On hyperparameter tuning in general clustering problems

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

Abstract

Tuning hyperparameters for unsupervised learning problems is difficult in general due to the lack of ground truth for validation. However, the success of most clustering methods depends heavily on the correct choice of the involved hyperparameters. Take for example the Lagrange multipliers of penalty terms in semidefinite programming (SDP) relaxations of community detection in networks, or the bandwidth parameter needed in the Gaussian kernel used to construct similarity matrices for spectral clustering. Despite the popularity of these clustering algorithms, there are not many provable methods for tuning these hyperparameters. In this paper, we provide an overarching framework with provable guarantees for tuning hyperparameters in the above class of problems under two different models. Our framework can be augmented with a cross validation procedure to do model selection as well. In a variety of simulation and real data experiments, we show that our framework outperforms other widely used tuning procedures in a broad range of parameter settings.

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 different combinations of plausible hyperparameter values and often has to rely on good heuristics and domain knowledge from the user.

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 (Hastie et al., 2005). 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 & Roeder, 2009). Other notable stability based methods for model selection in similar supervised settings include (Breiman et al., 1996; Bach, 2008; Meinshausen & Bühlmann, 2010; Lim & Yu, 2016). Finally, a large number of empirical methods exist in the machine learning literature for tuning hyperparameters in various training algorithms (Bergstra & 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 different probabilistic clustering models. Here the challenge is two-fold. 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) and Fang & 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. The analysis is done without model assumptions. The second difficulty arises if there is dependence structure in the datapoints, which necessitates careful splitting procedures for CV.

To illustrate the generality of our framework, we focus on

[1]Department of Statistics and Data Sciences, University of Texas at Austin [2]School of Mathematics and Statistics, University of Sydney. Correspondence to: <{xfan, yuguang}@utexas.edu, purna.sarkar@austin.utexas.edu, rachel.wang@sydney.edu.au>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

subgaussian mixtures and the statistical network models like the Stochastic Blockmodel (SBM) as two representative models for i.i.d. data and non i.i.d. data, where clustering is a natural problem. We show that our framework can provably tune **hyperparameters**, including the Lagrange multiplier of the penalty term in a type of semidefinite relaxation (SDP) for community detection problems in SBM, and the bandwidth parameter in kernel spectral clustering for subgaussian mixtures. In addition, the same framework can be used to do consistent **model selection** for both models.

1.1. Related Work

Hyperparameters and model selection in network models: While a number of methods exist for selecting the true number of communities (denoted by r) with consistency guarantees for SBM including Lei et al. (2016); Wang & Bickel (2017); Le & Levina (2015); Bickel & Sarkar (2016), these methods have not been generalized to other hyperparameter selection problems. For CV-based methods, existing strategies involve node splitting (Chen & Lei, 2018), or edge splitting (Li et al., 2016). 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 (Young & Scheinerman (2007), includes SBM as a special case) is shown. This method has also been empirically applied to tune other hyperparameters, though no provable guarantee was provided.

In the area of community detection, SDP-based methods have recently gained much attention. These can be divided into two broad categories. The first involves optimizing a penalized trace criterion (Amini et al., 2018; Cai et al., 2015; Chen & Lei, 2018; Guédon & Vershynin, 2016) over an unnormalized clustering matrix (see Section 2). The optimization problem itself does not need to know r. However, it is implicitly required in the final step which obtains the memberships from the clustering matrix. The second category uses a trace criterion with a normalized clustering matrix (see Section 2) (Peng & Wei, 2007; Yan & Sarkar, 2019; Mixon et al., 2017). Here the constraints involve r. Yan et al. (2017) uses 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. Consistency is typically achieved when the parameters lie within some range which is governed by unknown model parameters. The proposed method in Abbe & Sandon (2015) is agnostic of model parameters, but it involves a highly-tuned and hard to implement spectral clustering step (also noted by Perry & 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.

Hyperparameter tuning and model selection for mixture models: 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.

For model selection, there is an extensive repertoire of empirical and provable methods including the gap statistic (Tibshirani et al., 2001), silhouette index (Rousseeuw, 1987a), the slope criterion (Birgé & Massart, 2001), eigengap (Von Luxburg, 2007), to name a few. We compare our method to a subset of these.

We now present our problem setup in Section 2. Section 3 proposes and analyzes our hyperparameter tuning method MATR for SBM and subgaussian mixtures. In Section 4, we present MATR-CV and the related consistency guarantees for model selection for SBM and subgaussian mixtures. Finally, Section 5 contains detailed simulated and real data experiments and Section 6 concludes the paper.

2. Preliminaries and Notations

2.1. Notations

Let $(C_1, ..., C_r)$ denote a partition of n data points into r clusters; $m_i = |C_i|$ denote the size of C_i. Denote $\pi_{\min} = \min_i m_i/n, \pi_{\max} = \max_i m_i/n$. The cluster membership of each node 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^TZ is full rank. Given Z, the corresponding unnormalized clustering matrix is ZZ^T, and the normalized clustering matrix is $Z(Z^TZ)^{-1}Z^T$. X can be either a normalized or unnormalized clustering matrix, and will be made clear. 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 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 $\langle A, B\rangle = \text{trace}(A^TB)$. 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 as $n \to \infty$.

