Robust Subspace Segmentation by Simultaneously Learning Data Representations and Their Affinity Matrix

Xiaojie Guo
State Key Laboratory of Information Security
Institute of Information Engineering, Chinese Academy of Sciences
xj.max.guo@gmail.com

Abstract

The goal of subspace segmentation is to partition a set of data drawn from a union of subspace into their underlying subspaces. The performance of spectral clustering based approaches heavily depends on learned data affinity matrices, which are usually constructed either directly from the raw data or from their computed representations. In this paper, we propose a novel method to simultaneously learn the representations of data and the affinity matrix of representation in a unified optimization framework. A novel Augmented Lagrangian Multiplier based algorithm is designed to effectively and efficiently seek the optimal solution of the problem. The experimental results on both synthetic and real data demonstrate the efficacy of the proposed method and its superior performance over the state-of-the-art alternatives.

1 Introduction

In scientific data analysis applications, we often have to face a set of data $\boldsymbol{X} = [\boldsymbol{X}_1, ..., \boldsymbol{X}_n] \in \mathbb{R}^{d\times n}$ derived from a union of c subspaces $\{\mathcal{S}_s\}_{s=1}^{c}$, where d is the feature dimension and n is the number of data vectors. To characterize the given data as different groups such that the data in the same group are highly similar to each other (ideally drawn from one subspace), the subspace segmentation recently has been focus of considerable research in machine learning, computer vision and pattern recognition [Hong *et al.*, 2006; Rao *et al.*, 2010; Ho *et al.*, 2003; Liu *et al.*, 2013; Nie *et al.*, 2009].

1.1 Notation Summary

Lowercase letters $(u, v, ...)$ mean scalars and bold lowercase letters $(\boldsymbol{u}, \boldsymbol{v}, ...)$ vectors. $\boldsymbol{u}_j$ represents the j^{th} entry in $\boldsymbol{u}$. Bold uppercase letters $(\boldsymbol{U}, \boldsymbol{V}, ...)$ stand for matrices. $\boldsymbol{U}^T$ and $\boldsymbol{U}^{-1}$ are the transpose and inverse of $\boldsymbol{U}$, respectively. $\boldsymbol{U}_i$ stands for the i^{th} column of $\boldsymbol{U}$, while $\boldsymbol{U}_{ij}$ the j^{th} element in the i^{th} column of $\boldsymbol{U}$. $|\boldsymbol{U}_{ij}|$ is the absolute value of $\boldsymbol{U}_{ij}$. $\|\boldsymbol{U}\|_0$, $\|\boldsymbol{U}\|_1$, $\|\boldsymbol{U}\|_F$ and $\|\boldsymbol{U}\|_*$ denote the ℓ^0 norm (number of nonzero entries), ℓ^1 norm ($\sum_{i,j}|\boldsymbol{U}_{i,j}|$), ℓ^2 or Frobenius norm ($\sqrt{\sum_{i,j}\boldsymbol{U}_{i,j}^2}$), and nuclear norm (sum of all the singular values) of $\boldsymbol{U}$, respectively. $\|\boldsymbol{U}\|_{2,0}$ and $\|\boldsymbol{U}\|_{2,1}$ stand for the $\ell^{2,0}$ norm ($\sum_i \|\sqrt{\sum_j \boldsymbol{U}_{ij}^2}\|_0$) and $\ell^{2,1}$ norm ($\sum_i \sqrt{\sum_j \boldsymbol{U}_{ij}^2}$), respectively. $\langle \boldsymbol{U}, \boldsymbol{V} \rangle$ is the inner product of two matrices with identical size, which is equal to the trace of $\boldsymbol{U}^T\boldsymbol{V}$, *i.e.* $\mathrm{tr}(\boldsymbol{U}^T\boldsymbol{V})$. $\boldsymbol{U}\odot\boldsymbol{V}$ presents the Hadamard product of two matrices with identical size. Moreover, $\boldsymbol{0}$, $\boldsymbol{1}$ and $\boldsymbol{I}$ denote the vectors of all zeros, all ones and identity matrix with compatible sizes, respectively.

1.2 Related Work

Recently, many subspace segmentation methods have been proposed. From the perspective of their mechanisms of representing the subspaces, existing approaches can be roughly divided into four categories: iterative [Bradley and Mangasarian, 2000], statistical [Ma *et al.*, 2007], algebraic [Ma *et al.*, 2008; Vidal *et al.*, 2005] and spectral clustering methods [Lu *et al.*, 2012; Elhamifar and Vidal, 2009]. An elaborate review of these methods can be found in [Vidal, 2010]. Our method belongs to the spectral clustering based one, therefore we review the related work along this direction in the following.

The key of spectral clustering based approaches is to construct a "good" affinity matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, in which each element $\boldsymbol{A}_{ij}$ reflects the similarity between data points $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$. Ideally, the affinity should be 1 if they are from the same cluster, 0 otherwise. Directly computing distances on the raw data (*e.g.* k-NN using cosine or heat kernel distances) is possibly the most intuitive way to conduct the data affinity matrix. Nie *et al.* [Nie *et al.*, 2014] develop a more sophisticated method to learn the similarity matrix by adaptively assigning neighbors for each data point based on the local connectivity. *But, the affinity matrix constructed on the raw data is unable to well reveal the global subspace structure of data.*

Alternatively, inspired by the success of compressed sensing [Candès *et al.*, 2006; Donoho, 2006], a large body of research on exploiting the relationship of data representations $\boldsymbol{R} \in \mathbb{R}^{n\times n}$ has been carried out [Elhamifar and Vidal, 2009; Lu *et al.*, 2012; Liu *et al.*, 2013; Lu *et al.*, 2013; Saha *et al.*, 2013; Feng *et al.*, 2014], the formulation of which can be generally written as follows:

$$\min \Theta(\boldsymbol{E}) + \lambda\Psi(\boldsymbol{R}) \quad \text{s.t.}\, \boldsymbol{X} = \boldsymbol{X}\boldsymbol{R} + \boldsymbol{E}, \qquad (1)$$

where $\boldsymbol{E} \in \mathbb{R}^{d\times n}$ denotes the residual, $\Psi(\boldsymbol{R})$ stands for the regularizer on $\boldsymbol{R}$, and λ is the coefficient controlling the im-

portance of the regularizer. $\Theta(\boldsymbol{E})$ is the model of $\boldsymbol{E}$, which can be with different forms depending on the characteristic of data. For instance $\|\boldsymbol{E}\|_1$ is optimal for Laplacian residuals while $\|\boldsymbol{E}\|_F^2$ for Gaussian. The choice of the model is important for the task of subspace segmentation, but not the main focus of this work. Hence, we simply adopt the ℓ^2 for the rest of the paper as [Elhamifar and Vidal, 2009; Lu *et al.*, 2012; 2013; Nie *et al.*, 2014].

