Statistica Sinica Preprint No: SS-2021-0427

Title	A Unified Framework for Tuning Hyperparameters in Clustering Problems
Manuscript ID	SS-2021-0427
URL	http://www.stat.sinica.edu.tw/statistica/
DOI	10.5705/ss.202021.0427
Complete List of Authors	Xinjie Fan, Y. X. Rachel Wang, Purnamrita Sarkar and Yuguang Yue
Corresponding Authors	Y. X. Rachel Wang
E-mails	rachel.wang@sydney.edu.au

Notice: Accepted version subject to English editing.

Statistica Sinica

A UNIFIED FRAMEWORK FOR TUNING HYPERPARAMETERS IN CLUSTERING PROBLEMS

Xinjie Fan[1], Y. X. Rachel Wang[1, 2], Purnamrita Sarkar[2] and Yuguang Yue

University of Texas at Austin and University of Sydney

Abstract: Selecting hyperparameters for unsupervised learning problems is challenging in general due to the lack of ground truth for validation. Despite the prevalence of this issue in statistics and machine learning, especially in clustering problems, there are not many methods for tuning these hyperparameters with theoretical guarantees. In this paper, we provide a framework relying on maximizing a trace criterion connecting a similarity matrix with clustering solutions, which has provable guarantees for selecting hyperparameters in a number of distinct models. We consider both the sub-gaussian mixture model and network models to serve as examples of i.i.d. and non-i.i.d. data. We demonstrate that the same framework can be used to choose the Lagrange multipliers of penalty terms in semidefinite programming (SDP) relaxations for community detection, and the bandwidth parameter for constructing kernel similarity matrices for spectral clustering. By incorporating a cross-validation procedure, we show the framework can also do consistent model selection for network models. Using a variety

[1] Equal contribution

[2] Corresponding author

of simulated and real data examples, we show that our framework outperforms other widely used tuning procedures in a broad range of parameter settings.

Key words and phrases: Clustering, hyperparameter tuning, model selection, network models, sub-gaussian mixtures

1. Introduction

A standard statistical model has parameters, which characterize the underlying data distribution; an inference algorithm to learn these parameters typically involve hyperparameters (or tuning parameters). Popular examples include the penalty parameter in regularized regression models, the number of clusters in clustering analysis, the bandwidth parameter in kernel based clustering, nonparameteric density estimation or regression methods (Wasserman, 2006; Tibshirani et al., 2015), to name but a few. It is well-known that selecting these hyperparameters may require repeated training to search through combinations of plausible hyperparameter values and often has to rely on good heuristics and user's domain knowledge.

A classical method to do automated hyperparameter tuning is the nonparametric procedure Cross Validation (CV) (Stone, 1974; Zhang, 1993), which has been used extensively in machine learning and statistics (Friedman et al., 2001; Feng and Simon, 2020).CV has been studied extensively in supervised learning settings, particularly in low dimensional linear models (Shao, 1993; Yang et al., 2007) and penalized regression in high dimension (Wasserman and Roeder, 2009). Other notable stability based methods for model selection in similar supervised settings include Breiman et al. (1996); Bach (2008); Meinshausen and Bühlmann (2010); Lim and Yu (2016). Finally, a large number of empirical methods exist in the machine learning literature for tuning hyperparameters in various training algorithms (Bergstra and Bengio (2012); Bengio (2000); Snoek et al. (2012); Bergstra et al. (2011)), most of which do not provide theoretical guarantees.

In contrast to the supervised setting with i.i.d. data used in many of the above methods, in this paper, we consider *unsupervised* clustering problems with possible dependence structure in the datapoints. We propose an overarching framework for hyperparameter tuning and model selection for a variety of probabilistic clustering models. Here the challenge is two-fold. Firstly, since labels are not available, choosing a criterion for evaluation and in general a method for selecting hyperparameters is not easy. One may consider splitting the data in different folds and selecting the model or hyperparameter with the most stable solution. However, for multiple splits of the data, the inference algorithm may get stuck at the same local optima, and thus stability alone can lead to a suboptimal solution (Von Luxburg et al. (2010)). In Wang (2010); Fang and Wang (2012), the authors overcome this by redefining the number of clusters as one that gives the most stable clustering for a given algorithm. In Meila (2018), a semi-definite program (SDP) maximizing an inner product criterion is performed for each clustering solution, and the value of the objective function is used to evaluate the stability of the clustering. Their analysis is done without model assumptions, and when a large number of clustering solutions need to be evaluated, perfoming SDP for each solution could be computationally expensive. The second difficulty in the unsupervised clustering setting arises if there is dependence structure in the datapoints, which necessitates careful splitting procedures in a CV-based procedure.

To illustrate the generality of our framework, we focus on sub-gaussian mixtures and the statistical network models as two representative models for i.i.d. data and non i.i.d. data, where clustering is a natural problem. We diversify the models considered in Fan et al. (2020): (i) We use a different formulation for sub-gaussian mixtures with a more realistic noise structure; (ii) In addition to the Stochastic Blockmodel (SBM), we consider the more general Mixed Membership Stochastic Blockmodel (MMSB). By observing the fact that clustering algorithms typically operate on a similarity matrix arising from these models, which can be decomposed as signal plus noise, we propose a unified framework which measures the quality of a clustering solution by a trace criterion involving the similarity matrix. The framework provides provable guarantees to do hyperparameter tuning and model selection in these models. More specifically, our contributions can be summarized as below.

1. Our framework can provably tune the following **hyperparameters** in a computationally efficient way *without* the need for CV:

   (a) Lagrange multiplier of the penalty term in a type of semidefinite relaxation for community detection problems in SBM;

   (b) bandwidth parameter used in kernel spectral clustering for sub-gaussian mixture models.

2. We show the same framework incorporating a CV procedure can perform consistent **model selection** (i.e., determining number of clusters):

   (a) when the model selection problem is embedded in the choice of the Lagrange multiplier in another type of SDP relaxation for community detection in SBM;

   (b) general model selection for the Mixed Membership Stochastic Blockmodel (MMSB), which includes the SBM as a sub-model.

We choose to focus on model selection for network-structured data, because there already is an extensive repertoire of empirical and provable methods for i.i.d mixture models including the gap statistic (Tibshirani et al., 2001), silhouette index (Rousseeuw, 1987), the slope criterion (Birgé and Massart, 2001), eigen-gap (Von Luxburg, 2007), penalized maximum likelihood (Leroux, 1992), information theoretic approaches (AIC (Bozdogan, 1987), BIC (Keribin, 2000; Drton and Plummer, 2017), minimum message length (Figueiredo and Jain, 2002)), spectral clustering and diffusion based methods (Maggioni and Murphy, 2018; Little et al., 2017). Next we discuss the related work on models considered in this paper.

1.1 Related Work

Hyperparameters and model selection in network models: In network analysis, while a number of methods exist for selecting the true number of communities (denoted by r) with consistency guarantees including Lei et al. (2016); Wang and Bickel (2017); Le and Levina (2015); Bickel and Sarkar (2016) for SBM, and Fan et al. (2021) for more general models such as the degree-corrected mixed membership blockmodel, these methods have not been generalized to other hyperparameter selection problems. For CV-based methods, existing strategies involve node splitting (Chen and Lei (2018)), or edge splitting (Li et al. (2020)). In the former, it is established that CV prevents underfitting for model selection in SBM. In the latter, a similar one-sided consistency result for Random Dot Product Models (RDPG) (which includes SBM as a special case, see Young and Scheinerman (2007) and a comprehensive survey in Athreya et al. (2017)) is shown. This method has also been empirically applied to tune other hyperparameters, though no provable guarantee was provided.

