Open access • Proceedings Article • DOI:10.1109/CVPR.2011.5995365

A closed form solution to robust subspace estimation and clustering — Source link

Paolo Favaro, René Vidal, Avinash Ravichandran

Institutions: Heriot-Watt University, Johns Hopkins University, University of California, Los Angeles

Published on: 20 Jun 2011 - Computer Vision and Pattern Recognition

Topics: Matrix decomposition, Sparse matrix, Singular value decomposition, Subspace topology and Linear subspace

A Closed Form Solution to Robust Subspace Estimation and Clustering

Paolo Favaro
Heriot-Watt University

René Vidal
Johns Hopkins University

Avinash Ravichandran
University of California, Los Angeles

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/outliers. We pose this problem as a rank minimization problem, where the goal is to decompose the corrupted data matrix as the sum of a clean, self-expressive, low-rank dictionary plus a matrix of noise/outliers. Our key contribution is to show that, for noisy data, this non-convex problem can be solved very efficiently and in closed form from the SVD of the noisy data matrix. Remarkably, this is true for both one or more subspaces. An important difference with respect to existing methods is that our framework results in a polynomial thresholding of the singular values with minimal shrinkage. Indeed, a particular case of our framework in the case of a single subspace leads to classical PCA, which requires no shrinkage. In the case of multiple subspaces, our framework provides an affinity matrix that can be used to cluster the data according to the subspaces. In the case of data corrupted by outliers, a closed-form solution appears elusive. We thus use an augmented Lagrangian optimization framework, which requires a combination of our proposed polynomial thresholding operator with the more traditional shrinkage-thresholding operator.

1. Introduction

Subspace estimation and clustering are very important problems with widespread applications in computer vision and pattern recognition. In computer vision, for example, the number of pixels in an image can be rather large, yet most computer vision models use only 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 a low-dimensional representation of a high-dimensional data set. Conventional techniques, such as Principal Component Analysis (PCA) [10], assume that the data are drawn from a *single* low-dimensional subspace of a high-dimensional space. In practice, however the data could be drawn from *multiple subspaces* and the *membership* of the data points to the subspaces might be unknown. Therefore, more recent techniques, such as Generalized Principal Component Analysis (GPCA) [20], seek to simultaneously cluster the data into multiple subspaces and find a low-dimensional subspace fitting each group of points.

One of the main challenges faced by existing subspace estimation and clustering algorithms (see [19] for a review of the state-of-the-art methods) is that the data are often contaminated by noise, missing entries and outliers. Traditionally, these issues have been addressed within a probabilistic framework, by using hidden variable models [9] and mixture models [17]. A major drawback of those approaches is that, with few exceptions, they result in non-convex optimization problems. As a consequence, good initialization becomes critical for those approaches to work.

Recently, there has been a surge of methods based on sparse representation theory and rank minimization [2, 3, 4, 6, 7, 13, 14, 16]. In principle, such methods seek to minimize a non-convex function as well, such as the the number of nonzero entries of a vector or matrix, or the rank of a matrix. However, it has been shown that, under certain conditions, such non-convex problems can be solved by minimizing a convex surrogate. For example, it is shown in [3, 6] that for most underdetermined systems of linear equations, the sparsest solution coincides with the minimal ℓ_1-norm solution. Likewise, it is shown in [16] that the problem of finding a low-rank approximation to a given matrix can be solved by minimizing the nuclear norm in lieu of its rank, under certain restricted isometry conditions generalized to matrices. Moreover, it is shown in [2] that the exact solution to the problem of decomposing a matrix into the sum of a low-rank matrix and a sparse matrix can be found by minimizing the sum of the nuclear norm and the ℓ_1 norm. Remarkably, this provides a convex solution to the robust PCA problem when either the rank of the matrix or the percentage of outliers are small enough. The optimization involves simple iterative shrinkage and thresholding of the singular values of one matrix and of the entries of another matrix.

Sparse representation and rank minimization techniques have also been applied to the subspace clustering problem. For instance, it is shown in [7] that a point in a union of multiple subspaces admits a sparse representation with respect to the dictionary formed by all other data points. It is also shown in [7] that, if the subspaces are independent, the nonzero coefficients in the sparse representation of a point correspond to other points in the same subspace. Moreover, the nonzero coefficients can be obtained by ℓ_1 minimization. These nonzero coefficients are then used to cluster the

data according to the multiple subspaces. A very similar approach is presented in [13]. 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 others has been successfully used when the data are corrupted by noise and outliers, from a theoretical viewpoint it is not clear why the above methods are effective when using a corrupted dictionary.