2.2. Problem setup and motivation

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 $\mathscr{A}$ may or may not perform clustering on $\hat{S}$, and $\mathscr{A}$, $\hat{Z}$ 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): We require weak assortativity on the similarity matrix S defined as follows. Suppose for $i, j \in C_k$, $S_{ij} = a_{kk}$. Define the minimal difference between diagonal term and off-diagonal terms in the same row cluster as

$$p_{\text{gap}} = \min_k \left(a_{kk} - \max_{\substack{i \in C_k, j \in C_\ell \\ \ell \neq k}} S_{ij} \right). \tag{1}$$

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 $P = Z_i B Z_j^T 1(i \neq j)$. 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. Here we take $\hat{S}$ as A, and S as P (up to diagonal entries for self-loops).

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 for high-dimensional mixture model with d growing with n (see e.g. El Karoui et al. (2010); Amini & Razaee (2019)), where Y_i are generated from a mixture model with r clusters,

$$Y_i = \mu_a + \frac{W_i}{\sqrt{d}}, \quad \mathbb{E}(W_i) = 0, \quad Cov(W_i) = \sigma_a^2 I \tag{2}$$

where $a = 1, \ldots, r$, W_i's are independent subgaussian vectors, and this model can be thought of as low dimensional signal embedded in high dimensional noise. Here we take $\hat{S}$ as the negative pairwise distances; the exact forms of $\hat{S}$ and S will be made clear in Section 3.2.

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 often used in relaxations of the k-means objective (Mixon et al., 2017; Peng & Wei, 2007; Yan et al., 2017) in the context of SDP methods. The idea is that the criterion is large when datapoints within the same cluster are more similar. This criterion is also used by Meila (2018) for evaluating stability of a clustering solution, where the author uses SDP to maximize this criterion for each clustering solution. The criterion makes the implicit assumption that $\hat{S}$ (and S) is assortative, i.e. datapoints within the same cluster have high similarity based on $\hat{S}$. This is a reasonable assumption for subgaussian mixtures; for SBM, assortativity is already required by SDP methods for estimation consistency.

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 Supplement Section 10.

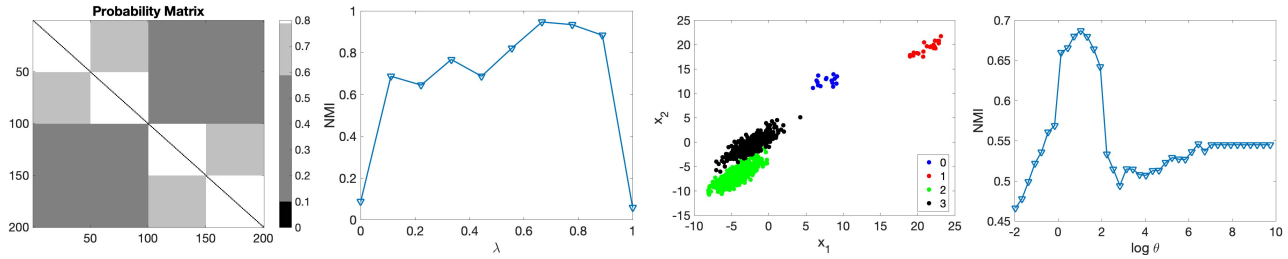
We first consider a SDP formulation (Li et al., 2018) for community detection under SBM, which has been widely used with slight variations in the literature (Amini et al., 2018; Perry & Wein, 2017; Guédon & Vershynin, 2016; Cai et al., 2015; Chen & Lei, 2018),

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

where λ is a hyperparameter. Typically, one then performs spectral clustering (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 shown in Figure 1 (a) 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.

As a second example, we consider a four-component Gaussian mixture model generated data shown in Figure 1 (c). We perform spectral clustering (k-means on the top r eigenvectors) on the widely used Gaussian kernel matrix (denoted K) with bandwidth parameter θ. Figure 1(d) shows the clustering performance using NMI as θ varies, and the flat region of suboptimal θ corresponds to cases when the two adjacent clusters cannot be separated well.

We show that in the case where the true cluster number r is known, an ideal hyperparameter λ can be chosen by simply



(a) SBM (b) NMI v.s. λ (c) Gaussian mixture (d) NMI v.s. θ

Figure 1: Tuning parameters in SDP and spectral clustering; accuracy measured by normalized mutual information.

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 inputs, and outputs a clustering result $\hat{Z}_{\lambda^*}$ with λ^* chosen by maximizing the trace criterion.

Algorithm 1 MAx-TRace (MATR) for known r.

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

$t^* = \text{argmax}(l_1, ..., l_T)$

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* (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 for $\hat{Z}_{\lambda^*}$.

In other words, as long as the range of λ considered 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 R and how close the estimated $\hat{X}_{\lambda_0}$ is to the ground truth X_0, i.e. the algorithm performance. If both ϵ and $\sup_{X \in \mathcal{X}_r} |\langle X, R \rangle|$ are $o_P(\tau)$, then MATR will yield a weakly consistent clustering matrix. The proof is in the Supplement Section 7. Next we apply MATR to select the Lagrange multiplier parameter in SDP-1 for SBM and the bandwidth parameter in spectral clustering for subgaussian mixtures.

