

# CAST: A Correlation-based Adaptive Spectral Clustering Algorithm on Multi-scale Data

Xiang Li<sup>§</sup>, Ben Kao<sup>§</sup>, Caihua Shan<sup>§</sup>, Dawei Yin<sup>†</sup>, Martin Ester<sup>‡</sup>

<sup>§</sup>The University of Hong Kong, Pokfulam Road, Hong Kong

<sup>†</sup>JD.com, Beijing, China

<sup>‡</sup>Simon Fraser University, Burnaby, BC, Canada

<sup>§</sup>{xli2, kao, chshan}@cs.hku.hk <sup>†</sup>yindawei@acm.org <sup>‡</sup>ester@sfu.ca

## ABSTRACT

We study the problem of applying spectral clustering to cluster multi-scale data, which is data whose clusters are of various sizes and densities. Traditional spectral clustering techniques discover clusters by processing a similarity matrix that reflects the proximity of objects. For multi-scale data, distance-based similarity is not effective because objects of a sparse cluster could be far apart while those of a dense cluster have to be sufficiently close. Following [16], we solve the problem of spectral clustering on multi-scale data by integrating the concept of objects’ “reachability similarity” with a given distance-based similarity to derive an objects’ coefficient matrix. We propose the algorithm CAST that applies *trace Lasso* to regularize the coefficient matrix. We prove that the resulting coefficient matrix has the “grouping effect” and that it exhibits “sparsity”. We show that these two characteristics imply very effective spectral clustering. We evaluate CAST and 10 other clustering methods on a wide range of datasets w.r.t. various measures. Experimental results show that CAST provides excellent performance and is highly robust across test cases of multi-scale data.

## KEYWORDS

Spectral clustering; robustness; multi-scale data

### ACM Reference Format:

Xiang Li<sup>§</sup>, Ben Kao<sup>§</sup>, Caihua Shan<sup>§</sup>, Dawei Yin<sup>†</sup>, Martin Ester<sup>‡</sup>. 2020. CAST: A Correlation-based Adaptive Spectral Clustering Algorithm on Multi-scale Data. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD ’20), August 23–27, 2020, Virtual Event, CA, USA*. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3394486.3403086>

## 1 INTRODUCTION

Cluster analysis is a fundamental task in machine learning and data mining, which seeks to group similar objects into same clusters and separate dissimilar objects into different clusters. Spectral clustering, which transforms clustering into a graph partitioning problem, has been shown to be effective in image segmentation [37], text

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
*KDD ’20, August 23–27, 2020, Virtual Event, CA, USA*

© 2020 Association for Computing Machinery.  
ACM ISBN 978-1-4503-7998-4/20/08...\$15.00  
<https://doi.org/10.1145/3394486.3403086>

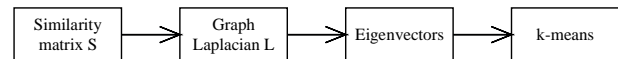


Figure 1: Spectral clustering pipeline

mining [8] and information network analysis [17]. These are fundamental tasks that are at the cores of many applications and services, such as text/media information retrieval systems, recommender systems, and viral marketing.

Given a set of objects  $X = \{x_1, x_2, \dots, x_n\}$  and a similarity matrix  $S$  such that each entry  $S_{ij}$  represents the affinity between objects  $x_i$  and  $x_j$ , standard spectral clustering methods first construct a graph  $G = (X, S)$ , where  $X$  denotes the set of vertices and  $S_{ij}$  gives the weight of the edge that connects  $x_i$  and  $x_j$ . Then, the graph Laplacian  $L$  of  $G$  is computed and eigen-decomposition is performed on matrix  $L$  to derive the  $k$  smallest eigenvectors  $\{e_1, e_2, \dots, e_k\}$ <sup>1</sup>, where  $k$  is the desired number of clusters and  $e_i$  is the  $i$ -th smallest eigenvector. These eigenvectors form a  $k \times n$  matrix, whose  $j$ -th column is taken as the feature vector of object  $x_j$ . (Essentially, objects are mapped into low-dimensional embeddings using the eigenvectors.) Finally, a post-processing step, e.g.,  $k$ -means, is applied on the objects with their feature vectors to return clusters. Figure 1 illustrates the general pipeline of spectral clustering.

Spectral clustering aims to optimize certain criterion that measures the quality of graph partitions. For example, the NCuts [29] method minimizes the *normalized cut* between clusters, which measures the weights of inter-cluster edges. Conventionally, objects’ affinity is given by some distance-based similarity. For multi-scale data, which consists of object clusters of different sizes and densities, distance-based similarity is often ineffective in capturing the correlations between objects [26, 38]. This leads to poor performance of spectral methods. For example, Fig. 2(a) shows a dense rectangular cluster located on top of a very sparse strip-shaped cluster. Objects at different ends of the strip-shaped cluster are far apart and hence their distance-based similarity is small. Fig. 2(b) shows the clustering given by NCuts, from which we see that the strip-shaped cluster is incorrectly segmented.

In [16], the ROSC algorithm was proposed to address the multi-scale data issue in spectral clustering. The idea is to rectify a given distance-based similarity matrix  $S$  by deriving a coefficient matrix  $Z$  that can better express the correlation among objects. Intuitively, each entry  $Z_{ij}$  in  $Z$  represents how well an object  $x_i$  characterizes

<sup>1</sup>We say that an eigenvector  $e_i$  is smaller than another eigenvector  $e_j$  if  $e_i$ ’s eigenvalue is smaller than that of  $e_j$ ’s.

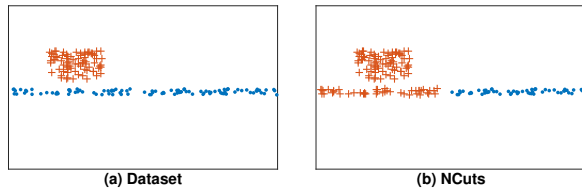


Figure 2: (a) A multi-scale dataset, (b) clustering by NCuts

another object  $x_j$ , and two objects are considered highly correlated (and thus should be put into the same cluster) if they give similar characterization to other objects. The coefficient matrix  $Z$  is constructed based on the similarity matrix  $S$  as well as a transitive  $K$ -nearest-neighbor (TKNN) graph. Specifically, two objects  $x_i$  and  $x_j$  are connected in the TKNN graph if there exists an object sequence  $\langle x_i, \dots, x_j \rangle$  such that adjacent objects in the sequence are  $K$ -nearest-neighbors of each other. For example, objects that are located at far ends of the strip-shaped cluster (Fig. 2) are connected by a chain of  $K$ -NN relations. An important property that was proven in [16] is that the matrix  $Z$  has the *grouping effect* [16, 24], which states that if two objects are similar in terms of both  $S$  and TKNN graph connectivity, their corresponding coefficient vectors in  $Z$  are also similar. Based on  $Z$ , ROSC constructs a new correlation matrix  $\tilde{Z}$ . The grouping effect of  $Z$  ensures that highly correlated objects are grouped together by applying spectral clustering on  $\tilde{Z}$ .

Besides expressing the correlation between objects of the same cluster, another important factor for correct clustering is to suppress the correlation between objects of different clusters. ROSC, however, focuses on enhancing the former by deriving a coefficient matrix  $Z$  that amplifies intra-cluster correlation; it does not promote the latter. Our objective is to study methods that deal with both factors. Specifically, our proposed algorithm CAST regularizes matrix  $Z$  so that it has grouping effect and it exhibits inter-cluster *sparsity*. By sparsity, we refer to the desired property that entries in the matrix that correspond to inter-cluster object pairs should be 0 or very small, hence the matrix is sparse.

One common approach to enforce sparsity is to apply  $\ell_1$  regularization on a solution matrix (i.e., by including the  $\ell_1$ -norm as a penalty term in an optimization problem). While using the  $\ell_1$ -norm helps sparsify inter-cluster correlation, it weakens intra-cluster correlation and hence it goes against establishing grouping effect. In contrast, ROSC uses the Frobenius norm to regularize the coefficient matrix  $Z$ , which is equivalent to regularizing each column vector of  $Z$  by the  $\ell_2$ -norm. The Frobenius norm has been shown to enhance grouping effect but not sparsity. Generally, a desired coefficient matrix should possess both grouping effect for objects in the same cluster and sparsity for objects from different clusters. To construct such a matrix, we introduce trace Lasso [9], which is a regularizer that falls in between the  $\ell_1$ -norm and the  $\ell_2$ -norm. The trace Lasso is adaptive depending on the correlation between objects. Given a set of objects  $\mathcal{X}$ , let  $X$  denote the feature matrix of objects, whose  $j$ -th column is the feature vector  $\mathbf{x}_j$  of an object  $x_j$ . (We assume the vectors are normalized, i.e.,  $\mathbf{x}_j^T \mathbf{x}_j = 1$  for all  $j$ .) If objects are highly correlated, i.e.,  $X^T X = \mathbf{1}\mathbf{1}^T$  ( $\mathbf{1}$  is the all-one vector), the trace Lasso is equivalent to the  $\ell_2$ -norm; If objects are

independent, i.e.,  $X^T X = I$  ( $I$  is the identity matrix), the trace Lasso will behave like the  $\ell_1$ -norm.

In this paper we study spectral clustering over multi-scale data. We propose the Correlation-based Adaptive Spectral clustering method using Trace lasso, or CAST. We discuss how CAST takes advantage of the trace Lasso to achieve robust spectral clustering. We summarize our main contributions as follows.

- We study the problem of applying spectral clustering on multi-scale data. We propose the CAST algorithm, which uses trace Lasso to construct and regularize a coefficient matrix  $Z$ . A correlation matrix that exhibits grouping effect and inter-cluster sparsity is subsequently derived for effective and robust spectral clustering.
- We mathematically prove that the derived matrix by CAST has grouping effect. This ensures high intra-cluster object correlation.
- We conduct extensive experiments to show the effectiveness of CAST. We compare CAST with 10 other methods w.r.t. various clustering quality measures over a wide range of datasets. Our results show that CAST consistently provides very good performance over the range of datasets. It is thus a very robust algorithm especially in handling multi-scale data.

The rest of the paper is organized as follows. Section 2 introduces related works. In Section 3 we describe the ROSC algorithm, give formal definitions of some important concepts based on which our algorithm is designed, and then present CAST. Section 4 presents experimental results. Finally, Section 5 concludes the paper.

## 2 RELATED WORK

Spectral clustering is a widely studied topic [2, 5, 6, 18, 34]. There are many works that study various aspects of spectral clustering, such as computational efficiency [4, 7, 35], clustering performance on data with different characteristics [12, 33, 40], and the theoretic foundations of the method [15, 25, 27]. An introduction to spectral clustering is given in [13, 28, 32].

