Low Rank Subspace Clustering (LRSC)

René Vidal[a], Paolo Favaro[b]

[a]Center for Imaging Science, Department of Biomedical Engineering, Johns Hopkins University, Baltimore, MD 21218, USA
[b]Institute of Informatics and Applied Mathematics, University of Bern, 3012, Switzerland

Abstract

We consider the problem of fitting one or more subspaces to a collection of data points drawn from the subspaces and corrupted by noise and/or gross errors. We pose this problem as a non-convex optimization problem, where the goal is to decompose the corrupted data matrix as the sum of a clean and self-expressive dictionary plus a matrix of noise and/or gross errors. By self-expressive we mean a dictionary whose atoms can be expressed as linear combinations of themselves with low-rank coefficients. In the case of noisy data, our key contribution is to show that this non-convex matrix decomposition problem can be solved in closed form from the SVD of the noisy data matrix. The solution involves a novel polynomial thresholding operator on the singular values of the data matrix, which requires minimal shrinkage. For one subspace, a particular case of our framework leads to classical PCA, which requires no shrinkage. For multiple subspaces, the low-rank coefficients obtained by our framework can be used to construct a data affinity matrix from which the clustering of the data according to the subspaces can be obtained by spectral clustering. In the case of data corrupted by gross errors, we solve the problem using an alternating minimization approach, which combines our polynomial thresholding operator with the more traditional shrinkage-thresholding operator. Experiments on motion segmentation and face clustering show that our framework performs on par with state-of-the-art techniques at a reduced computational cost.

Keywords: subspace clustering, low-rank and sparse methods, principal component analysis, motion segmentation, face clustering

1. Introduction

The past few decades have seen an explosion in the availability of datasets from multiple modalities. While such datasets are usually very high-dimensional, their intrinsic dimension is often much smaller than the dimension of the ambient space. For instance, the number of pixels in an image can be huge, yet most computer vision models use a few parameters to describe the appearance, geometry and dynamics of a scene. This has motivated the development of a number of techniques for finding low-dimensional representations of high-dimensional data.

One of the most commonly used methods is Principal Component Analysis (PCA), which models the data with *a single* low-dimensional subspace. In practice, however, the data points could be drawn from *multiple subspaces* and the *membership* of the data points to the subspaces could be unknown. For instance, a video sequence could contain several moving objects and different subspaces might be needed to describe the motion of different objects in the scene. Therefore, there is a need to simultaneously cluster the data into multiple subspaces and find a low-dimensional subspace fitting each group of points. This problem, known as *subspace clustering*, finds numerous applications in computer vision, e.g., image segmentation (Yang et al., 2008), motion segmentation (Vidal et al., 2008) and face clustering (Ho et al., 2003), image processing, e.g., image representation and compression (Hong et al., 2006), and systems theory, e.g., hybrid system identification (Vidal et al., 2003).

Prior Work on Subspace Clustering. Over the past decade, a number of subspace clustering methods have been developed. This includes algebraic methods (Boult and Brown, 1991; Costeira and Kanade, 1998; Gear, 1998; Vidal et al., 2005), iterative methods (Bradley and Mangasarian, 2000; Tseng, 2000; Agarwal and Mustafa, 2004; Lu and Vidal, 2006; Zhang et al., 2009), statistical methods (Tipping and Bishop, 1999; Sugaya and Kanatani, 2004; Gruber and Weiss, 2004; Yang et al., 2006; Ma et al., 2007; Rao et al., 2008, 2010), and spectral clustering-based methods (Boult and Brown, 1991; Yan and Pollefeys, 2006; Zhang et al., 2010; Goh and Vidal, 2007; Elhamifar and Vidal, 2009, 2010, 2013; Liu et al., 2010; Chen and Lerman, 2009). Among them, methods based on spectral clustering have been shown to perform very well for several applications in computer vision (see Vidal (2011) for a review and comparison of existing methods).

Spectral-clustering based methods decompose the subspace clustering problem in two steps. In the fist step, an affinity matrix C is constructed, where ideally $C_{ij} \approx 1$ if points i and j are in the same subspace and $C_{ij} \approx 0$ otherwise. In the second step, the segmentation of the data is obtained by applying spectral clustering techniques (see von Luxburg (2007) for a review) to the matrix C. Arguably, the most difficult step is to build a good affinity matrix. This is because two points could be very close to each other, but lie in different subspaces (e.g., near the intersection of two subspaces). Conversely, two points could be far from each other, but lie in the same subspace.

Earlier methods for building an affinity matrix (Boult and Brown, 1991; Costeira and Kanade, 1998) compute the singular value decomposition (SVD) of the data matrix $D = U\Sigma V^\top$ and let $C = V_r V_r^\top$, where the columns of V_r are the top $r = \text{rank}(D)$

singular vectors of D. The rationale behind this choice is that $C_{ij} = 0$ when points i and j are in different *independent* subspaces and the data are *uncorrupted*, as shown in Vidal et al. (2005). In practice, however, the data are often contaminated by noise and gross errors. In such cases, the equation $C_{ij} = 0$ does not hold, even if the rank of the noiseless D was given. Moreover, selecting a good value for r becomes very difficult, because D is full rank. Furthermore, the equation $C_{ij} = 0$ is derived under the assumption that the subspaces are linear. In practice, many datasets are better modeled by affine subspaces.

More recent methods for building an affinity matrix address these issues by using techniques from sparse and low-rank representation. For instance, it is shown in Elhamifar and Vidal (2009, 2010, 2013) that a point in a union of multiple subspaces admits a sparse representation with respect to the dictionary formed by all other data points, i.e., $D = DC$, where C is sparse. It is also shown in Elhamifar and Vidal (2009, 2010, 2013) that, if the subspaces are independent, the nonzero coefficients in the sparse representation of a point correspond to other points in the same subspace, i.e., if $C_{ij} \neq 0$, then points i and j belong to the same subspace. Moreover, the nonzero coefficients can be obtained by ℓ_1 minimization. The coefficients are then used to cluster the data according to the multiple subspaces. A very similar approach is presented in Liu et al. (2010). The major difference is that a low-rank representation is used in lieu of the sparsest representation. While the same principle of representing a point as a linear combination of other points has been successfully used when the data are corrupted by noise and gross errors, from a theoretical viewpoint it is not clear that the above methods are effective when using a corrupted dictionary.

Paper Contributions. In this paper, we propose a general optimization framework for solving the subspace estimation and clustering problem in the case of data drawn from multiple linear subspaces and corrupted by noise and/or gross errors. The proposed framework, which we call Low Rank Subspace Clustering (LRSC), is based on solving the following non-convex optimization problem:

$$(P_1) \quad \min_{A,C,E,G} \ \|C\|_* + \frac{\alpha}{2}\|G\|_F^2 + \gamma\|E\|_1 \quad \text{s.t.} \quad D = A + G + E, \ \ A = AC \ \text{ and } \ C = C^\top,$$

and its relaxation

$$(P_2) \quad \min_{A,C,E,G} \ \|C\|_* + \frac{\tau}{2}\|A - AC\|_F^2 + \frac{\alpha}{2}\|G\|_F^2 + \gamma\|E\|_1 \quad \text{s.t.} \quad D = A + G + E \ \text{ and } \ C = C^\top,$$

where $\|X\|_* = \sum_i \sigma_i(X)$, $\|X\|_F^2 = \sum_{ij} X_{ij}^2$ and $\|X\|_1 = \sum_{ij} |X_{ij}|$ are, respectively, the nuclear, Frobenius and ℓ_1 norms of X. In the above formulations, $A \in \mathbb{R}^{M\times N}$ is an unknown matrix whose columns are points drawn from a union of $n \geq 1$ low-dimensional linear subspaces of unknown dimensions $\{d_i\}_{i=1}^n$, where $d_i \ll M$. We assume that the entries of A are contaminated by noise represented by the matrix $G \in \mathbb{R}^{M\times N}$. We assume also that a small fraction $\rho \ll MN$ of the entries of A is contaminated by gross errors of arbitrary magnitude, which are represented by a matrix $E \in \mathbb{R}^{M\times N}$.

Given a corrupted data matrix $D = A+G+E$, we wish to find a self-expressive, noise-free and outlier-free (clean) data matrix A and a symmetric, low-rank affinity matrix $C \in \mathbb{R}^{N\times N}$. We do so by minimizing the cost function in P_2, which encourages:

- C to be low-rank (by minimizing $\|C\|_*$),
- A to be self-expressive (by minimizing $\|A - AC\|_F^2$),
- G to be small (by minimizing $\|G\|_F^2$), and
- E to be sparse (by minimizing $\|E\|_1$).

By self-expressive we mean that the clean data points can be expressed as linear combinations of themselves with coefficients C, i.e., $A = AC$. Notice that this constraint makes our problem non-convex, because both A and C are unknown. This is an important difference with respect to existing methods, which enforce $D = DC$ where D is the dictionary of corrupted data points. Another important difference is that we directly enforce C to be symmetric, while existing methods symmetrize C as a post-processing step.

The main contribution of our work is to show that important particular cases of P_2 (see Table 1) can be solved in closed form from the SVD of the data matrix. In particular, we show that in the absence of gross errors (i.e., $\gamma = \infty$), A and C can be obtained by thresholding the singular values of D and A, respectively. The thresholding is done using a novel polynomial thresholding operator, which reduces the amount of shrinkage with respect to existing methods. Indeed, when the self-similarity constraint $A = AC$ is enforced exactly (i.e., $\alpha = \infty$), the optimal solution for A reduces to classical PCA, which does not perform any shrinkage. Moreover, the optimal solution for C reduces to the affinity matrix for subspace clustering proposed by Costeira and Kanade (1998). In the case of data corrupted by gross errors, a closed-form solution appears elusive. We thus use an augmented Lagrange multipliers method. Each iteration of our method involves a polynomial thresholding of the singular values to reduce the rank and a regular shrinkage-thresholding to reduce gross errors.

Paper Outline. The remainder of the paper is organized as follows (see Table 1). Section 2 reviews existing results on sparse

Table 1: Particular cases of P_2 Solved in this Paper

	Relaxed	Exact
Uncorrupted	Section 3.1 $0 < \tau < \infty$ $\alpha = \infty$ $\gamma = \infty$	Section 3.2 $\tau = \infty$ $\alpha = \infty$ $\gamma = \infty$
Noise	Section 4.1 $0 < \tau < \infty$ $0 < \alpha < \infty$ $\gamma = \infty$	Section 4.2 $\tau = \infty$ $0 < \alpha < \infty$ $\gamma = \infty$
Gross Errors	Section 5.1 $0 < \tau < \infty$ $0 < \alpha < \infty$ $0 < \gamma < \infty$	Section 5.2 $\tau = \infty$ $0 < \alpha < \infty$ $0 < \gamma < \infty$

representation and rank minimization for subspace estimation and clustering as well as some background material needed for our derivations. Section 3 formulates the low rank subspace clustering problem for linear subspaces in the absence of noise or gross errors and derives a closed form solution for A and C. Section 4 extends the results of Section 3 to data contaminated by noise and derives a closed form solution for A and C based on the polynomial thresholding operator. Section 5 extends the results to data contaminated by both noise and gross errors and shows that A and C can be found using alternating minimization. Section 6 presents experiments that evaluate our method on synthetic and real data. Section 7 gives the conclusions.

2. Background

In this section we review existing results on sparse representation and rank minimization for subspace estimation (Section 2.1) and subspace clustering (Section 2.2). We also review some trace inequalities, which will be useful in our derivations (Section 2.3).

