Network Estimation by Mixing: Adaptivity and More *

Tianxi Li

Department of Statistics, University of Virginia

Can M. Le

Department of Statistics, University of California, Davis

June 8, 2021

Abstract

Networks analysis has been commonly used to study the interactions between units of complex systems. One problem of particular interest is learning the network's underlying connection pattern given a single and noisy instantiation. While many methods have been proposed to address this problem in recent years, they usually assume that the true model belongs to a known class, which is not verifiable in most real-world applications. Consequently, network modeling based on these methods either suffers from model misspecification or relies on additional model selection procedures that are not well understood in theory and can potentially be unstable in practice. To address this difficulty, we propose a mixing strategy that leverages available arbitrary models to improve their individual performances. The proposed method is computationally efficient and almost tuning-free; thus, it can be used as an off-the-shelf method for network modeling. We show that the proposed method performs equally well as the oracle estimate when the true model is included as individual candidates. More importantly, the method remains robust and outperforms all current estimates even when the models are misspecified. Extensive simulation examples are used to verify the advantage of the proposed mixing method. Evaluation of link prediction performance on 385 real-world networks from six domains also demonstrates the universal competitiveness of the mixing method across multiple domains.

1 Introduction

Networks are widely used to represent complex interactions between entities, and network analysis has been an intensively studied topic in statistics, computer science, physics, social science, and many areas in recent years. Analyzing the structures of network data can render salient insights about relation formulation or interaction mechanisms between individuals. Network analysis has been used in various applications in science, engineering, and social studies, revealing many new aspects in these studies [Barabási and Albert, 1999, Steglich et al., 2006, Newman, 2010, Fortunato, 2010, Lü and Zhou, 2011].

*Both authors contributed equally to this work.

Due to the noisy nature of network data, statistical modeling has been a common approach network analysis. Multiple probabilistic frameworks have been proposed for network data (e.g., Barabási and Albert [1999], Bollobas et al. [2007], Crane [2018], Ghafouri and Khasteh [2020]). These frameworks are understood as approximations of real-world network formulation from different aspects with pros and cons in various situations. Our discussion in this paper is embedded in arguably one of the most popular frameworks for statistical network modeling – the so-called "inhomogeneous Erdős-Rényi model" [Bollobas et al., 2007, Goldenberg et al., 2010, Newman, 2010]. This framework includes as special cases several important models in the network literature, such as the stochastic block model (SBM) [Holland et al., 1983] and its variants [Airoldi et al., 2008, Karrer and Newman, 2011, Jin et al., 2017, Sengupta and Chen, 2018, Noroozi et al., 2019] for community structures, the random dot product model [Young and Scheinerman, 2007] and latent space model [Hoff et al., 2002, Hoff, 2008] for latent vector spaces, and the graphon model node-exchangeable graphs [Aldous, 1981, Lovász and Szegedy, 2006, Diaconis and Janson, 2008].

While effective modeling strategies have been proposed and studied in many settings, it is observed that real-world networks may exhibit a much wider range of patterns [Ugander et al., 2013, Ghasemian et al., 2020, Miao and Li, 2021] and it is generally difficult to know which model is the proper one, or whether there is a proper choice. Therefore, an indispensable task in network modeling is to select a proper model or algorithm to use. Depending on the scope of generality, model selection procedures have been proposed in different categories. The model-based methods are designed under a family of models, such as testing procedures [Bickel and Sarkar, 2016, Gao and Lafferty, 2017, Lei, 2016, Mukherjee and Sen, 2017, Banerjee and Ma, 2017, Jin et al., 2019, Zhang and Amini, 2020] and criterion-based methods [Saldana et al., 2017, Wang and Bickel, 2017, Le and Levina, 2015, Yan et al., 2017]. The more general model selection approach is cross-validation [Chen and Lei, 2018, Li et al., 2020b, Chang et al., 2020], which can be used to compare models across different families. Cross-validation intuitively seeks the most proper model for the data even if none of the models under consideration are true. Despite the many good properties derived from these methods, model selection can be unstable [Breiman, 1996a] in the sense that a small perturbation of the data can result in selecting a different model. Moreover, even if one does know the correct class of candidate models, it is not always possible to effectively fit such models due to various constraints. For example, when the network is sparse, it is known that one cannot accurately estimate an SBM with a large number of communities [Rohe et al., 2011, Lei and Rinaldo, 2014, Gao et al., 2017, Li et al., 2020a] or a general graphon structure [Chatterjee, 2015, Zhang et al., 2017, Gao et al., 2015].

To address the aforementioned shortcomings, this paper aims for an "off-the-shelf" and principled network modeling strategy. From a practical perspective, it is computationally efficient for large-scale networks and does not require careful tuning or model specification. From the theoretical perspective, our strategy can be general enough to incorporate the well-studied models in the literature with a theoretical guarantee on performance. These properties are achieved by a simple but powerful idea: instead of selecting a single model, we take a constrained linear combination of all candidate models in a data-driven way, such that the aggregated estimate is provably more stable and adaptive to different underlying models.

Model aggregation is not unique to network data. The idea has been studied in statistical learning on metric space data, such as density estimation and regression problems, often known as the "mixing" or "stacking" procedure [Stone, 1974, Wolpert, 1992, Breiman, 1996b, LeBlanc and Tibshirani, 1996, Catoni, 1997, Haussler et al., 1998, Yang, 2000, 2001, Juditsky and Nemirovski, 2000, Tsybakov, 2003, Bunea et al., 2007]. However, the mixing strategy in network settings has not been well studied, except for the recent empirical work of Ghasemian et al. [2020]. Designing a network version of mixing estimation as an "off-the-shelf" method is non-trivial. The previous mixing

strategies either fail to produce good theoretical guarantees in network settings [Tsybakov, 2003] or are computationally prohibitive [Yang, 2000, 2001, Tsybakov, 2003, Bunea et al., 2007, van der Laan et al., 2007] and require additional tuning [Bunea et al., 2007]. The unique nature of network data (e.g., no i.i.d sample, discreteness, sparsity) also require special methodological designs and a different approach to theoretical analysis.

Our mixing methods rely on splitting the network node-pairs into a training and a validation data set, as in Li et al. [2020b], although other methods may be considered [Spielman and Srivastava, 2011, Rohe and Qin, 2013, Le, 2021]. Multiple individual models are fit using the training data and then aggregated according to the validation data. A simple aggregation approach with exponential weights is introduced to achieve the basic model adaptivity, an ideal property one could hope for the cross-validation procedure. Then, based on special geometrical properties of network data, we design a sign-constrained least square procedure to find the aggregation, which significantly improves the strategy. Our method is much simpler to implement and possesses strong theoretical guarantees. Our numerical and theoretical analyses show that the proposed mixing methods nearly achieve the oracle estimators' accuracy regardless of whether the true model is contained in the candidate model set. Furthermore, an evaluation of 385 real-world networks shows that the proposed methods are more accurate and much more computationally efficient than the stacking method of Ghasemian et al. [2020], which still lacks theoretical support.

The rest of the paper is organized as follows. Section 2 introduces the proposed methods and their theoretical properties. We start from the basic setup of the mixing strategy. Exponential mixing is first introduced as a warm-up, which is shown to achieve single-model adaptivity. Then we introduce the more general non-negative linear mixing method and show that the non-negative property is crucial in network settings. Section 3 includes extensive simulation studies of the (almost) tuning-free properties and compares the mixing estimators with several benchmark methods under both graphon (nonparametric) models and parametric models. In Section 4, we consider link prediction tasks on 385 real-world networks from 6 domains, introduced in Ghasemian et al. [2020]. The evaluation demonstrates the advantage of the proposed method as an "off-the-shelf" method for link prediction. Section 5 concludes the paper.

2 The network mixing method and its properties

Notations. Given a matrix M, we use $\|M\|$ and $\|M\|_F$ to denote its spectral norm and Frobenius norm, respectively, and let $\|M\|_\infty = \max_{ij} |M_{ij}|$. In addition, for any two matrices P_1, P_2 of the same dimensions, we define $\langle P_1, P_2 \rangle = \text{trace}(P_1^T P_2)$. Given a matrix $M \in \mathbb{R}^{n\times n}$ and an index set $\Omega \subset [n] \times [n]$, denote by $M(\Omega)$ the matrix obtained from M by setting all entries of M with indices in the complement Ω^c of Ω to zero. For two sequences $\{a_n\}$ and $\{b_n\}$, we write $a_n \lesssim b_n$ if $a_n = O(b_n)$, in which case we can also write it as $b_n \gtrsim a_n$. If $a_n \lesssim b_n$ and $a_n \gtrsim b_n$, we write $a_n \approx b_n$. We say that an event E occurs with high probability if $\mathbb{P}(E) \geq 1 - n^{-\delta}$ for some constant $\delta > 0$. We use $C > 0$ to denote an absolute constant whose value may change from line to line.

We focus on unweighted undirected networks in this paper. Let n be the network size. A network can be represented by a binary adjacency matrix $A \in \{0, 1\}^{n\times n}$, where $A_{ij} = 1$ if and only if node i and node j are connected. Since the network is undirected, we have $A = A^T$. Furthermore, assume $A_{ii} = 0$ for all i (we do not consider self-loops). Our discussion of networks will be embedded in the *inhomogeneous Erdős-Rényi network model* [Bollobas et al., 2007], which requires that the upper diagonal entries of A are independent Bernoulli random variables. Denote $P = \mathbb{E}A$ and assume that there exists a set $\mathcal{M}$ of m available methods for estimating P from A. These may be parametric

estimation procedures, such as those based on the stochastic block model [Chen and Lei, 2018, Li et al., 2020b] or the latent space model [Ma et al., 2020], or nonparametric algorithms [Chatterjee, 2015, Airoldi et al., 2013, Chan and Airoldi, 2014, Gao et al., 2015, Zhang et al., 2017]. Our goal is to leverage these methods to derive a new estimator $\hat{P}$ that can achieve a better approximation of P in terms of the mean squared error. A natural baseline target is, not surprisingly, to ensure that $\hat{P}$ is as good as the unknown optimal estimator in $\mathcal{M}$. Such a property is often referred to as *adaptivity* [Catoni, 1997, Yang, 2000, 2001]. Ultimately, our goal will be more ambitious than adaptivity: we would like to achieve optimality within a broader class of estimators than the m estimators of $\mathcal{M}$. We aim for an off-the-shelf option for modeling networks. That is, in addition to strong theoretical properties, empirical applicability is equally important. In a linear regression setting, procedures with provable adaptivity may not be computationally feasible or may require tuning parameters [Catoni, 1997, Yang, 2000, 2001, Tsybakov, 2003, Bunea et al., 2007, van der Laan et al., 2007], but we will introduce a computationally efficient and tuning-free method that can be scalable even to large networks.

Motivated by the adaptive mixing method in the classical regression setting, we resort to a data-splitting strategy to design our method. The strategy is based on the dyad-splitting of the network cross-validation procedure of Li et al. [2020b]. Specifically, given a fixed $p \in (0, 1)$, we randomly choose a subset of node pairs $\Omega \subset [n] \times [n]$ by independently selecting each node pair with probability p; in our setting, p is usually a constant between 0 and 0.5. Then $A(\Omega^c)$ can be viewed as an adjacency matrix generated from the inhomogeneous Erdős-Renyi model with probability matrix $(1-p)P$. Therefore, any model fitting methods for inhomogeneous Erdős-Renyi networks can be used with the same theoretical guarantees for consistency as their original version as discussed in Gao and Ma [2020] and Li et al. [2020c]. We apply the methods from $\mathcal{M}$ to $A(\Omega^c)$ to get estimators of $(1-p)P$, and multiply them by $(1-p)^{-1}$ to obtain estimators $\hat{P}^{(1)}, ..., \hat{P}^{(m)}$ of P. We assume that the entries of these estimators always fall between 0 and 1. If this is not the case, we can always threshold the entries without increasing the error. The mixing strategy is essentially an aggregated version of the m estimators for P.

