EGGS: Eigen-Gap Guided Search For Automated Spectral Clustering

Jicong Fan  Zhao Zhang  Mingbo Zhao  Yiheng Tu  Haijun Zhang

Abstract

The performance of spectral clustering heavily relies on the quality of affinity matrix. A variety of affinity-matrix-construction (AMC) methods have been proposed but they have hyperparameters to determine beforehand, which requires strong experience and lead to difficulty in real applications especially when the inter-cluster similarity is high or the dataset is large. In addition, we often need to choose different AMC methods for different datasets, which still depends on experience. To solve these two challenging problems, in this paper, we present a simple yet effective method for automated spectral clustering. The main idea is to find the most reliable affinity matrix among a set of candidates given by different AMC methods with different hyperparameters, where the reliability is quantified by the relative-eigen-gap of graph Laplacian introduced in this paper. We also implement the method using Bayesian optimization. We extend the method to large-scale datasets such as MNIST, on which the time cost is less than 90s and the clustering accuracy is state-of-the-art. Extensive experiments of natural image clustering show that our method is more versatile, accurate, and efficient than baseline methods.

1. Introduction

Clustering is an important approach to data mining and knowledge discovery. Particularly, spectral clustering (Weiss, 1999; Shi & Malik, 2000; Ng et al., 2002; Von Luxburg, 2007) has superior performance than k-means clustering (Steinhaus et al., 1956), hierarchical clustering (Johnson, 1967), DBSCAN (Ester et al., 1996), and mixtures of probabilistic principal component analyzers (Tipping & Bishop, 1999) in many applications. Roughly speaking, spectral clustering consists of two steps: 1) construct an affinity matrix in which each element denotes the similarity between two data points; 2) perform normalized cut (Shi & Malik, 2000) on the graph corresponding to the affinity matrix. K-nearest neighbors (k-NN) and Gaussian kernel

\[ k(x, y) = \exp(-\|x - y\|^2/(2\varsigma^2)) \]  

are two popular methods to construct affinity matrices, where \( k \) and \( \varsigma \) are hyperparameters to be determined beforehand. As the performance of spectral clustering heavily relies on the quality of affinity matrix, in recent years, a variety of methods have been proposed to construct or learn affinity matrices for spectral clustering. Many of them are in the framework of self-expressive (Roweis & Saul, 2000; Elhamifar & Vidal, 2013) model

\[
\text{minimize} \quad \frac{1}{2} \|X - XC\|^2_F + \lambda R(C). 
\]

Here the columns of \( X \in \mathbb{R}^{m \times n} \) are the data points drawn from a union of subspaces; \( C \in \mathbb{R}^{n \times n} \) is a coefficient matrix; \( R(C) \) denotes a regularization operator on \( C \); \( \lambda \) is a hyperparameter to be determined in advance. Elhamifar & Vidal (2013) proposed to use \( R(C) = \|C\|_1 := \sum_{i=1}^n \sum_{j=1}^n |c_{ij}| \) under a constraint \( \text{diag}(C) = 0 \). Since \( \| \cdot \|_1 \) is a convex surrogate of the number of nonzero elements of matrix, the \( C \) in (2) is encouraged to be sparse. In (Elhamifar & Vidal, 2013), the affinity matrix for spectral clustering is given by \( A = |C| + |C|^\top \). Theoretically, if \( x_i \) and \( x_j \) are from different subspaces, \( a_{ij} \) should be zero. The method is called Sparse Subspace Clustering (SSC) and has shown promising performance in several computer vision problems. Some theoretical results of SSC can be found in (Wang & Xu, 2013; Soltanolkotabi et al., 2014).

Following the self-expressive framework (2), Liu et al. (2013) let \( R(C) = \|C\|_* \) (nuclear norm of \( C \)) and proposed a Low-Rank Representation (LRR) method for subspace clustering. Lu et al. (2012) and (Pan Ji et al., 2014) used the least squares regression (LSR) model, (2) with \( R(C) = \|C\|^2_F \), for subspace clustering. A few variants of LRR and SSC can be found in (Patel & Vidal, 2014; Li & Vidal, 2015; Patel et al., 2015; Shen & Li, 2016; Li & Vidal, 2016; Lu et al., 2018). Recently, deep learning methods were also used to learn affinity matrices for spectral clustering (Ji et al., 2017; Zhang et al., 2019b; Kheirandishfard et al., 2020) and have achieved state-of-the-art performance on many benchmark datasets.

One common limitation of these spectral or subspace clus-
tering methods is that they have at least one\(^1\) hyperparameter to determine in advance\(^2\). Since clustering is an unsupervised learning problem, the hyperparameters cannot be tuned by cross validation that is widely used in supervised learning. Thus, we have to tune the hyperparameters in spectral clustering by experience, which is difficult when the given dataset is quite different from those in our experience or/and the inter-class similarity is high compared to the intra-cluster similarity. Note that SSC, LRR, and their kernel or deep learning extensions (Patel & Vidal, 2014; Ji et al., 2017; Zhang et al., 2019b) have quadratic or even cubic time complexity (per iteration of optimization), which further increases the difficulty of hyperparameter selection in clustering large datasets. On the other hand, different datasets often require different methods for affinity matrix construction, which is hard to tackle by experience.

Based on the above analysis, this paper aims at model and hyperparameter selection for spectral clustering and wants to improve the convenience, accuracy, and efficiency of spectral clustering. Our contributions are as follows.

- We propose an eigen-gap guided search method called EGGS for spectral clustering. The method finds the Laplacian matrix with largest relative-eigen-gap among a set of candidates constructed by different models with different hyperparameters. The search space of the proposed method is very easy to determine and can be arbitrarily large. In addition, the method has low computational cost because there is no iterative optimization when constructing affinity matrices.

- We also implement EGGS via Bayesian optimization. The method can select the possibly best model and optimize the hyperparameters automatically. Note that any AMC methods (e.g. SSC) can be included into the framework of EGGS.

- To cluster large-scale data, we propose to apply EGGS to a set of landmarks of the dataset. We then use neural network to learn a nonlinear mapping from the landmarks to the nullspace of the Laplacian matrix given by EGGS. The nonlinear mapping is used to encode the whole data into a cluster-informative low-dimensional space, in which k-means is performed.

- We provide theoretical analysis for the effectiveness of the proposed methods. For instance, we prove that least squares representation followed by a truncation operation can lead to correct clustering.

We test EGGS on seven benchmark image datasets and show that EGGS is more accurate and efficient than the baselines. In addition, the large-scale extension of EGGS is able to outperform state-of-the-art methods of large-scale subspace or spectral clustering. Most importantly, our method can select model and hyperparameter automatically.

2. Related work

2.1. Large-scale subspace or spectral clustering

A few researchers have developed large-scale subspace clustering methods (Chen & Cai, 2011; Peng et al., 2013; Cai & Chen, 2014; Wang et al., 2014; Peng et al., 2015; You et al., 2016ab; Li & Zhao, 2017; You et al., 2018; Li et al., 2020; Chen et al., 2020; Kang et al., 2021). For instance, You et al. (2016b) proposed to learn the sparse coefficient matrix of SSC by orthogonal matching pursuit (OMP), which greatly reduced the space and time complexity of SSC. Matsushima & Brbic (2019) presented a method called $S^5C$ that performs subset section via sparse representation iteratively and then represents all data points by the subset to construct an affinity matrix for spectral clustering. There is still a lot of room to improve the accuracy, efficiency, and convenience (e.g. hyperparameter tuning) of these methods.

2.2. Exploiting eigenvalue information for clustering

As the number of zero eigenvalues of a Laplacian matrix is equal to the number of connected components of the graph (von Luxburg, 2007), a few researchers took advantage of eigenvalue information in spectral clustering (Meila et al., 2005; Ji et al., 2015; Hu et al., 2017; Lu et al., 2018). For instance, Ji et al. (2015) utilized eigen-gap to determine the rank of the Shape Interaction Matrix. But the method requires to determine another hyperparameter $\gamma$ beforehand and needs to perform spectral clustering for multiple times. The methods of (Meila et al., 2005; Hu et al., 2017; Lu et al., 2018) are based on iterative optimization (need to perform eigenvalue decomposition at every iteration) and hence are not effective in handling large-scale datasets. In addition, the BDR method of (Lu et al., 2018) has two hyperparameters ($\lambda$, $\gamma$) to determine by experience, although it outperformed SSC and LRR on some datasets.

Figure 1 shows the clustering accuracy of SSC (Elhamifar & Vidal, 2013), LRR (Liu et al., 2013), and BDR-B (Lu et al., 2018) with different hyperparameters and our method EGGS on the Extended Yale Face B subset.

\(^1\)In the codes of SSC and its variants provided by their authors, there is usually one more thresholding parameter for affinity matrix, which affects the clustering accuracy a lot. In the deep learning based methods, we need to determine the network structures and regularization parameters, which is much more difficult.

\(^2\)Wang & Xu (2013) and Soltanolkotabi et al. (2014) provided lower and upper bounds for the $\lambda$ in SSC theoretically, which however depend on the unknown noise level.
Figure 1. Clustering accuracies of SSC, LRR, and BDR-B with different hyperparameter λ on the Extended Yale Face B database. The value of λ used in BDR-B has been divided by 10. The γ in BDR-B is chosen from {0.01, 0.1, 1} and the best one is used for each λ. In Case (b), the time costs of SSC, LRR, BDR-B, and EGGS are 9.5s, 33.0s, 7.6s, and 1.6s respectively.

(Kuang-Chih et al., 2005). SSC and BDR-B are sensitive to the value of λ, especially for the relatively difficult task, say Figure 1(b). LRR is not sensitive to the value of λ but its accuracy is low. EGGS is more accurate and efficient than other methods.

2.3. Automated machine learning

Automated model and hyperparameter selection for supervised learning has been extensively studied (Hutter et al., 2019). In contrast, the study for unsupervised learning is very limited. The reason is that in unsupervised learning there is no ground truth or reliable metric to evaluate the performance of algorithms. Concurrently to our work (the first version was done in 2020), Poulakis (2020) also attempted to do automated clustering. Specifically, Poulakis (2020) proposed to use meta-learning to select clustering algorithm and use a heuristic combination of some clustering validity metrics such as Silhouette coefficient (Liu et al., 2010) and Sildbw (Halkidi & Vazirgiannis, 2001) as an objective to maximize via grid search or Bayesian optimization (Jones et al., 1998). One problem is that these metrics are mainly based on Euclidean distance or densities and hence may not be suitable to evaluate the clustering performance of non-distance or non-density based clustering algorithms. Another problem is that there is no unified metric to compare different clustering algorithms.