In terms of algorithms for community detection or clustering, SDP methods have gained a lot of attention (Abbe et al. (2015); Amini et al. (2018); Guédon and Vershynin (2016); Cai et al. (2015); Hajek et al. (2016)) due to their strong theoretical guarantees. Typically, SDP based methods can be divided into two broad categories. The first one maximizes a penalized trace of the product of the adjacency matrix and an unnormalized clustering matrix (see definition in Section 2.2). Here the hyperparameter is the Lagrange multiplier of the penalty term Amini et al. (2018); Cai et al. (2015); Chen and Lei (2018); Guédon and Vershynin (2016). In this formulation, the optimization problem does not need to know the number of clusters. However, it is implicitly required in the final step which obtains the memberships from the clustering matrix.

The other class of SDP methods uses a trace criterion with a normalized clustering matrix (definition in Section 2.2) (Peng and Wei, 2007; Yan and Sarkar, 2019; Mixon et al., 2017). Here the constraints directly use the number of clusters. (Yan et al., 2017) use a penalized alternative of this SDP to do provable model selection for SBMs. However, most of these methods require appropriate tuning of the Lagrange multipliers, which are themselves hyperparameters. Usually the theoretical upper and lower bounds on these hyperparameters involve unknown model parameters, which are nontrivial to estimate. The proposed method in Abbe and Sandon (2015) is agnostic of model parameters, but it involves a highly-tuned and hard to implement spectral clustering step (also noted by Perry and Wein (2017)).

In this paper, we use a SDP from the first class (SDP-1) to demonstrate our provable tuning procedure, and another SDP from the second class (SDP-2) to establish consistency guarantee for our model selection method.

Spectral clustering with mixture models: In statistical machine learning literature, analysis of spectral clustering typically is done in terms of the Laplacian matrix built from an appropriately constructed similarity matrix of the datapoints. There has been much work (Hein et al., 2005; Hein, 2006; Belkin and Niyogi, 2003; Giné and Koltchinskii, 2006) on establishing different forms of asymptotic convergence of the Laplacian. Recently Löffler et al. (2019) have established error bounds for spectral clustering that uses the gram matrix as the similarity matrix. In Srivastava et al. (2019) error bounds are obtained for a variant of spectral clustering for the Gaussian kernel in presence of outliers. However, most of the existing tuning procedures for the bandwidth parameter of the Gaussian kernel are heuristic and do not have provable guarantees. Notable methods include Von Luxburg (2007), who choose an analogous parameter, namely the radius ϵ in an ϵ-neighborhood graph "as the length of the longest edge in a minimal spanning tree of the fully connected graph on the data points." Other discussions on selecting the bandwidth can be found in (Hein et al., 2005; Coifman et al., 2008) and (Schiebinger et al., 2015). Shi et al. (2008) propose a data dependent way to set the bandwidth parameter by suitably normalizing the 95% quantile of a vector containing 5% quantiles of distances from each point.

We now present our problem setup, which applies to both mixture and network models, in Section 2. Section 3 proposes and analyzes our hyperparameter tuning method MATR for networks and sub-gaussian mixtures. Next, in Section 4, we present MATR-CV and the related consistency guarantees for model selection for SBM and MMSB models. Finally, Section 5 contains detailed simulated and real data experiments, and we conclude with paper with a discussion in Section 6.

2. Preliminaries and Notations

2.1 Notations

Let $(C_1, ..., C_r)$ denote a partition of n data points (or nodes in a network) into r clusters; $m_i = |C_i|$ denote the size of C_i. Denote $\pi_{\min} = \min_i m_i/n$. The cluster membership of each data point is represented by a $n \times r$ matrix Z, with $Z_{ij} = 1$ if data point i belongs to cluster j, and 0 otherwise. Since r is the true number of clusters, $Z^T Z$ is full rank. Given Z, the corresponding unnormalized clustering matrix is ZZ^T, and the normalized clustering matrix is $Z(Z^T Z)^{-1} Z^T$. To ease notation, X can be either a normalized or unnormalized clustering matrix, and will be made clear in the context. We use $\tilde{X}$ to denote the matrix returned by SDP algorithms, which may not be a clustering matrix. Denote $\mathcal{X}_r$ as the set of all possible normalized clustering matrices with cluster number r. Let Z_0 and X_0 be the membership and the corresponding normalized clustering matrix from the ground truth. λ is a general hyperparameter; although with a slight abuse of notation, we also use λ to denote the Lagrange multiplier in SDP methods. For any matrix $X \in \mathbb{R}^{n \times n}$, let X_{C_k, C_ℓ} be a matrix such that $X_{C_k, C_\ell}(i, j) = X(i, j)$ if $i \in C_k, j \in C_\ell$, and 0 otherwise. E_n is the $n \times n$ all ones matrix. We write $\langle A, B \rangle = \text{trace}(A^T B)$. Standard notations of $o, O, o_P, O_P, \Theta, \Omega$ will be used. By "with high probability", we mean with probability tending to one.

2.2 Problem setup and the trace criterion

We consider a general clustering setting where the data $\mathcal{D}$ gives rise to a $n \times n$ observed similarity matrix $\hat{S}$, where $\hat{S}$ is symmetric. Denote $\mathscr{A}$ as a clustering algorithm which operates on the data $\mathcal{D}$ with a hyperparameter λ and outputs a clustering result in the form of $\hat{Z}$ or $\hat{X}$. Here note that $\mathscr{A}$, $\hat{Z}$ and $\hat{X}$ could all depend on λ. In this paper, we assume that $\hat{S}$ has the form $\hat{S} = S + R$, where R is a matrix of arbitrary noise, and S is the "population similarity matrix". As we consider different clustering models for network-structured data and iid mixture data, it will be made clear what $\hat{S}$ and S are in each context.

Assortativity (weak and strong): In some cases, we require weak assortativity on the similarity matrix S defined as follows. Suppose for data points $i, j \in C_k$, $S_{ij} = a_{kk}$. Define the minimal difference between diagonal term and off-diagonal terms in the same row cluster as

$$p_{\text{gap}} = \min_k \left(a_{kk} - \max_{\substack{i \in C_k, j \in C_\ell \\ \ell \neq k}} S_{ij} \right). \tag{2.1}$$

Weak assortativity requires $p_{\text{gap}} > 0$. This condition is similar to weak assortativity defined for blockmodels (e.g. Amini et al. (2018)). It is mild compared to strong assortativity requiring $\min_k a_{kk} - \max_{\substack{i \in C_k, j \in C_\ell \\ \ell \neq k}} S_{ij} > 0$.

Stochastic Blockmodel (SBM): The SBM is a generative model of networks with community structure on n nodes. By first partitioning the nodes into r classes which leads to a membership matrix Z, the $n \times n$ binary adjacency matrix A is sampled from probability matrix $P_{ij} = Z_i B Z_j^T 1(i \neq j)$. where Z_i and Z_j are the i^{th} and j^{th} row of matrix Z, B is the $r \times r$ block probability matrix. The aim is to estimate node memberships given A. We assume the elements of B have order $\Theta(\rho)$ with $\rho \to 0$ at some rate.

Mixed Membership Stochastic Blockmodel (MMSB): The SBM can be restrictive when it comes to modeling real world networks. As a result, various extensions have been proposed. The mixed membership stochastic blockmodel (MMSB, (Airoldi et al., 2008)) relaxes the requirement on the membership vector Z_i being binary and allows the entries to be in $[0, 1]^r$, such that they sum up to 1 for each i. We will denote this soft membership matrix by Θ.

