Statistica Sinica **35** (2025), 1763-1788
doi:https://doi.org/10.5705/ss.202023.0081

A PSEUDO-LIKELIHOOD APPROACH TO COMMUNITY DETECTION IN WEIGHTED NETWORKS

Andressa Cerqueira* and Elizaveta Levina

Federal University of São Carlos and University of Michigan

Abstract: Community structure is common in many real networks, with nodes clustered in groups sharing the same connections patterns. While many community detection methods have been developed for networks with binary edges, few of them are applicable to networks with weighted edges, which are common in practice. We propose a pseudo-likelihood community estimation algorithm derived under the weighted stochastic block model for networks with normally distributed edge weights, extending the pseudo-likelihood algorithm for binary networks, which offers some of the best combinations of accuracy and computational efficiency. We prove that the estimates obtained by the proposed method are consistent under the assumption of homogeneous networks, a weighted analogue of the planted partition model, and show that they work well in practice for both homogeneous and heterogeneous networks. We illustrate the method on simulated networks and on a fMRI dataset, where edge weights represent connectivity between brain regions and are expected to be close to normal in distribution by construction.

Key words and phrases: Community detection, pseudo-likelihood, weighted networks.

1. Introduction

Network models have been a useful general tool for understanding and modeling interactions between objects in many domains. The relationships (edges) between objects (nodes) can represent many things depending on the application: social interactions, trading partnerships, web links, packets sent between computers, disease contagion, neural connectivity, and so on. Some settings result in binary networks, where only the presence or absence of an edge is recorded, and other settings lead to weighted networks, where an edge is associated with a weight which typically quantifies the strength of the connection.

The probabilistic modeling of networks has traditionally focused on binary networks, starting from the classical Erdös-Rényi graph (Erdös and Rényi, 1960) which models edges as i.i.d. Bernoulli random variables. Yet many networks encountered in practice are weighted, and many of the binary networks in the literature are obtained by thresholding raw edge weights. For instance, the arguably most studied network with communities, the karate club dataset

*Corresponding author. E-mail: acerqueira@ufscar.br

(Zachary, 1977) is a binary network representing friendships, but the data were collected by observing frequency of the club members' interactions, and many other traditional social networks are recorded through similar mechanisms. Moreover, even when there is a true binary edge (e.g., friendship on Facebook), there is often an accompanying strength measurement (e.g., how long these people have been friends, how many of each other's posts they have liked, etc.).

Network communities, a term used to loosely refer to groups of nodes sharing connectivity patterns, have been observed in many real-world networks and studied extensively. The stochastic block model (SBM) (Holland, Laskey and Leinhardt, 1983), probably the most studied network model with communities, models a binary undirected graph with independent edges, with probability of an edge determined by community membership of the incident nodes. A lot is now known about theory and algorithms for recovering communities in this setting; see Abbe (2018) and references therein for a recent review. There are also multiple extensions of the binary SBM, such as the degree-corrected SBM (Karrer and Newman, 2011), and multiple models with overlapping communities, for example, Airoldi et al. (2008), Zhang, Levina and Zhu (2020), and Latouche, Birmelé and Ambroise (2011). A version called labeled SBM (Heimlicher, Lelarge and Massoulié, 2012; Lelarge, Massoulié and Xu, 2015) allows for multiple different types of edges between a pair of nodes, which one could think of as an edge vector or a discrete categorical edge weight.

Our focus in this paper is on networks with real-valued edge weights. They occur in many applications, notably brain connectivity networks obtained from various types of neuroimaging, and other biological networks such as gene-gene interactions. Typically the weights represent some quantitative similarity measure between the nodes, such as marginal or partial correlations. While these measures can be thresholded to obtain a binary network, there is at least empirical evidence that a lot of useful information is lost in the process (Arroyo et al., 2019).

Several models have been proposed to directly model networks with continuous-valued edge weights. For instance, Aicher, Jacobs and Clauset (2014) proposed a generalization of the SBM for weighted edges and a variational Bayes approach to learning the latent community labels. More recently, Xu, Jog and Loh (2020) obtained the optimal error rate of any clustering algorithm for weighted SBM through deriving an information-theoretic lower bound, and proposed an algorithm that achieves the optimal rate using discretization of weighted SBM into a labeled SBM. In the same line of research, Avrachenkov, Dreveton and Leskelä (2020) extended these results by allowing general probability distributions for the edge weights. Other methods proposed to estimate communities in a weighted network include algorithms developed in the complex systems community that do no rely on a generative statistical model (Reichardt and Bornholdt, 2006; Blondel et al., 2008; Rosvall and Bergstrom,

2008) and hierarchical clustering (Balakrishnan et al., 2011). Another line of work has focused on jointly analyzing multiple weighted networks, largely motivated by neuroimaging problems; see, for example, Levin, Lodhia and Levina (2022), MacDonald, Levina and Zhu (2021), and references therein. The work on community detection in multiple networks, however, has focused on binary networks to the best of our knowledge (Arroyo et al., 2021; Tang, Lu and Dhillon, 2009; Le, Levin and Levina, 2018; Bhattacharyya and Chatterjee, 2018; Wang et al., 2021).

Another related line of work is the submatrix localization problem, especially in the setting of a random matrix with Gaussian entries with a constant variance. The problem is to locate a submatrix of a known size of entries with positive means, while the rest of the matrix entries are assumed to have mean 0. The case of one hidden submatrix can be viewed as a special case of weighted SBM with two communities and was studied by Ma and Wu (2015) and Hajek, Wu and Xu (2018). Chen and Xu (2016) studied the case of more than one hidden submatrix, all of the same size.

In this paper, we consider the community detection problem for a single weighted SBM network. We model edge weights as Gaussian random variables (in practice this may be after a suitable transformation, for example, the Fisher transform for correlations), with means and variances determined by the communities of the incident nodes. Our main contribution is a fast and tractable pseudo-likelihood algorithm for fitting this model, inspired by the seminal pseudo-likelihood algorithm of Amini et al. (2013) for both the binary SBM and the degree-corrected binary SBM. Wang et al. (2023) extended it to a profile pseudo-likelihood approach, also for binary networks only. We prove that one iteration step of our pseudo-likelihood algorithm achieves the optimal error rate derived by Xu, Jog and Loh (2020), up to a constant. We characterize the optimal error rate for this setting explicitly in terms of the means and variances of the within- and between-community edge distributions, revealing interesting trade-offs that do not appear in the binary setting. We also show empirically that the pseudo-likelihood algorithm improves on other community detection methods when they are used to provide an initial value. We allow for any fixed number of communities of different sizes, and do not assume assortativity in any form. In contrast with the work by Aicher, Jacobs and Clauset (2014), we also allow the means and variances of the edges weights to depend on the number of nodes, thus controlling the overall expected "degree".

This paper is organized as follows. Section 2 introduces the weighted SBM and proposes the pseudo-likelihood based EM algorithm for fitting the model. Section 3 presents our main consistency results. The performance of the algorithm on simulated networks and comparison with other methods are presented in Section 4. An application to brain connectivity networks is studied in Section 5. We conclude with discussion in Section 6.

2. A Pseudo-Likelihood Algorithm for the Weighted Stochastic Block Model

Consider a weighted undirected network W with the node set $\{1, 2, \ldots, n\}$. Let $C_1, C_2, \ldots, C_n$ be independent and identically distributed random variables representing node community labels, with $\mathbb{P}(C_i = k) = \pi_k$, $k = 1, \ldots, K$ and $\pi = (\pi_1, \ldots, \pi_K)$. The *weighted stochastic block model* (WSBM) assumes that, conditionally on the node labels $c = (c_1, \ldots, c_n)$, the edge weights are drawn independently from a distribution that depends only on the labels of the incident nodes. In this paper, we study the model where the number of communities K is fixed and known, and the distribution of edge weights is Gaussian. Since our goal is to derive a pseudo-likelihood algorithm, some distributional assumption is needed, and many real weighted networks are reasonably well described by a Gaussian distribution of weights. In particular, the motivating example in Section 5 is on brain connectivity networks from fMRI, with edge weights measured as Fisher-transformed Pearson correlations, which are designed to be approximately Gaussian. Formally, we have

$$\mathscr{L}(W_{ij} \,|\, C_i = k, C_j = l) = \mathscr{N}(B_{kl}\,,\, \Sigma_{kl})\,, \qquad 1 \le i < j \le n \tag{2.1}$$

where $B \in \mathbb{R}^{K \times K}$ and $\Sigma \in \mathbb{R}_+^{K \times K}$ are symmetric matrices that contain the means and the variances of the edge weights, respectively. Diagonal elements of the matrix W typically carry no information, and in some applications may be set to 0; for completeness, we set the diagonal elements $W_{ii} = 0$, but their distribution can be left unspecified since they are not included in any calculations.

Zero entries in weighted networks are sometimes interpreted as non-observed edges, e.g., in Aicher, Jacobs and Clauset (2014) and Xu, Jog and Loh (2020). We simply treat this as an edge with weight 0 (which has probability 0 in the Gaussian setting), and assume all edge weights are observed, resulting in a dense weighted SBM. This is a common scenario in real weighted networks, including those from neuroimaging.