3. EGGS for Automated Spectral Clustering

3.1. Preliminary Knowledge

Let A ∈ ℝⁿ×ⁿ be an affinity matrix constructed from a given a data matrix X ∈ ℝᵐ×ⁿ. The corresponding graph is denoted by G = (V, E), where V = {v₁, …, vₙ} is the vertex set and E = {e₁, …, eₖ} is the edge set. The degree matrix of a graph G is defined as D = diag(A1), where 1 = [1, 1, …, 1]ᵀ. Our goal is to partition the vertices into k disjoint nonempty subsets C₁, …, Cₖ. Let C = {C₁, …, Cₖ}. It is expected to find a partition C that minimizes the following metric:

**Definition 3.1 (MNCut).** The multiway normalized cut (MNCut) (Meila, 2001) is defined as

\[
\text{MNCut}(C) = \sum_{i=1}^{k} \sum_{j \neq i} \frac{\text{Cut}(C_i, C_j)}{\text{Vol}(C_i)},
\]

where \(\text{Cut}(C_i, C_j) = \sum_{u \in C_i} \sum_{v \in C_j} A_{uv}\) and \(\text{Vol}(C_i)\) denotes the sum of the vertex degrees of \(C_i\).

The normalized graph Laplacian matrix is defined as

\[
L = I - D^{-1/2}AD^{-1/2},
\]

where I is an identity matrix. The normalized graph Laplacian is often more effective than unnormalized one in spectral clustering (some theoretical justification was given by Von Luxburg (Von Luxburg, 2007)). Let σ₁(L) be the i-th smallest eigenvalue of \(L\). The following claim shows the connection between MNCut(C) and \(\sigma_i(L)\).

**Claim 3.2.** The sum of the k smallest singular values of \(L\) quantifies the potential connectivity among \(C_1, \ldots, C_k\):

\[
\text{MNCut}(C) \geq \sum_{i=1}^{k} \sigma_i(L).
\]

The claim can be easily proved by using Lemma 4 of (Meila, 2001). We defer all proof of this paper to the appendices. Because the multiplicity k of the eigenvalue 0 of \(L\) equals to the number of connected components in \(G\) (Von Luxburg, 2007), we expect to construct an affinity matrix \(A\) from \(X\) such that \(L\) has k zero eigenvalues. Thus the optimal partition means \(\text{MNCut}(C) = \sum_{i=1}^{k} \sigma_i(L) = 0\).

3.2. Eigen-Gap Guided Search

In practice, we may construct an \(A\) such that \(\sum_{i=1}^{k} \sigma_i(L)\) is as small as possible because guaranteeing zero eigenvalues is difficult. But this is not enough because \(L\) may have \(k + 1\) or more very small or even zero eigenvalues.

The second smallest eigenvalue of the Laplacian matrix of a graph \(G\) is called the algebraic connectivity of \(G\) (denoted by \(ac(G)\)) (Fiedler, 1973). We have \(ac(G) = 0\) if and only if \(G\) is not connected. When \(G\) has \(k\) disjointed components, there are \(k\) algebraic connectivities, denoted by \(ac(C_1), \ldots, ac(C_k)\). Based on this, we have

**Claim 3.3.** The \(k+1\)-th smallest eigenvalue of \(L\) quantifies the least potential connectivity of partitions \(C_1, \ldots, C_k\) of \(C\):

\[
\min_{1 \leq i \leq k} \text{MNCut}(C_i) \geq \sigma_{k+1}(L).
\]

In other words, \(\sigma_{k+1}(L)\) measures the difficulty in segmenting each of \(C_i\) into two subsets. Hence, when \(\sigma_{k+1}(L)\)
is large, the partitions $C_1, \ldots, C_k$ are stable. Based on Claim 3.2 and Claim 3.3, we may construct an $A$ that has small $\sum_{i=1}^k \sigma_i(L)$ and large $\sigma_{k+1}(L)$ simultaneously, by solving

$$
\begin{align*}
\text{maximize } & \sigma_{k+1}(L) - \frac{1}{k} \sum_{i=1}^k \sigma_i(L), \\
\text{subject to } & L = I - D^{-1/2} A D^{-1/2}, \ A = f_\theta(X),
\end{align*}
$$

where $f_\theta$ denotes a function with parameter $\theta$. We have many choices for $f$. For instance, when considering the sparse self-expression, $f_\theta$ can be implicitly determined by

$$
X = XC, \ \text{diag}(C) = 0, \ A = |C| + |C^\top|, \ \lambda = \theta. \tag{7}
$$

We see that solving (6) directly can be difficult, though one may design an $f_\theta$ that is simpler than (7). On the other hand, in (6), we have to choose $f$ in advance, which requires domain expert or strong experience because different dataset usually needs different $f$. For instance, when one dataset has strong nonlinearity, we may use a nonlinear self-expressive model (Patel & Vidal, 2014).

Note that different $f$ can result in very different distribution of eigenvalues and the small eigenvalues are sensitive to $f$, $\theta$, and noise. Hence the objective in (6) is not effective to compare different $f$ and $\theta$. In this paper, we define a new metric relative-eigen-gap\(^3\) as follows

$$
\text{reg}(L) := \frac{\sigma_{k+1}(L) - \frac{1}{k} \sum_{i=1}^k \sigma_i(L)}{\frac{1}{k} \sum_{i=1}^k \sigma_i(L) + \epsilon}, \tag{8}
$$

where $\epsilon$ is a small constant (e.g. $10^{-6}$) to avoid zero denominator. $\text{reg}(L)$ is not sensitive to the scale of the small eigenvalues. Therefore, instead of (6), we propose to solve

$$
\begin{align*}
\text{maximize } & \text{reg}(L), \\
\text{subject to } & L = I - D^{-1/2} A D^{-1/2}, \ A = f_\theta(X),
\end{align*}
$$

where $\mathcal{F}$ is a set of pre-defined functions and $\Theta$ is a set of hyperparameters. In fact, (9) is equivalent to choosing one $A$ (or $L$) from a set of candidates constructed by different $f$ with different $\theta$, of which the relative-eigen-gap is largest.

The following theorem\(^4\) shows the connection between $\text{reg}(L)$ and the stability of the clustering $C$.

**Theorem 3.4.** Let $C$ and $C'$ be two partitions of the vertices of $G$, where $|C| = |C'| = k$. Define the distance between $C$ and $C'$ as

$$
\text{dist}(C, C') = 1 - \frac{1}{k} \sum_{C_i \in C} \sum_{C_j' \in C'} \frac{(\text{Vol}(C_i \cap C_j'))^2}{\text{Vol}(C_i) \text{Vol}(C_j')}. \tag{10}
$$

Suppose $\eta k \epsilon \geq \sum_{i=1}^k \sigma_i(L) \geq k \epsilon$ and $\text{reg}(L) > (k - 1) \eta / 2$. Let $\delta = \max (\text{MNCut}(C) - \sum_{i=1}^k \sigma_i(L), \text{MNCut}(C') - \sum_{i=1}^k \sigma_i(L))$. Then

$$
\text{dist}(C, C') < \frac{1.5 \delta \epsilon^{-1}}{\text{reg}(L) + (1 - k) \eta / 2}. \tag{11}
$$

It indicates that when $\text{reg}(L)$ is large and $\delta$ is small, the partitions $C$ and $C'$ are close to each other. Thus, the clustering has high stability. When $\sum_{i=1}^k \sigma_i(L) = k \epsilon$, we have

$$
\text{dist}(C, C') < \frac{6 \delta}{\sigma_{k+1}(L) - k \epsilon},
$$

which means the larger $\sigma_{k+1}(L)$ the more stable clustering.

### 3.3. Least Squares Representation Search Space

In (9), we have many choices for $\mathcal{F}$ and $\Theta$. For example, when $\mathcal{F}$ is given by SSC, $\Theta$ can be a set of values for the $\lambda$ in (2), e.g. $\Theta = \{0.01, 0.1, 1\}$. Nevertheless, solving (2) requires iterative optimization and hence is not effective when the dataset is large. We propose to solve

$$
\begin{align*}
\text{minimize } & \frac{1}{2} \|X - XC\|^2_F + \frac{1}{2} \|C\|^2_F, \\
\text{subject to } & X = XC, \ \text{diag}(C) = 0, \ \lambda = \theta.
\end{align*}
$$

of which the closed-form solution is

$$
C = (X^\top X + \lambda I)^{-1} X^\top X. \tag{13}
$$

Let $\text{diag}(C) = 0$ and $C \leftarrow |C|$, the affinity matrix can be constructed as $A = (C + C^\top)/2$. Lu et al. (2012) has shown that the least squares regression (12) is able to identify the subspaces, but determining $\lambda$ still relies on experience and should be selected from $\Theta$. Another problem is that the off-diagonal elements of $A$ are dense (leading to a connected graph), which can result in low clustering accuracy (Lu et al., 2018). Therefore, we propose to truncate $C$ by keeping only the largest $\tau$ elements of each column of $C$. Nevertheless, it is not easy to determine $\tau$ beforehand. When $\tau$ is too small, the corresponding graph will have $k + 1$ or more connected components. When $\tau$ is too large, the corresponding graph will have $k - 1$ or less connected components. We propose to select $\tau$ from a set $\mathcal{T}$ of candidates adaptively under the supervision of our $\text{reg}(L)$. Then the parameter search space we considered is $\Theta \times \mathcal{T}$.

In the case that the data have some low-dimensional nonlinear structures, the similarity between pair-wise columns of $X$ cannot be well recognized by the linear regression

---

\(^3\)Eigen-gap is usually defined as $\sigma_{k+1} - \sigma_k$. In this paper, we use “Eigen-gap” for convenience. On the other hand, $\sigma_k$ is sensitive to noise and an average of $\sigma_1, \ldots, \sigma_k$ is more stable. A comparative study is in Appendix D.5.

\(^4\)This theorem is a modified version of Theorem 1 in (Meila et al., 2005), which is for the eigen-gap $\sigma_{k+1} - \sigma_k$ of $L$. Here we consider $\text{reg}(L)$ instead.
It shows that when two data points in $X$, e.g., $x_i$ and $x_j$, are close to each other, the corresponding two elements in $\hat{c}$, e.g., $\hat{c}_i$ and $\hat{c}_j$, have small difference. Hence (16) with Gaussian kernel utilizes local information to enhance $C$.