Under the MMSB model, the $n \times n$ adjacency matrix A is sampled from the probability matrix P with $P_{ij} = \Theta_i B \Theta_j^T 1(i \neq j)$. We use an analogous definition for normalized clustering matrix: $X = \Theta(\Theta^T \Theta)^{-1}\Theta$. Note that this reduces to the usual normalized clustering matrix when Θ is a binary cluster membership matrix.

Mixture of sub-gaussian random variables: Let $Y = [Y_1, \ldots, Y_n]^T$

be a $n \times d$ data matrix. We consider a setting where Y_i are generated from a mixture model with r clusters,

$$Y_i = \mu_a + W_i, \quad \mathbb{E}(W_i) = 0, \quad Cov(W_i) = \sigma_a^2 I, \qquad a = 1, \ldots, r, \tag{2.2}$$

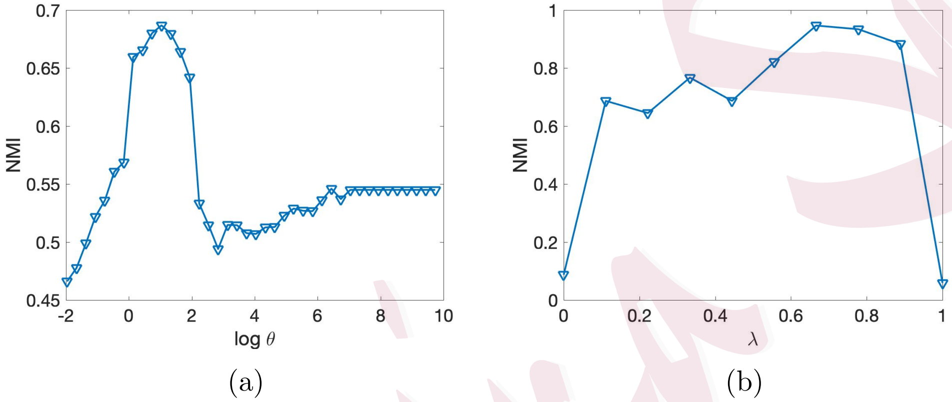
where W_i's are independent sub-gaussian vectors.

Trace criterion: Our framework is centered around the trace $\langle \hat{S}, X_\lambda \rangle$, where X_λ is the normalized clustering matrix associated with hyperparameter λ. This criterion is used in relaxations of the k-means objective (Mixon et al., 2017; Peng and Wei, 2007; Yan et al., 2017) for SDP methods and in evaluating stability of a clustering solution (Meila, 2018).

The idea is that the trace criterion is large when data points within the same cluster are more similar. As a result, this makes the implicit assumption that the similarity matrix $\hat{S}$ (and S) is assortative, i.e., data points within the same cluster have higher similarity based on $\hat{S}$. While this is reasonable for iid mixture models, the SBM or MMSB models may have a mixture of assortative and dis-assortative structure. In what follows, we assume weak assortativity for SBM since our algorithms of interest, SDP methods, operate on weakly assortative networks. For MMSB, which includes SBM as a sub-model, we show the same criterion still works without assortativity if we choose $\hat{S}$ to be A^2 with the diagonal removed.

3. Hyperparameter tuning with known r

In this section, we consider tuning hyperparameters when the true number of clusters r is known. First, we provide two simulation studies to motivate this section. The detailed parameter settings for generating the data can be found in the Supplementary Section S3.1.



(a)

(b)

Figure 1: Tuning hyperparameters in spectral clustering and SDP; accuracy measured by normalized mutual information (NMI). (a) NMI v.s. θ, where θ is the bandwidth parameter in kernel spectral clustering; (b) NMI v.s. λ, where λ is the Lagrange multiplier in SDP-1.

First, we consider a four-component Gaussian mixture model. We perform spectral clustering (k-means on the top r eigenvectors) on the widely used Gaussian kernel matrix (denoted K, where $K(i,j) = \exp\left(-\frac{\|Y_i - Y_j\|_2^2}{2\theta^2}\right)$ for datapoints Y_i and Y_j) with bandwidth parameter θ. Figure 1(a) shows the clustering performance against the ground truth as θ varies using normalized mutual information (NMI), which is a commonly used metric for comparing two partitions of points. The flat region of suboptimal θ reflects when the two adjacent clusters cannot be separated well.

As mentioned in Section 1.1, SDP is an important class of methods for community detection in SBM, but its performance can depend on the choice of the Lagrange multiplier parameter. We consider a SDP formulation (Li et al., 2018), which has been widely used with slight variations in the literature (Amini et al., 2018; Perry and Wein, 2017; Guédon and Vershynin, 2016; Cai et al., 2015; Chen and Lei, 2018),

$$\begin{aligned} \max \quad & \operatorname{trace}(AX) - \lambda \operatorname{trace}(XE_n) \\ \text{s.t.} \quad & X \succeq 0, X \geq 0, X_{ii} = 1 \text{ for } 1 \leq i \leq n, \end{aligned} \tag{SDP-1}$$

where λ is a hyperparameter. Typically, one then performs spectral clustering (that is, k-means on the top r eigenvectors) on the output of the SDP to get the clustering result. In Figure 1 (b), we generate an adjacency matrix from the probability matrix described in Supplementary Section S3.1 and use SDP-1 with tuning parameter λ from 0 to 1. The accuracy of the clustering result is measured by the normalized mutual information (NMI) and shown in Figure 1 (b). We can see that different λ values lead to widely varying clustering performance.

In the general case, we show that when the true cluster number r is known, an ideal hyperparameter λ can be chosen by simply maximizing the trace criterion introduced in Section 2.2. The tuning algorithm (MATR) is presented in Algorithm 1. It takes a general clustering algorithm $\mathscr{A}$, data $\mathcal{D}$ and similarity matrix $\hat{S}$ as input, and outputs a clustering result $\hat{Z}$ depending on λ^* chosen by maximizing the trace criterion.

Algorithm 1: MAx-TRace (MATR) based tuning algorithm for known number of clusters.

```
Input: clustering algorithm $\mathscr{A}$, data $\mathcal{D}$, similarity matrix $\hat{S}$, a set
  of candidates $\{\lambda_1, \cdots, \lambda_T\}$, number of clusters $r$;
Procedure:
for $t = 1 : T$ do
    run clustering on $\mathcal{D}$: $\hat{Z}_t = \mathscr{A}(\mathcal{D}, \lambda_t, r)$;
    compute normalized clustering matrix: $\hat{X}_t = \hat{Z}_t(\hat{Z}_t^T \hat{Z}_t)^{-1} \hat{Z}_t^T$;
    compute inner product: $l_t = \langle \hat{S}, \hat{X}_t \rangle$;
end for
$t^* = \text{argmax}(l_1, ..., l_T)$;
Output: $\hat{Z}_{t^*}$
```

We have the following theoretical guarantee for Algorithm 1.