2.1. Subspace Estimation by Sparse Representation and Rank Minimization

Low Rank Minimization. Given a data matrix corrupted by Gaussian noise $D = A + G$, where A is an unknown low-rank matrix and G represents the noise, the problem of finding a low-rank approximation of D can be formulated as

$$\min_{A} \ \|D - A\|_F^2 \ \text{ subject to } \ \text{rank}(A) \leq r. \tag{1}$$

The optimal solution to this (PCA) problem is given by $A = U\mathcal{H}_{\sigma_{r+1}}(\Sigma)V^\top$, where $D = U\Sigma V^\top$ is the SVD of D, σ_k is the k-th singular value of D, and $\mathcal{H}_\epsilon(x)$ is the *hard thresholding operator*:

$$\mathcal{H}_\epsilon(x) = \begin{cases} x & |x| > \epsilon \\ 0 & \text{else.} \end{cases} \tag{2}$$

When r is unknown, the problem of finding a low-rank approximation can be formulated as

$$\min_{A} \quad \text{rank}(A) + \frac{\alpha}{2}\|D - A\|_F^2, \tag{3}$$

where $\alpha > 0$ is a parameter. Since the optimal solution of (1) for a fixed rank $r = \text{rank}(A)$ is $A = U\mathcal{H}_{\sigma_{r+1}}(\Sigma)V^\top$, the problem in (3) is equivalent to

$$\min_{r} \ \ r + \frac{\alpha}{2}\sum_{k>r}\sigma_k^2(D). \tag{4}$$

The optimal r is the smallest r such that $\sigma_{r+1} \leq \sqrt{2/\alpha}$. Therefore, the optimal A is given by $A = U\mathcal{H}_{\sqrt{\frac{2}{\alpha}}}(\Sigma)V^\top$.

Since rank minimization problems are in general NP hard, a common practice (see Recht et al. (2010)) is to replace the rank of A by its nuclear norm $\|A\|_*$, i.e., the sum of its singular values, which leads to the following convex problem

$$\min_{A} \quad \|A\|_* + \frac{\alpha}{2}\|D - A\|_F^2, \tag{5}$$

where $\alpha > 0$ is a user-defined parameter.

It is shown in Cai et al. (2008) that the optimal solution to the problem in (5) is given by $A = U\mathcal{S}_{\frac{1}{\alpha}}(\Sigma)V^\top$, where $\mathcal{S}_\epsilon(x)$ is the *shrinkage-thresholding operator*

$$\mathcal{S}_\epsilon(x) = \begin{cases} x - \epsilon & x > \epsilon \\ x + \epsilon & x < -\epsilon \\ 0 & \text{else.} \end{cases} \tag{6}$$

Notice that the latter solution does not coincide with the one given by PCA, which performs hard-thresholding of the singular values of D without shrinking them by $1/\alpha$.

Principal Component Pursuit. While the above methods work well for data corrupted by Gaussian noise, they break down for data corrupted by gross errors. In Candès et al. (2011) this issue is addressed by assuming sparse gross errors, i.e., only a small percentage of the entries of D are corrupted. Hence, the goal is to decompose the data matrix D as the sum of a low-rank matrix A and a sparse matrix E, i.e.,

$$\min_{A,E} \quad \text{rank}(A) + \gamma\|E\|_0 \quad \text{s.t.} \quad D = A + E, \tag{7}$$

where $\gamma > 0$ is a parameter. Since this problem is in general NP hard, a common practice is to replace the rank of A by its nuclear norm and the ℓ_0 semi-norm by the ℓ_1 norm. It is shown in Candès et al. (2011) that, under broad conditions, the optimal solution to the problem in (7) is identical to that of the convex problem

$$\min_{A,E} \quad \|A\|_* + \gamma\|E\|_1 \quad \text{s.t.} \quad D = A + E. \tag{8}$$

While a closed form solution to this problem is not known, convex optimization techniques can be used to find the minimizer. We refer the reader to Lin et al. (2011) for a review of numerous approaches. One such approach is the Augmented Lagrange Multiplier (ALM) method, which considers the following optimization problem

$$\max_{Y} \min_{A,E} \|A\|_* + \gamma\|E\|_1 + \langle Y, D - A - E\rangle + \frac{\alpha}{2}\|D - A - E\|_F^2. \tag{9}$$

The third term enforces the equality constraint via the matrix of Lagrange multipliers Y, while the fourth term (which is zero at the optimum) makes the cost function strictly convex and thus improves the convergence. Notice that the minimization over A and E for a fixed Y can be re-written as

$$\min_{A,E} \|A\|_* + \gamma\|E\|_1 + \frac{\alpha}{2}\|D - A - E + \alpha^{-1}Y\|_F^2. \tag{10}$$

Given E and Y, it follows from the solution of (5) that the optimal solution for A is $A = U\mathcal{S}_{\alpha^{-1}}(\Sigma)V^\top$, where $U\Sigma V^\top$ is the SVD of $D - E + \alpha^{-1}Y$. Given A and Y, the optimal solution for E satisfies

$$-\alpha(D - A - E + \alpha^{-1}Y) + \gamma\,\text{sign}(E) = 0. \tag{11}$$

It is shown in Lin et al. (2011) that this equation can be solved in closed form using the shrinkage-thresholding operator as

$E = \mathcal{S}_{\gamma\alpha^{-1}}(D - A + \alpha^{-1}Y)$. Therefore, the inexact ALM method iterates the following steps till convergence

$$\begin{aligned}(U, \Sigma, V) &= \text{svd}(D - E_k + \alpha_k^{-1} Y_k) \\ A_{k+1} &= U \mathcal{S}_{\alpha_k^{-1}}(\Sigma) V^\top \\ E_{k+1} &= \mathcal{S}_{\gamma\alpha_k^{-1}}(D - A_{k+1} + \alpha_k^{-1} Y_k) \\ Y_{k+1} &= Y_k + \alpha_k (D - A_{k+1} - E_{k+1}) \\ \alpha_{k+1} &= \rho \alpha_k.\end{aligned} \tag{12}$$

This ALM method is essentially an iterated thresholding algorithm, which alternates between thresholding the SVD of $D - E + Y/\alpha$ to get A and thresholding $D - A + Y/\alpha$ to get E. The update for Y is simply a gradient ascent step. Also, to guarantee the convergence of the algorithm, the parameter α is updated by choosing the parameter ρ such that $\rho > 1$ so as to generate a sequence α_k that goes to infinity.

2.2. Subspace Clustering by Sparse Representation and Rank Minimization

Consider now the more challenging problem of clustering data drawn from multiple subspaces. In what follows, we discuss two methods based on sparse and low-rank representation for addressing this problem.

Sparse Subspace Clustering (SSC). The work of Elhamifar and Vidal (2009) shows that, in the case of uncorrupted data, an affinity matrix for solving the subspace clustering problem can be constructed by expressing each data point as a linear combination of all other data points. That is, we wish to find a matrix C such that $D = DC$ and $\text{diag}(C) = 0$. In principle, this leads to an ill-posed problem with many possible solutions. To resolve this issue, the principle of *sparsity* is invoked. Specifically, every point is written as a *sparse* linear combination of all other data points by minimizing the number of nonzero coefficients. That is

$$\min_C \sum_i \|C_i\|_0 \text{ s.t. } D = DC \text{ and } \text{diag}(C) = 0, \tag{13}$$

where C_i is the i-th column of C. Since this problem is combinatorial, a simpler ℓ_1 optimization problem is solved

$$\min_C \|C\|_1 \text{ s.t. } D = DC \text{ and } \text{diag}(C) = 0. \tag{14}$$

It is shown in Elhamifar and Vidal (2009, 2010, 2013) that under some conditions on the subspaces and the data, the solutions to the optimization problems in (13) and (14) coincide. It is also shown that $C_{ij} = 0$ when points i and j are in different subspaces. In other words, the nonzero coefficients of the i-th column of C correspond to points in the same subspace as point i. Therefore, one can use C to define an affinity matrix as $|C| + |C^\top|$. The segmentation of the data is then obtained by applying spectral clustering (von Luxburg, 2007) to this affinity.

In the case of data contaminated by noise G, the SSC algorithm assumes that each data point can be written as a linear combination of other data points up to an error G, i.e., $D = DC + G$, and solves the following convex problem

$$\min_{C,G} \|C\|_1 + \frac{\alpha}{2}\|G\|_F^2 \text{ s.t. } D = DC + G \text{ and } \text{diag}(C) = 0. \tag{15}$$

In the case of data contaminated also by gross errors E, the SSC algorithm assumes that $D = DC + G + E$, where E is sparse. Since both C and E are sparse, the equation $D = DC + G + E = [D\ I][C^\top\ E^\top]^\top + G$ means that each point is written as a sparse linear combination of a dictionary composed of all other data points plus the columns of the identity matrix I. Thus, one can find C by solving the following convex optimization problem

$$\min_{C,G,E} \|C\|_1 + \frac{\alpha}{2}\|G\|_F^2 + \gamma\|E\|_1 \tag{16}$$

$$\text{s.t. } D = DC + G + E \text{ and } \text{diag}(C) = 0. \tag{17}$$

While SSC works well in practice, until recently there was no theoretical guarantee that, in the case of corrupted data, the nonzero coefficients correspond to points in the same subspace.[1] Moreover, notice that the model is not really a subspace plus error model, because a contaminated data point is written as a linear combination of other contaminated points plus an error. To the best of our knowledge, there is no method that tries to simultaneously recover a clean dictionary and cluster the data within this framework.

Low Rank Representation (LRR). This algorithm (Liu et al., 2010) is very similar to SSC, except that it aims to find a low-rank representation instead of a sparse representation. This is motivated by the fact that, in the case of uncorrupted data drawn from n independent subspaces of dimensions $r = \{d_i\}_{i=1}^n$, the rank of the data matrix is $\text{rank}(D) = \sum_{i=1}^n d_i$. Thus, the LRR algorithm finds C by solving the following convex optimization problem

$$\min_C \|C\|_* \quad \text{s.t.} \quad D = DC. \tag{18}$$

It is shown in Liu et al. (2011) that in the case of uncorrupted data drawn from independent linear subspaces, the optimal solution to (18) is given by the matrix $C = V_1 V_1^\top$, where $D = U_1 \Sigma_1 V_1^\top$ is the rank r SVD of D. As shown in Vidal et al. (2008), this matrix is such that $C_{ij} = 0$ when points i and j are in different subspaces, hence it can be used to build an affinity matrix.

In the case of data contaminated by noise or gross errors, the LRR algorithm solves the convex optimization problem

$$\min_C \|C\|_* + \gamma\|E\|_{2,1} \text{ s.t. } D = DC + E, \tag{19}$$

where $\|E\|_{2,1} = \sum_{k=1}^N \sqrt{\sum_{j=1}^N |E_{jk}|^2}$ is the $\ell_{2,1}$ norm of the matrix of errors E. Notice that this problem is analogous to (15) and (17), except that the ℓ_1 and the Frobenius norms are replaced by the nuclear and the $\ell_{2,1}$ norms, respectively. It is argued in Liu et al. (2010) that this allows one to better handle outliers, since it is a convex relaxation to the number of corrupted data points, rather than the number of corrupted entries.

The LRR algorithm proceeds by solving the optimization problem in (19) using an ALM method. The optimal C is then used to define an affinity matrix $|C| + |C^\top|$. The segmentation of the data is then obtained by applying spectral clustering to the normalized Laplacian.

[1] We refer the reader to Soltanolkotabi et al. (2013) for very recent results in this direction.

2.3. The Von Neumann Trace Inequality

In this section, we review two matrix product inequalities, which we will use later in our derivations.

Lemma 1 (Von Neumann's Inequality). *For any $m \times n$ real valued matrices X and Y,*

$$trace(X^\top Y) \leq \sum_{i=1}^{n} \sigma_i(X)\sigma_i(Y), \tag{20}$$

where $\sigma_1(X) \geq \sigma_2(X) \geq \cdots \geq 0$ and $\sigma_1(Y) \geq \sigma_2(Y) \geq \cdots \geq 0$ are the descending singular values of X and Y respectively. The case of equality occurs if and only if it is possible to find unitary matrices U_X and V_X that simultaneously singular value-decompose *X and Y in the sense that*

$$X = U_X \Sigma_X V_X^\top \quad and \quad Y = U_X \Sigma_Y V_X^\top, \tag{21}$$

where Σ_X and Σ_Y denote the $m \times n$ diagonal matrices with the singular values of X and Y, respectively, down in the diagonal.