Despite the success of spectral clustering, previous works [26, 36] have pointed out that spectral methods can be adversely affected when data is multi-scale. To address the problem, the *self-tuning spectral clustering* method ZP [38] uses *local scaling* to extend a Gaussian kernel based similarity  $S_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$  to  $S_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma_i \sigma_j}\right)$ , where  $\mathbf{x}$  (boldface) denotes the feature vector of an object  $x$ , and  $\sigma$  is a scaling parameter. The original formulation uses a global scaling parameter  $\sigma$  for every object pair, which is difficult to set. When  $\sigma$  is set small,  $S_{ij}$  is small and it cannot effectively capture the high correlation between distant objects in a large sparse cluster. On the contrary, when  $\sigma$  is set large,  $S_{ij}$  is large. Objects from different but nearby dense clusters will then more likely to be mis-judged as similar. To address the issue, ZP introduces a local scaling parameter  $\sigma_i$  for each object  $x_i$ , which is defined as the distance between  $x_i$  and its  $l$ -th nearest neighbor ( $l$  can be empirically set). For an object  $x_i$  in a sparse cluster,  $\sigma_i$  is large. This enlarges the similarity between  $x_i$  and other distant objects in the same cluster of  $x_i$ . Also, a dense cluster gives a small  $\sigma_i$ , which effectively decreases the similarity between  $x_i$  and objects from nearby clusters.

Other previous works [1] have suggested to use more eigenvectors to capture more cluster separation information to improve

the effectiveness of spectral clustering on multi-scale data. There are methods that employ pseudo-eigenvectors generated by the power iteration (PI) technique. The PI method is used to compute the dominant eigenvector of a matrix. Lin et al. [20] point out that one can truncate the iterative PI process to obtain an intermediate pseudo-eigenvector. The pseudo-eigenvector is a weighted linear combination of all the eigenvectors and thus contains rich cluster separation information. They propose the *Power Iteration Clustering* (PIC) method based on the idea. Under PIC, however, each object has only one feature value given by the lone pseudo-eigenvector. When the number of clusters is large, a single pseudo-eigenvector is not enough to handle the cluster collision problem [19]. The PIC- $k$  method [19] is subsequently proposed to address this issue. PIC- $k$  runs PI multiple times to generate multiple pseudo-eigenvectors. These pseudo-eigenvectors provide more features of objects for more effective clustering. One issue of the PIC- $k$  method is that the pseudo-eigenvectors are not orthogonal and thus are redundant. To reduce redundancy, [30] proposes a *Deflation-based Power Iteration Clustering* (DPIC) method that uses Schur complement to generate orthogonal pseudo-eigenvectors. Another issue of the PIC-based methods is that the more dominant eigenvectors are assigned larger weights in the PI iteration. Generally, they overshadow other lesser but indispensable eigenvectors. To address this issue, the *Diverse Power Iteration Embedding* (DPIE) method [11] is proposed. When a new pseudo-eigenvector is generated in DPIE, information of previously generated pseudo-eigenvectors is removed from the new one. Ye et al. [36] put forward a *Full Spectral Clustering* (FUSE) method. It uses independent component analysis to rotate  $p > k$  generated pseudo-eigenvectors and make them statistically independent. The  $k$  most informative ones are used for clustering. Finally, in [16], a spectral clustering algorithm for multi-scale data, ROSC, is proposed. Details of ROSC will be covered in the next section.

### 3 ALGORITHMS

In this section we first outline the ROSC algorithm, which is the basis of our algorithm CAST. We define TKNN graph and grouping effect. Then, we discuss the effect of different regularization methods on the coefficient matrix  $Z$ , which is constructed to derive object correlation. After that, we present CAST and prove that the matrix  $Z$  CAST derives has grouping effect and that it promotes sparsity.

#### 3.1 ROSC

The basic idea of ROSC is to construct a coefficient matrix  $Z$  from a given similarity matrix  $S$  and then perform spectral clustering based on  $Z$ . To construct  $Z$ , ROSC first applies PI multiple times to generate  $p$  pseudo-eigenvectors. Whitening [14] is used to reduce the redundancy of these pseudo-eigenvectors. The  $p$  pseudo-eigenvectors together form a  $p \times n$  matrix  $X$ . The  $q$ -th column of  $X$  is taken as the feature vector  $\mathbf{x}_q$  of an object  $x_q$ . That is, ROSC takes the pseudo-eigenvectors as low dimensional embeddings of objects. Assuming the linear subspace model [22], ROSC characterizes each object by others with

$$X = XZ + O, \quad (1)$$

where  $Z \in \mathbb{R}^{n \times n}$  denotes a coefficient matrix such that each entry  $Z_{ij}$  describes how well an object  $x_i$  characterizes another object  $x_j$ , and  $O \in \mathbb{R}^{p \times n}$  is a matrix that captures the noise in the pseudo-eigenvectors. The more similar two objects are, the more likely one object can be represented by the other.

To correlate objects that are located at far ends of a cluster, ROSC regularizes  $Z$  by a TKNN graph.

#### Definition 1. (Transitive $K$ -nearest-neighbor (TKNN) graph)

Given a set of objects  $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$ , the TKNN graph  $\mathcal{G}_K = (\mathcal{X}, \mathcal{E})$  is an undirected graph, where  $\mathcal{X}$  is the set of vertices and  $\mathcal{E}$  is the set of edges. Specifically, the edge  $(x_i, x_j) \in \mathcal{E}$  iff there exists a sequence  $\langle x_i, \dots, x_j \rangle$  such that adjacent objects in the sequence are  $K$ -nearest-neighbors of each other. We use a reachability matrix  $\mathcal{W}$  to represent the TKNN graph, whose  $(i, j)$ -entry  $\mathcal{W}_{ij} = 1$  if  $(x_i, x_j) \in \mathcal{E}$ ; 0 otherwise.  $\square$

ROSC optimizes the objective function

$$\min_Z \|X - XZ\|_F^2 + \alpha_1 \|Z\|_F^2 + \alpha_2 \|Z - \mathcal{W}\|_F^2, \quad (2)$$

where the first term reduces noise  $O$ , the second term is the Frobenius norm of  $Z$  and the third term regularizes  $Z$  by the TKNN graph. It is shown in [16] that Eq. 2 has a closed-form solution  $Z^*$  that has the following grouping effect.

**Definition 2. (Grouping effect).** Given a set of objects  $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$ , let  $\mathbf{w}_q$  denote the  $q$ -th column of  $\mathcal{W}$ . Assume that objects' features are normalized, i.e.,  $\mathbf{x}_q^T \mathbf{x}_q = 1, \forall 1 \leq q \leq n$ . Let  $x_i \rightarrow x_j$  denote the conditions: (1)  $\mathbf{x}_i^T \mathbf{x}_j \rightarrow 1$  and (2)  $\|\mathbf{w}_i - \mathbf{w}_j\|_2 \rightarrow 0$ . A matrix  $Z$  is said to have grouping effect if

$$(x_i \rightarrow x_j) \Rightarrow (|Z_{ip} - Z_{jp}| \rightarrow 0 \forall 1 \leq p \leq n).$$

Grouping effect considers both feature similarity and reachability similarity of objects. Feature similarity is measured based on the closeness of objects' feature vectors (columns in  $X$ ), while reachability similarity is evaluated by columns in  $\mathcal{W}$  that express the connectivity of objects in the TKNN graph. Since the optimal solution  $Z^*$  has grouping effect, highly correlated objects in both similarities will have similar coefficient vectors in  $Z^*$ . ROSC will thus group the objects in the same cluster. Note that  $Z^*$  may be asymmetric and contain negative values. ROSC derives a correlation matrix  $\tilde{Z} = (|Z^*| + |(Z^*)^T|)/2$  and uses that as input to standard spectral clustering in place of the original similarity matrix  $S$ .

#### 3.2 Sparsity

ROSC focuses on generating a coefficient matrix  $Z$  that has grouping effect so that highly correlated objects have similar vector representations in  $Z$ . For effective clustering, we also require that objects from different clusters have sparse connections. To enforce sparsity, a common approach is to use  $\ell_1$  regularization. We thus modify Eq. 2 as

$$\begin{aligned} \min_Z \quad & \frac{1}{2} \|X - XZ\|_F^2 + \alpha_1 \|Z\|_1 + \frac{\alpha_2}{2} \|Z - \mathcal{W}\|_F^2, \\ \text{s.t.} \quad & \text{diag}(Z) = \mathbf{0}, \end{aligned} \quad (3)$$

where the second term is the  $\ell_1$ -norm of  $Z$ . To avoid self-representation of objects, we further add the constraint  $\text{diag}(Z) = \mathbf{0}$ , where  $\text{diag}(Z)$

is the main diagonal vector of  $Z$ . The optimization problem in Eq. 3 is convex and it is equivalent to solving the problem:

$$\begin{aligned} \min_{Z, J} \quad & \frac{1}{2} \|X - XJ\|_F^2 + \alpha_1 \|Z\|_1 + \frac{\alpha_2}{2} \|J - \mathcal{W}\|_F^2 \\ \text{s.t.} \quad & J = Z - \text{Diag}(Z), \end{aligned} \quad (4)$$

where  $\text{Diag}(Z)$  returns a diagonal matrix whose main diagonal vector is that of  $Z$ . Eq. 4 can be solved by the inexact Augmented Lagrange Multiplier (ALM) method [21, 39], which minimizes the following augmented Lagrangian function:

$$\begin{aligned} L(J, Z) = & \frac{1}{2} \|X - XJ\|_F^2 + \alpha_1 \|Z\|_1 + \frac{\alpha_2}{2} \|J - \mathcal{W}\|_F^2 \\ & + \text{tr}(Y^T (J - Z + \text{Diag}(Z))) + \frac{\mu}{2} \|J - Z + \text{Diag}(Z)\|_F^2, \end{aligned}$$

where  $Y$  is the Lagrangian multiplier and  $\mu > 0$  is a penalty parameter.  $L$  can be minimized by alternatively updating one variable with the others fixed. To update  $J$ , we set  $\frac{\partial L}{\partial J} = 0$  and derive

$$J = (X^T X + \alpha_2 I + \mu I)^{-1} (X^T X + \alpha_2 \mathcal{W} - Y + \mu Z - \mu \cdot \text{Diag}(Z)). \quad (5)$$

To update  $Z$ , we set  $\frac{\partial L}{\partial Z} = 0$  and derive

$$Z = A - \text{Diag}(A) \quad \text{and} \quad A = \mathcal{T}_{\frac{\alpha_1}{\mu}} \left( \frac{Y}{\mu} + J \right), \quad (6)$$

where  $\mathcal{T}_{\eta}(\cdot)$  is the shrinkage-thresholding operator acting on each entry of a given matrix, which is defined as  $\mathcal{T}_{\eta}(v) = (|v| - \eta)_+ \text{sgn}(v)$ . The operator  $(\cdot)_+$  returns the argument value if it is non-negative; 0 otherwise. The operator  $\text{sgn}(\cdot)$  gives the sign of the argument value. Algorithm 1 in Appendix shows the algorithm that uses inexact ALM to generate a sparse coefficient matrix  $Z$ . By the theory of inexact ALM, the convergence of Algorithm 1 is guaranteed [21].