Now we see that our search space is
\[
\mathcal{S} = \mathcal{F} \times \Theta \times \mathcal{T},
\]
where $\mathcal{F} = \{(12), (15)\}$, $\Theta = \{\lambda_1, \lambda_2, \ldots\}$, and $\mathcal{T} = \{\tau_1, \tau_2, \ldots\}$. The problem we want to solve becomes
\[
\max_{\mathcal{S}} \text{reg}(L_s).
\]

The whole algorithm of EGGS is shown in Algorithm 1. When $n \ll m$, using the push-through identity (Henderson & Searle, 1981), we reformulate (13) as
\[
C = X^T (\lambda I + XX^T)^{-1} X
\]
to reduce the computational cost from $O(n^3)$ to $O(mn^2)$. In (16), when $n$ is large (e.g., $> 5000$), we perform randomized SVD (Halko et al., 2011) on $K$: $K \approx V_r \Sigma_r V_r^T$. Then
\[
C \approx V_r \Sigma_r^{1/2} (\lambda I + \Sigma)^{-1} \Sigma_r^{-1/2} V_r^T,
\]
where $r = 20k$ works well in practical applications. The time complexity of computing $C$ is $O(r\tau n + r\tau n^2)$. In Line 11 of Algorithm 1, it is equivalent to compute the largest $k+1$ eigenvalues and eigenvectors of $D^{1/2}A D^{-1/2}$, which is sparse. The time complexity is $O(k\tau n)$. The total time complexity of Algorithm 1 (excluding Line 18) is
\[
O \left( |\Theta|(mn^2 + r\tau n + r\tau n^2) + 2|\Theta| |\mathcal{T}|k\tilde{r}n \right),
\]
where $\tilde{r}$ denotes the mean value in $\mathcal{T}$. The time complexity is at most $O(|\Theta|(mn^2 + |\mathcal{T}|kmn))$ when $\tau \leq r \leq m \leq n$. It is worth noting that Algorithm 1 can be easily implemented in parallel, which will reduce the time complexity to $O(\max(m, r) n^2 + kmn)$. On the contrary, SSC, LRR, and their variants require iterative optimization and hence their time complexity is about $O(tm n^2)$, where $t$ denotes the iteration number and is often larger than 100.

Figure 2 shows an intuitive example of the performance of EGGS in clustering a subset of the Extended Yale B database, where for (15) we use Gaussian kernel with $\varsigma = \frac{1}{\sqrt{2}} \sum_{i,j} ||x_i - x_j||$. We see that: 1) in linear regression and kernel regression, for a fixed $\lambda$ (or $\tau$), the $\sigma$ (or $\lambda$) with larger $\text{reg}(L)$ provides higher clustering accuracy; 2) for a fixed $\lambda$ and a fixed $\tau$, if the linear regression has a larger $\text{reg}(L)$, its clustering accuracy is higher than that of kernel regression, and vice versa. We conclude from Figure 2 that a larger $\text{reg}(L)$ indeed leads to a higher clustering accuracy, which is consistent with our theoretical analysis.
A.2
3.7
and
1998
A.3
in Appendix
2013
1
with
Klein et al.
Soltanolkotabi et al.

Figure 2. (Subspace Detection Property). A symmetric affinity matrix \( A \) obtained from \( X \) has subspace detection property if for all \( i \), the nonzero elements of \( a_i \) correspond to the columns of \( X \) in the same subspace as \( x_i \).

For convenience, let \( \pi(i) \) be the index of the subspace \( x_i \) belongs to and \( C_j \) be the index set of the columns of \( X \) in subspace \( j \). We consider the following deterministic model.

**Definition 3.7** (Deterministic Model). The columns of \( X \in \mathbb{R}^{m \times n} \) are drawn from a union of \( k \) different subspaces and are further corrupted by noise, where \( \dim(S_1 \cup \cdots \cup S_k) = d \leq m/2 \leq n/2 \). Let \( X = U \Sigma V^T \) be the SVD of \( X \), where \( \Sigma = \text{diag}(\sigma_1, \ldots, \sigma_n) \) and \( \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \). Let \( \gamma = \sigma_{d+1}/\sigma_d \). Denote \( v_i = (v_{i1}, \ldots, v_{in}) \) the \( i \)-th row of \( V \) and let \( v_i = (v_{i1}, \ldots, v_{id}) \). Suppose the following conditions hold:\( 1) \) for every \( i \in [n] \), the \( \bar{\tau} \)-th largest element of \( \{ \langle v_i^\top j \rangle : j \in \mathcal{C}_{x(i)} \} \) is greater than \( \alpha; \) \( 2) \) \( \max_{i \in [n]} \max_{j \in [n] \setminus \mathcal{C}_{x(i)}} |v_i^\top j| \leq \beta; \) \( 3) \) \( \max_{i,j,l} |v_i^\top v_j^\top v_l| \leq \mu; \) \( 4) \) \( \alpha - \beta > \frac{2\mu d^2}{1+\tau^2} \).

Then the following theorem verifies the effectiveness of (12) followed by the truncation operation in subspace detection.

**Theorem 3.8.** Suppose \( X \) is given by Definition 3.7 and \( C \)

is given by (12) with

\[
\begin{align*}
\lambda &> \left( \Delta (1+\gamma^2)-2\mu m\gamma^2 \right) \frac{1-\sqrt{1-\rho^{-1}}}{4m^2-2\Delta} \sigma_2^2, \\
\lambda &< \left( \Delta (1+\gamma^2)-2\mu m\gamma^2 \right) \frac{1+\sqrt{1-\rho^{-1}}}{4m^2-2\Delta} \sigma_2^2
\end{align*}
\]

(23)

where \( \rho = \left( \Delta (1+\gamma^2)-2\mu m\gamma^2 \right) \frac{1+\sqrt{1-\rho^{-1}}}{4m^2-2\Delta} \) and \( \Delta = \alpha - \beta \). Then the \( A \) given by line 9 of Algorithm 1 with \( \tau \leq \bar{\tau} \) has the subspace detection property.

In Theorem (3.8), the width of the range of \( \lambda \) is \( w = \left( \Delta (1+\gamma^2)-2\mu m\gamma^2 \right) \frac{1}{4m^2-2\Delta} \sigma_2^2 \). We see that larger \( \sigma_d, \Delta \), or smaller \( \gamma, d \) lead to wider range of \( \lambda \), which corresponds to simpler clustering problem. When \( \rho < 1 \) (too small \( \Delta \) and too large \( \gamma \)), \( \lambda \) does not exist. Theorem (3.8) can be extended to the kernel case (15) without the restriction of \( 2d \leq m \) even when the columns of \( X \) are drawn from a union of nonlinear low-dimensional manifolds. See Definition A.1, Definition A.2, and Theorem A.3 in Appendix A.

Based on Theorem (3.8) and Theorem A.3, the following proposition indicates that EGGS can cluster the data correctly.

**Proposition 3.9.** Suppose the affinity matrix \( A \) given by EGGS has the subspace or manifold detection property and \( \text{reg}(L) = \frac{\sigma_{d+1}}{\sigma_d} > 0 \). Then each component of \( G \) consists of all columns of \( X \) in the same subspace or manifold.

### 4. EGGS via Bayesian Optimization

Bayesian optimization (BO) (Jones et al., 1998) has become a promising tool for hyperparameter optimization of supervised machine learning algorithms (Snoek et al., 2012; Klein et al., 2017). Given a black-box function \( g : \mathcal{X} \to \mathbb{R} \), BO aims to find an \( \mathbf{x}^* \in \mathcal{X} \) that globally minimizes \( g \) and usually has three steps. The first step is finding the most promising point \( \mathbf{x}_{t+1} \in \text{argmax}_x a_p(g)(\mathbf{x}) \) by numerical optimization, where \( a_p(g) : \mathcal{X} \to \mathbb{R} \) is an acquisition function (e.g. Expected Improvement) relying on an prior \( p(g) \) (e.g. Gaussian processes (Williams & Rasmussen, 2006)). The second step is evaluating the expensive and possibly noisy function \( y_{t+1} \sim g(\mathbf{x}) + \mathcal{N}(0, \sigma^2) \) and add the new sample \( (\mathbf{x}_{t+1}, y_{t+1}) \) to observation set \( \mathcal{D}_t = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_t, y_t)\} \). The last step is updating \( p(g) \) and \( a_p(g) \) using \( \mathcal{D}_{t+1} \).

As an alternative to the grid search Algorithm 1, we can maximize \( \text{reg}(L_s) \) via BO. Suppose we have a set of different models, \( \mathcal{F} = \{f_1, f_2, \ldots, f_M\} \), for affinity matrix construction. For \( i = 1, 2, \ldots, M \), let \( g_i(s^{(i)}) := -\text{reg}(L(f_i(s^{(i)}), X)) \).
Table 1. Clustering performance on the six small datasets. As SSC-OMP and EGGS are faster than other methods, we run them for 20 times and report the mean values and standard deviations. For the MNIST1k, we report the average results of 20 trials because the subset is formed randomly. EGGS chose linear regression for Yale B and AR and chose kernel regression for others datasets. The NMI results are in Table 4 of Appendix D.3.

| Dataset     | SSC acc time | LRR acc time | EDSC acc time | KSSC acc time | SSC-OMP acc time | BDR-Z acc time | BDR-B acc time | EGGS acc time | EGGS$_BO$ acc time |
|-------------|--------------|--------------|---------------|---------------|------------------|----------------|----------------|---------------|-------------------|
| Yale B      | 0.723 273.8  | 0.643 928.1  | 0.806 58.6    | 0.649 464.3   | 0.765 (0.015)    | 0.596 719      | 0.719 368.8    | 0.897 (0.01)   | 0.909             |
| ORL         | 0.711 2.7    | 0.762 8.8    | 0.712 2.0     | 0.707 2.6     | 0.665 (0.022)    | 0.739 3.9      | 0.735 3.9      | 0.795 (0.011) | 0.803             |
| COIL20      | 0.871 61.8   | 0.729 221.2  | 0.759 15.4    | 0.912 100.6   | 0.658 (0.030)    | 0.713 86.8     | 0.791 86.8     | 0.782 (0.012) | 0.878             |
| AR          | 0.718 317.5  | 0.769 1220.6 | 0.673 69.1    | 0.726 100.6   | 0.669 (0.022)    | 0.745 25.7     | 0.751 25.7     | 0.786 (0.013) | 0.826             |
| MNIST-1k    | 0.596 0.054  | 0.513 0.037  | 0.536 0.035   | 0.577 0.053   | 0.542 0.038      | 0.576 24.9     | 0.578 24.9     | 0.615 0.041    | 0.619             |
| Fashion-MNIST-1k | 0.553 0.025 | 0.515 0.014  | 0.544 0.017   | 0.534 0.016   | 0.566 0.034      | 0.574 2.5      | 0.563 2.5      | 0.581 0.025    | 0.584             |

where $s^{(i)}$ denotes the hyperparameters in $f_1$ and $X$ denotes the data we want to cluster. Then we use BO to find

$$s^{(i)}_{\ast} = \arg\min_{s^{(i)} \in S^{(i)}} g_1(s^{(i)}),$$  \hspace{1cm} (24)

where $S^{(i)}$ denotes the constraints on $f_1$’s hyperparameters. Now we get the best model with its best hyperparameters

$$f_{s_{\ast}}(s^{(s)}_{\ast}|X), \hspace{1cm} \ast = \arg\min_{1 \leq i \leq M} g_1(s^{(i)}).$$  \hspace{1cm} (25)

For convenience, we denote the method by EGGS$_BO$. Taking model (15) with polynomial kernel as an example, the parameters are $s = (\lambda, b, q, \tau)^T$ and the constraints are given by $S = \{\lambda \in \mathbb{R} : \lambda_{\min} \leq \lambda \leq \lambda_{\max}, b \in \mathbb{R}, b_{\min} \leq b \leq b_{\max}, q \in \mathbb{Z}^+, g_{\min} \leq q \leq g_{\max}, \tau \in \mathbb{R}^+ : \tau_{\min} \leq \tau \leq \tau_{\max}\}$. More details about the implementation of EGGS$_BO$ are in Appendix B. Note that $F$ can include any affinity matrix construction methods such as k-NN, kernels, SSC, and even DSC (Li et al., 2017) (see Appendix D.6), though those without closed-form solutions will increase the time cost significantly.