In this paper, we extend existing sparse representation and rank minimization techniques in several dimensions. First, we propose a general framework that is applicable to both subspace estimation and subspace clustering by using a non-convex formulation. Second, we show that important particular cases of our framework can be solved in closed form from the SVD of the data matrix, as opposed to using ℓ_1 minimization or iterative shrinkage and thresholding. Third, our framework results in a novel polynomial thresholding operator, which reduces the amount of shrinkage with respect to existing methods. Indeed, in the case of a single subspace, our framework reduces to classical PCA, which does not perform any shrinkage. Fourth, our framework does not make use of a corrupted dictionary. Instead, given the corrupted data, we search for both the clean dictionary and the coefficients of the sparse representation.

The remainder of the paper is organized as follows. Section 2 reviews existing results on sparse representation and rank minimization for subspace estimation and clustering. Section 3 formulates the subspace estimation and clustering problem in the presence of noise as a rank minimization problem. Section 4 extends our results to the case of outliers. In this case, the solution is found by minimizing a convex cost via an augmented Lagrange multipliers method. Section 5 presents experiments that evaluate our method on synthetic and real data. Section 6 gives the conclusions.

2. Background

2.1. Subspace Estimation by Sparse Representation and Rank Minimization

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

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

The optimal solution to this (PCA) problem is given by $A = U_1\Sigma_1 V_1^T$, where U_1, Σ_1 and V_1 are obtained from the top r singular values and singular vectors of the data matrix D. We may also write this as $A = U\mathcal{H}_{\sigma_{r+1}}(\Sigma)V^T$, where $\mathcal{H}_\epsilon(x) = x1_{|x|>\epsilon}$ is the hard thresholding operator.

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

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

where $\alpha > 0$ is a parameter. Since this problem is in general NP hard, a common practice (see [16]) 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

$$\min_{A} \quad \|A\|_* + \frac{\alpha}{2}\|D - A\|_F^2. \tag{3}$$

It is shown in [1] that the optimal solution to this problem is given by $A = U\mathcal{S}_{\frac{1}{\alpha}}(\Sigma)V^T$, where $D = U\Sigma V^T$ is the SVD of D and $\mathcal{S}_{\frac{1}{\alpha}}(\Sigma)$ is the shrinkage-thresholding operator

$$\mathcal{S}_\epsilon(x) = \begin{cases} x - \epsilon & x > \epsilon \\ x + \epsilon & x < -\epsilon \\ 0 & \text{else} \end{cases}. \tag{4}$$

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 [2] this issue is addressed by assuming that the outliers are sparse, *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.*,

$$\min_{A,E} \quad \operatorname{rank}(A) + \gamma\|E\|_0 \quad \text{s.t.} \quad D = A + E, \tag{5}$$

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 and the ℓ_0 semi-norm by the ℓ_1 norm. It is shown in [2] that, under broad conditions, the optimal solution to the problem in (5) is identical to that of the convex problem

$$\min_{A,E} \quad \|A\|_* + \gamma\|E\|_1 \quad \text{s.t.} \quad D = A + E. \tag{6}$$

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 [11] for a review of numerous approaches. One such approach is the Augmented Lagrange Multiplier (ALM) method, which minimizes

$$\|A\|_* + \gamma\|E\|_1 + \langle Y, D-A-E\rangle + \frac{\alpha}{2}\|D-A-E\|_F^2. \tag{7}$$

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. The inexact ALM method then iterates the following steps till convergence

$$(U, S, V) = \operatorname{svd}(D - E_k + \alpha_k^{-1}Y_k) \tag{8}$$

$$A_{k+1} = U\mathcal{S}_{\alpha_k^{-1}}(S)V^T \tag{9}$$

$$E_{k+1} = \mathcal{S}_{\gamma\alpha_k^{-1}}(D - A_{k+1} + \alpha_k^{-1}Y_k) \tag{10}$$

$$Y_{k+1} = Y_k + \alpha_k(D - A_{k+1} - E_{k+1}) \tag{11}$$

$$\alpha_{k+1} = \rho\alpha_k \tag{12}$$

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 $\rho > 1$ 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 representation and rank minimization for addressing this problem.