### 3.3 CAST

Although the coefficient matrix  $Z$  derived with  $\ell_1$  regularization (Eq. 3) has sparse entries for uncorrelated objects, the regularization also weakens the connections between correlated objects, which goes against the grouping effect. To construct a matrix with both grouping effect for highly correlated objects and sparsity for uncorrelated ones, we regularize  $Z$  by the trace Lasso, which is a regularizer that takes object correlation into consideration. Let  $X$  be a feature matrix whose  $q$ -th column is the feature vector  $\mathbf{x}_q$  of an object  $x_q$ . We normalize  $\mathbf{x}_q$  such that  $\mathbf{x}_q^T \mathbf{x}_q = 1, \forall 1 \leq q \leq n$ . Let  $\mathbf{z}$  be the coefficient vector (a vector in the coefficient matrix  $Z$ ) that corresponds to an object  $x$ , the trace Lasso of  $\mathbf{z}$  is defined as

$$\Omega(\mathbf{z}) = \|X \text{Diag}(\mathbf{z})\|_*, \quad (7)$$

where  $\text{Diag}(\mathbf{z})$  is the diagonal matrix with  $\mathbf{z}$  as its main diagonal. The integration of  $X$  into the norm distinguishes the trace Lasso from other commonly used norms like the  $\ell_1$ -norm and the  $\ell_2$ -norm. Note that,

$$X \text{Diag}(\mathbf{z}) = \sum_{q=1}^n \mathbf{z}_q \mathbf{x}_q \mathbf{e}_q^T, \quad (8)$$

where  $\mathbf{e}_q$ 's are vectors of canonical basis. Consider  $X^T X$  as an encoding of object correlation. Then, if objects are uncorrelated, i.e.,

$X^T X = I$ , the trace Lasso equals the  $\ell_1$ -norm due to the orthogonality of  $\mathbf{x}_i$  and  $\mathbf{e}_i$ :

$$\|X \text{Diag}(\mathbf{z})\|_* = \sum_{q=1}^n \|\mathbf{x}_q\|_2 |z_q| = \sum_{q=1}^n |z_q| = \|\mathbf{z}\|_1. \quad (9)$$

On the other hand, if all the objects are highly correlated and have the same feature vector  $\mathbf{x}$ , i.e.,  $X^T X = \mathbf{1}\mathbf{1}^T$ , the trace Lasso is equivalent to the  $\ell_2$ -norm:

$$\|X \text{Diag}(\mathbf{z})\|_* = \|\mathbf{x}\mathbf{z}^T\|_* = \|\mathbf{x}\|_2 \|\mathbf{z}\|_2 = \|\mathbf{z}\|_2. \quad (10)$$

For other cases, the trace Lasso falls in between the  $\ell_1$ - and the  $\ell_2$ -norms:

$$\|\mathbf{z}\|_2 \leq \|X \text{Diag}(\mathbf{z})\|_* \leq \|\mathbf{z}\|_1. \quad (11)$$

Due to this adaptability, we apply the trace Lasso to regularize  $Z$  in Eq. 2. Specifically, given an object  $x$ , we optimize:

$$\min_{\mathbf{z}} \frac{1}{2} \|\mathbf{x} - X\mathbf{z}\|_2^2 + \alpha_1 \|X \text{Diag}(\mathbf{z})\|_* + \frac{\alpha_2}{2} \|\mathbf{z} - \mathbf{w}\|_2^2. \quad (12)$$

The optimization problem is convex and can be solved by the inexact ALM method. We first transform the problem into:

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{e}\|_2^2 + \alpha_1 \|J\|_* + \frac{\alpha_2}{2} \|\mathbf{h}\|_2^2 \\ \text{s.t.} \quad & \mathbf{e} = \mathbf{x} - X\mathbf{z}, \quad J = X \text{Diag}(\mathbf{z}), \quad \mathbf{h} = \mathbf{z} - \mathbf{w}. \end{aligned} \quad (13)$$

The augmented Lagrangian function of Eq. 13 is

$$\begin{aligned} \hat{L}(\mathbf{e}, J, \mathbf{h}, \mathbf{z}) = & \frac{1}{2} \|\mathbf{e}\|_2^2 + \alpha_1 \|J\|_* + \frac{\alpha_2}{2} \|\mathbf{h}\|_2^2 \\ & + \lambda_1^T (\mathbf{e} - \mathbf{x} + X\mathbf{z}) + \lambda_2^T (\mathbf{h} - \mathbf{z} + \mathbf{w}) + \text{tr}(Y^T (J - X \text{Diag}(\mathbf{z}))) \\ & + \frac{\mu}{2} (\|\mathbf{e} - \mathbf{x} + X\mathbf{z}\|_2^2 + \|J - X \text{Diag}(\mathbf{z})\|_F^2 + \|\mathbf{h} - \mathbf{z} + \mathbf{w}\|_2^2), \end{aligned}$$

where  $\lambda_1, \lambda_2$  and  $Y$  are Lagrangian multipliers, and  $\mu > 0$  is a penalty parameter. We adopt an alternative strategy to update variables of  $\hat{L}$  as in Sec. 3.2 and the update rules are as follows:

$$\mathbf{z} = (X^T X + I + \text{Diag}(X^T X))^{-1} \left( -\frac{X^T \lambda_1}{\mu} - X^T \mathbf{e} + X^T \mathbf{x} + \frac{\lambda_2}{\mu} + \mathbf{h} + \mathbf{w} + \text{diag}\left(\frac{Y}{\mu} + J\right)^T X \right). \quad (14)$$

Here, we overload the notation  $\text{Diag}(\cdot)$ , which returns a diagonal matrix whose main diagonal is that of the argument matrix. Also,  $\text{diag}(\cdot)$  returns the main diagonal vector of the argument matrix.

$$\mathbf{e} = \frac{\mu}{\mu + 1} \left( -\frac{\lambda_1}{\mu} + \mathbf{x} - X\mathbf{z} \right), \quad (15)$$

$$\mathbf{h} = \frac{\mu}{\alpha_2 + \mu} \left( -\frac{\lambda_2}{\mu} + \mathbf{z} - \mathbf{w} \right). \quad (16)$$

Furthermore, updating  $J$  is equivalent to solving the sub-problem:

$$\min_J \frac{\alpha_1}{\mu} \|J\|_* + \frac{1}{2} \|J - X \text{Diag}(\mathbf{z}) + \frac{Y}{\mu}\|_F^2, \quad (17)$$

which is convex and has a closed-form solution that can be solved by the Singular Value Thresholding (SVT) operator [3]. Suppose the singular value decomposition of a rank- $r$  matrix  $X \text{Diag}(\mathbf{z}) - \frac{Y}{\mu} = U\Sigma V^*$ , where  $\Sigma = \text{Diag}([\sigma_1, \dots, \sigma_r])$  and  $\sigma_i$  is the  $i$ -th largest singular value. Let  $\tau = \frac{\alpha_1}{\mu}$  and  $\mathcal{D}_\tau(\Sigma) = \text{Diag}([\sigma_1 - \tau]_+, \dots, [\sigma_r - \tau]_+)$ . The solution to Eq. 17 is:

$$J = U \mathcal{D}_\tau(\Sigma) V^*. \quad (18)$$

Algorithm 2, which is summarized in Appendix, shows the procedure for solving Eq. 12. It is pointed out in [22, 39] that the convergence of inexact ALM cannot be generally proved when there are three or more variables. However, the convexity of the Lagrangian function  $\hat{L}$  guarantees convergence to some extent [22]. Moreover, there are ways to ensure convergence, e.g., by observing that  $\mu$  is upper bounded by Step 10 of Alg. 2. While it is difficult to prove the convergence theoretically, inexact ALM has been empirically observed to perform well in practice [39].

### 3.4 Grouping Effect

Previous works [10, 16, 23, 24] have shown that spectral clustering is effective when applied to data with grouping effect. As we defined in Def. 2, with grouping effect, if two objects are highly correlated, their characterizations of other objects are similar. We measure object correlation by both a feature similarity and a reachability similarity. We next prove that the coefficient vectors  $\mathbf{z}$ 's regularized by the trace Lasso (Eq. 12) result in grouping effect.

**LEMMA 3.** *Given  $\mathbf{w} \in \mathbb{R}^d$ , let  $\bar{\mathbf{w}} = \mathbf{w}^T \mathbf{1}/d$ . The optimal solution  $\mathbf{y}^*$  to the problem:  $\min_{\mathbf{y}} \|\mathbf{y} - \mathbf{w}\|_2^2$ , s.t.  $\mathbf{y}^T \mathbf{1} = d\bar{y}$ , satisfies  $y_j^* = \bar{y} + (w_j - \bar{w})$ ,  $\forall 1 \leq j \leq d$ . Moreover, if  $\|\mathbf{w} - \bar{\mathbf{w}}\mathbf{1}\|_2 \leq \epsilon$ , then,  $\bar{y} - \epsilon \leq y_j^* \leq \bar{y} + \epsilon$ ,  $\forall 1 \leq j \leq d$ .*

**PROOF.** The Lagrangian function  $L'$  of the problem can be written as:  $L' = \sum_{j=1}^d (y_j - w_j)^2 + \beta(\sum_{j=1}^d y_j - d\bar{y})$ , where  $\beta$  is the Lagrangian multiplier. By setting  $\frac{\partial L'}{\partial y_j} = 0$ , we get  $y_j^* = w_j - \frac{\beta}{2}$ . Since  $\mathbf{y}$  satisfies  $\mathbf{y}^T \mathbf{1} = d\bar{y}$ , we substitute  $y_j^*$  into the equation and get  $\beta = 2(\bar{w} - \bar{y})$ ,  $y_j^* = \bar{y} + (w_j - \bar{w})$ . If  $\|\mathbf{w} - \bar{\mathbf{w}}\mathbf{1}\|_2 \leq \epsilon$ , then  $|w_j - \bar{w}| \leq \epsilon$ , i.e.,  $-\epsilon \leq w_j - \bar{w} \leq \epsilon$ . Hence,  $\bar{y} - \epsilon \leq y_j^* \leq \bar{y} + \epsilon$ ,  $\forall 1 \leq j \leq d$ .  $\square$