Under this model, the log-likelihood function for an observed network $W = w$ is given by

$$l(\pi, B, \Sigma; w) = \log \left\{ \sum_{e \in \{1,\ldots,K\}^n} p(w, e \,|\, \pi, B, \Sigma) \right\}, \tag{2.2}$$

where e is an arbitrary community assignment,

$$p(w, e \,|\, \pi, B, \Sigma) = \tag{2.3}$$
$$\prod_{k=1}^{K} \pi_k^{n_k(e)} \times \prod_{k,l=1}^{K} (2\pi\Sigma_{kl})^{-n_{kl}(e)/2} \exp\left\{ - \sum_{1 \le i < j \le n} \frac{(w_{ij} - B_{kl})^2}{2\Sigma_{kl}} \mathbb{1}\{e_i = k, e_j = l\} \right\},$$

and

$$n_k(e) = \sum_{i=1}^{n} \mathbb{1}\{e_i = k\} \ , \ n_{kl}(e) = \sum_{1 \le i < j \le n} \mathbb{1}\{e_i = k, e_j = l\}. \tag{2.4}$$

Given an observed weighted network $W = w$, our goal is to estimate the label assignments c. Since the log-likelihood function (2.2) involves a sum over all possible labels configurations (K^n terms), maximizing the likelihood or applying the EM algorithm directly is not tractable, as discussed for the binary networks case in Amini et al. (2013). Aicher, Jacobs and Clauset (2014) proposed a variational Bayes approach for the case of edge weights drawn from exponential family distributions and demonstrated its good empirical performance, but variational Bayes tends to scale poorly with the number of nodes and there are no theoretical performance guarantees available for this method. Another proposal is the discretization-based rate-optimal method by Xu, Jog and Loh (2020), which works by converting a weighted network to a labeled network and by applying spectral methods to labeled networks and does not make a distributional assumption about edge weights. This method achieves the optimal error rate for a discretization level that satisfies theoretical requirements. However, in practice the performance of this method depends significantly on the choice of the level of discretization.

Inspired by the pseudo-likelihood approach of Amini et al. (2013) for binary networks, we aim to replace the full intractable likelihood function (2.2) with an approximate likelihood of appropriate sums of edge weights. Let $e = (e_1, \ldots, e_n) \in \{1, \ldots, K\}^n$ be an arbitrary vector of node labels. For $i = 1, \ldots, n$ and $k = 1, \ldots, K$, we define $S_{ik}(e)$ as the sum of weights of edges between node i and all nodes in community k, that is,

$$S_{ik}(e) = \sum_{j=1}^{n} W_{ij} \mathbb{1}\{e_j = k\}\,, \tag{2.5}$$

and write s_{ik} for the corresponding observed quantity. For each node i, define the vector of block sums $S_i(e) = (S_{i1}(e), \ldots, S_{iK}(e))$. Let R be the $K \times K$ confusion matrix between the label vector e and the true labels c, defined by

$$R_{kl} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{e_i = k, c_i = l\}\,. \tag{2.6}$$

Conditionally on the true labels c, $\{S_{i1}(e), \ldots, S_{iK}(e)\}$ are mutually independent random variables. Moreover, again conditionally on c, for a node i with $c_i = k$, each variable S_{il}, $l = 1, \ldots, K$, follows the normal distribution $\mathcal{N}(P_{kl}\,, \Lambda_{kl})$, where the mean and variance are given by

$$P_{kl} = nR_{l\cdot}B_{\cdot k}\,, \ \Lambda_{kl} = nR_{l\cdot}\Sigma_{\cdot k}\,,$$

and $M_{k\cdot}$ and $M_{\cdot k}$ denote, respectively, the kth row and column of a matrix M.

The form of $K \times K$ parameter matrices P and Λ suggests a Gaussian mixture distribution, and indeed each random vector $S_i(e)$, conditioned on c, can be written as a mixture of K Gaussian random vectors with the probability distribution function

$$p(s_i(e); \pi, P, \Lambda) = \sum_{l=1}^{K} \pi_l p(s_i(e);\, P_{l\cdot}, \Lambda_{l\cdot})\,. \tag{2.7}$$

Since we consider undirected networks, the matrix W is symmetric, and the block sums for nodes i and j are not independent. Specifically, they share exactly one common summand, $W_{ij} = W_{ji}$, while all other terms in the sums are independent by assumption. The pseudo-likelihood part of our approach is to ignore this fairly weak dependence, and treat the row block sums as independent, leading to the pseudo log-likelihood function (up to a constant) given by

$$\ell_{\mathrm{PL}}(\pi, P, \Lambda; \{s_i(e)\}) = \sum_{i=1}^{n} \log \left(\sum_{l=1}^{K} \pi_l \prod_{k=1}^{K} \frac{1}{\sqrt{\Lambda_{lk}}} \exp \left[\frac{-\{s_{ik}(e) - P_{lk}\}^2}{2\Lambda_{lk}} \right] \right). \tag{2.8}$$

The pseudo-likelihood differs from the likelihood of the block sums under the WSBM by the one assumption of independence between W_{ij} and W_{ji}, namely treating them as two i.i.d. variables instead of setting $W_{ij} = W_{ji}$. We can reasonably hope that this minor departure from the problem structure should not affect the results too much, and ultimately we will prove that maximizing the objective function (2.8) results in a solution close to the truth under appropriate conditions.

The function (2.8) is the log-likelihood of a Gaussian mixture model, with mixture components labels matching the distribution of the latent communities, and thus fitting this model, which can be done by a standard EM algorithm for Gaussian mixture models, allowing us to estimate the true labels $c = (c_1, \ldots, c_n)$. The EM algorithm starts from an initial value of the labels, say c_0, estimates the parameters given the labels, then updates the labels, and repeats this process either until the parameters converge or for a fixed number of iterations T. The steps of the EM algorithm are given in Algorithm 1.

Once we have estimated the labels $\hat{c}$, the parameters π, B and Σ can be easily obtained in closed form by maximizing the complete likelihood (2.3). For any fixed label assignment e, the likelihood (2.3) is maximized by

$$\begin{aligned} \widehat{\pi}_k(e) &= \frac{n_k(e)}{n}\,, \\ \widehat{B}_{kl}(e) &= \frac{1}{n_{kl}(e)} \sum_{1 \le i < j \le n} w_{ij} \mathbb{1}\{e_i = k, e_j = l\}, \end{aligned}$$

Algorithm 1 The pseudo-likelihood algorithm for estimating labels.

Input: Initial labeling c_0, number of communities K and the network matrix W
Output: The estimated label vector $\hat{c}$

1. Initialize parameters $\widehat{\pi}_l(c_0)$, $\widehat{R} = \mathrm{diag}(\widehat{\pi}_1(c_0), \ldots, \widehat{\pi}_K(c_0))$, $\widehat{P}_{lk} = n\widehat{R}_k.\widehat{B}_{.l}$ and $\widehat{\Lambda}_{lk} = n\widehat{R}_k.\widehat{\Sigma}_{.l}$.
2. **Repeat** T times
3. **Repeat** until the parameter estimates $\hat{\pi}$, $\hat{P}$ and $\hat{\Lambda}$ converge
   3.1. Compute the block sums using (2.5).
   3.2. Estimate the probabilities $\mathbb{P}_{PL}(c_i = l|\mathbf{s}_i(e))$ by

$$\widehat{\pi}_{il} = \frac{\widehat{\pi}_l \prod_{k=1}^{K} \frac{1}{(\widehat{\Lambda}_{lk})^{1/2}} \exp\left[\frac{-\{s_{ik}(e) - \widehat{P}_{lk}\}^2}{2\widehat{\Lambda}_{lk}}\right]}{\sum_{m=1}^{K} \widehat{\pi}_m \prod_{k=1}^{K} \frac{1}{(\widehat{\Lambda}_{mk})^{1/2}} \exp\left[\frac{-\{s_{ik}(e) - \widehat{P}_{mk}\}^2}{2\widehat{\Lambda}_{mk}}\right]}. \tag{2.9}$$

   3.3. Update the parameter values:

$$\widehat{\pi}_l = \frac{1}{n}\sum_{i=1}^{n} \widehat{\pi}_{il}, \widehat{P}_{lk} = \frac{\sum_{i=1}^{n} \widehat{\pi}_{il} s_{ik}(e)}{\sum_{i=1}^{n} \widehat{\pi}_{il}}, \ \widehat{\Lambda}_{lk} = \frac{\sum_{i=1}^{n} \widehat{\pi}_{il}\{s_{ik}(e) - \widehat{P}_{lk}\}^2}{\sum_{i=1}^{n} \widehat{\pi}_{il}}. \tag{2.10}$$

4. Update the labels: $e_i = \mathrm{argmax}_{l=1,\ldots,K} \widehat{\pi}_{il}$.
Return: $\hat{c} = e$

$$\widehat{\Sigma}_{kl}(e) = \frac{1}{n_{kl}(e)} \sum_{1 \le i < j \le n} (w_{ij} - \widehat{B}_{kl})^2 \mathbb{1}\{e_i = k, e_j = l\}.$$

Plugging in the labels found by Algorithm 1 gives the final parameter estimates.

As always, how well an EM algorithm performs depends on the initial value c_0. The consistency analysis in Section 3 quantifies this dependence, at one step of the algorithm, in terms of the fraction of initial labels that match the true labels c. In practice, the EM algorithm can be initialized with the output of any clustering algorithm, such as spectral clustering; we discuss different choices of the initial value in detail in Section 4.

3. Consistency Results

We establish consistency of the estimated node labels $\hat{c}$ obtained from the pseudo-likelihood algorithm as the number of nodes n grows. We focus on what is called weak consistency in the literature: the fraction of mislabeled nodes converges to zero in probability as $n \to \infty$. In this framework, we treat c as an unknown parameter and define the error for an estimate $\hat{c}$ as

$$L(\hat{c}, c) = \min_{\phi \in \Phi_K} \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{\hat{c}_i \neq \phi(c_i)\}, \tag{3.1}$$

where Φ_K is the set of all permutations of community labels $\{1, \ldots, K\}$.

We will establish weak consistency of the estimator $\hat{c}$, meaning $\mathbb{P}\{L(\hat{c}, c) > \epsilon\} \to 0$ for all $\epsilon > 0$, as $n \to \infty$. We study consistency under the *homogeneous* weighted SBM, where all within-community edge weights have the same distribution, and so do all between-community edge weights. This is a common theoretical framework, often called the planted partition model for binary networks. In our setting, we assume that the $K \times K$ mean matrix B and the $K \times K$ variance matrix Σ are given by

$$B_{kl} = \begin{cases} a, \text{ if } k = l \\ b, \text{ if } k \neq l \end{cases} \qquad \text{and} \quad \Sigma_{kl} = \sigma^2, \text{ for all } k, l\,, \tag{3.2}$$

where $a, b \in \mathbb{R}$ and $\sigma^2 > 0$. All of the parameters a, b and σ^2 can vary with n, but most of the time we suppress this dependence to simplify notation.