Sparse Subspace Clustering (SSC). The work of [7] shows that, in the absence of noise, the subspace clustering problem can be solved 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

$$\min_{C} \sum_{i} \|C_i\|_0 \text{ s.t. } D = DC \text{ and } \text{diag}(C) = 0, \quad (13)$$

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

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

It is shown in [7, 8] that when the subspaces are either independent or disjoint, the solution 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 $|C| + |C^T|$. The segmentation of the data is then obtained by applying spectral clustering [22] to this affinity.

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

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

In the case of outliers, following [15], it is assumed that the outliers are sparse. Since both C and E are sparse, the equation $D = DC + E = [D\ I][C^T\ E^T]^T$ 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. Therefore, one can find C and E by solving the following convex optimization problem

$$\min_{C,E} \|C\|_1 + \|E\|_1 \text{ s.t. } D = DC + E \text{ and } \text{diag}(C) = 0. \quad (16)$$

While SSC works well in practice, in the case of corrupted data there is no theoretical guarantee that the nonzero coefficients correspond to points in the same subspace. 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.

Low Rank Representation (LRR). This algorithm [13] 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 n independent subspaces of dimensions $r = \{d_i\}_{i=1}^n$, the rank of the data matrix is $\sum_{i=1}^n d_i$. With noise free data, the LRR algorithm solves the following convex optimization problem

$$\min_{C} \|C\|_* \quad \text{s.t.} \quad D = DC. \quad (17)$$

It can be shown that when the noise free data are drawn from independent linear subspaces, the optimal solution to (17) is given by the matrix $C = V_1 V_1^T$, where $D = U_1 \Sigma_1 V_1^T$ is the rank r SVD of D. As shown in [21], 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 outliers, the LRR algorithm solves the convex optimization problem

$$\min_{C} \|C\|_* + \alpha\|E\|_{2,1} \text{ s.t. } D = DC + E, \quad (18)$$

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 (16), except that the ℓ_1 and the Frobenious norms are replaced by the nuclear and the $\ell_{2,1}$ norms, respectively. It is argued in [13] that this allows one to better handle outliers in one data point but not in others, since it is a convex relaxation to the number of corrupted data points.

The LRR algorithm proceeds by solving the optimization problem in (18) using an ALM method. The optimal C is then used to define an affinity matrix $|C| + |C^T|$. The segmentation of the data is then obtained by applying spectral clustering to the normalized Laplacian.

3. A Closed-Form Solution to Subspace Estimation and Clustering with Noise

In this section, we propose a unified framework for both subspace estimation and clustering in the presence of noise. Specifically, we propose to solve the following problem

$$\min_{A,C,E} \|C\|_* + \frac{\alpha}{2}\|E\|_F^2 \text{ s.t. } A = AC \text{ and } D = A + E. \quad (19)$$

While in principle this problem appears to be very similar to those in (15) and (18), there are a number of key differences.

First, notice that instead of expressing the noisy data as a linear combination of itself plu noise, *i.e.*, $D = DC+E$, 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.*, $D = A + E$. As a consequence, our method searches simultaneously for a clean dictionary, the sparse coefficients and the noise. Second, the main difference with (15) is that the ℓ_1 norm of the matrix of coefficients is replaced by the nuclear norm. Third, the main difference with (18) is that the $\ell_{2,1}$ norm of the matrix of noise is replaced by the Frobenius norm.

As we will show in this section, the above changes 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 (18) requires an ALM method, the solution to (19) can be computed in closed form from the SVD of the data matrix D. The proof of this result will be done in three steps. In Lemma 1 we will relax the constraint $A = AC$ and add a penalty $\frac{\tau}{2}\|A - AC\|_F^2$ to the cost. We will then show that the optimal solution for C, with A kept fixed, can be obtained in closed form from the SVD of A. Since the optimal E is $D - A$, we will not consider the term $\frac{\alpha}{2}\|E\|_F^2$. Then, in Lemma 2 we will optimize the relaxed cost over both A and C and show that the optimal A can be obtained in closed form from the SVD of D by applying a polynomial thresholding to the singular values of D. Finally, in Lemma 3 we will show that the solution to (19) is given by classical PCA, except that the number of principal components can be automatically determined.