Theorem 1. *Consider a clustering algorithm $\mathscr{A}$ with inputs $\mathcal{D}, \lambda, r$ and output $\hat{Z}_\lambda$. The similarity matrix $\hat{S}$ used for Algorithm 1(MATR) can be written as $\hat{S} = S + R$. We further assume S is weakly assortative with p_{gap} defined in Eq* (2.1)*, and X_0 is the normalized clustering matrix for the true binary membership matrix Z_0. Let $\pi_{\min}$ be the smallest cluster proportion, and $\tau := n\pi_{min}p_{gap}$. As long as there exists $\lambda_0 \in \{\lambda_1, \ldots, \lambda_T\}$, such that*

$\langle \hat{X}_{\lambda_0}, \hat{S}\rangle \geq \langle X_0, S\rangle - \epsilon$, Algorithm 1 will output a $\hat{Z}_{\lambda^*}$, such that

$$\left\| \hat{X}_{\lambda^*} - X_0 \right\|_F^2 \leq \frac{2}{\tau}(\epsilon + \sup_{X\in\mathcal{X}_r} |\langle X, R\rangle|),$$

where $\hat{X}_{\lambda^*}$ is the normalized clustering matrix associated with $\hat{Z}_{\lambda^*}$.

In other words, as long as the range of λ we consider covers some optimal λ value that leads to a sufficiently large trace criterion (compared with the true underlying X_0 and the population similarity matrix S), the theorem guarantees Algorithm 1 will lead to a normalized clustering matrix with small error. The deviation ϵ depends both on the noise matrix R and how close the estimated $\hat{X}_{\lambda_0}$ is to the ground truth X_0, i.e. the performance of the algorithm. To better interpret this trace lower bound, if we take $\epsilon = \langle \hat{X}_{\lambda_0} - X_0, S\rangle + \sup_{X\in\mathcal{X}_r} |\langle X, R\rangle|$, then the lower bound on the trace is automatically satisfied. The solution found by Algorithm 1 is then bounded by

$$\left\| \hat{X}_{\lambda^*} - X_0 \right\|_F^2 \leq \frac{2}{\tau}\left(\langle \hat{X}_{\lambda_0} - X_0, S\rangle + 2\sup_{X\in\mathcal{X}_r} |\langle X, R\rangle|\right). \tag{3.3}$$

In the bound, the second term is noise, while the first term measures the quality of the clustering solution at an ideal λ_0. If both terms are small, the output from MATR will be close to X_0. Later for specific models, we will give more details on how to interpret the first term. The proof of the theorem is in Supplementary Section S1.1.

In what follows, we apply MATR to more specific settings, namely to select the bandwidth parameter in spectral clustering for sub-gaussian mixtures and the Lagrange multiplier parameter in SDP-1 for SBM.

3.1 Hyperparameter tuning for mixtures of sub-gaussians

In this case, the data $\mathcal{D}$ is Y defined in Eq (2.2), the clustering algorithm $\mathscr{A}$ is spectral clustering (see the motivating example in Section 3) on the Gaussian kernel $K(i,j) = \exp\left(-\frac{\|Y_i - Y_j\|_2^2}{2\theta^2}\right)$. Note that one could use the similarity matrix as the kernel itself. However, this makes the trace criterion a function of the hyperparameter we are trying to tune, which compounds the difficulty of the problem. For simplicity, we use the negative squared distance matrix as $\hat{S}$, i.e. $\hat{S}_{ij} = -\|Y_i - Y_j\|_2^2$. The natural choice for S would be the conditional expectation of $\hat{S}$ given the cluster memberships, which is blockwise constant. However, this choice would lead to a suboptimal error rate. Therefore we use a slightly corrected variant of the matrix as S (also see (Mixon et al., 2017)), called the reference matrix:

$$S_{ij} = -\frac{d_{ab}^2}{2} - \max\left\{0, \frac{d_{ab}^2}{2} + 2(W_i - W_j)^T(\mu_a - \mu_b)\right\} 1(i \in C_a, j \in C_b), \tag{3.4}$$

where $d_{ab} := \|\mu_a - \mu_b\|$, W_i is defined in Eq (2.2). Note that for i, j in the same cluster $S_{ij} = 0$. Interestingly this reference matrix is random itself, which is a deviation from the S used for network models to be discussed below. Applying MATR to select θ, we have the following theoretical guarantee, the proof of which can be found in Supplementary Section S1.2.

Corollary 1. *Let $\hat{S}$ be the negative squared distance matrix, and let S be defined as in Eq 3.4. Let δ_{sep} denote the minimum distance between cluster centers, i.e. $\min_{k\neq\ell} \|\mu_k - \mu_\ell\|$. Denote $\alpha = \pi_{max}/\pi_{min}$. As long as there exists $\theta_0 \in \{\theta_1, \dots, \theta_T\}$, such that $\langle \hat{X}_{\theta_0}, \hat{S}\rangle \geq \langle X_0, S\rangle - n\pi_{min}\epsilon$, Algorithm 1 (MATR) will output a $\hat{Z}_{\theta^*}$, such that w.h.p.*

$$\|\hat{X}_{\theta^*} - X_0\|_F^2 \leq C\frac{\epsilon + r\alpha\sigma_{\max}^2(\alpha + \min\{r, d\})}{\delta_{sep}^2}$$

where $\sigma_{\max}$ is the largest operator norm of the covariance matrices of the mixture components, $\hat{X}_{\theta^*}$ is the normalized clustering matrix for $\hat{Z}_{\theta^*}$ and C is an universal constant.

In this setting, ϵ has to be much smaller than δ_{sep}^2 in order to guarantee small error. As mentioned after Theorem 1, to interpret this trace lower bound involving ϵ, one can set $n\pi_{\min}\epsilon = \langle \hat{X}_{\lambda_0} - X_0, S\rangle + \sup_{X\in\mathcal{X}_r} |\langle X, R\rangle|$, where the second term is absorbed into the noise term in the final bound as usual, and the first term boils down to requiring X_{λ_0} to be close to a computable SDP solution, which is close to X_0 itself. More details and the proof of Corollary 1 can be found in Supplementary Section S1.2.

3.2 Hyperparameter tuning for SBM

We consider choosing λ in SDP-1 for community detection in SBM. Here, the input to MATR – the data $\mathcal{D}$ and similarity matrix $\hat{S}$ – are both the adjacency matrix A. A natural choice of a weakly assortative S is the conditional expectation of A (denoted P) up to diagonal entries, which is blockwise constant. The assortativity condition on S translates naturally to the usual assortativity condition on B, as required by SDP programs. With suitable conditions on the block connectivity separation and estimation error, applying Algorithm 1 (MATR) to tune λ in SDP-1 yields a consistent normalized clustering matrix. For brevity, we defer the detailed statement and proofs to the Supplementary Section S1.3.

4. Hyperparameter tuning with unknown r

In this section, we adapt MATR to situations where the number of clusters is unknown to perform model selection. Similar to Section 3, we first explain the general tuning algorithm and state a general theorem to guarantee its performance. Then applications to specific models will be discussed in the following subsection. Since the applications we focus on are network models, we will present our algorithm with the data $\mathcal{D}$ being A for clarity. We present our algorithm using soft membership matrices Θ, which include binary membership matrices as a special case.

We show that MATR can be extended to model selection if we incorporate a cross-validation (CV) procedure. In Algorithm 2, we present the general MATR-CV algorithm which takes clustering algorithm $\mathscr{A}$, adjacency matrix A, and similarity matrix $\hat{S}$ as inputs. Compared with MATR, MATR-CV has two additional parts.

The first part (Algorithm 3) is to split nodes into two subsets for training and testing. This in turn partitions the adjacency matrix A into four submatrices A^{11}, A^{22}, A^{21} and its transpose, and similarly for $\hat{S}$. MATR-CV makes use of all the submatrices: A^{11} for training, A^{22} for testing, A^{11} and A^{21} for estimating the clustering result for nodes in A^{22} as shown in Algorithm 4, which is the second additional part. Algorithm 4 clusters testing nodes based on the training nodes cluster membership estimated from A^{11} and the connections between training nodes and testing nodes A^{21}, the details of which will be explained as we discuss specific models (see Section 4.1 for MMSB and Supplementary Section S2.3 for SBM).