Intuitively, the larger the absolute difference between the means $|a - b|$ is relative to σ, the easier it should be to recover communities. Besides that, as with all SBM-based community detection, the problem should get harder for larger K and for unbalanced community sizes. Yet even in the homogeneous setting, it is clear that the trade-offs for weighted SBM are more complex than they are for the binary SBM, not least because both the mean and the variance parameters are involved. Our goal is to establish conditions for consistency that make these trade-offs as explicit as possible.

We will study a single label update step (4) of the algorithm, which updates the estimated labels $\hat{c}^{(t)}$ to $\hat{c}^{(t+1)}$. Then for each node i, the label update is given by

$$\hat{c}_i^{(t+1)} = \underset{k=1,\ldots,K}{\operatorname{argmax}} \left(\log \hat{\pi}_k(\hat{c}^{(t)}) - \left[\sum_{m=1}^{K} \frac{\{s_{im}(\hat{c}^{(t)}) - \hat{P}_{km}(\hat{c}^{(t)})\}^2}{2\hat{\Lambda}_{km}(\hat{c}^{(t)})} + \frac{1}{2} \log \hat{\Lambda}_{km}(\hat{c}^{(t)}) \right] \right), \tag{3.3}$$

where $\hat{P}(\hat{c}^{(t)}) = n(R(\hat{c}^{(t)})\hat{B}(\hat{c}^{(t)}))^T$, $\hat{\Lambda}(\hat{c}^{(t)}) = n(R(\hat{c}^{(t)})\hat{\Sigma}(\hat{c}^{(t)}))^T$, and $\hat{c}^{(0)} \equiv c_0$.

Since the estimator (3.3) depends on $\hat{\pi}_k(\hat{c}^{(t)}), \hat{P}(\hat{c}^{(t)})$ and $\hat{\Lambda}(\hat{c}^{(t)})$ we would expect the consistency results to depend on how good these estimates are. For the parameter values a and b, we only require a very mild condition, a correct ordering of $\hat{a}(\hat{c}^{(t)})$ and $\hat{b}(\hat{c}^{(t)})$, and will assume the parameter estimates belong to the set

$$S_{a,b} = \{\, (\hat{a}, \hat{b}, \hat{\sigma}^2) \in \mathbb{R}^3 \,|\, (\hat{a} - \hat{b})(a - b) > 0 \text{ and } \hat{\sigma}^2 > 0 \,\}\,. \tag{3.4}$$

For the initial labels c_0, we will express all the consistency results in terms of the proportion of labels in c_0 that match the true community labels c, up to a permutation of community labels $1, \ldots, K$. All consistency results on community detection involve this permutation which is luckily irrelevant in practice, and to keep notation simple, we will omit it from further discussion. The proportion of correct initial values is not relevant asymptotically, but our empirical results

in Section 4 show, not surprisingly, that in practice starting from a good initial value is beneficial.

3.1. Balanced communities

We start from the case of balanced communities, that is, we assume each community has m nodes and $n = mK$. We further assume the initial labeling $c_0 \in \{1, \dots, K\}^n$ matches γm labels in each community, resulting in the overall matching proportion $\gamma \in (0, 1)$. The remaining misclassified $(1-\gamma)m$ nodes of each community are assumed to be distributed equally between the other $K-1$ community assignments. This set of initial partitions can be formally written as

$$\mathcal{E}_\gamma = \Bigg\{ e \in \{1, \dots, K\}^n : \text{ for all } k, l = 1, \dots, K, \\ \sum_{i=1}^{n} \mathbb{1}\{e_i = k, c_i = l\} = \gamma m \mathbb{1}(k = l) + \frac{(1-\gamma)m}{K-1} \mathbb{1}(k \neq l) \Bigg\}.$$

We emphasize that we do not know which initial labels are correct, only the proportion γ. The assumption that the misclassified nodes are equally distributed among the other $K-1$ communities is a technical assumption made only to simplify derivations of the confusion matrix in the proof of the Theorem 1 when $K > 2$; the proof under a relaxation of this assumption but with uglier algebra can be found in the Supplementary Material.

For any $\gamma \in (0, 1)$, let $h(\gamma) = -\gamma \log(\gamma) - (1-\gamma)\log(1-\gamma)$ be the binary entropy and let $\kappa_\gamma(n) = [\log n - \log\{4\pi\gamma(1-\gamma) + (1/3n)\}]/n$. The following theorem gives a probabilistic upper bound on the error of a single update step of the algorithm given by estimator (3.3). We analyze the first step of the algorithm, that is, set $t = 0$ and omit t from now on.

Theorem 1 (Balanced case). *Assume that $\pi_1 = \cdots = \pi_K = 1/K$. Let the initial labeling $e \in \mathcal{E}_\gamma$ and let $\hat{c}(e)$ be the estimate of the labels obtained from (3.3). For $a, b \in \mathbb{R}$, $a \neq b$, $\sigma^2 > 0$, $\gamma \in (0, 1)$, $\gamma \neq 1/K$ we have*

$$\sup_{e \in \mathcal{E}_\gamma} \sup_{\hat{a}, \hat{b}, \hat{\sigma}^2 \in S_{a,b}} \mathbb{E}\{L(\hat{c}(e), c)\} \leq (K-1) \exp\left\{ -\frac{1}{4} \frac{(\gamma K - 1)^2}{K(K-1)^2} \frac{n(a-b)^2}{\sigma^2} \right\}, \quad (3.5)$$

and

$$\mathbb{P}\left[\sup_{e \in \mathcal{E}_\gamma} \sup_{\hat{a}, \hat{b}, \hat{\sigma}^2 \in S_{a,b}} L(\hat{c}(e), c) > \exp\left\{ -\frac{1}{8} \frac{(\gamma K - 1)^2}{(K-1)^2} \frac{n}{K} \frac{(a-b)^2}{\sigma^2} \right\} \right] \\ \leq (K-1) \exp\left[-n \left\{ \frac{1}{8} \frac{(\gamma K - 1)^2}{K(K-1)^2} \frac{(a-b)^2}{\sigma^2} - C(n, \gamma) \right\} \right], \quad (3.6)$$

where $C(n,\gamma) = h(\gamma) + \kappa_\gamma(2n/K) + (1-\gamma)\log(K-1)$. *As long as*

$$\frac{1}{8}\frac{(\gamma K-1)^2}{K(K-1)^2}\frac{(a-b)^2}{\sigma^2} > C(n,\gamma)\,, \tag{3.7}$$

the pseudo-likelihood estimator is uniformly weakly consistent.

Theorem 1 holds both for $a > b$ and $a < b$. Further, when parameter values scale with the number of nodes n, the consistency result of Theorem 1 holds as long as

$$n\frac{(a_n-b_n)^2}{\sigma_n^2} \to \infty \qquad \text{when} \qquad n \to \infty\,. \tag{3.8}$$

Condition (3.7) in Theorem 1 can be viewed as a requirement on the minimum difference between the distributions of within-community and between-community edges. This is not a particularly strong requirement since the term $\kappa_\gamma(2n/K)$ in $C(n,\gamma)$ goes to zero as n grows and the binary entropy term is bounded by 1. Empirical studies in Section 4 confirm the method works well even when the difference in the means is small.

A recent result by Xu, Jog and Loh (2020) established the optimal error rate for the weighted SBM, for any clustering algorithm, as a function of the Renyi divergence of order 1/2 between the within- and between-community edge distributions. We compute this bound explicitly for our model (3.2), with $N(a,\sigma^2)$ and $N(b,\sigma^2)$ distributions for the within- and between-community edges, respectively. In this case, the lower bound from Xu, Jog and Loh (2020), for the balanced communities case, is given by

$$\exp\left[-\{1+o(1)\}\frac{n}{K}\frac{(a-b)^2}{4\sigma^2}\right], \tag{3.9}$$

and our upper bound (3.5) matches this theoretical lower bound up to a constant. Moreover, by (3.6)

$$\lim_{n\to\infty}\mathbb{P}\left[\sup_{e\in\mathcal{E}_\gamma}\sup_{\hat{a},\,\hat{b},\,\hat{\sigma}^2\in S_{a,b}} L\{\hat{c}\,(e),c\} > \exp\left\{-\frac{1}{8}\frac{(\gamma K-1)^2}{(K-1)^2}\frac{n}{K}\frac{(a-b)^2}{\sigma^2}\right\}\right] = 0\,, \tag{3.10}$$

when $(1/8)\{(\gamma K-1)^2/K(K-1)^2\}\{(a-b)^2/\sigma^2\} > C(n,\gamma)$. This implies that the pseudo-likelihood algorithm achieves the optimal rate up to a constant.

3.2. Unbalanced communities

The case of unbalanced communities is substantially more complicated, because now both the number of nodes and the proportion of correct initial labels in each community affect the performance. To keep the technical details manageable and focus on understanding trade-offs, we limit our study of the unbalanced case to $K = 2$.