Proof. See Mirsky (1975). ∎

Lemma 2. *For any $n \times n$ real valued, symmetric positive definite matrices X and Z,*

$$trace(XZ) \geq \sum_{i=1}^{n} \sigma_i(X)\sigma_{n-i+1}(Z), \tag{22}$$

where $\sigma_1(X) \geq \sigma_2(X) \geq \cdots \geq 0$ and $\sigma_1(Z) \geq \sigma_2(Z) \geq \cdots \geq 0$ are the descending singular values of X and Z, respectively. The case of equality occurs if and only if it is possible to find a unitary matrix U_X that simultaneously singular value-decomposes *X and Z in the sense that*

$$X = U_X \Sigma_X U_X^\top \quad and \quad Z = U_X \Pi \Sigma_Z \Pi^\top U_X^\top, \tag{23}$$

where Σ_X and Σ_Z denote the $n \times n$ diagonal matrices with the singular values of X and Z, respectively, down in the diagonal in descending order, and Π is a permutation matrix such that $\Pi\Sigma_Z\Pi^\top$ contains the singular values of Z in the diagonal in ascending order.

Proof. Let $Y = \lambda I - Z$, where $\lambda \geq \|Z\|_2$. Then $\text{trace}(XY) = \text{trace}(X(\lambda I - Z)) = \lambda\text{trace}(X) - \text{trace}(XZ)$. Also,

$$\text{trace}(XY) \leq \sum_{i=1}^{n} \sigma_i(X)\sigma_i(Y) = \sum_{i=1}^{n} \sigma_i(X)\sigma_i(\lambda I - Z) = \tag{24}$$

$$\sum_{i=1}^{n} \sigma_i(X)\big(\lambda - \sigma_{n-i+1}(Z)\big) = \lambda\text{trace}(X) - \sum_{i=1}^{n} \sigma_i(X)\sigma_{n-i+1}(Z).$$

It follows from Lemma 1 that $\text{trace}(XZ) \geq \sum_{i=1}^{n} \sigma_i(X)\sigma_{n-i+1}(Z)$, as claimed. Moreover, the equality is achieved if an only if there exists a matrix U_X (recall that X and Z are symmetric) such that $X = U_X\Sigma_X U_X^\top$ and $Y = U_X\Sigma_Y U_X^\top$. Therefore,

$$\begin{aligned} Z = \lambda I - Y &= \lambda I - U_X\Sigma_Y U_X^\top = \lambda I - U_X\Sigma_{\lambda I - Z} U_X^\top \\ &= \lambda I - U_X(\lambda I - \Pi\Sigma_Z\Pi^\top)U_X^\top \\ &= U_X\Pi\Sigma_Z\Pi^\top U_X^\top \end{aligned} \tag{25}$$

as claimed. ∎

3. Low Rank Subspace Clustering with Uncorrupted Data

In this section, we consider the low rank subspace clustering problem in the case of uncorrupted data. That is, we consider problems P_1 and P_2 with $\alpha = \infty$ and $\lambda = \infty$, so that $G = E = 0$ and $D = A$. In Section 3.1, we study the relaxed problem P_2 and show that the optimal solution for C can be obtained in closed form from the SVD of A by applying a nonlinear thresholding to its singular values. In Section 3.2, we study the exact problem P_1, whose optimal solution is obtained by hard thresholding of the singular values of A, as shown in Liu et al. (2011). However, we provide a much simpler derivation of the result.

3.1. Uncorrupted Data and Relaxed Constraints

Consider the following optimization problem

$$(P_3) \qquad \min_C \|C\|_* + \frac{\tau}{2}\|A - AC\|_F^2 \ \text{ s.t. } \ C = C^\top,$$

where $\tau > 0$ is a parameter. Notice that this cost function is convex on C, but not strictly convex. Therefore, we do not know a priory if the solution to P_3 is unique. The following theorem shows that the minimizer of P_3 is unique and can be computed in closed form from the SVD of A.

Theorem 1. *Let $A = U\Lambda V^\top$ be the SVD of A, where the diagonal entries of $\Lambda = diag(\{\lambda_i\})$ are the singular values of A in decreasing order. The optimal solution to P_3 is*

$$C = V\mathcal{P}_\tau(\Lambda)V^\top = V_1\left(I - \frac{1}{\tau}\Lambda_1^{-2}\right)V_1^\top, \tag{26}$$

where the operator $\mathcal{P}_\tau$ acts on the diagonal entries of Λ as

$$\mathcal{P}_\tau(x) = \begin{cases} 1 - \frac{1}{\tau x^2} & x > 1/\sqrt{\tau} \\ 0 & x \leq 1/\sqrt{\tau} \end{cases}, \tag{27}$$

and $U = [U_1\ U_2]$, $\Lambda = diag(\Lambda_1, \Lambda_2)$ and $V = [V_1\ V_2]$ are partitioned according to the sets $\mathbf{I}_1 = \{i : \lambda_i > 1/\sqrt{\tau}\}$ and $\mathbf{I}_2 = \{i : \lambda_i \leq 1/\sqrt{\tau}\}$. Moreover, the optimal value is

$$\Phi_\tau(A) = \sum_{i \in \mathbf{I}_1}\left(1 - \frac{1}{2\tau}\lambda_i^{-2}\right) + \frac{\tau}{2}\sum_{i \in \mathbf{I}_2}\lambda_i^2. \tag{28}$$

Proof. Let $A = U\Lambda V^\top$ be the SVD of A and $C = U_C\Delta U_C^\top$ be the eigenvalue decomposition (EVD) of C. The cost function of P_3 reduces to

$$\begin{aligned} &\|U_C\Delta U_C^\top\|_* + \frac{\tau}{2}\|U\Lambda V^\top(I - U_C\Delta U_C^\top)\|_F^2 = \\ &\|\Delta\|_* + \frac{\tau}{2}\|\Lambda V^\top U_C(I - \Delta)U_C^\top\|_F^2 = \|\Delta\|_* + \frac{\tau}{2}\|\Lambda W(I - \Delta)\|_F^2, \end{aligned} \tag{29}$$

where $W = V^\top U_C$. To minimize this cost with respect to W, we only need to consider the last term of the cost function, i.e.,

$$\|\Lambda W(I - \Delta)\|_F^2 = \text{trace}((I - \Delta)^2 W^\top \Lambda^2 W). \tag{30}$$

Applying Lemma 2 to $X = W(I-\Delta)^2W^\top$ and $Z = \Lambda^2$, we obtain that for all unitary matrices W

$$\min_W \text{trace}((I - \Delta)^2 W^\top \Lambda^2 W) = \sum_{i=1}^{N} \sigma_i((I - \Delta)^2)\sigma_{n-i+1}(\Lambda^2), \tag{31}$$

where the minimum is achieved by a permutation matrix $W = \Pi^\top$ that sorts the diagonal entries of Λ^2 in ascending order, i.e., the diagonal entries of $\Pi\Lambda^2\Pi^\top$ are in ascending order.

Let the i-th largest entry of $(I-\Delta)^2$ and Λ^2 be, respectively, $(1-\delta_i)^2 = \sigma_i((I-\Delta)^2)$ and $\gamma_{n-i+1}^2 = \lambda_i^2 = \sigma_i(\Lambda^2)$. Then

$$\min_W \|\Delta\|_* + \frac{\tau}{2}\|\Lambda W(I-\Delta)\|_F^2 = \sum_{i=1}^N |\delta_i| + \frac{\tau}{2}\sum_{i=1}^N \gamma_i^2(1-\delta_i)^2. \quad (32)$$

To find the optimal Δ, we take the derivative of the cost with respect to δ_i and set it to zero, which yields

$$\frac{\delta_i}{|\delta_i|} - \tau\gamma_i^2(1-\delta_i) = 0 \implies \delta_i + \frac{1}{\tau\gamma_i^2}\frac{\delta_i}{|\delta_i|} = 1. \quad (33)$$

This equation can be solved in closed form by using the shrinkage-thresholding operator in (6), which gives

$$\delta_i = \mathcal{S}_{\frac{1}{\tau\gamma_i^2}}(1) = \begin{cases} 1 - \frac{1}{\tau\gamma_i^2} & \gamma_i > 1/\sqrt{\tau} \\ 0 & \gamma_i \leq 1/\sqrt{\tau} \end{cases}. \quad (34)$$

Then, $\delta_i = \mathcal{P}_\tau(\lambda_{n-i+1})$, which can be compactly written as $\Delta = \Pi\mathcal{P}_\tau(\Lambda)\Pi^\top$. Therefore,

$$\Pi^\top\Delta\Pi = \mathcal{P}_\tau(\Lambda) = \begin{bmatrix} I - \frac{1}{\tau}\Lambda_1^{-2} & 0 \\ 0 & 0 \end{bmatrix}, \quad (35)$$

where $\Lambda = \mathrm{diag}(\Lambda_1, \Lambda_2)$ is partitioned according to the sets $\mathbf{I}_1 = \{i : \lambda_i > 1/\sqrt{\tau}\}$ and $\mathbf{I}_2 = \{i : \lambda_i \leq 1/\sqrt{\tau}\}$.

To find the optimal W, notice from Lemma 2 that the equality $\mathrm{trace}((I-\Delta)^2 W^\top\Lambda^2 W) = \sum_{i=1}^N (1-\delta_i)^2\lambda_{n-i+1}^2$ is achieved if and only if there exists a unitary matrix U_X such that

$$(I-\Delta)^2 = U_X(I-\Delta)^2 U_X^\top \text{ and } W^\top\Lambda^2 W = U_X\Pi\Lambda^2\Pi^\top U_X^\top. \quad (36)$$

Since the SVD of a matrix is unique up to the sign of the singular vectors associated with different singular values and up to a rotation and sign of the singular vectors associated with repeated singular values, we conclude that $U_X = I$ up to the aforementioned ambiguities of the SVD of $(I-\Delta)^2$. Likewise, we have that $W^\top = U_X\Pi$ up to the aforementioned ambiguities of the SVD of Λ^2. Now, if Λ^2 has repeated singular values, then $(I-\Delta)^2$ has repeated eigenvalues at the same locations. Therefore, $W^\top = U_X\Pi = \Pi$ up to a block-diagonal transformation, where each block is an orthonormal matrix that corresponds to a repeated singular value of Δ. Nonetheless, even though W may not be unique, the matrix C is always unique and equal to

$$\begin{aligned} C = U_C\Delta U_C^\top &= VW\Delta W^\top V^\top = V\Pi^\top\Delta\Pi V^\top \quad (37)\\ &= \begin{bmatrix} V_1 & V_2 \end{bmatrix}\begin{bmatrix} I - \frac{1}{\tau}\Lambda_1^{-2} & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} V_1 & V_2 \end{bmatrix}^\top = V_1(I - \frac{1}{\tau}\Lambda_1^{-2})V_1^\top, \end{aligned}$$

as claimed.

Finally, the optimal C is such that $AC = U_1(\Lambda_1 - \frac{1}{\tau}\Lambda_1^{-1})V_1^\top$ and $A - AC = U_2\Lambda_2 V_2^\top + \frac{1}{\tau}U_1\Lambda_1 V_1^\top$. This shows (28), because

$$\|C\|_* + \frac{\tau}{2}\|A - AC\|_F^2 = \sum_{i\in\mathbf{I}_1}\left(1 - \frac{1}{\tau}\lambda_i^{-2}\right) + \frac{\tau}{2}\left(\sum_{i\in\mathbf{I}_1}\frac{\lambda_i^{-2}}{\tau^2} + \sum_{i\in\mathbf{I}_2}\lambda_i^2\right),$$

as claimed. ∎

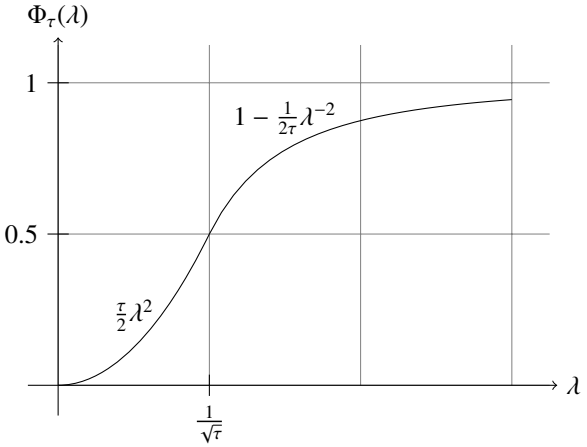

Figure 1: Plot of $\Phi_\tau(\lambda)$.