3.1. Hyperparameter tuning for SBM

We consider the problem of choosing λ in SDP-1 for community detection in SBM. Here, the input to Algorithm 1 – 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, i.e. P up to diagonal entries: let $\tilde{P} = ZBZ^T$. Note that $\tilde{P}$ is blockwise constant, and assortativity condition on $\tilde{P}$ translates naturally to the usual assortativity condition on B. As the output matrix $\tilde{X}$ from SDP-1 may not necessarily be a clustering matrix, we use spectral clustering on $\tilde{X}$ to get the membership matrix $\hat{Z}$ required in Algorithm 1. SDP-1 together with spectral clustering is used as $\mathscr{A}$.

In Proposition 12 of the Supplement, we show that SDP-1 is strongly consistent, when applied to a general strongly assortative SBM with known r, as long as λ satisfies:

$$\max_{k \neq l} B_{k,l} + \Omega(\sqrt{\rho \log n / n\pi_{\min}}) \leq \lambda \leq \min_k B_{kk} + O(\sqrt{\rho \log n / n\pi_{\max}^2}) \quad (3)$$

An empirical way of choosing λ was provided in (Cai et al., 2015), which we will compare with in Section 5. We show a result complementary to Eq 3 under a SBM model with weakly assortative B, that for a specific region of λ, the normalized clustering matrix from SDP-1 will merge two clusters with high probability. This highlights the importance of selecting an appropriate λ since different values can lead to drastically different clustering result. The detailed statement and proof can be found in Proposition 11 of the Supplement Section 7.2.

When we use Algorithm 1 to tune λ for $\mathscr{A}$, we have the following theoretical guarantee.

Corollary 2. *Consider $A \sim SBM(B, Z_0)$ with weakly assortative B and r number of communities. Denote $\tau := n\pi_{min}\min_k(B_{kk} - \max_{\ell\neq k} B_{k\ell})$. If we have $\epsilon = o_P(\tau)$, $r\sqrt{n\rho} = o(\tau)$, $n\rho \geq c\log n$, for some constant $c > 0$, then as long as there exists $\lambda_0 \in \{\lambda_1, \ldots, \lambda_T\}$, such that $\langle \hat{X}_{\lambda_0}, A\rangle \geq \langle X_0, P\rangle - \epsilon$, with $\mathscr{A}$ Algorithm 1(MATR) will output a $\hat{Z}_{\lambda^*}$, such that $\|\hat{X}_{\lambda^*} - X_0\|_F^2 = o_P(1)$, where $\hat{X}_{\lambda^*}$, X_0 are the normalized clustering matrices for $\hat{Z}_{\lambda^*}$, Z_0 respectively.*

Remark 3. *(i) Since $\lambda \in [0, 1]$, to ensure the range of λ considered overlaps with the optimal range in Eq (3), it suffices to consider λ choices from $[0, 1]$. Then for λ satisfying Eq (3), SDP-1 produces $\tilde{X} = X_0$ w.h.p. if B is strongly assortative. Since $\langle X_0, R\rangle = O_P(r\sqrt{n\rho})$, we can take $\epsilon = O(r\sqrt{n\rho})$, and the conditions in this corollary imply $\frac{r}{\sqrt{n\rho}\pi_{min}} \to 0$. Suppose all the communities are of comparable sizes, i.e. $\pi_{min} = \Theta(1/r)$, then the conditions only require $r = O(\sqrt{n})$ since $n\rho \to \infty$.*

(ii) Since the proofs of Theorem 1 and Corollary 2 are general, the conclusion is not limited to SDP-1 and applies to more general community detection algorithms for SBM when r is known. It is easy to see that a sufficient condition for the consistency of $\hat{X}_{\lambda^*}$ to hold is that there exists λ_0 in the range considered, such that $|\langle \hat{X}_{\lambda_0} - X_0, P\rangle| = o_P(\tau)$.

(iii) We note that the specific application of Corollary 2 to SDP-1 leads to weak consistency of $\hat{X}_{\lambda^*}$ instead of strong consistency as originally proved for SDP-1. This is partly due to the generality of theorem (including the relaxation of strong assortativity on B to weak assortativity) as discussed above, and the fact that we are estimating λ.

3.2. Hyperparameter tuning for mixtures of subgaussians

In this case, the data $\mathcal{D}$ is Y defined in Eq (2), the clustering algorithm $\mathscr{A}$ is spectral clustering (k-means on the top r eigenvectors) on the Gaussian kernel $K(i, j) = \exp\left(-\frac{\|Y_i - Y_j\|_2^2}{2\theta^2}\right)$ and outputs a membership matrix $\hat{Z}$. 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. We use the negative squared distance matrix as $\hat{S}$, i.e. $\hat{S}_{ij} = -\|Y_i - Y_j\|_2^2$. Its population version S is blockwise constant with values $a_{k\ell} = -(d_{k\ell}^2 + \sigma_k^2 + \sigma_\ell^2)$, for $d_{k\ell} = \|\mu_k - \mu_\ell\|_2$. Again we apply MATR to select θ and have the following theoretical guarantee, the proof of which is in Supplement Section 7.4.

Corollary 4. *Consider $\hat{S}$ and S defined above. Assuming S is weakly assortative, denote $\tau := n\pi_{min}\min_k(a_{kk} - \max_{\ell\neq k} a_{k\ell})$. If the following conditions hold,*

$$\epsilon = o_P(\tau), n\sqrt{\log n/d} = o(\tau),$$

then, as long as there exists $\lambda_0 \in \{\lambda_1, \ldots, \lambda_T\}$, such that $\langle \hat{X}_{\lambda_0}, \hat{S}\rangle \geq \langle X_0, S\rangle - \epsilon$, with $\mathscr{A}$, Algorithm 1(MATR) will output a $\hat{Z}_\lambda$, such that $\|\hat{X}_\lambda - X_0\|_F^2 = o_P(1)$, where $\hat{X}_\lambda$ is the normalized clustering matrix for $\hat{Z}_\lambda$.