The main difference among the methods mentioned above lies in the regularization on $\boldsymbol{R}$. Specifically, Sparse Subspace Clustering (SSC: $\Psi(\boldsymbol{R}) \equiv \|\boldsymbol{R}\|_0$) [Elhamifar and Vidal, 2009] introduces compressed sensing techniques to subspace segmentation. Because the ℓ^0 norm is non-convex and NP-hard to approximate, replacing it with its tightest convex surrogate ℓ^1 norm makes the problem tractable and gives the optimal solution to the original problem under some condition. The main drawback of SSC is that it processes data individually and thus lacks optimality due to the existence of inherent joint structure between the representations of data points. Instead of a sparse representation, Liu *et al.* [Liu *et al.*, 2013] propose the Low Rank Representation (LRR: $\Psi(\boldsymbol{R}) \equiv \mathrm{rank}(\boldsymbol{R})$) to jointly find a low rank representation by minimizing the rank of $\boldsymbol{R}$. As the $\mathrm{rank}(\cdot)$ is also intractable to directly optimize, its convex replacement, the nuclear norm, is employed. Least Squares Regression (LSR: $\Psi(\boldsymbol{R}) \equiv \|\boldsymbol{R}\|_F^2$) [Lu *et al.*, 2012] is a much more efficient solver for subspace segmentation than LRR with a similar grouping effect. To further refine the representation, Feng *et al.* [Feng *et al.*, 2014] impose a block diagonal prior on the representation, which shows a reasonable improvement on the segmentation results. To simultaneously take into account the grouping effect and sparsity of representation, Grouping Sparse Coding (GSC: $\Psi(\boldsymbol{R}) \equiv \|\boldsymbol{R}\|_{2,0}$) [Saha *et al.*, 2013] was developed. For efficiently solving the GSC problem, the $\ell^{2,0}$ norm needs to be convex relaxed to $\ell^{2,1}$. Besides, Correlation Adaptive Subspace Segmentation (CASS: $\Psi(\boldsymbol{R}) \equiv \sum_{i=1}^{n} \|\boldsymbol{X}\,\mathrm{Diag}(\boldsymbol{R}_i)\|_*$) is designed [Lu *et al.*, 2013] to better explore the subspace structure, which can be viewed as an adaptive interpolation between SSC and LSR. However, the computational load of CASS is relatively heavy as it involves a series of SVD operations for dealing with every single data.

Traditionally, after solving the problem (1), the representation is utilized to define the affinity matrix of an undirected graph in the way of $\frac{|\boldsymbol{R}_{ij}|+|\boldsymbol{R}_{ji}|}{2}$ for the data vectors $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$, then spectral clustering algorithms such as Normalized Cuts [Shi and Malik, 2000] are employed to segment the data into c clusters. *Although this kind of affinity is somehow valid, the meaning of which is already not the same as the original definition.* In this paper, we propose a method to construct a meaningful affinity matrix under the assumption that the data points should have a larger probability to be in the same cluster if their representations have a smaller distance.

1.3 Contribution

The contribution of this paper can be summarized in three aspects. 1) We propose a novel subspace segmentation method to jointly learn the representations of data and their affinity matrix in a unified optimization framework. 2) We design a new Augmented Lagrange Multiplier based algorithm to efficiently and effectively seek the solution of the associated optimization problem. 3) To demonstrate the efficacy and the superior performance of the proposed algorithm over the state-of-the-art alternatives, extensive experiments on both synthetic data and several datasets are conducted.

2 Problem Formulation

Given a set of clean data points sufficiently sampled from c independent subspaces $\{\mathcal{S}_{s=1}^c\}$, $\boldsymbol{X} = [\boldsymbol{X}_1, ..., \boldsymbol{X}_n]$. By introducing a hypothesized permutation matrix Γ that arranges the data to the true segmentation of data, we have $\boldsymbol{X}^* = \boldsymbol{X}\Gamma = [\boldsymbol{X}^1, ..., \boldsymbol{X}^c] \in \mathbb{R}^{d\times n}$, where $\boldsymbol{X}^s$ denotes a collection of n_s data points from the s^{th} subspace $\mathcal{S}_s$ with $n = \sum_{s=1}^{c} n_s$. The data can be self-represented by a linear combination of the items in $\boldsymbol{X}$ as $\boldsymbol{X} = \boldsymbol{X}\boldsymbol{R}$. To avoid the trivial solution, we should impose some constraint on $\boldsymbol{R}$. Considering the simplicity and the effectiveness leads us to choose the ℓ^2 norm $\|\boldsymbol{R}\|_F^2$ to do the job, any other choices can, of course, be selected. Recall our assumption that the data points should have a larger probability to be in the same cluster if their representations have a smaller distance, thus we naturally propose the following constraint:

$$\min_{\forall i\ \boldsymbol{A}_i^T\mathbf{1}=1, \boldsymbol{A}_i\succeq 0} \sum_{i=1}^{n}\sum_{j=1}^{n} \|\boldsymbol{R}_i - \boldsymbol{R}_j\|_F^2 \boldsymbol{A}_{ij}, \tag{2}$$

where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is the desired affinity matrix, $\boldsymbol{A}_{ij}$ reflects the probability of the data points $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ from the same cluster based on the distance between their representations $\boldsymbol{R}_i$ and $\boldsymbol{R}_j$. The constraints $\boldsymbol{A}_i^T\mathbf{1} = 1$ and $\boldsymbol{A}_i \succeq 0$ are to guarantee the probability property of $\boldsymbol{A}_i$. However, simply solving the problem (2) results in that only the nearest representation (or equally the nearest data) is assigned as the neighbor of $\boldsymbol{R}_i$ (or $\boldsymbol{X}_i$) with probability 1 and all the others with probabilities 0. Similar to $\boldsymbol{R}$, we again enforce minimizing $\|\boldsymbol{A}\|_F^2$ to prevent from the trivial solution.