5. EGGS+NSE for Large-Scale Data

Since the time and space complexity of EGGS are quadratic with $n$, it cannot be directly applied to large-scale datasets. To solve the problem, we propose to perform Algorithm 1 on a set of landmarks of the data (denoted by $\hat{X}$) to get $\hat{Z}$. The landmarks can be generated by k-means or randomly. Then we regard $\hat{Z}$ as a feature matrix and learn a map $g : \mathbb{R}^m \to \mathbb{R}^k$ from $X$ to $\hat{Z}$. According to Lines 4-10 and 17 in Algorithm 1, $g$ should be nonlinear and nonsmooth. According to the universal approximation theorem (Sonoda & Murata, 2017) of neural networks, we approximate $g$ by a two-layer neural network and solve

$$\min_{W_1, W_2, b_1, b_2} \frac{1}{2} \| \hat{Z} - W_2 \text{ReLU}(W_1 \hat{X} + b_1 1^n) - b_2 1^n \|_F^2 + \frac{\nu}{2} \left( \| W_1 \|^2_F + \| W_2 \|^2_F \right),$$  \hspace{1cm} (26)

where $W_1 \in \mathbb{R}^{d \times m}$, $W_2 \in \mathbb{R}^{k \times d}$, $b_1 \in \mathbb{R}^d$, and $b_2 \in \mathbb{R}^k$. Since $A$ is sparse, $k$ is often less than $m$, and a neural network is used, we call (26) Neural Sparse Embedding (NSE). We use mini-batch Adam (Kingma & Ba, 2014) to solve NSE. It is worth mentioning that NSE is different from (Li et al., 2020). In (Li et al., 2020), the regression is for an affinity matrix, which led to high computational cost.

The neural network learned from (26) is applied to the whole data $X$ to extract a k-dimensional feature matrix $Z$:

$$Z = \hat{g}(X) = W_2 \text{ReLU}(W_1 X + b_1 1^n) + b_2 1^n.$$

Finally, we perform k-means on $Z$ to get the clusters. The procedures are summarized in Algorithm 2 in Appendix C. The following proposition shows that a small number of hidden nodes are sufficient to make the clustering succeed.

Proposition 5.1. Suppose the columns (with unit $\ell_2$ norm) of $X$ are drawn from a union of k independent subspaces of dimension $r$: $\sum_{j=1}^k \text{dim}(S_j) = \text{dim}(S_1 \cup \cdots \cup S_k) = kr$. For $j = 1, \ldots, k$, let $U^j$ be the bases of $S_j$ and $\bar{x}_i = U^j v_i$, if $x_i \in S_j$. Suppose $\max\{|U_i^j|^2 : 1 \leq l \leq r, 1 \leq i \neq j \leq k\} \leq \mu$. Suppose that for all $i = 1, \ldots, n$, $\max\{|v_i_1, \ldots, v_i_k| > \mu$. Then there exist $W_1 \in \mathbb{R}^{k \times m}$, $W_2 \in \mathbb{R}^{k \times d}$, $b_1 \in \mathbb{R}^d$, and $b_2 \in \mathbb{R}^k$ such that performing k-means on $Z$ given by (27) identifies the clusters correctly, where $d = kr$. 
Table 2. Clustering accuracy (mean value and standard deviation) and time cost (second) on MNIST and Fashion-MNIST. The time cost of S\(^5\)COMP is divided by 20 (the parameter \(T\) in (Chen et al., 2020)) because we haven’t used the parallel algorithm of the method (limited by the RAM). “—” means the computation is out of memory.

|                  | LSC-K | SSCC | SSC-OMP | S\(^5\)C | S\(^5\)COMP-C | EGGS+NSE | EGGS\(_{BO}\)+NSE |
|------------------|-------|------|---------|---------|--------------|---------|-------------------|
| MNIST-10k        | acc   | time |         |         |              |         |                   |
|                  | 0.652 (0.037) | 18.9 | 0.529 (0.031) | 31.2 | 0.431 (0.014) | 26.4 | 0.646 (0.045) | 82.3 | 0.623 (0.028) | 280.6 | 0.687 (0.035) | 710.4/20 | 16.2 |
| MNIST            | acc   | time |         |         |              |         |                   |
|                  | 0.665 (0.021) | 329.2 | 0.548 (0.025) | 97.8 | 0.453 (0.017) | 1178.3 | 0.627 (0.025) | — | 0.755 (0.022) | 86.9 |
| Fashion-MNIST-10k| acc   | time |         |         |              |         |                   |
|                  | 0.571 (0.025) | 18.6 | 0.537 (0.016) | 29.1 | 0.509 (0.038) | 26.8 | 0.565 (0.021) | 107.3 | 0.569 (0.024) | 707.2/20 | 17.3 |
| Fashion-MNIST    | acc   | time |         |         |              |         |                   |
|                  | 0.561 (0.015) | 335.1 | 0.528 (0.013) | 94.6 | 0.359 (0.017) | 1156.6 | 0.559 (0.013) | — | 0.586 (0.008) | 88.7 |

Table 3. Clustering accuracy (mean value and standard deviation) and time cost (second) on MNIST and Fashion-MNIST with feature extraction. “—” means the computation is out of memory and “/” means the algorithm was performed on a computational platform not comparable to ours. The underlined values are from (Chen et al., 2020).

|                  | LSC-K | SSCC | SSC-OMP | S\(^5\)C | S\(^5\)COMP-C | EGGS+NSE | EGGS\(_{BO}\)+NSE |
|------------------|-------|------|---------|---------|--------------|---------|-------------------|
| MNIST            | acc   | time |         |         |              |         |                   |
|                  | 0.8659 (0.0215) | 273.6 | 0.8229 (0.0503) | 83.3 | 0.8159 | 280.6 | 0.7829 (0.0283) | 907.5 | 0.9632 | 316.8 | 0.9775 (0.0034) | 59.2 |
| Fashion-MNIST    | acc   | time |         |         |              |         |                   |
|                  | 0.6131 (0.0298) | 251.8 | 0.6220 (0.0159) | 71.64 | 0.3796 (0.0217) | 1013.9 | 0.6057 (0.0227) | — | 0.6398 (0.0133) | 61.9 |
| GTSRB            | acc   | time |         |         |              |         |                   |
|                  | 0.8711 (0.0310) | 31.2 | 0.8104 (0.0321) | 37.1 | 0.8252 | 18 / | 0.9044 (0.0267) | 98.7 | 0.9554 | 16 / | 0.9873 (0.0126) | 16.8 |

6. Experiments

We test the proposed method on Extended Yale B Face (Kuang-Chih et al., 2005), ORL Face (Samaria & Harter, 1994), COIL20 (Nene et al., 1996), AR Face (Martínez & Kak, 2001), MNIST (LeCun et al., 2010), Fashion-MNIST (Xiao et al., 2017), GTSRB (Stallkamp et al., 2012), subsets and extracted features of MNIST and Fashion-MNIST. More details about the datasets are in Appendix D.1.

6.1. Performance on small datasets

We first compare our methods with SSC (Elhamifar & Vidal, 2013), LRR (Liu et al., 2013), EDSC (Pan Ji et al., 2014), KSSC (Patel & Vidal, 2014), SSC-OMP (You et al., 2016b), BDR-Z (Lu et al., 2018), and BDR-B (Lu et al., 2018), on the six smaller datasets. To save space, the hyperparameter settings are moved into Appendix D.2.

The clustering accuracy and time cost on six small datasets are reported in Table 1. On Yale B and ORL datasets, EGGS and EGGS\(_{BO}\) outperformed other methods significantly. SSC and KSSC have high clustering accuracy on COIL20. The reason is that the different poses of the images can be well recognized and exploited by the sparse self-expressive models in SSC and KSSC. On MNIST1k, the difference among SSC, KSSC, BDR, EGGS, and EGGS\(_{BO}\) are not significant. SSC-OMP and EGGS are more efficient than other methods. The clustering accuracy of EGGS is higher than that of SSC-OMP in every case. Note that we didn’t report the time cost of EGGS\(_{BO}\) because it optimized all hyperparameters, which is unfair to compare the time cost.

Note that the search space of EGGS can be arbitrarily large and the performance of EGGS is stable. For example, when \(\Theta = \{10^{-8}, 10^{-7}, \ldots, 10^{0}\}\) and \(T = \{3, 4, \ldots, 100\}\), the clustering accuracy of EGGS on Yale B is still 0.897. See Appendix D.4. But we use a relatively compact search space to avoid the highly unnecessary computation. Numerical results of EGGS\(_{BO}\) for SSC and KSSC in the comparison to (12) and (15) are in Appendix D.6.

6.2. Performance on large datasets

We compare our EGGS+NSE and EGGS\(_{BO}\)+NSE with LSC-K (Chen & Cai, 2011), SSCC (Peng et al., 2013), SSC-OMP (You et al., 2016b), and S\(^5\)C (Matsumisha & Brbic, 2019), and S\(^5\)COMP-C (Chen et al., 2020). The parameter settings and more results are in Appendix D.7 and Appendix D.8 respectively. Table 2 shows the clustering accuracy and standard deviation of 10 repeated trials on the raw-pixel data of MNIST and Fashion-MNIST. We see that EGGS+NSE and EGGS\(_{BO}\)+NSE have highest clustering accuracy in every case. The improvement of EGGS+NSE on MNIST is 13.5% over the second best method LSC-K. Moreover,
EGGS+NSE is always more efficient than other methods. The results on MNIST and Fashion-MNIST features and GTSRB are reported in Table 3. Our two methods outperformed other methods. Note that on Fashion-MNIST, the accuracy of our methods are lower than the deep learning method of (Zhang et al., 2019b), which however has many hyperparameters to determine and has high computational cost. It is worth mentioning that if NSE is ablated, the accuracy of EGGS on MNIST-10k, -20k, and -30k are 0.9772, 0.9783 and 0.9864 respectively, higher than that of EGGS+NSE generally, although the time costs increased a lot.