We use the hold-out entries $A(\Omega)$ to determine a reasonable way to combine the available estimators. A potential approach for this purpose is the cross-validation method of Li et al. [2020b], which calculates $\|A(\Omega) - \hat{P}^{(r)}(\Omega)\|_F^2, 1 \le r \le m$, and picks the one with the smallest error:

$$\hat{P} = \hat{P}^{(\hat{r})}, \quad \hat{r} = \underset{1 \le r \le m}{\operatorname{argmin}} \|A(\Omega) - \hat{P}^{(r)}(\Omega)\|_F^2. \tag{1}$$

(In practice, one often repeats this procedure multiple times and uses the average validation error as the criterion. Here we focus on the single split for conceptual simplicity.) This is one of the most common model selection strategies in statistical modeling [Hastie et al., 2009]. It has been verified, for example, in the context of multivariate outcome prediction and density estimation [van der Laan et al., 2006, Feng and Yu, 2019, Lei, 2020]. But so far it is unclear whether such an approach is valid for network cross-validation. For example, although the estimator in (1) works well for moderately dense networks when $\mathcal{M}$ contains the true model, its performance deteriorates quickly in a sparse regime or under model misspecification (as we show in Section 3). We now introduce a soft-selection version of the cross-validation method to address this shortcoming.

2.1 Network mixing with exponential weights and estimation adaptivity

We view the validation error $\|A(\Omega) - \hat{P}^{(r)}(\Omega)\|_F^2$ as a goodness-of-fit metric for the rth model. To match the performance of the optimal model in $\mathcal{M}$, we focus on the models with small valida-

tion errors. In particular, we propose the following simple rule to combine the models based on exponential weights:

$$\hat{P}^{(\exp)} = \sum_{r=1}^{m} \pi_r \hat{P}^{(r)}, \qquad \pi_r = \frac{\exp\left(-\|A(\Omega) - \hat{P}^{(r)}(\Omega)\|_F^2\right)}{\sum_{r=1}^{m} \exp\left(-\|A(\Omega) - \hat{P}^{(r)}(\Omega)\|_F^2\right)}. \tag{2}$$

Compared with (1), this estimator is a soft-selection version of the cross-validation procedure. Despite its simplicity, we now show that it achieves model adaptivity, matching the performance of the unknown optimal estimator produced by $\mathcal{M}$. Since Ω is independent of the data, all theoretical analysis is conditioned on Ω. The data-splitting proportion p is a fixed constant, independent of the network size n.

Theorem 1 (Mixing estimator with exponential weights)**.** Let A be the adjacency matrix of a random network drawn from the inhomogeneous Erdős-Rényi model with probability matrix $P = \mathbb{E}A$. Denote by $\hat{P}^{(r)}$, $1 \le r \le m$, the estimators of P obtained by applying the methods from $\mathcal{M}$ to $A(\Omega)$, and let $\hat{P}$ be the convex combination of these estimates defined by (2). Assume $m = o(n^2)$. Then with high probability,

$$\left\|\hat{P}^{(\exp)}(\Omega) - P(\Omega)\right\|_F \le \min_{1\le r\le m} \left\|\hat{P}^{(r)}(\Omega) - P(\Omega)\right\|_F + C\varepsilon, \tag{3}$$

where $C > 0$ is a constant and

$$\varepsilon = \left(\log n \cdot \max_{i,j} P_{ij}\right)^{1/2} + \min\left\{\frac{\log(nm)}{\min_{1\le r\le m} \|P^{(r)}(\Omega) - P(\Omega)\|_F},\ \log^{1/2}(nm)\right\}.$$

Remark 1. Although Theorem 1 describes the error restricted to Ω, extending the error bound to the entire matrix P is trivial. Replacing Ω by $\Omega^c = [n]^2 \setminus \Omega$ in the estimating procedure and applying Theorem 1 would generate another estimator $\breve{P}$ that admits the same type of error bound. Then combining $\hat{P}(\Omega)$ and $\breve{P}(\Omega^c)$ as a full estimator would produce the same error bound for the entire matrix P. For simplicity, we will state our results only in terms of $P(\Omega)$ in all theoretical discussions to follow.

Theorem 1 states that the estimator given in (2) is nearly as good as the best estimate produced by a single method from $\mathcal{M}$. To better understand the additional error ε, assume that the network is generated from a stochastic block model, under which nodes are partitioned into k groups according to a label vector $c \in [n]^k$ and edges are formed independently between nodes with probabilities $P_{ij} = B_{c_i c_j}$ for some fixed matrix $B \in \mathbb{R}^{k\times k}$. According to Gao et al. [2015],

$$\min_{1\le r\le m} \left\|\hat{P}^{(r)}(\Omega) - P(\Omega)\right\|_F^2 \gtrsim k^2 + n\log k,$$

while the square of the additional error is

$$\varepsilon^2 \lesssim \log n + \min\left\{\frac{\log^2(nm)}{k^2 + n\log k}, \log(nm)\right\} \approx \log n + \frac{\log^2(nm)}{k^2 + n\log k}.$$

It is easy to see that ε^2 is smaller than the rate-optimal error $k^2 + n\log k$. Therefore, our estimator can always match the optimal estimator from $\mathcal{M}$.

As a connection to similar properties of statistical estimation problems, Catoni [1997] and Yang [2000, 2001] introduce mixing methods for regression and density estimation that can achieve estimation adaptivity. Unfortunately, their strategies are computationally expensive even in regression

settings, let alone in network problems, the sample size of which scales in $O(n^2)$. In contrast, our estimator (2) achieves the same type of adaptivity with negligible computational cost in addition to the m model-fitting procedures.

2.2 Beyond adaptivity: Non-negative linear network mixing

The exponential mixing estimator (2) performs well when at least one of the m individual estimators is accurate. In practice, however, all the available estimators may be inaccurate, for example, when the network is very sparse. Given the m estimators at hand, it is natural to be more ambitious: to seek a better candidate than any single model estimate.

Consider the class of linear combinations of the m estimators

$$\hat{P} = \sum_{r=1}^{m} \pi_r \hat{P}^{(r)}. \tag{4}$$

The most natural option is the *linear mixing estimator*, which uses the weights provided by solving the ordinary least squares (OLS) problem:

$$\hat{\pi}^{(\text{ols})} = \underset{\pi \in \mathbb{R}^m}{\operatorname{argmin}} \left\| \sum_{r=1}^{m} \pi_r \hat{P}^{(r)}(\Omega) - A(\Omega) \right\|_F^2. \tag{5}$$

Although linear mixing performs very well in many settings, it is a generic method and does not leverage many features of network estimation. For example, all the entries of P are non-negative, and therefore it is expected that $\langle \hat{P}^{(r)}(\Omega), P(\Omega) \rangle > 0$ for any reasonable estimate $\hat{P}^{(r)}(\Omega)$. Similarly, $\langle \hat{P}^{(r)}(\Omega), \hat{P}^{(s)}(\Omega) \rangle > 0$ for all $1 \le r, s \le m$ (unless their supports are disjoint), and the angles between reasonably good estimators tend to be small. Moreover, in sparse networks the noises (entries of $A - P$) are heavy-tailed random variables, so the available estimators of P can be very noisy. These observations motivate us to further improve the linear mixing method by introducing a natural non-negative constraint to the OLS weights. In particular, we consider the *non-negative linear* (NNL) mixing estimator based on the following weights:

$$\hat{\pi}^{(\text{nnl})} = \underset{\pi \succeq 0}{\operatorname{argmin}} \left\| \sum_{r=1}^{m} \pi_r \hat{P}^{(r)}(\Omega) - A(\Omega) \right\|_F^2, \tag{6}$$

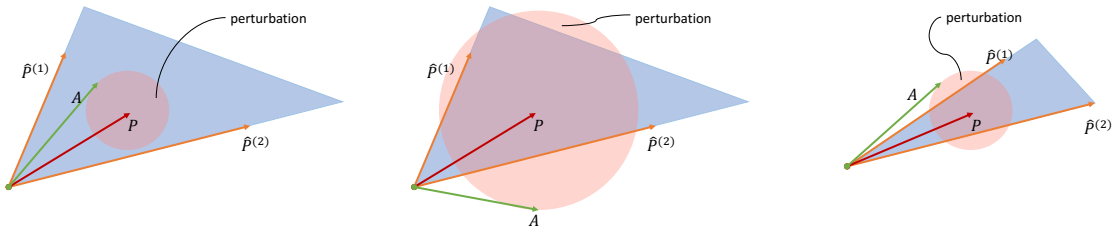
where $\pi \succeq 0$ denotes the constraint that all the entries of the weight vector π are non-negative. This is a simple convex optimization problem and can be solved efficiently by either a projected quasi-Newton algorithm or a sequential coordinate descent algorithm [Kim et al., 2006, Chen and Plemmons, 2010]. The NNL mixing estimator is defined to be

$$\hat{P}^{(\text{nnl})} = \sum_{r=1}^{m} \hat{\pi}_r^{(\text{nnl})} \hat{P}^{(r)}. \tag{7}$$

We can see that $\hat{P}^{(\text{nnl})}(\Omega)$ is the projection of $A(\Omega)$ on the convex cone formed by the conical combination of m individual estimators $\hat{P}^{(r)}(\Omega), 1 \le r \le m$. The non-negative sign constraint directly imposes a regularization effect that significantly helps the method handle the potentially large number of individual estimators. This is crucial for a tuning-free procedure and matches our aim for an "off-the-shelf" method.

The non-negative constraint has proved effective in high-dimensional linear regression problems [Slawski and Hein, 2011, Slawski et al., 2013, Meinshausen et al., 2013], with properties similar to the LASSO estimator. However, in the current context, NNL estimation is not motivated by the curse of dimensionality. In network mixing problems, the sample size for (5) is of order n^2, and m is usually much smaller than n^2. Therefore, though m can be large if one wants to ensure the expressiveness of the mixing estimator, (5) is usually not an ultra-high-dimensional problem. Instead, the strong correlation between the $\hat{P}^{(r)}$s and the low concentration of the adjacency matrix (due to the network sparsity) complicate the matter and result in the deterioration of the OLS estimator, even in a relatively low-dimensional setting.

To see why the non-negative constraint can help, let us assume that the true signal $P(\Omega)$ is within the convex cone of the estimators. Figure 1a illustrates an ideal situation. When the two individual estimators form a large angle, and the perturbation range of A around P is not too large, the observed adjacency matrix is likely to stay within or close to the convex cone, so the OLS and NNL estimators perform similarly. When the perturbation range is large, as in Figure 1b (which happens in sparse networks [Le et al., 2017]), A may be far from the cone (and P), and the projection to the cone can result in a better estimator. On the other hand, when the two estimators align well, the convex cone is smaller (Figure 1c), and A is unlikely to belong to the cone. In this case, the NNL estimator can also outperform the OLS estimator. Overall, these observations show that sparse networks and strong alignment between estimators tend to make the OLS estimator vulnerable to increasing m, even when m is much smaller than n^2.



(a) Good concentration and weak alignment

(b) Bad concentration and weak alignment

(c) Good concentration and strong alignment

Figure 1: Illustration of the relation between the convex cone and the concentration of the adjacency matrix.

We now present the theoretical support for the NNL estimator and its claimed advantage over linear mixing. Let

$$\mathcal{L} = \Big\{ \sum_{r=1}^{m} \pi_r \hat{P}^{(r)}(\Omega), \pi_r \in \mathbb{R}, 1 \le r \le m \Big\}, \quad \mathcal{C} = \Big\{ \sum_{r=1}^{m} \pi_r \hat{P}^{(r)}(\Omega), \pi_r \ge 0, 1 \le r \le m \Big\}$$

be the linear subspace and the cone generated by $\{\hat{P}^{(r)}(\Omega), 1 \le r \le m\}$. Denote by $\Pi_{\mathcal{L}}$ and $\Pi_{\mathcal{C}}$ the projections onto $\mathcal{L}$ and $\mathcal{C}$, respectively.