Given a set of objects  $\mathcal{X} = \{x_1, \dots, x_n\}$ , let  $X$  denote the feature matrix of the objects and  $\mathbf{z}^*$  denote the optimal solution of Eq. 12. We rearrange  $X$  as  $X = [\hat{X}, \tilde{X}]$ , where  $\hat{X} \in \mathbb{R}^{d \times q}$  consists of  $q$  column vectors that are similar to each other and  $\tilde{X} \in \mathbb{R}^{d \times (n-q)}$  consists of the remaining columns. In particular,  $\tilde{X}$  satisfies:

$$\max\{\|\tilde{X} - \bar{\mathbf{x}}_0 \mathbf{1}^T\|_*, \|\tilde{X} - \bar{\mathbf{x}}_0 \mathbf{1}^T\|_F, \|\tilde{X} - \bar{\mathbf{x}}_0 \mathbf{1}^T\|_2\} \leq \epsilon,$$

where  $\epsilon$  is a small positive value,  $\mathbf{1} \in \mathbb{R}^q$  is the all one's vector and  $\bar{\mathbf{x}}_0 = \tilde{X} \mathbf{1}/q$  is the mean vector of columns of  $\tilde{X}$ . Similarly, we rearrange  $\mathbf{z}^* = [\hat{\mathbf{z}}, \tilde{\mathbf{z}}]$ . To prove  $\mathbf{z}^*$  has grouping effect, we only need to show that if  $\|\tilde{\mathbf{z}} - \bar{\mathbf{z}}\mathbf{1}\|_2 > \delta$ , then  $f([\hat{\mathbf{z}}, \tilde{\mathbf{z}}]) > f([\hat{\mathbf{z}}, \bar{\mathbf{z}}\mathbf{1}])$ , where  $\bar{\mathbf{z}} = \mathbf{1}^T \tilde{\mathbf{z}}/q$  is the average value of  $\tilde{\mathbf{z}}$ ,  $\delta$  is a positive value and  $f(\mathbf{z}) = \frac{1}{2} \|\mathbf{x} - X\mathbf{z}\|_2^2 + \alpha_1 \|X \text{Diag}(\mathbf{z})\|_* + \frac{\alpha_2}{2} \|\mathbf{z} - \mathbf{w}\|_2^2$ . Formally,

**THEOREM 4.** *Given a matrix  $X = [\hat{X}, \tilde{X}]$  and a vector  $\mathbf{w} = [\hat{\mathbf{w}}, \tilde{\mathbf{w}}]$ , let  $\mathbf{y}^*$  be the optimal solution to the problem:  $\min_{\mathbf{y}} \|\mathbf{y} - \tilde{\mathbf{w}}\|_2^2$ , s.t.  $\mathbf{y}^T \mathbf{1} = q\bar{z}$ .  $\tilde{X}$  satisfies  $\max\{\|\tilde{X} - \bar{\mathbf{x}}_0 \mathbf{1}^T\|_*, \|\tilde{X} - \bar{\mathbf{x}}_0 \mathbf{1}^T\|_F, \|\tilde{X} - \bar{\mathbf{x}}_0 \mathbf{1}^T\|_2\} \leq \epsilon$ , and  $\tilde{\mathbf{w}}$  satisfies  $\bar{\mathbf{w}} = \mathbf{1}^T \tilde{\mathbf{w}}/d$  and  $\|\tilde{\mathbf{w}} - \bar{\mathbf{w}}\mathbf{1}\|_2 \leq \epsilon$ . If  $\|\tilde{\mathbf{z}} - \bar{\mathbf{z}}\mathbf{1}\|_2 > \delta$ ,  $f([\hat{\mathbf{z}}, \tilde{\mathbf{z}}]) > f([\hat{\mathbf{z}}, \bar{\mathbf{z}}\mathbf{1}])$ , where*

$$\delta = \sqrt{\frac{(2\gamma - \alpha_2 \sum_{j=1}^q [(y_j^* - \bar{z})(y_j^* + \bar{z} - 2\tilde{w}_j)]) (\|\hat{X} \text{Diag}(\hat{\mathbf{z}}) \bar{\mathbf{x}}_0 \bar{\mathbf{z}}^T\|_2)}{\alpha_1 \|\bar{\mathbf{x}}_0\|_2^2}}$$

and  $\gamma = ((\alpha_1 + \|\mathbf{x} - \hat{X}\hat{\mathbf{z}} - \tilde{X}\bar{\mathbf{z}}\mathbf{1}\|_2) \|\tilde{\mathbf{z}}\|_2 + \alpha_1 |\bar{\mathbf{z}}|) \epsilon$ .

The proof of Theorem 4 is given in the Appendix.

**THEOREM 5.**  *$\mathbf{z}^*$  has grouping effect.*

**PROOF.** Given two objects  $x_i$  and  $x_j$  in  $\tilde{X}$ . When  $\epsilon \rightarrow 0$ ,  $\tilde{X}$  will be close to  $\bar{\mathbf{x}}_0 \mathbf{1}^T$  and  $\tilde{\mathbf{w}}$  will be close to  $\bar{\mathbf{w}}\mathbf{1}$ . Hence,  $x_i \rightarrow x_j$ . From Lemma 3,  $\mathbf{y}^*$  in Theorem 4 satisfies  $-\epsilon \leq y_j^* - \bar{z} \leq \epsilon$ . If  $\epsilon \rightarrow 0$ , we have  $\gamma \rightarrow 0$  and  $\sum_{j=1}^q [(y_j^* - \bar{z})(y_j^* + \bar{z} - 2\tilde{w}_j)] \rightarrow 0$ . Further, we get  $\delta \rightarrow 0$ . According to Theorem 4,  $\tilde{\mathbf{z}}$  has to be very close to  $\bar{\mathbf{z}}\mathbf{1}$ . As a result, given two highly correlated objects  $x_i$  and  $x_j$  such that  $x_i \rightarrow x_j$ , we have  $z_i^* \rightarrow z_j^*$ .  $\mathbf{z}^*$  thus has grouping effect.  $\square$

### 3.5 Clustering Procedure

Given a set of objects  $\mathcal{X} = \{x_1, \dots, x_n\}$ , we solve Eq. 12 for each object and construct an optimal solution for the coefficient matrix  $Z^* = [z_1^*, \dots, z_n^*]$ . To prevent self-representation, when we determine  $z_i^*$  using Eq. 12, we remove the  $i$ -th column vector from  $X$ . We note that  $Z^*$  may be asymmetric and contain negative values. To fix, CAST computes a new matrix  $\tilde{Z} = (|Z^*| + |(Z^*)^T|)/2$  as in [22]. It is easy to prove that  $|Z^*|$ ,  $|(Z^*)^T|$  and thus  $\tilde{Z}$  all have grouping effect. Moreover, since the trace Lasso automatically self-adjusts to either the  $\ell_1$ - or the  $\ell_2$ - norm,  $Z^*$  enhances sparsity for objects of different clusters. (We will further illustrate this sparsity effect in the next section.) The matrix  $\tilde{Z}$  is then fed to the pipeline of a standard spectral clustering method (e.g., NCuts) in place of the original similarity matrix  $S$ . Algorithm 3 in Appendix outlines CAST.

## 4 EXPERIMENT

In this section, we conduct extensive experiments to evaluate the performance of CAST. We compare CAST with 10 other clustering methods on a wide range of datasets w.r.t. three popular measures, namely, *purity*, *adjusted mutual information (AMI)*, and *rand index (RI)*. These measures evaluate the clustering quality with values in the range from 0 to 1. A larger value indicates a better clustering quality. For details of the three measures, see [20, 31]. We include the experiment settings in the Appendix.

### 4.1 Algorithms for comparison

We group the algorithms into the following four categories.

- **(Standard spectral clustering methods):** NCuts and NJW are two standard methods. They differ in the way they normalize the graph Laplacian,  $D - S$ , where  $D$  is a diagonal matrix with  $D_{ii} = \sum_{j=1}^n S_{ij}$ . NCuts uses the random-walk-based normalization  $D^{-1}(D - S)$  while NJW employs the symmetric normalization  $D^{-\frac{1}{2}}(D - S)D^{-\frac{1}{2}}$ .
- **(Power iteration (PI)-based methods):** PIC, PIC- $k$ , DPIC, and DPICe apply PI to generate pseudo-eigenvectors as a replacement of eigenvectors. They were described in Section 2.
- **(Multi-scale-data-oriented methods):** ZP and FUSE are two methods that are specifically designed to handle multi-scale data. They were discussed in Section 2. Note that ZP automatically estimates the number of clusters. For a fair comparison, we modify ZP so that it returns  $k$  (the number of true) clusters.
- **(Matrix-reconstruction methods):** This group includes ROSC and CAST. We also consider a version of ROSC that regularizes the  $Z$  matrix using the  $\ell_1$ -norm instead of the Frobenius norm. (That is, we replace Eq. 2 by Eq. 3 and solve the optimization problem using Alg. 1.) We call the method *ROSC with Sparse Matrix*, or **ROSC-S**.

## 4.2 Performance results

**[Synthetic datasets]** We first use two synthetic datasets to illustrate the characteristics of the 11 methods. Both datasets consist of clusters with various densities and sizes. Fig. 3(a) shows SYN1, in which a sparse rectangular cluster (magenta) is sandwiched between a small dense circular cluster (yellow) and a large dense rectangular one (navy blue). The second dataset SYN2 is illustrated in Fig. 4(a), in which two dense square clusters are very close to a sparse half-ring cluster (red). In both datasets, an object in an elongated cluster can be closer to an object of another cluster than to an object that is at a far end of the same cluster. This is purposely done to make clustering very difficult and so that we can visually compare the matrix-reconstruction methods.

Tables 1 and 2 show the clustering performance of the methods for the datasets SYN1 and SYN2, respectively. From the tables, we see that matrix-reconstruction methods generally perform very well. For SYN1 (Table 1), w.r.t. Purity measure, ROSC, ROSC-S and CAST either perform better or comparably with the other methods. For measures AMI and RI, the matrix-reconstruction methods outperform others. For SYN2 (Table 2), CAST and ROSC outperform others by wide margins w.r.t. all three measures. ROSC-S has either better or comparable performance with other non-matrix-reconstruction methods. Moreover, CAST achieves the best performance over all three measures on both SYN1 and SYN2.

We visually compare the performance of the matrix-reconstruction methods for SYN1 in Fig. 3. Recall that ROSC-S uses the  $\ell_1$ -norm to regularize the coefficient matrix. This promotes sparsity. However, as shown in Fig. 3(b), the sparse magenta cluster is incorrectly chipped off on the right side. In contrast, ROSC, which uses the Frobenius norm and promotes object correlation, is connecting objects aggressively. This causes some clusters to overspread into close neighboring clusters (see the regions enclosed in red boxes in Fig. 3(c)). From Fig. 3(d), we see that CAST rectifies the two problems by striking a balance between object connectivity and sparsity using the trace Lasso regularizer. This explains CAST being the best method for SYN1. Figs. 4(b), (c), (d) visually compare the three methods for SYN2. Again, ROSC-S promotes sparsity, and for SYN2, it inadvertently segments the half-ring cluster into three parts. ROSC, which promotes object correlation, recovers more objects of the half-ring cluster, but the half-ring is still split into three segments. For SYN2, CAST avoids the merging of objects on the right side of the half-ring cluster with the square cluster. This shows the adaptability of the trace Lasso regularizer.

