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, nonparametric 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
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 & Levinia (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 & Le) 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 \( \epsilon \) in an \( \epsilon \)-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 (Birge & Massart, 2001), eigen-gap (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 \( (C_1, \ldots, C_r) \) 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, \( Z^T Z \) is full rank. Given \( Z \), the corresponding unnormalized clustering matrix is \( ZZ^T \), and the normalized clustering matrix is \( Z(Z^T Z)^{-1}Z^T \). \( X \) 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 \( \lambda \) 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. \( \lambda \) is a general hyperparameter; although with a slight abuse of notation, we also use \( \lambda \) 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 \( X_{C_k,C_l}(i,j) = X(i,j) \) if \( i \in C_k, j \in C_l \), 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
We consider a general clustering setting where the data $D$ gives rise to a $n \times n$ observed similarity matrix $\hat{S}$, where $S$ is symmetric. Denote $\mathcal{A}$ as a clustering algorithm which operates on the data $D$ with a hyperparameter $\lambda$ and outputs a clustering result in the form of $\hat{Z}$ or $\hat{X}$. Here note that $\mathcal{A}$ may or may not perform clustering on $S$, and $\mathcal{A}$, $\hat{Z}$ and $\hat{X}$ could all depend on $\lambda$. 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

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

(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 $\min_k a_{kk} - \max_{\ell \neq k} \max_{i \in C_k, j \in C_\ell} S_{ij} > 0$.

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 $P = Z_i B^T Z_j (i \neq j)$, 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,

$$Y_i = \mu_a + \frac{W_i}{\sqrt{d}} \quad E(W_i) = 0, \quad \text{Cov}(W_i) = \sigma^2_a I \quad (2)$$

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 $X_\lambda$ is the normalized clustering matrix associated with hyperparameter $\lambda$. 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).

$$\max \quad \text{trace}(AX) - \lambda \text{trace}(XE_n)$$

s.t. $X \succeq 0, X \succeq 0, X_{ii} = 1$ for $1 \leq i \leq n$, \quad (SDP-1)

where $\lambda$ 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 $\lambda$ 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 $\lambda$ 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 $\theta$. Figure 1 (d) shows the clustering performance using NMI as $\theta$ varies, and the flat region of suboptimal $\theta$ 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 $\lambda$ can be chosen by simply...
Algorithm 1 (MATR) can be written as
\[ \lambda^* = \arg\max_{\lambda} \langle X, \mathbb{E}_X \rangle \] with \( \lambda^* \) chosen by maximizing the trace criterion. The tuning algorithm (MATR) is presented in Algorithm 1.

Consider a clustering algorithm \( \mathcal{A} \), data \( D \) and similarity matrix \( \hat{S} \) as inputs, and outputs a clustering result \( \hat{Z}_\lambda \) by maximizing the trace criterion introduced in Section 2.2. Theorem 1. The tuning algorithm (MATR) is presented in Algorithm 1.

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 \( \lambda \) in [SDP-1] for community detection in SBM. Here, the input to Algorithm 1 – the data \( D \) and similarity matrix \( \hat{S} \) are both the adjacency matrix \( A \). A natural choice of a weakly assortative \( S \) is the conditional expectation of \( X \), i.e. \( P \) up to diagonal entries: let \( \hat{P} = ZBZ^T \). Note that \( \hat{P} \) is blockwise constant, and assortativity condition on \( \hat{P} \) translates naturally to the usual assortativity condition on \( B \). As the output matrix \( \hat{X} \) from [SDP-1] may not necessarily be a clustering matrix, we use spectral clustering on \( \hat{X} \) to get the membership matrix \( \hat{Z} \) required in Algorithm 1. [SDP-1] together with spectral clustering is used as \( \mathcal{A} \).

In Proposition 1.2 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 \( \lambda \) satisfies:

\[
\max_{k \neq i} B_{k,i} + \frac{\Omega(\sqrt{\rho \log n/n\pi}}{n,n\pi_{\min}} \leq \lambda \leq \min_k B_{k,k} + O(\sqrt{\rho \log n/n\pi_{\max}^2})
\] (3)

An empirical way of choosing \( \lambda \) 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 \( \lambda \), the normalized clustering matrix from [SDP-1] will merge two clusters with high probability. This highlights the importance of selecting an appropriate \( \lambda \) since different values can lead to drastically different clustering results. The detailed statement and proof can be found in Proposition 1.1 of the Supplement Section 7.2.