Theorem 2 (Non-negative mixing estimator, positive inner products)**.** Let A be the adjacency matrix of a random network drawn from the inhomogeneous Erdős-Rényi model with probability matrix $P = \mathbb{E}A$. Denote by $\hat{P}^{(r)}$, $1 \le r \le m$, the estimates of P obtained by applying the methods from $\mathcal{M}$ to $A(\Omega^c)$, and let $\hat{P}^{(\mathrm{nnl})}$ be the NNL estimator defined by (7). Assume that

$$\delta = \min_{1 \le r,s \le m} \frac{\langle \hat{P}^{(r)}(\Omega), \hat{P}^{(s)}(\Omega) \rangle}{\|\hat{P}^{(r)}(\Omega)\|_F \cdot \|\hat{P}^{(s)}(\Omega)\|_F} > 0. \tag{8}$$

Then there exists a constant $C > 0$ such that with high probability,

$$\|\hat{P}^{(\text{nnl})}(\Omega) - P(\Omega)\|_F \quad \leq \quad \|\Pi_{\mathcal{C}} P(\Omega) - P(\Omega)\|_F + C\delta^{-1/2}\varepsilon, \tag{9}$$

where $C > 0$ is a constant and

$$\varepsilon = \max_{1\leq r\leq m} \frac{\|\hat{P}^{(r)}(\Omega)\|_\infty}{\|\hat{P}^{(r)}(\Omega)\|_F} \cdot \log(n+m) + \sqrt{\|P(\Omega)\|_\infty \log(n+m)}. \tag{10}$$

Theorem 2 states that the proposed estimator $\hat{P}^{(\text{nnl})}(\Omega)$ is nearly as accurate as the non-negative linear oracle $\Pi_{\mathcal{C}} P(\Omega)$. Given a fixed δ, the extra error term ε can be at most of the order $\log(n+m)$, although it will grow much more slowly if the entries of P and $\hat{P}^{(r)}$ are of similar orders and the network is relatively sparse.

As a consequence of Theorem 2, the next corollary shows that the NNL estimator can be strictly more accurate than the OLS estimator $\Pi_{\mathcal{L}} A(\Omega)$, especially when m is large (a typical scenario in model aggregation).

Corollary 1 (Comparison of NNL and OLS estimators)**.** Let A be the adjacency matrix of a random network drawn from the inhomogeneous Erdős-Rényi model with probability matrix $P = \mathbb{E}A$. Denote by $\hat{P}^{(r)}$, $1 \leq r \leq m$, the estimators of P obtained by applying the methods from $\mathcal{M}$ to $A(\Omega^c)$, and let $\hat{P}^{(\text{ols})}$ be the OLS estimator defined by (5). With high probability,

$$\|\hat{P}^{(\text{ols})}(\Omega) - P(\Omega)\|_F^2 \leq \|\Pi_{\mathcal{L}} P(\Omega) - P(\Omega)\|_F^2 + m\|P(\Omega)\|_\infty + \left(m \log n \|P(\Omega)\|_\infty\right)^{1/2}. \tag{11}$$

Furthermore, consider the setting in Theorem 2 and denote

$$\rho = \left(1 - \|P(\Omega)\|_\infty\right)^2 \cdot \min_{(i,j)\in\Omega} P_{ij}^2, \quad \Delta = \|\hat{P}^{(\text{ols})}(\Omega) - P(\Omega)\|_F^2 - \|\hat{P}^{(\text{nnl})}(\Omega) - P(\Omega)\|_F^2.$$

Assume $m \geq C\|P(\Omega)\|_\infty \rho^{-1} \log n$ for some sufficiently large constant $C > 0$. Then with high probability,

$$\Delta \geq \frac{m\rho^{1/2}}{2} - \Psi, \tag{12}$$

where

$$\Psi = \frac{C' \log^2(n+m)}{\delta} \Big(\max_{1\leq r\leq m} \frac{\|\hat{P}^{(r)}(\Omega)\|_\infty^2}{\|\hat{P}^{(r)}(\Omega)\|_F^2} + \|P(\Omega)\|_\infty \Big) + \|\Pi_{\mathcal{C}} \Pi_{\mathcal{L}} P(\Omega) - \Pi_{\mathcal{L}} P(\Omega)\|_F^2$$

for some constant $C' > 0$.

Note that the term $\|\Pi_{\mathcal{C}} \Pi_{\mathcal{L}} P(\Omega) - \Pi_{\mathcal{L}} P(\Omega)\|_F$ is the distance from the linear oracle $\Pi_{\mathcal{L}} P(\Omega)$ to the cone $\mathcal{C}$, measuring the degree of violation of the non-negative cone assumption. Corollary 1 demonstrates the effects of the main factors on performance. It can be seen that the advantage of NNL mixing over OLS mixing increases with m and also with the correlation δ, as we have discussed above. Next, we illustrate the effect of network density.

Example 1. Assume a sparse network and that all the entries of P are of the same order $d/n = o(1)$, where d is the average degree. Also, assume that the m individual estimators are reasonable, so all the $\hat{P}_{ij}^{(r)}$s are also of the same order as the P_{ij}s. When the conical assumption holds and

$m \gg n \log n/d$, taking the relative error to cancel the scaling effect of P, we have

$$\frac{\Delta}{\|P(\Omega)\|_F^2} \gtrsim \frac{m}{nd} - \frac{\log^2(n+m)}{\delta d^2}.$$

For sparser networks with small d, the advantage of the NNL estimator is more dramatic. This theoretical prediction is further supported by our empirical evidence in Section 3.

Theorem 2 requires $\hat{P}^{(r)}(\Omega)$, $1 \le r \le m$, to have positive inner products. This assumption is reasonable because they are all estimators of $P(\Omega)$ and must have non-negative entries. More importantly, we can directly calculate δ from data.

Next, we consider an even weaker assumption, which is true in all reasonable settings we can think of. We replace the assumption (8) in Theorem 2 with the following assumption on $\hat{P}^{(r)}(\Omega)$, $1 \le r \le m$, known as the *self-regularizing property* [Slawski and Hein, 2011]. Denote by $\Sigma \in \mathbb{R}^{m\times m}$ the matrix with entries $\Sigma_{rs} = \langle \hat{P}^{(r)}(\Omega), \hat{P}^{(s)}(\Omega)\rangle$ and $\|\Sigma\|_\infty = \max_{1\le r,s\le m} |\Sigma_{rs}|$. We say that Σ satisfies the self-regularizing property with constant κ, $0 < \kappa \le 1$, if

$$\beta^T \Sigma \beta \ge \kappa \|\Sigma\|_\infty, \text{ for all } \beta \succeq 0 \text{ and } \textstyle\sum_{r=1}^m \beta_r = 1. \tag{13}$$

Notice that, like (8), this condition can be numerically verified from the data by solving a convex problem. More importantly, in our current setting of network mixing, this condition will almost always hold due to the following property, taken directly from the discussion of Slawski and Hein [2011].

Proposition 1. If there exists a partition $\{Q_t\}_{t=1}^T$ of $[m]$ such that

$$\min_{r,s\in Q_t} \langle \hat{P}^{(r)}(\Omega), \hat{P}^{(s)}(\Omega)\rangle \ge \kappa \max_{r\in Q_t} \|\hat{P}^{(r)}(\Omega)\|_F^2 > 0, \text{ for all } 1 \le t \le T,$$

then Σ is self-regularizing with constant κ/T.

To see why (13) is weaker than (8), notice that without loss of generality we can assume that all the $\hat{P}^r(\Omega)$s have the same norm, because rescaling does not change our linear fitting. So when (8) holds, Σ also satisfies (13), with $\kappa = \rho$. Moreover, even if we are in the extreme situation where

$$\langle \hat{P}^{(r)}(\Omega), \hat{P}^{(s)}(\Omega)\rangle = 0$$

for some r, s, Proposition 1 indicates that (13) can still hold as long as we separate such pairs in different groups. Even in the worst case, the m-way partition guarantees the self-regularizing property, with $\kappa = 1/m$.

Theorem 3. Consider the setting of Theorem 2 with condition (8) replaced by the self-regularizing property (13). Then there exists a constant $C > 0$ such that with high probability,

$$\|\hat{P}^{(\text{nnl})}(\Omega) - P(\Omega)\|_F \quad \le \quad \inf_{\eta\in\mathcal{C}} \|\eta - \Pi_{\mathcal{L}} P(\Omega)\|_F + C\kappa^{-1/2}\epsilon',$$

where

$$\begin{aligned} \epsilon' &= \left(\frac{\Phi \log(n+m)}{\sqrt{\|\Sigma\|_\infty}} + \sqrt{\Phi \|\pi^*\|_1 \log(n+m)}\right), \\ \Phi &= \max_{1\le r\le m} \|\hat{P}^{(r)}(\Omega)\|_\infty + \|P(\Omega)\|_\infty^{1/2} \|\Sigma\|_\infty, \end{aligned}$$

and π^* is the non-negative linear coefficient of the oracle estimator such that

$$\Pi_{\mathcal{C}} P(\Omega) = \sum_r \pi_r^* \hat{P}^{(r)}(\Omega).$$

Compared with Theorem 2, the price we pay for the weaker assumption in Theorem 3 is greater additive error. As a simple demonstration, consider the setting of Example 1 and assume that all the $\hat{P}^r(\Omega)$s have the same norm. In this case, $\kappa = \delta$. At best, the ϵ' of Theorem 3 has

$$\epsilon' \approx \frac{d^{3/2}}{n^{1/2}} \log(n+m) + \frac{d^{5/4}}{n^{1/4}} \log^{1/2}(n+m).$$

In contrast, the ϵ of Theorem 2 gives

$$\epsilon \approx \frac{1}{n} \log(n+m) + \frac{d^{1/2}}{n^{1/2}} \log^{1/2}(n+m),$$

which is clearly of a lower order.

2.3 Candidate set and other practical considerations

The mixing strategy involves determining a set of candidate estimators and the hold-out proportion, p. For the OLS and NNL mixing methods, the scales of the individual estimators do not matter, and they do not need to be accurate individually. This property is essential for a mixing method to be applicable for general link prediction problems (see Section 4). But exponential mixing does not make sense if the $\hat{P}^{(r)}$s are in the wrong scale or if they are all inaccurate.

In determining the candidate set $\mathcal{M}$, there is a necessary tradeoff between computational efficiency and expressiveness. We recommend using spectral clustering together with SBM fitting for $1 \le k \le K_{\max}$, spherical spectral clustering [Rohe et al., 2011, Lei and Rinaldo, 2014] and DCBM fitting for $1 \le k \le K_{\max}$, and the universal singular value thresholding (USVT) of Chatterjee [2015], with the improvement mentioned in Zhang et al. [2017], by taking the first $n^{1/3}$ singular components. This set of candidate estimators can be calculated efficiently for large networks, with the main computational burden being the one-time calculation of SVD. The block models, despite their simplicity, have been shown to have great approximation power if they are used properly [Airoldi et al., 2013], and USVT also comes with good expressiveness. When networks are sparse, the block components likely receive higher mixing weights, helping stabilize the estimator. In contrast, when networks are sufficiently dense, USVT can be more effective. Empirically, we observe that this setup gives very effective estimation performance. In the above recommendation, $K_{\max}$ is a reasonably large positive integer, and in Sections 3 and 4, we show that our method is very robust to $K_{\max}$ and ρ within reasonable ranges, which is again due to the regularizing property of the non-negative constraint.

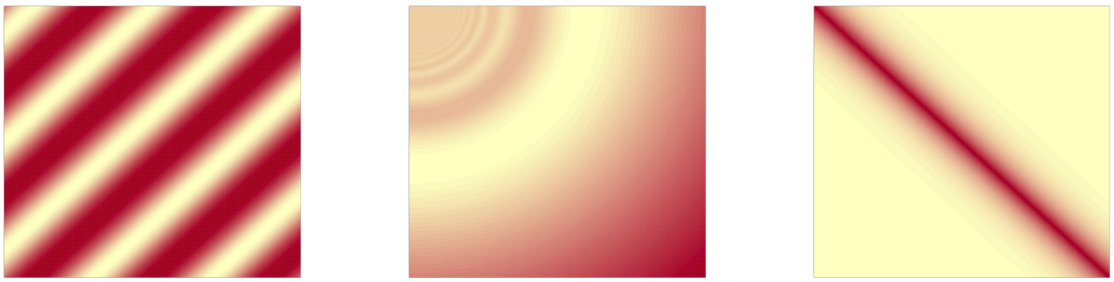
The discussion so far has focused on one random split, Ω. In practice, one can also randomly repeat the procedure multiple times and take the average as the output. This may slightly improve accuracy in our limited evaluation but will also increase the computational cost by a multiplicative factor. For this reason, we only consider single splits in this study.

The mixing method, and any aggregation methods, do have limitations. The primary performance metric of the mixing approach is estimation accuracy. Combining multiple estimators may destroy the structural interpretations of each individual estimator, such as block structures or smoothness.

However, when special structures are desirable, one can always apply the same structural extraction strategy, such as community detection, to the estimated $\hat{P}$. A more accurate estimate of P may lead to more accurate structure extraction.

