Biometrika (2020), **107**, 2, *pp.* 257–276
Printed in Great Britain
doi: 10.1093/biomet/asaa006
Advance Access publication 4 April 2020

Network cross-validation by edge sampling

BY TIANXI LI

Department of Statistics, University of Virginia, B005 Halsey Hall, 148 Amphitheater Way, Charlottesville, Virginia 22904, U.S.A.
tianxili@virginia.edu

ELIZAVETA LEVINA AND JI ZHU

Department of Statistics, University of Michigan, 459 West Hall, 1085 South University Avenue, Ann Arbor, Michigan 48105, U.S.A.
elevina@umich.edu jizhu@umich.edu

SUMMARY

While many statistical models and methods are now available for network analysis, resampling of network data remains a challenging problem. Cross-validation is a useful general tool for model selection and parameter tuning, but it is not directly applicable to networks since splitting network nodes into groups requires deleting edges and destroys some of the network structure. In this paper we propose a new network resampling strategy, based on splitting node pairs rather than nodes, that is applicable to cross-validation for a wide range of network model selection tasks. We provide theoretical justification for our method in a general setting and examples of how the method can be used in specific network model selection and parameter tuning tasks. Numerical results on simulated networks and on a statisticians' citation network show that the proposed cross-validation approach works well for model selection.

Some key words: Cross-validation; Model selection; Parameter tuning; Random network.

1. INTRODUCTION

Statistical methods for analysing networks have received a great deal of attention because of their wide-ranging applications in areas such as sociology, physics, biology and the medical sciences. Statistical network models provide a principled approach to extracting salient information about the network structure while filtering out noise. Perhaps the simplest statistical network model is the famous Erdős–Rényi model (Erdős & Rényi, 1960), which served as a building block for a large body of more complex models, including the stochastic block model (Holland et al., 1983), the degree-corrected stochastic block model (Karrer & Newman, 2011), the mixed membership block model (Airoldi et al., 2008) and the latent space model (Hoff et al., 2002), to name a few.

While there has been plenty of work on models for networks and algorithms for fitting them, inference for these models is often lacking, making it hard to take advantage of the full power of statistical modelling. Data splitting methods provide a general, simple and relatively model-free inference framework and are commonly used in modern statistics, with cross-validation being the tool of choice for many model selection and parameter tuning tasks. For networks, both of these tasks are important; while there are plenty of models to choose from, it is a lot less clear how to select the best model for the data and how to choose tuning parameters for the selected model,

which is often necessary in order to fit it. In classical settings where the data points are assumed to be an independent and identically distributed sample, cross-validation works by splitting the data into multiple parts, or folds, holding out one fold at a time as a test set, fitting the model on the remaining folds and computing its error on the held-out fold, and finally averaging the errors across all folds to obtain the cross-validation error. The model or the tuning parameter is then chosen to minimize this error. To explain the challenge of applying this idea to networks, we first introduce a probabilistic framework.

Let $\mathcal{V} = \{1, 2, \ldots, n\} =: [n]$ denote the node set of a network, and let A be its $n \times n$ adjacency matrix, where $A_{ij} = 1$ if there is an edge from node i to node j and 0 otherwise. We view the elements of A as realizations of independent Bernoulli variables, with $E(A) = M$, where M is a matrix of probabilities. For undirected networks, $A_{ji} = A_{ij}$, so both A and M are symmetric matrices. We further assume that the unique edges A_{ij}, for $i < j$, are independent Bernoulli variables. The general network analysis task is to estimate M from the data A, under various structural assumptions we might make to address the difficulty of having a single realization of A.

To perform cross-validation on networks, one has to decide how to split the data contained in A and how to treat the resulting partial data which no longer form a complete network. To the best of our knowledge, there is little work available on this topic. Cross-validation was used by Hoff (2008) under a particular latent space model, and Chen & Lei (2018) proposed a novel cross-validation strategy for model selection under the stochastic block model and its variants. In this paper we do not assume a specific model for the network, but instead make a more general structural assumption of M being approximately of low rank, which holds for most of the popular network models. We propose a new general edge cross-validation strategy for networks, splitting node pairs rather than nodes into different folds, which is a natural yet crucial choice. Treating the network after removing the entries of A for some node pairs as a partially observed network, we apply low-rank matrix completion to complete the network and then fit the relevant model. This reconstructed network has the same rate of concentration around the true model as the full network adjacency matrix, allowing for valid analysis. Our method is valid for directed and undirected, binary, and weighted networks. As concrete examples, we show how edge cross-validation can be used to determine the latent space dimension of random dot product graph models, select between block model variants, tune regularization for spectral clustering, and tune neighbourhood smoothing for graphon models.

2. THE EDGE CROSS-VALIDATION ALGORITHM

2.1. *Notation and model*

For simplicity of presentation, we derive all our results for binary networks, but it will be clear that our framework is directly applicable to weighted networks, which are prevalent in practice, and in fact the application in § 5 is on a weighted network.

Recall that n is the number of nodes and A is the $n \times n$ adjacency matrix. Let $D = \mathrm{diag}(d_1, \ldots, d_n)$ be the diagonal matrix with the node degrees $d_i = \sum_j A_{ij}$ on the diagonal. The normalized Laplacian of a network is defined as $L = D^{-1/2} A D^{-1/2}$. Finally, we write I_n for the $n \times n$ identity matrix and $\mathbb{1}_n$ for the $n \times 1$ column vector of ones, suppressing the dependence on n when it is clear from the context. For any matrix M, $\|M\|$ will denote its spectral norm and $\|M\|_{\mathrm{F}}$ its Frobenius norm.

Throughout the paper, we work with the widely used inhomogeneous Erdős–Rényi model for networks, defined by an $n \times n$ matrix of probabilities, M, with unique edges A_{ij} drawn as independent Bernoulli variables such that $\mathrm{pr}(A_{ij} = 1) = M_{ij}$. All the information about the

structure of the network is thus contained in M. While the M_{ij} can all be different, with no additional assumptions on M inference is impossible, since we only have one observation. On the other hand, we would like to avoid assuming a specific parametric model, since choosing the type of model is one of the primary applications of cross-validation. As a compromise, we make a generic structural assumption on M, assuming that it is of low rank, which is true for many popular network models. We describe three classes of examples as follows.

Example 1 (The stochastic block model and its generalizations). The stochastic block model is perhaps the most widely used undirected network model with communities. The model assumes that $M = ZBZ^{\mathrm{T}}$ where $B \in [0,1]^{K\times K}$ is a symmetric probability matrix and $Z \in \{0,1\}^{n\times K}$ has exactly one 1 in each row, with $Z_{ik} = 1$ if node i belongs to community k. Let $c = (c_1, \ldots, c_n)$ be the vector of node membership labels, with c_i taking values in $\{1, \ldots, K\}$. In particular, it is assumed that $\mathrm{pr}(A_{ij} = 1) = B_{c_i c_j}$, that is, the probability of an edge between two nodes depends only on the communities they belong to. One of the commonly pointed-out limitations of the stochastic block model is that it forces equal expected degrees for all the nodes in the same community, thereby ruling out hubs. The degree-corrected stochastic block model corrects this by allowing individual degree parameters θ_i to be associated with each node i and assuming models $\mathrm{pr}(A_{ij} = 1) = \theta_i\theta_j B_{c_i c_j}$. The degree-corrected model needs a constraint to ensure identifiability, and here we use the constraint $\sum_{c_i=k} \theta_i = 1$ for each k, proposed in the original paper of Karrer & Newman (2011). The popular configuration model of Chung & Lu (2002) can be viewed as a special case of the degree-corrected model, and both these models have a probability matrix M of rank K. There are multiple other low-rank variants of the stochastic block model, such as the mixed membership block model (Airoldi et al., 2008) and the popularity-adjusted model (Sengupta & Chen, 2018). For a review of recent developments with regard to this class of model, see Abbe (2018).

Example 2 (The random dot product graph model). The random dot product graph model (Young & Scheinerman, 2007) is a general low-rank network model. It assumes that each node of the network is associated with a latent K-dimensional vector $Z_i \in \mathbb{R}^K$ and that $M_{ij} = Z_i^{\mathrm{T}} Z_j$. This model has been successfully applied to a number of network problems (Sussman et al., 2014; Tang et al., 2017), and its limiting behaviour has also been studied (Tang & Priebe, 2018). More details can be found in the review paper of Athreya et al. (2017). The random dot product graph model can include the stochastic block model as a special case, but only if the probability matrix M of the stochastic block model is positive semidefinite.