Let $n_k = \sum_{i=1}^{n} \mathbb{1}\{c_i = k\}$ be the number of nodes in community k, $k = 1, 2$, and let $\pi_k = n_k/n$. Assume that the initial labeling $c_0 \in \{1,2\}^n$, up to a permutation of labels, matches $\gamma_1 n_1$ labels in community 1 and $\gamma_2 n_2$ labels in community 2, or in other words, the initial label vector belongs to the set

$$\mathcal{E}_{\gamma_1,\gamma_2} = \left\{ e \in \{1,2\}^n : \sum_{i=1}^{n} \mathbb{1}\{e_i = k, c_i = k\} = \gamma_k n_k, k = 1, 2 \right\}. \tag{3.11}$$

For $e \in \mathcal{E}_{\gamma_1,\gamma_2}$, the confusion matrix R is given by

$$R(e) = \begin{pmatrix} \gamma_1 \dfrac{n_1}{n} & (1-\gamma_2)\dfrac{n_2}{n} \\ (1-\gamma_1)\dfrac{n_1}{n} & \gamma_2 \dfrac{n_2}{n} \end{pmatrix} = \begin{pmatrix} \gamma_1\pi_1 & (1-\gamma_2)\pi_2 \\ (1-\gamma_1)\pi_1 & \gamma_2\pi_2 \end{pmatrix}. \tag{3.12}$$

For any $e \in \mathcal{E}_{\gamma_1,\gamma_2}$, observe that the number of nodes in each community can be expressed as

$$\begin{aligned} \tilde{n}_1 &= \sum_{i=1}^{n} \mathbb{1}\{e_i = 1\} = \gamma_1 n_1 + (1-\gamma_2) n_2, \\ \tilde{n}_2 &= \sum_{i=1}^{n} \mathbb{1}\{e_i = 2\} = (1-\gamma_1) n_1 + \gamma_2 n_2. \end{aligned}$$

The corresponding proportions of nodes in each community of $e \in \mathcal{E}_{\gamma_1,\gamma_2}$ are then $\tilde{\pi}_1 = \gamma_1\pi_1 + \pi_2(1-\gamma_2)$ and $\tilde{\pi}_2 = (1-\gamma_1)\pi_1 + \pi_2\gamma_2$. Conditional on the true labels c, the proportion of nodes in each community defined by e depends only on the known matching proportions γ_1 and γ_2, and on the true proportions π_1 and π_2. Define the quantities

$$\begin{aligned} \beta_1 &= \tilde{\pi}_2 \left\{ (1-\gamma_2)\pi_2 - \gamma_1\pi_1 \right\}, \\ \beta_2 &= \tilde{\pi}_1 \left\{ (1-\gamma_1)\pi_1 - \gamma_2\pi_2 \right\}. \end{aligned} \tag{3.13}$$

For parameter estimates $\hat{a}, \hat{b}, \hat{\sigma}^2 \in \hat{P}_{a,b,\sigma^2}$, define the function

$$\begin{aligned} F(x,y) = &(-2x + \hat{a} + \hat{b}) \left\{ \beta_1\gamma_1\pi_1 - \beta_2(1-\gamma_1)\pi_1 \right\} \\ &+(-2y + \hat{a} + \hat{b}) \left\{ \beta_1(1-\gamma_2)\pi_2 - \beta_2\gamma_2\pi_2 \right\}. \end{aligned} \tag{3.14}$$

By (3.13), the values of β_1 and β_2 depend only on γ_1, γ_2, π_1 and π_2, and we suppress this dependence to simplify notation. Theorem 2 gives an upper bound on the probability of misclassification of each node i in terms of $\gamma_1, \gamma_2, \hat{a}, \hat{b}, \hat{\sigma}^2, \pi_1$ and π_2. The proof of this theorem relies on bounding the probability of misclassification of a node by bounding the probability that the label update, given by (3.3), chooses the wrong label. We will express this bound in terms of two quantities, t_1 and t_2, that depend on all the initial values in a fairly complicated way, and on how well they approximate the true values.

Theorem 2 (Unbalanced case). *Initialize the algorithm with a label vector* $e \in \mathcal{E}_{\gamma_1,\gamma_2}$ *and the corresponding parameter estimates* $\hat{a}, \hat{b}, \hat{\sigma}^2 \in S_{a,b}$. *Let*

$$t_1 = \frac{2\hat{\sigma}^2\tilde{\pi}_1\tilde{\pi}_2}{n(\hat{a}-\hat{b})}\log\left(\frac{\tilde{\pi}_1}{\tilde{\pi}_2}\right) + F(a,b), \quad t_2 = \frac{2\hat{\sigma}^2\tilde{\pi}_1\tilde{\pi}_2}{n(\hat{a}-\hat{b})}\log\left(\frac{\tilde{\pi}_1}{\tilde{\pi}_2}\right) + F(b,a). \quad (3.15)$$

If $a > b$, $t_1 \geq 0$ *and* $t_2 < 0$ *or if* $a < b$, $t_1 < 0$ *and* $t_2 \geq 0$, *the probability of misclassification for any node* $i \in \{1,\ldots,n\}$, *conditionally on* $\hat{a}, \hat{b}$ *and* $\hat{\sigma}^2$, *is given by*

$$\mathbb{P}(\hat{c}_i(e) \neq 1 | c_i = 1) \leq \exp\bigg(-\frac{n}{8\sigma^2\tau^2}t_1^2\bigg), \quad (3.16)$$

$$\mathbb{P}(\hat{c}_i(e) \neq 2 | c_i = 2) \leq \exp\bigg(-\frac{n}{8\sigma^2\tau^2}t_2^2\bigg), \quad (3.17)$$

where $\tau^2 = \beta_1^2\pi_1 + \beta_2^2\pi_2$.

The sign assumptions on t_1 and t_2 are technical assumptions needed to apply the Gaussian tail bounds used in the proof to the correct tail; the bounds themselves only depend on t_1^2 and t_2^2. For the balanced case, the symmetry ensures that the assumption of alignment of signs of $\hat{a}-\hat{b}$ and $a-b$ made in (3.4) is sufficient. For the unbalanced case, we did not find a simple way to express this assumption in terms of how the initial values relate to the parameters, but intuitively it also reflects the quality of starting values, and thus can be expected to hold if the starting values are good enough.

The result (3.15) implies that the probabilities of misclassification (3.16) and (3.17) go to zero as n grows. Note that in this analysis K remains fixed while n grows. The imbalance between community sizes does not matter in this situation asymptotically, though in practice estimation does suffer from unbalanced community sizes, as the simulations studies in Section 4 show.

To get more intuition about the tradeoffs indicated by Theorem 2, take a $\delta_n > 0$ such that $|\hat{a}_n - a_n| < \delta_n$, $|\hat{b}_n - b_n| < \delta_n$ and assume the variance is known, with $\hat{\sigma}_n^2 = \sigma_n^2$. We can then rewrite the bound (3.16) in Theorem 2 as

$$\exp\Bigg[-\frac{n(a_n-b_n)^2}{8\tau^2\sigma_n^2} \times \Bigg\{\frac{2{\sigma_n}^2\tilde{\pi}_1\tilde{\pi}_2}{n(a_n-b_n+2\delta_n)(a_n-b_n)}\log\left(\frac{\tilde{\pi}_1}{\tilde{\pi}_2}\right) + C_1^{\gamma,\pi} - \frac{2\delta C_2^{\gamma,\pi}}{(a_n-b_n)}\Bigg\}^2\Bigg],$$

where $C_1^{\gamma,\pi} = \beta_1\pi_2 + \beta_2\pi_1 - (\beta_1+\beta_2)(\pi_1\gamma_1 + \pi_2\gamma_2)$ and $C_2^{\gamma,\pi} = \beta_1\pi_2 - \beta_2\pi_1 + (\beta_1+\beta_2)(\pi_1\gamma_1 - \pi_2\gamma_2)$. If

$$\frac{\sigma_n^2}{n} \to 0 \quad \text{and} \quad \frac{n(a_n-b_n)^2}{\sigma_n^2} \to \infty,$$

then the probabilities of misclassification (3.16) and (3.17) go to zero as $n \to \infty$.

Combining the two bounds of Theorem 2, we can conclude that the expected error of $\hat{c}(e)$, for any $e \in \mathcal{E}_{\gamma_1,\gamma_2}$, satisfies

$$\mathbb{E}\{L(\hat{c}(e), c)\} \leq \pi_1 \exp\left(-\frac{n}{8\sigma^2\tau^2}t_1^2\right) + \pi_2 \exp\left(-\frac{n}{8\sigma^2\tau^2}t_2^2\right). \tag{3.18}$$

In particular, when $\pi = (1/2, 1/2)$ and $\gamma_1 = \gamma_2 = \gamma$, we have $\tilde{\pi}_1 = \tilde{\pi}_2 = 1/2$, $\beta_1 = \beta_2 = (1-2\gamma)/4$ and $-t_2 = t_1 = (1-2\gamma)^2(a-b)/4$. Thus, the expected error given by (3.18) matches, up to a constant, the expected error (3.5) obtained for the balanced case.