3 Simulation examples

We now evaluate the mixing methods using simulation examples. We focus on the task of estimating the network connection probability matrix, P. In all the examples below, we set the network size to $n = 1000$, with varying density. We first evaluate our method in the graphon model settings. In particular, we use the three connection graphon connection matrices W from Zhang et al. [2017], for which we set $P = \alpha \cdot W$ and use α to control the expected average degree of the networks (Figure 2).



(a) Model 1 (b) Model 2 (c) Model 3

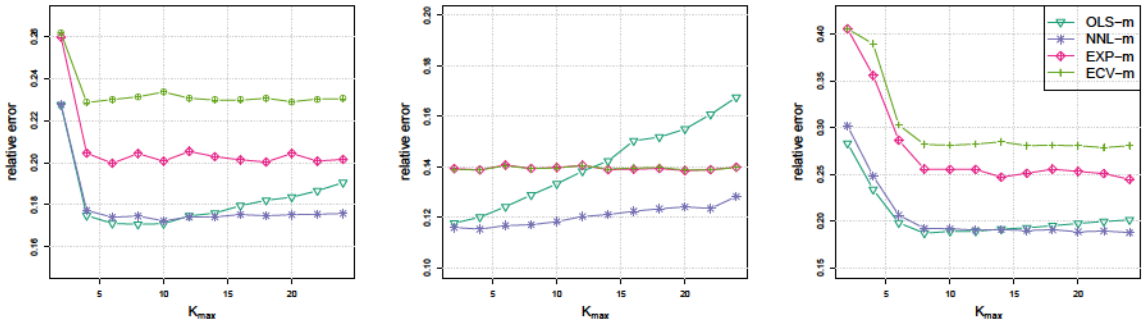
Figure 2: Three network models based on the graphon setup from Zhang et al. [2017]. The first model is of rank 3. The other two are full-rank models.

The mixing methods are based on the recommended setting discussed in Section 2.3, using SBM and DCBM fitting for $1 \le k \le K_{\max}$, and USVT. We first show that the proposed method is robust to $K_{\max}$ within a reasonable range. After that, we compare the mixing estimator with a few benchmarks. We use $K_{\max} = 15$ and $p = 0.1$ as the default setting in all the benchmark comparisons. Given an estimator $\hat{P}$, performance is measured by the relative Frobenius error, defined as

$$\|\hat{P} - P\|_F^2 / \|P\|_F^2.$$

3.1 Comparison of mixing aggregation strategies

We first want to compare different aggregation strategies for the mixing method, including linear mixing (OLS-m), NNL mixing (NNL-m), exponential mixing (EXP-m), and edge cross-validation model selection (ECV-m). Since $K_{\max}$ needs to be specified, we investigate its impact on the mixing method's performance. To demonstrate that the method is almost tuning-free, we want to show that it remains stable for a reasonable range of $K_{\max}$ values. Figure 3 shows the estimation performance with varying $K_{\max}$ for networks with expected average node degree 20.

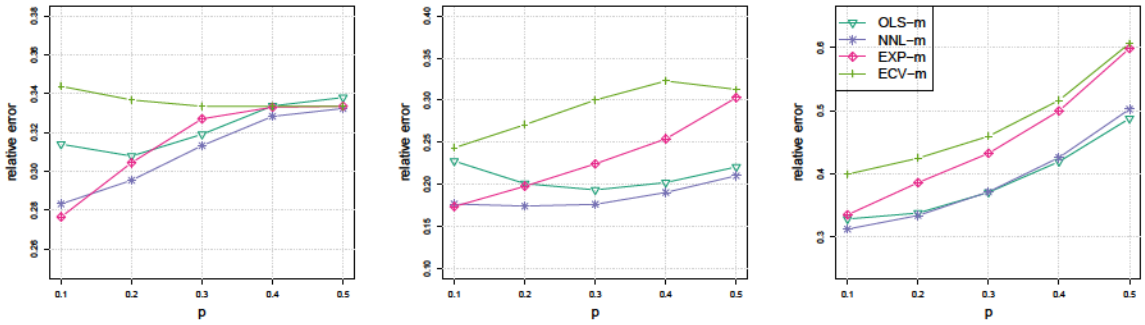


(a) Model 1 (b) Model 2 (c) Model 3

Figure 3: Relative estimation error of P with various values of $K_{\max}$ and $p = 0.1$. The networks under evaluation have $n = 1000$ nodes and average node degree 20.

As can be seen, small values of $K_{\max}$ result in underfitting and a large error. As $K_{\max}$ increases, the error drops until overfitting kicks in. However, overall, exponential mixing, NNL mixing, and cross-validation do not suffer from overparameterization of block approximations. Linear combination, in contrast, degrades when $K_{\max}$ is large. This observation matches our theoretical understanding of the linear estimator as we are fitting the model with an increasing number of variables. Exponential mixing significantly outperforms cross-validation, while NNL mixing is much better than both.

Next, we evaluate the robustness of the different strategies with respect to the hold-out proportion, p. We evaluate their performance when p varies from 0.1 to 0.5 and $K_{\max}$ is fixed at 15 (Figure 4). Overall, all the aggregation methods tend to improve as p decreases within this range. This indicates that the aggregation weight determination step is easier and requires a smaller sample size than the individual model estimation step. Of all the aggregation methods, non-negative linear combination is the most robust. Linear combination also delivers competitive estimation accuracy. Based on this observation, we recommend using $K_{\max} = 15, p = 0.1$ as the default configuration for network mixing estimation.

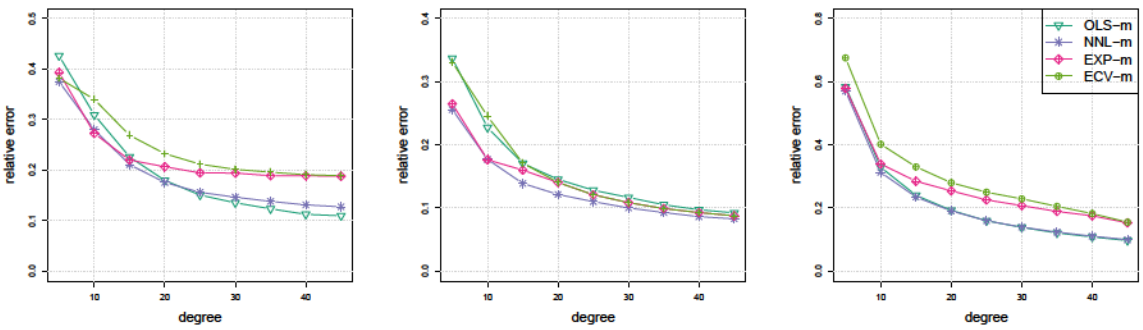


(a) Model 1 (b) Model 2 (c) Model 3

Figure 4: Relative estimation error of P with varying p and $K_{\max} = 15$. The networks under evaluation have $m = 1000$ and average node degree 20.

Now we fix $K_{\max} = 15$ and $p = 0.1$ and evaluate the estimation performance for a range of network sparsity levels, varying the expected average degree from 5 to 45 (Figure 5). In the sparse regime, NNL mixing outperforms the others, while linear mixing catches up as the network becomes denser.

This is predicted in Corollary 1 as well. Exponential mixing eventually coincides with the cross-validation method, which also matches our theory.



(a) Model 1 (b) Model 2 (c) Model 3

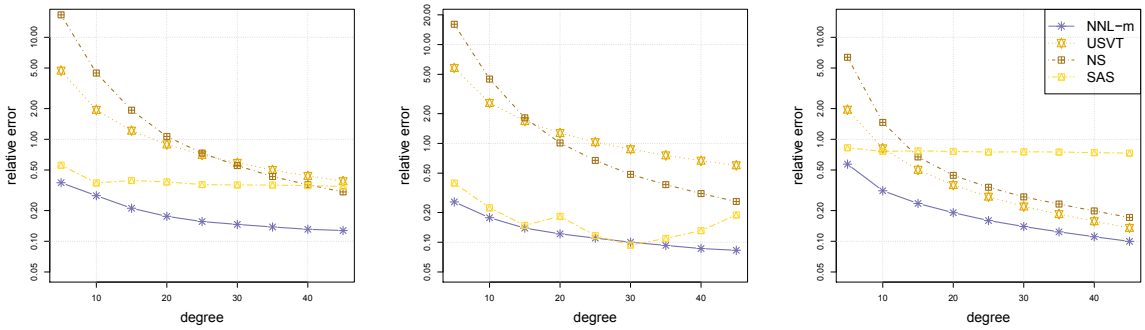
Figure 5: Relative estimation error of P with various expected average degrees. The networks under evaluation contain $n = 1000$ nodes, and $K_{\max} = 15, p = 0.1$ are used.

Overall, NNL mixing is preferable compared to the other methods. Linear mixing is less robust to the choice of $K_{\max}$ and is inferior to non-negative combination in the sparse setting. However, when the network is denser, it outperforms the others. In all of the experiments to follow, we use $K_{\max} = 15$ and $p = 0.1$, and we believe this configuration can be used in almost all applicable tasks.

3.2 Comparison with benchmark network estimation methods

Now we compare the mixing method with a few benchmark network estimation methods. We split this into two parts. First, we consider the three graphon models in Figure 2, and we compare the mixing method to the graphon methods, which have theoretical guarantees and reasonable computational cost. These include the USVT estimator of Chatterjee [2015], the neighborhood smoothing (NS) method of Zhang et al. [2017], and the sort-and-smooth (SAS) method of block approximation from Airoldi et al. [2013] and Chan and Airoldi [2014]. Second, we generate networks from two special parametric models: the SBM and the latent space model [Hoff et al., 2002]. Under these models, oracle estimations can be achieved by parametric model fitting, and they are included for comparison. The mixing, USVT, NS, and latent space model fitting [Ma et al., 2020] are all based on the R package *randnet* [Li et al., 2021], while SAS is based on the R package *graphon* [You, 2020].

Figure 6 shows the model estimation performance of the proposed mixing strategies and the three graphon estimation benchmarks. All four variant mixing strategies uniformly outperform the benchmark methods with all three models. The advantage of the mixing method is clear when the networks are sparse. Of the three graphon estimation methods, SAS is more accurate for the first two models, while USVT and NS are better for the third one. The difference between various mixing strategies is negligible.



(a) Model 1 (b) Model 2 (c) Model 3

Figure 6: Estimation performance of the proposed mixing methods and three benchmark graphon estimation methods on synthetic networks generated from three graphon models.

Next, we generate networks from two parametric models. The first one is the SBM, with six communities of equal size, following the configuration of Zhang et al. [2017]. Under the SBM, we include two oracle estimators. Oracle1 requires the true community labels and estimates the probability matrix P by averaging entries within corresponding blocks of the adjacency matrix A. Oracle2 knows the true SBM with six clusters but not the true community labels; it uses spectral clustering to find the node labels and averages entries within estimated blocks of A. In particular, notice that Oracle2 itself is automatically included as one of the individual models in the mixing.

The second model is the latent space model (LSM), with

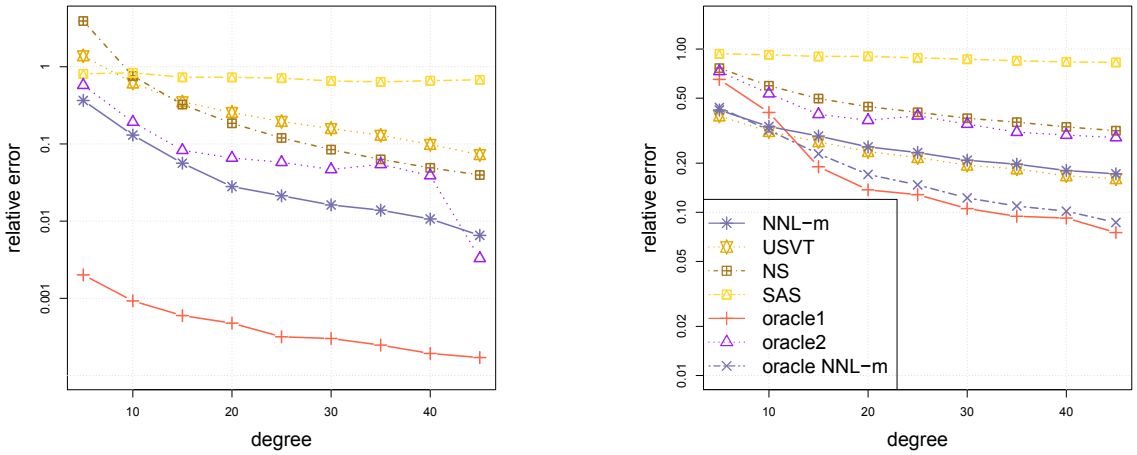
$$\text{logit}(P_{ij}) = \alpha_i + \alpha_j + \langle Z_i, Z_j \rangle,$$

where Z_i, Z_j are latent vectors in $\mathbb{R}^4$, generated by $N(0, I_4)$ and then centralized according to Ma et al. [2020]. Oracle1 in this case is the oracle version of the model that uses the true model structure and also the true latent dimension, estimated by the gradient descent method of Ma et al. [2020] with the recommended initialization. Oracle2 still assumes the correct model structure, but uses the wrong dimension (3 instead of 4), representing the possibility of dimensionality mis-specification. For reference, we also include an oracle version of the NNL mixing method, which has Oracle1 as an individual component.