Like other CV procedures, we note that MATR-CV requires specifying a training ratio γ_{train} and the number of repetitions J. Choosing any $\gamma_{\text{train}} = \Theta(1)$ does not affect our asymptotic results. Repetitions of splits are used empirically to enhance stability; theoretically we show asymptotic consistency for any random split. The general theoretical guarantee and the role of the trace gap Δ are given in the next theorem.

Algorithm 2: MATR-CV.

```
Input: clustering algorithm $\mathscr{A}$,
  adjacency matrix $A$, similarity
  matrix $\hat{S}$, candidates $\{r_1, \cdots, r_T\}$,
  number of repetitions $J$, training
  ratio $\gamma_{\text{train}}$, trace gap $\Delta$;
for $j = 1 : J$ do
    for $t = 1 : T$ do
        $\hat{S}^{11}, \hat{S}^{21}, \hat{S}^{22} \leftarrow$
          NodeSplitting($\hat{S}$, $n$, $\gamma_{\text{train}}$);
        $A^{11}, A^{21}, A^{22} \leftarrow$
          NodeSplitting($A$, $n$, $\gamma_{\text{train}}$);
        $\hat{\Theta}^{11} = \mathscr{A}(A^{11}, r_t)$;
        $\hat{\Theta}^{22} =$ ClusterTest($A^{21}, \hat{\Theta}^{11}$);
        $\hat{X}^{22} = \hat{\Theta}^{22}(\hat{\Theta}^{22^T}\hat{\Theta}^{22})^{-1}\hat{\Theta}^{22^T}$;
        $l_{r_t,j} = \langle \hat{S}^{22}, \hat{X}^{22} \rangle$;
    end for
    $r_j^* = \min\{r_t : l_{r_t,j} \geq$
      $\max_t l_{r_t,j} - \Delta\}$;
end for
$\hat{r} = \text{median}\{r_j^*\}$
Output: $\hat{r}$
```

Algorithm 3: NodeSplitting

```
Input: $A$, $n$,
  $\gamma_{\text{train}}$;
Randomly split $[n]$
  into $Q_1$, $Q_2$ of
  size $n\gamma_{\text{train}}$ and
  $n(1 - \gamma_{\text{train}})$
$A^{11} \leftarrow A_{Q_1,Q_1}$,
  $A^{21} \leftarrow A_{Q_2,Q_1}$,
  $A^{22} \leftarrow A_{Q_2,Q_2}$
Output:
  $A^{11}, A^{21}, A^{22}$
```

Algorithm 4: ClusterTest

```
Input: $A^{21}$, $\hat{\Theta}^{11}$;
  Estimate testing
  node
  memberships
  using $\hat{\Theta}^{11}$ and
  $A^{21}$.
Output: $\hat{\Theta}^{22}$
```

Theorem 2. *Given a candidate set of cluster numbers $\{r_1, \ldots, r_T\}$ containing the true number of cluster r, let $\hat{X}^{22}_{r_t}$ be the normalized clustering matrix obtained from r_t clusters, as described in MATR-CV. Assume the following is true:*

(i) with probability at least $1-\delta_{under}$, $\max_{r_t<r}\langle \hat{S}^{22}, \hat{X}^{22}_{r_t}\rangle \leq \langle \hat{S}^{22}, X^{22}_0\rangle - \epsilon_{under}$;

(ii) with probability at least $1-\delta_{over}$, $\max_{r<r_t\leq r_T}\langle \hat{S}^{22}, \hat{X}^{22}_{r_t}\rangle \leq \langle \hat{S}^{22}, X^{22}_0\rangle + \epsilon_{over}$;

(iii) for the true r, with probability at least $1-\delta_{est}$, $\langle \hat{S}^{22}, \hat{X}^{22}_{r}\rangle \geq \langle \hat{S}^{22}, X^{22}_0\rangle - \epsilon_{est}$;

(iv) there exists $\Delta > 0$ such that $\epsilon_{est} + \epsilon_{over} \leq \Delta < \epsilon_{under} - \epsilon_{est}$.

Here $\epsilon_{under}, \epsilon_{est}, \epsilon_{over} > 0$. Then with probability at least $1 - \delta_{under} - \delta_{over} - \delta_{est}$, MATR-CV will recover the true r with trace gap Δ.

The proof is deferred to the Supplementary Section S2.

Remark 1. 1. MATR-CV is also compatible with tuning multiple hyperparameters. For example, for SDP-1, if the number of clusters is unknown, then for each $\hat{r}$, we can run MATR to find the best λ for the given $\hat{r}$, followed by running a second level MATR-CV to find the best $\hat{r}$. As long as the conditions in Theorems 1 and 2 are met, $\hat{r}$ and the clustering matrix returned will be consistent.

2. As will be seen in the applications below, the derivations of ϵ_{under} and ϵ_{over} are general and only depend on the properties of $\hat{S}$. On the other hand, ϵ_{est} measures the estimation error associated with the algorithm of interest and depends on its performance.

In what follows, we demonstrate MATR-CV can be applied to do model selection for MMSB, which includes SBM as a sub-model.

4.1 Model selection for MMSB

In this section, we consider model selection for the MMSB model as introduced in Section 2.2 with a soft membership matrix Θ. As an example of estimation algorithm, we consider the SPACL algorithm proposed by Mao et al. (2017), which gives consistent parameter estimation when given the correct r. As mentioned in Section 2.2, a normalized clustering matrix in this case is defined analogously as $X = \Theta(\Theta^T\Theta)^{-1}\Theta^T$ for any Θ. X is still a projection matrix, and $X\mathbf{1}_n = \Theta(\Theta^T\Theta)^{-1}\Theta^T\mathbf{1}_n = \Theta(\Theta^T\Theta)^{-1}\Theta^T\Theta\mathbf{1}_r = \mathbf{1}_n$, since $\Theta\mathbf{1}_r = \mathbf{1}_n$. Following Mao et al. (2017), we consider a Bayesian setting for Θ: each row of Θ, $\Theta_i \sim \text{Dirichlet}(\boldsymbol{\alpha}), \boldsymbol{\alpha} \in \mathbb{R}_+^r$. We assume r, $\boldsymbol{\alpha}$ are all fixed constants. Note that the Bayesian setting here is only for convenience, and can be replaced with equivalent assumptions bounding the eigenvalues of $\Theta^T\Theta$. We also assume there is at least one pure node for each of the r communities for consistent estimation at the correct r.

MATR-CV can be applied to the MMSB model by (i) taking $\hat{S} = A^2 - \text{diag}(A^2)$, $S = P^2 - \text{diag}(P^2)$. This allows us to remove the assortativity requirement on P and replace it with a full rank condition on B, which is commonly assumed in the MMSB literature. The fact that P^2 is always positive semi-definite will be used in the proof. The removal of $\text{diag}(A^2)$ and $\text{diag}(P^2)$ leads to better concentration, since $\text{diag}(A^2)$ is centered around a different mean; (ii) Noting that $P^{12} = \Theta^{11}B(\Theta^{22})^T$, we can view the estimation of Θ^{22} as a regression problem with plug-in estimators of Θ^{11} and B. In Algorithm 4, we use an estimate of the form $\hat{\Theta}^{22} = A^{21}\hat{\Theta}^{11}((\hat{\Theta}^{11})^T\hat{\Theta}^{11})^{-1}\hat{B}^{-1}$, where $\hat{B}$, $\hat{\Theta}^{11}$ are estimated from A^{11}.