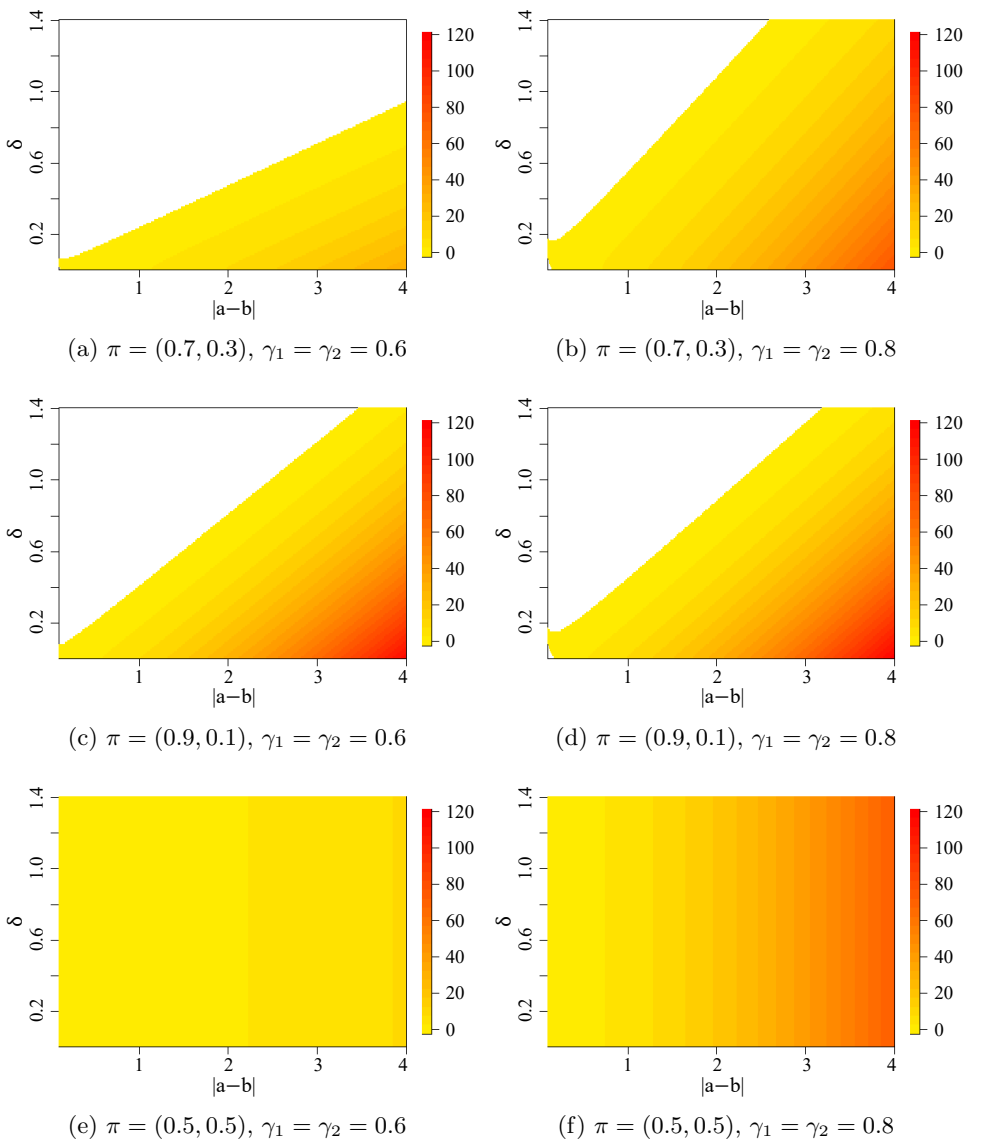
Figure 1 shows the negative logarithm of the bound on the expected error (higher values means tighter bound) as a function of the signal strength $|a-b|$, the difference between the within- and between-community means, and the quality of the initial parameter estimates δ. We fix $n = 100$ and the variance $\hat{\sigma}^2 = 1$. We plot the value of the log-bound as a heatmap, for the set of parameters that satisfy conditions (3.15). As one would expect, the error decreases as δ decreases (better initial value for the parameters), as $|a-b|$ increases (easier problem), and as the proportion of matches γ increases (better initial value for the labels). We also see that a better initial value for the labels (higher γ) leads to a larger set of parameters satisfying the conditions, and to lower errors. Note, however, that the error bound is a complicated function of all the relevant parameters (π, γ, δ, etc.), and thus the overall effect of any one parameter may be hard to isolate; for example, the comparison of (a) to (b) may seem counter-intuitive, because the bound is tighter for the less balanced case, but this is an artifact of a complex dependence on the initial values of the quantities t_1 and t_2, and the fact the bound is not optimal. In our empirical results, the errors are higher in less balanced cases, all other things being equal.

4. Empirical Evaluation on Simulated Networks

Our empirical investigations focus on two goals: understanding how the various parameters of the problem affect the performance of the pseudo-likelihood algorithm (PL), and comparing it to other ways of estimating communities from weighted networks. We simulate networks from the model (3.2) with $K = 3$ and other parameters as specified below. The number of iterations of the PL algorithm is fixed at 20. Performance is evaluated by the error in community assignments defined in (3.1), averaged over 100 replications.

4.1. Impact of problem difficulty and initial values

Figure 2 shows the performance of the PL algorithm, as implemented in Algorithm 1, as a function of several quantities that control the difficulty of the problem: the signal strength $|a-b|$ (with fixed $\sigma^2 = 1$), and several values of the number of nodes n and the correct fraction of initial labels γ. The results are



(a) $\pi = (0.7, 0.3)$, $\gamma_1 = \gamma_2 = 0.6$

(b) $\pi = (0.7, 0.3)$, $\gamma_1 = \gamma_2 = 0.8$

(c) $\pi = (0.9, 0.1)$, $\gamma_1 = \gamma_2 = 0.6$

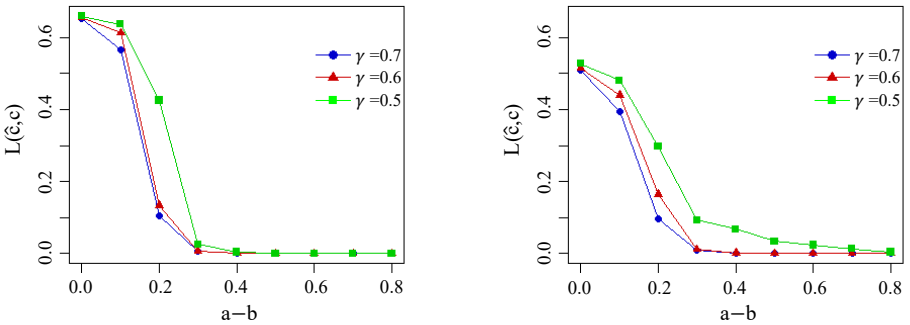
(d) $\pi = (0.9, 0.1)$, $\gamma_1 = \gamma_2 = 0.8$

(e) $\pi = (0.5, 0.5)$, $\gamma_1 = \gamma_2 = 0.6$

(f) $\pi = (0.5, 0.5)$, $\gamma_1 = \gamma_2 = 0.8$

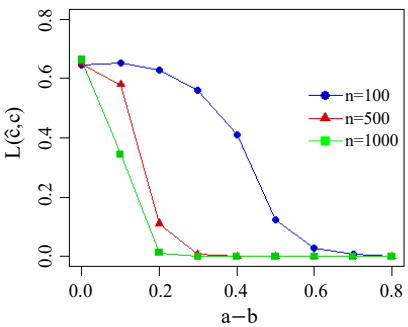
Figure 1. The negative logarithm of the upper bound of the expected error (3.18) when $n = 100$, $\hat{\sigma}^2 = \sigma^2 = 1$, $|\hat{a}_n - a_n| < \delta_n$ and $|\hat{b}_n - b_n| < \delta_n$ for different proportion of matches γ_1 and γ_2.

intuitive: the error rate decreases as the signal gets stronger (larger $|a - b|$), the initial value improves (larger γ), and the number of nodes grows. An encouraging finding is that these results are not especially sensitive to γ, which we cannot easily control in practice. When there is little difference between the means a and b, the error rate gets close to random guessing (2/3 in this case, as $K = 3$), as one would expect. The comparison between balanced and unbalanced community sizes is not straightforward with $K = 3$, but overall the balanced case is easier,

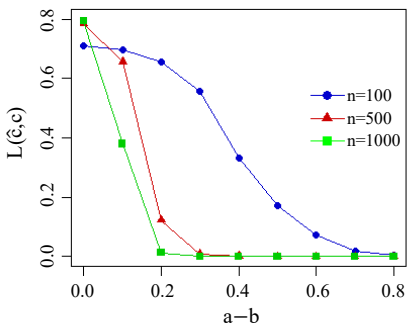


(a) $n = 500$ and $\pi = (1/3, 1/3, 1/3)$.

(b) $n = 500$ and $\pi = (0.2, 0.5, 0.3)$.



(c) $\gamma = 0.7$ and $\pi = (1/3, 1/3, 1/3)$.



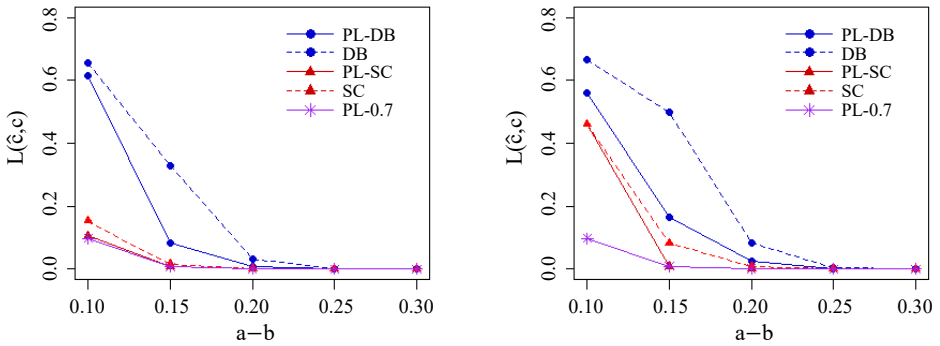
(d) $\gamma = 0.7$ and $\pi = (0.2, 0.5, 0.3)$.

Figure 2. Error of the PL algorithm as a function of the difference between the means $|a-b|$, for several values of the correct proportion γ and the number of nodes n, averaged over 100 replications. The variance is fixed at $\sigma^2 = 1$.

as one would expect. The higher errors for the balanced case when the signal is very low are an artifact of the fact that putting all nodes into the single largest community gives the error rate of 66% for the balanced case but only 50% for the unbalanced case.

We compare two choices of the initial labels for the PL algorithm. One is spectral clustering (SC), implemented as described in Lei and Rinaldo (2015) for unweighted networks, applied directly to the matrix of weights W. The second option is the discretization-based algorithm (DB) of Xu, Jog and Loh (2020), which first discretizes the matrix of weights and then applies clustering. The DB method itself depends on the choice of the discretization level; we followed the DB authors' recommendation and set it to $\lfloor 0.4(\log\log n)^4 \rfloor$.