Lemma 1 *Let $A = U\Lambda V^T$ be the SVD of a given matrix A. The optimal solution to $\min_C \|C\|_* + \frac{\tau}{2}\|A - AC\|_F^2$ is*

$$\widehat{C} = V_1(I - \frac{1}{\tau}\Lambda_1^{-2})V_1^T, \tag{20}$$

where $U = [U_1\ U_2]$, $\Lambda = diag(\Lambda_1, \Lambda_2)$ 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

$$\phi(A) = \sum_{i\in\mathbf{I}_1}(1 - \frac{1}{2\tau}\lambda_i^{-2}) + \frac{\tau}{2}\sum_{i\in\mathbf{I}_2}\lambda_i^2. \tag{21}$$

Proof. In order for $\widehat{C}$ to be the minimizer, the first order sub-differential of the cost

$$\partial_C \text{Cost} = \partial\|C\|_* - \tau A^T A(I - C), \tag{22}$$

evaluated at $\widehat{C}$ should contain the zero matrix, *i.e.*, $0 \in \partial_{\widehat{C}}\text{Cost}$. We now show that $\widehat{C}$ satisfies this condition. For, recall that the sub-differential of the nuclear norm of a matrix C with compact SVD $C = U_C\Sigma_C V_C^T$ is given by

$$\partial\|C\|_* = \{U_C V_C^T + W : U_C^T W = 0, W V_C = 0, \|W\|_2 \leq 1\}. \tag{23}$$

Substituting this in (22) for $U_C = V_C = V_1$ yields

$$V_1V_1^T + W - \tau A^T A(I - C) = 0. \tag{24}$$

Since $I - \widehat{C} = I - V_1(I - \frac{1}{\tau}\Lambda_1^{-2})V_1^T = \frac{1}{\tau}V_1\Lambda_1^{-2}V_1^T + V_2V_2^T$ and $A^TA = V\Lambda^2V^T = V_1\Lambda_1^2V_1^T + V_2\Lambda_2^2V_2^T$, we obtain

$$V_1V_1^T + W - \tau(\frac{1}{\tau}V_1V_1^T + V_2\Lambda_2^2V_2^T) = 0. \tag{25}$$

This gives us $W = \tau V_2\Lambda_2V_2^T$, which is such that $V_1^TW = WV_1 = 0$ and $\|W\|_F^2 \leq 1$. Finally, if we replace the optimal solution into the cost function, we get

$$\begin{aligned}\phi(A) &= \|\widehat{C}\|_* + \frac{\tau}{2}\|A - A\widehat{C}\|_F^2 \\ &= \|I - \frac{1}{\tau}\Lambda_1^{-1}\|_* + \frac{\tau}{2}\|\frac{1}{\tau}U_1\Lambda_1^{-1}V_1^T + U_2\Lambda_2V_2^T\|_F^2 \\ &= \sum_{i\in\mathbf{I}_1}(1 - \frac{1}{\tau}\lambda_i^{-2}) + \frac{\tau}{2}(\sum_{i\in\mathbf{I}_1}\frac{1}{\tau^2}\lambda_i^{-2} + \sum_{i\in\mathbf{I}_2}\lambda_i^2)\end{aligned} \tag{26}$$

from which we get the desired result. ∎

Lemma 2 *Let $D = U\Sigma V^T$ be the SVD of the data matrix D. The optimal solution to*

$$\min_{A,C}\|C\|_* + \frac{\tau}{2}\|A - AC\|_F^2 + \frac{\alpha}{2}\|D - A\|_F^2 \tag{27}$$

is given by

$$\widehat{A} = U\Lambda V^T \ \text{ and } \ \widehat{C} = V_1(I - \frac{1}{\tau}\Lambda_1^{-2})V_1^T, \tag{28}$$

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

$$\sigma = \psi(\lambda) = \begin{cases}\lambda + \frac{1}{\alpha\tau}\lambda^{-3} & \text{if } \lambda > 1/\sqrt{\tau} \\ \lambda + \frac{\tau}{\alpha}\lambda & \text{if } \lambda \leq 1/\sqrt{\tau}\end{cases}, \tag{29}$$

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

Proof. For $\widehat{A}$ to be the minimizer, the first derivative of the cost in (27) with respect to A should be equal to zero, *i.e.*,

$$\tau A(I - C)(I - C)^T - \alpha(D - A) = 0. \tag{30}$$

Let $A = [U_1\ U_2]\text{diag}(\Lambda_1, \Lambda_2)[V_1\ V_2]^T$ be the SVD of A partitioned according to $\mathbf{I}_1$ and $\mathbf{I}_2$. Notice that we do not know yet that the SVDs of A and D are related, *i.e.* we do not know that $U = [U_1\ U_2]$ and $V = [V_1\ V_2]$. However, we know from Lemma 1 that the optimal C can be obtained from the SVD of A as $\widehat{C} = V_1(I - \frac{1}{\tau}\Lambda_1^{-2})V_1^T$. Therefore,

$$\begin{aligned}A(I - \widehat{C})^2 &= (U_1\Lambda_1V_1^T + U_2\Lambda_2V_2^T)(\frac{1}{\tau}V_1\Lambda_1^{-2}V_1^T + V_2V_2^T)^2 \\ &= \frac{1}{\tau^2}U_1\Lambda_1^{-3}V_1^T + U_2\Lambda_2V_2^T\end{aligned} \tag{31}$$

