Biometrika (2017), **104**, 4, *pp.* 771–783
Printed in Great Britain

doi: 10.1093/biomet/asx042
Advance Access publication 15 September 2017

Estimating network edge probabilities by neighbourhood smoothing

By YUAN ZHANG

Department of Statistics, Ohio State University, 404 Cockins Hall,
1958 Neil Avenue, Columbus, Ohio 43210, U.S.A,
yzhanghf@stat.osu.edu

ELIZAVETA LEVINA and JI ZHU

Department of Statistics, University of Michigan, 311 West Hall,
1085 South University Avenue, Ann Arbor, Michigan 48109, U.S.A.
elevina@umich.edu   jizhu@umich.edu

Summary

The estimation of probabilities of network edges from the observed adjacency matrix has important applications to the prediction of missing links and to network denoising. It is usually addressed by estimating the graphon, a function that determines the matrix of edge probabilities, but this is ill-defined without strong assumptions on the network structure. Here we propose a novel computationally efficient method, based on neighbourhood smoothing, to estimate the expectation of the adjacency matrix directly, without making the structural assumptions that graphon estimation requires. The neighbourhood smoothing method requires little tuning, has a competitive mean squared error rate and outperforms many benchmark methods for link prediction in simulated and real networks.

Some key words: Graphon estimation; Network analysis; Nonparametric statistics.

1. Introduction

Statistical network analysis spans a wide range of disciplines, including network science, statistics, physics, computer science and sociology, and has an equally wide range of applications and analysis tasks such as community detection and link prediction. In this paper, we study the problem of inferring the generative mechanism of an undirected network based on a single realization of the network. The data consist of the network adjacency matrix $A \in \{0,1\}^{n\times n}$, where n is the number of nodes and $A_{ij} = A_{ji} = 1$ if there is an edge between nodes i and j. We assume that the observed adjacency matrix A is generated from an underlying probability matrix P, so that for $i \leqslant j$, the A_{ij} are independent $\mathrm{Ber}(P_{ij})$ trials and the P_{ij} are edge probabilities.

It is impossible to estimate P from a single realization of A unless one assumes some form of structure in P. When the network is expected to have communities, arguably the most popular assumption is that of the stochastic block model, where each node belongs to one of K blocks and the probability of an edge between two nodes is determined by the block to which the nodes belong. In this case, the $n \times n$ matrix P is parameterized by the $K \times K$ matrix of within- and between-block edge probabilities, and thus it is possible to estimate P from a single realization. The main challenge in fitting the stochastic block model lies in estimating the blocks themselves, and that has been the focus of the literature to date; see for example Bickel & Chen (2009), Rohe et al. (2011), Amini et al. (2013), Saade et al. (2014) and Guédon & Vershynin (2016). Once the blocks have been estimated, P can be estimated efficiently by a plug-in moment estimator. Many extensions and alternatives to the stochastic block model have been proposed to model networks with communities, including those of Hoff (2008), Airoldi et al. (2008), Karrer & Newman (2011), Cai & Li (2015) and Zhang et al. (2015), but their properties are generally known only under the correctly specified model with communities. Here we are interested in estimating P for more general networks.

A general representation of the matrix P for unlabelled exchangeable networks goes back to Aldous (1981) and a 1979 preprint by D. N. Hoover entitled 'Relations on probability spaces and arrays of random variables'. Formally, a network is exchangeable if for any permutation π of the set $\{1, \ldots, n\}$, the distribution of edges remains the same. That is, if the adjacency matrix $A = [A_{ij}]$ is drawn from the probability matrix P, which we write as $A \sim P$, then for any permutation π,

$$\left[A_{\pi(i)\pi(j)}\right] \sim P. \tag{1}$$

In Aldous (1981) and in the preprint by Hoover, it was shown that an exchangeable network always admits the following Aldous–Hoover representation:

Definition 1. *For any network satisfying* (1)*, there exists a function* $f : [0, 1] \times [0, 1] \to [0, 1]$ *and a set of independent and identically distributed random variables* $\xi_i \sim \mathrm{Un}[0, 1]$ *such that*

$$P_{ij} = f(\xi_i, \xi_j).$$

Following the literature, we call f the graphon function. Unfortunately, as pointed out in Diaconis & Janson (2007), f in this representation is neither unique nor identifiable, since for any measure-preserving one-to-one transformation $\sigma : [0, 1] \to [0, 1]$, both $f\{\sigma(u), \sigma(v)\}$ and $f(u, v)$ yield the same distribution of A. An identifiable and unique canonical representation can be defined if one requires $g(u) = \int_0^1 f(u, v)\,\mathrm{d}v$ to be nondecreasing (Bickel & Chen, 2009). Chan & Airoldi (2014) show that f and the ξ_i are jointly identifiable when $g(u)$, which can be interpreted as the expected node degree, is strictly monotone. This assumption is strong and excludes the stochastic block model.

In practice, the main purpose of estimating f is to estimate P, and thus identifiability of f or lack thereof may not matter if P itself can be estimated. In the preprint by Hoover and in Diaconis & Janson (2007), it was shown that the measure-preserving map σ is the only source of nonidentifiability. Wolfe & Olhede (2013) and Choi & Wolfe (2014) proposed estimating f up to a measure-preserving transformation σ via step-function approximations based on fitting the stochastic block model with a larger number of blocks, K. This approximation does not assume that the network itself follows the block model, and some theoretical guarantees have been obtained under more general models. In related work, Olhede & Wolfe (2014) proposed to approximate the graphon with so-called network histograms, that is, stochastic block models with many blocks of equal size, akin to histogram bins. Another method for computing a network histogram was proposed by Amini & Levina (2016), as an application of their semidefinite programming approach to fitting block models with equal-size blocks. Gao et al. (2015a) established the minimax error rate for estimating P and proposed a least-squares-type estimator to achieve this rate, which obtains the estimated probability P by averaging the adjacency matrix elements within a given block partition. A similar estimator was proposed in Choi (2017), applicable also to nonsmooth graphons. However, these methods are in principle computationally infeasible since they require an exhaustive enumeration of all possible block partitions. Cai et al. (2015) proposed an iterative algorithm to fit a stochastic block model and approximate the graphon, but its error rate is unknown for general graphons. A Bayesian approach using block priors proposed by Gao et al. (2015b) achieves the minimax error rate adaptively, but it still requires evaluation of the posterior likelihood over all possible block partitions to obtain the posterior mode or the expectation for the probability matrix.

