. 4421–4429.

[11] J. Q. Jiang, Stochastic block model and exploratory analysis in signed networks, Physical Review E 91 (6) (2015) 062805.

[12] Y. Chen, X. Wang, B. Yuan, B. Tang, Overlapping community detection in networks with positive and negative links, Journal of Statistical Mechanics: Theory and Experiment 2014 (3) (2014) P03021.

[13] B. Yang, X. Liu, Y. Li, X. Zhao, Stochastic blockmodeling and variational bayes learning for signed network analysis, IEEE Transactions on Knowledge and Data Engineering 29 (9) (2017) 2026–2039.

[14] X. Zhao, X. Liu, H. Chen, Network modelling and variational bayesian inference for structure analysis of signed networks, Applied Mathematical Modelling 61 (2018) 237–254.

[15] D. Liu, X. Liu, W. Wang, H. Bai, Semi-supervised community detection based on discrete potential theory, Physica A: Statistical Mechanics and its Applications 416 (2014) 173–182.

[16] P. Zhang, C. Moore, L. Zdeborová, Phase transitions in semisupervised clustering of sparse networks, Physical Review E 90 (5) (2014) 052802.

[17] L. Yang, X. Cao, D. Jin, X. Wang, D. Meng, A unified semi-supervised community detection framework using latent space graph regularization, IEEE transactions on cybernetics 45 (11) (2015) 2585–2598.

[18] E. Mossel, J. Xu, Local algorithms for block models with side information, in: Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, ACM, 2016, pp. 71–80.

[19] X. Ma, B. Wang, L. Yu, Semi-supervised spectral algorithms for community detection in complex networks based on equivalence of clustering methods, Physica A: Statistical Mechanics and its Applications 490 (2018) 786–802.

[20] M. Ganji, J. Chan, P. J. Stuckey, J. Bailey, C. Leckie, K. Ramamohanarao, L. Park, Semi-supervised blockmodelling with pairwise guidance, in: M. Berlingerio, F. Bonchi, T. Gärtner, N. Hurley, G. Ifrim (Eds.), Machine Learning and Knowledge Discovery in Databases, Springer International Publishing, Cham, 2019, pp. 158–174.

[21] S. Fortunato, D. Hric, Community detection in networks: A user guide, Physics Reports 659 (2016) 1–44.

[22] D. Liu, H.-Y. Bai, H.-J. Li, W.-J. Wang, Semi-supervised community detection using label propagation, International Journal of Modern Physics B 28 (29) (2014) 1450208.

[23] X. Ma, L. Gao, X. Yong, L. Fu, Semi-supervised clustering algorithm for community structure detection in complex networks, Physica A: Statistical Mechanics and its Applications 389 (1) (2010) 187–197.

[24] Z.-Y. Zhang, Community structure detection in complex networks with partial background information, EPL (europhysics letters) 101 (4) (2013) 48005.

[25] C. Moore, X. Yan, Y. Zhu, J.-B. Rouquier, T. Lane, Active learning for node classification in assortative and disassortative networks, in: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2011, pp. 841–849.

[26] L. Peel, Supervised blockmodelling, arXiv preprint arXiv:1209.5561.

[27] P. Holland, K. Laskey, S. Leinhardt, Stochastic blockmodels: First steps 5 (1983) 109–137.

[28] M. D. Hoffman, D. M. Blei, C. Wang, J. Paisley, Stochastic variational inference, The Journal of Machine Learning Research 14 (1) (2013) 1303–1347.

[29] B. Yang, X. Zhao, On the scalable learning of stochastic blockmodel., in: AAAI, 2015, pp. 360–366.

[30] X. Zhu, Z. Ghahramani, J. Lafferty, et al., Semi-supervised learning using gaussian fields and harmonic functions, in: ICML, Vol. 3, 2003, pp. 912–919.

[31] L. I. Kuncheva, S. T. Hadjitodorov, Using diversity in cluster ensembles, in: 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583), Vol. 2, 2004, pp. 1214–1219 vol.2. doi:10.1109/ICSMC.2004.1399790.

[32] B. Yang, X. Zhao, X. Liu, Bayesian approach to modeling and detecting communities in signed network, in: AAAI, 2015, pp. 1952–1958.

[33] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Physical review E 78 (4) (2008) 046110.

[34] P. Doreian, A. Mrvar, A partitioning approach to structural balance, Social networks 18 (2) (1996) 149–168.

[35] S. Yuan, X. Wu, Y. Xiang, SNE: signed network embedding, in: Advances in Knowledge Discovery and Data Mining - 21st Pacific-Asia Conference, PAKDD 2017, Jeju, South Korea, May 23-26, 2017, Proceedings, Part II, 2017, pp. 183–195.

[36] D. Lusseau, M. E. Newman, Identifying the role that animals play in their social networks, Proceedings of the Royal Society of London B: Biological Sciences 271 (Suppl 6) (2004) S477–S481.

[37] M. Girvan, M. E. Newman, Community structure in social and biological networks, PNAS 99 (12) (2002) 7821–7826.

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

[39] M. E. Newman, Finding community structure in networks using the eigenvectors of matrices, Physical review E 74 (3) (2006) 036104.

[40] L. A. Adamic, N. Glance, The political blogosphere and the 2004 u.s. election: Divided they blog, in: Proceedings of the 3rd International Workshop on Link Discovery, LinkKDD '05, ACM, New York, NY, USA, 2005, pp. 36–43. doi:10.1145/1134271.1134277.
URL http://doi.acm.org/10.1145/1134271.1134277