Example 3 (The latent space model and graphon models). The latent space model (Hoff et al., 2002) is another popular inhomogeneous Erdős–Rényi model. Like the random dot product graph, it assumes that the nodes correspond to n latent positions $Z_i \in \mathbb{R}^K$ and that the probability matrix is some function of the latent positions; examples include the distance model $f(M_{ij}) = \alpha - \|Z_i - Z_j\|$ and the projection model $f(M_{ij}) = \alpha - Z_i^{\mathrm{T}} Z_j/(\|Z_i\|\|Z_j\|)$, where f is a known function such as the logit function. More generally, the Aldous–Hoover representation (Aldous, 1981; Diaconis & Janson, 2007) says that the probability matrix of any exchangeable random graph can be written as $M_{ij} = f(\xi_i, \xi_j)$ for ξ_i $(i \in [n])$ independent uniform random variables on $[0,1]$ and a function $f : [0,1] \times [0,1] \to [0,1]$ that is symmetric in its two arguments, determined up to a measure-preserving transformation. There is a substantial literature on estimating the function f, called the graphon, under various assumptions (Wolfe & Olhede, 2013; Choi & Wolfe, 2014; Gao et al., 2015). Under this framework M is random, but the network follows an inhomogeneous Erdős–Rényi model conditionally on M, and so our method is applicable conditionally. The latent space model and the more general graphon models typically do not assume that M is of

low rank, instead enforcing certain smoothness conditions on the function f. Fortunately, when these smoothness assumptions hold, the corresponding matrix M can typically be approximated reasonably well by a low-rank matrix (Chatterjee, 2015; Zhang et al., 2017). In this setting, the edge cross-validation procedure works with the best low-rank approximation to the model; see the details in § 4.2.

2.2. *The edge cross-validation procedure*

For notational simplicity, we only present the algorithm for directed networks; the only modification needed for undirected networks is to treat node pairs (i,j) and (j,i) as one pair. The key idea of the edge cross-validation procedure is to split node pairs rather than nodes, resulting in a partially observed network. We randomly sample node pairs, regardless of the value of A_{ij}, with a fixed probability $1-p$ of being in the held-out set. By the exchangeable model assumption, the values of A corresponding to held-out node pairs are independent of those corresponding to the rest. The leftover training network now has missing edge values, which means that many models and methods cannot be applied to it directly. Our next step is to reconstruct a complete network $\hat{A}$ from the training node pairs. Fortunately, the missing entries are missing completely at random by construction, and this is the classic setting for matrix completion. Any low-rank-based matrix completion algorithm (e.g., Candès & Plan, 2010; Davenport et al., 2014) can now be used to fill in the missing entries. We defer the details of the algorithm to § 2.3.

Once we complete $\hat{A}$ through matrix completion, we can fit the candidate models on $\hat{A}$ and evaluate the relevant loss on the held-out entries of A, just as in standard cross-validation. There may be more than one way to evaluate the loss on the held-out set if the loss function itself is designed for binary input; we will elaborate on this issue in the examples in § 3. The general algorithm is summarized as Algorithm 1. We present the version with many random splits into training and test pairs, but the same procedure is obviously applicable to K-fold cross-validation if the computational cost of many random splits is prohibitive.

Algorithm 1. The general edge cross-validation procedure.

Input: an adjacency matrix A, a loss function L, a set $\mathcal{C}$ of Q candidate models or tuning parameter values to select from, the training proportion p, and the number of replications N.

Step 1. Select the rank $\hat{K}$ for matrix completion, either from prior knowledge or using the model-free cross-validation procedure in § 3.1.

Step 2. For $m = 1, \ldots, N$:

(a) Randomly choose a subset of node pairs $\Omega \subset \mathcal{V} \times \mathcal{V}$ by selecting each pair independently with probability p.

(b) Apply a low-rank matrix completion algorithm to (A, Ω) to obtain $\hat{A}$ with rank $\hat{K}$.

(c) For each of the candidate models $q = 1, \ldots, Q$, fit the model on $\hat{A}$ and evaluate its loss $L_q^{(m)}$ by averaging the loss function L with the estimated parameters over the held-out set $A_{ij}, (i,j) \in \Omega^{\mathrm{c}}$.

Step 3. Let $L_q = \sum_{m=1}^{N} L_q^{(m)}/N$ and return $\hat{q} = \arg\min_q L_q$, the best model from set $\mathcal{C}$.

The two crucial parts of edge cross-validation are splitting node pairs at random and applying low-rank matrix completion to obtain a full matrix $\hat{A}$. The two internal parameters that need to

be set for the edge cross-validation are the selection probability p and the number of repetitions N. Our numerical experiments suggest that the accuracy is stable for $p \in (0.85, 1)$ and that the choice of N does not have much effect after applying stability selection; see the Supplementary Material. In all of our examples, we take $p = 0.9$ and $N = 3$.

2.3. *Network recovery by matrix completion*

There are many algorithms that can be used to recover $\hat{A}$ from the training pairs. Define the operator $P_\Omega : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ by $(P_\Omega A)_{ij} = A_{ij}\,\mathrm{I}\{(i,j) \in \Omega\}$, where I denotes the indicator function, which replaces held-out entries by zeros. A generic low-rank matrix completion procedure solves the problem

$$\min_W F(P_\Omega W, P_\Omega A) \quad \text{subject to} \quad \operatorname{rank}(W) \leqslant \hat{K}, \tag{1}$$

where $\hat{K}$ is the rank constraint and F is a loss function measuring the discrepancy between W and A on entries in Ω, e.g., the sum of squared errors or the binomial deviance. Since the problem is nonconvex due to the rank constraint, many computationally feasible variants of (1) have been proposed for use in practice, which are obtained via convex relaxation and/or problem reformulation. While any such method can be used in edge cross-validation, for concreteness we follow the singular value thresholding procedure to construct a low-rank approximation

$$\hat{A} = S_H\left(\frac{1}{p}P_\Omega A,\ \hat{K}\right), \tag{2}$$

where $S_H(P_\Omega A, \hat{K})$ denotes the rank-$\hat{K}$ truncated singular value decomposition of a matrix $P_\Omega A$; that is, if the singular value decomposition of $P_\Omega A$ is $P_\Omega A = UDV^{\mathrm{T}}$ where $D = \operatorname{diag}(\sigma_1, \ldots, \sigma_n)$ with $\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_n \geqslant 0$, then $S_H(P_\Omega A, \hat{K}) = UD_{\hat{K}}V^{\mathrm{T}}$ where $D_{\hat{K}} = \operatorname{diag}(\sigma_1, \ldots, \sigma_{\hat{K}}, 0, \ldots, 0)$.

This matrix completion procedure is similar to the universal singular value thresholding method of Chatterjee (2015), except that we fix K and always use the top K eigenvalues rather than a universal constant to threshold the σ_i. This method is computationally efficient as it requires only a partial singular value decomposition of the adjacency matrix with held-out entries replaced by zeros, which is typically sparse. It runs easily on a network of size $10^4 - 10^5$ on a laptop computer. In principle, one can use any matrix completion algorithm satisfying a bound similar to the one in Theorem 1. One can choose a more sophisticated method such as that of Keshavan et al. (2009) or Mazumder et al. (2010) if the size of the network allows, but since cross-validation is already computationally intensive, we have prioritized lowering the computational cost. Additionally, imputation accuracy is not the primary goal; we expect, and in fact need, noisy versions of A. As a small-scale illustration, we compare our singular value decomposition method with the iterative hardImpute algorithm of Mazumder et al. (2010) in the context of edge cross-validation; see the Supplementary Material. We find that while the hardImpute algorithm improves the accuracy of matrix completion by itself, it takes longer to compute and does not provide any tangible improvement in model selection, which is the ultimate goal here.

Remark 1. In some situations, the rank of M itself is directly associated with the model to be selected; see the examples in the Supplementary Material. In such cases, the matrix completion rank $\hat{K}$ should be selected as part of the model selection, omitting Step 1 in Algorithm 1 and instead merging Steps 2(b) and 2(c) and using a value of $\hat{K}$ corresponding to the model being evaluated. See § § 3.1 and 3.2 for details.

Remark 2. If an upper bound on $\|M\|_\infty = \max_{ij}|M_{ij}|$ is available, say $\|M\|_\infty \leqslant \bar{d}/n$, an improved estimator $\tilde{A}$ can be obtained by truncating the entries of $\hat{A}$ to the interval $[0, \bar{d}/n]$, as in Chatterjee (2015). A trivial option of truncating to the interval $[0, 1]$ is always available, which ensures that $\tilde{A}$ will be a better estimator of M than $\hat{A}$ is in Frobenius norm. We did not observe any substantial improvement in model selection performance from truncation, however. In some applications, a binary adjacency matrix may be needed for subsequent model fitting; if that is the case, a binary matrix can be obtained from $\tilde{A}$ by using one of the standard link prediction methods, for example by thresholding at 0.5.