Figure 7 shows the performance of all the methods under the two parametric models. In the SBM setting, the mixing methods are inferior only to the unbeatable Oracle1 and are even better than Oracle2. Since the mixing procedure includes Oracle2 as an individual component, this result demonstrates the effects of ensembling multiple models. In the difficult regime, including multiple models may help stabilize the estimation and further improve the estimate of the true model when it is fitted separately. In the LSM setting, the mixing method is again inferior only to the perfect oracle parametric estimation (Oracle1) in the dense setting and, again, it is even better than it in the sparse setting. USVT is also very effective in this setting, as indicated by the theory of Chatterjee [2015], and the mixing method adapts to it. NNL-m and USVT are even better than Oracle2 estimate. The oracle NNL-m adapts to Oracle1 in the dense setting and outperforms it in the sparse setting. The comparison between the mixing method and the oracle methods under these two parametric models highlights two advantages of the mixing approach:

- In the sparse regime, the estimation accuracy may be poor even if the true model is known. The mixing strategy provides a mechanism for combining simpler models to obtain a more stable estimate.

- In the dense regime, the mixing approach may remain effective even if the mixing ensemble does not include the true model and can match the oracle otherwise.



(a) Stochastic block model

(b) Latent space model

Figure 7: Comparison with benchmark graphon estimation methods on synthetic networks generated from two parametric models.

4 Link prediction in real-world networks

Link prediction is a widely used procedure in network science and many scientific domains to predict missing links/non-links based on a partially observed network [Liben-Nowell and Kleinberg, 2007, Lichtenwalter et al., 2010, Lü and Zhou, 2011]. A link prediction algorithm usually provides a set of scores corresponding to the missing links/non-links. A higher score means that it is more likely that a link exists between the corresponding nodes. When the data are generated from a random network model, the estimated edge probabilities, if available, provide natural prediction scores. One can predict that edges exist for the node pairs with scores higher than a given threshold. Ghasemian et al. [2020] test a large body of link prediction algorithms on 550 real-world networks from six application domains (biological, economic, social, technological, information and transportation). They observe that no method provides superior prediction accuracy across all domains. To achieve reasonable adaptivity, they propose the optimal link prediction (OLP) strategy, which aggregates a large ensemble of link prediction algorithms using random forest. They first train random forest models and then refit the link prediction algorithms using all the training data. For this data set, OLP gives nearly optimal performance across all domains. In this section, we explore the adaptivity of the mixing method for this task and compare it with OLP.

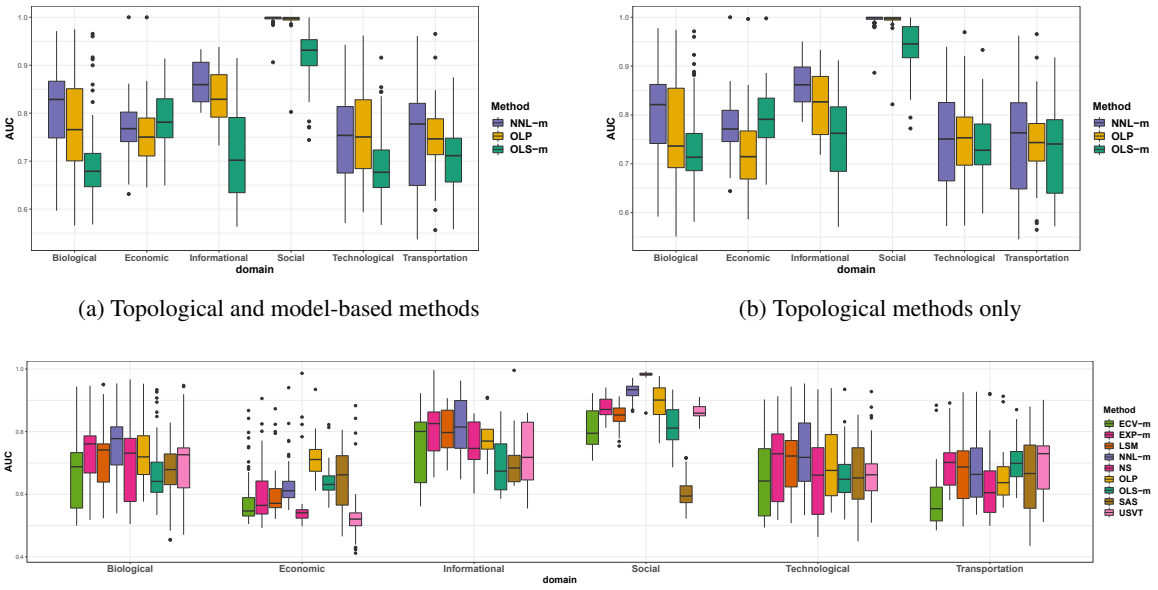
We first explain how to apply the mixing method to the link prediction task. In Ghasemian et al. [2020], three categories of link prediction algorithms are used in the ensemble: topological methods, model-based methods, embedding methods. The linear and NNL mixing methods we have introduced can include the link prediction scores from topological/embedding algorithms as individual predictions and then use either linear combination or non-negative linear combination to aggregate them. Therefore, the mixing method in this situation is similar to OLP, except that we use linear methods instead of random forest to aggregate the available estimates. Conceptually, the advantages of the mixing method are straightforward. First, it is much more computationally efficient than OLP,

especially for large networks and validation sets; this is because the sample size for random forest fitting in OLP scales as $O(n^2)$. Second, theoretical guarantees are available for the mixing method.

The data set of all the networks and the Python implementation of OLP are provided by Ghasemian [2020]. Ghasemian et al. [2020] use 42 topological network features for link prediction. We expand this set by adding the model estimates from the canonical mixing procedure – the SBM and DCBM estimates with $K = 1, 2, ..., 15$, and the USVT estimate. We use three different configurations of features to investigate the effectiveness of different categories of features: link prediction based on all 73 features (42 topological plus 31 model-based); link prediction based on the 42 topological features; and prediction based on the 31 model-based features only. In the model-based category, we also include the three graphon methods and the latent space model studied in Section 3. Notice that the model-based methods are computationally efficient, while calculating some of the 42 topological features is much more time-consuming. Therefore, the topological approaches (using all 42 features) can be computationally prohibitive for large networks.

For a stable evaluation of performance, we consider only networks with more than 200 nodes, giving in 385 networks in total. Based on their origins, they are labeled as biological (72), economic (112), informational (10), social (108), technological (55), and transportation networks (28). Given each network, we randomly sample 10% (capped at 20,000) of the node pairs to form the test set and use the complement subnetwork for training. This sampling scheme is different from the one used by Ghasemian et al. [2020], where the test set is sampled in a balanced manner to maintain similar amounts of edges and non-edges. Although that can provide better predictive performance, it may be unrealistic in practice when one has no control over the test set. We therefore take the more natural approach to sample the test set randomly. We measure the link prediction performance of a method by the predictive area under the ROC curve (AUC) based on the test data and average it over ten independent repetitions. The predictive AUC is a widely used performance metric for the link prediction task [Huang and Ling, 2005].

Figure 8 shows the results for the three categories of link prediction features. The overall link prediction accuracy varies widely across domains. Social networks are easier for link prediction, while economic, technological, and transportation networks are more difficult for the same task. When topological algorithms and mode-based algorithms are implemented, the mixing method with non-negative linear combination and OLP perform similarly in five of the six categories. OLP gives slightly worse results for the sixth category, informational networks. The mixing method with linear combination is inferior to the other two in five domains but is better for economic networks. The inferiority of linear mixing may be because 73 predictors are used, and linear combination suffers from this large number of predictors. The comparison remains similar for the methods based on topological algorithms only. Overall, for model-based algorithms, the two mixing methods are still better than the other methods. The LSM and graphon estimators might be good for one category (e.g., NS for social networks) but inferior in others. The mixing methods and OLP are reasonably good in all categories. In particular, the mixing methods are still comparable to or even better than the OLP method.



(a) Topological and model-based methods

(b) Topological methods only

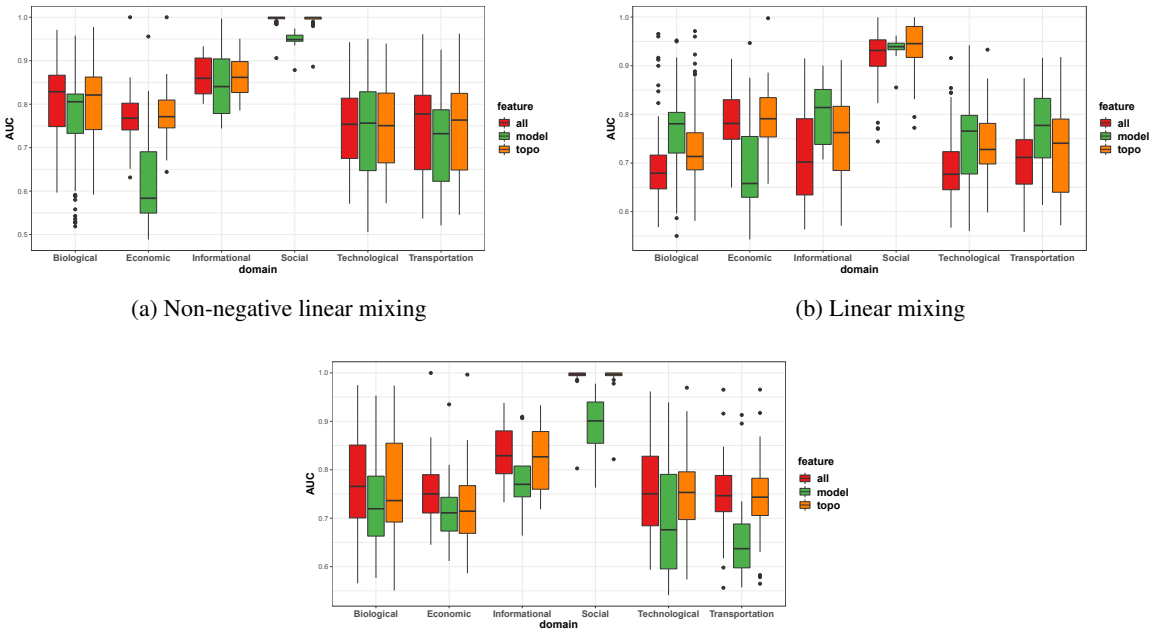
(c) Model-based methods only

Figure 8: Link prediction performance on 385 real-world networks from six domains.

Figure 9 compares the performance of each mixing strategy for different categories of features. Both OLP and non-negative linear mixing show the same pattern: the model-based prediction is worse than the topology-based prediction, while the version using both the model and topological features has similar performance as the topology-based algorithms. Linear mixing using all features is never superior in all domains. This is likely because it becomes unstable when many mixing features are used.

As practical guidance, we also want to briefly mention the speed of the above (model-based) methods. USVT is only based on the SVD of the adjacency matrix, which is usually very sparse. It is usually the most efficient one. The mixing methods need additional modeling fitting after SVD (note that the mixing needs only one round of SVD as well), but that additional cost is relatively small. So with an efficient SVD implementation [Baglama et al., 2019, Qiu and Mei, 2019], SVD and USVT can easily handle networks of size $O(10^5)$ on a single laptop. In principle, SAS has a similar speed, though we observe that it is slightly slower than mixing. OLP can generate all the features in the same way as the mixing methods, but random forest fitting can be much slower, and model-fitting cost becomes the major bottleneck. In our experiments, while OLP is feasible for networks with a few thousand nodes, it is much slower than mixing. The gradient method for the latent space model and neighborhood smoothing can handle networks of moderate size but may become too slow if the size is larger than 3000.