Other recent efforts on graphon estimation focus on the case of monotone node degrees, which make the graphon identifiable. The sort-and-smooth methods of Yang et al. (2014) and Chan & Airoldi (2014) estimate the graphon under this assumption by first sorting nodes by their degrees and then smoothing the matrix A locally to estimate edge probabilities. The monotone degree assumption is crucial for the success of these methods, and, as we later show, the sort-and-smooth methods perform poorly when it does not hold. Finally, general matrix denoising methods can be applied to this problem if one considers A to be a noisy version of its expectation P; a good general representative of this class of methods is the universal singular-value thresholding approach of Chatterjee (2015). Since this is a general method, we cannot expect its error rate to be especially competitive for this specific problem, and indeed its mean squared error rate is slower than the cubic root of the minimax rate.

In this paper, we propose a novel computationally efficient method for edge probability matrix estimation based on neighbourhood smoothing, for piecewise-Lipschitz graphon functions. The key to this method is adaptive neighbourhood selection, which allows us to avoid making strong assumptions about the graphon. A node's neighbourhood consists of nodes with similar rows in the adjacency matrix, which intuitively correspond to nodes with similar values of the latent node positions ξ_i. To the best of our knowledge, our estimator achieves the best error rate among existing computationally feasible methods; it allows easy parallelization. The size of the neighbourhood is controlled by a tuning parameter, similar to bandwidth in nonparametric regression; the rate of this bandwidth parameter is determined by theory, and we show empirically that the method is robust with respect to the choice of the constant. Experiments on synthetic networks demonstrate that our method performs very well under a wide range of graphon models, including those of low rank and full rank, with and without monotone degrees. We also test its performance on the link prediction problem, using both synthetic and real networks.

2. The neighbourhood smoothing estimator and its error rate

2·1. *Neighbourhood smoothing for edge probability estimation*

Our goal is to estimate the probabilities P_{ij} from the observed network adjacency matrix A, where each A_{ij} is independently drawn from $\text{Ber}(P_{ij})$. While $P_{ij} = f(\xi_i, \xi_j)$, where the ξ_i are latent, our goal is to estimate P for the single realization of the ξ_i that gave rise to the data, rather than the function f. We think of f as a fixed unknown smooth function on $[0, 1]^2$, with formal smoothness assumptions to be stated later. Let $e_{ij} = e_{ij}(P_{ij})$ denote the Bernoulli error and omit its dependence on P. We can then write

$$A_{ij} = P_{ij} + e_{ij} = f(\xi_i, \xi_j) + e_{ij}. \qquad (2)$$

Formulation (2) resembles a nonparametric regression problem, except that the ξ_i are not observed. This has important consequences: for example, assuming further smoothness in f beyond order one does not improve the minimax error rate when estimating P (Gao et al., 2015a). Our approach is to apply neighbourhood smoothing, which would be natural had the latent variables ξ_i been observed. Intuitively, if we had a set $\mathcal{N}_i$ of neighbours of a node i, in the sense that $\mathcal{N}_i = \{i' : P_{i'\cdot} \approx P_{i\cdot}\}$, where $P_{i\cdot}$ represents the ith row of P, then we could estimate $P_{i\cdot}$ by averaging $A_{i'\cdot}$ over $i' \in \mathcal{N}_i$. Postponing the question of how to select $\mathcal{N}_i$ until § 2·2, we define a general neighbourhood smoothing estimator by

$$\tilde{P}_{ij} = \frac{\sum_{i' \in \mathcal{N}_i} A_{i'j}}{|\mathcal{N}_i|}. \qquad (3)$$

When the network is symmetric, we instead use a symmetric estimator

$$\hat{P} = \big(\tilde{P} + \tilde{P}^{\mathrm{T}}\big)/2. \qquad (4)$$

For simplicity, we focus on undirected networks. A natural alternative is to average over $\mathcal{N}_i \times \mathcal{N}_j$, but (3) and (4) allow vectorization and are thus more computationally efficient. Our estimator can also be viewed as a relaxation of step-function approximations such as those given in Olhede & Wolfe (2014). In step-function approximations, the neighbourhood for each node comprises the nodes from its block, so the neighbourhoods for two nodes from the same block are very similar, and the blocks have to be estimated first; in contrast, neighbourhood smoothing provides for more flexible neighbourhoods that differ from node to node, as well as an efficient way to select the neighbourhood, which we will discuss next.