Remark 3. An alternative to matrix completion is to simply replace all the held-out entries with zeros and use the resulting matrix A^0 for model estimation. The resulting model estimate $\hat{M}^0$ of the probability matrix $E(A^0)$ is a biased estimator of M, but since we know the sampling probability p, we can remove this bias by setting $\hat{M}^* = \hat{M}^0/p$, as in Chatterjee (2015) and Gao et al. (2016), and then use $\hat{M}^*$ for prediction and for calculating the cross-validation error. This method is valid as long as the adjacency matrix is binary, and it is probably the simplest of all methods, though surprisingly we did not find any explicit references to it in the literature. In particular, for the stochastic block model it is equivalent to our general edge cross-validation procedure when (2) is used for matrix completion. However, in applications beyond block models these two approaches will give different results, and we have empirically observed that edge cross-validation with matrix completion works better and is much more robust to the choice of p. Moreover, filling in zeros, unlike performing matrix completion, does not work for weighted networks since it would clearly change the weight distribution, which cannot be fixed via a simple rescaling by p. We will not pursue this approach further.

Remark 4. Another matrix completion option is the 1-bit matrix completion approach (Cai & Zhou, 2013; Davenport et al., 2014; Bhaskar & Javanmard, 2015), which uses binomial deviance instead of the least squares loss, and assumes that some smooth transformation of M is of low rank. In particular, the special case of the projection latent space model fits within this framework. However, 1-bit matrix completion methods are generally much more computationally demanding than Frobenius norm-based completion, and given that computational cost is paramount for cross-validation while accurate matrix imputation is of secondary concern, we will not pursue 1-bit matrix completion further.

2.4. *Theoretical justification*

Intuitively, edge cross-validation should work well if $\hat{A}$ reflects relevant structural properties of the true underlying model. The following theorem formalizes this intuition. All our results will be expressed in terms of the number of nodes n, the sampling probability p, which controls the size of the training set, the rank K of the true matrix M, and an upper bound on the expected node degree d, defined to be any value such that $\max_{ij} M_{ij} \leqslant d/n$, a crucial quantity for network concentration results. We can always trivially set $d = n$, but we will also consider the sparse network case with $d = o(n)$.

THEOREM 1. *Let M be a probability matrix of rank K and let d be as defined above. Let A be an adjacency matrix with edges sampled independently and $E(A) = M$. Let Ω be an index matrix for a set of node pairs selected independently with probability $p \geqslant C_1(\log n)/n$ for some absolute constant C_1, where $\Omega_{ij} = 1$ if the node pair (i, j) is selected and 0 otherwise. If $d \geqslant C_2 \log n$ for some absolute constant C_2, then with probability at least $1 - 3n^{-\delta}$ for some $\delta > 0$, the completed*

matrix $\hat{A}$ defined in (2) with $\hat{K} = K$ satisfies

$$\|\hat{A} - M\| \leqslant \tilde{C} \max\left\{\left(\frac{Kd^2}{np}\right)^{1/2}, \left(\frac{d}{p}\right)^{1/2}, \frac{(\log n)^{1/2}}{p}\right\}, \tag{3}$$

where $\tilde{C} = \tilde{C}(\delta, C_1, C_2)$ is a constant that depends only on C_1, C_2 and δ. This also implies

$$\frac{\|\hat{A} - M\|_{\mathrm{F}}^2}{n^2} \leqslant \frac{\tilde{C}^2}{2} \max\left(\frac{K^2 d^2}{n^3 p}, \frac{Kd}{n^2 p}, \frac{K \log n}{n^2 p^2}\right).$$

This theorem holds for both directed and undirected networks; it can also be equivalently written in terms of the size $|\Omega|$, since $|\Omega| \sim n^2 p$. From now on, we treat p as a constant for simplicity, considering it is a user-chosen parameter. We first compare Theorem 1 with known rates for previously studied network problems. In this case, the spectral norm error bound (3), taking into account the assumption $d \geqslant C_2 \log n$, becomes

$$\|\hat{A} - M\| \leqslant \tilde{C} \max\left\{\left(\frac{Kd}{n}\right)^{1/2}, 1\right\} d^{1/2}. \tag{4}$$

The bound (4) implies that the rate of concentration of $\hat{A}$ around M is the same as the rate of concentration of the full adjacency matrix A around its expectation (Lei & Rinaldo, 2014; Chin et al., 2015; Le et al., 2017), as long as $Kd/n \leqslant 1$. The sparser the network, the weaker the requirement on K. For instance, when the network is moderately sparse with $d = O(\log n)$, we only need $K \leqslant n/(\log n)$. Although it may seem counter-intuitive, this is because the dependence on K in the bound comes entirely from M itself. A sparse network means that most entries of M are very small, so replacing the missing entries in A with zeros does not contribute much to the overall error and thus the requirement on K can be less stringent. While for sparse networks the estimator is noisier, the noise bounds are of the same order for the complete and the incomplete networks when p is a constant, and so the two concentration bounds still match.

Theorem 1 essentially indicates that $\|\hat{A} - M\| \approx \|A - M\|$ if we assume $Kd \leqslant n$. Thus, in the sense of concentration in spectral norm, we can treat $\hat{A}$ as a network sampled from the same model. Under many specific models, such a concentration of $\hat{A}$ is sufficient to ensure model estimation consistency at the same rate as can be obtained by using the original matrix A, and it also gives good properties with respect to model selection.

3. Examples of edge cross-validation for model selection

3.1. *Model-free rank estimators*

The rank constraint in the matrix completion problem for Algorithm 1 may be unknown, and we need to choose or estimate it in order to apply edge cross-validation. When the true model is a generic low-rank model such as the random dot product graph model, selecting $\hat{K}$ is essentially the same as selecting its latent space dimension. Rank selection for a general low-rank matrix, not a network, by cross-validation was studied in Owen & Perry (2009) and Kanagal & Sindhwani (2010), where the matrix was split by blocks or multiple blocks instead of by individual entries, and performance was evaluated on the task of nonnegative matrix factorization, a completely different setting from ours. More generally, selection of $\hat{K}$ can itself be treated as a

model selection problem, since the completed matrix $\hat{A}$ itself is a low-rank approximation to the unknown underlying probability matrix M.

To find a suitable value for $\hat{K}$, one has to compare the completions $\hat{A}$ for each candidate rank with A in some way. We take the natural approach of directly comparing the values of $\hat{A}$ and A on the held-out set. We can use the sum of squared errors on the held-out entries, $\sum_{(i,j)\in\Omega^c}(A_{ij}-\hat{A}_{ij})^2$, or, when A is binary, the binomial deviance as the loss function to be optimized. Another possibility is to evaluate how well $\hat{A}$ predicts links for unweighted networks. We can predict $\tilde{A}_{ij} = \mathrm{I}\{\hat{A}_{ij} > c\}$ for all entries in the held-out set Ω^c for a threshold c, and vary c to obtain a sequence of link prediction results. A common measure of prediction performance is the area under the receiver operating characteristic curve, which compares false positive rates and true positive rates for all values of c, with perfect prediction corresponding to an area under the curve of 1, and random guessing corresponding to an area of 0.5. Therefore, the negative area under the curve can also be used as the loss function. In summary, the completion rank K can be chosen as described in Algorithm 2.

Algorithm 2. Model-free edge cross-validation for rank selection.

Input: an adjacency matrix A, the training proportion p, the maximum possible rank $K_{\max}$, and the number of replications N.

Step 1. For $m = 1, \ldots, N$:

(a) Randomly choose a subset of node pairs $\Omega \subset \mathcal{V} \times \mathcal{V}$ by selecting each pair independently with probability p.
(b) Apply a low-rank matrix completion algorithm to (A, Ω) to obtain $\hat{A}$ with rank $\hat{K}$.
(c) For each of the candidate models $k = 1, \ldots, K_{\max}$, apply the low-rank matrix completion algorithm to (A, Ω) with rank k to obtain $\hat{A}$; calculate the value of the loss function for using $\hat{A}$ to predict A on Ω^c, denoted by $L_k^{(m)}$.

Step 2. Let $L_k = \sum_{m=1}^{N} L_k^{(m)}/N$ and return $\hat{K} = \min_k L_k$.

If the loss is the sum of squared errors, Algorithm 2 can be viewed as a network analogue of the tuning strategy of Mazumder et al. (2010). In practice, we have observed that both the imputation error and the area under the curve work well in general rank estimation tasks. For block models, they perform comparably to likelihood-based methods most of the time.

From a theoretical perspective, the model of rank K is a special case of the model of rank $K+1$, and while the former is preferable for parsimony, the two models will give very similar model fits, unless overfitting occurs. A reasonable goal for model selection in this situation is to guarantee that the selected rank is not underselected; the same guarantee was provided by Chen & Lei (2018). The lack of protection against overselection is a known issue for cross-validation in many problems and has been rigorously shown for regression (Shao, 1993; Zhang, 1993; Yang, 2007; Lei, 2017) whenever the training proportion is nonvanishing.