Notice that the optimal value of P_3, $\Phi_\tau(A)$, is a decomposable function of the singular values of A, as are the Frobenius and nuclear norms of A, $\|A\|_F^2 = \sum\lambda_i^2$ and $\|A\|_* = \sum\lambda_i$, respectively. However, unlike $|A|_F$ or $|A|_*$, $\Phi_\tau(A)$ is not a convex function of A because $\Phi_\tau(\lambda)$ is quadratic near zero and saturates as λ increases, as illustrated in Figure 1. Interestingly, as τ goes to infinity, Φ_τ approaches rank(A), as we shall see. Therefore, we may view $\Phi_\tau(A)$ as a non-convex relaxation of rank(A).

3.2. Uncorrupted Data and Exact Constraints

Consider now the optimization problem

$$(P_4) \qquad \min_C \|C\|_* \text{ s.t. } A = AC \text{ and } C = C^\top.$$

The following Theorem shows that the Costeira and Kanade affinity matrix $C = V_1V_1^\top$ is the optimal solution to P_5. The theorem follows from Theorem 1 by letting $\tau \to \infty$. An alternative proof can be found in Liu et al. (2011). Here, we provide a simpler and more direct proof.

Theorem 2. *Let $A = U\Lambda V^\top$ be the SVD of A, where the diagonal entries of $\Lambda = diag(\{\lambda_i\})$ are the singular values of A in decreasing order. The optimal solution to P_4 is*

$$C = V_1V_1^\top, \quad (38)$$

where $V = [V_1\ V_2]$ is partitioned according to the sets $\mathbf{I}_1 = \{i : \lambda_i > 0\}$ and $\mathbf{I}_2 = \{i : \lambda_i = 0\}$. Moreover, the optimal value is

$$\Phi_\infty(A) = \sum_{i\in\mathbf{I}_1} 1 = rank(A). \quad (39)$$

Proof. Let $C = U_C\Delta U_C^\top$ be the EVD of C. Then $A = AC$ can be rewritten as $U\Lambda V^\top = U\Lambda V^\top U_C\Delta U_C^\top$, which reduces to

$$\Lambda V^\top U_C = \Lambda V^\top U_C\Delta \quad (40)$$

since $U^\top U = I$ and $U_C^\top U_C = I$. Let $W = V^\top U_C = \begin{bmatrix} w_1, \cdots, w_N \end{bmatrix}$. Then, $\Lambda w_j = \Lambda w_j\delta_j$ for all $j = 1, \ldots, N$. This means that $\delta_j = 1$ if $\Lambda w_j \neq \mathbf{0}$ and δ_j is arbitrary otherwise. Since our

goal is to minimize $\|C\|_* = \|\Delta\|_* = \sum_{j=1}^{N} |\delta_j|$, we need to set as many δ_j's to zero as possible. Since $A = AC$ implies that $\text{rank}(A) \leq \text{rank}(C)$, we can set at most $N - \text{rank}(A)$ δ_j's to zero and the remaining $\text{rank}(A)$ δ_j's must be equal to one. Now, if $\delta_j = 0$, then $\Lambda w_j = \Lambda_1 V_1^\top U_C e_j = \mathbf{0}$, where e_j is the j-th column of the identity. This means that the columns of U_C associated to $\delta_j = 0$ must be orthogonal to the columns of V_1, and hence the columns of U_C associated with $\delta_j = 1$ must be in the range of V_1. Thus, $U_C = \begin{bmatrix} V_1 R_1 & U_2 R_2 \end{bmatrix} \Pi$ for some rotation matrices R_1 and R_2, and permutation matrix Π, and so the optimal C is

$$C = \begin{bmatrix} V_1 R_1 & V_2 R_2 \end{bmatrix} \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} V_1 R_1 & V_2 R_2 \end{bmatrix}^\top = V_1 V_1^\top, \quad (41)$$

as claimed. ∎

4. Low Rank Subspace Clustering with Noisy Data

In this section, we consider the low rank subspace clustering problem in the case of noisy data. That is, we consider problems P_1 and P_2 with $\lambda = \infty$, so that $E = 0$ and $D = A + G$. While in principle the resulting problems appear to be very similar to those in (15) and (19), there are a number of differences. First, notice that instead of expressing the noisy data as a linear combination of itself plus noise, i.e., $D = DC + G$, we search for a clean dictionary, A, which is self-expressive, i.e., $A = AC$. We then assume that the data are obtained by adding noise to the clean dictionary, i.e., $D = A + G$. As a consequence, our method searches simultaneously for a clean dictionary A, the coefficients C and the noise G. Second, the main difference with (15) is that the ℓ_1 norm of the matrix of the coefficients is replaced by the nuclear norm. Third, the main difference with (19) is that the $\ell_{2,1}$ norm of the matrix of the noise is replaced by the Frobenius norm. Fourth, our method enforces the symmetry of the affinity matrix as part of the optimization problem, rather than as a post-processing step.

As we will show in this section, these modifications result in a key difference between our method and the state of the art: while the solution to (15) requires ℓ_1 minimization and the solution to (19) requires an ALM method, the solutions to P_5 and P_6 can be computed in closed form from the SVD of the data matrix D. For the relaxed problem, P_2, the closed-form solution for A is found by applying a polynomial thresholding to the singular values of D, as we will see in Section 4.1. For the exact problem, P_1, the closed-form solution for A is given by classical PCA, except that the number of principal components can be automatically determined, as we will see in Section 4.2.

4.1. Noisy Data and Relaxed Constraints

In this section, we assume that the data are contaminated by noise, i.e., $D = A + E$, and relax the constraint $A = AC$ by adding a penalty to the cost. More specifically, we consider the optimization problem

$$(P_5) \quad \min_{A,C} \ \|C\|_* + \frac{\tau}{2}\|A - AC\|_F^2 + \frac{\alpha}{2}\|D - A\|_F^2 \ \text{ s.t. } \ C = C^\top.$$

The key difference with respect to the problem considered in Section 3.1 is that A is unknown. Hence the cost function in P_5 is not convex in (A, C) because of the product AC. Nonetheless, we will show in this subsection that the optimal solution is still unique, unless one of the singular values of D satisfies a constraint that depends on α and τ. In such a degenerate case, the problem has two optimal solutions. Moreover, the optimal solutions for both A and C can be computed in closed form from the SVD of D, as stated in the following theorem.

Theorem 3. *Let $D = U\Sigma V^\top$ be the SVD of the data matrix D. The optimal solutions to P_5 are of the form*

$$A = U\Lambda V^\top \ \text{ and } \ C = V\mathcal{P}_\tau(\Lambda)V^\top, \quad (42)$$

where each entry of $\Lambda = diag(\lambda_1, \ldots, \lambda_n)$ is obtained from each entry of $\Sigma = diag(\sigma_1, \ldots, \sigma_n)$ as the solutions to

$$\sigma = \psi(\lambda) = \begin{cases} \lambda + \frac{1}{\alpha\tau}\lambda^{-3} & \text{if } \lambda > 1/\sqrt{\tau} \\ \lambda + \frac{\tau}{\alpha}\lambda & \text{if } \lambda \leq 1/\sqrt{\tau} \end{cases}, \quad (43)$$

that minimize

$$\phi(\lambda, \sigma) = \frac{\alpha}{2}(\sigma - \lambda)^2 + \begin{cases} 1 - \frac{1}{2\tau}\lambda^{-2} & \lambda > 1/\sqrt{\tau} \\ \frac{\tau}{2}\lambda^2 & \lambda \leq 1/\sqrt{\tau} \end{cases}. \quad (44)$$

The solution for each λ, hence for A and C, is unique, except when D has a singular value σ such that (56) *holds.*

The proof of this result will be done in three steps. First, we will use Theorem 1 to show that C can be computed in closed form from the SVD of A. Second, we will show that the optimal A can be obtained in closed form from the SVD of D. Third, we will study conditions under which the solution is unique.

A Closed-Form Solution for C. Notice that when A is fixed, P_5 reduces to P_3. Therefore, it follows from Theorem 1 that the optimal solution for C is $C = V\mathcal{P}_\tau(\Lambda)V^\top$, where $A = U\Lambda V^\top$ is the SVD of A. Moreover, it follows from (28) that if we replace the optimal C into the cost of P_5, then P_5 is equivalent to

$$\min_A \ \Phi_\tau(A) + \frac{\alpha}{2}\|D - A\|_F^2. \quad (45)$$

A Closed-Form Solution for A. To solve (45), let $D = U\Sigma V^\top$ and $A = U_A\Lambda V_A^\top$ be the SVDs of D and A, respectively. Then,

$$\begin{aligned} \|D - A\|_F^2 &= \|U\Sigma V^\top - U_A\Lambda V_A^\top\|_F^2 \\ &= \|\Sigma\|_F^2 - 2\text{trace}(V\Sigma U^\top U_A\Lambda V_A^\top) + \|\Lambda\|_F^2 \\ &= \|\Sigma\|_F^2 - 2\text{trace}(\Sigma W_1 \Lambda W_2^\top) + \|\Lambda\|_F^2, \end{aligned} \quad (46)$$

where $W_1 = U^\top U_A$ and $W_2 = V^\top V_A$. Therefore, the minimization over A in (45) can be carried out by minimizing first with respect to W_1 and W_2 and then with respect to Λ.

The minimization over W_1 and W_2 is equivalent to

$$\max_{W_1, W_2} \text{trace}(\Sigma W_1 \Lambda W_2^\top). \quad (47)$$

By letting $X = \Sigma$ and $Y = W_1 \Lambda W_2^\top$ in Lemma 1, we obtain

$$\max_{W_1, W_2} \operatorname{trace}(\Sigma W_1 \Lambda W_2^\top) = \sum_{i=1}^{n} \sigma_i(\Sigma)\sigma_i(\Lambda) = \sum_{i=1}^{n} \sigma_i \lambda_i. \quad (48)$$

Moreover, the maximum is achieved if and only if there exist orthogonal matrices U_W and V_W such that

$$\Sigma = U_W \Sigma V_W^\top \quad \text{and} \quad W_1 \Lambda W_2^\top = U_W \Lambda V_W^\top. \quad (49)$$

Hence, the optimal solutions are $W_1 = U_W = I$ and $W_2 = V_W = I$ up to a unitary transformation that accounts for the sign and rotational ambiguities of the singular vectors of Σ. This means that A and D have the same singular vectors, i.e., $U_A = U$ and $V_A = V$, and that $\|D - A\|_F^2 = \|U(\Sigma - \Lambda)V^\top\|_F^2 = \|\Sigma - \Lambda\|_F^2$. By substituting this expression for $\|D - A\|_F^2$ into (45), we obtain

$$\min_{\Lambda} \sum_{i \in \mathbf{I}_1} (1 - \frac{1}{2\tau}\lambda_i^{-2}) + \frac{\tau}{2} \sum_{i \in \mathbf{I}_2} \lambda_i^2 + \frac{\alpha}{2} \sum_i (\sigma_i - \lambda_i)^2. \quad (50)$$

where $\mathbf{I}_1 = \{i : \lambda_i > 1/\sqrt{\tau}\}$ and $\mathbf{I}_2 = \{i : \lambda_i \leq 1/\sqrt{\tau}\}$.

It follows from the above equation that the optimal λ_i can be obtained independently for each σ_i by minimizing the ith term of the above summation, which is of the form $\phi(\lambda, \sigma)$ in (44). The first order derivative of ϕ is given by

$$\frac{\partial \phi}{\partial \lambda} = \alpha(\lambda - \sigma) + \begin{cases} \frac{1}{\tau}\lambda^{-3} & \lambda > 1/\sqrt{\tau} \\ \tau\lambda & \lambda \leq 1/\sqrt{\tau} \end{cases}. \quad (51)$$

Therefore, the optimal λ's can be obtained as the solution of the nonlinear equation $\sigma = \psi(\lambda)$, as claimed in (43).