2·2. *Neighbourhood selection*

Selecting the neighbourhood $\mathcal{N}_i$ in (4) is the core of our method. Since we estimate $P_{i\cdot}$ by averaging over $A_{i'\cdot}$ for $i' \in \mathcal{N}_i$, good neighbourhood candidates i' should have $f(\xi_{i'}, \cdot)$ close to $f(\xi_i, \cdot)$, which implies $P_{i'\cdot}$ close to $P_{i\cdot}$. We use the ℓ_2 distance between graphon slices to quantify this, defining

$$d(i,i') = \|f(\xi_i,\cdot) - f(\xi_{i'},\cdot)\|_2 = \left\{\int_0^1 |f(\xi_i,v) - f(\xi_{i'},v)|^2 \,\mathrm{d}v\right\}^{1/2}.$$

While one may consider more general ℓ_p or other distances, the ℓ_2 distance is particularly easy to work with theoretically. For the purpose of neighbourhood selection, it is not necessary to estimate $d(i,i')$; it suffices to provide a tractable upper bound. For integrable functions g_1 and g_2 defined on $[0,1]$, define $\langle g_1, g_2\rangle = \int_0^1 g_1(u) g_2(u)\,\mathrm{d}u$. Then we can write

$$d^2(i,i') = \langle f(\xi_i,\cdot), f(\xi_i,\cdot)\rangle + \langle f(\xi_{i'},\cdot), f(\xi_{i'},\cdot)\rangle - 2\langle f(\xi_i,\cdot), f(\xi_{i'},\cdot)\rangle. \qquad (5)$$

The third term in (5) can be estimated by $2\langle A_{i\cdot}, A_{i'\cdot}\rangle/n$, where $A_{i\cdot}$ and $A_{i'\cdot}$ are nearly independent up to a single duplicated entry, owing to symmetry. The first two terms in (5) are more difficult, since $\langle A_{i\cdot}, A_{i\cdot}\rangle/n$ is not a good estimator for $\langle f(\xi_i,\cdot), f(\xi_i,\cdot)\rangle$. Here we present the intuition and provide a full theoretical justification in Theorem 1. For simplicity, assume for now that f is Lipschitz with a Lipschitz constant of 1. The idea is to use nodes with graphon slices similar to i and i' to make the terms in the inner product distinct graphon slices. With high probability, for each i, we can find $\tilde{i} \neq i$ such that $|\xi_{\tilde{i}} - \xi_i| \leqslant e_n$, where the sequence e_n is a function of

n and represents the error rate to be specified later. Then $\|f(\xi_i,\cdot)-f(\xi_{\tilde{i}},\cdot)\|_2 \leqslant e_n$, and we can approximate $\langle f(\xi_i,\cdot), f(\xi_i,\cdot)\rangle$ by $\langle f(\xi_i,\cdot), f(\xi_{\tilde{i}},\cdot)\rangle$, where the latter can now be estimated by $\langle A_{i\cdot}, A_{\tilde{i}\cdot}\rangle/n$. The same technique can be used to approximate the second term in (5), but all these approximations depend on the unknown ξ. To deal with this, we rearrange the terms in (5) as follows:

$$\begin{aligned} d^2(i,i') &= \langle f(\xi_i,\cdot)-f(\xi_{i'},\cdot), f(\xi_i,\cdot)\rangle - \langle f(\xi_i,\cdot)-f(\xi_{i'},\cdot), f(\xi_{i'},\cdot)\rangle \\ &\leqslant \left|\langle f(\xi_i,\cdot)-f(\xi_{i'},\cdot), f(\xi_{\tilde{i}},\cdot)\rangle\right| + \left|\langle f(\xi_i,\cdot)-f(\xi_{i'},\cdot), f(\xi_{\tilde{i}'},\cdot)\rangle\right| + 2e_n \\ &\leqslant 2\max_{k\neq i,i'} |\langle f(\xi_i,\cdot)-f(\xi_{i'},\cdot), f(\xi_k,\cdot)\rangle| + 2e_n. \end{aligned} \tag{6}$$

The inner product on the right-hand side of (6) can be estimated by

$$\tilde{d}^2(i,i') = \max_{k\neq i,i'} |\langle A_{i\cdot}-A_{i'\cdot}, A_{k\cdot}\rangle| \,/n. \tag{7}$$

Intuitively, the neighbourhood $\mathcal{N}_i$ should consist of i' with small $\tilde{d}(i,i')$. To formalize this, let $q_i(h)$ denote the hth sample quantile of the set $\{\tilde{d}(i,i') : i' \neq i\}$, where h is a tuning parameter, and set

$$\mathcal{N}_i = \{i' \neq i : \tilde{d}(i,i') \leqslant q_i(h)\}, \tag{8}$$

where for notational simplicity we suppress the dependence of $\mathcal{N}_i$ on h. Thresholding at a quantile rather than at some absolute value is convenient since real networks vary in their average node degrees and other parameters, which leads to very different values and distributions of $\tilde{d}$. Empirically, thresholding at a quantile shows significant advantages in stability and performance compared with an absolute threshold. The choice of h will be guided by both the theory in § 2·3, which suggests the order of h, and empirical performance, which suggests the constant factor. More details are included in the Supplementary Material.

An important feature of this definition is that the neighbourhood admits nodes with similar graphon slices, but not necessarily similar ξ. For example, in the stochastic block model, all nodes from the same block would be equally likely to be included in each other's neighbourhoods, regardless of their ξ. Even though we use ξ_i and $\xi_{i'}$ to motivate (6), we always work with the function values $f(\xi_i,\xi_j)$ and never attempt to estimate the ξ_i or f by themselves. This contrasts with the approaches of Chan & Airoldi (2014) and Yang et al. (2014), and gives us a substantial computational advantage as well as much more flexibility in assumptions.

2·3. *Consistency of the neighbourhood smoothing estimator*

We study the theoretical properties of our estimator for a family of piecewise-Lipschitz graphon functions, defined as follows.