We have the following guarantee for $\hat{r}$ returned by MATR-CV.

Theorem 3. *Let A be generated from a MMSB model (see Section 2.2) satisfying $\lambda^*(B) = \Omega(\rho)$, where $\lambda^*(B)$ is the smallest singular value of B. We assume $\sqrt{n\rho}/(\log n)^{1+\xi} \to \infty$ for some arbitrarily small $\xi > 0$. Given a candidate set of $\{r_1, \ldots, r_T\}$ containing r and $r_T = \Theta(1)$, with high probability for large n, MATR-CV returns the true cluster number r if $\Delta = O((n\rho)^{3/2}(\log n)^{1.01})$.*

Proof sketch. We first show w.h.p., the underfitting and overfitting errors in Theorem 2 are $\epsilon_{\text{under}} = \Omega(n^2\rho^2)$, $\epsilon_{\text{over}} = O(n\rho\sqrt{\log n})$. To obtain ϵ_{est}, we show that given the true cluster number, the convergence rate of the parameter estimates for the testing nodes obtained from the regression algorithm is the same as the convergence rate for the training nodes. This leads to $\epsilon_{\text{est}} = O((n\rho)^{3/2}(\log n)^{1+\xi})$. For convenience we pick $\xi = 0.01$. For details, see Section S2.2 of the Supplementary. □

Remark 2. 1. The new choice of S and $\hat{S}$ allows our framework to work for more general B, which can have negative eigenvalues in Theorem 3. If B is positive semi-definite with full rank (an assumption commonly used in many MMSB papers), we can still use A and P as $\hat{S}$ and S respectively. Similar analysis applies and the same type of consistency result holds.

2. Compared with Fan et al. (2021), which consider the more general degree-corrected MMSB model, our result holds for $\rho \to 0$ at a faster rate.

3. A practical note: due to the constant in the estimation error being tedious to determine, in this case we only know the asymptotic order of the gap Δ. As has been observed in many other methods based on asymptotic properties (e.g. Bickel and Sarkar (2016); Lei et al. (2016); Wang and Bickel (2017); Hu et al. (2017)), performing an adjustment for finite samples often

improves the empirical performance. In practice we find that if the constant factor in Δ is too large, we tend to underfit. To guard against this, we note that at the correct r, the trace difference $\delta_{r,r-1} := \langle \hat{S}, \hat{X}_r \rangle - \langle \hat{S}, \hat{X}_{r-1} \rangle$ should be much larger than Δ. We start with $\Delta = (n\rho)^{3/2}(\log n)^{1.01}$ and find $\hat{r}$ by Algorithm 2; if $\delta_{\hat{r},\hat{r}-1}$ is smaller than Δ, we reduce Δ by half and repeat the step of finding r_j^* in Algorithm 2 until $\delta_{\hat{r},\hat{r}-1} > \Delta$. This adjustment is much faster than bootstrap corrections and works well empirically.

4. As an example of applying Algorithm 2 to SBM, we consider a different type of SDP algorithm introduced in Peng and Wei (2007); Yan et al. (2017), where the model selection problem is embedded in the algorithm as a hyperparameter tuning problem. In this case, $\hat{S}$ is simply A itself, and the estimation error ϵ_{est} can achieve zero. The detailed statement and proofs can be found in Section S2.3 of the Supplementary.

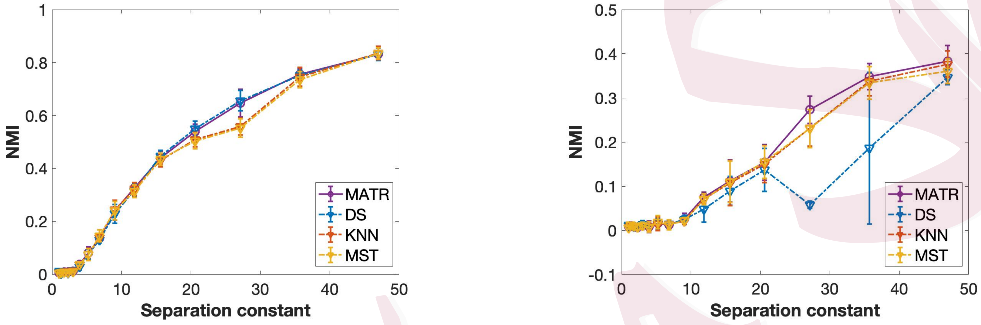
5. Numerical experiments

In this section, we present extensive numerical results on simulated and real data by applying MATR and MATR-CV to different settings considered in Sections 3 and 4.

5.1 MATR with known number of clusters

Spectral clustering for mixture models. We use MATR-CV to select the bandwidth parameter θ in spectral clustering applied to mixture data when given the correct number of clusters. In all the examples, our candidate set of θ is $\{t\alpha/20\}$ for $t = 1, \cdots, 20$ and $\alpha = \max_{i,j} \|Y_i - Y_j\|_2$. We compare MATR with three other well-known heuristic methods. The first one was proposed by (Shi et al., 2008) (DS), where, for each data point Y_i, the 5% quantile of $\{\|Y_i - Y_j\|_2, j = 1, ..., n\}$ is denoted q_i and then θ is set to be $\frac{\text{95\% quantile of } \{q_1,\ldots,q_n\}}{\sqrt{\text{95\% quantile of } \chi^2_d}}$. We also compare with two other methods in Von Luxburg (2007): a method based on k-nearest neighbor (KNN) and a method based on minimal spanned tree (MST). For KNN, θ is chosen in the order of the mean distance of a point to its k-th nearest neighbor, where $k \sim \log(n) + 1$. For MST, θ is set as the length of the longest edge in a minimal spanning tree of the fully connected graph on the data points.

Simulated data. We first conduct experiments on simulated data generated from a 3-component Gaussian mixture with $d = 20$. The means are multiplied by a separation constant which controls clustering difficulty (larger, the easier). Detailed descriptions of the parameter settings can be found in Section S3.2 of the Supplementary. $n = 500$ datapoints are generated for each mixture model and random runs are used to calculate standard deviations for each parameter setting. Figure 2 (a) and (b) show the NMI of different methods against the separation constant for equal and unequal mixing proportions respectively. For all these settings, MATR performs the best or comparably to DS, KNN and MST.

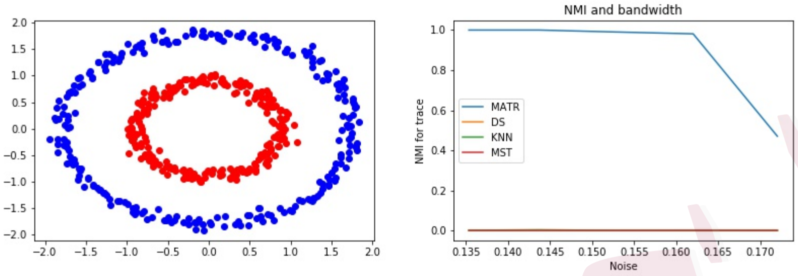


(a)

(b)

Figure 2: Comparison of NMI for tuning bandwidth in spectral clustering for mixture models with (a) equal and (b) unequal mixing coefficients.

To illustrate the robustness of our method on non-Gaussian data, we also apply MATR to tune the bandwidth θ for the two rings dataset (Fig 3 (a)) by setting the similarity matrix $\hat{S}$ to be a RBF kernel to account for nonlinearity. To alleviate the problem that the trace objective is now also dependent on θ via $\hat{S}$, we use a rough guess, e.g., 10^{th} percentile of pairwise distances, in $\hat{S}$. A rough guess here is enough to pick up the right trend. We then apply MATR to select θ in spectral clustering. As seen in Fig 3 (b), MATR outperforms the other methods by a large margin.