Figure 3 shows that the PL algorithm improves substantially upon both initial values. The SC algorithm is generally more accurate than DB, and thus leads to better PL solutions when used as the initial value. For comparison, the PL algorithm starting with $\gamma = 0.7$ correct labels is included in all scenarios. Spectral clustering is known to favor balanced solutions, and thus for unbalanced



(a) $\pi = (1/3, 1/3, 1/3)$ (b) $\pi = (0.2, 0.5, 0.3)$

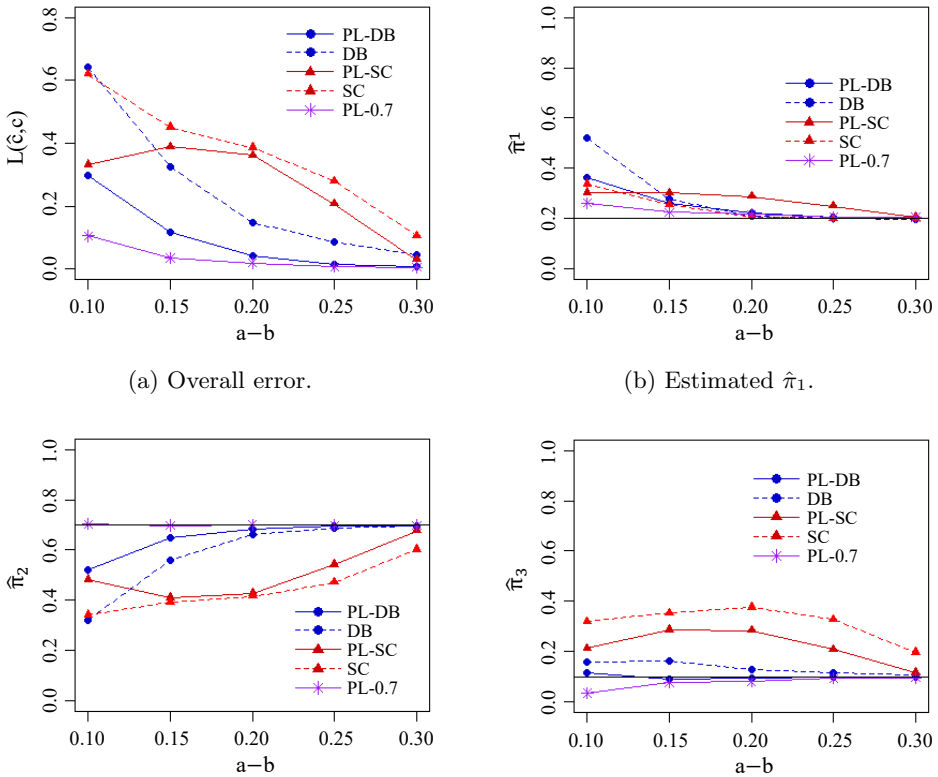
Figure 3. Balanced vs. unbalanced community sizes. Overall error for PL initialized with $\gamma = 0.7$, SC, DB, and PL initialized with either SC (PL-SC) or DB (PL-DB). For both settings, $\sigma^2 = 0.5$, $n = 1000$, and results are averaged over 100 replications.

networks with a small community containing only 10% of the nodes (Figure 4a), the SC estimates are the worst among all the methods, but even then the PL method is able to improve somewhat on the initial value provided by spectral clustering.

4.2. Robustness to the Gaussian assumption

To investigate robustness to the Gaussian assumption, we also considered the case of heavier-tailed edge weights. We generated the weights from a mixture of Gaussian and a noncentral t distribution $t_{\mu,d}$ with d degrees of freedom and the noncentrality parameter μ. The within-community and between-community edge weights are generated by the mixture $\alpha\mathcal{N}(0.2, 0.25) + (1 - \alpha)t_{0.2,4}$ and $\alpha\mathcal{N}(0, 0.25) + (1 - \alpha)t_{0,4}$, respectively. Figure 5a illustrates the within and between densities for $\alpha = 0.4$. Figure 5a shows that the PL method performs well even for small values of α, when the edge weights are heavy-tailed.

Figure 5b illustrates a different violation of the distributional assumption, with within-community edge weights generated from a bimodal mixture of Gaussians $0.5\mathcal{N}(-0.3, 0.25) + 0.5\mathcal{N}(b, 0.25)$, and the between-community edge weights are $\mathcal{N}(0, 0.25)$. We vary b from 0.3 to 0.6, with the mean of within-community edge weights varying from 0 to 0.15, while the between-community mean is always 0. As expected, the overall error of the PL estimates decreases as the difference between the within and between means increases. The DB algorithm has an advantage in this case because the discretization step helps overcome this departure from normality, which was also pointed out by the authors.



(a) Overall error.

(b) Estimated $\hat{\pi}_1$.

(c) Estimated $\hat{\pi}_2$.

(d) Estimated $\hat{\pi}_3$.

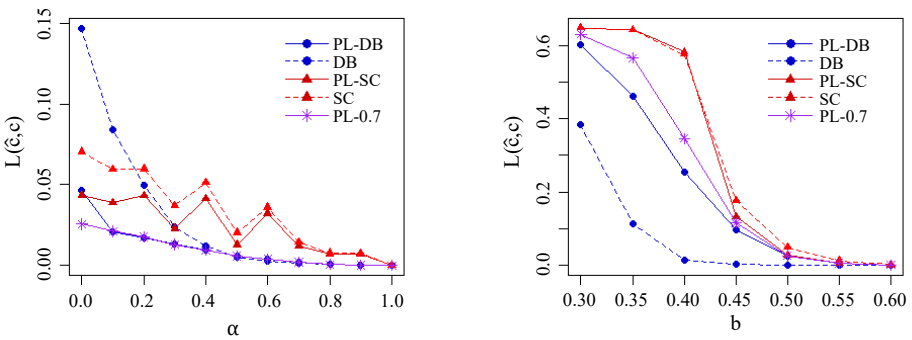
Figure 4. Unbalanced case, $\pi = (0.2, 0.7, 0.1)$. Overall error and estimated proportion of nodes for PL initialized with $\gamma = 0.7$, SC, DB, and PL initialized with either SC (PL-SC) or DB (PL-DB). For every setting, $\sigma^2 = 0.5$, $n = 1000$, and results are averaged over 100 replications.

4.3. Estimating the number of communities

In practice, the number of communities K is not known. While developing an estimator for K in this setting is outside the scope of this work, we empirically investigate the algorithm's performance when K has to be estimated. Following the penalized likelihood model selection approach for binary networks proposed by Wang and Bickel (2017), who proved its consistency for the binary SBM setting, we use a similar penalty, $n \log(n)$ times the number of parameters, and estimate the number of communities by

$$\hat{K} = \underset{k}{\operatorname{argmax}} \left\{ \log p(w, \hat{z} \mid \hat{\pi}, \hat{B}, \hat{\Sigma}) - k(k+1) n \log n \right\}, \tag{4.1}$$

where $p(w, \hat{z} \mid \hat{\pi}, \hat{B}, \hat{\Sigma})$ is the maximum complete likelihood described in (2.3), obtained by plugging in the estimated labels $\hat{z}$ and the parameter estimates obtained by the pseudo-likelihood method. The number of parameters in this



(a) Heavy tails (b) Bimodal

Figure 5. Overall error for PL initialized with $\gamma = 0.7$, SC, DB, and PL initialized with either SC (PL-SC) or DB (PL-DB). (a) Edge weights are generated from a mixture of Gaussian and a noncentral t-distribution with mixture probability α and $n = 1000$. (b) Within-community edge weights are generated from a mixture of Gaussians and between-community weights from a Gaussian distribution.

case is the number of mean and variance parameters of the edge weights.

First, we simply evaluate whether the estimator $\hat{K}$ is correct; the proportion of times $\hat{K} = K$ is shown in Table 1. In all of the settings we consider in the table, the estimated $\hat{K}$ is either correct and equal to $K = 3$, or else $\hat{K} = 2$. This is common for all estimators of the number of communities—in more difficult settings they underestimate the true K. To compare the true communities with the communities estimated with $\hat{K}$, we also show, in Figure 6, the Normalized Mutual Information (NMI) between the two communities assignments (Yao, 2003). NMI is well defined even for partitions with different number of communities and takes values between 0 (the two partitions are independent random assignments) and 1 (perfect match). For the hardest setting for signal strength $b = 1.3$, we get $\hat{K} = 2$ while the true $K = 3$. Figure 6 shows that as the number of nodes n grows, NMI approaches 1 as the $\hat{K}$ converges to the true value, but it takes much longer in more challenging settings.

4.4. Running times comparison

Finally, we compare the running times of different methods in Figure 7a, on the same standard single core. The time reported for the PL method does not include the time needed to generate the initial labeling. As n grows, the PL algorithm becomes cheaper to compute than SC and DB themselves, suggesting it is an effective tool for improving their performance, increasing statistical accuracy at little computational cost.

Table 1. Proportion (out of 100 replications) that $\hat{K} = 3$ is correctly estimated by the penalized log maximum complete likelihood, for different values of b and the number of nodes n, with fixed $a = 2$, $\sigma^2 = 0.5$. In all of the settings considered, the estimator returns either $\hat{K} = 3$ or $\hat{K} = 2$.

	b \ n	700	800	900	1,000	1,100	1,200
$\pi = (1/3, 1/3, 1/3)$	1.1	1.00	1.00	1.00	1.00	1.00	1.00
	1.2	0.52	1.00	1.00	1.00	1.00	1.00
	1.3	0.00	0.00	0.38	1.00	1.00	1.00
$\pi = (0.2, 0.5, 0.3)$	1.1	0.00	0.00	0.06	0.33	0.89	1.00
	1.2	0.00	0.00	0.00	0.00	0.00	0.19
	1.3	0.00	0.00	0.00	0.00	0.00	0.00

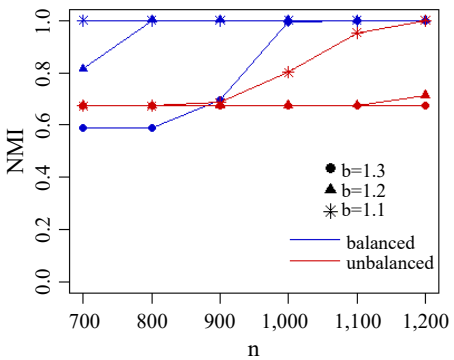
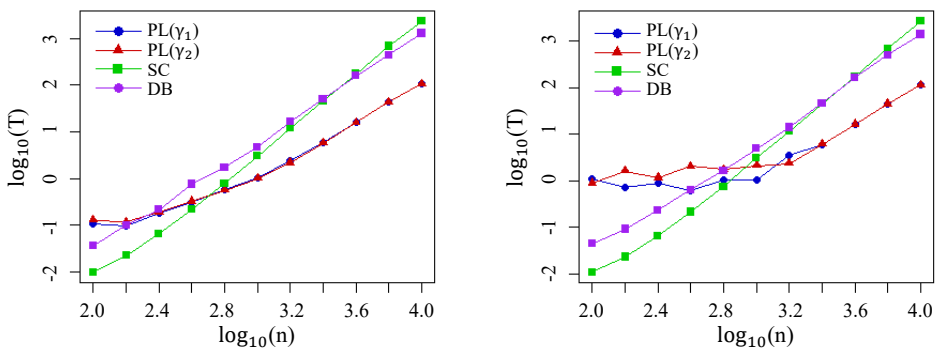


Figure 6. The NMI between true and estimated communities obtained using $\hat{K}$, averaged over 100 replications, for different values of b and the number of nodes n, while fixing $a = 2$, $\sigma^2 = 0.5$ and the true number of communities $K = 3$.