Definition 2 (Piecewise-Lipschitz graphon family). *For any $\delta, L > 0$, let $\mathcal{F}_{\delta;L}$ denote a family of piecewise-Lipschitz graphon functions $f : [0,1]^2 \to [0,1]$ such that* (i) *there exists an integer $K \geqslant 1$ and a sequence $0 = x_0 < \cdots < x_K = 1$ satisfying $\min_{0\leqslant s\leqslant K-1}(x_{s+1}-x_s) \geqslant \delta$, and* (ii) *both $|f(u_1,v)-f(u_2,v)| \leqslant L|u_1-u_2|$ and $|f(u,v_1)-f(u,v_2)| \leqslant L|v_1-v_2|$ hold for all $u, u_1, u_2 \in [x_s, x_{s+1}]$, $v, v_1, v_2 \in [x_t, x_{t+1}]$ and $0 \leqslant s, t \leqslant K-1$.*

For any $P, Q \in \mathbb{R}^{m\times m}$, define $d_{2,\infty}$, the normalized $(2,\infty)$ matrix norm, by

$$d_{2,\infty}(P,Q) = m^{-1/2}\|P-Q\|_{2,\infty} = \max_i m^{-1/2}\|P_{i\cdot} - Q_{i\cdot}\|_2.$$

Then we have the following error rate bound.

Theorem 1. *Assume that L is a global constant and $\delta = \delta(n)$ depends on n, satisfying $\lim_{n\to\infty} \delta/(n^{-1}\log n)^{1/2} \to \infty$. Then the estimator $\tilde{P}$ defined in (4), with neighbourhood $\mathcal{N}_i$ defined in (8) and $h = C(n^{-1}\log n)^{1/2}$ for any global constant $C \in (0,1]$, satisfies*

$$\max_{f\in\mathcal{F}_{\delta;L}} \mathrm{pr}\left\{ d_{2,\infty}(\tilde{P},P)^2 \geqslant C_1\left(\frac{\log n}{n}\right)^{1/2}\right\} \leqslant n^{-C_2}, \tag{9}$$

where C_1 and C_2 are positive global constants.

Since for any $P, Q \in \mathbb{R}^{m\times m}$ we have $d_{2,\infty}(P,Q) \geqslant m^{-1}\|P-Q\|_{\mathrm{F}}$, Theorem 1 yields the following corollary.

Corollary 1. *Under the conditions of Theorem 1,*

$$\max_{f\in\mathcal{F}_{\delta;L}} \mathrm{pr}\left\{ \frac{1}{n^2}\|\tilde{P}-P\|_F^2 \geqslant C_1\left(\frac{\log n}{n}\right)^{1/2}\right\} \leqslant n^{-C_2}. \tag{10}$$

The bound (10) continues to hold if we replace $\tilde{P}$ by $\hat{P}$, but (9) may not hold. Next, we show that under the $(2,\infty)$ norm, our estimator $\tilde{P}$ is nearly rate-optimal, up to a $\log n$ factor.

Theorem 2. *Under the conditions of Theorem 1, we have*

$$\inf_{\hat{P}} \sup_{f\in\mathcal{F}_{\delta;L}} E\{d_{2,\infty}^2(\hat{P},P)\} \geqslant C(n\log n)^{-1/2}$$

for some global constant $C > 0$.

To the best of our knowledge, the result (9) is the only $(2,\infty)$ error rate available for polynomial-time graphon estimation methods. Most previous work has focused on the mean squared error and only considered the special case $\delta = 1$. For $\delta = 1$, the minimax error rate $\log n/n$ established by Gao et al. (2015a) has so far only been achieved by methods that require combinatorial optimization or evaluation, as in, for example, Gao et al. (2015a) and Klopp et al. (2017). The rate $(\log n/n)^{1/2}$ was previously achieved by combinatorial methods, as in, for example, Wolfe & Olhede (2013) and Olhede & Wolfe (2014). Among computationally efficient methods, singular-value thresholding (Chatterjee 2015, Theorem 2.7) achieves $n^{-1/3}$. Additionally, the sort-and-smooth method proposed by Chan & Airoldi (2014) achieves the minimax error rate under the strong assumption that f has strictly monotone expected node degrees $d_f(v) = \int_0^1 f(u,v)\,\mathrm{d}u$. A referee sent us a proof that thresholding the leading k singular values of the matrix A achieves the mean squared error of $k/n + k^{-2}$, where the variance k/n is due to Candès & Plan (2011) and k^{-2} is the bias. Taking $k = n^{1/3}$ gives the best known mean squared error rate of $n^{-2/3}$ for a computationally efficient algorithm. For the graphon family $f \in \mathcal{F}_{\delta;L}$ where $\delta/(n^{-1}\log n)^{1/2} \to \infty$ that we study, the $n^{-1/3}$ singular-value thresholding method and our method achieve the same mean squared error rate.

Table 1. *Synthetic graphons*

Graphon	Function $f(u, v)$	Monotone degrees	Rank	Local structure
1	$k/(K+1)$ if $u, v \in ((k-1)/K, k/K)$, $0{\cdot}3/(K+1)$ otherwise; $K = \lfloor \log n \rfloor$	Yes	$\lfloor \log n \rfloor$	No
2	$\sin\{5\pi(u+v-1)+1\}/2 + 0{\cdot}5$	No	3	No
3	$1 - [1 + \exp\{15(0{\cdot}8\lvert u-v\rvert)^{4/5} - 0{\cdot}1\}]^{-1}$	No	Full	No
4	$(u^2+v^2)/3 \cos\{1/(u^2+v^2)\} + 0{\cdot}15$	No	Full	Yes

For the case of general δ, we can show that the minimax rate of $\log n/n$ established by Gao et al. (2015a) still holds for the family $\mathcal{F}_{\delta;L}$, in Proposition 1; see the Supplementary Material.

Proposition 1. *Under the conditions of Theorem 1, when $\delta/(\log n/n)^{1/2} \to \infty$, there exists a global constant $C_3 > 0$ such that*

$$\inf_{\tilde{P}} \max_{f \in \mathcal{F}_{\delta;L}} E\left\{ \frac{1}{n^2} \|\hat{P} - P\|_{\mathrm{F}}^2 \right\} \asymp \frac{\log n}{n}.$$

Whether this minimax error rate can be achieved by a computationally efficient method remains an open question.