Putting the concerns together with slight algebraic transformation gives the following formulation:

$$\begin{aligned} \min \lambda_1\|\boldsymbol{R}\|_F^2 + \lambda_2\,\mathrm{tr}(\boldsymbol{R}\boldsymbol{L}_A\boldsymbol{R}^T) + \lambda_3\|\boldsymbol{A}\|_F^2 \\ \mathrm{s.\,t.}\ \boldsymbol{X} = \boldsymbol{X}\boldsymbol{R};\ \forall i\ \boldsymbol{A}_i^T\mathbf{1} = 1;\ \boldsymbol{A}_i \succeq 0, \end{aligned} \tag{3}$$

where $\boldsymbol{L}_A$ is the Laplacian matrix of $\boldsymbol{A}$, which is constructed in the way of $\boldsymbol{D}_A - \boldsymbol{A}$. The degree matrix $\boldsymbol{D}_A$ is defined as a diagonal matrix where the i^{th} diagonal element is $\sum_j \boldsymbol{A}_{ij}$. In addition, λ_1, λ_2 and λ_3 are three non-negative weights balancing the corresponding terms.

In real world applications, the noise free and independent subspaces assumption may not be satisfied. It is desirable to extend the problem (3) to be robust to noises. With the introduction of the noise term $\|\boldsymbol{E}\|_F^2$, the problem can be finally formulized as follows:

$$\begin{aligned} \min \|\boldsymbol{E}\|_F^2 + \lambda_1\|\boldsymbol{R}\|_F^2 + \lambda_2\,\mathrm{tr}(\boldsymbol{R}\boldsymbol{L}_A\boldsymbol{R}^T) + \lambda_3\|\boldsymbol{A}\|_F^2 \\ \mathrm{s.\,t.}\ \boldsymbol{X} = \boldsymbol{X}\boldsymbol{R} + \boldsymbol{E};\ \forall i\ \boldsymbol{A}_i^T\mathbf{1} = 1;\ \boldsymbol{A}_i \succeq 0. \end{aligned} \tag{4}$$

In the next section, we will propose a novel algorithm to effectively and efficiently solve the problem (4).

3 Optimization

As we have seen in (4), it has combined all aforementioned priors and constraints for learning the representations of data and finding the meaningful affinity matrix with respect to the data representations in a unified optimization framework. Although the objective (4) is not jointly convex in $\boldsymbol{A}$ and $\boldsymbol{R}$, but convex with respect to each of them when the other is fixed. The Augmented Lagrange Multiplier (ALM) with Alternating Direction Minimizing (ADM) strategy [Lin *et al.*, 2011] has proven to be an efficient and effective solver of problems like (4). To apply ALM-ADM on our problem, we need to make our objective function separable. Thus we introduce one auxiliary variable $\boldsymbol{Q}$ to replace $\boldsymbol{R}$ in the trace term of the objective function (4). Accordingly, $\boldsymbol{Q} = \boldsymbol{R}$ acts as the additional constraint. Note that the probability properties of every $\boldsymbol{A}_i$ are enforced as hard constraints. The augmented Lagrangian function of (4) $\mathcal{L}_{\{\forall i\ \boldsymbol{A}_i^T\mathbf{1}=1;\boldsymbol{A}_i\succeq 0\}}$ is

$$\begin{cases} \|\boldsymbol{E}\|_F^2 + \lambda_1\|\boldsymbol{R}\|_F^2 + \lambda_2\,\mathrm{tr}(\boldsymbol{Q}\boldsymbol{L}_A\boldsymbol{Q}^T) + \lambda_3\|\boldsymbol{A}\|_F^2 \\ \quad +\Phi(\boldsymbol{Z}_1, \boldsymbol{X} - \boldsymbol{X}\boldsymbol{R} - \boldsymbol{E}) + \Phi(\boldsymbol{Z}_2, \boldsymbol{Q} - \boldsymbol{R}) \end{cases} \tag{5}$$

with the definition $\Phi(\boldsymbol{Z}, \boldsymbol{C}) \equiv \frac{\mu}{2}\|\boldsymbol{C}\|_F^2 + \langle \boldsymbol{Z}, \boldsymbol{C}\rangle$, where μ is a positive penalty scalar and, $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$ are the Lagrangian multipliers. Besides the Lagrangian multipliers, there are four variables, including $\boldsymbol{E}$, $\boldsymbol{R}$, $\boldsymbol{Q}$ and $\boldsymbol{A}$, to solve. The solver iteratively updates one variable at a time by fixing the others. The solutions of the subproblems are as follows:

$\boldsymbol{E}$-subproblem: For computing $\boldsymbol{E}^{(t+1)}$, we take derivative of $\mathcal{L}$ with respect to $\boldsymbol{E}$ with the unrelated terms fixed and set it to zero, then obtain $\boldsymbol{E}^{(t+1)} =:$

$$\begin{aligned} &\operatorname*{argmin}_{\boldsymbol{E}} \|\boldsymbol{E}\|_F^2 + \Phi(\boldsymbol{Z}_1^{(t)}, \boldsymbol{X} - \boldsymbol{X}\boldsymbol{R}^{(t)} - \boldsymbol{E}) \\ &= \frac{\boldsymbol{Z}_1^{(t)} + \mu^{(t)}(\boldsymbol{X} - \boldsymbol{X}\boldsymbol{R}^{(t)})}{2 + \mu^{(t)}}, \end{aligned} \tag{6}$$

where $\{\mu^{(t)}\}$ is a monotonically increasing sequence.