Remark 5. *The conditions in the corollary are satisfied as long as the spectral clustering algorithm is supplied with an appropriate bandwidth parameter that leads to small error in estimating X_0, and $d/\log n \to \infty$ for fixed $\pi_{\min}$ and $a_{k\ell}$. The existence of such a bandwidth is guaranteed using the results in (Yan & Sarkar, 2016). Also, since we consider low dimensional signal obscured by high dimensional model, it is reasonable to assume that $d_{k\ell}$ (and thus $a_{k\ell}$) is fixed.*

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 algorithm and state a general theorem to guarantee its performance, then apply it to SBM and subgaussian mixture.

Algorithm 2 MATR-CV.

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

Algorithm 3 Splitting

Input: $\hat{S}$, n, γ_{train}
Randomly split $[n]$ into Q_1, Q_2 of size $n\gamma_{\text{train}}$ and $n(1 - \gamma_{\text{train}})$
$\hat{S}^{11} \leftarrow \hat{S}_{Q_1,Q_1}$, $\hat{S}^{21} \leftarrow \hat{S}_{Q_2,Q_1}$, $\hat{S}^{22} \leftarrow \hat{S}_{Q_2,Q_2}$
Output: $\hat{S}^{11}, \hat{S}^{21}, \hat{S}^{22}$

Algorithm 4 ClusterTest

Input: $\hat{S}^{21} \in \{0, 1\}^{n\times m}$, $\hat{Z}^{11} \in \{0, 1\}^{m\times k}$
$M \leftarrow \hat{S}^{21}\hat{Z}^{11}(\hat{Z}^{11T}\hat{Z}^{11})^{-1}$
for $i = 1 : n$ **do**
  $\hat{Z}^{22}(i, \arg\max M(i, :)) = 1$
end
Output: $\hat{Z}^{22}$

We present MATR-CV in Algorithm 2, which augments MATR with a cross-validation (CV) procedure. MATR-CV takes clustering algorithm $\mathscr{A}$ and similarity matrix $\hat{S}$ as inputs. $\mathscr{A}$ directly operates on a similarity matrix.

Algorithm 3 splits $\hat{S}$ into submatrices $\hat{S}^{11}$, $\hat{S}^{22}$, $\hat{S}^{21}$ and its transpose. MATR-CV makes use of all the submatrices: $\hat{S}^{11}$ for training, $\hat{S}^{22}$ for testing, $\hat{S}^{11}$ and $\hat{S}^{21}$ for estimating the clustering result for datapoints in $\hat{S}^{22}$ as shown in Algorithm 4. For each test datapoint, using the estimated membership $\hat{Z}^{11}$ and $\hat{S}^{21}$, we compute average similarities to different clusters of nodes in the training set. Because of our assortativity assumption, $\hat{Z}^{22}$ can be determined by a majority vote.

For the training ratio γ_{train}, as long as $\Theta(1)$, our asymptotic results remain unaffected. 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, with proof deferred to the Supplement Section 8.

Theorem 6. *Given a candidate set of cluster numbers $\{r_1, \dots, r_T\}$ containing the true cluster number r, let $\hat{X}^{22}_{r_t}$ be the normalized clustering matrix obtained from r_t clusters according to MATR-CV. Assume the following:*

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

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

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

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

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.

Remark 7. *(i) 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 6 are met, $\hat{r}$ and the clustering matrix returned will be consistent.*

(ii) 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.

4.1. Model selection for SBM

For model selection, we use the SDP in SDP-2-λ. Here X is a normalized clustering matrix, and in the case of exact recovery $\text{trace}(X)$ is equal to the number of clusters. Since r is implicitly chosen through λ, most of the existing model selection methods with consistency guarantees do not apply directly. Yan et al. (2017) proposed to recover the clustering and r simultaneously, where λ still needs to be empirically selected first. In the Supplement Proposition 18, we show suboptimal choices of λ can lead to merged clusters, which motivates us to choose λ in a systematic way.

$$\begin{aligned} \max_{X} \quad & \text{trace}(AX) - \lambda \text{trace}(X) \\ \text{s.t.} \quad & X \succeq 0, X \ge 0, X\mathbf{1} = \mathbf{1} \end{aligned} \tag{SDP-2-λ}$$

We consider applying MATR-CV to an alternative form of SDP-2-λ as shown in SDP-2, where the cluster number r' appears explicitly in the constraint and is part of the input. SDP-2 returns an estimated normalized clustering matrix, to which we apply spectral clustering to compute the cluster memberships. We name this algorithm $\mathscr{A}_{\text{SDP-2}}$. In this case, we use A as $\hat{S}$, so P is the population similarity matrix.

$$\begin{aligned} \max_{X} \quad & \text{trace}(AX) \\ \text{s.t.} \quad & X \succeq 0, X \ge 0, \text{trace}(X) = r', X\mathbf{1} = \mathbf{1} \end{aligned} \tag{SDP-2}$$

We have the following result ensuring MATR-CV returns a consistent cluster number.