(a) The two-ring dataset (b) NMI comparison.

Figure 3: Results on the ring dataset.

Real data. We also test MATR for tuning θ on a real dataset, the Olivetti faces dataset, provided by scikit-learn (Pedregosa et al., 2011). The data consists of 40 classes with 10 examples in each class. We standardize the dataset before clustering. MATR achieves the highest NMI value of 0.83. Both KNN and MST obtain NMI values around 0.82, while DS yields a θ much smaller than the other methods, leading to similarity matrices that are highly unstable when spectral clustering is applied.

Additional results for SBM. We apply MATR to tune λ in SDP-1 for known r and compare with two existing data driven methods (Cai et al. (2015) and Li et al. (2020)) using simulated and real networks. The details can be found in Section S3.3 of the Supplementary.

5.2 Model selection with MATR-CV

MMSB. We compare MATR-CV with Universal Singular Value Thresholding (USVT) (Chatterjee et al., 2015), ECV (Li et al., 2020) and SIMPLE (Fan et al., 2021) in terms of model selection with MMSB. For ECV and MATR-CV, we consider the candidate set $r \in \{1, 2, \cdots, \lfloor \hat{\rho} n \rfloor\}$, where $\hat{\rho} = \sum_{i<j} A_{ij} / \binom{n}{2}$.

Simulated data. We first apply all the methods to simulated data. We consider $B = \rho \times \{(p-q)I_r + qE_r\}$. Following (Mao et al., 2018), we sample $\Theta_i \sim \text{Dirichlet}(\boldsymbol{\alpha})$ and $\boldsymbol{\alpha} = \mathbf{1}_r/r$. We generate networks with $n = 2000$ nodes with $r = 4$ and $r = 8$ respectively. We set $p = 1, q = 0.1$ for $r = 4$ and $p = 1, q = 0.01$ for $r = 8$ for a range of ρ. In Table 1a and 1b, we report the fractions of exactly recovering the true cluster number r over 40 runs for each method across different average degrees. We observe that in both $r = 4$ and $r = 8$ cases, MATR-CV outperforms the other three methods by a large margin on sparse graphs. We find that SIMPLE tends to underfit in our sparsity regime, since their theoretical guarantees hold for a denser degree regime in order to generalize to a broader model than MMSB. An example generated from a non-assortative B can be found in Supplementary Section S3.4.

ρ	0.01	0.02	0.06	0.08	0.11	0.13
MATR-CV	0.35	0.83	0.93	1	1	1
USVT	0	0	1	1	1	1
ECV	0	0	1	0.95	1	1

(a)

ρ	0.02	0.05	0.09	0.12	0.16	0.21
MATR-CV	0.10	0.43	0.95	0.93	0.95	1
USVT	0	0	0.58	1	1	1
ECV	0	0	0	0.93	1	1

(b)

Table 1: Model selection on simulated MMSB. Exact recovery fractions for (a) 4 clusters; (b) 8 clusters.

Real data. We also test MATR-CV with MMSB on a real network, the political books network, which contains 105 nodes in 3 clusters. Here fitting a MMSB model is reasonable since each book can have mixed political inclinations, e.g. a "conserved" book may be in fact mixed between "neutral" and "conservative". With MATR-CV, we found 3 clusters, agreeing with the ground truth. USVT, ECV and SIMPLE found fewer than 3 clusters.

Additional results for SBM. We apply MATR-CV to tune the SDP in Yan et al. (2017) for model selection. Comparisons with existing methods on simulated and real networks can be found in Supplementary Section S3.5.

6. Discussion

Clustering data, both in i.i.d and network structured settings have received a lot of attention both from applied and theoretical communities. However, methods for tuning hyperparameters involved in clustering problems are mostly heuristic. In this paper, we present MATR, a provable MAx-TRace based hyperparameter tuning framework for general clustering problems. We prove the effectiveness of this framework for tuning SDP relaxations for community detection under the block model and for learning the bandwidth parameter of the gaussian kernel in spectral clustering over a mixture of sub-gaussians. Our framework can also be used to do model selection using a cross validation based extension (MATR-CV) which can be used to consistently estimate the number of clusters in blockmodels and mixed membership blockmodels. Using a variety of simulation and real experiments we show the advantage of our method over other existing heuristics.

The framework presented in this paper is general and can be applied to doing model selection or tuning for broader model classes like degree corrected blockmodels (Karrer and Newman, 2011), since there are many exact recovery based algorithms for estimation in these settings (Chen et al., 2018). We believe that our framework can be extended to the broader class of degree corrected mixed membership blockmodels (Jin et al., 2017) which includes the topic model (Mao et al., 2018). However, the derivation of the estimation error ϵ_{est} involves tedious derivations of parameter estimation error, which has not been done by existing works. Furthermore, even though our work uses node sampling, we believe we can extend the MATR-CV framework to get consistent model selection for other sampling procedures like edge sampling (Li et al., 2020).

Supplementary Materials

The Supplementary Materials contains detailed proofs of the main results together with additional theoretical and numerical results.

References

Abbe, E., A. S. Bandeira, and G. Hall (2015). Exact recovery in the stochastic block model. *IEEE Transactions on Information Theory 62*(1), 471–487.

Abbe, E. and C. Sandon (2015). Recovering communities in the general stochastic block model without knowing the parameters. In *Advances in NIPS*, pp. 676–684.

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

Amini, A. A., E. Levina, et al. (2018). On semidefinite relaxations for the block model. *Ann. Statist. 46*(1), 149–179.

Athreya, A., D. E. Fishkind, M. Tang, C. E. Priebe, Y. Park, J. T. Vogelstein, K. Levin,

V. Lyzinski, and Y. Qin (2017). Statistical inference on random dot product graphs: a survey. *J. Mach. Learn. Res. 18*(1), 8393–8484.

Bach, F. R. (2008). Bolasso: model consistent lasso estimation through the bootstrap. In *Proceedings of the 25th international conference on Machine learning*, pp. 33–40. ACM.

Belkin, M. and P. Niyogi (2003, June). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput. 15*(6), 1373–1396.

Bengio, Y. (2000). Gradient-based optimization of hyperparameters. *Neural comput. 12*(8), 1889–1900.

Bergstra, J. and Y. Bengio (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research 13*(Feb), 281–305.

Bergstra, J. S., R. Bardenet, Y. Bengio, and B. Kégl (2011). Algorithms for hyper-parameter optimization. In *Advances in NIPS*, pp. 2546–2554.

Bickel, P. J. and P. Sarkar (2016). Hypothesis testing for automated community detection in networks. *JRSSb 78*(1), 253–273.

Birgé, L. and P. Massart (2001, Aug). Gaussian model selection. *Journal of the European Mathematical Society 3*(3), 203–268.

Bozdogan, H. (1987). Model selection and akaike's information criterion (aic): The general theory and its analytical extensions. *Psychometrika 52*, 345–370.

Breiman, L. et al. (1996). Heuristics of instability and stabilization in model selection. *Ann.*

Statist. 24(6), 2350–2383.

Cai, T. T., X. Li, et al. (2015). Robust and computationally feasible community detection in the presence of arbitrary outlier nodes. *Ann. Statist. 43*(3), 1027–1059.

Chatterjee, S. et al. (2015). Matrix estimation by universal singular value thresholding. *Ann. Statist. 43*(1), 177–214.