Assumption 1. We have $M = \rho_n M^0$ with $M^0 = U\Sigma^0 U^{\mathrm{T}}$ being a probability matrix, where $\Sigma^0 = \mathrm{diag}(\lambda_1^0, \ldots, \lambda_K^0)$ is the diagonal matrix of nonincreasing eigenvalues and $U = (U_1, \ldots, U_K)$ contains the corresponding eigenvectors. There exists a positive constant ψ_1 such that $n\psi_1^{-1} \leqslant \lambda_K^0 \leqslant \lambda_1^0 \leqslant \psi_1 n$, and the minimum gap between any two distinct eigenvalues is at least $n/(2\psi_1)$. Also, $\max_{i\in[n]} \sum_{j\in[n]} M_{ij}^0 \geqslant \psi_2 n \max_{ij} M_{ij}^0$ for some positive constant ψ_2, i.e., the values of M^0 are all of similar magnitude. With this parameterization, the expected node degree is bounded by $\lambda_n = n\rho_n$.

Another quantity that we need is matrix coherence, introduced by Candès & Recht (2009). Under the parameterization of Assumption 1, coherence of P is defined as

$$\mu(M) = \max_{i\in[n]} \frac{n}{K}\|U^{\mathrm{T}} e_i\|^2 = \frac{n}{K}\|U\|_{2,\infty}^2.$$

To control prediction errors on the held-out entries in edge cross-validation, we need matrix completion to work well for most entries, for which it is generally believed in the literature that matrix incoherence is necessary (Chen et al., 2015; Chi & Li, 2019). We will follow this literature and assume $\mu(M)$ to be bounded, although in our context this assumption can be relaxed at the cost of a stronger condition on the network density.

Assumption 2 (Incoherent matrix). Under Assumption 1, the coherence of P^0 is bounded, with $\mu(P^0) \leqslant a$ for some constant $a > 1$.

Intuitively, Assumption 2 says that the mass of the eigenvectors of P is not concentrated on a small number of coordinates. There is a large class of matrices satisfying the above bounded incoherence condition (Candès & Recht, 2009; Candès & Tao, 2010). In the context of networks, it is easy to verify that, for example, the stochastic block model with a positive-semidefinite probability matrix and nonvanishing communities satisfies both Assumptions 1 and 2. In the special case of a fixed K and $B = \rho\cdot[(1-\beta)I + \beta\,\mathbb{1}\mathbb{1}^{\mathrm{T}}]$, a sufficient condition for positive semidefiniteness is $\beta \leqslant 1/K$, implying a certain degree of assortativity in the network. The degree-corrected stochastic block model and the configuration model also satisfy these assumptions with similar restrictions on the parameters, as long as the variability in the node degrees is of the same order for all nodes. In general, any model that does not have spiky nodes that are very different from the other nodes should satisfy these assumptions, possibly with some additional constraints on the parameter space.

We next state a result on model selection, which is the primary task of edge cross-validation.

THEOREM 2 (CONSISTENCY UNDER THE RANDOM DOT PRODUCT GRAPH MODEL). *Assume that A is generated from a random dot product graph model satisfying Assumptions 1 and 2, with latent space dimension K. Let $\hat{K}$ be the output of Algorithm 2. If the sum of squared errors is used as the loss and the expected degree satisfies $\lambda_n/(n^{1/3}\log^{4/3} n) \to \infty$, then*

$$\mathrm{pr}(\hat{K} < K) \to 0.$$

To the best of our knowledge, Theorem 2 gives the first model selection guarantee under the random dot product graph model.

3.2. *Model selection for block models*

We now apply edge cross-validation to model selection under the stochastic block model and its degree-corrected version, referred to as block models for conciseness. The choice of fitting method is not crucial for model selection, and many consistent methods are now available for fitting both models (Karrer & Newman, 2011; Zhao et al., 2012; Amini et al., 2013; Bickel et al., 2013). Here we use one of the simplest, fastest and most common methods, namely spectral clustering on the Laplacian $L = D^{1/2}AD^{1/2}$, where D is the diagonal matrix of node degrees. For the stochastic block model, spectral clustering takes the K leading eigenvectors of L, arranges them in an $n \times K$ matrix U, and applies the K-means clustering algorithm to the rows of U to obtain cluster assignments for the n nodes. For the degree-corrected block model, the rows are normalized first and then the same algorithm applied.

Spectral clustering enjoys asymptotic consistency under the block models when the average degree grows at least as fast as $\log n$ (Rohe et al., 2011; Lei & Rinaldo, 2014; Sarkar & Bickel, 2015). The possibility of strong consistency for spectral clustering was recently discussed in Eldridge et al. (2017), Abbe et al. (2017) and Su et al. (2019). Variants of spectral clustering, such as spherical spectral clustering (Qin & Rohe, 2013; Lei & Rinaldo, 2014) and the SCORE method (Jin, 2015), are consistent under the degree-corrected model.

Since both the stochastic block model and the degree-corrected model are undirected network models, we use the undirected edge cross-validation procedure, selecting edges at random from pairs (i,j) with $i < j$ and including the pair (j,i) whenever (i,j) is selected. Once node memberships are estimated, the other parameters are easy to estimate by conditioning on node labels, for example using the maximum likelihood estimation evaluated on the available node pairs. Let $\hat{C}_k = \{i : (i,j) \in \Omega, \hat{c}_i = k\}$ be the estimated member sets for each group $k = 1, \ldots, K$. Then we can estimate the entries of the probability matrix B as

$$\hat{B}_{kl} = \frac{\sum_{(i,j)\in\Omega} A_{ij}\, \mathrm{I}(\hat{c}_i = k,\, \hat{c}_j = l)}{\hat{n}^{\Omega}_{kl}}, \tag{5}$$

where

$$\hat{n}^{\Omega}_{kl} = \begin{cases} |\{(i,j) \in \Omega : \hat{c}_i = k,\, \hat{c}_j = l\}|, & k \neq l, \\ |\{(i,j) \in \Omega : i < j,\, \hat{c}_i = \hat{c}_j = k\}|, & k = l. \end{cases}$$

Under the degree-corrected model, the probability matrix can be estimated similarly to Karrer & Newman (2011), Zhao et al. (2012) and Joseph & Yu (2016) via the Poisson approximation, by letting

$$\hat{O}^{*}_{kl} = \sum_{(i,j)\in\Omega} A_{ij}\, \mathrm{I}(\hat{c}_i = k,\, \hat{c}_j = l)$$

and setting

$$\hat{\theta}_i = \frac{\sum_{j:\,(i,j)\in\Omega} A_{ij}}{\sum_{k=1}^{K} \hat{O}^{*}_{\hat{c}_i,k}}, \quad \hat{P}_{ij} = \hat{\theta}_i \hat{\theta}_j \hat{O}^{*}_{\hat{c}_i\hat{c}_j} / p.$$

The probability estimate $\hat{P}$ is scaled by p to reflect missing edges, which makes it slightly different from the estimator for the fully observed degree-corrected model (Karrer & Newman, 2011). This rescaling happens automatically in the estimator (5) since the sums in both the numerator and the denominator range over Ω only. Finally, the loss function can again be the sum of squared errors or the binomial deviance; we have found that the L_2 loss works slightly better in practice for the block models.

The model selection task here includes the choice of the stochastic block model versus the degree-corrected model and the choice of K. Suppose we consider the number of communities ranging from 1 to $K_{\max}$. The candidate set of models in Algorithm 1 is then both of the block models with K varying from 1 to $K_{\max}$. The edge cross-validation procedure for this task is presented next, in Algorithm 3.

Algorithm 3. Parametric edge cross-validation procedure.

Input: an adjacency matrix A, the largest number $K_{\max}$ of communities to consider, the training proportion p, and the number of replications N.

Step 1. For $m = 1, \ldots, N$:

(a) Randomly choose a subset of node pairs Ω, selecting each pair (i,j), for $i < j$, independently with probability p, and adding (j,i) if (i,j) is selected.

(b) For $k = 1, \ldots, K_{\max}$:

(i) Apply matrix completion to (A, Ω) with rank constraint k to obtain $\hat{A}_k$.

(ii) Run spectral clustering on $\hat{A}_k$ to obtain the estimated stochastic block model membership vector $\hat{c}_{1,k}^{(m)}$, and use spherical spectral clustering to obtain the estimated degree-corrected model $\hat{c}_{2,k}^{(m)}$.

(iii) Estimate the two models' probability matrices, $\hat{M}_{1,k}^{(m)}$ and $\hat{M}_{2,k}^{(m)}$ based on $\hat{c}_{1,k}^{(m)}$ and $\hat{c}_{2,k}^{(m)}$, respectively, and evaluate the corresponding losses $L_{q,k}^{(m)}$ $(q = 1, 2)$ by applying the loss function L with the estimated parameters to A_{ij} for $(i,j) \in \Omega^{\mathrm{c}}$.

Step 2. Let $L_{q,k} = \sum_{m=1}^{N} L_{q,k}^{(m)}/N$. Return $(\hat{q}, \hat{K}) = \arg\min_{q=1,2} \min_{k=1,\ldots,K_{\max}} L_{q,k}$ as the best model, with $\hat{q} = 1$ indicating no degree correction and $\hat{q} = 2$ indicating degree correction.