7. Conclusion

We have proposed a novel clustering method called EGGS with theoretical guarantees for automated spectral clustering. The automatic model and hyperparameter searching and high efficiency of EGGS provide us huge convenience and reliability in real applications. Extensive experiments showed the effectiveness and superiority of our methods over baseline methods. Future work may focus on the theoretical guarantee of EGGS for non-deterministic models instead of the deterministic model given by Definition 3.7.

References

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn., 3(1):1–122, 2011. ISSN 1935-8237. doi: 10.1561/2200000016.

Bruna, J. and Mallat, S. Invariant scattering convolution networks. IEEE transactions on pattern analysis and machine intelligence, 35(8):1872–1886, 2013.

Cai, D. and Chen, X. Large scale spectral clustering via landmark-based sparse representation. IEEE transactions on cybernetics, 45(8):1669–1680, 2014.

Chen, X. and Cai, D. Large scale spectral clustering with landmark-based representation. In Twenty-fifth AAAI conference on artificial intelligence. Citeseer, 2011.

Chen, Y., Li, C.-G., and You, C. Stochastic sparse subspace clustering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4155–4164, 2020.

Elhamifar, E. and Vidal, R. Sparse subspace clustering: Algorithm, theory, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(11):2765–2781, 2013. ISSN 0162-8828. doi: 10.1109/TPAMI.2013.57.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In kdd, volume 96, pp. 226–231, 1996.

Fan, J., Zhang, Y., and Udell, M. Polynomial matrix completion for missing data imputation and transductive learning. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pp. 3842–3849, 2020.

Fiedler, M. Algebraic connectivity of graphs. Czechoslovak mathematical journal, 23(2):298–305, 1973.

Halkidi, M. and Vazirgiannis, M. Clustering validity assessment: finding the optimal partitioning of a data set. In Proceedings 2001 IEEE International Conference on Data Mining, pp. 187–194, 2001. doi: 10.1109/ICDM.2001.989517.

Halko, N., Martinsson, P. G., and Tropp, J. A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. SIAM Review, 53(2):217–288, 2011.

Henderson, H. V. and Searle, S. R. On deriving the inverse of a sum of matrices. SIAM Review, 23(1):53–60, 1981.

Hu, J., Qian, Q., Pei, J., Jin, R., and Zhu, S. Finding multiple stable clusterings. Knowledge and Information Systems, 51(3):991–1021, 2017.

Hutter, F., Kotthoff, L., and Vanschoren, J. Automated machine learning: methods, systems, challenges. Springer Nature, 2019.

Ji, P., Zhang, T., Li, H., Salzmann, M., and Reid, I. Deep subspace clustering networks. In Advances in Neural Information Processing Systems, pp. 24–33, 2017.

Johnson, S. C. Hierarchical clustering schemes. Psychometrika, 32(3):241–254, 1967.

Jones, D. R., Schonlau, M., and Welch, W. J. Efficient global optimization of expensive black-box functions. Journal of Global optimization, 13(4):455–492, 1998.

Kang, Z., Lin, Z., Zhu, X., and Xu, W. Structured graph learning for scalable subspace clustering: From single view to multiview. IEEE Transactions on Cybernetics, pp. 1–31, 2021. doi: 10.1109/TCYB.2021.3061660.
Kheirandishfard, M., Zohrizadeh, F., and Kamangar, F. Deep low-rank subspace clustering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

Klein, A., Falkner, S., Bartels, S., Hennig, P., and Hutter, F. Fast bayesian optimization of machine learning hyperparameters on large datasets. In Artificial Intelligence and Statistics, pp. 528–536. PMLR, 2017.

Kuang-Chih, L., Ho, J., and Kriegman, D. J. Acquiring linear subspaces for face recognition under variable lighting. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(5):684–698, 2005. ISSN 0162-8828. doi: 10.1109/TPAMI.2005.92.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

Li, C.-G. and Vidal, R. Structured sparse subspace clustering: A unified optimization framework. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 277–286, 2015.

Li, C.-G. and Vidal, R. A structured sparse plus structured low-rank framework for subspace clustering and completion. IEEE Transactions on Signal Processing, 64(24):6557–6570, 2016.

Li, J. and Zhao, H. Large-scale subspace clustering by fast regression coding. In IJCAI, 2017.

Li, J., Liu, H., Tao, Z., Zhao, H., and Fu, Y. Learnable subspace clustering. IEEE Transactions on Neural Networks and Learning Systems, pp. 1–15, 2020. doi: 10.1109/TNNLS.2020.3040379.

Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y., and Ma, Y. Robust recovery of subspace structures by low-rank representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(1):171–184, 2013. ISSN 0162-8828. doi: 10.1109/TPAMI.2012.88.

Liu, Y., Li, Z., Xiong, H., Gao, X., and Wu, J. Understanding of internal clustering validation measures. In 2010 IEEE international conference on data mining, pp. 911–916. IEEE, 2010.

Lu, C., Feng, J., Lin, Z., Mei, T., and Yan, S. Subspace clustering by block diagonal representation. IEEE transactions on pattern analysis and machine intelligence, 41(2):487–501, 2018.

Lu, C.-Y., Min, H., Zhao, Z.-Q., Zhu, L., Huang, D.-S., and Yan, S. Robust and efficient subspace segmentation via least squares regression. In European conference on computer vision, pp. 347–360. Springer, 2012.

Martínez, A. M. and Kak, A. C. PCA versus LDA. IEEE transactions on pattern analysis and machine intelligence, 23(2):228–233, 2001.

Matérn, B. Spatial variation, volume 36. Springer Science & Business Media, 2013.

Matsushima, S. and Brbic, M. Selective sampling-based scalable sparse subspace clustering. In Advances in Neural Information Processing Systems, pp. 12416–12425, 2019.

Meila, M. The multicut lemma. UW Statistics Technical Report, 417, 2001.

Meila, M., Shortreed, S., and Xu, L. Regularized spectral learning. In AISTATS. PMLR, 2005.

Nene, S. A., Nayar, S. K., and Murase, H. Columbia object image library (coil-20). Report, Columbia University, 1996.

Ng, A. Y., Jordan, M. I., and Weiss, Y. On spectral clustering: Analysis and an algorithm. In Advances in neural information processing systems, pp. 849–856, 2002.

Pan Ji, Salzmann, M., and Hongdong Li. Efficient dense subspace clustering. In IEEE Winter Conference on Applications of Computer Vision, pp. 461–468, 2014. doi: 10.1109/WACV.2014.6836065.

Patel, V. M. and Vidal, R. Kernel sparse subspace clustering. In 2014 IEEE International Conference on Image Processing (ICIP), pp. 2849–2853, Oct 2014. doi: 10.1109/ICIP.2014.7025576.

Patel, V. M., Hien Van, N., and Vidal, R. Latent space sparse and low-rank subspace clustering. Selected Topics in Signal Processing, IEEE Journal of, 9(4):691–701, 2015. ISSN 1932-4553. doi: 10.1109/JSTSP.2015.2402643.

Peng, X., Zhang, L., and Yi, Z. Scalable sparse subspace clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 430–437, 2013.

Peng, X., Tang, H., Zhang, L., Yi, Z., and Xiao, S. A unified framework for representation-based subspace clustering of out-of-sample and large-scale data. IEEE transactions on neural networks and learning systems, 27(12):2499–2512, 2015.
Poulakis, G. Unsupervised automl: a study on automated machine learning in the context of clustering. Master’s thesis, Πανεπιστήμιο Πειραιώς, 2020.

Roweis, S. T. and Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. Science, 290 (5500):2323–2326, 2000.

Samaria, F. S. and Harter, A. C. Parameterisation of a stochastic model for human face identification. In Proceedings of 1994 IEEE Workshop on Applications of Computer Vision, pp. 138–142, Dec 1994.

Shen, J. and Li, P. Learning structured low-rank representation via matrix factorization. In Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, volume 51 of Proceedings of Machine Learning Research, pp. 500–509, Cadiz, Spain, 09–11 May 2016. PMLR.

Shi, J. and Malik, J. Normalized cuts and image segmentation. IEEE Transactions on pattern analysis and machine intelligence, 22(8):888–905, 2000.

Snoek, J., Larochelle, H., and Adams, R. P. Practical bayesian optimization of machine learning algorithms. Advances in neural information processing systems, 25, 2012.

Soltanolkotabi, M., Elhamifar, E., and Candes, E. J. Robust subspace clustering. The Annals of Statistics, 42(2):669–699, 2014.

Sonoda, S. and Murata, N. Neural network with unbounded activation functions is universal approximator. Applied and Computational Harmonic Analysis, 43(2):233 – 268, 2017. ISSN 1063-5203. doi: https://doi.org/10.1016/j.acha.2015.12.005. URL http://www.sciencedirect.com/science/article/pii/S1063520315001748.

Stallkamp, J., Schliping, M., Salmen, J., and Igel, C. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. Neural networks, 32:323–332, 2012.

Steinhaus, H. et al. Sur la division des corps matériels en parties. Bull. Acad. Polon. Sci, 1(804):801, 1956.

Tipping, M. E. and Bishop, C. M. Mixtures of probabilistic principal component analyzers. Neural computation, 11 (2):443–482, 1999.

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

Wang, S., Tu, B., Xu, C., and Zhang, Z. Exact subspace clustering in linear time. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, pp. 2113–2120, 2014.

Wang, Y.-X. and Xu, H. Noisy sparse subspace clustering. In International Conference on Machine Learning, pp. 89–97. PMLR, 2013.

Weiss, Y. Segmentation using eigenvectors: a unifying view. In Proceedings of the Seventh IEEE International Conference on Computer Vision, volume 2, pp. 975–982 vol.2, 1999. doi: 10.1109/ICCV.1999.790354.

Williams, C. K. and Rasmussen, C. E. Gaussian processes for machine learning, volume 2. MIT press Cambridge, MA, 2006.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

You, C., Li, C.-G., Robinson, D. P., and Vidal, R. Oracle based active set algorithm for scalable elastic net subspace clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3928–3937, 2016a.

You, C., Robinson, D., and Vidal, R. Scalable sparse subspace clustering by orthogonal matching pursuit. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3918–3927, 2016b.

You, C., Li, C., Robinson, D. P., and Vidal, R. Scalable exemplar-based subspace clustering on class-imbalanced data. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 67–83, 2018.