Uniqueness of the Closed Form Solution. When $3\tau \leq \alpha$, the solution for λ is unique, as shown in Figure 2. This is because

$$\frac{\partial^2 \phi}{\partial \lambda^2} = \begin{cases} \alpha - \frac{3}{\tau}\lambda^{-4} & \lambda > 1/\sqrt{\tau} \\ \alpha + \tau & \lambda \leq 1/\sqrt{\tau} \end{cases} \quad (52)$$

is strictly positive, hence ϕ is a strictly convex function of λ.

When $3\tau > \alpha$, the solution is unique if $\sigma < \sigma_1 \triangleq \frac{4}{3}\sqrt[4]{\frac{3}{\alpha\tau}}$ or $\sigma > \sigma_3 \triangleq \frac{\alpha+\tau}{\alpha\sqrt{\tau}}$, as illustrated in Figure 3. However, when $\sigma_1 \leq \sigma \leq \sigma_3$ there could be up to three different solutions. The first candidate solution can be computed in closed form as

$$\lambda_1(\sigma) = \frac{\alpha}{\alpha + \tau}\sigma. \quad (53)$$

The remaining two candidate solutions λ_2 and λ_3 can be computed as the two real roots of the polynomial

$$p(\lambda) = \lambda^4 - \sigma\lambda^3 + \frac{1}{\alpha\tau} = 0, \quad (54)$$

with λ_2 being the smallest and λ_3 being the largest root. The other two roots of p are complex. Out of the three candidate solutions, λ_1 and λ_3 correspond to a minimum and λ_2 corresponds to a maximum. This is because

$$\lambda_1 \leq 1/\sqrt{\tau}, \quad \lambda_2 < \sqrt[4]{\frac{3}{\alpha\tau}} \quad \text{and} \quad \lambda_3 > \sqrt[4]{\frac{3}{\alpha\tau}}, \quad (55)$$

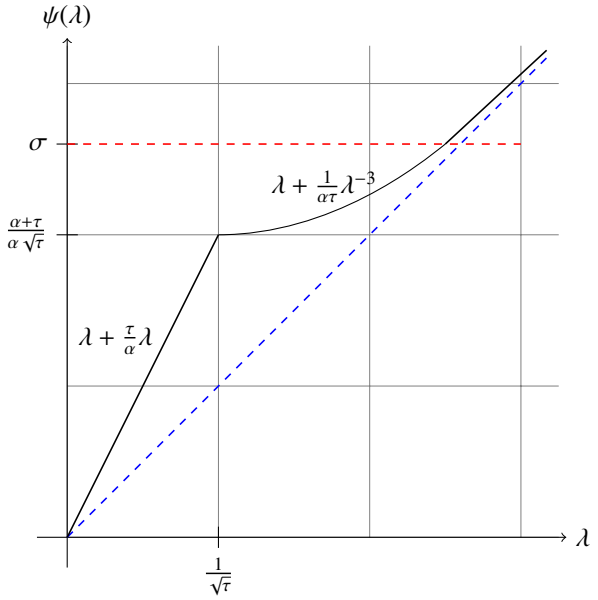

Figure 2: Plot of $\psi(\lambda)$ when $3\tau \leq \alpha$.

and so $\frac{\partial^2 \phi}{\partial \lambda^2}$ is positive for λ_1, negative for λ_2 and positive for λ_3.

Out of the two possible minimizers, only one of them will be a global minimum whenever

$$\phi(\lambda_1(\sigma), \sigma) < \phi(\lambda_3(\sigma), \sigma) \text{ or } \phi(\lambda_1(\sigma), \sigma) > \phi(\lambda_3(\sigma), \sigma).$$

In either of such cases the solution for Λ, hence for A and C, will be unique. The only case in which the solution is not unique is when D has a singular value σ such that

$$\phi(\lambda_1(\sigma), \sigma) = \phi(\lambda_3(\sigma), \sigma), \quad (56)$$

which implies that

$$\frac{\alpha\tau}{2(\alpha + \tau)}\sigma^2 = \frac{\alpha}{2}(\sigma - \lambda_3(\sigma))^2 + 1 - \frac{1}{2\tau}\lambda_3(\sigma)^{-2}. \quad (57)$$

This completes the proof of Theorem 3.

4.1.1. The Polynomial Thresholding Operator $\mathcal{P}_{\alpha,\tau}$

Theorem 3 gives us a way to obtain A from the SVD of the data matrix in closed form. Remarkably, the solution is obtained by applying a *polynomial thresholding* operator $\lambda = \mathcal{P}_{\alpha,\tau}(\sigma)$ to the singular values of D. In what follows, we show that this operator can be computed as

$$\lambda = \mathcal{P}_{\alpha,\tau}(\sigma) = \begin{cases} \lambda_3(\sigma) & \text{if } \sigma > \sigma_* \\ \lambda_1(\sigma) & \text{if } \sigma \leq \sigma_*, \end{cases} \quad (58)$$

for some $\sigma_* > 0$. Moreover, we show that, for some values of α and τ, σ_* can be computed in closed form. Specifically, when $3\tau \leq \alpha$, there is a unique solution for λ, which is given by

$$\lambda = \begin{cases} \lambda_1 & \text{if } \lambda \leq 1/\sqrt{\tau} \\ \lambda_2 = \lambda_3 & \text{if } \lambda > 1/\sqrt{\tau}. \end{cases} \quad (59)$$

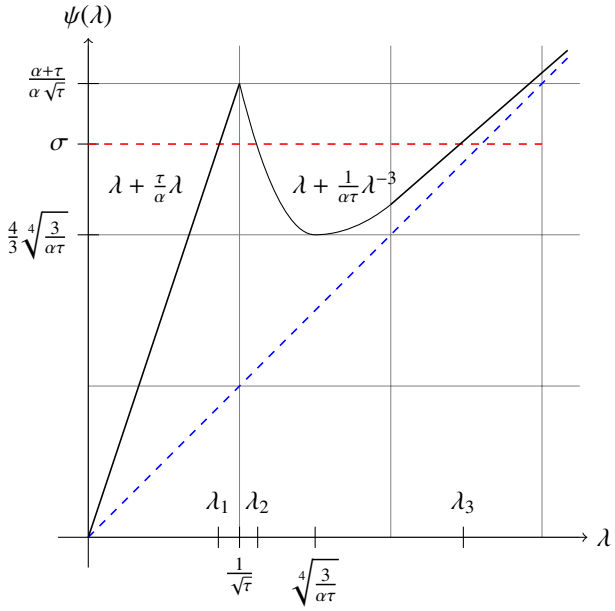

Figure 3: Plot of $\psi(\lambda)$ when $3\tau > \alpha$.

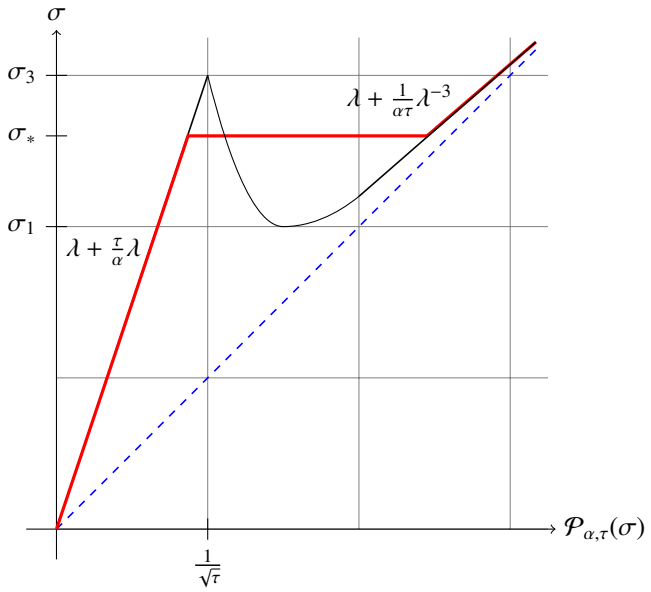

Figure 4: The polynomial thresholding operator.

Thus, when $3\tau \leq \alpha$ we have

$$\sigma_* = \psi\Big(\frac{1}{\sqrt{\tau}}\Big) = \frac{\alpha+\tau}{\alpha\sqrt{\tau}}. \tag{60}$$

When $3\tau > \alpha$, the solution is $\lambda = \lambda_1$ or $\lambda = \lambda_3$ depending on whether $\phi(\lambda_1(\sigma)) < \phi(\lambda_3(\sigma))$ or $\phi(\lambda_1(\sigma)) > \phi(\lambda_3(\sigma))$, respectively. We thus need to show that there exists a $\sigma_* > 0$ such that $\phi(\lambda_1(\sigma)) < \phi(\lambda_3(\sigma))$ for $\sigma < \sigma_*$ and $\phi(\lambda_1(\sigma)) > \phi(\lambda_3(\sigma))$ for $\sigma > \sigma_*$. Because of the intermediate value theorem, it is sufficient to show that

$$f(\sigma) = \phi(\lambda_1(\sigma), \sigma) - \phi(\lambda_3(\sigma), \sigma) \tag{61}$$

is continuous and increasing for $\sigma \in [\sigma_1, \sigma_3]$, negative at σ_1 and positive at σ_3, so that there is a $\sigma_* \in (\sigma_1, \sigma_3)$ such that $f(\sigma_*) = 0$. Recall that $\sigma_1 = \frac{4}{3}\sqrt[4]{\frac{3}{\alpha\tau}}$ and $\sigma_3 = \frac{\alpha+\tau}{\alpha\sqrt{\tau}}$. The function f is continuous in $[\sigma_1, \sigma_3]$, because a) ϕ is a continuous function of (λ, σ), b) the roots of a polynomial (λ_1 and λ_2) vary continuously as a function of the coefficients (σ) and c) the composition of two continuous functions is continuous. Also, f is increasing in $[\sigma_1, \sigma_3]$, because

$$\begin{aligned}\frac{df}{d\sigma} &= \frac{\partial\phi}{\partial\lambda}\Big|_{(\lambda_1,\sigma)}\frac{d\lambda_1}{d\sigma} + \frac{\partial\phi}{\partial\sigma}\Big|_{(\lambda_1,\sigma)} - \frac{\partial\phi}{\partial\lambda}\Big|_{(\lambda_3,\sigma)}\frac{d\lambda_3}{d\sigma} - \frac{\partial\phi}{\partial\sigma}\Big|_{(\lambda_3,\sigma)} \\ &= 0 + \alpha(\sigma - \lambda_1) - 0 - \alpha(\sigma - \lambda_3) = \alpha(\lambda_3 - \lambda_1) > 0.\end{aligned}$$

Now, notice from Figure 3 that when $\sigma < \sigma_1$ the optimal solution is $\lambda = \lambda_1$. When $\sigma = \sigma_1$, $\lambda_1 = \frac{4\alpha}{3(\alpha+\tau)}\sqrt[4]{\frac{3}{\alpha\tau}}$ is a minimum and $\lambda_2 = \lambda_3 = \sqrt[4]{\frac{3}{\alpha\tau}}$ is an inflection point, thus the optimal solution is $\lambda = \lambda_1$.[2] When $\sigma > \sigma_3$, the optimal solution is λ_3. Finally, when $\sigma = \sigma_3$, $\lambda_1 = \lambda_2 = \frac{1}{\sqrt{\tau}}$ is a maximum and λ_3 is a minimum, thus the optimal solution is $\lambda = \lambda_3$. Therefore, the threshold for σ must lie in the range

$$\frac{4}{3}\sqrt[4]{\frac{3}{\alpha\tau}} < \sigma_* < \frac{\alpha+\tau}{\alpha\sqrt{\tau}}. \tag{62}$$