As a special case, one can also consider the task of just choosing K under a specific model, the stochastic block model or the degree-corrected model, for which there are many methods available (Latouche et al., 2012; McDaid et al., 2013; Bickel & Sarkar, 2016; Lei, 2016; Saldana et al., 2017; Wang & Bickel, 2017; Chen & Lei, 2018; Le & Levina, 2019). In particular, Theorem 1 can be modified, as detailed in the Supplementary Material, to show that the parametric edge cross-validation procedure, Algorithm 3, achieves one-sided consistency in choosing K under the stochastic block model, under the following standard assumption (Lei & Rinaldo, 2014).

Assumption 3. The probability matrix $B^{(n)}$ is of the form $B^{(n)} = \rho_n B_0$, where B_0 is a fixed $K \times K$ symmetric nonsingular matrix with all entries in $[0, 1]$ and with K fixed, and therefore the expected node degree is $\lambda_n = n\rho_n$. There exists a constant $\gamma > 0$ such that $\min_k n_k > \gamma n$, where $n_k = |\{i : c_i = k\}|$.

THEOREM 3 (CONSISTENCY UNDER THE STOCHASTIC BLOCK MODEL). *Let A be the adjacency matrix of a network generated from the stochastic block model satisfying Assumption 3, and suppose that, as prior knowledge, the model is known to be the stochastic block model, but the number of communities K is to be estimated. Assume $\lambda_n / \log n \to \infty$. Let $\hat{K}$ be the selected number of communities obtained using Algorithm 3 with the L_2 loss. Then*

$$\mathrm{pr}(\hat{K} < K) \to 0.$$

If we assume $\lambda_n n^{-2/3} \to \infty$ and that all entries of B_0 are positive, then the same result holds also for the binomial deviance loss.

The theorem requires a stronger assumption for the binomial deviance result than for the L_2 loss. While these conditions may not be tight, empirically the L_2 loss performs better, as shown in the Supplementary Material, which can intuitively be explained by the instability of the binomial

deviance near 0. Just as in our result for the random dot product graph and in Chen & Lei (2018), we have a one-sided guarantee, but the assumption on the expected degree is much weaker than that of Theorem 2. This reflects a natural trade-off between gaining better rates under a parametric version of the edge cross-validation and imposing additional model assumptions.

3.3. *Parameter tuning in graphon estimation*

Graphon, or probability matrix, estimation is another general task which often relies on tuning parameters that can be determined by cross-validation. Zhang et al. (2017) proposed a method, called neighbourhood smoothing, for estimating M instead of f under the assumption that f is a piecewise-Lipschitz function, avoiding the ambiguity associated with the measure-preserving transformation. They showed that their method achieves a nearly optimal rate while requiring only polynomial complexity for computation. The method depends on a tuning parameter h which controls the degree of smoothing. The theory suggests taking $h = \tau(\log n/n)^{1/2}$ for some τ.

This is a setting in which we have no reason to assume a known rank for the true probability matrix and M does not have to be of low rank. However, for a smooth graphon function, a low-rank matrix can approximate M reasonably well (Chatterjee, 2015). The edge cross-validation procedure under the graphon model now has to select the best rank for its internal matrix completion step. Specifically, in each split we can run the rank estimation procedure discussed in § 3.1 to estimate the best rank for approximation and the corresponding $\hat{A}$ as the input for the neighbourhood smoothing algorithm. The selected tuning parameter is the one that minimizes the average prediction error.

The edge cross-validation algorithm can also be used in other tuning parameter selection problems. In the Supplementary Material we demonstrate its application to tuning network regularization in spectral clustering.

3.4. *Stability selection*

Stability selection (Meinshausen & Bühlmann, 2010) was proposed as a general method for reducing noise by repeating model selection many times over random splits of the data and keeping only those features that are selected in the majority of splits; any cross-validation procedure can benefit from stability selection since it relies on random data splits. An additional advantage of stability selection in our context is increased robustness to the choices of p and N; see the Supplementary Material. Chen & Lei (2018) used this idea as well, repeating the procedure multiple times and choosing the most frequently selected model. We adopt the same strategy for edge cross-validation, along with the cross-validation method of Chen & Lei (2018), choosing the model selected most frequently out of 20 replications. When we need to select a numerical parameter rather than a model, we can also average the values selected over the 20 replications (and round to an integer if needed, say for the number of communities). Overall, picking the most frequent selection is more robust to different tasks, although picking the average may work better in certain situations. More details are given in § 4.

4. Numerical performance evaluation

4.1. *Model selection under block models*

Following Chen & Lei (2018), we evaluate the performance under the block models in choosing both the model, with or without degree correction, and the number K of communities simultaneously. The setting for all simulated networks in this section is as follows. For the degree-corrected block model, we first sample 300 values from the power-law distribution with lower

bound 1 and scaling parameter 5, and then set the node degree parameters θ_i ($i = 1, \ldots, n$) by randomly and independently choosing one of these 300 values. For the stochastic block model, we set $\theta_i = 1$ for all i. We take the size of the communities to be equal; the imbalanced community situation is studied in the Supplementary Material. Let $B_0 = (1 - \beta)I + \beta \mathbb{1}\mathbb{1}^{\mathrm{T}}$ and $B \propto \Theta B_0 \Theta$, so that β is the out-in ratio, the ratio of the between-block probability to the within-block probability of an edge. The scaling is selected so that the average node degree is λ. We consider several combinations of the size and the number of communities: $(n = 600, K = 3)$, $(n = 600, K = 5)$ and $(n = 1200, K = 5)$. For each configuration we then vary two aspects of the model.

(I) Sparsity: set the expected average degree λ to 15, 20, 30 or 40, fixing $t = 0$ and $\beta = 0.2$.
(II) Out-in ratio: set β to 0, 0.25 or 0.5, fixing $\lambda = 40$ and $t = 0$.

All results are based on 200 replications. The four methods compared on this task are: the edge cross-validation procedure, Algorithm 3, with the L_2 loss; its stable version where the most frequent selection of 20 independent repetitions is returned; and the corresponding versions of the procedure of Chen & Lei (2018). Here we present only the results from using the L_2 loss for model selection, since we have observed that it works better than binomial deviance for both methods. The performance using the binomial deviance as the loss is reported in the Supplementary Material.

Table 1 shows the fraction of times the correct model was selected when the true model is the degree-corrected model. Across all settings, stability selection improves performance as long as the single cross-validation is working reasonably well to start with. This is to be expected, since stability selection is only a variance-reduction step, and it cannot help if the original procedure is not working. Although the method of Chen & Lei (2018) works well in easier settings that involve a smaller number of communities, e.g., a denser network or a smaller out-in ratio, it quickly loses accuracy in model selection as the problem becomes harder. In contrast, the edge cross-validation method yields better selection in all cases, and in more difficult settings the difference is very large. In the Supplementary Material we present results from the experiment with the stochastic block model as the underlying truth, and the findings remain the same.

Another popular model selection problem under the block models is selection of the number of communities assuming that the true model, with or without degree correction, is known. We have conducted extensive simulation experiments on this task by comparing the two cross-validation methods above and a few other model-based methods; the details are given in the Supplementary Material. Between the two cross-validation methods, the edge cross-validation procedure is again a clear winner. However, the model-based methods are overall more effective than cross-validation methods, as expected.

4.2. *Tuning nonparametric graphon estimation*

In this subsection we demonstrate the performance of edge cross-validation in tuning τ in the neighbourhood smoothing estimation for a graphon model used in Zhang et al. (2017).

The tuning procedure is very stable for the graphon problem and stability selection is unnecessary. Figure 1 shows the tuning results for two graphon examples taken from Zhang et al. (2017), both for networks with $n = 500$ nodes. Graphon 1 is a block model, though this information is never used, which is a piecewise-constant function, and M is of low rank. Graphon 2 is a smoothly varying function that is not of low rank; see Zhang et al. (2017) for more details. The errors are displayed as the median over 200 replications with a 95% confidence interval, calculated by bootstrap, of the normalized Frobenius error $\|\hat{M} - M\|_{\mathrm{F}}/\|M\|_{\mathrm{F}}$. For graphon 1, which is of low rank, the edge cross-validation procedure works extremely well and picks the best τ from

Table 1. *Overall model selection by two cross-validation methods; the true model is the degree-corrected block model*

Configuration				Proposed method		Chen & Lei (2018)	
K	n	λ	β	L_2 loss	L_2 loss + stability	L_2 loss	L_2 loss + stability
3	600	15	0.2	0.73	0.87	0.00	0.00
		20	0.2	0.97	0.99	0.02	0.00
		30	0.2	1.00	1.00	0.43	0.40
		40	0.2	1.00	1.00	0.88	0.98
5	600	15	0.2	0.49	0.58	0.00	0.00
		20	0.2	0.90	0.95	0.00	0.00
		30	0.2	0.99	1.00	0.05	0.01
		40	0.2	0.99	1.00	0.27	0.24
5	1200	15	0.2	0.67	0.76	0.00	0.00
		20	0.2	0.99	0.99	0.00	0.00
		30	0.2	1.00	1.00	0.04	0.00
		40	0.2	1.00	1.00	0.41	0.33
3	600	40	0.1	1.00	1.00	0.99	1.00
		40	0.2	1.00	1.00	0.88	0.98
		40	0.5	0.95	0.97	0.00	0.00
5	600	40	0.1	1.00	1.00	0.79	0.96
		40	0.2	0.99	1.00	0.27	0.24
		40	0.5	0.00	0.00	0.00	0.00
5	1200	40	0.1	1.00	1.00	0.90	0.99
		40	0.2	1.00	1.00	0.41	0.33
		40	0.5	0.00	0.00	0.00	0.00

the candidate set most of the time. For graphon 2, which is not of low rank and therefore more challenging for a procedure based on a low-rank approximation, the edge cross-validation does not always choose the best τ, but it still achieves a fairly competitive error rate by successfully avoiding the bad range of τ values. This example illustrates that the choice of constant can lead to a big difference in estimation error, and that the edge cross-validation method is successful at choosing it.