We thus have

$$
\begin{aligned}
D &= \frac{\tau}{\alpha} A(I - \widehat{C})^2 + A \\
&= \frac{1}{\tau\alpha} U_1 \Lambda_1^{-3} V_1^T + \frac{\tau}{\alpha} U_2 \Lambda_2 V_2^T + U_1 \Lambda_1 V_1^T + U_2 \Lambda_2 V_2^T \\
&= \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Lambda_1 + \frac{1}{\tau\alpha}\Lambda_1^{-3} & 0 \\ 0 & \Lambda_2 + \frac{\tau}{\alpha}\Lambda_2 \end{bmatrix} \begin{bmatrix} V_1 & V_2 \end{bmatrix}^T. \quad (32)
\end{aligned}
$$

The last expression gives a valid SVD for D, modulo reordering of the singular values. Therefore, the optimal solution for A is $A = U\Lambda V^T$, where the entries of Λ are related to those of Σ by equation (29).

Notice that $\psi(\lambda)$ is a strictly increasing function of λ when $3\tau \leq \alpha$. Therefore, in this case there is a unique λ for each σ. Moreover, the singular values of Λ have the same order as those of Σ. When $3\tau > \alpha$, the solution for λ may not be unique. In particular, up to four different solutions could be obtained from the polynomial $\lambda^4 - \sigma\lambda^3 + \frac{1}{\alpha\tau} = 0$. Nonetheless, only one of the solutions corresponds to the global minimum. To find the best solution, notice from (26) that the cost function in (27) reduces to

$$
\begin{aligned}
&\sum_{i \in \mathbf{I}_1} (1 - \frac{1}{2\tau}\lambda_i^{-2}) + \frac{\tau}{2} \sum_{i \in \mathbf{I}_2} \lambda_i^2 + \frac{\alpha}{2} \|D - A\|_F^2 = \\
&\sum_{i \in \mathbf{I}_1} (1 - \frac{1}{2\tau}\lambda_i^{-2}) + \frac{\tau}{2} \sum_{i \in \mathbf{I}_2} \lambda_i^2 + \frac{\alpha}{2} \sum_i (\sigma_{\pi(i)} - \lambda_i)^2,
\end{aligned} \quad (33)
$$

where π is an unknown permutation that sorts the singular values of Σ according to those of Λ. It follows from the above equation that the best λ_i for each $\sigma_{\pi(i)}$ can be found as the one that minimizes the ith term of the above summation. Specifically, for each σ_i, we find one or more candidate solutions λ_{ik} by solving (29) and then choose the optimal λ associated with σ_i as λ_{ik^*}, where K^* is given by

$$
\arg\min_k \frac{\alpha}{2}(\sigma_i - \lambda_{ik})^2 + \begin{cases} 1 - \frac{1}{2\tau}\lambda_{ik}^{-2} & \lambda_{ik} > 1\sqrt{\tau} \\ \frac{\tau}{2}\lambda_{ik}^2 & \lambda_{ik} \leq 1\sqrt{\tau} \end{cases}. \quad (34)
$$

Notice that the above procedure can be carried out without knowing π, because we can simply find a λ for each σ and determine the order of the λ's, hence π, at the end. ∎

Lemma 2 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 practice, when the term $\frac{1}{\alpha\tau} \simeq 0$ (relative to σ) there is a much simpler procedure to obtain the thresholding function. In this case, the quartic 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 hence

$$
\lambda = \sigma \qquad \forall \sigma : \sigma > 1/\sqrt{\tau}. \quad (35)
$$

Therefore, 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 is approximated (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 the cost (27) and compare them. We obtain

$$
\frac{\alpha\tau}{2(\alpha + \tau)}\sigma_*^2 = 1 - \frac{1}{2\tau\sigma_*^2}. \quad (36)
$$

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

$$
\sigma_* = \sqrt{\frac{\alpha + \tau}{\alpha\tau} + \sqrt{\frac{\alpha + \tau}{\alpha^2\tau}}}. \quad (37)
$$