Zhang, J., Li, C.-G., You, C., Qi, X., Zhang, H., Guo, J., and Lin, Z. Self-supervised convolutional subspace clustering network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5473–5482, 2019a.

Zhang, T., Ji, P., Harandi, M., Huang, W., and Li, H. Neural collaborative subspace clustering. In International Conference on Machine Learning, pp. 7384–7393, 2019b.
A. More theoretical results

**Definition A.1** (Polynomial Deterministic Model). The columns of \( X_0 \in \mathbb{R}^{m \times n} \) are drawn from a union of \( k \) different polynomials \( \{g_j : \mathbb{R}^r \to \mathbb{R}^m, r < m\}_{j=1}^k \) of order at most \( p \) and are further corrupted by noise, say \( X = X_0 + E \). Denote the eigenvalue decomposition of the kernel matrix \( K \) of \( X = V \Sigma V^\top \), where \( \Sigma = \text{diag}(\sigma_1, \ldots, \sigma_n) \) and \( \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \). Let \( \gamma = \sigma_{d+1}/\sigma_d \). Denote \( v_i = (v_{i1}, \ldots, v_{in}) \) the \( i \)-th row of \( V \) and let \( \bar{v}_i = (v_{i1}, \ldots, v_{id}) \), where \( d < n/2 \). Suppose the following conditions hold: 1) for every \( i \in [n] \), the \( \tau \)-th largest element of \( \{\bar{v}_i^\top \bar{v}_j : j \in C_{\pi(i)}\} \) is greater than \( \alpha \); 2) \( \max_{i \in [n]} \max_{j \in [n]\setminus C_{\pi(i)}} |\bar{v}_i^\top \bar{v}_j| \leq \beta \); 3) \( \lambda = \sigma_1 \geq \cdots \geq \sigma_{d+1} > \alpha - \beta > 2\mu_n^2 \). Here we consider polynomials because they are easy to analyze and can well approximate smooth functions provided that \( p \) is sufficiently large. Clustering the columns of \( X \) given by Definition A.1 according to the polynomials is actually a manifold clustering problem beyond the setting of subspace clustering. Similar to the subspace detection property, we define

**Definition A.2** (Manifold Detection Property). A symmetric affinity matrix \( A \) obtained from \( X \) has manifold detection property if for all \( i \), the nonzero elements of \( a_i \) correspond to the columns of \( X \) lying on the same manifold as \( x_i \).

The following theorem verifies the effectiveness of (15) followed by the truncation operation in manifold detection.

**Theorem A.3.** Suppose \( X \) and \( K \) are given by Definition A.1 and \( C \) is given by (15), where the kernel function is a polynomial kernel of order \( q \), \( \text{rank}(K_0) = d \) (\( K_0 \) is from \( X_0 \)), and

\[
\begin{align*}
\lambda &> \frac{(\Delta(1+\gamma^2)-2\mu_n\gamma^2)(1-\sqrt{1-\rho^{-1}})\sigma_d^2}{4\mu_d^2-2\Delta} \\
\lambda &< \frac{(\Delta(1+\gamma^2)-2\mu_n\gamma^2)(1+\sqrt{1-\rho^{-1}})\sigma_d^2}{4\mu_d^2-2\Delta}
\end{align*}
\]

(28)

where \( \rho = \frac{(\Delta(1+\gamma^2)-2\mu_n\gamma^2)^2}{4(2\mu_d-2\Delta)(2\mu_d-2\mu_n-2\Delta)} \) and \( \Delta = \alpha - \beta \). Then \( d \leq k(\tau^{(p+q)^2}) \) and the \( A \) obtained by line 9 of Algorithm 1 with \( \tau \leq \bar{\tau} \) has the manifold detection property.

In the theorem, \( \sigma_d \) can be much larger than \( \sigma_{d+1} \) provided that the noise is small enough. Then we get a wide range for \( \lambda \). Compared to Theorem 3.8, Theorem A.3 allows a much larger \( d \), which means the kernel method is able to handle more difficult clustering problems than the linear method.

B. More details about EGGS\(_{BO}\)

In EGGS\(_{BO}\), we use Expected Improvement (EI) acquisition function

\[
a_{\text{EI}}(s|D_k) = E_p[\max(g_{\text{min}} - g(s), 0)],
\]

(29)

where \( g_{\text{min}} \) is the best function value known. The closed-form formulation is

\[
a_{\text{EI}}(s|D_k) = (g_{\text{min}} - \mu)\Phi \left( \frac{g_{\text{min}} - \mu}{\sigma} \right) + \phi \left( \frac{g_{\text{min}} - \mu}{\sigma} \right),
\]

(30)

where \( \mu = \mu(s|D_k, \theta_K) \) and \( \sigma = \sigma(s|D_k, \theta_K) \) are the mean value and variance of the Gaussian process, \( \phi \) and \( \Phi \) are standard Gaussian cumulative density function and probability density function respectively, and \( \theta_K \) denotes the hyperparameters of the Gaussian process. For the covariance function, we use the automatic relevance determination (ARD) Matérn 5/2 kernel (Matérn, 2013):

\[
k_{\text{MAT}}(s, s') = \theta_0 \left( 1 + \sqrt{\frac{5r^2(s, s')}{\theta_0^2}} + \frac{5}{3} \frac{2}{\theta_0^2} \right) \times \exp \left( -\sqrt{5r^2(s, s')} \right),
\]

(31)

where \( r^2(s, s') = \sum_{j=1}^d (s_j - s_j')^2/\theta_j^2 \). EGGS\(_{BO}\) is implemented in MATLAB.

C. Algorithm of EGGS+NSE

**Algorithm 2** EGGS+NSE

**Input:** \( X, k, \Sigma, \Theta, \tilde{n}, \).

1. Select \( \tilde{n} \) landmarks from \( X \) by k-means to form \( \tilde{X} \).
2. Apply Algorithm 1 to \( \tilde{X} \) and get \( \tilde{Z} \).
3. Use mini-batch Adam to solve (26).
4. Compute \( Z \) by (27).
5. Perform k-means on \( Z \).

**Output:** \( k \) clusters: \( C_1, \ldots, C_k \).

D. More about the experiments

D.1. Dataset description

The description for the benchmark image datasets considered in this paper are as follows.

- **Extended Yale B Face** (Kuang-Chih et al., 2005) (Yale B for short): face images \( 192 \times 168 \) of 38 subjects. Each subject has about 64 images under various illumination conditions. We resize the images into \( 32 \times 32 \).
- **ORL Face** (Samaria & Harter, 1994): face images \( 112 \times 92 \) of 40 subjects. Each subject has 10 images with different poses and facial expressions. We resize the images into \( 32 \times 32 \).
D.2. Hyperparameter settings for the small datasets

The parameter $\lambda$ in each of SSC, LRR, and KSSC is chosen from $\{0.01, 0.02, 0.05, 0.1, 0.2, \ldots, 0.5\}$. The $\lambda$ in BDR is chosen from $\{5, 10, 20, \ldots, 80\}$. The $\gamma$ in BDR-B and BDR-Z is chosen from $\{0.01, 0.1, 1\}$. The parameter $s$ in SSC-OMP is chosen from $\{3, 4, \ldots, 15\}$. We report the results of these methods with their best hyperparameters. In EGGS, we set $\Theta = \{0.01, 0.1, 1\}$ and $\mathcal{T} = \{5, 6, \ldots, 15\}$. In EGGS$_{BO}$, we consider two models: 1) Gaussian kernel (1) similarity; 2) (15) with polynomial kernel; 3) (15) with Gaussian kernel, in which the hyperparameters of kernels are optimized adaptively. Then we needn’t to consider (12) explicitly because it is a special case of (15) with polynomial kernel. See Appendix D.6.

D.3. Clustering results in terms of NMI

In addition to the clustering accuracy reported in Table 1, here we also compare the normalized mutual information (NMI) in Table 4. We see that the comparative performance of all methods are similar to the results in Table 1 and our methods EGGS and EGGS$_{BO}$ outperformed other methods in almost all cases.

D.4. The stability of EGGS

Though we have used a relatively compact search space in EGGS to reduce the highly unnecessary computational cost, the search space can be arbitrarily large. Figure 3 shows the clustering accuracy and the corresponding relative-eigen-gap. We can see that the region with highest relative-eigen-gap is in accordance with the region with highest clustering accuracy.

D.5. Eigen-Gap versus relative-eigen-gap

In the paper, we use relative-eigen-gap (reg) defined by (11) in the main paper because the eigen-gap (eg) rely on the scale of the small eigenvalues, which are sensitive to noise and may change significantly when using different values of $F$, $\Theta$, and $\mathcal{T}$. Table 5 shows the relative-eigen-gap we defined can provide much higher clustering accuracy, compared to the ordinary eigen-gap.

D.6. EGGS$_{BO}$ for SSC and KSSC

For SSC, we consider the following problem

$$
\min_c \frac{1}{2}\text{Tr} (K - 2KC + C^\top KC) + \lambda \|C\|_1,
$$

where $K$ is an $n \times n$ kernel matrix with $[K]_{ij} = k(x_i, x_j)$. Note that when we use a linear kernel function, (32) reduces to the vanilla SSC. We solve the optimization via alternating direction method of multipliers (ADMM) (Boyd et al., 2011), where the Lagrange parameter is 0.1 and max number of iterations is 500. In this study, we

All experiments are conducted in MATLAB on a MacBook Pro with 2.3 GHz Intel i5 Core and 8GB RAM.