In summary, though the OLP method is claimed to be nearly optimal for link prediction, we find that the NNL mixing strategy delivers comparable or slightly better accuracy and adaptivity across all domains. Given its additional theoretical guarantees and much higher computational efficiency, we believe that the mixing method can generally serve as an off-the-shelf link prediction algorithm.



(a) Non-negative linear mixing

(b) Linear mixing

(c) OLP

Figure 9: Mixing performance using different features.

5 Discussion

We have introduced a mixing strategy for network estimation. It can be used as an off-the-shelf method for network modeling with both flexibility and scalability. The method is designed according to geometrical insights into the network modeling problems, and we have shown the advantage of the design theoretically. We have also demonstrated its competitive performance empirically in link prediction problems.

The current study could be extended in several directions for future follow-up. In this paper, we focus on the accuracy of the mixing estimator for a network model. In many network analysis problems, the crucial quantities can differ from the properties of the network model P itself. For example, in regression inference of network-linked data [Le and Li, 2020], the recovery of the network projection operator is critical. Intuitively, using the mixing estimator should render a more robust inference. The theory of this type of inference warrants further study. Another direction is to extend the mixing strategy to dynamic network modeling problems [Kim et al., 2018]. Flexible dynamic network models tend to be computationally challenging to fit. Given its flexibility and computational efficiency, it would be interesting to see whether the mixing method can help alleviate this difficulty.

Acknowledgement

C. M. Le is supported in part by the NSF grant DMS-2015134. T. Li is supported in part by the NSF grant DMS-2015298 and the Quantitative Collaborative Award from the College of Arts and Sciences at the University of Virginia.

References

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

E. M. Airoldi, T. B. Costa, and S. H. Chan. Stochastic blockmodel approximation of a graphon: Theory and consistent estimation. *Advances in Neural Information Processing Systems 26*, 2013.

D. J. Aldous. Representations for partially exchangeable arrays of random variables. *Journal of Multivariate Analysis*, 11(4):581–598, 1981.

J. Baglama, L. Reichel, and B. W. Lewis. *irlba: Fast Truncated Singular Value Decomposition and Principal Components Analysis for Large Dense and Sparse Matrices*, 2019. URL https://CRAN.R-project.org/package=irlba. R package version 2.3.3.

D. Banerjee and Z. Ma. Optimal hypothesis testing for stochastic block models with growing degrees. *arXiv:1705.05305*, 2017.

A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439): 509–512, 1999.

P. J. Bickel and P. Sarkar. Hypothesis testing for automated community detection in networks. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(1):253–273, 2016.

B. Bollobas, S. Janson, and O. Riordan. The phase transition in inhomogeneous random graphs. *Random Structures and Algorithms*, 31:3–122, 2007.

L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996a.

L. Breiman. Stacked regressions. *Machine learning*, 24(1):49–64, 1996b.

F. Bunea, A. B. Tsybakov, M. H. Wegkamp, et al. Aggregation for gaussian regression. *The Annals of Statistics*, 35(4):1674–1697, 2007.

O. Catoni. The mixture approach to universal model selection. In *École Normale Supérieure*. Citeseer, 1997.

S. Chan and E. Airoldi. A consistent histogram estimator for exchangeable graph models. In *International Conference on Machine Learning*, pages 208–216. PMLR, 2014.

J. Chang, E. D. Kolaczyk, and Q. Yao. Discussion of 'network cross-validation by edge sampling'. *Biometrika*, 107(2):277–280, 2020.

S. Chatterjee. Matrix estimation by universal singular value thresholding. *The Annals of Statistics*, 43(1):177–214, 2015.

D. Chen and R. J. Plemmons. Nonnegativity constraints in numerical analysis. In *The birth of numerical analysis*, pages 109–139. World Scientific, 2010.

K. Chen and J. Lei. Network cross-validation for determining the number of communities in network data. *Journal of the American Statistical Association*, 113(521):241–251, 2018.

H. Crane. *Probabilistic foundations of statistical network analysis*. Chapman and Hall/CRC, 2018.

P. Diaconis and S. Janson. Graph limits and exchangeable random graphs. *Rend. Mat. Appl.*, 28(1): 33—-61, 2008.

Y. Feng and Y. Yu. The restricted consistency property of leave-nv-out cross-validation for high-dimensional variable selection. *Statistica Sinica*, 29(3):1607–1630, 2019.

S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

C. Gao and J. Lafferty. Testing for global network structure using small subgraph statistics. *arXiv preprint arXiv:1710.00862*, 2017.

C. Gao and Z. Ma. Discussion of 'network cross-validation by edge sampling'. *Biometrika*, 107(2): 281–284, 2020.

C. Gao, Y. Lu, and H. H. Zhou. Rate-optimal graphon estimation. *The Annals of Statistics*, 43(6): 2624–2652, 2015.

C. Gao, Z. Ma, A. Y. Zhang, and H. H. Zhou. Achieving optimal misclassification proportion in stochastic block models. *The Journal of Machine Learning Research*, 18(1):1980–2024, 2017.

S. Ghafouri and S. H. Khasteh. A survey on exponential random graph models: an application perspective. *PeerJ Computer Science*, 6:e269, 2020.

A. Ghasemian. Optimal link prediction. https://github.com/Aghasemian/OptimalLinkPrediction, 2020.

A. Ghasemian, H. Hosseinmardi, A. Galstyan, E. M. Airoldi, and A. Clauset. Stacking models for nearly optimal link prediction in complex networks. *Proceedings of the National Academy of Sciences*, 117(38):23393–23400, 2020.

A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi. A survey of statistical network models. *Foundations and Trends® in Machine Learning*, 2(2):129–233, 2010.

T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*, volume 2. Springer, 2009.

D. Haussler, J. Kivinen, and M. K. Warmuth. Sequential prediction of individual sequences under general loss functions. *IEEE Transactions on Information Theory*, 44(5):1906–1925, 1998.

P. Hoff. Modeling homophily and stochastic equivalence in symmetric relational data. In *Advances in Neural Information Processing Systems*, pages 657–664, 2008.

P. D. Hoff, A. E. Raftery, and M. S. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002.

P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.

J. Huang and C. X. Ling. Using auc and accuracy in evaluating learning algorithms. *IEEE Transactions on knowledge and Data Engineering*, 17(3):299–310, 2005.

J. Jin, Z. T. Ke, and S. Luo. Estimating network memberships by simplex vertex hunting. *arXiv preprint arXiv:1708.07852*, 2017.

J. Jin, Z. T. Ke, and S. Luo. Optimal adaptivity of signed-polygon statistics for network testing. *arXiv preprint arXiv:1904.09532*, 2019.

A. Juditsky and A. Nemirovski. Functional aggregation for nonparametric regression. *Annals of Statistics*, pages 681–712, 2000.

B. Karrer and M. E. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107, 2011.

B. Kim, K. H. Lee, L. Xue, and X. Niu. A review of dynamic network models with latent variables. *Statistics surveys*, 12:105, 2018.

D. Kim, S. Sra, and I. S. Dhillon. *A new projected quasi-newton approach for the nonnegative least squares problem*. Citeseer, 2006.

Y. Klochkov and N. Zhivotovskiy. Uniform hanson-wright type concentration inequalities for unbounded entries via the entropy method. *Theory of Probability and Mathematical Statistics*, 25 (22):1–30, 2020.

C. M. Le. Edge sampling using local network information. *Journal of Machine Learning Research*, 22(88):1–29, 2021.

C. M. Le and E. Levina. Estimating the number of communities in networks by spectral methods. *arXiv preprint arXiv:1507.00827*, 2015.

C. M. Le and T. Li. Linear regression and its inference on noisy network-linked data. *arXiv preprint arXiv:2007.00803*, 2020.

C. M. Le, E. Levina, and R. Vershynin. Concentration and regularization of random graphs. *Random Structures & Algorithms*, 51(3):538–561, 2017.

M. LeBlanc and R. Tibshirani. Combining estimates in regression and classification. *Journal of the American Statistical Association*, 91(436):1641–1650, 1996.

J. Lei. A goodness-of-fit test for stochastic block models. *The Annals of Statistics*, 44(1):401–424, 2016.

J. Lei. Cross-validation with confidence. *Journal of the American Statistical Association*, 115(532): 1978–1997, 2020.

J. Lei and A. Rinaldo. Consistency of spectral clustering in stochastic block models. *The Annals of Statistics*, 43(1):215–237, 2014.

T. Li, L. Lei, S. Bhattacharyya, K. Van den Berge, P. Sarkar, P. J. Bickel, and E. Levina. Hierarchical community detection by recursive partitioning. *Journal of the American Statistical Association*, pages 1–18, 2020a.

T. Li, E. Levina, and J. Zhu. Network cross-validation by edge sampling. *Biometrika*, 107(2): 257–276, 2020b.

T. Li, E. Levina, and J. Zhu. Rejoinder:'network cross-validation by edge sampling'. *Biometrika*, 107(2):289–292, 2020c.

T. Li, E. Levina, and J. Zhu. *randnet: Random Network Model Estimation, Selection and Parameter Tuning*, 2021. URL https://CRAN.R-project.org/package=randnet. R package version 0.3.

D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.

R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla. New perspectives and methods in link prediction. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 243–252, 2010.

L. Lovász and B. Szegedy. Limits of dense graph sequences. *J. Combin. Theory Ser. B*, 96(6): 933—-957, 2006.

L. Lü and T. Zhou. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, 390(6):1150–1170, 2011.

Z. Ma, Z. Ma, and H. Yuan. Universal latent space model fitting for large networks with edge covariates. *Journal of Machine Learning Research*, 21(4):1–67, 2020.

N. Meinshausen et al. Sign-constrained least squares estimation for high-dimensional regression. *Electronic Journal of Statistics*, 7:1607–1631, 2013.

R. Miao and T. Li. Informative core identification in complex networks. *arXiv preprint arXiv:2101.06388*, 2021.

R. Mukherjee and S. Sen. Testing degree corrections in stochastic block models. *arXiv preprint arXiv:1705.07527*, 2017.

M. Newman. *Networks: an introduction*. Oxford university press, 2010.

M. Noroozi, M. Pensky, and R. Rimal. Sparse popularity adjusted stochastic block model. *arXiv preprint arXiv:1910.01931*, 2019.

Y. Qiu and J. Mei. *RSpectra: Solvers for Large-Scale Eigenvalue and SVD Problems*, 2019. URL https://CRAN.R-project.org/package=RSpectra. R package version 0.16-0.

K. Rohe and T. Qin. The blessing of transitivity in sparse and stochastic networks. *arXiv:1307.2302*, 2013.

K. Rohe, S. Chatterjee, and B. Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915, 2011.

D. Saldana, Y. Yu, and Y. Feng. How many communities are there? *Journal of Computational and Graphical Statistics*, 26:171–181, 2017.

S. Sengupta and Y. Chen. A block model for node popularity in networks with community structure. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(2):365–386, 2018.

M. Slawski and M. Hein. Sparse recovery by thresholded non-negative least squares. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

M. Slawski, M. Hein, et al. Non-negative least squares for high-dimensional linear models: Consistency and sparse recovery without regularization. *Electronic Journal of Statistics*, 7:3004–3056, 2013.

D. A. Spielman and N. Srivastava. Graph sparsification by effective resistances. *SIAM J. Comput.*, 40(6):1913–1926, 2011.

C. Steglich, T. A. Snijders, and P. West. Applying siena: An illustrative analysis of the coevolution of adolescents' friendship networks, taste in music, and alcohol consumption. *Methodology: European Journal of Research Methods for the Behavioral and Social Sciences*, 2(1):48, 2006.

M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):111–133, 1974.

A. B. Tsybakov. Optimal rates of aggregation. *Learning Theory and Kernel Machines. Lecture Notes in Computer Science*, 2777:303–313, 2003.

J. Ugander, L. Backstrom, and J. Kleinberg. Subgraph frequencies: Mapping the empirical and extremal geography of large graph collections. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1307–1318, 2013.