When we use Algorithm 1 to tune \( \lambda \) for \( \mathcal{A} \), 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 $\tau := n\pi_{\min}(B_{kk} - \max_{\ell \neq k} B_{k\ell})$. If we have $\epsilon = o_P(\tau), \frac{\sqrt{n}}{mp} = o(\tau), np \geq c\log n$, for some constant $c > 0$, then as long as there exists $\lambda_0 \in \{\lambda_1, \ldots, \lambda_T\}$, such that $(\hat{X}_{\lambda_0}, A) \geq (X_0, P) - \epsilon$, with $\mathcal{A}$, Algorithm 1 (MATR) will output a $\hat{Z}_X$, such that $\|X_X - X_0\|_2^2 = o_P(1)$, where $X_X$ is the normalized clustering matrix for $Z_0$.

Remark 3. (i) Since $\lambda \in [0, 1]$, to ensure the range of $\lambda$ considered overlaps with the optimal range in Eq (3), it suffices to consider $\lambda$ choices from $[0, 1]$. Then for $\lambda$ satisfying Eq (3), SDP-I produces $\hat{X} = X_0$ w.h.p. if $B$ is strongly assortative. Since $(X_0, P) = O_P(\sqrt{np})$, we can take $\epsilon = O(\sqrt{np})$, and the conditions in this corollary imply $\frac{\sqrt{n}}{mp} = o(\lambda_0)$. Suppose all the communities are of comparable sizes, i.e. $\pi_{\min} = \Theta(1/r)$, then the conditions only require $\epsilon = O(\sqrt{n})$ since $n \rho \rightarrow \infty$.

(ii) Since the proofs of Theorem 4 and Corollary 2 are general, the conclusion is not limited to SDP-I 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 $\lambda_0$ in the range considered, such that $\|X_{\lambda_0} - X_0, P\| = o_P(\tau)$.

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

3.2. Hyperparameter tuning for mixtures of subgaussians

In this case, the data $D$ is $Y$ defined in Eq (2), the clustering algorithm $\mathcal{A}$ is spectral clustering ($k$-means on the top $r$ eigenvectors) on the Gaussian kernel $K(i, j) = \exp\left(-\frac{\|Y_i - Y_j\|_2^2}{2\sigma^2}\right)$ 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 $a_{kk} = -(d_{kk}^2 + \sigma_k^2 + \sigma_i^2)$, for $d_{kk} = \|\mu_k - \mu_k\|_2$. Again we apply MATR to select $\theta$ 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 $\tau := n\pi_{\min}(a_{kk} - \max_{\ell \neq k} a_{k\ell})$. Then, as long as there exists $\lambda_0 \in \{\lambda_1, \ldots, \lambda_T\}$ such that $(\hat{X}_{\lambda_0}, \hat{S}) \geq (X_0, S) - \epsilon$, with $\mathcal{A}$, Algorithm 1 (MATR) will output a $\hat{Z}_X$, such that $\|X_X - X_0\|_2^2 = o_P(1)$, where $X_X$ is the normalized clustering matrix for $\hat{Z}_X$.

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 $\mathcal{A}$, similarity matrix $\hat{S}$, candidates $\{r_1, \ldots, r_T\}$, number of repetitions $J$, training ratio $\gamma_{\text{train}}$, trace gap $\Delta$.

for $j = 1 : J$ do

for $l = 1 : T$ do

$\hat{S}_{11}^{l}, \hat{S}_{21}^{l}, \hat{Z}_{22} \leftarrow \text{NodeSplitting}(\hat{S}, n, \gamma_{\text{train}})$

$\hat{Z}_{11}^{l} = \mathcal{A}(\hat{S}_{11}^{l}, r_l)$

$\hat{Z}_{22} = \text{ClusterTest}(\hat{S}_{21}^{l}, \hat{Z}_{11}^{l})$

$X_{22} = \hat{Z}_{22}(\hat{Z}_{22}^T \hat{Z}_{22})^{-1} \hat{Z}_{22}^T$

$l_{r_l,j} = \langle \hat{S}_{22}^{l}, X_{22}^{l}\rangle$

end

$r_j^* = \min\{r_l : l_{r_l,j} \geq \max_{r_l\neq j} l_{r_l,j} - \Delta\}$

end

$\hat{r} = \text{median}\{r_j^*\}$

Output: $\hat{r}$

Algorithm 3 Splitting

Input: $\hat{S}, n, \gamma_{\text{train}}$

Randomly split $[n]$ into $Q_1, Q_2$ of size $n\gamma_{\text{train}}$ and $n(1 - \gamma_{\text{train}})$

$\hat{S}_{11}^{l} \leftarrow \hat{S}_{Q_1, Q_1}, \hat{S}_{21}^{l} \leftarrow \hat{S}_{Q_2, Q_1}, \hat{S}_{22}^{l} \leftarrow \hat{S}_{Q_2, Q_2}$