$\boldsymbol{R}$-subproblem: It is obvious that all the terms related with $\boldsymbol{R}$ are quadratic, thus dropping the constant terms and taking derivative of $\mathcal{L}$ with respect to $\boldsymbol{R}$ gives $\boldsymbol{R}^{(t+1)} =:$

$$\begin{aligned} &\operatorname*{argmin}_{\boldsymbol{R}} \lambda_1\|\boldsymbol{R}\|_F^2 + \Phi(\boldsymbol{Z}_1^{(t)}, \boldsymbol{X} - \boldsymbol{X}\boldsymbol{R} - \boldsymbol{E}^{(t+1)}) \\ &+ \Phi(\boldsymbol{Z}_2^{(t)}, \boldsymbol{Q}^{(t)} - \boldsymbol{R}) = \left(\frac{2\lambda_1 + \mu^{(t)}}{\mu^{(t)}}\boldsymbol{I} + \boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{T}, \end{aligned} \tag{7}$$

where $\boldsymbol{T} \equiv \boldsymbol{X}^T(\boldsymbol{X} - \boldsymbol{E}^{(t+1)} + \frac{\boldsymbol{Z}_1^{(t)}}{\mu^{(t)}}) + \boldsymbol{Q}^{(t)} + \frac{\boldsymbol{Z}_2^{(t)}}{\mu^{(t)}}$.

$\boldsymbol{Q}$-subproblem: In a similar way to updating $\boldsymbol{E}$ and $\boldsymbol{R}$, the closed form solution of this subproblem can be easily calculated by $\boldsymbol{Q}^{(t+1)} =:$

$$\begin{aligned} &\operatorname*{argmin}_{\boldsymbol{Q}} \lambda_2\,\mathrm{tr}(\boldsymbol{Q}\boldsymbol{L}_A^{(t)}\boldsymbol{Q}^T) + \Phi(\boldsymbol{Z}_2^{(t)}, \boldsymbol{Q} - \boldsymbol{R}^{(t+1)}) \\ &= (\mu^{(t)}\boldsymbol{R}^{t+1} - \boldsymbol{Z}_2^{(t)})(2\lambda_2\boldsymbol{L}_A^{(t)} + \mu^{(t)}\boldsymbol{I})^{-1}. \end{aligned} \tag{8}$$

$\boldsymbol{A}$-subproblem: The update of $\boldsymbol{A}^{(t+1)}$ can be done via solving the following optimization problem:

$$\begin{aligned} &\operatorname*{argmin}_{\boldsymbol{A}} \lambda_2\,\mathrm{tr}(\boldsymbol{Q}^{(t+1)}\boldsymbol{L}_A\boldsymbol{Q}^{(t+1)T}) + \lambda_3\|\boldsymbol{A}\|_F^2 \\ &\qquad \text{s.t.}\ \ \forall i\ \boldsymbol{A}_i^T\mathbf{1} = 1;\ \ \boldsymbol{A}_i \succeq 0. \end{aligned} \tag{9}$$

Algorithm 1: Proposed Robust Subspace Segmentation

Input: Data matrix $\boldsymbol{X} \in \mathbb{R}^{D\times n}$, cluster number c, nearest neighbor number k, $\lambda_1 \geq 0$, $\lambda_2 \geq 0$ and $\lambda_3 \geq 0$.

Initial.: $\boldsymbol{E}^{(0)} = \boldsymbol{Z}_1^{(0)} = \mathbf{0} \in \mathbb{R}^{D\times n}$, $\boldsymbol{F}^{(0)} = \mathbf{0} \in \mathbb{R}^{c\times n}$, $\boldsymbol{R}^{(0)} = \boldsymbol{Q}^{(0)} = \boldsymbol{A}^{(0)} = \boldsymbol{Z}_2^{(0)} = \mathbf{0} \in \mathbb{R}^{n\times n}$, $\mu^{(0)} = 1.25$, $\rho > 1$, $t = 0$

while *not converged* **do**

- Construct $\boldsymbol{L}_A^{(t)}$ based on $\boldsymbol{A}^t$;
- Update $\boldsymbol{E}^{(t+1)}$ via Eq. (6);
- Update $\boldsymbol{R}^{(t+1)}$ via Eq. (7);
- Update $\boldsymbol{Q}^{(t+1)}$ via Eq. (8);
- **for** i *from* 1 *to* n **do**
  - Update $\boldsymbol{A}_i^{(t+1)}$ via Eq. (10);
- **end**
- Balance $\boldsymbol{A}^{(t+1)}$ by $\frac{\boldsymbol{A}^{(t+1)}+\boldsymbol{A}^{(t+1)T}}{2}$;
- Update the multipliers via Eq. (11);
- $\mu^{t+1} = \mu^t\rho$; $t = t + 1$;

end

Segment the data into c groups by Normalized cuts.

Output: Final Data Segmentation

As can be seen from (9), it can be separated into a set of smaller independent problems, *i.e.*:

$$\forall i\ \boldsymbol{A}_i^{(t+1)} = \operatorname*{argmin}_{\boldsymbol{A}_i\in\{\boldsymbol{a}|\boldsymbol{a}^T\mathbf{1}=1;\boldsymbol{a}\succeq 0\}} \|\boldsymbol{A}_i + \boldsymbol{d}_i^{Q(t+1)}\|_F^2$$

where $\boldsymbol{d}_i^{Q(t+1)} \in \mathbb{R}^{n\times 1}$ is a vector, the j^{th} element of which is $\boldsymbol{d}_{ij}^{Q(t+1)} = \frac{\lambda_2\|\boldsymbol{Q}_i^{(t+1)} - \boldsymbol{Q}_j^{(t+1)}\|_F^2}{4\lambda_3}$. For each $\boldsymbol{A}_i$, the closed form solution is:

$$\boldsymbol{A}_i^{(t+1)} = \left(\frac{1 + \sum_{j=1}^k \tilde{\boldsymbol{d}}_{ij}^{Q(t+1)}}{k}\mathbf{1} - \boldsymbol{d}_i^{Q(t+1)}\right)_+, \tag{10}$$

where the operator $(\boldsymbol{u})_+$ turns negative elements in $\boldsymbol{u}$ to 0 while keeps the rest. Please notice that the parameter $k \in \{1, ..., n\}$ is introduced to control the number of nearest neighbors of $\boldsymbol{Q}_i$ (or $\boldsymbol{X}_i$) that could have chance to connect to $\boldsymbol{Q}_i$ (or $\boldsymbol{X}_i$). In addition, the elements of $\tilde{\boldsymbol{d}}_i^{Q(t+1)}$ are those of $\boldsymbol{d}_i^{Q(t+1)}$ but with the ascending order. For clarity and completeness, the detailed proof of the closed form solution of (10) can be found in the appendix. As the graph constructed according to $\boldsymbol{A}$ obtained by (10) is generally an unbalanced digraph, we employ $\frac{\boldsymbol{A}+\boldsymbol{A}^T}{2}$ to achieve the balance.