3. Probability matrix estimation on synthetic networks

In this section, we evaluate the performance of our symmetric estimator (4) in estimating the probability matrix for synthetic networks. We generate the networks from the four graphons listed in Table 1, selected to have different features in different combinations, monotone degrees, low rank, etc. The corresponding probability matrices are pictured in the first column of Fig. 1 in the lower triangular half. All networks have $n = 2000$ nodes.

Additional empirical results in the Supplementary Material show that our method is robust with respect to the choice of the constant factor C in the bandwidth h; for simplicity, we set $C = 1$ for the rest of this paper. Here we focus on comparison with benchmarks. From the general matrix denoising methods, we include the widely used method of universal singular-value thresholding (Chatterjee, 2015) and the $n^{1/3}$ leading-singular-value thresholding method suggested by a referee. We also compare with the sort-and-smooth methods of Chan & Airoldi (2014) and Yang et al. (2014). These methods differ only in that the latter employs singular-value thresholding to denoise the network as a pre-processing step. Owing to space constraints, we present both methods in Table 2 but only Chan & Airoldi (2014) in the figures, since they are visually very similar.

We also include two approximations based on fitting a stochastic block model, called network histograms by Olhede & Wolfe (2014). One is the oracle stochastic block model, where the blocks are based on the true values of the latent ξ_i. This cannot be done in practice, but we use it as the gold standard for a step-function approximation. The feasible version of this is an approximation based on a stochastic block model with estimated blocks; we fit it by regularized spectral clustering (Chaudhuri et al., 2012). Any other algorithm for fitting the stochastic block model can be used to estimate the blocks; for example, Olhede & Wolfe (2014) used a local updating algorithm initialized with spectral clustering to compute their network histograms. Here we have chosen regularized spectral clustering because of its speed and good empirical performance. For both approximations, we set the number of blocks to $n^{1/2}$, as in Olhede & Wolfe (2014).

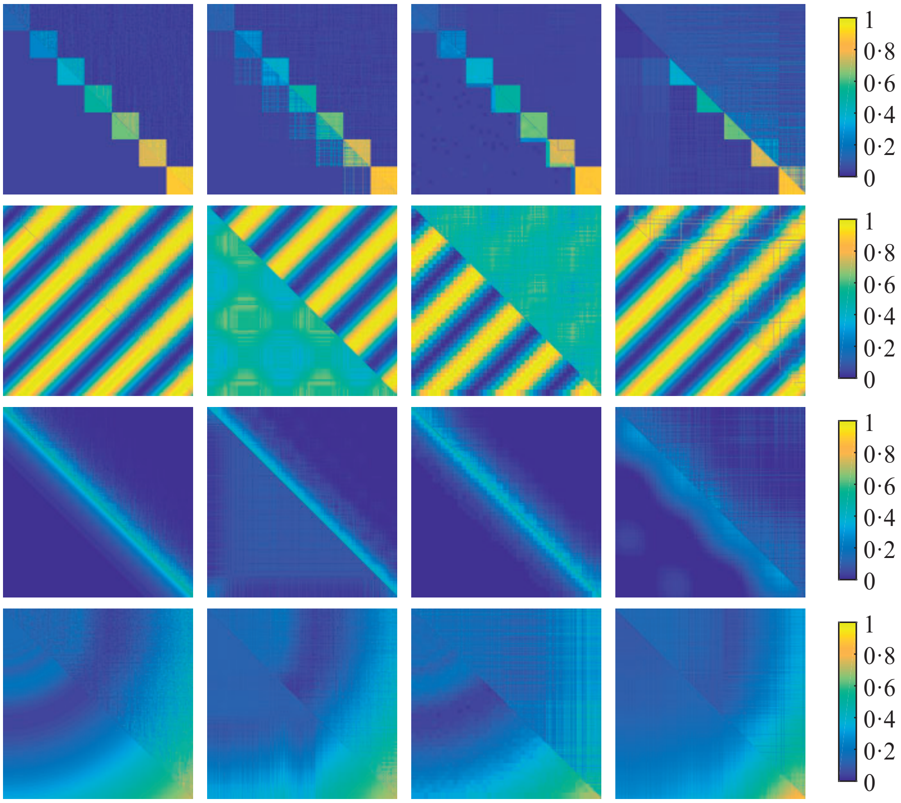

Fig. 1. Estimated probability matrices for graphons 1–4, shown in rows 1–4. Column 1: true P (lower) and our method (upper). Column 2: Chan & Airoldi (2014) (lower) and $n^{1/3}$ singular-value thresholding (upper). Column 3: Block model oracle (lower) and spectral clustering (upper). Column 4: Chatterjee (2015) (lower) and Airoldi's method (upper).

A recent as yet unpublished method kindly shared with us by E. Airoldi proposes a stochastic block model approximation, adapting the method of Airoldi et al. (2013) to work with a single adjacency matrix. It uses a dissimilarity measure $\sum_{k \neq i,i'} |\langle A_{i\cdot} - A_{k\cdot}, A_{k\cdot}\rangle|$, which we considered before choosing (7) because it leads to a better guaranteed error rate. Airoldi's method then builds blocks by starting with one not-yet-clustered node i and including all nodes whose dissimilarity from i is below a threshold Δ as neighbours. We found that our strategy of thresholding by quantile instead of a fixed threshold is more efficient and stable, and the theoretical error rate is better for our method.

We present heat maps of results for a single realization in Fig. 1, and the root mean squared errors and the mean absolute errors of $\hat{P}$ in Table 2. While these two errors mostly agree on method ranking, the few cases where they disagree indicate whether the errors come primarily from a small number of poorly estimated entries or are more uniformly distributed throughout the matrix.