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

Chen, Y., X. Li, and J. Xu (2018). Convexified modularity maximization for degree-corrected stochastic block models. *Ann. Statist. 46*(4), 1573–1602.

Coifman, R. R., Y. Shkolnisky, F. J. Sigworth, and A. Singer (2008, October). Graph laplacian tomography from unknown random projections. *Trans. Img. Proc. 17*(10), 1891–1899.

Drton, M. and M. Plummer (2017). A bayesian information criterion for singular models. *JRSSb 79*(2), 323–380.

Fan, J., Y. Fan, X. Han, and J. Lv (2021). Simple: Statistical inference on membership profiles in large networks. *JRSSb to appear*.

Fan, X., Y. Yue, P. Sarkar, and Y. X. R. Wang (2020). On hyperparameter tuning in general clustering problems. In *ICML*.

Fang, Y. and J. Wang (2012). Selection of the number of clusters via the bootstrap method. *Computational Statistics & Data Analysis 56*(3), 468–477.

Feng, J. and N. Simon (2020). An analysis of the cost of hyperparameter selection via split-sample validation, with applications to penalized regression. *Statistica Sinica 30*(1), 511–530.

Figueiredo, M. A. T. and A. K. Jain (2002, March). Unsupervised learning of finite mixture models. *IEEE Transactions on PAMI 24*(3), 381–396.

Friedman, J., T. Hastie, R. Tibshirani, et al. (2001). *The elements of statistical learning*, Volume 1. Springer series in statistics New York.

Giné, E. and V. Koltchinskii (2006). *Empirical graph Laplacian approximation of Laplace–Beltrami operators: Large sample results*, Volume Number 51 of *Lecture Notes–Monograph Series*, pp. 238–259. IMS.

Guédon, O. and R. Vershynin (2016, Aug). Community detection in sparse networks via grothendieck's inequality. *Probability Theory and Related Fields 165*(3), 1025–1049.

Hajek, B., Y. Wu, and J. Xu (2016). Achieving exact cluster recovery threshold via semidefinite programming. *IEEE Transactions on Information Theory 62*(5), 2788–2797.

Hein, M. (2006). Uniform convergence of adaptive graph-based regularization. In *Proceedings of COLT*, COLT'06, Berlin, Heidelberg, pp. 50–64. Springer-Verlag.

Hein, M., J.-Y. Audibert, and U. von Luxburg (2005). From graphs to manifolds – weak and strong pointwise consistency of graph laplacians. In *COLT*, pp. 470–485.

Hu, J., H. Qin, T. Yan, J. Zhang, and J. Zhu (2017). Using maximum entry-wise deviation to test the goodness-of-fit for stochastic block models. *arXiv preprint arXiv:1703.06558*.

Jin, J., Z. T. Ke, and S. Luo (2017). Estimating network memberships by simplex vertex hunting.

Karrer, B. and M. E. J. Newman (2011, Jan). Stochastic blockmodels and community structure in networks. *Phys. Rev. E 83*, 016107.

Keribin, C. (2000, 01). Consistent estimate of the order of mixture models. *Sankhy=a, Series A 62*, 49–66.

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

Lei, J. et al. (2016). A goodness-of-fit test for stochastic block models. *Ann. Statist. 44*(1), 401–424.

Leroux, B. G. (1992, 09). Consistent estimation of a mixing distribution. *Ann. Statist. 20*(3), 1350–1360.

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

Li, X., Y. Chen, and J. Xu (2018). Convex relaxation methods for community detection. *arXiv preprint arXiv:1810.00315*.

Lim, C. and B. Yu (2016). Estimation stability with cross-validation (escv). *Journal of Computational and Graphical Statistics 25*(2), 464–492.

Little, A., M. Maggioni, and J. Murphy (2017, 12). Path-based spectral clustering: Guarantees, robustness to outliers, and fast algorithms. *None*.

Löffler, M., A. Y. Zhang, and H. H. Zhou (2019). Optimality of spectral clustering for gaussian mixture model.

Maggioni, M. and J. M. Murphy (2018). Learning by unsupervised nonlinear diffusion. *ArXiv abs/1810.06702*.

Mao, X., P. Sarkar, and D. Chakrabarti (2017). Estimating mixed memberships with sharp eigenvector deviations. *ArXiv abs/1709.00407*.

Mao, X., P. Sarkar, and D. Chakrabarti (2018). Overlapping clustering models, and one (class) svm to bind them all. In *Advances in Neurips*, pp. 2126–2136.

Meila, M. (2018). How to tell when a clustering is (approximately) correct using convex relaxations. In *Advances in Neural Information Processing Systems*, pp. 7407–7418.

Meinshausen, N. and P. Bühlmann (2010). Stability selection. *JRSSb 72*(4), 417–473.

Mixon, D. G., S. Villar, and R. Ward (2017, 03). Clustering subgaussian mixtures by semidefinite programming. *Information and Inference: A Journal of the IMA 6*(4), 389–415.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research 12*, 2825–2830.

Peng, J. and Y. Wei (2007, February). Approximating k-means-type clustering via semidefinite programming. *SIAM J. on Optimization 18*(1), 186–205.

Perry, A. and A. S. Wein (2017). A semidefinite program for unbalanced multisection in the stochastic block model. In *2017 International Conference on Sampling Theory and Applications (SampTA)*, pp. 64–67. IEEE.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics 20*, 53 – 65.

Schiebinger, G., M. J. Wainwright, and B. Yu (2015, 04). The geometry of kernelized spectral clustering. *Ann. Statist. 43*(2), 819–846.

Shao, J. (1993). Linear model selection by cross-validation. *JASA 88*(422), 486–494.

Shi, T., M. Belkin, and B. Yu (2008). Data spectroscopy: Learning mixture models using eigenspaces of convolution operators. In *Proceedings of the 25th international conference on Machine learning*, pp. 936–943. ACM.

Snoek, J., H. Larochelle, and R. P. Adams (2012). Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pp. 2951–2959.

Srivastava, P. R., P. Sarkar, and G. A. Hanasusanto (2019). A robust spectral clustering algorithm for sub-gaussian mixture models with outliers.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *JRSSb 36*(2), 111–133.

Tibshirani, R., M. Wainwright, and T. Hastie (2015). *Statistical learning with sparsity: the lasso and generalizations*. Chapman and Hall/CRC.

Tibshirani, R., G. Walther, and T. Hastie (2001). Estimating the number of clusters in a data set via the gap statistic. *JRSSb 63*(2), 411–423.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Stat. & comput. 17*(4), 395–416.

Von Luxburg, U. et al. (2010). Clustering stability: an overview. *Foundations and Trends® in Machine Learning 2*(3), 235–274.

Wang, J. (2010). Consistent selection of the number of clusters via crossvalidation. *Biometrika 97*(4), 893–904.

Wang, Y. X. R. and P. J. Bickel (2017, 04). Likelihood-based model selection for stochastic block models. *Ann. Statist. 45*(2), 500–528.

Wasserman, L. (2006). *All of nonparametric statistics*. Springer Science & Business Media.

Wasserman, L. and K. Roeder (2009). High dimensional variable selection. *Annals of statistics 37*(5A), 2178.

Yan, B. and P. Sarkar (2019). Covariate regularized community detection in sparse graphs. *JASA theory and methods*.

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

Yang, Y. et al. (2007). Consistency of cross validation for comparing regression procedures.

Ann. Statist. 35(6), 2450–2473.

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

Zhang, P. (1993). Model selection via multifold cross validation. *Ann. Statist.*, 299–313.