Multipliers: Besides, there are still two multipliers to update, which are simply done through:

$$\begin{aligned} \boldsymbol{Z}_1^{(t+1)} &= \boldsymbol{Z}_1^{(t)} + \mu^t(\boldsymbol{X} - \boldsymbol{X}\boldsymbol{R}^{(t+1)} - \boldsymbol{E}^{(t+1)}); \\ \boldsymbol{Z}_2^{(t+1)} &= \boldsymbol{Z}_2^{(t)} + \mu^t(\boldsymbol{Q}^{(t+1)} - \boldsymbol{R}^{(t+1)}). \end{aligned} \tag{11}$$

The procedure of solving the problem (4) terminates when $\|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{R}^{t+1} - \boldsymbol{E}^{t+1}\|_F \leq \delta\|\boldsymbol{X}\|_F$ with $\delta = 10^{-7}$ or the maximal number of iterations is reached. After obtaining the affinity matrix $\boldsymbol{A}$ and the representation matrix $\boldsymbol{R}$, the

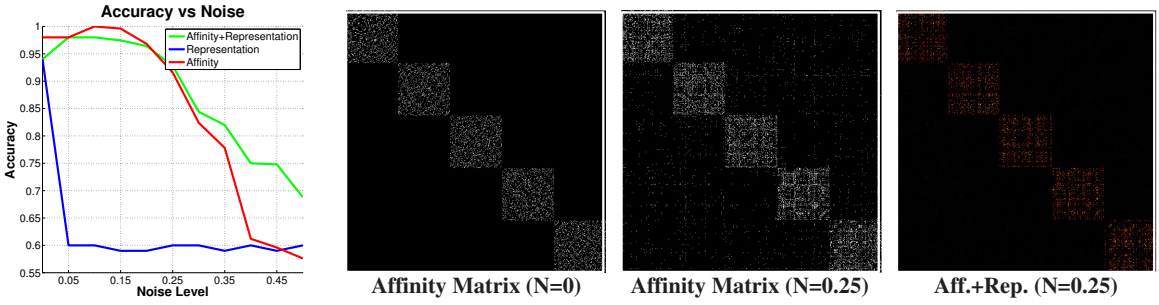

Figure 1: **Left:** clustering accuracy vs noise. **Mid-Left:** the affinity on synthesized data without noise. **Rest:** the affinity and, the Hadamard product of the affinity and the representation on synthesized data with 0.25 white Gaussian noise, respectively.

spectral clustering techniques, Normalized Cuts for all the involved methods in this paper, can be executed to segment the input data into c clusters. The entire algorithm of subspace segmentation is summarized in Algorithm 1.

It is worth noting that although there is no established theory of global convergence in literature for ADM algorithms applied to non-convex problem as the one solved in this work, it is guaranteed that the proposed algorithm converges to at least a stationary point (first order optimality condition). In addition, empirical evidence on both synthesized and real data presented in the next section suggests that the proposed algorithm have very strong and stable convergence behavior.

4 Experimental Verification

In this section, we conduct experiments on synthetic data to reveal the efficacy of our proposed method, and on real data to demonstrate the superior performance of our method over the state-of-the-art alternatives including k-NN using heat kernel distance, CAN and PCAN [Nie *et al.*, 2014], SSC [Elhamifar and Vidal, 2009], LRR[1] [Liu *et al.*, 2013], LSR[2] [Lu *et al.*, 2012] and CASS [Lu *et al.*, 2013], the codes for which are downloaded from the authors' webpages. To obtain the best possible performance of the compared methods for different cases, we tune their corresponding parameters. Specifically, for k-NN, CAN and PCAN, the free parameter k is tuned from 1 to 10. For SSC, the space of the regularizer weight on $\boldsymbol{R}$ is $\alpha \in \{2, 4, ..., 20\}$, $\lambda \in \{0.1, 0.2, ..., 5.0\}$ for LSR, $\lambda \in \{0.0001, 0.001, 0.01, 0.1, 1.0, 2.0, 3.0\}$ for LRR, while $\lambda \in \{0.0001, 0.001, 0.01, 0.1, 1.0\}$ for CASS. To simplify our parameters, we let $\lambda_1 = \lambda_2 = \lambda_3 = \hat{\lambda} \in \{0.1, 0.2, ..., 1.0\}$, although the simplification may very likely exclude the best performance for our method. By doing so, the parameter space is significantly shrunken, but contained by the original space. In other words, if the proposed algorithm with the shrunken parameter space outperforms the state-of-the-arts, the original parameter space can also achieve the same or possibly better performance. Due to space limit, we do not give the influence analysis of each parameter individually. Please note that our model also involves the parameter k, which will be fixed in the experiments according to a k effect testing (discussed in Sec. 4.2). Normalized Cuts [Shi and Malik, 2000] is employed to segment the input data into clusters for all the competitors, average segmentation accuracies[3] over 10 independent trials are finally reported.

4.1 Synthetic Data

This part attempts to verify the robustness of our method to different levels of noise. We generate 5 independent subspaces $\{\mathcal{S}_s\}_{s=1}^{5}$ of 4 dimensions, whose ambient dimension is 250. There are 100 unit data points randomly sampled from each subspace, a part of which are chosen to be corrupted with different levels of white Gaussian noise $\mathcal{N}(0, 1)$. For this experiment, the proportion of polluted data is fixed to 50%, the nearest neighbor number and $\hat{\lambda}$ are set to 10 and 0.1 respectively, and the noise level varies from 0 (no noise) to 0.5. We evaluate the clustering performance of executing the Normalized Cuts on the affinity only ($\boldsymbol{A}$, red curve), the presentation only ($\boldsymbol{R}$, blue curve) and their Hadamard product ($\boldsymbol{A} \odot \boldsymbol{R}$, green curve). The combination of $\boldsymbol{R}$ and $\boldsymbol{A}$ is motivated by that the probability of two data points drawn from different subspaces simultaneously having high responses in R and A should be low. As can be seen from the first picture in Fig. 1, all the three schemes achieve very high accuracies when data are clean. But, as the noise level increases, the clustering performance of $\boldsymbol{R}$ only sharply drops to about 0.6. While the accuracies of $\boldsymbol{A}$ only (for most cases) and $\boldsymbol{A} \odot \boldsymbol{R}$ are much higher. Please notice that, the red curve is always superior to the green until the noise level is up to 0.2. Afterwards, the green precedes the red. That is to say, using the affinity matrix only can provide a promising result on slightly polluted data, while further introducing the representation matrix is able to significantly boost the robustness to heavily corrupted data. The second picture in Fig. 1 displays the affinity matrix obtained by our method corresponding to the case of zero noise, which shows the perfect block sparsity. The rest two pictures in Fig. 1 give the affinity matrix and, the