Recall that the objective of the matrix-reconstruction methods is to construct a new matrix in place of the original similarity matrix  $S$  and use the constructed matrix as input to the spectral clustering pipeline. Figs. 5 and 6 display the constructed matrices (together with the original similarity matrix  $S$ ) for SYN1 and SYN2, respectively. Each figure displays values in a matrix by pixel brightness. Rows and columns in the matrix are reordered by gold-standard clusters. Readers are advised to view the figures magnified on a computer screen. Intuitively, each luminous rectangular block corresponds to a cluster. Ideally, a figure should have 3 blocks (because there are 3 clusters in each dataset); each block is brightly lit (showing high intra-cluster correlation); and pixels outside the blocks are black (showing sparse inter-cluster correlation).

From Figs. 5(a) and 6(a), we see that the original similarity matrices  $S$  do not have the desired properties. In particular, pixels in blocks are sparse making the blocks not very visible. This results in poor clustering when spectral clustering is applied on  $S$  directly. From Figs. 5(c) and 6(c), we see that by using the Frobenius norm, ROSC is able to significantly amplify intra-cluster correlations, resulting in brightly-lit blocks. However, inter-cluster correlations are inadvertently amplified as well, which is particularly striking in Fig 6(c). On the other hand, ROSC-S, which uses the  $\ell_1$ -norm to promote sparsity, reduces inter-cluster correlation at the expense of less defined blocks. Finally, Figs. 5(d) and 6(d) show that CAST strikes a better balance between intra-cluster correlation and inter-cluster sparsity. This gives better-lit blocks with dimer regions outside the blocks compared with those of ROSC.

We further study how all 11 methods perform on multi-scale data by varying the densities and sizes of some clusters in the synthetic datasets. We make two changes: (1) increase the density of a cluster while keeping its size unchanged, and (2) increase the size of a cluster while maintaining its density unchanged. Here, we show some representative results. Specifically, we increase the density of the middle magenta cluster in SYN1 and use  $\Delta d$  to denote the density change (e.g.,  $\Delta d = 20\%$  means that the density of the cluster is 1.2 times larger than the original one). We change the size of the cluster by enlarging the length sideways with the height fixed. We use  $\Delta s$  to denote the size change (e.g.,  $\Delta s = 100\%$  means that the size of the cluster is doubled). We make similar changes to the half-ring cluster of SYN2. In particular, we gradually enlarge the size of the cluster from a half ring ( $\ominus$ ,  $\Delta s = 0\%$ ) to a whole ring ( $\bigcirc$ ,  $\Delta s = 100\%$ ). The clustering results are shown in Fig. 7. From the figure, we see that CAST gives the best and the most stable performances among all the methods over all the test cases. This shows that a similarity matrix with intra-cluster correlation and inter-cluster sparsity contributes positively to the clustering of multi-scale data. In contrast, ROSC-S, which constructs a sparse matrix but reduces connections between highly correlated objects, and ROSC, which computes a matrix with grouping effect but amplifies the inter-cluster correlations, are thus much less robust.

**[Real datasets]** We further compare the methods using 5 real datasets. They are: *COIL20* (images), *glass* (UCI repository), *MNIST0127* (hand-written digit images), *isolet\_5CLASS* (speech, UCI repository), and *Yale\_5CLASS* (facial images). Some statistics of these datasets are given in the Appendix.

Tables 3, 4 and 5 show the performance results. Since we evaluate the methods on 5 datasets w.r.t. 3 measures, there are in total 15 “contests”. Each row in the tables corresponds to one contest. We highlight the winner’s score of each contest in bold. We also give the ranking of CAST in each contest next to its score. From the tables, we make the following observations:

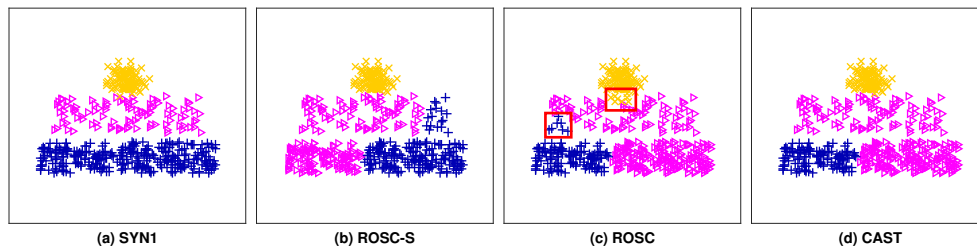
- Matrix-reconstruction methods (ROSC, ROSC-S and CAST) win in all but one contest (*AMI-glass*). In this case, CAST’s score (0.3390) is very close to that of the winner NJW (0.3469). This shows that matrix-reconstruction methods are superior in dealing with multi-scale data. Compared with other competitors, these methods derive new matrices that can more effectively capture object correlations, which explains their excellent performance. For example, all the three methods significantly outperform the rests for *COIL20*.

Measure	NJW	NCuts	PIC	PIC- $k$	DPIC	DPIE	ZP	FUSE	ROSC-S	ROSC	CAST
Purity	0.8000	0.8000	0.7229	0.7220	0.6085	0.7564	0.8000	0.7607	0.7786	0.7826	<b>0.8122</b>
AMI	0.4213	0.4216	0.4092	0.4221	0.1406	0.4523	0.4217	0.4691	0.5060	0.4874	<b>0.5430</b>
RI	0.6953	0.6956	0.6474	0.6586	0.5421	0.6605	0.6958	0.6898	0.7070	0.7021	<b>0.7438</b>

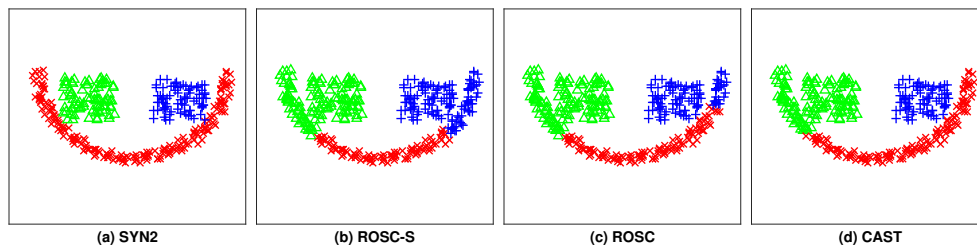
**Table 1: Purity, AMI, and RI scores of methods for dataset SYN1**

Measure	NJW	NCuts	PIC	PIC- $k$	DPIC	DPIE	ZP	FUSE	ROSC-S	ROSC	CAST
Purity	0.6923	0.6917	0.6741	0.6648	0.5556	0.5971	0.6917	0.7298	0.7257	0.7797	<b>0.8188</b>
AMI	0.4472	0.4468	0.4361	0.4158	0.2202	0.1485	0.4468	0.4856	0.4859	0.5611	<b>0.6340</b>
RI	0.6635	0.6632	0.6470	0.6354	0.5139	0.4637	0.6632	0.6976	0.6888	0.7397	<b>0.7683</b>

**Table 2: Purity, AMI, and RI scores of methods for dataset SYN2**



**Figure 3: Clustering results for SYN1**



**Figure 4: Clustering results for SYN2**

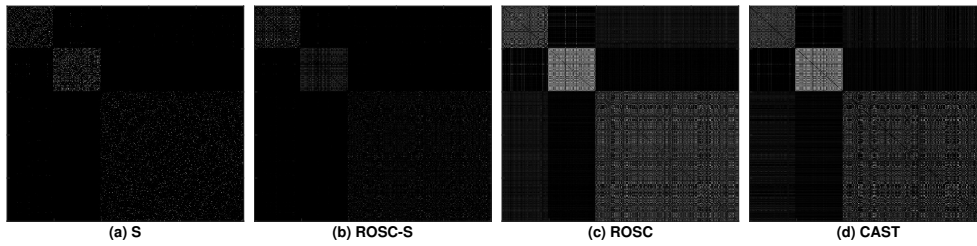
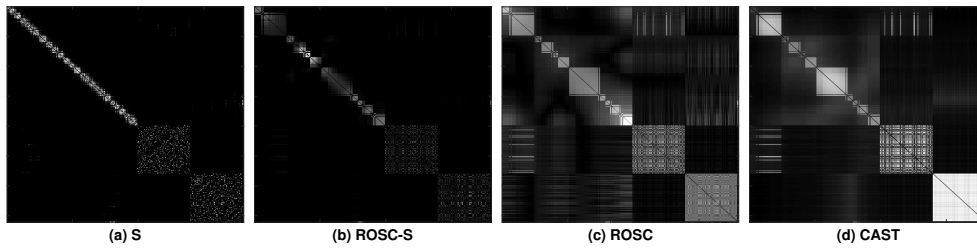
- ROSC and ROSC-S each win in 5 and 6 contests, respectively. We also observe that there are quite a few cases in which their performances differ significantly. For example, ROSC-S beats ROSC 0.8038 to 0.7518 in AMI-*isolet\_5CLASS*, while ROSC outperforms ROSC-S 0.5822 to 0.5497 in Purity-*glass*. This is because ROSC lacks sparsity for inter-cluster connections while ROSC-S loses grouping effect for highly correlated objects. The results thus show that the relative performance of ROSC-S and ROSC varies across datasets. They are thus relatively unstable in their performance.
- CAST provides a more stable performance across the datasets compared with ROSC and ROSC-S. First, CAST wins in 9 contests and ranks 2nd in 5 others. For the case that CAST is not top-2 (e.g., AMI-*glass*), it is the best algorithm among the matrix-reconstruction methods. With regularization using the trace Lasso, CAST takes advantage of both grouping effect and sparsity. It is thus more robust when applied to multi-scale data of different characteristics.

We end this section with a parameter analysis. CAST uses two parameters  $\alpha_1$  and  $\alpha_2$  to control the trace Lasso regularization term and the TKNN graph regularization term, respectively. We fix one parameter and vary the other. Fig. 8 shows the parameter analysis

on the datasets *isolet\_5CLASS* and *Yale\_5CLASS*. From the figure, we see that CAST gives very stable performance over a wide range of parameter values.

## 5 CONCLUSIONS

In this paper, we studied the performance of spectral clustering on data with various sizes and densities. We reviewed existing spectral methods in handling multi-scale data. In particular, we observed that the ROSC algorithm constructs a matrix with grouping effect, but it fails to weaken connections between clusters. We thus proposed the CAST algorithm, which uses trace Lasso to balance the effect of  $\ell_1$  and  $\ell_2$  regularizations. We mathematically proved that the matrix  $\tilde{Z}$  constructed by CAST has grouping effect. We also show that the matrix achieves sparsity for uncorrelated objects. We conducted extensive experiments to evaluate CAST’s performance and compared CAST against other competitors using both synthetic and real datasets. Our experimental results showed that CAST performed very well against its competitors over all the datasets. It is thus robust when applied to multi-scale data of different properties.