**References**

- **COIL20** (Nene et al., 1996): images ($32 \times 32$) of 20 objects. Each object has 72 images of different poses.
- **AR Face** (Martínez & Kak, 2001): face images ($165 \times 120$) of 50 males and 50 females. Each subject has 26 images with different facial expressions, illumination conditions, and occlusions. We resize the images into $42 \times 30$.
- **MNIST** (LeCun et al., 1998): 70,000 grey images ($28 \times 28$) of handwritten digits $0 \sim 9$.
- **MNIST-1k(10k)**: a subset of MNIST containing 1000(10000) samples, 100(1000) randomly selected samples per class.
- **Fashion-MNIST** (Xiao et al., 2017): 70,000 grey images ($28 \times 28$) of 10 types of fashion product.
- **Fashion-MNIST-1k(10k)**: a subset of Fashion-MNIST containing 1000(10000) samples, 100(1000) randomly selected samples per class.
- **MNIST-feature**: following the same procedures of (Chen et al., 2020), we compute a feature vector of dimension 3,472 using the scattering convolution network (Bruna & Mallat, 2013) and then reduce the dimension to 500 using PCA.
- **Fashion-MNIST-feature**: similar to MNIST-feature.
- **GTSRB** (Stallkamp et al., 2012): consisting of 12,390 images of street signs in 14 categories. Following (Chen et al., 2020), we extract a 1568-dimensional HOG feature, and reduce the dimension to 500 by PCA.
Table 4. Normalized Mutual Information on the six small datasets

| Dataset         | SSC | LRR | EDSC | KSSC | SSC-OMP | BDR-Z | BDR-B | EGGS | EGGS\textsubscript{BO} |
|-----------------|-----|-----|------|------|---------|-------|-------|------|-----------------------|
| Yale B          | 0.817 | 0.703 | 0.835 | 0.730 | 0.841 | 0.666 | 0.743 | 0.919 | 0.928 |
| ORL             | 0.849 | 0.872 | 0.856 | 0.872 | 0.815 | 0.875 | 0.865 | 0.907 | 0.903 |
| COIL20          | 0.954 | 0.706 | 0.843 | 0.983 | 0.671 | 0.843 | 0.873 | 0.897 | 0.963 |
| AR              | 0.818 | 0.872 | 0.825 | 0.809 | 0.691 | 0.865 | 0.861 | 0.887 | 0.904 |
| MNIST-1k        | 0.612 | 0.538 | 0.631 | 0.983 | 0.671 | 0.843 | 0.873 | 0.963 | 0.652 |
| Fashion-MNIST-1k| 0.616 | 0.601 | 0.621 | 0.621 | 0.559 | 0.614 | 0.614 | 0.667 | 0.652 |

Table 5. The comparison of EGGS with eigen-gap and EGGS with relative-eigen-gap

| Dataset       | eg | reg | accuracy |
|---------------|----|-----|----------|
| YaleB         | 0.790 | 0.768 | 0.775 | 0.663 | 0.562 |
| ORL           | 0.897 | 0.795 | 0.782 | 0.786 | 0.755 | 0.595 |

D.7. Hyperparameter settings of large-scale clustering

On MNIST-10k, MNIST, Fashion-MNIST-10k, and Fashion-MNIST, the parameter settings of (Chen & Cai, 2011), SSSC (Peng et al., 2013), SSC-OMP (You et al., 2016b), and S\textsuperscript{3}C (Matsushima & Brbic, 2019), and S\textsuperscript{3}COMP-C (Chen et al., 2020), and EGGS+NSE are shown in Table 7. These hyper parameters have been de-
termined via grid search and the best (as possible) values are used.

Table 7. Hyper-parameter settings of the compared methods on MNIST-10k, MNIST, Fashion-MNIST-10k, and Fashion-MNIST. s denotes the number of landmark data points. In the optimization (mini-batch Adam) of EGGS+NSE, the epoch number, batch size, and step size are 200, 128, and $10^{-3}$ respectively.

| Method      | s     | d     | γ     |
|-------------|-------|-------|-------|
| LSC-K       | 1000  | 200   | $10^{-5}$ |
| SSC         | 1000  | 200   | 0.01  |
| SSC-OMP     | K = 10 (sparsity) |       |       |
| S³C         | s = 1000, λ = 0.1 or 0.2 |       |       |
| S³COMP-C    | T = 20, λ = 0.4, δ = 0.9 |       |       |
| EGGS+NSE    | s = 1000, d = 200, γ = $10^{-5}$ |       |       |
| EGGS_{BO}+NSE | s = 1000, d = 200, γ = $10^{-5}$ |       |       |

D.8. Influence of hyper-parameters in EGGS+NSE

We investigate the effects of the type of activation function and the number (d) of nodes in the hidden layer of NSE. For convenience, we used a fixed random seed of MATLAB (rng(1)). Figure 6 shows the clustering accuracy on MNIST given by EGGS+NSE with different activation function and different d. We see that ReLU outperformed tanh consistently. The reason is that the nonlinear mapping g from the data space to the nullspace of the Laplacian matrix is nonsmooth and ReLU is more effective than tanh in approximating nonsmooth functions. In addition, when d increases, the clustering accuracy of EGGS+NSE with ReLU often becomes higher because wider network often has higher ability of function approximation.

Figure 7 shows the clustering accuracy on MNIST given by EGGS+NSE with different γ and α. When α is too small (say $10^{-4}$), the clustering accuracy is low, because the training error is quite large in 200 epochs. In fact, by increasing the training epochs, the clustering accuracy can be improved, which however will increase the time cost. When α is relatively large, the clustering accuracy is often higher than 0.755. On the other hand, EGGS+NSE is not sensitive to γ provided that it is not too large.

Figure 8 shows the mean value and standard deviation (10 repeated trials) of the clustering accuracy on MNIST given by EGGS+NSE with different number (denoted by s) of landmark points. It can be found that when the s increases, the clustering accuracy increases and its standard deviation becomes smaller. When s is large enough, the improvement is not significant.

E. Proof for theoretical results

E.1. Proof for Claim 3.2

Proof. The stochastic transition matrix of $G$ is defined as

$$P = D^{-1}A.$$ (33)

In (Meila, 2001), it was showed that

$$\text{MNCut}(\mathcal{C}) \geq k - \sum_{i=1}^{k} \varrho_i(P),$$ (34)

where $\varrho_i(P)$ denotes the $i$-th largest eigenvalue of $P$ and $1 = \varrho_1(P) \geq \varrho_2(P) \geq \cdots \varrho_k(P)$. According to Lemma
Figure 8. Influence of the number of landmark points in EGGS+NSE on MNIST. We set \( d = 200 \), \( \gamma = 10^{-5} \), and \( \alpha = 10^{-3} \). The shadow denotes the standard deviation of 10 trials.

Red from (Meila, 2001), we have

\[
\sigma_i(L) = 1 - q_i(P), \quad \forall i = 1, \ldots, n. \tag{35}
\]

Substituting (35) into (34), we have

\[
\text{MNCut}(C) \geq \sum_{i=1}^{k} \sigma_i(L). \tag{36}
\]

**Remark E.1.** \( C \) can be any partition of the nodes of \( G \). Let \( C^* \) be the optimal partition. Then \( \text{MNCut}(C^*) = \sum_{i=1}^{k} \sigma_i(L) \). If \( \sum_{i=1}^{k} \sigma_i(L) = 0 \), there are no connections (edges) among \( C_1^* \), \ldots, \( C_k^* \).

**E.2. Proof for Claim 3.3**

**Proof.** For \( i = 1, \ldots, k \), we aim to partition \( C_i \) into two subsets, denoted by \( C_i^1 \) and \( C_i^2 \). Then we define

\[
\text{MNCut}(C_i) = \frac{\text{Cut}(C_i^1, C_i^2)}{\text{Vol}(C_i^1)} + \frac{\text{Cut}(C_i^2, C_i^1)}{\text{Vol}(C_i^2)}. \tag{37}
\]

It follows that

\[
\text{MNCut}(C_i) \geq \sum_{j=1}^{2} \sigma_j(L_{C_i}) \geq \sigma_2(L_{C_i}) = \text{ac}(C_i), \tag{38}
\]

where \( L_{C_i} \) denotes the Laplacian matrix of \( C_i \) an \( i = 1, \ldots, k \). Since \( \sigma_{k+1}(L) = \min\{\text{ac}(C_1), \ldots, \text{ac}(C_k)\} \), we have

\[
\min_{1 \leq i \leq k} \text{MNCut}(C_i) \geq \sigma_{k+1}(L). \tag{39}
\]

Therefore, \( \sigma_{k+1}(L) \) measures the least connectivity of \( C_1, \ldots, C_k \). This finished the proof.

**Remark E.2.** When \( \sigma_{k+1}(L) \) is large, the connectivity in each of \( C_1, \ldots, C_k \) is strong. Otherwise, the connectivity in each of \( C_1, \ldots, C_k \) is weak. When \( \sigma_{k+1}(L) = 0 \), at least one of \( C_1, \ldots, C_k \) contains at least two components, which means the nodes of \( G \) can be partitioned into \( k + 1 \) or more clusters.

**E.3. Proof for Theorem 3.4**

**Proof.** According to Theorem 1 of (Meila et al., 2005), we have

\[
\text{dist}(C, C') < \frac{3\delta}{\sigma_{k+1}(L) - \sigma_k(L)}. \tag{40}
\]

Since \( \text{reg}(L) = \frac{1}{n} \sum_{i=1}^{k} \sigma_i(L) \), we have

\[
\sigma_{k+1}(L) - \sigma_k(L) = \text{reg}(L)(\hat{\sigma} + \epsilon) + \bar{\sigma} - \sigma_k(L), \tag{41}
\]

where \( \bar{\sigma} = \frac{1}{k} \sum_{i=1}^{k} \sigma_i(L) \geq \epsilon \). Invoking (41) into (40), we arrive at

\[
\text{dist}(C, C') < \frac{3\delta}{2\text{reg}(L) + (1 - k)\eta\epsilon}. \tag{42}
\]

This finished the proof.

**E.4. Proof for Proposition 3.5**

**Proof.** Since \( \hat{\epsilon} \) is the optimal solution, we have

\[
\phi(x_i)^T(\phi(y) - \phi(X)\hat{\epsilon}) + \lambda\hat{\epsilon}_i = 0, \tag{43}
\]

\[
\phi(x_j)^T(\phi(y) - \phi(X)\hat{\epsilon}) + \lambda\hat{\epsilon}_j = 0. \tag{44}
\]

It follows that

\[
\|\hat{\epsilon}_i - \hat{\epsilon}_j\| = \|((\phi(x_i) - \phi(x_j))^T(\phi(y) - \phi(X)\hat{\epsilon}))\| \leq \|\phi(x_i) - \phi(x_j)\|\|\phi(y) - \phi(X)\hat{\epsilon}\|
\]

\[
= \sqrt{k(x_i, x_i) - 2k(x_i, x_j) + k(x_j, x_j)} \times \|\phi(y) - \phi(X)\hat{\epsilon}\|
\]

\[
= \sqrt{2 - 2k(x_i, x_j)} \|\phi(y) - \phi(X)\hat{\epsilon}\|
\]

\[
\leq \sqrt{2 - 2k(x_i, x_j)}\|\phi(y)\|
\]

\[
= \sqrt{2 - 2\exp\left(-\frac{\|x_i - x_j\|^2}{2\epsilon^2}\right)}. \tag{42}
\]
In the second and last equalities, we used the fact that \( \|\phi(y)\| = \|\phi(x)\| = 1 \). In the second inequality, we used the fact that \( \frac{1}{2}\|\phi(y) - \phi(X)\|^2 + \frac{1}{2}\|\bar{c}\|^2 \leq \frac{1}{2}\|\phi(y)\|^2 \) because \( \bar{c} \) is the optimal solution. \( \square \)