[1] As the authors proposed two versions of LRR with different models of $\boldsymbol{E}$, *i.e.* the ℓ^1 and $\ell^{2,1}$, we test both of them and denote them as LRR_1 and LRR_{21}, respectively. For more details, please refer to [Liu *et al.*, 2013].

[2] LSR has two implementations, which are denoted as LSR_1 and LSR_2, respectively. Please refer to [Lu *et al.*, 2012].

[3] The metric, segmentation accuracy, is calculated by finding the best matching between cluster labels and ground truth labels.

Table 1: Performance Comparison on Extended Yale B

Methods	*k-NN*	*CAN*	*PCAN*	*SSC*	*LRR*$_1$	*LRR*$_{21}$	*LSR*$_1$	*LSR*$_2$	*CASS*	*Ours* $\boldsymbol{A}$	*Ours* $\boldsymbol{A}\odot\boldsymbol{R}$
Free Para.	k	k	k	α	λ	λ	λ	λ	λ	$\hat{\lambda}$	$\hat{\lambda}$
5 sub.	71.56	69.94	72.19	*97.19*	65.94	83.25	86.44	94.06	94.03	**99.06**	95.63
Para.	2	2	3	4.0	0.1	1.0	0.3	0.3	-	0.1	0.1
10 sub.	49.59	48.02	48.88	63.97	60.56	60.00	57.03	61.73	81.88	**92.28**	*87.70*
Para.	2	5	2	10.0	0.01	2.0	1.0	0.1	-	0.1	0.1
30 sub.	52.59	38.79	42.78	50.19	58.23	61.24	57.77	58.38	*NA*	**76.84**	*76.35*
Para.	2	5	3	10.0	0.1	2.0	0.2	0.3	-	0.1	0.1
38 sub.	47.53	38.24	40.89	45.89	55.11	57.39	56.13	57.73	*NA*	**74.48**	*71.91*
Para.	2	5	3	10.0	0.1	2.0	0.5	0.5	-	0.1	0.1

combination of affinity and representation with respect to the case with 0.25 noise, the block sparsities of which are not perfect but very well preserved.

4.2 Extended Yale B

We compare the proposed method with other state-of-the-art methods for face clustering on the Extended Yale B dataset [Lee *et al.*, 2005]. The dataset contains face images of 38 subjects. For each subject, there are about 64 frontal face images taken under different illuminations. More than half of the data vectors in this dataset have been corrupted by "shadows", which makes the task difficult. In this experiment, we resize the images into 32×32 and use the raw pixel values to form data vectors of 1024 dimensions.

***k* Effect.** We use first 10 subjects to test the parameter effect of k, say the number of nearest neighbors. To eliminate the effect from other parameters, we empirically set $\hat{\lambda}$ to 0.1. In addition, the three kinds of matrix including $\boldsymbol{R}$ only, $\boldsymbol{A}$ only and $\boldsymbol{A}\odot\boldsymbol{R}$ are again employed to see the difference. As displayed in the left graph of Fig. 2 , it is easy to see that using $\boldsymbol{A}$ only and $\boldsymbol{A}\odot\boldsymbol{R}$ give much more promising results when k ranges from 2 to 30 than using $\boldsymbol{R}$ only. Similar to the conclusion drawn from Sec. 4.1, $\boldsymbol{A}\odot\boldsymbol{R}$ shows a better robustness than $\boldsymbol{A}$ only in this experiment. Based on this testing, we will fix $k = 3$ for our method for the rest experiments.

Convergence Speed. Without loss of generality, the convergence speed of Algorithm 1 by setting $\hat{\lambda} = 0.1$ on 10 subjects is given in the right picture of Fig. 2, in which the stop criterion sharply drops to the level of 10^{-6} with about 10 iterations and to 10^{-7} using 27 iterations. Our algorithm takes 4s to finish the computation on our PC, which is slower than LSR that spends 0.06s, but much more efficient than SSC (39s), LRR (60s) and CASS (34, 560s). This indicates that our proposed algorithm can converge sufficiently fast. Moreover, all the experiments conducted in this paper by our algorithm are converged with about $25 - 40$ iterations.

Performance Comparison. Table 1 provides the quantitative comparison among the competitors on the Extended Yale B dataset. We evaluate the performance of the competitors on the tasks with different numbers of subject including 5, 10, 20, 30 and 38. The bold and italic numbers in each row represent the best and the second best results, respectively, for the corresponding task. Our parameter $\hat{\lambda}$ is determined according to the highest accuracy of the case with 5 subjects,

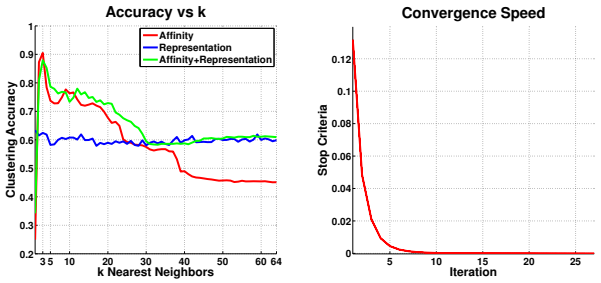

Figure 2: **Left:** parameter effect of k. **Right:** convergence speed of the proposed algorithm.