Graphon 1 has $K = \lfloor \log n \rfloor = 7$ blocks with different within-block edge probabilities, which all dominate the low between-block probability. The best results are obtained by our method, singular-value thresholding, spectral clustering and the oracle stochastic block model approximation, which is expected given that the data are generated from a stochastic block model. The oracle uses $n^{1/2}$ blocks rather than the true K, and thus makes substantial errors on the block boundaries, but not anywhere else. The method of Chan & Airoldi (2014) correctly estimates

Table 2. *Root mean squared errors and mean absolute errors with standard errors, all multiplied by* 10^2*, averaged over* 2000 *replications. The largest relative error is less than* 4%

	Graphon 1		Graphon 2		Graphon 3		Graphon 4	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Our method	1·92	1·33	3·06	2·25	3·00	1·41	3·55	2·76
Chan & Airoldi (2014)	8·78	3·09	34·17	30·16	11·27	8·04	4·46	3·58
Yang et al. (2014)	9·56	4·14	34·18	30·19	11·47	8·59	5·67	4·87
$n^{1/3}$ singular-value	2·99	2·25	4·74	3·59	3·16	1·79	5·86	4·33
Block model spectral	1·72	0·75	33·06	28·80	3·98	1·78	9·08	6·64
Block model oracle	5·48	1·42	5·11	3·80	1·62	0·75	1·06	0·83
Chatterjee (2015)	4·09	2·25	1·89	1·47	6·39	3·81	5·67	4·87
Airoldi's method	15·94	8·92	15·82	9·23	9·40	5·74	4·60	3·16

RMSE, root mean squared error; MAE, mean absolute error.

the main blocks because they have different expected degrees, but suffers from boundary effects due to smoothing over the entire region. In contrast, our method, which determines smoothing neighbourhoods based on similarities of graphon slices, does not suffer from boundary effects. Chatterjee (2015) does a good job on denser blocks but thresholds away sparser blocks. Airoldi's method captures tightly connected communities, but does not do as well on weaker communities.

Graphon 2 lacks node-degree monotonicity, and thus the method of Chan & Airoldi (2014) does not work here. Spectral clustering also performs poorly, probably because it uses too many $n^{1/2}$ eigenvectors that add noise. Airoldi's method and the stochastic block model oracle give grainy but reasonable approximations to P, and the best results are obtained by our method, that of Chatterjee (2015) and singular-value thresholding with $n^{1/3}$ eigenvalues. The latter two are expected to work well, since this is a low-rank matrix.

Graphon 3 is a diagonal-dominated matrix, and our method is the best among computationally efficient methods. The method of Chatterjee (2015) does not perform well, because this is not a low-rank matrix; spectral clustering, on the other hand, does fine, because there are many nonzero eigenvalues and the $n^{1/2}$ eigenvectors contain enough information. The $n^{1/3}$ singular-value thresholding does better than Chatterjee (2015) and provides a lower-resolution denoising. Airoldi's method shows the structure only roughly, probably because of the similarity measure that it uses. The method of Chan & Airoldi (2014) fails since all node expected degrees are almost the same.

Graphon 4 is difficult to estimate for all methods. It is of full rank, with structure at different scales. This makes it a difficult setting for low-rank approximations, among which the $n^{1/3}$ singular-value thresholding alone uses enough eigenvalues to produce a reasonable result, albeit with boundary effects. This graphon is not a block matrix, and thus spectral clustering does not perform well. The expected node degrees are not the same, but their ordering does not match the ordering of the latent node positions, so this graphon is also difficult for the sort-and-smooth method of Chan & Airoldi (2014). Our method successfully picks up the global structure and the curvature. While visually it is fairly similar to the result of $n^{1/3}$ singular-value thresholding, our method has significantly better errors in Table 2. Overall, this example illustrates a limitation of all global methods when there are subtle local differences.

Table 2 shows the mean squared errors and the mean absolute errors of all methods on the four graphons averaged over 2000 replications. The results generally agree with those shown in the figures. The few relative discrepancies between the root mean squared errors and ℓ_1 errors occur when there are a small number of large errors, such as the boundary effects for the oracle for graphon 1, which affect the root mean squared errors more than the ℓ_1 error.

For graphon 1, our method and spectral clustering perform best. For graphon 2, our method is only outperformed by universal singular-value thresholding, whereas $n^{1/3}$ leading-eigenvalue thresholding selects fewer eigenvalues than needed. For graphon 3, our method is comparable to $n^{1/3}$ leading-eigenvalue thresholding, and they are both better than the other methods, not counting the oracle. For graphon 4, our method shows a significant advantage over all other methods except for the oracle. Thus, in all cases, our method shows very competitive performance compared with benchmarks.

Overall, the results in this section show that various previously proposed methods can perform very well under their respective assumptions, which may be monotone degrees or low rank or an underlying block model, but they fail when these assumptions are not satisfied. Our method is the only one among those compared that performs well in a large range of scenarios, because it learns the structure from data via neighbourhood selection instead of imposing a priori structural assumptions. The $n^{1/3}$ singular-value thresholding method also shows consistent performance across all graphons, although in all cases somewhat worse than ours. It performs very well in the low-rank case, but if the leading singular values decay slowly, our method performs better.