Theorem 8. *Suppose A is generated from a SBM model with r clusters and a weakly assortative B. We assume r is fixed, and $\pi_{\min} \ge \delta > 0$ for some constant δ, and $n\rho / \log n \to \infty$. Given a candidate set of $\{r_1, \dots, r_T\}$ containing true cluster number r and $r_T = \Theta(r)$, with high probability for n large, MATR-CV returns the true number of clusters with $\Delta = (1 + B_{\max})\sqrt{r_{\max} \log n} + B_{\max} r_{\max}$, where $r_{\max} := \arg\max_{r_t} \langle A, \hat{X}_{r_t} \rangle$.*

Proof sketch. We provide a sketch of the proof here, the details can be found in Supplement Section 8.2. We derive the three errors in Theorem 6. In this case, we show that w.h.p., $\epsilon_{\text{under}} = \Omega(n p_{\text{gap}} \pi^2_{\min} / r^2)$, $\epsilon_{\text{over}} = (1 + B_{\max})\sqrt{r_T \log n} + B_{\max} r$, and MATR-CV achieves exact recovery when given the true r, that is, $\epsilon_{\text{est}} = 0$. Since $\epsilon_{\text{under}} \gg \epsilon_{\text{over}}$ under the conditions of the theorem, by Theorem 6, taking $\Delta = \epsilon_{\text{over}}$ MATR-CV returns the correct r w.h.p. Furthermore, we can remove the dependence of Δ on unknown r by noting that $r_{\max} := \arg\max_{r_t} \langle A, \hat{X}_{r_t} \rangle \ge r$ w.h.p., then it suffices to consider the candidate range $\{r_1, \dots, r_{\max}\}$. Thus r_T, r in Δ can be replaced with $r_{\max}$. □

Remark 9. *(i) Although we have assumed fixed r, it is easy to see from the order of ϵ_{under} and ϵ_{over} that the theorem holds for $r^5/n \to 0$, $r^{4.5}\sqrt{\log n}/(n\rho) \to 0$ if we let $\pi_{\min} = \Omega(1/r)$. Many other existing works on SBM model selection assume fixed r. (Lei et al., 2016) considered the regime*

$r = o(n^{1/6})$. *(Hu et al., 2017) allowed r to grow lineary up to a logarithmic factor, but at the cost of making ρ fixed.*

(ii) Asymptotically, Δ is equivalent to $\Delta_{SDP\text{-}2} := \sqrt{r_{\max} \log n}$. We use $\Delta_{SDP\text{-}2}$ in practice when r is fixed.

4.2. Model selection for mixture models

In this subsection, we show that MATR-CV can also recover the number of mixture components in the subgaussian mixture model described in Eq (2) with $\hat{S}$ being the negative squared distance matrix as in Section 3.2. In this case, $\mathcal{A}$ does spectral clustering on $\hat{S}$ directly, which does not contain a bandwidth parameter.

Theorem 10. *Suppose Y is generated from the model in Eq (2). We assume r is fixed, and $\pi_{\min} \geq \delta > 0$ for some constant δ, and $d/\log n \to \infty$. Given a candidate set of $\{r_1, \ldots, r_T\}$ containing true cluster number r and $r_T = \Theta(r)$, with probability tending to one as $n \to \infty$, MATR-CV returns the true number of clusters with $\Delta = n\sqrt{\frac{(\log n)^{1.1}}{d}}$.*

Proof sketch. The proof is analogous to that of Theorem 8. The only difference is that in this case $\hat{S}_{22}$ and $\hat{X}_{22}$ are dependent. However, for the model specified in Eq (2) we have elementwise concentration for $\hat{S}$ around its population counterpart, which alleviates this difficulty. We first show that $\epsilon_{\text{under}} = \Omega(n)$, whereas $\epsilon_{\text{over}} = O(n\sqrt{\log n/d})$. Surprisingly, even though the spectral clustering algorithm is in fact weakly consistent, after the majority voting step in Algorithm 4, we get exact recovery for the test set so $\epsilon_{\text{est}} = 0$. This is similar to the results in (Abbe et al., 2016). The additional $(\log n)^{1.1}$ term is used in the gap so that it is asymptotically of a larger order than ϵ_{over}. □

5. Numerical experiments

Now 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. *More experimental details can be found in Supplement Section 10.*

5.1. MATR on SBM with known r

We apply MATR to tune λ in SDP-1 for known r. Since $\lambda \in [0, 1]$ for SDP-1, we choose $\lambda \in \{0, \cdots, 20\}/20$ in all the examples. For comparison we choose two existing data driven methods. The first method (CL, (Cai et al., 2015)) sets λ as the mean connectivity density in a subgraph determined by nodes with "moderate" degrees. The second is ECV (Li et al., 2016) which uses CV with edge sampling to select the λ giving the smallest loss on the test edges from a model estimated on training edges. We use a training ratio of 0.9 and the L_2 loss throughout.