4.1.2. *Approximate Polynomial Thresholding Operator*

Notice, however, that finding a closed-form formula for σ_* is not straightforward, because it requires solving (56). While this equation can be solved numerically for each α and τ, a simple closed form formula can be obtained when $\frac{1}{\alpha\tau} \simeq 0$ (relative to σ). In this case, the quartic becomes $p(\lambda) = \lambda^4 - \sigma\lambda^3 = 0$, which can be immediately solved and yields three solutions that are equal to 0 and are hence out of the range $\lambda > 1/\sqrt{\tau}$. The only valid solution to the quartic is

$$\lambda = \sigma \qquad \forall\sigma : \sigma > 1/\sqrt{\tau}. \tag{63}$$

Thus, a simpler thresholding procedure can be obtained by approximating the thresholding function with two piecewise linear functions. One is exact (when $\lambda \leq 1/\sqrt{\tau}$) and the other one

[2] One can also show that $\phi(\lambda_1(\sigma_1), \sigma_1) < \phi(\lambda_3(\sigma_1), \sigma_1)$ as follows:

$$\begin{aligned}\phi(\lambda_1(\sigma_1), \sigma_1) &= \frac{\alpha\tau}{2}\frac{1}{\alpha+\tau}\frac{16}{9}\sqrt{\frac{3}{\alpha\tau}} = \frac{8\tau}{3(\alpha+\tau)}\sqrt{\frac{\alpha}{3\tau}} \\ \phi(\lambda_3(\sigma_1), \sigma_1) &= 1 - \frac{1}{2\tau}\lambda_3^{-2} + \frac{\alpha}{2}(\sigma - \lambda_3)^2 = 1 - \frac{1}{2\tau}\lambda_3^{-2} + \frac{\alpha}{18}\lambda_3^2 \\ &= 1 - \frac{9 - \alpha\tau\lambda_3^4}{18\tau\lambda_3^2} = 1 - \frac{1}{3\tau\lambda_3^2} = 1 - \frac{1}{3}\sqrt{\frac{\alpha}{3\tau}}.\end{aligned}$$

Therefore, $\phi(\lambda_1(\sigma_1), \sigma_1) < \phi(\lambda_3(\sigma_1), \sigma_1)$ because

$$\frac{1}{3}\Big(\frac{8\tau}{\alpha+\tau} + 1\Big)\sqrt{\frac{\alpha}{3\tau}} < 1 \iff \Big(\frac{9\tau+\alpha}{\alpha+\tau}\Big)^2\frac{\alpha}{27\tau} < 1 \iff (3\tau - \alpha)^3 > 0,$$

which follows from the fact that $3\tau > \alpha$.

is approximate (when $\lambda > 1/\sqrt{\tau}$). The approximation, however, is quite accurate for a wide range of values for α and τ. Since we have two linear functions, we can easily find a threshold for σ as the value σ_* at which the discontinuity happens. To do so, we can plug in the given solutions in (56). We obtain

$$\frac{\alpha\tau}{2(\alpha+\tau)}\sigma_*^2 = 1 - \frac{1}{2\tau\sigma_*^2}. \tag{64}$$

This gives 4 solutions, out of which the only suitable one is

$$\sigma_* = \sqrt{\frac{\alpha+\tau}{\alpha\tau} + \sqrt{\frac{\alpha+\tau}{\alpha^2\tau}}}. \tag{65}$$

Finally, the approximate polynomial thresholding operator can be written as

$$\lambda = \widetilde{\mathcal{P}}_{\alpha,\tau}(\sigma) = \begin{cases} \sigma & \text{if } \sigma > \sigma_* \\ \frac{\alpha}{\alpha+\tau}\sigma & \text{if } \sigma \leq \sigma_*. \end{cases} \tag{66}$$

Notice that as τ increases, the largest singular values of D are preserved, rather than shrank by the operator $\mathcal{S}_{\alpha^{-1}}$ in (6). Notice also that the smallest singular values of D are shrank by scaling them down, as opposed to subtracting a threshold.

4.2. Noisy Data and Exact Constraints

In this section, we assume that the data is generated from the exact self-expressive model, $A = AC$, and contaminated by noise, i.e., $D = A + G$. This leads to the optimization problem

$$(P_6) \quad \min_{A,C} \ \|C\|_* + \frac{\alpha}{2}\|D - A\|_F^2 \ \text{ s.t. } \ A = AC \text{ and } \ C = C^\top.$$

This problem can be seen as the limiting case of P_5 with $\tau \to \infty$. In this case, the polynomial thresholding operator reduces to the hard thresholding operator $\mathcal{H}_\epsilon$ in (2) with threshold $\epsilon = \sigma_* = \sqrt{\frac{2}{\alpha}}$. Therefore, the optimal A can be obtained from the SVD of $D = U\Sigma V^\top$ as $A = U\mathcal{H}_{\sqrt{\frac{2}{\alpha}}}(\Sigma)V^\top$, while the optimal C is given by Theorem 2. We thus have the following result.

Theorem 4. *Let $D = U\Sigma V^\top$ be the SVD of the data matrix D. The optimal solution to P_6 is given by*

$$A = U_1\Sigma_1 V_1^\top \quad \textit{and} \quad C = V_1 V_1^\top, \tag{67}$$

where Σ_1 contains the singular values of D that are larger than $\sqrt{\frac{2}{\alpha}}$, and U_1 and V_1 contain the corresponding singular vectors.

5. Low Rank Subspace Clustering with Corrupted Data

In this section, we consider the low-rank subspace clustering problem in the case of data corrupted by noise and gross errors, i.e., we consider problems P_1 and P_2. Similar to the case of noisy data discussed in Section 4, the major difference between these optimization problems and those in (17) and (19) is that, rather than using a corrupted dictionary, we search simultaneously for a clean dictionary A, the low-rank coefficients C and the sparse errors E. Also, notice that the ℓ_1 norm of the matrix of coefficients is replaced by the nuclear norm, that the $\ell_{2,1}$ norm of the matrix of errors is replaced by the ℓ_1 norm, and that we enforce the symmetry of the affinity matrix as part of the optimization problem, rather than as a post-processing. A closed form solution to the low-rank subspace clustering problem in the case of data corrupted by noise and gross errors appears elusive at this point. Therefore, we propose to solve P_1 and P_2 using an alternating minimization approach, as described next.

5.1. Corrupted Data and Relaxed Constraints

Iterative Polynomial Thresholding (IPT). We begin by considering the relaxed problem P_2, which is equivalent to

$$\begin{aligned} &\min_{A,C,E} \ \|C\|_* + \frac{\tau}{2}\|A - AC\|_F^2 + \frac{\alpha}{2}\|D - A - E\|_F^2 + \gamma\|E\|_1 \\ &\text{s.t. } \ C = C^\top. \end{aligned} \tag{68}$$

When E is fixed, this problem reduces to P_5, except that D is replaced by $D - E$. Therefore, it follows from Theorem 3 that A and C can be computed from the SVD of $D - E = U\Sigma V^\top$ as

$$A = U\mathcal{P}_{\alpha,\tau}(\Sigma)V^\top \quad \text{and} \quad C = V\mathcal{P}_\tau(\mathcal{P}_{\alpha,\tau}(\Sigma))V^\top, \tag{69}$$

where P_τ is the operator in (27) and $P_{\alpha,\tau}$ is the polynomial thresholding operator in (58).

When A and C are fixed, the optimal solution for E satisfies

$$-\alpha(D - A - E) + \gamma\,\text{sign}(E) = 0. \tag{70}$$

This equation can be solved in closed form by using the shrinkage-thresholding operator in (6) and the solution is

$$E = \mathcal{S}_{\frac{\gamma}{\alpha}}(D - A). \tag{71}$$

This suggest an iterative thresholding algorithm that, starting from $A_0 = D$ and $E_0 = 0$, alternates between applying polynomial thresholding to $D - E_k$ to obtain A_{k+1} and applying shrinkage-thresholding to $D - A_{k+1}$ to obtain E_{k+1}, i.e.

$$\begin{aligned} (U_k, \Sigma_k, V_k) &= \text{svd}(D - E_k) \\ A_{k+1} &= U_k\mathcal{P}_{\alpha,\tau}(\Sigma_k)V_k^\top \\ E_{k+1} &= \mathcal{S}_{\gamma\alpha^{-1}}(D - A_{k+1}). \end{aligned} \tag{72}$$

Notice that, since the updates for A and E do not depend on C, we do not need to compute C at each iteration: we can simply obtain C from A upon convergence.

Although the optimization problem in (68) is non-convex, the algorithm in (72) is guaranteed to converge, as shown in Tseng (2001). Specifically, it follows from Theorem 1 that the optimization problem in (68) is equivalent to the minimization of the cost function

$$f(A, E) = \Phi_\tau(A) + \frac{\alpha}{2}\|D - A - E\|_F^2 + \gamma\|E\|_1. \tag{73}$$

It is easy to see that the algorithm in (72) is a coordinate descent method applied to the minimization of f. This function is continuous, has a compact level set $\{(A, E) : f(A, E) \leq f(A_0, E_0)\}$,

and has at most one minimum in E as per (71). Therefore, it follows from Theorem 4.1 part (c) in Tseng (2001) that the algorithm in (72) converges to a coordinate-wise minimum of f.

Notice, however, this minimum is not guaranteed to be a global minimum. Moreover, in practice its convergence can be slow as observed in Lin et al. (2011) for similar problems.

Alternating Direction Method of Multipliers (ADMM). We now propose an alternative solution to P_2 in which we enforce the constraint $D = A + E$ exactly. This means that we tolerate outliers, but we do not tolerate noise. Using the method of multipliers, this problem can be formulated as

$$\max_{Y} \min_{A,E,C:C=C^\top} \|C\|_* + \frac{\tau}{2}\|A - AC\|_F^2 + \frac{\mu}{2}\|D - A - E\|_F^2 + \langle Y, D - A - E\rangle + \gamma\|E\|_1. \quad (74)$$

In this formulation, the term with μ does not play the role of penalizing the noise $G = D-A-E$, as before. Instead, it augments the Lagrangian with the squared norm of the constraint.

To solve the minimization problem over (A, E, C), notice that when E is fixed the optimization over A and C is equivalent to

$$\min_{A,C} \|C\|_* + \frac{\tau}{2}\|A - AC\|_F^2 + \frac{\mu}{2}\|D - A - E + \mu^{-1}Y\|_F^2 \text{ s.t. } C = C^\top.$$

It follows from Theorem 3 that the optimal solutions for A and C can be computed from the SVD of $D - E + \mu^{-1}Y = U\Sigma V^\top$ as $A = U\mathcal{P}_{\mu,\tau}(\Sigma)V^\top$ and $C = V\mathcal{P}_{\tau}(\mathcal{P}_{\mu,\tau}(\Sigma))V^\top$. Conversely, when A and C are fixed, the optimization problem over E reduces to

$$\min_{E} \frac{\mu}{2}\|D - A - E + \mu^{-1}Y\|_F^2 + \gamma\|E\|_1. \quad (75)$$

As discussed in Section 2.1, the optimal solution for E is given as $E = \mathcal{S}_{\gamma/\mu}(D - A + \mu^{-1}Y)$.

Given A and E, the ADMM algorithm updates Y using gradient ascent with step size μ, which gives $Y \leftarrow Y + \mu(D - A - E)$. Therefore, starting from $A_0 = D$, $E_0 = 0$ and $Y_0 = 0$, we obtain the following ADMM for solving the low-rank subspace clustering problem in the presence of gross corruptions,

$$\begin{aligned}(U_k, \Sigma_k, V_k) &= \text{svd}(D - E_k + \mu_k^{-1}Y_k)\\ A_{k+1} &= U_k\mathcal{P}_{\mu_k,\tau}(\Sigma_k)V_k^\top\\ E_{k+1} &= \mathcal{S}_{\gamma\mu_k^{-1}}(D - A_{k+1} + \mu_k^{-1}Y_k)\\ Y_{k+1} &= Y_k + \mu_k(D - A_{k+1} - E_{k+1})\\ \mu_{k+1} &= \rho\mu_k,\end{aligned} \quad (76)$$

where $\rho > 1$ is a parameter. As in the case of the IPT method, C is obtained from A upon convergence. Experimentally, we have observed that our method always converges. However, while the convergence of the ADMM is well studied for convex problems, we are not aware of any extensions to the nonconvex case.