Output: $\hat{S}_{11}^{l}, \hat{S}_{21}^{l}, \hat{S}_{22}^{l}$

Algorithm 4 ClusterTest

Input: $\hat{S}_{21}^{l} \in \{0, 1\}^{n \times m}, \hat{Z}_{11}^{l} \in \{0, 1\}^{m \times k}$

$M \leftarrow \hat{S}_{21}^{l} \hat{Z}_{11}^{l T} (\hat{Z}_{11}^{l T} \hat{Z}_{11}^{l})^{-1}$

for $i = 1 : n$ do

$\hat{Z}_{22}^{l}(i, \arg \max M(i,:)) = 1$

end

Output: $\hat{Z}_{22}^{l}$
We present MATR-CV in Algorithm 2 which augments MATR with a cross-validation (CV) procedure. MATR-CV takes clustering algorithm \( \mathcal{A} \) and similarity matrix \( S \) as inputs. \( \mathcal{A} \) directly operates on a similarity matrix.

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

For the training ratio \( \gamma_{\text{train}} \), 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 \( \Delta \) are given in the next theorem, with proof deferred to the Supplement Section 6.

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

(i) with probability at least \( 1 - \delta_{\text{under}} \),
\[ \max_{r_t < r} (S^{22}, X_{r_t}^{22}) \leq (S^{22}, X_0^{22}) - \epsilon_{\text{under}}; \]

(ii) with probability at least \( 1 - \delta_{\text{over}} \),
\[ \max_{r_t \leq r} (S^{22}, X_{r_t}^{22}) \leq (S^{22}, X_0^{22}) + \epsilon_{\text{over}}; \]

(iii) for the true \( r \), with probability at least \( 1 - \delta_{\text{est}} \),
\[ (S^{22}, X_r^{22}) \geq (S^{22}, X_0^{22}) - \epsilon_{\text{est}}; \]

(iv) for each \( r \), with probability at least \( 1 - \delta_{\text{under}} \), \( \Delta < \epsilon_{\text{under}} - \epsilon_{\text{est}}. \)

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

Remark 7. (i) MATR-CV is also compatible with tuning multiple hyperparameters. For example, for Algorithm 3 if the number of clusters is unknown, then for each \( \hat{r} \), we can run MATR to find the best \( \lambda \) 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 2 and 3 are met, \( \hat{r} \) and the clustering matrix returned will be consistent.

(ii) The derivations of \( \epsilon_{\text{under}}, \epsilon_{\text{est}}, \) and \( \epsilon_{\text{over}} \) are general and only depend on the properties of \( S \). On the other hand, \( \epsilon_{\text{est}} \) 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 Algorithm 4. For each test datapoint, using the estimated membership \( X_{r_t}^{11} \) and \( X_{r_t}^{21} \), we compute average similarities to different clusters of nodes in the training set. Because of our assortativity assumption, \( Z^{22} \) can be determined by a majority vote.

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 5. In this case, we show that w.h.p.,
\[ \epsilon_{\text{under}} = \Omega(\frac{\lambda_{\text{gap}}^2}{n \pi_{\text{min}}^2} r^2), \epsilon_{\text{over}} = (1 + B_{\text{max}}^2) \sqrt{n \log n} + B_{\text{max}}r_{\text{max}}, \] and MATR-CV achieves exact recovery when given the true \( r \), that is, \( \epsilon_{\text{est}} = 0. \) Since \( \epsilon_{\text{under}} > \epsilon_{\text{over}} \), MATR-CV returns the true number of clusters with \( \Delta = (1 + B_{\text{max}}) \sqrt{n \log n} + B_{\text{max}}r_{\text{max}}, \) where \( r_{\text{max}} := \arg \max_{r_t} \langle A, X_r \rangle. \)

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 5. In this case, we show that w.h.p.,
\[ \epsilon_{\text{under}} = \Omega(\frac{\lambda_{\text{gap}}^2}{n \pi_{\text{min}}^2} r^2), \epsilon_{\text{over}} = (1 + B_{\text{max}}^2) \sqrt{n \log n} + B_{\text{max}}r_{\text{max}}, \] and MATR-CV achieves exact recovery when given the true \( r \), that is, \( \epsilon_{\text{est}} = 0. \) Since \( \epsilon_{\text{under}} > \epsilon_{\text{over}} \), MATR-CV returns the true number of clusters with \( \Delta = (1 + B_{\text{max}}) \sqrt{n \log n} + B_{\text{max}}r_{\text{max}}, \) where \( r_{\text{max}} := \arg \max_{r_t} \langle A, X_r \rangle. \) Furthermore, we can remove the dependence of \( \Delta \) on unknown \( r \) by noting that \( r_{\text{max}} := \arg \max_{r_t} \langle A, X_r \rangle \geq r \) w.h.p., then it suffices to consider the candidate range \( \{r_1, \ldots, r_{\text{max}}\} \).