Simulated data. Consider a strongly assortative SBM as required by SDP-1 for both equal sized and unequal sized clusters. Specifically, we consider the following linkage probability matrix, with two well separated clusters, each of which again have two clusters, thus leading to a hierarchical structure as below:

$$B = \rho \times \begin{bmatrix} 0.8 & 0.6 & 0.3 & 0.3 \\ 0.6 & 0.8 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0.8 & 0.6 \\ 0.3 & 0.3 & 0.6 & 0.8 \end{bmatrix}. \tag{4}$$

For the equal sized case, each cluster has 100 nodes. For the unequal sized case, the first and third clusters have 100 nodes each, while the second and fourth have 50 nodes each. The sparsity parameter ρ ranges from 0.2 to 1. Standard deviations are calculated based on random runs of each parameter setting. We present NMI comparisons for equal sized SBM ($n = 400$, $r = 4$) in Figure 2(A), and unequal sized SBM in Figure 2(B). In both, MATR outperforms others by a large margin as degree grows.

Real data. We compare MATR with ECV and CL on the football (Girvan & Newman, 2002), political books and the political blogs (Adamic & Glance, 2005) datasets. All of them are binary networks with $115, 105$ and 1490 nodes respectively. The clustering performance of each method realtive to ground truth is evaluated by NMI and shown in Table 1. MATR performs the best out of the three methods on the football dataset, and is tied with ECV on the political books dataset. MATR is not as good as CL on the poligical blogs dataset, but still outperforms ECV.

	MATR	ECV	CL
Football	0.924	0.895	0.883
Political blogs	0.258	0.142	0.423
Political books	0.549	0.549	0.525

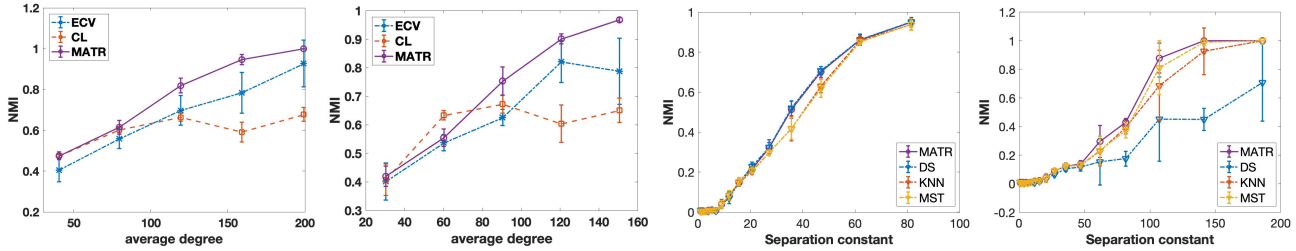
Table 1: Hyperparameter tuning on real data

5.2. MATR on subgaussian mixtures with known r

We use MATR to select the bandwidth parameter θ in spectral clustering applied to data from a gaussian mixture. 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 widely used heuristics. In DS Shi et al. (2008), first 5% quantiles of each node's distance to all other nodes is computed. θ is estimated as a suitably normalized 95% quantile of the previously computed vector. In KNN Von Luxburg (2007), 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 Von Luxburg (2007), 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.

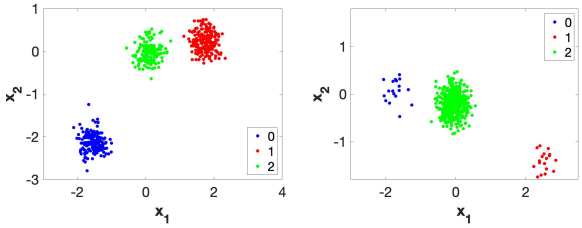
We generate $n = 500$ samples from a 3-component 20



(A) (B) (C) (D)

Figure 2: Comparison of NMI for tuning λ for SDP-1 for equal (A) and unequal sized (B) SBMs. Comparison of NMI for tuning bandwidth in spectral clustering for mixture models with (C) equal and (D) unequal cluster assignment probabilities.

dimensional isotropic Gaussian mixture (each component having identity covariance matrix, see Eq 2). The means are generated from a isotropic Gaussian with covariance $0.01I$. To impose sparsity on the means, we set all but the first two dimensions to zero. To change the level of clustering difficulty, we multiply the means with a separation constant c (larger c corresponding to larger separation and easier clustering). We vary c from 0 to 200. For Figure 2 (c), the probabilities of cluster assignment are equal, while for Figure 2 (d), each point belongs to one of the three clusters with probability $(0.9, 0.05, 0.05)$. 2D projections of the datapoints for the two settings are shown in Figure 3.

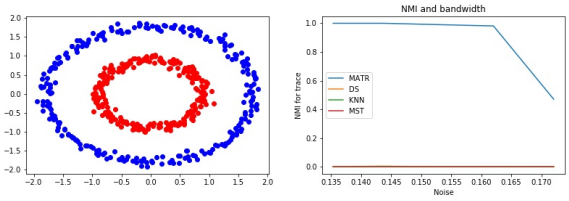


(a) Equal sized clusters (b) Unequal sized clusters

Figure 3: 2D projections of the datapoints for Gaussian mixtures.

We report the mean and standard error of NMI over multiple random runs. In Figure 2 (C) and (D) we plot NMI on the Y axis against the separation along the X axis for mixture models with equal and unequal mixture proportions, respectively. For all these settings, MATR performs as well or better than the best among DS, KNN and MST.

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 4 (a)) by setting the similarity matrix $\hat{S}$ to be a RBF kernel to account for nonlinearity. This is problematic since it makes the trace objective dependent on θ via $\hat{S}$ as well as $\hat{X}$. To alleviate this, for $\hat{S}$ we use a rough guess, e.g., 10^{th} percentile of pairwise distances, because a rough guess is enough to pick up the right trend. We then apply MATR to select θ in spectral clustering. As seen in Fig 4 (b), MATR outperforms the other methods by a large margin.