4. APPLICATION TO LINK PREDICTION

Direct evaluation of probability matrix estimation on real networks is difficult, since the true probability matrix is unknown. We assess the practical utility of our method by applying it to link prediction, a task that relies on estimating the probability matrix. Here we think of the true adjacency matrix A^{true} as unobserved, with binary edges drawn independently with probabilities given by P, also unobserved. Instead, we observe $A_{ij}^{\text{obs}} = M_{ij}A_{ij}^{\text{true}}$, where the unobserved M_{ij} are independent $\text{Ber}(1-p)$, and p is unknown. Therefore $A_{ij}^{\text{obs}} = 1$ is always a true edge, but $A_{ij}^{\text{obs}} = 0$ could be either a true zero or a false negative. This setting is different from and perhaps more realistic than the link prediction setting in Gao et al. (2016), who assumed that M_{ij} are observed. Under their setting, the missing rate p can be estimated by the empirical missing rate $\hat{p}$, and all estimators can be corrected for missingness simply by dividing them by $1-\hat{p}$.

A link prediction method usually outputs a nonnegative score matrix $\hat{A}$, with scores giving the estimated propensity of a node pair to form an edge. For methods that estimate the probability matrix, $\hat{A}$ can be taken to be $\hat{P}$; other link prediction methods construct a binary $\hat{A}$ by working directly on A. Both types of method essentially output a ranked list of most likely missing links, useful in practice for follow-up confirmatory analysis.

We compare various link prediction methods via their receiver operating characteristic curves. For each $t > 0$, we define the false-positive and true-positive rates by

$$r_{\text{FP}}(t) = \sum_{ij} 1\big(\hat{A}_{ij} > t, A_{ij}^{\text{true}} = 0, M_{ij} = 0\big) \Big/ \sum_{ij} 1\big(A_{ij}^{\text{true}} = 0, M_{ij} = 0\big),$$

$$r_{\text{TP}}(t) = \sum_{ij} 1\big(\hat{A}_{ij} > t, A_{ij}^{\text{true}} = 1, M_{ij} = 0\big) \Big/ \sum_{ij} 1\big(\hat{A}_{ij} = 1, M_{ij} = 0\big).$$

Then, by varying t, we obtain the receiver operating characteristic curve. In practice, t is often selected to output a fixed number of most likely links.

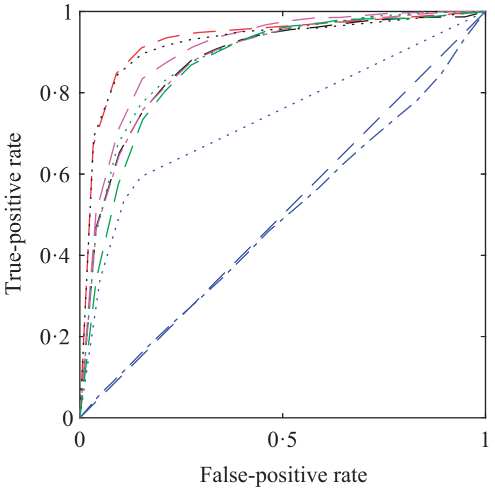

Fig. 2. Receiver operating characteristic curves for link prediction on the political blogs network; 10% of edges are missing at random. The red dashed curve is for our method, the black dotted curve for $n^{1/3}$ singular-value thresholding, the blue dashed curve for the method of Zhao et al. (2017), the blue dash-dotted curve for the Jaccard index, the blue dotted curve for PropFlow, the black dashed curve for the method of Chatterjee (2015), the magenta dashed curve for the method of Chan & Airoldi (2014), the magenta dash-dotted curve for the method of Yang et al. (2014), and the green dashed curve for the block model with spectral clustering.

In this section, we include three additional benchmark methods that produce score matrices rather than estimated probability matrices. One standard score is the Jaccard index $\langle A_{i\cdot}, A_{j\cdot}\rangle \big/ \{(\sum_k A_{ik})(\sum_k A_{jk})\}$; see for example Lichtenwalter et al. (2010). The method of Zhao et al. (2017) computes scores so that similar node pairs have similar predicted scores. The PropFlow algorithm of Lichtenwalter et al. (2010) uses an expected random walk distance between nodes as the score.

We first compare all methods on simulated networks generated from the graphons in Table 1. We set $n = 500$ because of the computational cost of some of the benchmarks and we set $p = 10\%$. All experiments are repeated 1000 times. Figure 2 in the Supplementary Material shows the receiver operating characteristic curves for four graphons. Most differences between the methods can be inferred from Fig. 1. Overall, the methods based on graphon estimation outperform score-based methods. Our method outperforms all other methods on this task, producing a receiver operating characteristic curve very close to that based on the true probability matrix P.

We have also applied our method and others to the political blogs network (Adamic & Glance, 2005). This network consists of 1222 manually labelled blogs: 586 liberal and 636 conservative. The network clearly shows two communities, with heterogeneous node degrees, i.e., there are hubs. We removed 10% of edges at random and calculated the receiver operating characteristic curve for predicting the missing links, shown in Fig. 2. Again, methods based on estimating the probability matrix performed much better than the scoring methods, and our method performs best overall. Sort-and-smooth methods slightly outperformed spectral clustering and the method of Chatterjee (2015), perhaps because of the presence of hubs.

5. Discussion

The strength of our method is the adaptive neighbourhood choice, which works well under many different conditions; it is also computationally efficient, easy to implement and essentially tuning-free. Its main limitation is the piecewise-Lipschitz condition, which occasionally leads to oversmoothing. Our method does not achieve the minimax error rate, and its rate cannot be improved; whether the minimax rate can be achieved by any polynomial-time method is, to the best of our knowledge, an open problem. Another major future challenge is relaxing the assumption of independent edges to allow better fit to real-world networks.

Acknowledgement

The authors thank an associate editor and two referees for very helpful suggestions, and E. M. Airoldi of Harvard University for sharing a copy of his manuscript and code and for great comments. Levina was supported by the National Science Foundation and the Office of Naval Research. Zhu was supported by the National Science Foundation and the Key Laboratory for Applied Statistics, Ministry of Education, China.