i.e. 0.1, and fixed for all the cases in this experiment. While, for different cases, each of the alternatives reports its best accuracies individually obtained by tuning its parameters in the corresponding parameter space. Please notice that the results of CASS with respect to the 5- and 10-subject cases are the best results reported by the authors of CASS [Lu *et al.*, 2013]. For the rest cases, we do not provide the results as CASS takes too much time to handle even the 20-subject task (and not reported in [Lu *et al.*, 2013]). As can be observed from Table 1, both our methods using $\boldsymbol{A}$ only and $\boldsymbol{A}\odot\boldsymbol{R}$ with the uniform setting greatly outperform the others with tuned parameters for all the involved cases. We can also see that in this experiment $\boldsymbol{A}\odot\boldsymbol{R}$ is slightly behind $\boldsymbol{A}$ only. The reason may be that although the data in this dataset are corrupted by different illuminations, they are well aligned and thus largely preserve the subspace structure. In [Feng *et al.*, 2014], the authors state that their proposed scheme can improve the performance of LRR and LSR by $3\%-6\%$ on the (only reported) cases with 5 and 10 subjects[4], even though, our method still significantly outperforms [Feng *et al.*, 2014].

4.3 USPS

Further, we compare the performance of SSC, LRR, LSR, CASS and our method on the USPS dataset[5], which consists of 10 classes corresponding to 10 handwritten digits, $0 \sim 9$. We use the first 100 examples with the size 16×16 of each subject for this experiment. The examples of each class are

[4]Since the code of [Feng *et al.*, 2014] is not available when this paper is prepared, we do not explicitly compare with it.

[5]www.csie.ntu.edu.tw/˜cjlin/libsvmtools/datasets/multiclass.html

Table 2: Performance Comparison on USPS

Methods	*SSC*	*LRR*$_1$	*LRR*$_{21}$	*LSR*$_1$	*LSR*$_2$	*CASS*	*Ours* $\boldsymbol{A}$	*Ours* $\boldsymbol{A} \odot \boldsymbol{R}$
Free Para.	α	λ	λ	λ	λ	λ	$\hat{\lambda}$	$\hat{\lambda}$
10 subjects	73.72	74.40	74.40	72.40	72.20	72.70	*80.79*	**82.58**
Parameter	4.0	0.1	0.01	4.8	4.1	0.1	0.8	0.8

Table 3: Performance Comparison on UMIST

Methods	*SSC*	*LRR*$_1$	*LRR*$_{21}$	*LSR*$_1$	*LSR*$_2$	*CASS*	*Ours* $\boldsymbol{A}$	*Ours* $\boldsymbol{A} \odot \boldsymbol{R}$
Free Para.	α	λ	λ	λ	λ	λ	$\hat{\lambda}$	$\hat{\lambda}$
20 subjects	67.84	52.38	49.70	53.57	53.39	51.50	*68.09*	**70.12**
Parameter	4.0	0.01	0.001	1.3	1.0	0.1	1.0	1.0

with many variations on appearance, and of different classes may share some features (*e.g.* digits 3 and 8), which violates the assumption of independent subspaces and thus increases the difficulty of clustering.

Performance Comparison. As shown in Table 2, the best possible clustering accuracies of SSC, LRR, LSR and CASS are very close to each other, which fall into the range $[72.20, 74.40]$. These results are reasonably good although the USPS is more challenging than the Extended Yale B, as the subject number of the USPS is only 10 and the amount (sampling) of each subject of USPS is more than that of the Extended Yale B. As for our method, the performance of the scheme using $\boldsymbol{A}$ achieves 80.79, while that of the scheme using $\boldsymbol{A} \odot \boldsymbol{R}$ obtains 82.58, which significantly improve the clustering accuracy on USPS compared with the others. It is worth mentioning that, in this experiment, the $\boldsymbol{A} \odot \boldsymbol{R}$ scheme emerges its advantage over the $\boldsymbol{A}$ only.

4.4 UMIST

Moreover, we attempt to test the abilities of different approaches on a more challenging dataset UMIST [Graham and Allinson, 1998]. The UMIST collects 575 images from 20 subjects, which are resized into 56×46 for this experiment. Each subject has about 28 images with different poses, which significantly breaks the assumed subspace structure. Therefore, the performance of SSC, LRR, LSR, CASS and our method on subspace segmentation may degenerate or even fail, because they are primarily designed under the assumption of strong subspace structure.

Performance Comparison. Table 3 shows the performance comparison on the UMIST, from which we can see that the segmentation accuracies of LRR, LSR and CASS are around 0.51. This verifies the fact that the UMIST is a very challenging subspace segmentation dataset. Surprisingly, SSC and our method achieve reasonably high accuracies, *i.e.* 67.84, 68.09 and 70.12, respectively. This is mainly because SSC processes data individually instead of enforcing the subspace structure, which is more suitable for this dataset than the strategies of LRR and CASS. While our method introduces k nearest neighbor concept (with relatively small k) in the affinity matrix, which may connect the faces of the same subject with slight pose changes.

5 Conclusion

Subspace segmentation is an important yet challenging problem in many research fields, such as machine learning, computer vision and pattern recognition. Differently to previous spectral clustering based work that computes the affinity based either directly on the distance between data or on the similarity of data representations, this paper has shown a novel method that simultaneously learns data representations and their affinity matrix. We have formulated the problem into a unified optimization framework and designed an efficient Augmented Lagrangian Multiplier based algorithm to seek the solution. The experimental results, compared to the state-of-the-art alternatives, have demonstrated the clear advantages of the proposed method.

Appendix

Given a problem with the following shape:

$$\underset{\boldsymbol{a}}{\operatorname{argmin}} \|\boldsymbol{a}+\boldsymbol{d}\|_F^2 \quad \text{s.t. } \boldsymbol{a}^T\mathbf{1}=1;\ \boldsymbol{a} \succeq 0, \tag{12}$$

where $\boldsymbol{a} \in \mathbb{R}^{n\times 1}$ is the target, and $\boldsymbol{d} \in \mathbb{R}^{n\times 1}$ is a known (distance) vector. To be more general, we can further appoint the number of nonzero elements in $\boldsymbol{a}$ as $k \in \{1, ..., n\}$. The closed form solution of the problem (12) is as follows:

$$\boldsymbol{a} = \left(\frac{1+\sum_{j=1}^{k}\tilde{\boldsymbol{d}}_j}{k}\mathbf{1} - \boldsymbol{d}\right)_+, \tag{13}$$

where the elements of $\tilde{\boldsymbol{d}} \in \mathbb{R}^{n\times 1}$ are those of $\boldsymbol{d}$ but with the ascending order.