Figure 5: Similarity matrix  $S$  (a) and constructed matrices by ROSC-S (b), ROSC (c), and CAST (d) for SYN1Figure 6: Similarity matrix  $S$  (a) and constructed matrices by ROSC-S (b), ROSC (c), and CAST (d) for SYN2

Dataset	NJW	NCuts	PIC	PIC- $k$	DPIC	DPIE	ZP	FUSE	ROSC-S	ROSC	CAST
COIL20	0.4115	0.3926	0.2801	0.2801	0.2361	0.3496	0.5028	0.4177	<b>0.9500</b>	<b>0.9500</b>	<b>0.9500 (1)</b>
glass	0.5234	0.5187	0.4976	0.5029	0.5245	0.5158	0.5374	0.5390	0.5497	<b>0.5822</b>	0.5785 (2)
MNIST0127	0.5066	0.4970	0.4975	0.4924	0.5898	0.4395	0.5066	0.6436	0.6971	0.6776	<b>0.7146 (1)</b>
isolet_5CLASS	0.8120	0.7967	0.5863	0.5867	0.3033	0.8572	0.7767	0.7825	<b>0.8860</b>	0.8253	0.8671 (2)
Yale_5CLASS	0.5273	0.5091	0.4516	0.4596	0.4000	0.5225	0.5091	0.5458	0.5422	0.5693	<b>0.5753 (1)</b>

Table 3: Purity scores, real datasets

Dataset	NJW	NCuts	PIC	PIC- $k$	DPIC	DPIE	ZP	FUSE	ROSC-S	ROSC	CAST
COIL20	0.4718	0.4258	0.2989	0.2781	0.2507	0.3642	0.5702	0.4448	<b>0.9758</b>	<b>0.9758</b>	<b>0.9758 (1)</b>
glass	<b>0.3469</b>	0.3465	0.3162	0.3193	0.2807	0.2683	0.3426	0.2589	0.3245	0.2988	0.3390 (4)
MNIST0127	0.4353	0.4241	0.3623	0.3822	0.3714	0.2059	0.4219	0.4125	0.5243	0.4731	<b>0.5311 (1)</b>
isolet_5CLASS	0.7595	0.7204	0.5280	0.5292	0.0489	0.7481	0.7379	0.6516	<b>0.8038</b>	0.7518	0.7662 (2)
Yale_5CLASS	0.3121	0.3321	0.2357	0.2320	0.1468	0.3305	0.2788	0.3465	0.3218	0.3475	<b>0.3477 (1)</b>

Table 4: AMI scores, real datasets

Dataset	NJW	NCuts	PIC	PIC- $k$	DPIC	DPIE	ZP	FUSE	ROSC-S	ROSC	CAST
COIL20	0.7303	0.6245	0.4940	0.4481	0.7737	0.6114	0.8534	0.7424	<b>0.9938</b>	<b>0.9938</b>	<b>0.9938 (1)</b>
glass	0.6890	0.6880	0.6808	0.6851	0.6556	0.6281	0.6949	0.6693	0.6992	<b>0.7117</b>	0.7022 (2)
MNIST0127	0.5683	0.5459	0.5941	0.5887	0.6598	0.4648	0.6018	0.7022	0.7693	0.7533	<b>0.7867 (1)</b>
isolet_5CLASS	0.9058	0.8942	0.7288	0.7296	0.6792	0.9123	0.8993	0.8695	<b>0.9293</b>	0.9026	0.9132 (2)
Yale_5CLASS	0.7626	0.7519	0.6772	0.6843	0.6846	0.7542	0.7600	0.7363	0.7721	0.7817	<b>0.7833 (1)</b>

Table 5: Rand index scores, real datasets

## 6 ACKNOWLEDGMENTS

This research is supported by Hong Kong Research Grants Council GRF HKU 17254016.

## REFERENCES

- [1] Charles J Alpert and So-Zen Yao. 1995. Spectral partitioning: the more eigenvectors, the better. In *Proceedings of the 32nd annual ACM/IEEE Design Automation Conference*. ACM, 195–200.
- [2] Aleksandar Bojchevski, Yves Matkovic, and Stephan Günnemann. 2017. Robust Spectral Clustering for Noisy Data: Modeling Sparse Corruptions Improves Latent Embeddings. In *KDD*. 737–746.
- [3] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. 2010. A singular value thresholding algorithm for matrix completion. *SIAM Journal on optimization* 20, 4 (2010), 1956–1982.
- [4] Xinlei Chen and Deng Cai. 2011. Large Scale Spectral Clustering with Landmark-Based Representation. In *AAAI*. 313–318.
- [5] Xiaojun Chen, Weijun Hong, Feiping Nie, Dan He, Min Yang, and Joshua Zhexue Huang. 2018. Spectral clustering of large-scale data by directly solving normalized cut. In *KDD*. ACM, 1206–1215.
- [6] Carlos D Correa and Peter Lindstrom. 2012. Locally-scaled spectral clustering using empty region graphs. In *KDD*. 1330–1338.



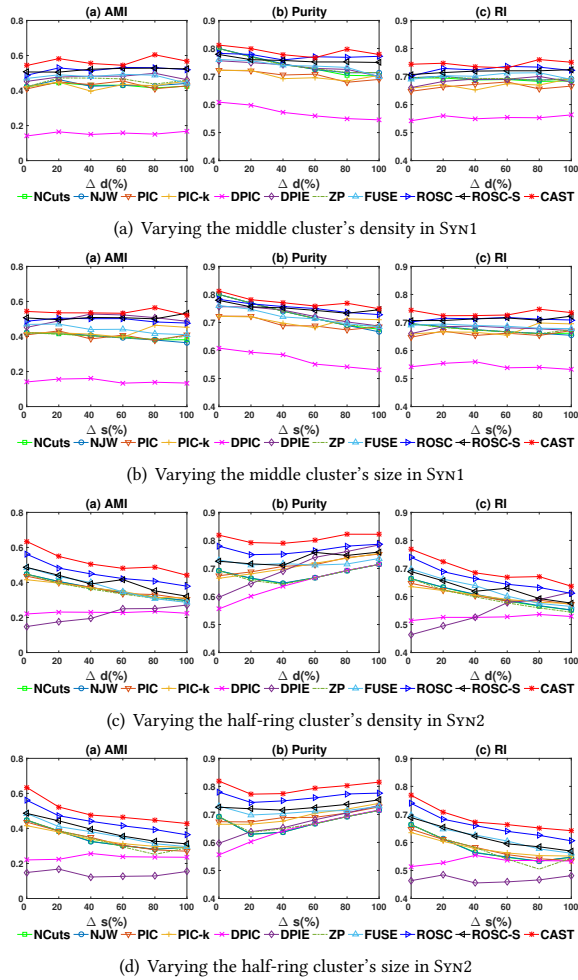


Figure 7: Results vs. varying the clusters in SYN1 and SYN2

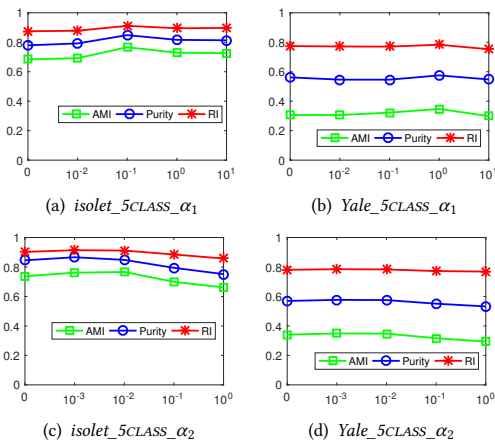


Figure 8: CAST's performance scores vs.  $\alpha_1$  and  $\alpha_2$

[7] Jane K Cullum and Ralph A Willoughby. 2002. *Lanczos algorithms for large symmetric eigenvalue computations: Vol. I: Theory*. SIAM.

[8] Inderjit S Dhillon. 2001. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD*. 269–274.

[9] Edouard Grave, Guillaume R Obozinski, and Francis R Bach. 2011. Trace lasso: a trace norm regularization for correlated designs. In *NeurIPS*. 2187–2195.

[10] Han Hu, Zhouchen Lin, Jianjiang Feng, and Jie Zhou. 2014. Smooth representation clustering. In *CVPR*. 3834–3841.

[11] Hao Huang, Shinjae Yoo, Dantong Yu, and Hong Qin. 2014. Diverse power iteration embeddings and its applications. In *ICDM*. 200–209.

[12] Ling Huang, Donghui Yan, Nina Taft, and Michael I Jordan. 2009. Spectral clustering with perturbed data. In *NeurIPS*. 705–712.

[13] Ravi Kannan, Santosh Vempala, and Adrian Vetta. 2004. On clusterings: Good, bad and spectral. *JACM* 51, 3 (2004), 497–515.

[14] Agnan Kessy, Alex Lewin, and Korbinian Strimmer. 2017. Optimal whitening and decorrelation. *The American Statistician* (2017).

[15] Stephane Lafon and Ann B Lee. 2006. Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *TPAMI* 28, 9 (2006), 1393–1403.

[16] Xiang Li, Ben Kao, Siqiang Luo, and Martin Ester. 2018. ROSC: robust spectral clustering on multi-scale data. In *WWW*. 157–166.

[17] Xiang Li, Ben Kao, Zhaochun Ren, and Dawei Yin. 2019. Spectral Clustering in Heterogeneous Information Networks. In *AAAI*. 4221–4228.

[18] Zhenguo Li, Jianzhuang Liu, Shifeng Chen, and Xiaoou Tang. 2007. Noise robust spectral clustering. In *ICCV*. 1–8.

[19] Frank Lin. 2012. *Scalable methods for graph-based unsupervised and semi-supervised learning*. Ph.D. Dissertation. Carnegie Mellon University.

[20] Frank Lin and William W Cohen. 2010. Power iteration clustering. In *ICML*. 655–662.

[21] Zhouchen Lin, Minming Chen, and Yi Ma. 2010. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055* (2010).

[22] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. 2013. Robust recovery of subspace structures by low-rank representation. *TPAMI* 35, 1 (2013), 171–184.

[23] Canyi Lu, Jiashi Feng, Zhouchen Lin, and Shuicheng Yan. 2013. Correlation adaptive subspace segmentation by trace lasso. In *ICCV*. 1345–1352.

[24] Can-Yi Lu, Hai Min, Zhong-Qiu Zhao, Lin Zhu, De-Shuang Huang, and Shuicheng Yan. 2012. Robust and efficient subspace segmentation via least squares regression. In *ECCV*. 347–360.

[25] Marina Meila and Jianbo Shi. 2001. A random walks view of spectral segmentation. (2001).

[26] Boaz Nadler and Meirav Galun. 2006. Fundamental limitations of spectral clustering. In *NeurIPS*. 1017–1024.

[27] Boaz Nadler, Stephane Lafon, Ronald Coifman, and Ioannis Kevrekidis. 2005. Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators. In *NeurIPS*. 955–962.

[28] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. 2001. On spectral clustering: Analysis and an algorithm. In *NeurIPS*. 849–856.

[29] Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. *TPAMI* 22, 8 (2000), 888–905.