(a) $(a - b)/\sigma = 2$ (b) $(a - b)/\sigma = 0.2$

Figure 7. The running time T (in seconds) of the algorithms PL ($\gamma_1 = 0.9$ and $\gamma_2 = 0.5$), SC and DB as a function of the number of nodes n.

5. Application to fMRI Data

Here we apply the pseudo-likelihood method to the COBRE data (Aine et al., 2017), consisting of resting state fMRI brain images of 54 schizophrenic patients and 69 healthy patients. Each fMRI scan was processed following a standard pipeline at the time of data collection, and converted to a weighted graph with $n = 264$ nodes representing the regions of the brain; see Arroyo et al. (2019) for details. The edge weights represent functional connectivity between the brain regions, measured by Fisher-transformed correlations between the time series of blood oxygenation levels at the corresponding regions. Since the Fisher transform of the correlation coefficient is designed to make the distribution approximately normal, this is a natural application for the normally distributed edge weights model. We average the 69 weighted networks corresponding to healthy patients using the weighted network average method of Levin, Lodhia and Levina (2022) to obtain an estimate for the healthy population, and similarly for the schizophrenic patients, resulting in two "prototypical" weighted networks.

5.1. Community detection in fMRI brain networks

There are no ground truth communities in this problem, but we can still compare the healthy and the schizophrenic populations. We can also compare results from community detection to previously published known brain atlases such as Power et al. (2011). The true number of communities K is also unknown, and we simply vary K from 2 to 20, a range based on previous findings for fMRI connectivity networks. As before, we use both DB and SC as potential initial values for our pseudo-likelihood method. Using the same formula as in the simulation study, the discretization level for DB method is set to 10.

We start from comparing fitted likelihoods of different starting values and the PL solutions initialized with them, shown in Figure 8. While we plot these as a function of the number of community K for convenience, the values across different Ks are not directly comparable, since we are not applying any penalization for model complexity. The plots generally agree with what we saw in simulations: the PL algorithm finds a better fit than the initial value it starts from, and the DB method especially does not fit the data as well.

We also looked at how different solutions differ from each other and from their initial values. Figure 9 shows the proportion of nodes labeled differently (after finding the best permutation of community labels) between pairs of methods. We see similar patterns for the healthy and the schizophrenic populations, and while the two initial values are substantially different, the solutions of the PL method are closer, suggesting that it moves in the direction of a higher likelihood from both initial values, giving us more confidence in the PL solutions.

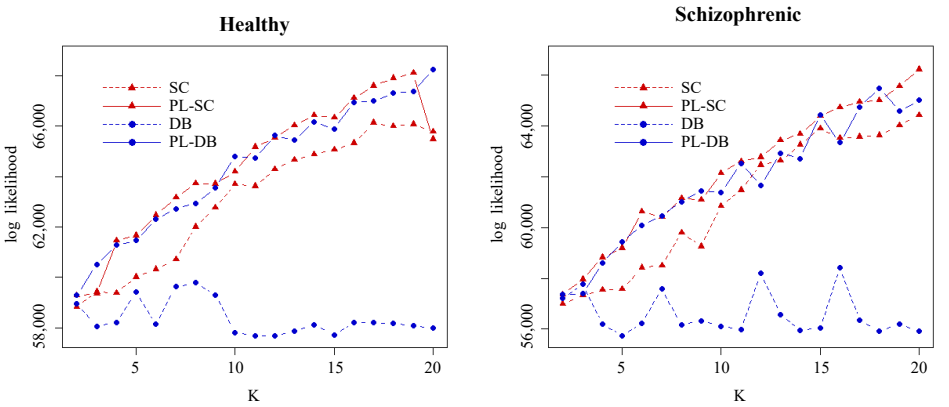


Figure 8. The log of the complete likelihood using the communities from spectral clustering (SC), discretization-based method (DB), and pseudo-likelihood with two initial values, PL-SC and PL-DB.

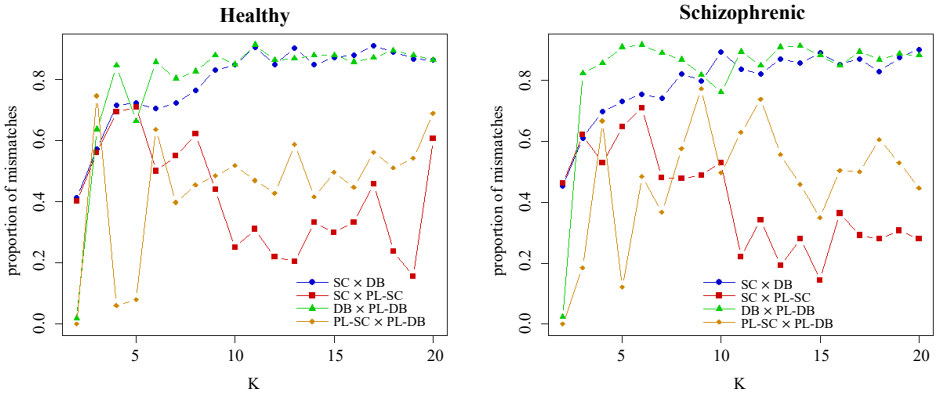


Figure 9. Proportion of mismatched nodes between different pairs of methods as a function of the number of communities K.

5.2. Comparative analysis of brain systems

We also compared our solutions to the Power parcellation (Power et al., 2011), an assignment of the same 264 ROIs (nodes) to 14 functional brain systems (regions), one of which is labeled "uncertain" and others correspond to known brain functions, as described in Table 2. This parcellation was obtained from a different dataset which included only healthy subjects.

Fixing $K = 14$ to match the Power parcellation, we estimated 14 communities for both populations by the PL algorithm using SC as the initial value. The estimated communities for the healthy population were relabeled to match their numbers as closely as possible to the Power regions using the Hungarian algorithm (Kuhn, 1955). Table 3 compares the parcellation estimated from the healthy population (H1–H14) to both the Power regions (P1–P14) and the parcellation

Table 2. Brain regions in the Power parcellation.

Region	Function	Nodes	Region	Function	Nodes
P1	Sensory/somatomotor Hand	30	P8	Fronto-pariental Task Control	25
P2	Sensory/somatomotor Mouth	5	P9	Salience	18
P3	Cingulo-opercular Task Control	14	P10	Subcortical	13
P4	Auditory	13	P11	Ventral attention	9
P5	Default mode	58	P12	Dorsal attention	11
P6	Memory retrieval	5	P13	Cerebellar	4
P7	Visual	31	P14	Uncertain	28

Table 3. Proportion of common nodes of the estimated regions of the healthy network (H1–H14) with the correspondent Power parcellation regions (P1–P14) and the estimated regions of the schizophrenic network (S1–S14).

Region	Region	Region	Region	Region	Region
H1	P1 (0.70)	S1 (0.59)	H8	P7 (0.48)	S8 (1.00)
H2	P5 (0.61)	S5 (0.44)	H9	P8 (0.61)	S9 (1.00)
H3	P1,P3 (0.40)	S11 (0.40)	H10	P10 (0.35)	S10 (0.67)
H4	P3,P4,P5 (0.26)	S4 (0.95)	H11	P5 (0.65)	S11 (0.59)
H5	P5 (1.00)	S5 (0.85)	H12	P12 (0.69)	S12 (1.00)
H6	P5 (0.62)	S6 (0.44)	H13	P5 (0.33)	S13 (0.67)
H7	P7 (0.69)	S7 (0.61)	H14	P14 (1.00)	S14 (1.00)

estimated for the schizophrenic population (S1–S14), listing the region(s) with the highest overlap for each of the H1–H14 and the proportion of shared nodes. The Sankey diagram in Figure 10 shows the correspondence between the healthy and Power parcellation for reference, and the correspondence between the healthy and the schizophrenic populations. We see that only a few regions are strongly different between the healthy and the schizophrenic parcellations; in particular, H2 and H11, which both appear to split off from P5, which is the default mode network (DMN). The DMN has been implicated in schizophrenia previously (Broyd et al., 2009; Öngür et al., 2010), and we have observed in previous work (Kim, Kessler and Levina, 2023) that different parts of the DMN seem to play different roles, and so the splitting into multiple parts makes sense. We also note that the most stable Power regions, obtained from different patients, seem to correspond to two attention regions: visual and fronto-parietal task control. This could indicate that these are the strongest communities. This is, of course, exploratory analysis, and making formal inferences requires further study.