Proof. The Lagrangian function of the problem (12) is as:

$$\mathcal{C} = \frac{1}{2}\|\boldsymbol{a}+\boldsymbol{d}\|_F^2 - \alpha(\boldsymbol{a}^T\mathbf{1}-1) - \boldsymbol{\omega}^T\boldsymbol{a}, \tag{14}$$

where $\alpha \succeq 0$ and $\boldsymbol{\omega} \succeq 0$ are the Lagrangian multipliers. It is easy to verify the optimal solution of $\boldsymbol{a}$ can be obtained through solving the following equation system, with the help of the KKT condition:

$$\frac{\partial \mathcal{C}}{\partial \boldsymbol{a}} = \boldsymbol{a}+\boldsymbol{d}-\alpha\mathbf{1}-\boldsymbol{\omega}=0;\ \boldsymbol{a}^T\mathbf{1}-1=0;\ \boldsymbol{\omega}^T\boldsymbol{a}=0. \tag{15}$$

The third equation holds when the condition that if $\boldsymbol{a}_j > 0$ then $\boldsymbol{\omega}_j = 0$ is satisfied, thus we have

$$\boldsymbol{a} = \left(\alpha \mathbf{1} - \boldsymbol{d}\right)_+. \tag{16}$$

And there are k positive elements in $\boldsymbol{a} \succeq 0$. Namely:

$$\alpha - \tilde{\boldsymbol{d}}_k > 0 \quad \text{and} \quad \alpha - \tilde{\boldsymbol{d}}_{k+1} \leq 0. \tag{17}$$

According to Eq. (16) together with $\boldsymbol{a}^T \mathbf{1} = 1$, we have

$$\sum_{j=1}^{k} (\alpha - \tilde{\boldsymbol{d}}_j) = 1 \Rightarrow \alpha = \frac{1 + \sum_{j=1}^{k} \tilde{\boldsymbol{d}}_j}{k}. \tag{18}$$

Substituting α in Eq. (16) with $\frac{1+\sum_{j=1}^{k} \tilde{\boldsymbol{d}}_j}{k}$ in Eq. (18) recognizes the form stated in Eq. (13). Please notice that only those data points with representation distances to the target smaller than $\frac{1+\sum_{j=1}^{k} \tilde{\boldsymbol{d}}_j}{k}$ are its neighbors. □

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61402467 and in part by the Excellent Young Talent Program through the Institute of Information Engineering, Chinese Academy of Sciences.

References

[Bradley and Mangasarian, 2000] P. Bradley and O. Mangasarian. K-plane clustering. *Journal of Global Optimization*, 16(1):23–32, 2000.

[Candès *et al.*, 2006] E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Image Processing*, 52(2):489–509, 2006.

[Donoho, 2006] D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[Elhamifar and Vidal, 2009] E. Elhamifar and R. Vidal. Space subspace clustering. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2790–2797, 2009.

[Feng *et al.*, 2014] J. Feng, Z. Lin, H. Xu, and S. Yan. Robust subspace segmentation with block-diagonal prior. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3818–3825, 2014.

[Graham and Allinson, 1998] D. Graham and N. Allinson. Characterizing virtual eigensignatures for general purpose face recognition. In *Face Recognition: From Theory to Applications ; NATO ASI Series F, Computer and Systems Sciences*, volume 163, pages 446–456, 1998.

[Ho *et al.*, 2003] J. Ho, M. Yang, J. Lim, K. Lee, and D. Kriegman. Clustering appearances of objects under varying illumination conditions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 11–18, 2003.

[Hong *et al.*, 2006] W. Hong, J. Wright, K. Huang, and Y. Ma. Multiscale hybrid linear models for lossy image representation. *IEEE Transactions on Image Processing*, 15(12):3655–3671, 2006.

[Lee *et al.*, 2005] K. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):684–698, 2005.

[Lin *et al.*, 2011] Z. Lin, R. Liu, and Z. Su. Linearized alternating direction method with adaptive penalty for low rank representation. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 612–620, 2011.

[Liu *et al.*, 2013] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1):171–184, 2013.

[Lu *et al.*, 2012] C. Lu, H. Min, Z. Zhao, L. Zhu, D. Huang, and S. Yan. Robust and efficient subspace segmentation via least squares regression. In *Proceedings of European Conference on Computer Vision*, pages 347–360, 2012.

[Lu *et al.*, 2013] C. Lu, J. Feng, Z. Lin, and S. Yan. Correlation adaptive subspace segmentation by trace lasso. In *Proceedings of International Conference on Computer Vision*, pages 1345–1352, 2013.

[Ma *et al.*, 2007] Y. Ma, H. Derksen, W. Hong, and J. Wright. Segmentation of multivariate mixed data via lossy coding and compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1546–1562, 2007.

[Ma *et al.*, 2008] Y. Ma, A. Yang, H. Derksen, and R. Fossum. Estimation of subspace arrangements with applications in modeling and segmenting mixed data. *SIAM Review*, 50(3):413–458, 2008.

[Nie *et al.*, 2009] F. Nie, D. Xu, I. Tsang, and C. Zhang. Spectral enbedded clustering. In *Proceedings of International Joint Conference on Artifical Intelligence*, pages 1181–1186, 2009.

[Nie *et al.*, 2014] F. Nie, X. Wang, and H. Huang. Clustering and projected clustering with adaptive neighbors. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 977–986, 2014.

[Rao *et al.*, 2010] S. Rao, R. Tron, R. Vidal, and Y. Ma. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1832–1845, 2010.

[Saha *et al.*, 2013] B. Saha, D. Pham, D. Phung, and S. Venkatesh. Sparse subspace clustering via group sparse coding. In *Proceedings of SIAM International Conference on Data Mining*, pages 130–138, 2013.

[Shi and Malik, 2000] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[Vidal *et al.*, 2005] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1–15, 2005.

[Vidal, 2010] R. Vidal. A tutorial on subspace clustering. *IEEE Signal Processing Magzine*, 28:52–68, 2010.