Remark 9. (i) Although we have assumed fixed \( r \), it is easy to see from the order of \( \epsilon_{\text{under}} \) and \( \epsilon_{\text{over}} \) that the theorem holds for \( r^5/n \to 0, r^{-4.5} \sqrt{n \log n} (n \pi_{\text{min}}^2) \to 0 \) if we let \( \pi_{\text{min}} = \Omega(1/r) \). Many other existing works on SBM model selection assume fixed \( r \). (Lei et al. 2016) considered the regime
Asymptotically, $\Delta$ is equivalent to $\Delta_{SDP-2} := \sqrt{r_{\text{max}}} \log n$. We use $\Delta_{SDP-2}$ 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 $S$ being the negative squared distance matrix as in Section 3.2. In this case, $A$ does spectral clustering on $S$ 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 $\pi_{\text{min}} \geq \delta > 0$ for some constant $\delta$ 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 $\Delta = n \sqrt{\log n}/d$.

**Proof sketch.** The proof is analogous to that of Theorem 8. The only difference is that in this case $S_{22}$ and $X_{22}$ are dependent. However, for the model specified in Eq (2) we have elementwise concentration for $S$ around its population counterpart, which alleviates this difficulty. We first show that $\epsilon_{\text{under}} = O(n)$, 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 1, 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 $\epsilon_{\text{over}}$.

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 3.3 and 3.4. More experimental details can be found in Supplement Section 10.

5.1. MATR on SBM with known $r$

We apply MATR to tune $\lambda$ in [SDP-1] for known $r$. Since $\lambda \in [0, 1]$ for SDP-1, we choose $\lambda \in \{0, \ldots, 20\}/20$ in all the examples. For comparison we choose two existing data driven methods. The first method (CL (Jia et al., 2015)) sets $\lambda$ 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 $\lambda$ 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.

5.2. MATR on subgaussian mixtures with known $r$

We use MATR to select the bandwidth parameter $\theta$ in spectral clustering applied to data from a subgaussian mixture. In all the examples, our candidate set of $\theta$ is $\{t\alpha/20\}$ for $t = 1, \ldots, 20$ and $\alpha = 0.95$. 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. $\theta$ is estimated as a suitably normalized 95% quantile of the previously computed vector. In KNN (Von Luxburg, 2007), $\theta$ 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), $\theta$ 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.** 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:

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

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 $\rho$ 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 dataset (Girvan & Newman, 2002), political books and the political blogs dataset (Adamic & Glance, 2005) datasets. All of them are binary networks with 115, 105 and 1490 nodes respectively. The clustering performance of each method relative 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 political blogs dataset, but still outperforms ECV.

![Table 1: Hyperparameter tuning on real data](image-url)

We generate $n = 500$ samples from a 3-component 20
Figure 2: Comparison of NMI for tuning $\lambda$ 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 $0.01 I$.

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.

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 $\theta$ 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 $\theta$ via $\hat{S}$ as well as $X$. To alleviate this, for $\hat{S}$ we use a rough guess, e.g., $10^{\text{th}}$ percentile of pairwise distances, because a rough guess is enough to pick up the right trend. We then apply MATR to select $\theta$ in spectral clustering. As seen in Fig 4 (b), MATR outperforms the other methods by a large margin.

Real data. We tune $\theta$ 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 $\rho$ 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.

<table>
<thead>
<tr>
<th></th>
<th>Truth</th>
<th>MATR-CV</th>
<th>ECV</th>
<th>BH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Football</td>
<td>12</td>
<td>12</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Polblogs</td>
<td>2</td>
<td>6</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>Polbooks</td>
<td>3</td>
<td>6</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>separation</th>
<th>1</th>
<th>1.5</th>
<th>2.2</th>
<th>3.3</th>
<th>5.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATR-CV</td>
<td>0.2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>GAP</td>
<td>0</td>
<td>0.7</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SIL</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

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 a single node of two Xeon E5-2690 v3 with 24 cores.

<table>
<thead>
<tr>
<th></th>
<th>Truth</th>
<th>MATR-CV</th>
<th>GAP</th>
<th>SIL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avila</td>
<td>12</td>
<td>11</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

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
On hyperparameter tuning in general clustering problems

References


On hyperparameter tuning in general clustering problems