5.2. Corrupted Data and Exact Constraints

Let us now consider the subspace estimation and clustering problem P_1, where the constraint $A = AC$ is enforced. We can solve P_1 as the limiting case of P_2 when $\tau \to \infty$. In this case, the polynomial thresholding operator $\mathcal{P}_{\alpha,\tau}$ becomes the hard thresholding operator $\mathcal{H}_{\sqrt{\frac{2}{\alpha}}}$. Therefore, we can solve P_1 using the IST and ADMM algorithms described in Section 5.1 with $\mathcal{P}_{\alpha,\tau}$ replaced by $\mathcal{H}_{\sqrt{\frac{2}{\alpha}}}$.

6. Experiments

In this section we evaluate the performance of LRSC on two computer vision tasks: motion segmentation and face clustering. Using the *subspace clustering error*,

$$\text{subspace clustering error} = \frac{\text{\# of misclassified points}}{\text{total \# of points}}, \quad (77)$$

as a measure of performance, we compare LRSC to state-of-the-art subspace clustering algorithms based on spectral clustering, such as LSA (Yan and Pollefeys, 2006), SCC (Chen and Lerman, 2009), LRR (Liu et al., 2010), and SSC (Elhamifar and Vidal, 2013). We choose these methods as a baseline, because they have been shown to perform very well on the above tasks, as reported in Vidal (2011). For the state-of-the-art algorithms, we use the implementations provided by their authors. Following the experimental setup in Elhamifar and Vidal (2013), the parameters of the different methods are set as shown in Table 2.

Table 2: Parameter setup of different algorithms. K is the number of nearest neighbors used by LSA to fit a local subspace around each data point, d is the dimension of each subspace assumed by LSA and SCC, τ is a parameter weighting the self-expressiveness error, α is a parameter weighting noise, and γ is a parameter weighting gross errors by the ℓ_1 (SSC, LRSC) or $\ell_{2,1}$ (LRR) norms.

Parameter	LSA	SCC	LRR	SSC	LRSC
Motion Segmentation					
K	8				
d	4	3			
τ					420
α				$\frac{800}{\min_i \max_{j\neq i} \lvert \boldsymbol{d}_i^\top \boldsymbol{d}_j\rvert}$	∞, 3000, 5000
γ			4	∞	∞, 5
ρ				1.0	1.1
Face Clustering					
K	7				
d	5	9			
τ					0.045, 0.075
α				∞	∞
γ			0.18	$\frac{20}{\min_i \max_{j\neq i} \|\boldsymbol{d}_j\|_1}$	∞, 11
ρ				1.0	1.1

Notice that the SSC and LRR algorithms in Elhamifar and Vidal (2013) and Liu et al. (2010), respectively, apply spectral clustering to a similarity graph built from the solution of their proposed optimization programs. Specifically, SSC uses the affinity $|C| + |C|^\top$, while LRR uses the affinity $|C|$. However, the implementation of the SSC algorithm normalizes the columns of C to be of unit ℓ_1 norm. To investigate the effect of this post-processing step, we report the results for both cases of without (SCC) and with (SCC-N) the column normalization step. Also, the code of the LRR algorithm in Liu et al. (2012) applies a heuristic post-processing step to the low-rank solution prior to building the similarity graph, similar to Lauer and Schnörr (2009). Thus, we report the results for both without (LRR) and with (LRR-H) the heuristic post-processing step.

Notice also that the original published code of LRR contains the function "compacc.m" for computing the misclassification rate, which is erroneous, as noted in Elhamifar and Vidal (2013). Here, we use the correct code for computing the

Figure 5: Motion segmentation: given feature points on multiple rigidly moving objects tracked in multiple frames of a video (top), the goal is to separate the feature trajectories according to the moving objects (bottom).

misclassification rate and as a result, the reported performance for LRR-H is different from the published results in Liu et al. (2010) and Liu et al. (2012). Likewise, our results for LRSC are different from those in our prior work Favaro et al. (2011), where we had also used the erroneous function "compacc.m".

Finally, since LSA and SCC need to know the number of subspaces a priori and the estimation of the number of subspaces from the eigenspectrum of the graph Laplacian in the noisy setting is often unreliable, to have a fair comparison, we provide the number of subspaces as an input to all the algorithms.

6.1. Experiments on Motion Segmentation

Motion segmentation refers to the problem of clustering a set of 2D point trajectories extracted from a video sequence into groups corresponding to different rigid-body motions. Here, the data matrix D is of dimension $2F \times N$, where N is the number of 2D trajectories and F is the number of frames in the video. Under the affine projection model, the 2D trajectories associated with a single rigid-body motion live in an affine subspace of $\mathbb{R}^{2F}$ of dimension $d = 1, 2$ or 3 (Tomasi and Kanade, 1992). Therefore, the trajectories associated with n different moving objects lie in a union of n affine subspaces in $\mathbb{R}^{2F}$, and the motion segmentation problem reduces to clustering a collection of point trajectories according to multiple affine subspaces. Since LRSC is designed to cluster linear subspaces, we apply LRSC to the trajectories in homogeneous coordinates, i.e., we append a constant $\lambda = 0.1$ and work with $2F + 1$ dimensional vectors.

We use the Hopkins155 motion segmentation database (Tron and Vidal, 2007) to evaluate the performance of LRSC against that of other algorithms. The database, which is available online at `http://www.vision.jhu.edu/data/hopkins155`, consists of 155 sequences of two and three motions. For each sequence, the 2D trajectories are extracted automatically with a tracker and outliers are manually removed. Figure 5 shows some sample images with the feature points superimposed.

Tables 3 and 4 give the average subspace clustering error obtained by different variants of LRSC on the Hopkins 155 motion segmentation database. We can see that most variants of LRSC have a similar performance. This is expected, because the trajectories are corrupted by noise, but do not have gross errors. Therefore, the Frobenius norm on the errors performs almost as well as the ℓ_1 norm. However, the performance depends on

Table 3: Clustering error (%) of different variants of the LRSC algorithm on the Hopkins 155 database with the $2F$-dimensional data points. The parameters in the first four columns are set as $\tau = 420$, $\alpha = 3000$ for 2 motions, $\alpha = 5000$ for 3 motions and $\gamma = 5$. The parameters in the last four columns are set as $\tau = \frac{4.5\times10^4}{\sqrt{MN}}$ and $\alpha = 3000$ for two motions, $\tau = \frac{6\times10^4}{\sqrt{MN}}$ and $\alpha = 5000$ for 3 motions, and $\gamma = 5$. For P_2-ADMM, we also set $\mu_0 = 100$ and $\rho = 1.1$.

Method	P_3	P_5	P_2 ADMM	P_2 IPT	P_3	P_5	P_2 ADMM	P_2 IPT
2 Motions								
Mean	3.39	3.27	3.13	3.27	2.58	**2.57**	2.62	**2.57**
Median	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
3 Motions								
Mean	7.28	7.29	7.31	7.29	6.68	**6.64**	6.76	6.67
Median	2.53	2.53	2.53	2.53	**1.76**	**1.76**	**1.76**	**1.76**
All								
Mean	4.25	4.16	4.05	4.16	3.49	**3.47**	3.53	3.48
Median	**0.00**	0.19	**0.00**	0.19	0.09	0.09	**0.00**	0.09

Table 4: Clustering error (%) of different variants of the LRSC algorithm on the Hopkins 155 database with the data projected onto a $4n$-dimensional space using PCA. The parameters for LRSC are chosen as in Table 3.

Method	P_3	P_5	P_2 ADMM	P_2 IPT	P_3	P_5	P_2 ADMM	P_2 IPT
2 Motions								
Mean	3.19	3.28	3.93	3.28	2.59	**2.57**	3.43	**2.57**
Median	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
3 Motions								
Mean	7.27	7.28	8.07	7.28	6.64	6.67	8.13	**6.62**
Median	2.54	2.54	3.76	2.54	**1.76**	**1.76**	2.30	**1.76**
All								
Mean	4.09	4.16	4.85	4.17	3.49	3.48	4.48	**3.47**
Median	**0.19**	0.19	**0.21**	0.19	0.19	0.09	0.19	**0.00**

Table 5: Clustering error (%) of different algorithms on the Hopkins 155 database with the $2F$-dimensional data points.

Method	LSA	SCC	LRR	LRR-H	SSC	SSC-N	LRSC
2 Motions							
Mean	4.23	2.89	4.10	2.13	2.07	**1.52**	2.57
Median	0.56	**0.00**	0.22	**0.00**	**0.00**	**0.00**	**0.00**
3 Motions							
Mean	7.02	8.25	9.89	**4.03**	5.27	4.40	6.64
Median	1.45	**0.24**	6.22	1.43	0.40	0.56	1.76
All							
Mean	4.86	4.10	5.41	2.56	2.79	**2.18**	3.47
Median	0.89	**0.00**	0.53	**0.00**	**0.00**	**0.00**	0.09

Table 6: Clustering error (%) of different algorithms on the Hopkins 155 database with the $4n$-dimensional data points obtained by applying PCA.

Method	LSA	SCC	LRR	LRR-H	SSC	SSC-N	LRSC
2 Motions							
Mean	3.61	3.04	4.83	3.41	2.14	**1.83**	2.57
Median	0.51	**0.00**	0.26	0.00	**0.00**	**0.00**	**0.00**
3 Motions							
Mean	7.65	7.91	9.89	4.86	5.29	**4.40**	6.62
Median	1.27	1.14	6.22	1.47	0.40	**0.56**	1.76
All							
Mean	4.52	4.14	5.98	3.74	2.85	**2.41**	3.47
Median	0.57	**0.00**	0.59	0.00	**0.00**	**0.00**	**0.00**

the choice of the parameters. In particular, notice that choosing τ that depends on the number of motions and size of each sequence gives better results than using a fixed τ.

Tables 5 and 6 compare the best results of LRSC against the state-of-the-art results. Overall, LRSC compares favorably against LSA and SCC and LRR without post-processing of the affinity matrix. Relative to LRR with post-processing, LRSC performs worse when the data is not projected, and better when the data is projected. However, LRSC does not perform as well as either version of SSC (with or without post-processing).

Overall, we can see that even the simplest version of LRSC (P_3), whose solution can be computed in closed form, performs on par with state-of-the-art motion segmentation methods, which require solving a convex optimization problem.

6.2. Experiments on Face Clustering

Face clustering refers to the problem of clustering a set of face images from multiple individuals according to the identity of each individual. Here, the data matrix D is of dimension $P \times N$, where P is the number of pixels, and N is the number of images. For a Lambertian object, the set of all images taken under all lighting conditions, but the same viewpoint and expression, forms a cone in the image space, which can be well approximated by a low-dimensional subspace (Basri and Jacobs, 2003). In practice, a few pixels deviate from the Lambertian model due to cast shadows and specularities, which can be modeled as sparse outlying entries. Therefore, the face clustering problem reduces to clustering a set of images according to multiple subspaces and corrupted by sparse gross errors.

We use the Extended Yale B database (Lee et al., 2005) to evaluate the performance of LRSC against that of state-of-the-art methods. The database includes 64 frontal face images of 38 individuals acquired under 64 different lighting conditions. Each image is cropped to 192×168 pixels. Figure 6 shows sample images from the database. To reduce the computational cost and the memory requirements of all algorithms, we downsample the images to 48×42 pixels and treat each $2,016$-dimensional vectorized image as a data point.

Following the experimental setup of Elhamifar and Vidal (2013), we divide the 38 subjects into 4 groups, where the first three groups correspond to subjects 1 to 10, 11 to 20, 21 to 30, and the fourth group corresponds to subjects 31 to 38. For each of the first three groups we consider all choices of $n \in \{2, 3, 5, 8, 10\}$ subjects and for the last group we consider all choices of $n \in \{2, 3, 5, 8\}$. Finally, we apply clustering algorithms for each trial, i.e., each set of n subjects.