5. Community detection in a statistician citation network

In this section, we demonstrate model selection on a publicly available dataset compiled by Ji & Jin (2016). This dataset contains information on the title, author, year, citations and DOI of all papers published between 2003 and 2012 in the four top statistics journals, *The Annals of Statistics*, *Biometrika*, *Journal of the American Statistical Association* and *Journal of the Royal Statistical Society, Series B*, and includes 3607 authors and 3248 papers in total. The dataset was carefully curated by Ji & Jin (2016) to resolve name ambiguities and is relatively interpretable, at least to statisticians.

Because the citations of all the papers are available, we can construct the citation network between authors, as well as between papers; here we focus on authors as we are looking for research communities of people. We therefore construct a weighted undirected network between authors, where the weight is the total number of their mutual citations. The largest connected component of the network contains 2654 authors. Thresholding the weight to binary resulted in all methods for estimating K selecting an unrealistically large and uninterpretable value, suggesting

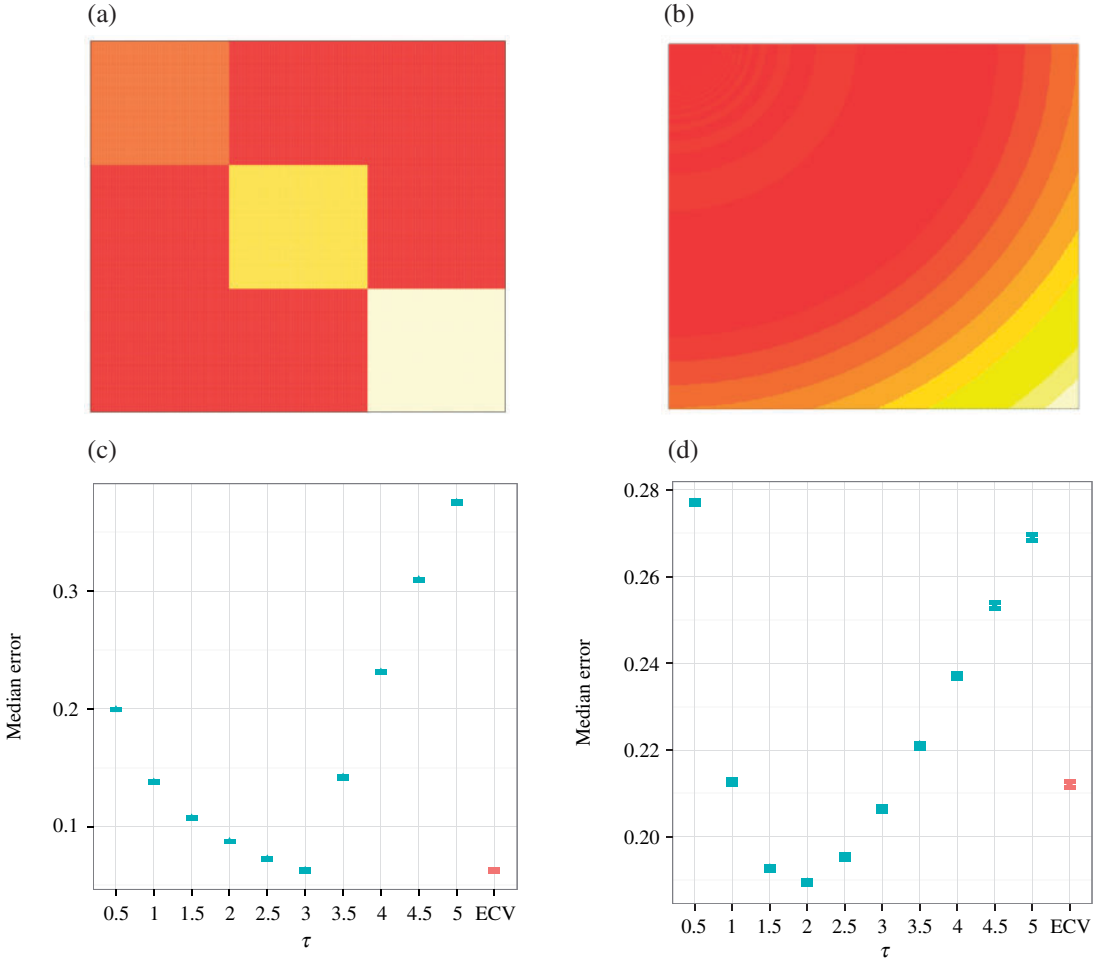

Fig. 1. Parameter tuning for piecewise-constant graphon estimation, comparing the edge cross-validation (ECV, red) and predefined (blue) methods: (a) graphon 1 heatmap; (b) graphon 2 heatmap; (c) graphon 1 errors; (d) graphon 2 errors.

that the network is too complex to be adequately described by a binary block model. Since the weights are available and contain much more information than just the presence of an edge, we analyse the weighted network instead; seamlessly switching between binary and weighted networks is a strength of the edge cross-validation approach. Many real-world networks display a core-periphery structure, and citation networks are especially likely to have this form. We focus on analysing the core of the citation network, following the procedure proposed by Wang & Rohe (2016) to extract it: we delete nodes with fewer than 15 mutual citations and their corresponding edges, and repeat until the network no longer changes. This results in a network with 706 authors, as shown in Fig. 2. The individual node citation count ranges from 15 to 703, with a median of 30.

Block models are not defined for weighted networks, but the Laplacian is still well-defined and so the spectral clustering algorithm for community detection can be applied. The model-free version, Algorithm 2, can be used to determine the number of communities. We apply the edge cross-validation procedure with the sum-of-squared-errors loss and repeat it 200 times, with the candidate values for K ranging from 1 to 50. The stable version of the edge cross-validation method selects $K = 20$. We also used edge cross-validation to tune the regularization parameter for spectral clustering, as described in the Supplementary Material. It turns out that

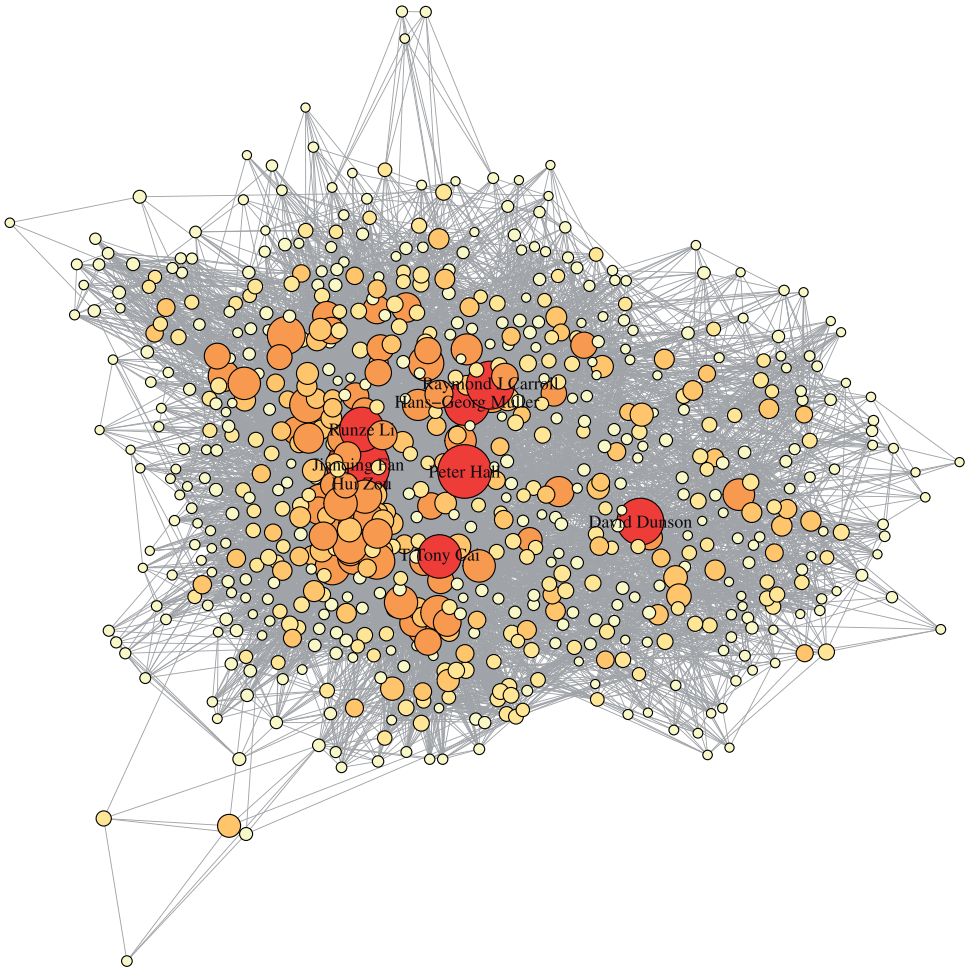

Fig. 2. The core of the statistician citation network: the network has 706 nodes, and the node citation count, ignoring directions, ranges from 15 to 703; node size and colour indicate the citation count, with nodes having larger citation counts being larger and darker.