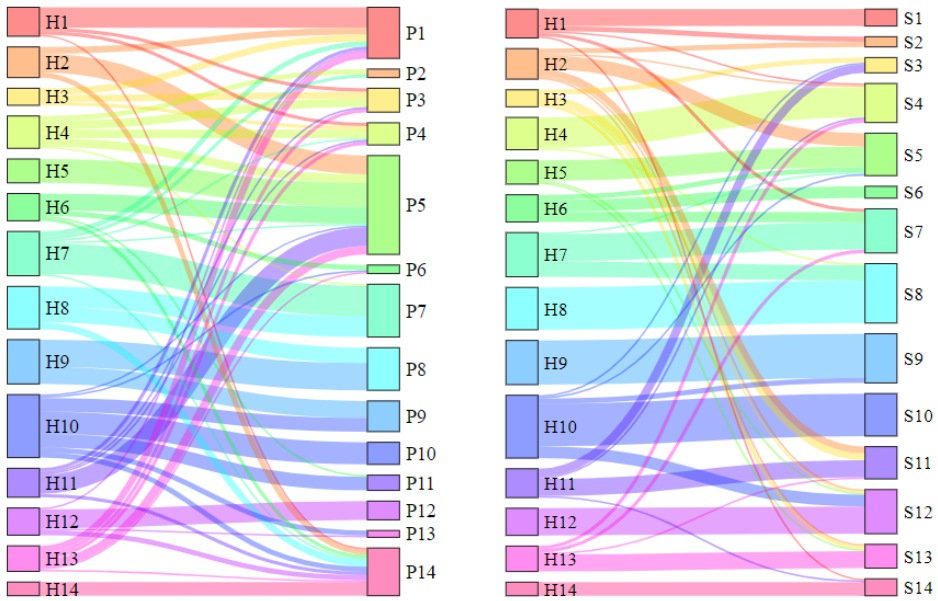


Figure 10. A Sankey diagram comparing the communities obtained by the PL-SC algorithm for the healthy population (H1–H14) with the Power regions (P1–P14) and the communities estimated for the schizophrenic patients (S1–S14).

6. Discussion

In this paper, we proposed a pseudo-likelihood method for community detection on weighted networks, a much less studied type of networks than binary but just as common in practice, or even more common, once we account for the fact that many binary networks are obtained by thresholding weighted networks obtained directly from the data. Like many other community detection methods, we provided the analysis under the weighted stochastic block model, but empirically the algorithm appears reasonably robust to model misspecification. Our theoretical analysis shows that the proposed method achieves the optimal error rate up to a constant, and illustrates the trade-offs between the means, variances, and community sizes. Like all iterative algorithms, our algorithm depends on how good the initial value is. We were able to quantify this dependence in theory, and showed empirically that the algorithm is robust to the choice of initial value. In particular, the initial value provided by spectral clustering is a fast and reliable way to initialize pseudo-likelihood, which can still improve upon it substantially.

There are many directions in which this work can be taken further. We only considered theoretical analysis for homogenous models, where all within-communities edge distributions are the same, and between-communities edge

distributions are also the same. This simplification allows for bounds that make the within- and between- tradeoff explicit, which is arguably the main value of obtaining such bounds. The algorithm itself, however, is equally applicable to heterogeneous models, and theoretical properties under that scenario remain to be investigated. Another useful extension would be to incorporate edge distributions with a point mass at zero, so that the network can be truly sparse. More generally, relaxing parametric assumptions on edge distributions and developing more general versions of the pseudo-likelihood approach to community detection on weighted networks would increase the range of applications where the method could be applied not just based on empirical evidence, but with provable guarantees. Finally, further investigation of model selection options and of theoretical properties of the penalized likelihood model selection method we proposed remain an open question.

Supplementary Material

The online Supplementary Material contains the proof of Theorems 1 and 2, and a version of Theorem 1 without the assumption of equal proportions of true labels in each community on the initial value.

Acknowledgments

The authors thank Min Xu, Varun Jog and Po-Ling Loh for sharing their code with us, and Keith Levin for sharing the average networks from the COBRE data. This research was supported by the São Paulo Research Foundation (FAPESP) grant 2018/18115-2 to A.C. and by the NSF grants DMS-1916222, 2052918, and 2210439 to E.L. This research was conducted while A.C. was a Visiting Postdoctoral Researcher in the Department of Statistics at the University of Michigan, and she thanks the department for its hospitality and support.

References

Abbe, E. (2018). Community detection and stochastic block models: Recent developments. *Journal of Machine Learning Research* **18**, 1–86.

Aicher, C., Jacobs, A. Z. and Clauset, A. (2014). Learning latent block structure in weighted networks. *Journal of Complex Networks* **3**, 221–248.

Aine, C., Bockholt, H. J., Bustillo, J. R., Cañive, J. M., Caprihan, A., Gasparovic, C. et al. (2017). Multimodal neuroimaging in schizophrenia: Description and dissemination. *Neuroinformatics* **15**, 343–364.

Airoldi, E. M., Blei, D. M., Fienberg, S. E. and Xing, E. P. (2008). Mixed membership stochastic blockmodels. *Journal of Machine Learning Research* **9**, 1981–2014.

Amini, A. A., Chen, A., Bickel, P. J. and Levina, E. (2013). Pseudo-likelihood methods for community detection in large sparse networks. *The Annals of Statistics* **41**, 2097–2122.

Arroyo, J., Athreya, A., Cape, J., Chen, G., Priebe, C. E. and Vogelstein, J. T. (2021). Inference for multiple heterogeneous networks with a common invariant subspace. *Journal of Machine Learning Research* **22**, 1–49.

Arroyo, J., Kessler, D., Levina, E. and Taylor, S. F. (2019). Network classification with applications to brain connectomics. *The Annals of Applied Statistics* **13**, 1648–1677.

Avrachenkov, K., Dreveton, M. and Leskelä, L. (2020). Community recovery in non-binary and temporal stochastic block models. *arXiv:2008.04790*.

Balakrishnan, S., Xu, M., Krishnamurthy, A. and Singh, A. (2011). Noise thresholds for spectral clustering. In *Advances in Neural Information Processing Systems* (Edited by J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira and K. Weinberger). Curran Associates, Inc.

Bhattacharyya, S. and Chatterjee, S. (2018). Spectral clustering for multiple sparse networks: I. *arXiv:1805.10594*.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R. and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, P10008.

Broyd, S. J., Demanuele, C., Debener, S., Helps, S. K., James, C. J. and Sonuga-Barke, E. J. (2009). Default-mode brain dysfunction in mental disorders: A systematic review. *Neuroscience and Biobehavioral Reviews* **33**, 279–296.

Chen, Y. and Xu, J. (2016). Statistical-computational tradeoffs in planted problems and submatrix localization with a growing number of clusters and submatrices. *Journal of Machine Learning Research* **17**, 1–57.

Erdös, P. and Rényi, A. (1960). On the evolution of random graphs. *Publication of the Mathematical Institute of the Hungarian Academy of Sciences* **5**, 17–60.

Hajek, B., Wu, Y. and Xu, J. (2018). Submatrix localization via message passing. *Journal of Machine Learning Research* **18**, 1–52.

Heimlicher, S., Lelarge, M. and Massoulié, L. (2012). Community detection in the labelled stochastic block model. *arXiv:1209.2910*.

Holland, P. W., Laskey, K. B. and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social Networks* **5**, 109–137.

Karrer, B. and Newman, M. E. (2011). Stochastic blockmodels and community structure in networks. *Physical Review E* **83**, 016107.

Kim, Y., Kessler, D. and Levina, E. (2023). Graph-aware modeling of brain connectivity networks. *The Annals of Applied Statistics* **17**, 2095–2117.

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics Quarterly* **2**, 83–97.

Latouche, P., Birmelé, E. and Ambroise, C. (2011). Overlapping stochastic block models with application to the French political blogosphere. *The Annals of Applied Statistics* **5**, 309–336.

Le, C. M., Levin, K. and Levina, E. (2018). Estimating a network from multiple noisy realizations. *Electronic Journal of Statistics* **12**, 4697–4740.

Lei, J. and Rinaldo, A. (2015). Consistency of spectral clustering in stochastic block models. *The Annals of Statistics* **43**, 215–237.

Lelarge, M., Massoulié, L. and Xu, J. (2015). Reconstruction in the labelled stochastic block model. *IEEE Transactions on Network Science and Engineering* **2**, 152–163.

Levin, K., Lodhia, A. and Levina, E. (2022). Recovering shared structure from multiple networks with unknown edge distributions. *Journal of Machine Learning Research* **23**, 1–48.

Ma, Z. and Wu, Y. (2015). Computational barriers in minimax submatrix detection. *The Annals of Statistics* **43**, 1089–1116.

MacDonald, P. W., Levina, E. and Zhu, J. (2021). Latent space models for multiplex networks with shared structure. *Biometrika* **109**, 683–706.

Öngür, D., Lundy, M., Greenhouse, I., Shinn, A. K., Menon, V., Cohen, B. M. et al. (2010). Default mode network abnormalities in bipolar disorder and schizophrenia. *Psychiatry Research* **183**, 59–68.

Power, J. D., Cohen, A. L., Nelson, S. M., Wig, G. S., Barnes, K. A., Church, J. A. et al. (2011). Functional network organization of the human brain. *Neuron* **72**, 665–678.

Reichardt, J. and Bornholdt, S. (2006). Statistical mechanics of community detection. *Physical Review E* **74**, 016110.

Rosvall, M. and Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences* **105**, 1118–1123.

Tang, W., Lu, Z. and Dhillon, I. S. (2009). Clustering with multiple graphs. In *2009 Ninth IEEE International Conference on Data Mining*, 1016–1021.

Wang, J., Zhang, J., Liu, B., Zhu, J. and Guo, J. (2023). Fast network community detection with profile-pseudo likelihood methods. *Journal of the American Statistical Association* **118**, 1359–1372.

Wang, S., Arroyo, J., Vogelstein, J. T. and Priebe, C. E. (2021). Joint embedding of graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **43**, 1324–1336.

Wang, Y. X. R. and Bickel, P. J. (2017). Likelihood-based model selection for stochastic block models. *The Annals of Statistics* **45**, 500–528.

Xu, M., Jog, V. and Loh, P.-L. (2020). Optimal rates for community estimation in the weighted stochastic block model. *The Annals of Statistics* **48**, 183–204.

Yao, Y. Y. (2003). Information-theoretic measures for knowledge discovery and data mining. In *Entropy Measures, Maximum Entropy Principle and Emerging Applications*, 115–136. Springer, Berlin, Heidelberg.

Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of Anthropological Research* **33**, 452–473.

Zhang, Y., Levina, E. and Zhu, J. (2020). Detecting overlapping communities in networks using spectral methods. *SIAM Journal on Mathematics of Data Science* **2**, 265–283.

(Received March 2023; accepted April 2024)