Table 7 shows the average and median subspace clustering errors of different algorithms. In this experiment, we first apply the Robust Principal Component Analysis (RPCA) algorithm of Candès et al. (2011) to the face images of each subject and then apply different subspace clustering algorithms to the low-rank component of the data obtained by RPCA. While this cannot be done in practice, because the clustering of the data is not known beforehand, this experiment illustrates some of the challenges of the face clustering and validates several conclusions about the performances of different algorithms. In particular, notice that LSA and SCC do not perform well, even with de-corrupted data. Notice also that LRR-H does not perform well for more than 8 subjects, showing that the post processing step on the obtained low-rank coefficient matrix not always improves the result of LRR. SSC and LRSC, on the other hand, perform very well, with LRSC achieving perfect performance.

Table 7: Clustering error (%) of different algorithms on the Extended Yale B database after applying RPCA separately to the data points in each subject.

Algorithm	LSA	SCC	LRR	LRR-H	SSC-N	LRSC
2 Subjects						
Mean	6.15	1.29	0.09	0.05	0.06	**0.00**
Median	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
3 Subjects						
Mean	11.67	19.33	0.12	0.10	0.08	**0.00**
Median	2.60	8.59	**0.00**	**0.00**	**0.00**	**0.00**
5 Subjects						
Mean	21.08	47.53	0.16	0.15	0.07	**0.00**
Median	19.21	47.19	**0.00**	**0.00**	**0.00**	**0.00**
8 Subjects						
Mean	30.04	64.20	4.50	11.57	0.06	**0.00**
Median	29.00	63.77	0.20	15.43	**0.00**	**0.00**
10 Subjects						
Mean	35.31	63.80	0.15	13.02	0.89	**0.00**
Median	30.16	64.84	**0.00**	13.13	0.31	**0.00**

Table 8: Clustering error (%) of different algorithms on the Extended Yale B database without pre-processing the data.

Algorithm	LSA	SCC	LRR	LRR-H	SSC-N	LRSC
2 Subjects						
Mean	32.80	16.62	9.52	2.54	**1.86**	5.32
Median	47.66	7.82	5.47	0.78	**0.00**	4.69
3 Subjects						
Mean	52.29	38.16	19.52	4.21	**3.10**	8.47
Median	50.00	39.06	14.58	2.60	**1.04**	7.81
5 Subjects						
Mean	58.02	58.90	34.16	6.90	**4.31**	12.24
Median	56.87	59.38	35.00	5.63	**2.50**	11.25
8 Subjects						
Mean	59.19	66.11	41.19	14.34	**5.85**	23.72
Median	58.59	64.65	43.75	10.06	**4.49**	28.03
10 Subjects						
Mean	60.42	73.02	38.85	22.92	**10.94**	30.36
Median	57.50	75.78	41.09	23.59	**5.63**	28.75

Table 8 shows the results of applying different clustering algorithms to the original data, without first applying RPCA to each group. Notice that the performance of LSA and SCC deteriorates dramatically, showing that these methods are very sensitive to gross errors. The performance of LRR is better, but the errors are still very high, especially as the number of subjects increases. In this case, the post processing step of LRR-H does help to significantly reduce the clustering error.

Finally, Figure 7 shows the average computational time of each algorithm as a function of the number of subjects (or equivalently the number of data points). Note that the computational time of SCC is drastically higher than other algorithms. This comes from the fact that the complexity of SCC increases exponentially in the dimension of the subspaces, which in this

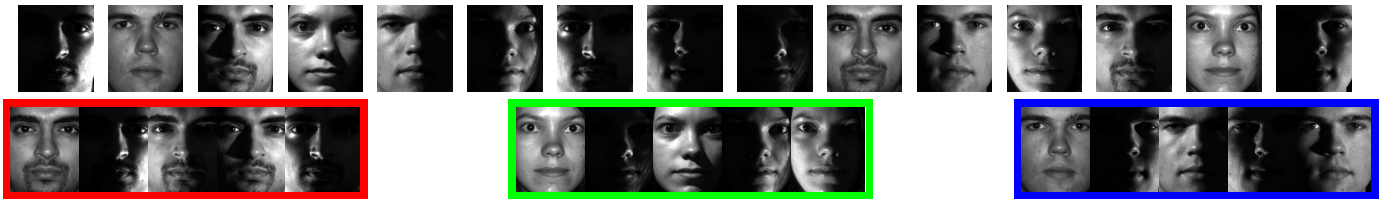

Figure 6: Face clustering: given face images of multiple subjects (top), the goal is to find images that belong to the same subject (bottom).

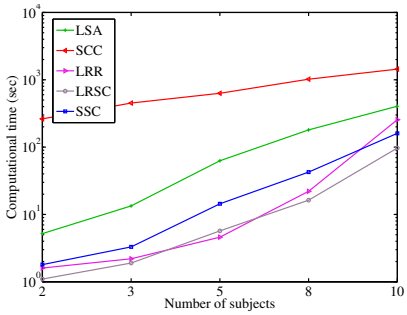

Figure 7: Average computational time (sec.) of the algorithms on the Extended Yale B database as a function of the number of subjects.

case is $d = 9$. On the other hand, SSC, LRR and LRSC use fast and efficient convex optimization techniques which keeps their computational time lower than other algorithms. Overall, LRR and LRSC are the fastest methods.

7. Discussion and Conclusion

We have proposed a new algorithm for clustering data drawn from a union of subspaces and corrupted by noise/gross errors. Our approach was based on solving a non-convex optimization problem whose solution provides an affinity matrix for spectral clustering. Our key contribution was to show that important particular cases of our formulation can be solved in closed form by applying a polynomial thresholding operator to the SVD of the data. A drawback of our approach to be addressed in the future is the need to tune the parameters of our cost function. Further research is also needed to understand the correctness of the resulting affinity matrix in the presence of noise and corruptions. Finally, all existing methods decouple the learning of the affinity from the segmentation of the data. Further research is needed to integrate these two steps into a single objective.

References

Agarwal, P., Mustafa, N., 2004. k-means projective clustering. In: ACM Symposium on Principles of database systems.

Basri, R., Jacobs, D., 2003. Lambertian reflection and linear subspaces. IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (3), 218–233.

Boult, T., Brown, L., 1991. Factorization-based segmentation of motions. In: IEEE Workshop on Motion Understanding. pp. 179–186.

Bradley, P. S., Mangasarian, O. L., 2000. k-plane clustering. Journal of Global Optimization 16 (1), 23–32.

Cai, J.-F., Candés, E. J., Shen, Z., 2008. A singular value thresholding algorithm for matrix completion. SIAM Journal of Optimization 20 (4), 1956–1982.

Candès, E., Li, X., Ma, Y., Wright, J., 2011. Robust principal component analysis? Journal of the ACM 58 (3).

Chen, G., Lerman, G., 2009. Spectral curvature clustering (SCC). International Journal of Computer Vision 81 (3), 317–330.

Costeira, J., Kanade, T., 1998. A multibody factorization method for independently moving objects. International Journal of Computer Vision 29 (3), 159–179.

Elhamifar, E., Vidal, R., 2009. Sparse subspace clustering. In: IEEE Conference on Computer Vision and Pattern Recognition.

Elhamifar, E., Vidal, R., 2010. Clustering disjoint subspaces via sparse representation. In: IEEE International Conference on Acoustics, Speech, and Signal Processing.

Elhamifar, E., Vidal, R., 2013. Sparse subspace clustering: Algorithm, theory, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence.

Favaro, P., Vidal, R., Ravichandran, A., 2011. A closed form solution to robust subspace estimation and clustering. In: IEEE Conference on Computer Vision and Pattern Recognition.

Gear, C. W., 1998. Multibody grouping from motion images. Int. Journal of Computer Vision 29 (2), 133–150.

Goh, A., Vidal, R., 2007. Segmenting motions of different types by unsupervised manifold clustering. In: IEEE Conference on Computer Vision and Pattern Recognition.

Gruber, A., Weiss, Y., 2004. Multibody factorization with uncertainty and missing data using the EM algorithm. In: IEEE Conf. on Computer Vision and Pattern Recognition. Vol. I. pp. 707–714.

Ho, J., Yang, M. H., Lim, J., Lee, K., Kriegman, D., 2003. Clustering appearances of objects under varying illumination conditions. In: IEEE Conference on Computer Vision and Pattern Recognition.

Hong, W., Wright, J., Huang, K., Ma, Y., 2006. Multi-scale hybrid linear models for lossy image representation. IEEE Trans. on Image Processing 15 (12), 3655–3671.

Lauer, F., Schnörr, C., 2009. Spectral clustering of linear subspaces for motion segmentation. In: IEEE International Conference on Computer Vision.

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

Lin, Z., Chen, M., Wu, L., Ma, Y., 2011. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. arXiv:1009.5055v2.

Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y., Ma, Y., 2011. Robust recovery of subspace structures by low-rank representation. In: http://arxiv.org/pdf/1010.2955v1.

Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y., Ma, Y., 2012. Robust recovery of subspace structures by low-rank representation. IEEE Transactions on Pattern Analysis and Machine Intelligence.

Liu, G., Lin, Z., Yu, Y., 2010. Robust subspace segmentation by low-rank representation. In: International Conference on Machine Learning.

Lu, L., Vidal, R., 2006. Combined central and subspace clustering on computer vision applications. In: International Conference on Machine Learning. pp. 593–600.

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

Mirsky, L., 1975. A trace inequality of John von Neumann. Monatshefte für Mathematic 79, 303–306.

Rao, S., Tron, R., Ma, Y., Vidal, R., 2008. Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In: IEEE Conference on Computer Vision and Pattern Recognition.

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

Recht, B., Fazel, M., Parrilo, P., 2010. Guaranteed minimum-rank solutions

of linear matrix equations via nuclear norm minimization. SIAM Review 52 (3), 471–501.

Soltanolkotabi, M., Elhamifar, E., Candes, E., 2013. Robust subspace clustering. http://arxiv.org/abs/1301.2603.

Sugaya, Y., Kanatani, K., 2004. Geometric structure of degeneracy for multi-body motion segmentation. In: Workshop on Statistical Methods in Video Processing.

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

Tomasi, C., Kanade, T., 1992. Shape and motion from image streams under orthography: A factorization method. International Journal of Computer Vision 9, 137–154.

Tron, R., Vidal, R., 2007. A benchmark for the comparison of 3-D motion segmentation algorithms. In: IEEE Conference on Computer Vision and Pattern Recognition.

Tseng, P., 2000. Nearest q-flat to m points. Journal of Optimization Theory and Applications 105 (1), 249–252.

Tseng, P., 2001. Convergence of a block coordinate descent method for nondifferentiable minimization. Journal of Optimization Theory and Applications 109 (3), 475–494.

Vidal, R., March 2011. Subspace clustering. IEEE Signal Processing Magazine 28 (3), 52–68.

Vidal, R., Ma, Y., Sastry, S., 2005. Generalized Principal Component Analysis (GPCA). IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (12), 1–15.

Vidal, R., Soatto, S., Ma, Y., Sastry, S., 2003. An algebraic geometric approach to the identification of a class of linear hybrid systems. In: Conference on Decision and Control. pp. 167–172.

Vidal, R., Tron, R., Hartley, R., August 2008. Multiframe motion segmentation with missing data using PowerFactorization and GPCA. International Journal of Computer Vision 79 (1), 85–105.

von Luxburg, U., 2007. A tutorial on spectral clustering. Statistics and Computing 17.

Yan, J., Pollefeys, M., 2006. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In: European Conf. on Computer Vision. pp. 94–106.

Yang, A., Wright, J., Ma, Y., Sastry, S., 2008. Unsupervised segmentation of natural images via lossy data compression. Computer Vision and Image Understanding 110 (2), 212–225.

Yang, A. Y., Rao, S., Ma, Y., 2006. Robust statistical estimation and segmentation of multiple subspaces. In: Workshop on 25 years of RANSAC.

Zhang, T., Szlam, A., Lerman, G., 2009. Median k-flats for hybrid linear modeling with many outliers. In: Workshop on Subspace Methods.

Zhang, T., Szlam, A., Wang, Y., Lerman, G., 2010. Hybrid linear modeling via local best-fit flats. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1927–1934.