E.5. Proof for Theorem 3.8

Proof. Invoking the SVD of \( X \) into the closed-form solution (13), we get

\[
C = V \text{diag} \left( \frac{\sigma_1^2}{\sigma_1^2 + \lambda}, \ldots, \frac{\sigma_n^2}{\sigma_n^2 + \lambda} \right) V^\top. \tag{43}
\]

It means

\[
c_{ij} = \sum_{l=1}^{n} \frac{v_i^l v_j^l \sigma_l^2}{\sigma_l^2 + \lambda} \tag{44}
= \hat{v}_i^\top \hat{v}_j - \sum_{l=1}^{d} \frac{v_i^l v_j^l \lambda}{\sigma_l^2 + \lambda} + \sum_{l=d+1}^{n} \frac{v_i^l v_j^l \sigma_l^2}{\sigma_l^2 + \lambda}.
\]

Suppose \( j \in C_π(i) \) and \( k \in [n] \setminus C_π(i) \). We have

\[
|c_{ij} - c_{ik}| = \left| \hat{v}_i^\top \hat{v}_j - \sum_{l=1}^{d} \frac{v_i^l v_j^l \lambda}{\sigma_l^2 + \lambda} + \sum_{l=d+1}^{n} \frac{v_i^l v_j^l \sigma_l^2}{\sigma_l^2 + \lambda} \right|
- \left| \hat{v}_i^\top \hat{v}_k - \sum_{l=1}^{d} \frac{v_i^l v_k^l \lambda}{\sigma_l^2 + \lambda} + \sum_{l=d+1}^{n} \frac{v_i^l v_k^l \sigma_l^2}{\sigma_l^2 + \lambda} \right|
\leq \left| \hat{v}_i^\top \hat{v}_j - \sum_{l=1}^{n} \frac{v_i^l v_j^l \lambda}{\sigma_l^2 + \lambda} - \sum_{l=d+1}^{n} \frac{v_i^l v_j^l \sigma_l^2}{\sigma_l^2 + \lambda} \right|
- \left| \hat{v}_i^\top \hat{v}_k - \sum_{l=1}^{n} \frac{v_i^l v_k^l \lambda}{\sigma_l^2 + \lambda} - \sum_{l=d+1}^{n} \frac{v_i^l v_k^l \sigma_l^2}{\sigma_l^2 + \lambda} \right|
\leq |v_i^\top v_j| - |v_i^\top v_k| - 2\mu \sum_{l=1}^{d} \frac{\sigma_l^2}{\sigma_l^2 + \lambda} - 2\mu \sum_{l=d+1}^{n} \frac{\sigma_l^2}{\sigma_l^2 + \lambda}
\geq |v_i^\top v_j| - |v_i^\top v_k| - 2\mu \frac{\sigma_d^2}{\sigma_d^2 + \lambda} - 2\mu \frac{\sigma_{d+1}^2}{\sigma_{d+1}^2 + \lambda}, \tag{45}\]

where \( \alpha = \min(m, n) - d = m - d \).

To ensure that there exist at least \( \bar{\tau} \) elements of \( \{|c_{ij}| : j \in C_\pi(i)\} \) greater than \( |c_{ik}| \) for all \( k \in [n] \setminus C_\pi(i) \), we need

\[
|v_i^\top v_j| - |v_i^\top v_k| - 2\mu \frac{\sigma_d^2}{\sigma_d^2 + \lambda} - 2\mu \frac{\sigma_{d+1}^2}{\sigma_{d+1}^2 + \lambda} > 0 \tag{46}\]

holds at least for \( \bar{\tau} \) different \( j \), where \( j \in C_\pi(i) \). It is equivalent to ensure that

\[
\alpha - \beta - 2\mu \frac{\sigma_d^2}{\sigma_d^2 + \lambda} - 2\mu \frac{\sigma_{d+1}^2}{\sigma_{d+1}^2 + \lambda} > 0. \tag{47}\]

We rewrite (47) as

\[
u_1 \lambda^2 + u_2 \lambda + u_3 > 0, \tag{48}\]

where \( u_1 = \alpha - \beta - 2\mu d, u_2 = (\alpha - \beta)(\sigma_d^2 + \sigma_{d+1}^2) - 2\mu (d + a) \sigma_{d+1}^2, \) and \( u_3 = (\alpha - \beta - 2\mu a) \sigma_d^2 \).

The definition of \( \mu, \alpha, \) and \( \beta \) imply \( u_1 < 0 \). Then we solve (47) and obtain

\[
\begin{align*}
\lambda > \frac{(\alpha - \beta)(\sigma_d^2 + \sigma_{d+1}^2) - 2\mu d (d + a) \sigma_{d+1}^2 - \sqrt{w}}{2(2\mu d - (\alpha - \beta))} \\
\lambda < \frac{(\alpha - \beta)(\sigma_d^2 + \sigma_{d+1}^2) - 2\mu d (d + a) \sigma_{d+1}^2 + \sqrt{w}}{2(2\mu d - (\alpha - \beta))} \tag{49}
\end{align*}
\]

where \( w = u_2^2 - 4u_1 u_3 \). To simplify the notations, we let \( \Delta = \alpha - \beta, \sigma_{d+1} = \gamma \sigma_d \) and get

\[
\begin{align*}
\lambda > \frac{(\Delta + \gamma)^2 - 2\mu m \gamma^2}{4(2\mu d - (\Delta - 2\mu))} \\
\lambda < \frac{(\Delta + \gamma)^2 - 2\mu m \gamma^2}{4(2\mu d - (\Delta - 2\mu))} \tag{50}
\end{align*}
\]

Further, let \( \rho = \frac{(\Delta + \gamma)^2 - 2\mu m \gamma^2}{4(\Delta - 2\mu)(\Delta - 2\mu_m)} \), we arrive at

\[
\begin{align*}
\lambda > \left( \frac{\Delta + \gamma}{4(2\mu d - (\Delta - 2\mu))} \right) \left( 1 - \sqrt{1 - \rho^{-1}} \right) \sigma_d^2 \\
\lambda < \left( \frac{\Delta + \gamma}{4(2\mu d - (\Delta - 2\mu))} \right) \left( 1 + \sqrt{1 - \rho^{-1}} \right) \sigma_d^2 \tag{51}
\end{align*}
\]

That means, if (51) holds, for every \( i \), the indices of the largest \( \bar{\tau} \) absolute elements in the \( i \)-th column of \( C \) are in \( C_\pi(i) \). Therefore, the truncation operation with parameter \( \tau \leq \bar{\tau} \) ensures the subspace detection property. This finished the proof. \( \square \)

E.6. Proof for Proposition 3.9

Proof. The condition of reg means

\[
\frac{\sigma_{k+1}(L)}{\frac{1}{k} \sum_{i=1}^{k} \sigma_i(L)} - \frac{1}{k} \sum_{i=1}^{k} \sigma_i(L) \leq \frac{\sigma_{k+1}(L)}{\epsilon} \geq 0.
\]

For convenience, denote \( \vartheta = \frac{1}{k} \sum_{i=1}^{k} \sigma_i(L) \). We have

\[
-\vartheta \epsilon = \partial \sigma_{k+1}. \tag{52}
\]

It indicates \( \vartheta = 0 \) and \( \sigma_{k+1} \neq 0 \). Therefore the graph has exactly \( k \) connected components. Since the subspace or manifold detection property hold for \( A \), each component of \( G \) is composed of the columns of \( X \) in the same subspace or manifold. Thus, all the columns of \( X \) in the same subspace or manifold must be in the same component. Otherwise, the number of connected components is larger than \( k \). \( \square \)
E.7. Proof for Theorem A.3

The proof is nearly the same as that for Theorem 3.8, except that $d \leq n/2$ and $\text{rank}(K_0) \leq k^{(r+pq)}$, where $K_0 = \phi(X_0)^T \phi(X_0)$. In this case, $K$ can be well approximately by a low-rank matrix of rank at most $k^{(r+pq)}$ provided that the noise is small enough. More details about $K_0$ can be found in (Fan et al., 2020).

E.8. Proof for Proposition 5.1

Proof. We only need to provide an example of $W_1 \in \mathbb{R}^{d \times m}$, $W_2 \in \mathbb{R}^{k \times d}$, $b_1 \in \mathbb{R}^d$, and $b_2 \in \mathbb{R}^k$, where $d = kr$, such that the clusters can be recognized by k-means.

We organize the rows of $W_1$ into $k$ groups: $W_1^j \in \mathbb{R}^{r \times m}$, $j = 1, \ldots, k$. Let $W_1^j = U_j^T$, $j = 1, \ldots, k$. Let $W_1 x_i = \alpha_i = (\alpha^1_i, \ldots, \alpha^k_i)$. When $x_i \in S_j$, we have

$$\alpha^j_i = U_j^T x_i = U_j^T U_j v_i = v_i.$$  \hfill (52)

It follows from the assumption that

$$\max_p \alpha^j_{pi} > \mu.$$  \hfill (53)

Let $b_1 = [b^1_1; \ldots; b^k_1] = -\mu 1$. Then $h^j_i = \text{ReLU}(\alpha^j_i + b^j_1)$ has at least one positive element. On the other hand, since

$$\alpha^l_i = U^l_T x_i = U^l_T U^l v_i, \quad l \neq j,$$  \hfill (54)

using the assumption of $\mu$, we have

$$|\alpha^j_{pi}| = |U^j_{lp}^T U^j v_i| \leq \|U^j_{lp}^T U^j\| \|v_i\| \leq \mu,$$  \hfill (55)

where we have used the fact $\|v_i\| = 1$ because $\|x_i\| = 1$. It follows that

$$h^j_i = \text{ReLU}(\alpha^j_i + b^j_1) = 0, \quad l \neq j.$$  

Now we formulate $W_2$ as

$$W_2 = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1k} \\ q_{21} & q_{22} & \cdots & q_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ q_{k1} & q_{k2} & \cdots & q_{kk} \end{bmatrix},$$  \hfill (56)

where $q_{lj} \in \mathbb{R}^{1 \times r}$, $l, j = 1, \ldots, k$. We have

$$z_{ji} = q_{j1} h^1_i + q_{j2} h^2_i + \cdots + q_{jk} h^k_i = q_{ji} h^j_i,$$

and

$$z_{li} = q_{l1} h^1_i + q_{l2} h^2_i + \cdots + q_{lk} h^k_i = q_{li} h^j_i.$$  

Here we have let $b_2 = 0$. Let $q_{ji} \geq 0$ and $q_{lj} = 0$, we have

$$z_{ji} > z_{li} = 0.$$  

Therefore, if $x_i \in S_j$, we have $z_{ji} > 0$ and $z_{li} = 0 \quad \forall 1 \leq j \neq l \leq k$. Now performing k-means on $Z = [z_1, \ldots, z_n]$ can identify the clusters trivially.

$\square$