the regularization does make the result more interpretable. We list the 20 communities in Table 2, with each community represented by the 10 authors having the largest number of citations, along with subjective and tentative names that we assigned to these communities. The names were chosen according to the interests or area of contributions of the majority of the authors; as they are based exclusively on data collected in the period 2003–2012, people who have worked on many topics over many years tend to appear under the topic they devoted the most attention to in that time period. Many communities can be easily identified by their common research interests; high-dimensional inference, a topic that many researchers published on in that period, is subdivided into several subcommunities that are themselves interpretable: communities 1, 2, 4, 5, 10, 12 and 15 in Table 2. Overall, these groups are fairly easy to interpret for those familiar with the statistics literature of the decade 2003–2012.

6. DISCUSSION

The general scheme of leaving out entries at random followed by matrix completion may be useful for other resampling-based methods. In particular, an interesting question we plan to

Table 2. *The eight authors with the largest total citation numbers, ignoring the direction, in each of* 20 *communities, along with the community interpretations; the communities are ordered by size, and authors within a community are ordered by mutual citation count*

	Interpretation [size]	Authors
1	High-dimensional inference (multiple testing, machine learning) [57]	T. Tony Cai, Jiashun Jin, Larry Wasserman, Christopher Genovese, Bradley Efron, John D. Storey, David L. Donoho, Yoav Benjamini
2	High-dimensional inference (sparse penalties) [53]	Hui Zou, Ming Yuan, Yi Lin, Trevor J. Hastie, Robert J. Tibshirani, Xiaotong Shen, Jinchi Lv, Gareth M. James
3	Functional data analysis [52]	Hans-Georg Muller, Jane-Ling Wang, Fang Yao, Yehua Li, Ciprian M. Crainiceanu, Jeng-Min Chiou, Alois Kneip, Hulin Wu
4	High-dimensional inference (theory and sparsity) [45]	Peter Buhlmann, Nicolai Meinshausen, Cun-Hui Zhang, Alexandre B. Tsybakov, Emmanuel J. Candes, Terence Tao, Marten H. Wegkamp, Bin Yu
5	High-dimensional covariance estimation [43]	Peter J. Bickel, Ji Zhu, Elizaveta Levina, Jianhua Z. Huang, Mohsen Pourahmadi, Clifford Lam, Wei Biao Wu, Adam J. Rothman
6	Bayesian machine learning [41]	David Dunson, Alan E. Gelfand, Abel Rodriguez, Michael I. Jordan, Peter Muller, Gareth Roberts, Gary L. Rosner, Omiros Papaspiliopoulos
7	Spatial statistics [41]	Tilmann Gneiting, Marc G. Genton, Sudipto Banerjee, Adrian E. Raftery, Haavard Rue, Andrew O. Finley, Bo Li, Michael L. Stein
8	Biostatistics (machine learning) [40]	Donglin Zeng, Dan Yu Lin, Michael R. Kosorok, Jason P. Fine, Jing Qin, Guosheng Yin, Guang Cheng, Yi Li
9	Sufficient dimension reduction [39]	Lixing Zhu, R. Dennis Cook, Bing Li, Chih-Ling Tsai, Liping Zhu, Yingcun Xia, Lexin Li, Liqiang Ni
10	High-dimensional inference (penalized methods) [38]	Jianqing Fan, Runze Li, Hansheng Wang, Jian Huang, Heng Peng, Song Xi Chen, Chenlei Leng, Shuangge Ma
11	Bayesian (general) [33]	Jeffrey S. Morris, James O. Berger, Carlos M. Carvalho, James G. Scott, Hemant Ishwaran, Marina Vannucci, Philip J. Brown, J. Sunil Rao
12	High-dimensional theory and wavelets [33]	Iain M. Johnstone, Bernard W. Silverman, Felix Abramovich, Ian L. Dryden, Dominique Picard, Richard Nickl, Holger Dette, Marianna Pensky
13	Mixed (causality + theory + Bayesian) [32]	James R. Robins, Christian P. Robert, Paul Fearnhead, Gilles Blanchard, Zhiqiang Tan, Stijn Vansteelandt, Nancy Reid, Jae Kwang Kim
14	Semiparametrics and nonparametrics [28]	Hua Liang, Naisyin Wang, Joel L. Horowitz, Xihong Lin, Enno Mammen, Arnab Maity, Byeong U. Park, Wolfgang Karl Hardle
15	High-dimensional inference (machine learning) [27]	Hao Helen Zhang, J. S. Marron, Yufeng Liu, Yichao Wu, Jeongyoun Ahn, Wing Hung Wong, Peter L. Bartlett, Michael J. Todd
16	Semiparametrics [24]	Peter Hall, Raymond J. Carroll, Yanyuan Ma, Aurore Delaigle, Gerda Claeskens, David Ruppert, Alexander Meister, Huixia Judy Wang
17	Mixed (causality + financial) [22]	Qiwei Yao, Paul R. Rosenbaum, Yacine Ait-Sahalia, Yazhen Wang, Marc Hallin, Dylan S. Small, Davy Paindaveine, Jian Zou
18	Biostatistics (survival, clinical trials) [22]	L. J. Wei, Lu Tian, Tianxi Cai, Zhiliang Ying, Zhezhen Jin, Peter X.-K. Song, Hui Li, Bin Nan
19	Biostatistics (genomics) [21]	Joseph G. Ibrahim, Hongtu Zhu, Jiahua Chen, Amy H. Herring, Heping Zhang, Ming-Hui Chen, Stuart R. Lipsitz, Denis Heng-Yan Leung
20	Bayesian (nonparametrics) [15]	Subhashis Ghosal, Igor Prunster, Antonio Lijoi, Stephen G. Walker, Aad van der Vaart, Anindya Roy, Judith Rousseau, J. H. van Zanten

investigate in the future is whether this strategy can be used to create something akin to bootstrap samples from a single network realization. Another direction that we have not explored in this paper is cross-validation under alternatives to the inhomogeneous Erdős–Rényi model, such as the models of Crane & Dempsey (2018) and Lauritzen et al. (2018). Edge cross-validation can also be modified for the setting in which additional node features are available (Newman & Clauset, 2016; Li et al., 2019). We leave these questions for future work.

Acknowledgement

The authors thank the associate editor and the referees for very helpful suggestions. Levina was partially supported by the U.S. National Science Foundation and Office of Naval Research. Zhu was partially supported by the U.S. National Science Foundation.

Supplementary material

Supplementary material available at *Biometrika* online includes proofs of the theoretical properties and additional simulation results.

References

Abbe, E. (2018). Community detection and stochastic block models: Recent developments. *J. Mach. Learn. Res.* **18**, 1–86.

Abbe, E., Fan, J., Wang, K. & Zhong, Y. (2017). Entrywise eigenvector analysis of random matrices with low expected rank. *arXiv:* 1709.09565.

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

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**, 2097–122.

Athreya, A., Fishkind, D. E., Levin, K., Lyzinski, V., Park, Y., Qin, Y., Sussman, D. L., Tang, M., Vogelstein, J. T. & Priebe, C. E. (2017). Statistical inference on random dot product graphs: A survey. *arXiv:* 1709.05454.

Bhaskar, S. A. & Javanmard, A. (2015). 1-bit matrix completion under exact low-rank constraint. In *Proc. 49th Annu. Conf. Information Sciences and Systems (CISS)*. New York: Curran Associates, pp. 1–6.

Bickel, P., Choi, D., Chang, X. & Zhang, H. (2013). Asymptotic normality of maximum likelihood and its variational approximation for stochastic blockmodels. *Ann. Statist.* **41**, 1922–43.

Bickel, P. J. & Sarkar, P. (2016). Hypothesis testing for automated community detection in networks. *J. R. Statist. Soc.* B **78**, 253–73.