(a) two rings dataset (b) NMI comparison.

Figure 4: Results on ring dataset.

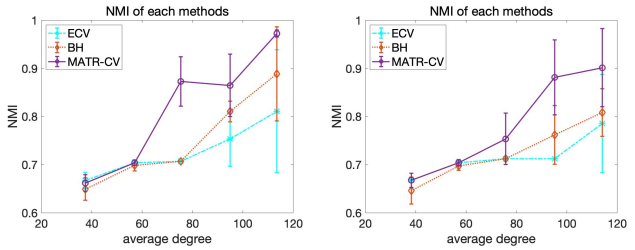
Real data. We tune θ for spectral clustering on the test set provided by (Pedregosa et al., 2011) of the Optical Recognition of Handwritten Digits Data Set with $n = 1797$ and $r = 10$. The clustering done with tuning using MATR, DS, KNN and MST achieve NMI values of 0.64, 0.45, 0.64 and 0.62 respectively. Thus, MATR performs similarly to KNN but outperforms DS and MST. A visual comparison of the clustering results can be found in Supplement Section 10.

5.3. Model selection for SBM

We make comparisons among MATR-CV, Bethe-Hessian estimator (BH) (Le & Levina, 2015) and ECV (Li et al., 2016). For ECV and MATR-CV, we consider $r \in \{1, \cdots \sqrt{n}\}$.

Simulated data. We simulate networks from a 4-cluster assortative SBM with equal and unequal sized blocks. We use a B matrix similar to Eq 4 (details in the Supplement). We select 5 sparsity parameters ρ from 0.2 to 0.6 with even spacing in Fig 5. In Figure 5, we show NMI on Y axis vs. average degree on Y axis. In Figure 5(a) and (b) we respectively consider equal sized (4 clusters of size 100) and unequal sized networks (two with 120 nodes and two with 80 nodes). In all cases, MATR-CV has the highest NMI.

Table 5 in Section 10 of the Supplement shows median number of clusters selected by each method.



(a) NMI for equal sized case (b) NMI for unequal sized case

Figure 5: Comparison of NMI with model selection for equal and unequal sized cases.

Real data. For model selection, we compare MATR-CV, ECV and BH on the same three real network datasets as before. The results are shown in Table 2, where MATR-CV finds the ground truth for the football dataset. On the other two datasets, none of the methods can estimate the r correctly.

	Truth	MATR-CV	ECV	BH
Football	12	12	10	10
Polblogs	2	6	1	8
Polbooks	3	6	2	4

Table 2: Model selection on real networks.

5.4. Model selection for subgaussian mixtures

For model selection experiments on mixture model, we compare MATR-CV, with the Gap statistics (Tibshirani et al., 2001) (GAP) and Silhouette score (Rousseeuw, 1987b) (SIL). For all methods, we use spectral clustering directly on the negative squared distance matrix to do clustering.

Simulated data. We follow the same simulation setting as in Section 5.2 but with $r = 4$. In Table 3, we report the fractions of finding the true cluster number for each method on mixture model with unequal mixing probabilities for different separation constants. MATR-CV outperforms the other two methods by a large margin for not well-separated cases, where GAP and SIL tend to underfit. For mixture model with equal mixing probabilities, MATR-CV performs similarly as GAP but better than SIL, and the results can be found in Supplement Table 6.

separation	1	1.5	2.2	3.3	5.0
MATR-CV	0.2	1	1	1	1
GAP	0	0.7	1	1	1
SIL	0	0	0	0	1

Table 3: Exact recovery fractions for unbalanced 4 clusters

Real data. We apply MATR-CV to the Avila dataset [1] with 10430 data points, 12 clusters and 10 attributes. The dataset is extracted from images of the 'Avila Bible' for copyist identification, which correspond to the different clusters. As shown in Table 4, MATR-CV picks the number of clusters closest to the ground truth. For all the methods, we set the maximal number of clusters to be square root of the dataset size. Because of the scale of Avila dataset, we apply a hierarchical searching strategy to reduce running time. More specifically, we first run a coarse grid search ($K_{coarse} = 10, 20, \cdots, 100$), then pick the $\hat{K}_{coarse}$ with largest trace and conduct a finer grid search between $\hat{K}_{coarse} - 10$ and $\hat{K}_{coarse} + 10$. MATR-CV takes around 2 hours to complete while SIL takes around 7 hours and GAP takes around 30 hours to finish on an single node of two Xeon E5-2690 v3 with 24 cores.

	Truth	MATR-CV	GAP	SIL
Avila	12	11	2	2

Table 4: Number of clusters selected by different methods

6. Concluding remarks

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 under SBM and for learning the bandwidth parameter of the gaussian kernel in spectral clustering on subgaussian mixtures. 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 both models. 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 more general models like degree corrected blockmodels (Karrer & Newman, 2011), since there are many exact recovery based algorithms for estimation in these settings (Chen et al., 2018).

Acknowledgements

YXRW was partially supported by the ARC DECRA fellowship. PS was partially supported by NSF DMS 1713082.

[1] https://archive.ics.uci.edu/ml/datasets/Avila

References

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

Abbe, E., Bandeira, A. S., and Hall, G. Exact recovery in the stochastic block model. *IEEE Transactions on Information Theory*, 62(1):471–487, Jan 2016. ISSN 1557-9654. doi: 10.1109/TIT.2015.2490670.