M. J. van der Laan, D. Sandrine, and A. W. van der Vaart. The cross-validated adaptive epsilon-net estimator. *Statistics & Risk Modeling*, 24(3):1–23, 2006.

M. J. van der Laan, E. C. Polley, and A. E. Hubbard. Super learner. *Statistical Applications in Genetics and Molecular Biology*, 6(25), 2007.

Y. R. Wang and P. J. Bickel. Likelihood-based model selection for stochastic block models. *The Annals of Statistics*, 45(2):500–528, 2017.

D. H. Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

B. Yan, P. Sarkar, and X. Cheng. Provable estimation of the number of blocks in block models. *arXiv preprint arXiv:1705.08580*, 2017.

Y. Yang. Mixing strategies for density estimation. *The Annals of Statistics*, 28(1):75–87, 2000.

Y. Yang. Adaptive regression by mixing. *Journal of the American Statistical Association*, 96(454): 574–588, 2001.

K. You. *graphon: A Collection of Graphon Estimation Methods*, 2020. URL https://CRAN.R-project.org/package=graphon. R package version 0.3.4.

S. J. Young and E. R. Scheinerman. Random dot product graph models for social networks. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 138–149. Springer, 2007.

L. Zhang and A. A. Amini. Adjusted chi-square test for degree-corrected block models. *arXiv:2012.15047*, 2020.

Y. Zhang, E. Levina, and J. Zhu. Estimating network edge probabilities by neighbourhood smoothing. *Biometrika*, 104(4):771–783, 2017.

A Exponential weight mixing

Proof of Theorem 1. From (2), we have

$$\begin{aligned}\pi_r &\propto \exp\left(-\sum_{(i,j)\in\Omega}\left(A_{ij}-\hat{P}_{ij}^{(r)}\right)^2\right)\\ &\propto \exp\left(-\sum_{(i,j)\in\Omega}\left(A_{ij}-\hat{P}_{ij}^{(r)}\right)^2+\sum_{(i,j)\in\Omega}(A_{ij}-P_{ij})^2\right) =: \exp\left(-S^{(r)}\right),\end{aligned}$$

where $\propto$ denotes "proportional to". Rewrite $S^{(r)}$ as follows:

$$S^{(r)}=\sum_{(i,j)\in\Omega}\left(P_{ij}-\hat{P}_{ij}^{(r)}\right)\left(2A_{ij}-P_{ij}-\hat{P}_{ij}^{(r)}\right) =: \sum_{(i,j)\in\Omega} X_{ij}^{(r)}.$$

Denote $\rho = \max_{i,j} P_{ij}$. Conditioning on $A(\Omega^c)$ (thus on $\hat{P}^{(r)}$), $S^{(r)}$ is the sum of independent random variables $X_{ij}^{(r)}$ with mean and variance given by

$$\mathbb{E}X_{ij}^{(r)}=\left(P_{ij}-\hat{P}_{ij}^{(r)}\right)^2, \quad \text{var}\left(X_{ij}^{(r)}\right)=4P_{ij}(1-P_{ij})\left(P_{ij}-\hat{P}_{ij}^{(r)}\right)^2\le 4\rho\left(P_{ij}-\hat{P}_{ij}^{(r)}\right)^2.$$

Since $\left|X_{ij}^{(r)}\right|\le 8$, by Bernstein's inequality,

$$\mathbb{P}\left(\left|S^{(r)}-\mathbb{E}S^{(r)}\right|>t\right)\le 2\exp\left(\frac{-t^2/2}{4\rho\sum_{(i,j)\in\Omega}\left(P_{ij}-\hat{P}_{ij}^{(r)}\right)^2+8t/3}\right).$$

Since $m=o(n^2)$, we have

$$\left|S^{(r)}-\sum_{(i,j)\in\Omega}\left(P_{ij}-\hat{P}_{ij}^{(r)}\right)^2\right|\le C\left(\rho\log n\sum_{(i,j)\in\Omega}(P_{ij}-\hat{P}_{ij}^{(r)})^2\right)^{1/2}+C\log n \tag{14}$$

for all $1\le r\le m$ with high probability for a sufficiently large constant C. In other words, $S^{(r)}$ concentrates well around the mean $\sum_{(i,j)\in\Omega}\left(P_{ij}-P_{ij}^{(r)}\right)^2$, and the mixture of $P^{(r)}$ puts exponentially large weight on the estimator $P^{(r)}$ with the smallest error $\|P^{(r)}(\Omega)-P(\Omega)\|_F^2$.

Let $r^*=\text{argmin}_{1\le r\le m}\|P^{(r)}(\Omega)-P(\Omega)\|_F^2$ and denote

$$x_r=\|P^{(r)}(\Omega)-P(\Omega)\|_F, \quad x=\|P^{(r^*)}(\Omega)-P(\Omega)\|_F.$$

Consider the index set

$$I=\left\{r\in[m]: x_r\le x+2C\sqrt{\rho\log n}+\frac{2C\log(nm)}{x}\right\}.$$

By the triangle inequality,

$$\Big\| \sum_{r=1}^{m} \pi_r P^{(r)}(\Omega) - P(\Omega) \Big\|_F \le \sum_{r=1}^{m} \pi_r \|P^{(r)}(\Omega) - P(\Omega)\|_F = \sum_{r\in I} \pi_r x_r + \sum_{r\notin I} \pi_r x_r.$$

From the definition of I we get

$$\sum_{r\in I} \pi_r x_r \le x + 2C\sqrt{\rho \log n} + \frac{2C\log(nm)}{x}.$$

For the second sum, consider $r \notin I$. Recall that $\pi_r \propto \exp(-S^{(r)})$ and by (14), $S^{(r)}$ and $S^{(r^*)}$ concentrate around x_r^2 and x^2, respectively. Therefore with high probability we have

$$\pi_r = \frac{\exp(-S^{(r)})}{\sum_r \exp(-S^{(r)})} \quad \le \quad \exp\left(S^{(r*)} - S^{(r)}\right) \le \exp\left(x^2 - x_r^2 + 2C\sqrt{\rho\log n}(x + x_r) + 2C\log n\right).$$

Since $r \notin I$,

$$x_r^2 - x^2 - 2C\sqrt{\rho\log n}(x_r + x) = (x_r + x)(x_r - x - 2C\sqrt{\rho\log n}) \ge 4C\log(nm),$$

and consequently,

$$\pi_r \le \exp\left(-2C\log(nm)\right) = (nm)^{-2C}.$$

Since $x_r \le n$, by choosing $C > 1$, we get

$$\sum_{r\notin I} \pi_r \|P^{(r)}(\Omega) - P(\Omega)\|_F \le \sum_{r\notin I} n(nm)^{-2C} = o(1).$$

In summary, we have proved that with high probability,

$$\Big\| \sum_{r=1}^{m} \pi_r P^{(r)}(\Omega) - P(\Omega) \Big\|_F \le \min_{1\le r\le m} \big\|P^{(r)}(\Omega) - P(\Omega)\big\|_F + \delta,$$

where

$$\delta = 3C\Big(\log n \cdot \max_{i,j} P_{i,j}\Big)^{1/2} + \frac{3C\log(nm)}{\min_{1\le r\le m}\|P^{(r)}(\Omega) - P(\Omega)\|_F}.$$

In case $\min_{1\le r\le m}\|P^{(r)}(\Omega) - P(\Omega)\|_F \le 1$, we can replace $\log(nm)/x$ in the definition of I by $\log^{1/2}(nm)$ and repeat the above argument. Again, for constant $C > 1$, the same derivations can still go through and the resulting error would be

$$\delta = 3C\Big(\log n \cdot \max_{i,j} P_{i,j}\Big)^{1/2} + 3C\log^{1/2}(nm).$$

Consequently, the error can be improved as follows:

$$\delta = 3C\Big(\log n \cdot \max_{i,j} P_{ij}\Big)^{1/2} + 3C\min\left\{\frac{\log(nm)}{\min_{1\le r\le m}\|P^{(r)}(\Omega) - P(\Omega)\|_F},\ \log^{1/2}(nm)\right\}.$$

Rewriting $3C$ as a new constant C completes the proof. □

B Non-negative linear aggregation

Proof of Theorem 2. Denote $E = A(\Omega) - P(\Omega)$. By the triangle inequality,

$$\|\Pi_{\mathcal{C}} A(\Omega) - P(\Omega)\|_F \le \|\Pi_{\mathcal{C}} E\|_F + \|\Pi_{\mathcal{C}} P(\Omega) - P(\Omega)\|_F. \tag{15}$$

We now show that $\|\Pi_{\mathcal{C}} E\|_F$ is small due to condition (8). Since $\mathcal{C}$ is a cone, whether $\|\Pi_{\mathcal{C}} E\|_F = 0$ or $\|\Pi_{\mathcal{C}} E\|_F > 0$, depending on the relative location of E to $\mathcal{C}$. If $\|\Pi_{\mathcal{C}} E\|_F > 0$ then there exists $\nu \in \mathcal{C}$ with $\|\nu\|_F = 1$ such that $\|\Pi_{\mathcal{C}} E\|_F = \langle \nu, E\rangle$. Let $\nu^{(r)}$ be the normalized predictor

$$\nu^{(r)} = \|\hat{P}^{(r)}(\Omega)\|_F^{-1} \hat{P}^{(r)}(\Omega)$$

and $\nu = \sum_{r=1}^m \lambda_r \nu^{(r)}$ for some $\lambda_1, ..., \lambda_m \ge 0$. Then by (8),

$$1 = \|\nu\|_F^2 = \sum_{1\le r,s\le m} \lambda_r \lambda_s \langle \nu^{(r)}, \nu^{(s)}\rangle \ge \delta \Big(\sum_{r=1}^m \lambda_r \Big)^2,$$

which implies $\sum_{r=1}^m \lambda_r \le \delta^{-1/2}$. Therefore

$$\|\Pi_{\mathcal{C}} E\|_F = \langle \nu, E\rangle = \sum_{r=1}^m \lambda_r \langle \nu^{(r)}, E\rangle \le \delta^{-1/2} \max_{1\le r\le m} \langle \nu^{(r)}, E\rangle.$$

For each $\nu^{(r)} = \big(\nu_{ij}^{(r)}\big)$ with $\|\nu^{(r)}\| := \max_{(i,j)\in\Omega} |\nu_{ij}^{(r)}|$, by Bernstein's inequality,

$$\begin{aligned} \mathbb{P}\big(|\langle \nu^{(r)}, E\rangle| > t\big) &\le \exp\left(\frac{-t^2/2}{\sum_{(i,j)\in\Omega} \big(\nu_{ij}^{(r)}\big)^2 \mathrm{var}(E_{ij}) + \|\nu^{(r)}\|_\infty t/3} \right) \\ &\le \exp\left(\frac{-t^2/2}{\|P(\Omega)\|_\infty + \|\nu^{(r)}\|_\infty t/3} \right). \end{aligned}$$

Choosing $t \approx \max_{1\le r\le m} \|\nu^{(r)}\|_\infty \log(n+m) + (\|P(\Omega)\|_\infty \log(n+m))^{1/2}$ and applying the union bound, we obtain that with high probability,

$$\|\Pi_{\mathcal{C}} E\|_F \le C\delta^{-1/2} \left(\max_{1\le r\le m} \|\nu^{(r)}\|_\infty \log(n+m) + (\|P(\Omega)\|_\infty \log(n+m))^{1/2} \right).$$

The proof is complete. □

Proof of Corollary 1. Denote $E = A(\Omega) - P(\Omega)$. We can view E and $\Pi_{\mathcal{L}}$ as a vector and a matrix in $\mathbb{R}^{|\Omega|}$ and $\mathbb{R}^{|\Omega|\times|\Omega|}$, respectively. Then $\|\Pi_{\mathcal{L}} A(\Omega) - \Pi_{\mathcal{L}} P(\Omega)\|_F^2 = \|\Pi_{\mathcal{L}} E\|_F^2$. Denoting by $\circ$ the entry-wise product, we get

$$\mathbb{E}\|\Pi_{\mathcal{L}} E\|_F^2 = \mathbb{E}\langle \Pi_{\mathcal{L}} E, \Pi_{\mathcal{L}} E\rangle = \mathrm{trace}(\Pi_{\mathcal{L}} \mathrm{diag}(P(\Omega) \circ (I - P(\Omega))) \Pi_{\mathcal{L}}^T) \tag{16}$$