Supplementary material

Supplementary material available at *Biometrika* online includes numerical results on the bandwidth constant in Theorem 1, $(2,\infty)$-norm errors and comparisons with benchmarks on link prediction for synthetic graphons from § 3, and the proofs of Theorems 1 and 2 and Proposition 1.

References

Adamic, L. A. & Glance, N. (2005). The political blogosphere and the 2004 U.S. election: Divided they blog. In *Proc. 3rd Int. Workshop on Link Discovery*, LinkKDD '05. New York: ACM, pp. 36–43.

Airoldi, E. M., Blei, D. M., Fienberg, S. E. & Xing, E. P. (2008). Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.* **9**, 1981–2014.

Airoldi, E. M., Costa, T. B. & Chan, S. H. (2013). Stochastic blockmodel approximation of a graphon: Theory and consistent estimation. In *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani & K. Q. Weinberger, eds. Red Hook, New York: Curran Associates, pp. 692–700.

Aldous, D. J. (1981). Representations for partially exchangeable arrays of random variables. *J. Mult. Anal.* **11**, 581–98.

Amini, A. A., Chen, A., Bickel, P. J. & Levina, E. (2013). Pseudo-likelihood methods for community detection in large sparse networks. *Ann. Statist.* **41**, 2097–22.

Amini, A. A. & Levina, E. (2016). On semidefinite relaxations for the block model. *arXiv*: 1406.5647v3.

Bickel, P. J. & Chen, A. (2009). A nonparametric view of network models and Newman–Girvan and other modularities. *Proc. Nat. Acad. Sci.* **106**, 21068–73.

Cai, D., Ackerman, N. & Freer, C. (2015). An iterative step-function estimator for graphons. *arXiv*: 1412.2129v2.

Cai, T. T. & Li, X. (2015). Robust and computationally feasible community detection in the presence of arbitrary outlier nodes. *Ann. Statist.* **43**, 1027–59.

Candès, E. J. & Plan, Y. (2011). Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements. *IEEE Trans. Info. Theory* **57**, 2342–59.

Chan, S. H. & Airoldi, E. (2014). A consistent histogram estimator for exchangeable graph models. *J. Mach. Learn. Res. Workshop Conf. Proc.* **32**, 208–16.

Chatterjee, S. (2015). Matrix estimation by universal singular value thresholding. *Ann. Statist.* **43**, 177–214.

Chaudhuri, K., Chung, F. & Tsiatas, A. (2012). Spectral clustering of graphs with general degrees in the extended planted partition model. *J. Mach. Learn. Res.* **2012**, 1–23.

Choi, D. (2017). Co-clustering of nonsmooth graphons. *Ann. Statist.* **45**, 1488–515.

Choi, D. & Wolfe, P. J. (2014). Co-clustering separately exchangeable network data. *Ann. Statist.* **42**, 29–63.

DIACONIS, P. & JANSON, S. (2007). Graph limits and exchangeable random graphs. *arXiv:* 0712.2749.

GAO, C., LU, Y., MA, Z. & ZHOU, H. H. (2016). Optimal estimation and completion of matrices with biclustering structures. *J. Mach. Learn. Res.* **17**, 1–29.

GAO, C., LU, Y. & ZHOU, H. H. (2015a). Rate-optimal graphon estimation. *Ann. Statist.* **43**, 2624–52.

GAO, C., VAN DER VAART, A. W. & ZHOU, H. H. (2015b). A general framework for Bayes structured linear models. *arXiv:* 1506.02174.

GUÉDON, O. & VERSHYNIN, R. (2016). Community detection in sparse networks via Grothendieck's inequality. *Prob. Theory Rel. Fields* **165**, 1025–49.

HOFF, P. (2008). Modeling homophily and stochastic equivalence in symmetric relational data. *Adv. Neural Info. Proc. Syst.* **20**, 657–64.

KARRER, B. & NEWMAN, M. E. (2011). Stochastic blockmodels and community structure in networks. *Phys. Rev.* E **83**, 016107.

KLOPP, O., TSYBAKOV, A. B. & VERZELEN, N. (2017). Oracle inequalities for network models and sparse graphon estimation. *Ann. Statist.* **45**, 316–54.

LICHTENWALTER, R. N., LUSSIER, J. T. & CHAWLA, N. V. (2010). New perspectives and methods in link prediction. In *Proc. 16th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, KDD '10. New York: ACM, pp. 243–52.

OLHEDE, S. C. & WOLFE, P. J. (2014). Network histograms and universality of blockmodel approximation. *Proc. Nat. Acad. Sci.* **111**, 14722–7.

ROHE, K., CHATTERJEE, S. & YU, B. (2011). Spectral clustering and the high-dimensional stochastic blockmodel. *Ann. Statist.* **39**, 1878–915.

SAADE, A., KRZAKALA, F. & ZDEBOROVÁ, L. (2014). Spectral clustering of graphs with the Bethe Hessian. In *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence & K. Q. Weinberger, eds. Red Hook, New York: Curran Associates, pp. 406–14.

WOLFE, P. J. & OLHEDE, S. C. (2013). Nonparametric graphon estimation. *arXiv:* 1309.5936.

YANG, J. J., HAN, Q. & AIROLDI, E. M. (2014). Nonparametric estimation and testing of exchangeable graph models. *J. Mach. Learn. Res. Workshop Conf. Proc.* **33**, 1060–7.

ZHANG, Y., LEVINA, E. & ZHU, J. (2015). Detecting overlapping communities in networks using spectral methods. *arXiv:* 1412.3432v4.

ZHAO, Y., WU, Y.-J., LEVINA, E. & ZHU, J. (2017). Link prediction for partially observed networks. *J. Comp. Graph. Statist.,* doi:10.1080/10618600.2017.1286243.

[*Received on* 29 *September* 2015. *Editorial decision on* 21 *June* 2017]