Adamic, L. A. and Glance, N. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pp. 36–43. ACM, 2005.

Amini, A. A. and Razaee, Z. S. Concentration of kernel matrices with application to kernel spectral clustering. *arXiv preprint arXiv:1909.03347*, 2019.

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

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

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

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

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

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

Birgé, L. and Massart, P. Gaussian model selection. *Journal of the European Mathematical Society*, 3(3): 203–268, Aug 2001. ISSN 1435-9855. doi: 10.1007/s100970100031. URL https://doi.org/10.1007/s100970100031.

Breiman, L. et al. Heuristics of instability and stabilization in model selection. *Ann. Statist.*, 24(6):2350–2383, 1996.

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

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

Chen, Y., Li, X., and Xu, J. Convexified modularity maximization for degree-corrected stochastic block models. *Ann. Statist.*, 46(4):1573–1602, 08 2018. doi: 10.1214/17-AOS1595. URL https://doi.org/10.1214/17-AOS1595.

Coifman, R. R., Shkolnisky, Y., Sigworth, F. J., and Singer, A. Graph laplacian tomography from unknown random projections. *Trans. Img. Proc.*, 17(10):1891–1899, October 2008. ISSN 1057-7149. doi: 10.1109/TIP.2008.2002305. URL https://doi.org/10.1109/TIP.2008.2002305.

El Karoui, N. et al. On information plus noise kernel random matrices. *Ann. Statist.*, 38(5):3191–3216, 2010.

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

Girvan, M. and Newman, M. E. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

Guédon, O. and Vershynin, R. Community detection in sparse networks via grothendieck's inequality. *Probability Theory and Related Fields*, 165(3):1025–1049, Aug 2016. ISSN 1432-2064. doi: 10.1007/s00440-015-0659-z. URL https://doi.org/10.1007/s00440-015-0659-z.

Hastie, T., Tibshirani, R., Friedman, J., and Franklin, J. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2): 83–85, 2005.

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

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

Karrer, B. and Newman, M. E. J. Stochastic blockmodels and community structure in networks. *Phys. Rev. E*, 83:016107, Jan 2011. doi: 10.1103/PhysRevE.83.016107. URL https://link.aps.org/doi/10.1103/PhysRevE.83.016107.

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

Lei, J. and Rinaldo, A. Consistency of spectral clustering in stochastic block models. *Ann. Statist.*, 43(1):215–237, 02 2015. doi: 10.1214/14-AOS1274. URL https://doi.org/10.1214/14-AOS1274.

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

Li, T., Levina, E., and Zhu, J. Network cross-validation by edge sampling. *arXiv preprint arXiv:1612.04717*, 2016.

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

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

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

Meinshausen, N. and Bühlmann, P. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

Mixon, D. G., Villar, S., and Ward, R. Clustering subgaussian mixtures by semidefinite programming. *Information and Inference: A Journal of the IMA*, 6(4): 389–415, 03 2017. ISSN 2049-8764. doi: 10.1093/imaiai/iax001. URL https://doi.org/10.1093/imaiai/iax001.

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

Peng, J. and Wei, Y. Approximating k-means-type clustering via semidefinite programming. *SIAM J. on Optimization*, 18(1):186–205, February 2007. ISSN 1052-6234. doi: 10.1137/050641983. URL http://dx.doi.org/10.1137/050641983.

Perry, A. and Wein, A. S. 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, 2017.

Rousseeuw, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53 – 65, 1987a. ISSN 0377-0427. doi: https://doi.org/10.1016/0377-0427(87)90125-7. URL http://www.sciencedirect.com/science/article/pii/0377042787901257.

Rousseeuw, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987b.

Schiebinger, G., Wainwright, M. J., and Yu, B. The geometry of kernelized spectral clustering. *Ann. Statist.*, 43 (2):819–846, 04 2015. doi: 10.1214/14-AOS1283. URL https://doi.org/10.1214/14-AOS1283.

Shao, J. Linear model selection by cross-validation. *Journal of the American statistical Association*, 88(422):486–494, 1993.

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

Skala, M. Hypergeometric tail inequalities: ending the insanity. *arXiv preprint arXiv:1311.5939*, 2013.

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

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

Tibshirani, R., Walther, G., and Hastie, T. Estimating the number of clusters in a data set via the gap statistic. *JRSSb*, 63(2):411–423, 2001. doi: 10.1111/1467-9868.00293. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00293.

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

Von Luxburg, U. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

von Luxburg, U. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, Dec 2007. ISSN 1573-1375. doi: 10.1007/s11222-007-9033-z. URL https://doi.org/10.1007/s11222-007-9033-z.

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

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

Wang, Y. X. R. and Bickel, P. J. Likelihood-based model selection for stochastic block models. *Ann. Statist.*, 45 (2):500–528, 04 2017. doi: 10.1214/16-AOS1457. URL https://doi.org/10.1214/16-AOS1457.

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

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

Yan, B. and Sarkar, P. On robustness of kernel clustering. In *Advances in Neural Information Processing Systems*, pp. 3098–3106, 2016.

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

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

Yang, Y. et al. Consistency of cross validation for comparing regression procedures. *Ann. Statist.*, 35(6):2450–2473, 2007.

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

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