Cai, T. & Zhou, W.-X. (2013). A max-norm constrained minimization approach to 1-bit matrix completion. *J. Mach. Learn. Res.* **14**, 3619–47.

Candès, E. J. & Plan, Y. (2010). Matrix completion with noise. *Proc. IEEE* **98**, 925–36.

Candès, E. J. & Recht, B. (2009). Exact matrix completion via convex optimization. *Foundat. Comp. Math.* **9**, 717–72.

Candès, E. J. & Tao, T. (2010). The power of convex relaxation: Near-optimal matrix completion. *IEEE Trans. Info*. Theory **56**, 2053–80.

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

Chen, K. & Lei, J. (2018). Network cross-validation for determining the number of communities in network data. *J. Am. Statist. Assoc.* **113**, 241–51.

Chen, Y., Bhojanapalli, S., Sanghavi, S. & Ward, R. (2015). Completing any low-rank matrix, provably. *J. Mach. Learn. Res.* **16**, 2999–3034.

Chi, E. C. & Li, T. (2019). Matrix completion from a computational statistics perspective. *WIREs Comp. Statist.* **11**, e1469.

Chin, P., Rao, A. & Vu, V. (2015). Stochastic block model and community detection in sparse graphs: A spectral algorithm with optimal rate of recovery. *Proc. Mach. Learn. Res.* **40**, 391–423.

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

Chung, F. & Lu, L. (2002). The average distances in random graphs with given expected degrees. *Proc. Nat. Acad. Sci.* **99**, 15879–82.
Crane, H. & Dempsey, W. (2018). Edge exchangeable models for interaction networks. *J. Am. Statist. Assoc.* **113**, 1311–26.
Davenport, M. A., Plan, Y., van den Berg, E. & Wootters, M. (2014). 1-bit matrix completion. *Info. Infer.* **3**, 189–223.
Diaconis, P. & Janson, S. (2007). Graph limits and exchangeable random graphs. *arXiv:* 0712.2749.
Eldridge, J., Belkin, M. & Wang, Y. (2017). Unperturbed: Spectral analysis beyond Davis-Kahan. *arXiv:* 1706.06516.
Erdős, P. & Rényi, A. (1960). On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci.* **5**, 17–61.
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. (2015). Rate-optimal graphon estimation. *Ann. Statist.* **43**, 2624–52.
Hoff, P. (2008). Modeling homophily and stochastic equivalence in symmetric relational data. In *Advances in Neural Information Processing Systems 20 (NIPS 2007)*. San Diego, California: Neural Information Processing Systems Foundation, pp. 657–64.
Hoff, P. D., Raftery, A. E. & Handcock, M. S. (2002). Latent space approaches to social network analysis. *J. Am. Statist. Assoc.* **97**, 1090–8.
Holland, P. W., Laskey, K. B. & Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social Networks* **5**, 109–37.
Ji, P. & Jin, J. (2016). Coauthorship and citation networks for statisticians. *Ann. Appl. Statist.* **10**, 1779–812.
Jin, J. (2015). Fast community detection by SCORE. *Ann. Statist.* **43**, 57–89.
Joseph, A. & Yu, B. (2016). Impact of regularization on spectral clustering. *Ann. Statist.* **44**, 1765–91.
Kanagal, B. & Sindhwani, V. (2010). Rank selection in low-rank matrix approximations: A study of crossvalidation for NMFs. In *Advances in Neural Information Processing Systems*, vol. 1. San Diego, California: 635 Neural Information Processing Systems Foundation, pp. 10–15.
Karrer, B. & Newman, M. E. (2011). Stochastic blockmodels and community structure in networks. *Phys. Rev.* E **83**, 016107.
Keshavan, R., Montanari, A. & Oh, S. (2009). Matrix completion from noisy entries. In *Advances in Neural Information Processing Systems 22 (NIPS 2009)*. San Diego, California: Neural Information Processing Systems 640 Foundation, pp. 952–60.
Latouche, P., Birmele, E. & Ambroise, C. (2012). Variational Bayesian inference and complexity control for stochastic block models. *Statist. Mod.* **12**, 93–115.
Lauritzen, S., Rinaldo, A. & Sadeghi, K. (2018). Random networks, graphical models and exchangeability. *J. R. Statist. Soc.* B **80**, 481–508.
Le, C. M. & Levina, E. (2019). Estimating the number of communities in networks by spectral methods. *arXiv:* 1507.00827v2.
Le, C. M., Levina, E. & Vershynin, R. (2017). Concentration and regularization of random graphs. *Random Struct. Algor*. **51**, 538–61.
Lei, J. (2016). A goodness-of-fit test for stochastic block models. *Ann. Statist.* **44**, 401–24.
Lei, J. (2017). Cross-validation with confidence. *arXiv:* 1703.07904v2.
Lei, J. & Rinaldo, A. (2014). Consistency of spectral clustering in stochastic block models. *Ann. Statist.* **43**, 215–37.
Li, T., Levina, E. & Zhu, J. (2019). Prediction models for network-linked data. *Ann. Appl. Statist.* **13**, 132–64.
Mazumder, R., Hastie, T. & Tibshirani, R. (2010). Spectral regularization algorithms for learning large incomplete matrices. *J. Mach. Learn. Res*. **11**, 2287–322.
McDaid, A. F., Murphy, T. B., Friel, N. & Hurley, N. J. (2013). Improved Bayesian inference for the stochastic block model with application to large networks. *Comp. Statist. Data Anal.* **60**, 12–31.
Meinshausen, N. & Bühlmann, P. (2010). Stability selection. *J. R. Statist. Soc.* B **72**, 417–73.
Newman, M. E. J. & Clauset, A. (2016). Structure and inference in annotated networks. *Nature Commun.* **7**, 11863.
Owen, A. B. & Perry, P. (2009). Bi-cross-validation of the SVD and the nonnegative matrix factorization. *Ann. Appl. Statist*. **3**, 564–94.
Qin, T. & Rohe, K. (2013). Regularized spectral clustering under the degree-corrected stochastic blockmodel. In *Proc. 26th Int. Conf. Neural Information Processing Systems*. New York: Association for Computing Machinery, pp. 3120–8.
Rohe, K., Chatterjee, S. & Yu, B. (2011). Spectral clustering and the high-dimensional stochastic blockmodel. *Ann. Statist.* **39**, 1878–915.
Saldana, D., Yu, Y. & Feng, Y. (2017). How many communities are there? *J. Comp. Graph. Statist.* **26**, 171–81.
Sarkar, P. & Bickel, P. J. (2015). Role of normalization in spectral clustering for stochastic blockmodels. *Ann. Statist.* **43**, 962–90.
Sengupta, S. & Chen, Y. (2018). A block model for node popularity in networks with community structure. *J. R. Statist. Soc.* B **80**, 365–86.
Shao, J. (1993). Linear model selection by cross-validation. *J. Am. Statist. Assoc.* **88**, 486–94.

Su, L., Wang, W. & Zhang, Y. (2019). Strong consistency of spectral clustering for stochastic block models. *arXiv:* 1710.06191v3.
Sussman, D. L., Tang, M. & Priebe, C. E. (2014). Consistent latent position estimation and vertex classification for random dot product graphs. *IEEE Trans. Pat. Anal. Mach. Intel.* **36**, 48–57.
Tang, M., Athreya, A., Sussman, D. L., Lyzinski, V. & Priebe, C. E. (2017). A nonparametric two-sample hypothesis testing problem for random graphs. *Bernoulli* **23**, 1599–630.
Tang, M. & Priebe, C. E. (2018). Limit theorems for eigenvectors of the normalized Laplacian for random graphs. *Ann. Statist.* **46**, 2360–415.
Wang, S. & Rohe, K. (2016). Discussion of "Coauthorship and citation networks for statisticians". *Ann. Appl. Statist.* **10**, 1820–6.
Wang, Y. X. R. & Bickel, P. J. (2017). Likelihood-based model selection for stochastic block models. *Ann. Statist.* **45**, 500–28.
Wolfe, P. J. & Olhede, S. C. (2013). Nonparametric graphon estimation. *arXiv:* 1309.5936.
Yang, Y. (2007). Consistency of cross validation for comparing regression procedures. *Ann. Statist.* **35**, 2450–73.
Young, S. J. & Scheinerman, E. R. (2007). Random dot product graph models for social networks. In *International Workshop on Algorithms and Models for the Web-Graph*. Cham, Switzerland: Springer, pp. 138–49.
Zhang, P. (1993). Model selection via multifold cross validation. *Ann. Statist.* **21**, 299–313.
Zhang, Y., Levina, E. & Zhu, J. (2017). Estimating network edge probabilities by neighbourhood smoothing. *Biometrika* **104**, 771–83.
Zhao, Y., Levina, E. & Zhu, J. (2012). Consistency of community detection in networks under degree-corrected stochastic block models. *Ann. Statist.* **40**, 2266–92.

[*Received on* 10 *May* 2018. *Editorial decision on* 17 *October* 2019]