So we have

$$m\sqrt{\rho} = m \cdot \min_{(i,j)\in\Omega} P_{ij} \cdot \Big(1 - \max_{(i,j)\in\Omega} P_{ij}\Big) \le \mathbb{E}\|\Pi_{\mathcal{L}} E\|_F^2 \le m \cdot \max_{(i,j)\in\Omega} P_{ij}. \tag{17}$$

Furthermore, by the Cauchy–Schwarz inequality,

$$(\mathbb{E}\|\Pi_{\mathcal{L}}E\|_F)^2 \le \mathbb{E}\|\Pi_{\mathcal{L}}E\|_F^2 \le m \cdot \|P(\Omega)\|_\infty. \tag{18}$$

Since $\|\Pi_{\mathcal{L}}\| = 1$, by Hanson-Wright inequality [Klochkov and Zhivotovskiy, 2020, Theorem 1.1] and (17), for any $t \ge \max\{\mathbb{E}\|\Pi_{\mathcal{L}}E\|_F, 1\}$ we have

$$\begin{aligned}\mathbb{P}\left(\left|\|\Pi_{\mathcal{L}}E\|_F^2 - \mathbb{E}\|\Pi_{\mathcal{L}}E\|_F^2\right| > t\right) &\le \exp\left(-c\min\left\{\frac{t^2}{(\mathbb{E}\|\Pi_{\mathcal{L}}E\|_F)^2}, t\right\}\right)\\ &\le \exp\left(-c\min\left\{\frac{t^2}{\mathbb{E}\|\Pi_{\mathcal{L}}E\|_F^2}, t\right\}\right).\end{aligned}$$

Choosing $t = (m\log n \cdot \|P(\Omega)\|_\infty)^{1/2}$ and using (18), we see that

$$t > \max\{\mathbb{E}\|\Pi_{\mathcal{L}}E\|_F, 1\}.$$

Then (17) leads to

$$\|\Pi_{\mathcal{L}}E\|_F^2 \le m \cdot \max_{(i,j)\in\Omega} P_{ij} + (m\log n \cdot \|P(\Omega)\|_\infty)^{1/2} \tag{19}$$

with high probability. Combining the assumption $m \ge 4\rho^{-1}\|P(\Omega)\|_\infty \log n$ and (17), we can see that

$$\mathbb{E}\|\Pi_{\mathcal{L}}E\|_F^2 \ge m\sqrt{\rho} > 2\sqrt{m\|P(\Omega)\|_\infty \log n} = 2t > t$$

and therefore with high probability

$$\|\Pi_{\mathcal{L}}E\|_F^2 \ge m \cdot \min_{(i,j)\in\Omega} P_{ij} \cdot \left(1 - \|P(\Omega)\|_\infty\right) - (m\log n \cdot \|P(\Omega)\|_\infty)^{1/2} > \frac{m\sqrt{\rho}}{2}. \tag{20}$$

Notice that (19) directly indicates (11). Now we have

$$\begin{aligned}\Delta &= \|\hat{P}^{(\mathrm{ols})}(\Omega) - P(\Omega)\|_F^2 - \|\hat{P}^{(\mathrm{nnl})}(\Omega) - P(\Omega)\|_F^2\\ &= \|\Pi_{\mathcal{L}}A(\Omega) - \Pi_{\mathcal{L}}P(\Omega)\|_F^2 + \|\Pi_{\mathcal{L}}P(\Omega) - P(\Omega)\|_F^2\\ &\quad -\|\Pi_{\mathcal{C}}A(\Omega) - \Pi_{\mathcal{L}}P(\Omega)\|_F^2 - \|\Pi_{\mathcal{L}}P(\Omega) - P(\Omega)\|_F^2\\ &\ge \|\Pi_{\mathcal{L}}E\|_F^2 - \|\Pi_{\mathcal{C}}E\|_F^2 - \|\,\Pi_{\mathcal{C}}\Pi_{\mathcal{L}}P(\Omega) - \Pi_{\mathcal{L}}P(\Omega)\|_F^2.\end{aligned}$$

On the other hand, by Theorem 2 and (20),

$$\Delta \ge \frac{m\rho^{1/2}}{2} - \frac{C^2}{\delta}\epsilon^2 - \|\Pi_{\mathcal{C}}\Pi_{\mathcal{L}}P(\Omega) - \Pi_{\mathcal{L}}P(\Omega)\|_F^2$$

with high probability, and the proof is completed by using the basic inequality $(a+b)^2 \le 2(a^2+b^2)$ for ϵ^2. □

Proof of Theorem 3. We will mainly follow the proof strategy of Slawski and Hein [2011]. However, there is one key step in their proof (right after B.5) that may not go through. So our proof can be seen as a corrected version of that.

Recall the definition of $\hat{\beta}$ in (6). Since this is a constrained linear regression problem, for the notation simplicity, let us denote $X_r = \hat{P}^{(r)}(\Omega)$, $Y = A(\Omega)$, $\mu = P(\Omega)$, and $E = A(\Omega) - P(\Omega) = Y - \mu$.

We view them as column vectors and further denote $X = (X_1, ..., X_m)$. The optimization problem (6) is equivalent to

$$\hat{\pi} = \underset{\pi \succeq 0}{\operatorname{argmin}} \|Y - X\pi\|^2. \tag{21}$$

Consider the oracle parameter

$$\pi^* = \underset{\pi \succeq 0}{\operatorname{argmin}} \|\mu - X\pi\|^2$$

and define $\delta = \pi^* - \pi$ for any $\pi \succeq 0$. In particular, we write $\hat{\delta} = \pi^* - \hat{\pi}$.

Our goal is to compare the NNL mixing estimate $X\hat{\pi}$ with the best linear approximation $X\pi^*$ in the noiseless setting when μ is known. We first rewrite the objective function in (21) as follows:

$$\begin{aligned} \|Y - X\pi\|^2 &= \|\mu - X\pi^* + X(\pi^* - \pi) + E\|^2 \\ &= \|\mu - X\pi^*\|^2 + \|X\delta\|^2 + \|E\|^2 + 2E^T(\mu - X\pi^*) + 2E^T X\delta. \end{aligned}$$

Note that the constraint $\pi \succeq 0$ is equivalent to $\delta \preceq \pi^*$, and only the second and the last terms of the expression above depend on δ, it follows that

$$\hat{\delta} = \underset{\delta \preceq \pi^*}{\operatorname{argmin}} \left\{ \|X\delta\|^2 + 2E^T X\delta \right\}. \tag{22}$$

Since the objective function on the right-hand side is zero when $\delta = 0$, its value at the minimizer $\delta = \hat{\delta}$ is at most zero. Equivalently, we have

$$\|X\hat{\delta}\|^2 \leq 2E^T X\hat{\delta}.$$

For a vector $x \in \mathbb{R}^m$, denote

$$S_+(x) = \{i \in [m] : x_i \geq 0\}, \quad S_-(x) = \{i \in [m] : x_i < 0\}.$$

Let x_P be the vector obtained from x by setting all entries within $S_-(x)$ to zero and $x_N = x - x_P$. The inequality above implies

$$\|X\hat{\delta}\|^2 \leq 2\|E^T X\|_\infty \|\hat{\delta}\|_1 \leq 2\|E^T X\|_\infty \big(\|\hat{\delta}_P\|_1 + \|\hat{\delta}_N\|_1\big) \leq 2\|E^T X\|_\infty \big(\|\pi^*\|_1 + \|\hat{\delta}_N\|_1\big), \tag{23}$$

where $\|\hat{\delta}_P\|_1 \leq \|\pi^*\|_1$ because $\hat{\delta} = \pi^* - \hat{\pi} \leq \pi^*$ for $\hat{\pi} \succeq 0$. It remains to bound $\|\hat{\delta}_N\|_1$.

Denote $\Sigma = X^T X$. For each $\delta \leq \pi^*$, let Σ_{PP}, Σ_{NN} and Σ_{NP} be the matrices obtained from Σ by setting all entries with indices (i, j) outside $S_+(\delta) \times S_+(\delta)$, $S_-(\delta) \times S_-(\delta)$ and $S_-(\delta) \times S_+(\delta)$ to zero, respectively. Then (22) is equivalent to

$$\hat{\delta} = \underset{\delta \preceq \pi^*}{\operatorname{argmin}} \left\{ \delta_P^T \Sigma_{PP} \delta_P + \delta_N^T \Sigma_{NN} \delta_N + 2\delta_N^T \Sigma_{NP} \delta_P + 2E^T X(\delta_N + \delta_P) \right\}.$$

Define $\mathcal{T} = \{x \in \mathbb{R}^m : x \leq \pi^*, x_i \leq 0, i \in S_-(\hat{\delta}), x_i = 0, i \in S_+(\hat{\delta})\}$. Replacing δ_P in the objective function above on the right-hand side with $\hat{\delta}_P$, we see that

$$\hat{\delta}_N = \underset{\delta_N \in \mathcal{T}}{\operatorname{argmin}} \left\{ \delta_N^T \Sigma_{NN} \delta_N + 2\delta_N^T \Sigma_{NP} \hat{\delta}_P + 2E^T X \delta_N \right\}.$$

Since the objective function on the right-hand side is zero when $\delta_N = 0 \in \Omega$,

$$\hat{\delta}_N^T \Sigma_{NN} \hat{\delta}_N + 2\hat{\delta}_N^T \Sigma_{NP} \hat{\delta}_P + 2E^T X \hat{\delta}_N \leq 0.$$

Denote $\|\Sigma\|_\infty = \max_{1\leq r,s\leq m} |\Sigma_{rs}|$. By the self-regularizing property (13), we have

$$\begin{aligned} 0 &\geq \kappa \|\hat{\delta}_N\|_1^2 \|\Sigma\|_\infty - 2\|\hat{\delta}_N\|_1 \|\hat{\delta}_P\|_1 \|\Sigma\|_\infty + 2E^T X \hat{\delta}_N \\ &\geq \kappa \|\hat{\delta}_N\|_1^2 \|\Sigma\|_\infty - 2\|\hat{\delta}_N\|_1 \|\hat{\delta}_P\|_1 \|\Sigma\|_\infty - 2\|E^T X\|_\infty \|\hat{\delta}_N\|_1. \end{aligned}$$

It follows that

$$\|\hat{\delta}_N\|_1 \leq \frac{2}{\kappa} \left(\|\hat{\delta}_P\|_1 + \frac{\|E^T X\|_\infty}{\|\Sigma\|_\infty} \right) \leq \frac{2}{\kappa} \left(\|\pi^*\|_1 + \frac{\|E^T X\|_\infty}{\|\Sigma\|_\infty} \right). \tag{24}$$

From (23) and (24), we get

$$\|X\hat{\pi} - X\pi^*\|^2 \leq \frac{4\|E^T X\|_\infty^2}{\kappa \|\Sigma\|_\infty} + \frac{(2\kappa + 4)\|E^T X\|_\infty \|\pi^*\|_1}{\kappa}.$$

By Bernstein's inequality on the Bernoulli error E, for each $1 \leq r \leq m$, we have

$$\mathbb{P}\left(|E^T X_r| > t \right) \leq 2 \exp\left(\frac{-t^2/2}{\|P(\Omega)\|_\infty \|\hat{P}^{(r)}(\Omega)\|_F^2 + t\|\hat{P}^{(r)}(\Omega)\|_\infty / 3} \right).$$

Choosing $t \approx \|\hat{P}^{(r)}(\Omega)\|_\infty \log(n+m) + \|\hat{P}^{(r)}(\Omega)\|_F (\|P(\Omega)\|_\infty \log(n+m))^{1/2}$ for each r and using the union bound across r, we obtain the final bound for $\|X\hat{\pi} - X\pi^*\|^2 = \|\hat{P}(\Omega) - \Pi_{\mathcal{C}} P(\Omega)\|_F^2$ with

$$\Phi = \max_r \|\hat{P}^{(r)}(\Omega)\|_\infty + \|P(\Omega)\|_\infty^{1/2} \max_r \|\hat{P}^{(r)}(\Omega)\|_F = \max_r \|\hat{P}^{(r)}(\Omega)\|_\infty + \|P(\Omega)\|_\infty^{1/2} \|\Sigma\|_\infty.$$

The proof is complete. □