[30] Anh Pham The, Nguyen Duc Thang, La The Vinh, Young-Koo Lee, and Sungyoung Lee. 2013. Deflation-based power iteration clustering. *Applied Intelligence* 39, 2 (2013), 367–385.

[31] Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2010. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *JMLR* 11 (2010), 2837–2854.

[32] Ulrike Von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and computing* 17, 4 (2007), 395–416.

[33] Ulrike Von Luxburg, Mikhail Belkin, and Olivier Bousquet. 2008. Consistency of spectral clustering. *The Annals of Statistics* (2008), 555–586.

[34] Tao Xiang and Shaogang Gong. 2008. Spectral clustering with eigenvector selection. *Pattern Recognition* 41, 3 (2008), 1012–1029.

[35] Donghui Yan, Ling Huang, and Michael I Jordan. 2009. Fast approximate spectral clustering. In *KDD*. 907–916.

[36] Wei Ye, Sebastian Goebel, Claudia Plant, and Christian Böhm. 2016. FUSE: Full Spectral Clustering. In *KDD*. 1985–1994.

[37] Stella X. Yu and Jianbo Shi. 2003. Multiclass spectral clustering. In *ICCV*. 313–319.

[38] Lihi Zelnik-Manor and Pietro Perona. 2004. Self-tuning spectral clustering. In *NeurIPS*. 1601–1608.

[39] Yin Zhang. 2010. Recent advances in alternating direction methods: Practice and theory. In *IPAM workshop on continuous optimization*.

[40] Xiatian Zhu, Chen Change Loy, and Shaogang Gong. 2014. Constructing robust affinity graphs for spectral clustering. In *CVPR*. 1450–1457.

## A EXPERIMENT

### A.1 Dataset statistics

We summarize the statistics of datasets used in our experiments in Table 6.

Dataset	#objects	#dimensions	#clusters
COIL20	1,440	1,024	20
glass	214	9	6
MNIST0127	1,666	784	4
isolet_5CLASS	300	617	5
Yale_5CLASS	55	1024	5

Table 6: Statistics of 5 real datasets

### A.2 Experiment settings

For all the datasets, the similarity matrix  $S$  is computed based on Euclidean distance of objects' attributes.  $S$  is also locally scaled as is done in ZP. All the methods employ  $k$ -means as the last step of the clustering pipeline to return clusters. For this step, we run  $k$ -means 100 times with random starting centroids and the most frequent cluster assignment is used [20]. For ROSC, ROSC-S and CAST, we set  $K = 4$  in constructing the TKNN graph as suggested in [16], and fine tune the parameters by grid search for  $\alpha_1, \alpha_2 \in \{0.001, 0.01, 0.1, 1, 10\}$  to report the best results. For other methods, parameters are set according to their original papers. For each method and dataset, we run the experiment 50 times and report average results. Our codes and datasets are publicly available at <https://github.com/lixiang3776/CAST>.

## B ALGORITHMS

We give the details of the pseudocodes. Algorithm 1 and 2 introduce solving Eq. 3 and Eq. 12 by inexact ALM, respectively. Algorithm 3 summarizes CAST.

## C PROOF

In this section we prove Theorem 4. We first consider two lemmas.

LEMMA 6. Given  $\mathbf{z} \in \mathbb{R}^n$ ,  $X \in \mathbb{R}^{d \times n}$ ,  $\|X \text{Diag}(\mathbf{z})\|_* \leq \|X\|_F \|\mathbf{z}\|_2$ .

LEMMA 7. If  $\eta_i \geq \mu_i \geq 0$ ,  $i = 1, \dots, n$ , and  $C = \sum_{i=1}^n (\eta_i - \mu_i)$ , then  $\sum_{i=1}^n \sqrt{\eta_i} \geq \sum_{i=1}^n \sqrt{\mu_i} + \frac{C}{2\sqrt{\max\{\eta_i\}}}$ .

For proofs of both lemmas, see [23]. Next, we prove Theorem 4. Let  $\hat{X}_{\hat{\mathbf{z}}} = \hat{X} \text{Diag}(\hat{\mathbf{z}})$ . We get

$$f([\hat{\mathbf{z}}; \tilde{\mathbf{z}}]) = \frac{1}{2} \|\mathbf{x} - \hat{X}\hat{\mathbf{z}} - \tilde{X}\tilde{\mathbf{z}}\|_2^2 + \alpha_1 \|\hat{X}_{\hat{\mathbf{z}}} \tilde{X} \text{Diag}(\tilde{\mathbf{z}})\|_* + \frac{\alpha_2}{2} \|[\hat{\mathbf{z}}; \tilde{\mathbf{z}}] - \mathbf{w}\|_2^2.$$

Rewrite  $f([\hat{\mathbf{z}}; \tilde{\mathbf{z}}]) = \Omega_1 + \Omega_2 + \Omega_3$ , where  $\Omega_1 = \frac{1}{2} \|\mathbf{x} - \hat{X}\hat{\mathbf{z}} - \tilde{x}_0 \mathbf{1}^T \tilde{\mathbf{z}} + (\tilde{x}_0 \mathbf{1}^T \tilde{\mathbf{z}} - \tilde{X}\tilde{\mathbf{z}})\|_2^2$ ,  $\Omega_2 = \alpha_1 \|\hat{X}_{\hat{\mathbf{z}}} \tilde{x}_0 \mathbf{1}^T \text{Diag}(\tilde{\mathbf{z}})\| + [0 (\tilde{X} - \tilde{x}_0 \mathbf{1}^T) \text{Diag}(\tilde{\mathbf{z}})]\|_*$  and  $\Omega_3 = \frac{\alpha_2}{2} \|\hat{\mathbf{z}} - \tilde{\mathbf{w}}\|_2^2 + \frac{\alpha_2}{2} \|(\tilde{\mathbf{z}} - \tilde{\mathbf{w}})\|_2^2$ . Since  $\mathbf{y}^*$  is the optimal solution to the problem:  $\min_{\mathbf{y}} \|\mathbf{y} - \tilde{\mathbf{w}}\|_2^2$ , s.t.  $\mathbf{y}^T \mathbf{1} = q\tilde{z}$ , we have

$$\begin{aligned} \|\tilde{\mathbf{z}} - \tilde{\mathbf{w}}\|_2^2 &\geq \|\mathbf{y}^* - \tilde{\mathbf{w}}\|_2^2 - \|\tilde{\mathbf{z}}\mathbf{1} - \tilde{\mathbf{w}}\|_2^2 + \|\tilde{\mathbf{z}}\mathbf{1} - \tilde{\mathbf{w}}\|_2^2 \\ &= \sum_{j=1}^q [(y_j^* - \tilde{z})(y_j^* + \tilde{z} - 2\tilde{w}_j)] + \|\tilde{\mathbf{z}}\mathbf{1} - \tilde{\mathbf{w}}\|_2^2. \end{aligned}$$

---

### Algorithm 1 Solving Eq. 3 by inexact ALM

---

**Input:**  $X, \mathcal{W}, k, \rho, \mu_{\max}, \epsilon$

**Output:**  $Z$

- 1: Initialize  $J, Z, Y, \mu$
  - 2: **while**  $\|J - Z + \text{Diag}(Z)\|_{\infty} > \epsilon$  **do**
  - 3:   Update  $J$  by Eq. 5 with the others fixed
  - 4:   Update  $Z$  by Eq. 6 with the others fixed
  - 5:   Update the multiplier  $Y = Y + \mu(J - Z + \text{Diag}(Z))$
  - 6:   Update  $\mu = \min(\rho\mu, \mu_{\max})$
  - 7: **end while**
  - 8: **return**  $Z$
- 

---

### Algorithm 2 Solving Eq. 12 by inexact ALM

---

**Input:**  $\mathbf{x}, X, \mathbf{w}, k, \rho, \mu_{\max}, \epsilon$

**Output:**  $\mathbf{z}$

- 1: Initialize  $J, \mathbf{z}, \mathbf{e}, \mathbf{h}, \lambda_1, \lambda_2, Y, \mu$
  - 2: **while**  $\|\mathbf{e} - \mathbf{x} + X\mathbf{z}\|_{\infty} > \epsilon$  or  $\|\mathbf{h} - \mathbf{z} + \mathbf{w}\|_{\infty} > \epsilon$  or  $\|J - X \text{Diag}(\mathbf{z})\|_{\infty} > \epsilon$  **do**
  - 3:   Update  $\mathbf{z}$  by Eq. 14 with other variables fixed
  - 4:   Update  $\mathbf{e}$  by Eq. 15 with other variables fixed
  - 5:   Update  $\mathbf{h}$  by Eq. 16 with other variables fixed
  - 6:   Update  $J$  by Eq. 18 with other variables fixed
  - 7:   Update the multiplier  $\lambda_1 = \lambda_1 + \mu(\mathbf{e} - \mathbf{x} + X\mathbf{z})$
  - 8:   Update the multiplier  $\lambda_2 = \lambda_2 + \mu(\mathbf{h} - \mathbf{z} + \mathbf{w})$
  - 9:   Update the multiplier  $Y = Y + \mu(J - X \text{Diag}(\mathbf{z}))$
  - 10:   Update  $\mu = \min(\rho\mu, \mu_{\max})$
  - 11: **end while**
  - 12: **return**  $\mathbf{z}$
- 

---

### Algorithm 3 CAST

---

**Input:**  $S, k$ .

**Output:**  $C = \{C_1, \dots, C_k\}$

- 1: Compute the TKNN graph and the weight matrix  $\mathcal{W}$
  - 2: Calculate  $W = D^{-1}S$ , where  $D_{ii} = \sum_j S_{ij}$
  - 3: Apply PI on  $W$  and generate  $p$  pseudo-eigenvectors  $\{\mathbf{v}_r\}_{r=1}^p$
  - 4:  $X = \{\mathbf{v}_1^T; \mathbf{v}_2^T; \dots; \mathbf{v}_p^T\}$ ;  $X = \text{whiten}(X)$
  - 5: Normalize each column vector  $\mathbf{x}$  of  $X$  such that  $\mathbf{x}^T \mathbf{x} = 1$
  - 6: **for**  $i = 1$  to  $n$  **do**
  - 7:   Solve Eq. 12 for an object  $x_i$  by inexact ALM and get  $\mathbf{z}_i^*$
  - 8: **end for**
  - 9: Calculate the coefficient matrix  $Z^* = [\mathbf{z}_1^*, \dots, \mathbf{z}_n^*]$
  - 10: Construct  $\check{Z} = (|Z^*| + |(Z^*)^T|)/2$
  - 11: Run NCuts on  $\check{Z}$  to obtain clusters  $C = \{C_r\}_{r=1}^k$
  - 12: **return**  $C = \{C_1, \dots, C_k\}$
- 

Let  $\Omega_4 = \sum_{j=1}^q [(y_j^* - \tilde{z})(y_j^* + \tilde{z} - 2\tilde{w}_j)]$ . Since  $\|\mathbf{y}^* - \tilde{\mathbf{w}}\|_2^2$  is the minimum value, we have  $\Omega_4 \leq 0$  and  $\|\tilde{\mathbf{z}} - \tilde{\mathbf{w}}\|_2^2 \geq \|\tilde{\mathbf{z}}\mathbf{1} - \tilde{\mathbf{w}}\|_2^2 + \Omega_4$ . We derive lower bounds for  $\Omega_1$ ,  $\Omega_2$  and  $\Omega_3$ :

$$\begin{aligned} \Omega_1 &\geq \frac{1}{2} \|\mathbf{x} - \hat{X}\hat{\mathbf{z}} - \tilde{x}_0 \mathbf{1}^T \tilde{\mathbf{z}}\|_2^2 - \|\mathbf{x} - \hat{X}\hat{\mathbf{z}} - \tilde{x}_0 \mathbf{1}^T \tilde{\mathbf{z}}\|_2 \|\tilde{x}_0 \mathbf{1}^T - \tilde{X}\|_2 \|\tilde{\mathbf{z}}\|_2 \\ &\geq \frac{1}{2} \|\mathbf{x} - \hat{X}\hat{\mathbf{z}} - \tilde{x}_0 \mathbf{1}^T \tilde{\mathbf{z}}\|_2^2 - \|\mathbf{x} - \hat{X}\hat{\mathbf{z}} - \tilde{x}_0 \mathbf{1}^T \tilde{\mathbf{z}}\|_2 \|\tilde{x}_0 \mathbf{1}^T - \tilde{X}\|_2 \|\tilde{\mathbf{z}}\|_2 \end{aligned} \quad (19)$$

Based on Lemma 6,

$$\begin{aligned} \Omega_2 &\geq \alpha_1 \|[\hat{X}_{\hat{z}} \bar{\mathbf{x}}_0 \mathbf{1}^T \text{Diag}(\hat{z})]\|_* - \alpha_1 \|(\tilde{X} - \bar{\mathbf{x}}_0 \mathbf{1}^T) \text{Diag}(\tilde{z})\|_* \\ &\geq \alpha_1 \|[\hat{X}_{\hat{z}} \bar{\mathbf{x}}_0 \tilde{z}^T]\|_* - \alpha_1 \|\tilde{z}\|_2 \|\tilde{X} - \bar{\mathbf{x}}_0 \mathbf{1}^T\|_F, \end{aligned} \quad (20)$$

$$\Omega_3 \geq \frac{\alpha_2}{2} \|[\hat{z}; \bar{\mathbf{z}}\mathbf{1}]\|_2^2 + \frac{\alpha_2}{2} \Omega_4 \quad (21)$$

Combining Eqs. 19-21, we have,

$$\begin{aligned} f([\hat{z}; \tilde{z}]) &\geq \frac{1}{2} \|\mathbf{x} - \hat{X}\hat{z} - \tilde{X}(\bar{\mathbf{z}}\mathbf{1})\|_2^2 - (\alpha_1 + \|\mathbf{x} - \hat{X}\hat{z} - \tilde{X}(\bar{\mathbf{z}}\mathbf{1})\|_2) \|\tilde{z}\|_2 \epsilon \\ &\quad + \alpha_1 \|[\hat{X}_{\hat{z}} \bar{\mathbf{x}}_0 \tilde{z}^T]\|_* + \frac{\alpha_2}{2} \|[\hat{z}; \bar{\mathbf{z}}\mathbf{1}]\|_2^2 + \frac{\alpha_2}{2} \Omega_4 \end{aligned}$$

Let  $\Omega_5 = \frac{1}{2} \|\mathbf{x} - \hat{X}\hat{z} - \tilde{X}(\bar{\mathbf{z}}\mathbf{1})\|_2^2 - (\alpha_1 + \|\mathbf{x} - \hat{X}\hat{z} - \tilde{X}(\bar{\mathbf{z}}\mathbf{1})\|_2) \|\tilde{z}\|_2 \epsilon$  and  $\Omega_6 = \frac{\alpha_2}{2} \|[\hat{z}; \bar{\mathbf{z}}\mathbf{1}]\|_2^2 + \frac{\alpha_2}{2} \Omega_4$ , we have,

$$f([\hat{z}; \tilde{z}]) \geq \Omega_5 + \alpha_1 \|[\hat{X}_{\hat{z}} \bar{\mathbf{x}}_0 \tilde{z}^T]\|_* + \Omega_6. \quad (22)$$

Let  $Y = \hat{X}_{\hat{z}} \hat{X}_{\hat{z}}^T$  and  $\lambda_i(M)$  denote the  $i$ -th largest eigenvalue of a matrix  $M$ . We have,

$$\begin{aligned} \sum_{i=1}^d \lambda_i(Y + \|\tilde{z}\|_2^2 \bar{\mathbf{x}}_0 \bar{\mathbf{x}}_0^T) &= \text{tr}(Y + \|\tilde{z}\|_2^2 \bar{\mathbf{x}}_0 \bar{\mathbf{x}}_0^T) \\ &= \text{tr}(Y + \|\bar{\mathbf{z}}\mathbf{1}\|_2^2 \bar{\mathbf{x}}_0 \bar{\mathbf{x}}_0^T) + \text{tr}((\|\tilde{z}\|_2^2 - \|\bar{\mathbf{z}}\mathbf{1}\|_2^2) \bar{\mathbf{x}}_0 \bar{\mathbf{x}}_0^T) \\ &= \sum_{i=1}^d \lambda_i(Y + \|\bar{\mathbf{z}}\mathbf{1}\|_2^2 \bar{\mathbf{x}}_0 \bar{\mathbf{x}}_0^T) + (\|\tilde{z}\|_2^2 - \|\bar{\mathbf{z}}\mathbf{1}\|_2^2) \|\bar{\mathbf{x}}_0\|_2^2 \end{aligned} \quad (23)$$

Since  $\mathbf{1}^T \tilde{z} = q\bar{z}$ , we get  $\|\tilde{z}\|_2^2 \geq \|\bar{\mathbf{z}}\mathbf{1}\|_2^2$  and  $\lambda_i(Y + \|\tilde{z}\|_2^2 \bar{\mathbf{x}}_0 \bar{\mathbf{x}}_0^T) \geq \lambda_i(Y + \|\bar{\mathbf{z}}\mathbf{1}\|_2^2 \bar{\mathbf{x}}_0 \bar{\mathbf{x}}_0^T) \geq 0$ . Moreover,  $\|\tilde{z}\|_2^2 - \|\bar{\mathbf{z}}\mathbf{1}\|_2^2 = \|\tilde{z} - \bar{\mathbf{z}}\mathbf{1}\|_2^2$ . Based

on Eq. 23 and Lemma 7, we get

$$\begin{aligned} \|[\hat{X}_{\hat{z}} \bar{\mathbf{x}}_0 \tilde{z}^T]\|_* &= \sum_{i=1}^d \sqrt{\lambda_i(Y + \|\tilde{z}\|_2^2 \bar{\mathbf{x}}_0 \bar{\mathbf{x}}_0^T)} \\ &\geq \sum_{i=1}^d \sqrt{\lambda_i(Y + \|\bar{\mathbf{z}}\mathbf{1}\|_2^2 \bar{\mathbf{x}}_0 \bar{\mathbf{x}}_0^T)} + \frac{\|\tilde{z} - \bar{\mathbf{z}}\mathbf{1}\|_2^2 \|\bar{\mathbf{x}}_0\|_2^2}{2\sqrt{\lambda_1(Y + \|\tilde{z}\|_2^2 \bar{\mathbf{x}}_0 \bar{\mathbf{x}}_0^T)}} \\ &\geq \|[\hat{X}_{\hat{z}} \bar{\mathbf{z}}\bar{\mathbf{x}}_0 \mathbf{1}^T]\|_* + \frac{\|\bar{\mathbf{x}}_0\|_2^2}{2\|[\hat{X}_{\hat{z}} \bar{\mathbf{x}}_0 \tilde{z}^T]\|_2} \delta^2 \end{aligned} \quad (24)$$

Moreover,

$$\begin{aligned} \|[\hat{X}_{\hat{z}} \bar{\mathbf{z}}\bar{\mathbf{x}}_0 \mathbf{1}^T]\|_* &= \|[\hat{X}_{\hat{z}} \tilde{X} \text{Diag}(\bar{\mathbf{z}}\mathbf{1})] + [0 (\bar{\mathbf{z}}\bar{\mathbf{x}}_0 \mathbf{1}^T - \tilde{X} \text{Diag}(\bar{\mathbf{z}}\mathbf{1}))]\|_* \\ &\geq \|[\hat{X}_{\hat{z}} \tilde{X} \text{Diag}(\bar{\mathbf{z}}\mathbf{1})]\|_* - |\bar{z}| \|\bar{\mathbf{x}}_0 \mathbf{1}^T - \tilde{X}\|_* \\ &\geq \|[\hat{X}_{\hat{z}} \tilde{X} \text{Diag}(\bar{\mathbf{z}}\mathbf{1})]\|_* - |\bar{z}| \epsilon. \end{aligned} \quad (25)$$

Substituting Eq. 24 and 25 into Eq. 22, we have

$$\begin{aligned} f([\hat{z}; \tilde{z}]) &\geq \Omega_5 + \alpha_1 (\|[\hat{X}_{\hat{z}} \tilde{X} \text{Diag}(\bar{\mathbf{z}}\mathbf{1})]\|_* - |\bar{z}| \epsilon) + \frac{\|\bar{\mathbf{x}}_0\|_2^2}{2\|[\hat{X}_{\hat{z}} \bar{\mathbf{x}}_0 \tilde{z}^T]\|_2} \delta^2 + \Omega_6 \\ &= f([\hat{z}; \bar{\mathbf{z}}\mathbf{1}]) + \left( \frac{\alpha_1 \|\bar{\mathbf{x}}_0\|_2^2}{2\|[\hat{X}_{\hat{z}} \bar{\mathbf{x}}_0 \tilde{z}^T]\|_2} \right) \delta^2 + \frac{\alpha_2}{2} \Omega_4 - \Omega_7 \end{aligned} \quad (26)$$

where  $\Omega_7 = ((\alpha_1 + \|\mathbf{x} - \hat{X}\hat{z} - \tilde{X}(\bar{\mathbf{z}}\mathbf{1})\|_2) \|\tilde{z}\|_2 + \alpha_1 |\bar{z}|) \epsilon$ . From Eq. 26, we see that if  $\|\tilde{z} - \bar{\mathbf{z}}\mathbf{1}\|_2 > \delta$ ,  $f([\hat{z}; \tilde{z}]) > f([\hat{z}; \bar{\mathbf{z}}\mathbf{1}])$ , where

$$\delta = \sqrt{\frac{(2\Omega_7 - \alpha_2 \Omega_4) (\|[\hat{X}_{\hat{z}} \bar{\mathbf{x}}_0 \tilde{z}^T]\|_2)}{\alpha_1 \|\bar{\mathbf{x}}_0\|_2^2}}.$$