Finally, our thresholding function can be written as

$$
\lambda = \widetilde{\mathcal{P}}_{\alpha,\tau}(\sigma) = \begin{cases} \sigma_i & \text{if } \sigma > \sigma_* \\ \frac{\alpha}{\alpha+\tau}\sigma_i & \text{if } \sigma_i \leq \sigma_*. \end{cases} \quad (38)
$$

As τ increases, notice that the largest singular values of D are preserved, rather than shrank by the operator $\mathcal{S}_{\frac{1}{\alpha}}$ in (4). Notice also that the smallest singular values of D are shrank by scaling them down, as opposed to subtracting a threshold. Moreover, notice that in the limiting case, where $\tau \to \infty$, a hard thresholding function is obtained, where the threshold is given by $\sigma_* = \sqrt{\frac{2}{\alpha}}$. That is, A can be obtained from the SVD of $D = U\Sigma V^T$ as $\widehat{A} = U\mathcal{H}_{\sqrt{\frac{2}{\alpha}}}(\Sigma)V^T$, where $\mathcal{H}_\epsilon(x) = x1_{|x|>\epsilon}$ is the hard thresholding operator. An alternative derivation of this result is given next.

Lemma 3 *Let $D = U\Sigma V^T$ be the SVD of the data matrix D. The optimal solution to*

$$
\min_{A,C} \|C\|_* + \frac{\alpha}{2}\|D - A\|_F^2 \quad s.t. \quad A = AC \quad (39)
$$

is given by $\widehat{A} = U_1\Sigma_1 V_1^T$ and $\widehat{C} = V_1 V_1^T$, where Σ_1, U_1 and V_1 correspond to the the top $r = \arg\min_k k + \frac{\alpha}{2}\sum_{i>k}\sigma_k^2$ singular values and singular vectors of D, respectively.

Proof. Using the method of Lagrange multipliers, with Y as the Lagrange multiplier for $A = AC$, we need to optimize

$$
\|C\|_* + \frac{\alpha}{2}\|D - A\|_F^2 + < Y, A - AC > . \quad (40)
$$

Without loss of generality, let us parameterize A by its rank-r SVD, $A = U_1\Lambda_1 V_1^T$, where r is arbitrary. Also, let V_2 be a basis for the orthonormal to V_1, so that $I = V_1V_1^T + V_2V_2^T$. It is shown in [12] that the optimal solution for C given A is $C = V_1V_1^T$. The first order conditions are thus given by

$$
V_1V_1^T + W = A^TY = V_1\Lambda_1U_1^TY \quad (41)
$$

$$
Y(I - C^T) = \alpha(D - A) \Rightarrow D = A + \frac{1}{\alpha}YV_2V_2^T. \quad (42)
$$

From the first equation we obtain $V_2^TW = 0$. Since we also have $V_1^TW = 0$, we conclude that $W = 0$. This implies that $U_1^TY = \Lambda_1^{-1}V_1^T$ and so Y must be of the form $Y = U_1\Lambda_1^{-1}V_1^T + U_2B$ for some B. Substituting this into

the second equation yields $D = A + \frac{1}{\alpha} U_2 B V_2^T$ and so $\|D - A\|_F^2 = \alpha^{-2} \|B V_2\|_F^2$. This cost is minimized when BV_2 is a diagonal matrix, say Λ_2, which can be chosen to be nonnegative without loss of generality. This means that $D = [U_1\ U_2] \text{diag}(\Lambda_1, \Lambda_2/\alpha) [V_1\ V_2]^T$ is a valid SVD for D. Therefore, we can choose $U = [U_1\ U_2]$, $V = [V_1\ V_2]$, $\Lambda_1 = \Sigma_1$ and $\Lambda_2 = \alpha \Sigma_2$. Substituting this into the cost gives

$$\|V_1 V_1^T\|_* + \frac{\alpha}{2} \|D - A\|_F^2 = r + \frac{\alpha}{2} \|\Sigma_2\|_F^2 = r + \frac{\alpha}{2} \sum_{k>r} \sigma_k^2. \quad (43)$$

r is then chosen as the minimizer of this cost. Equivalently, one can threshold the singular values of D at $\sqrt{2/\alpha}$. ■

4. Iterative Subspace Estimation and Clustering Algorithms in the Presence of Outliers

In this section, we propose a unified framework for both subspace estimation and clustering in the presence of outliers. Specifically, we propose to solve the problem

$$\min_{A,C,E} \|C\|_* + \gamma \|E\|_1 \text{ s.t. } A = AC \text{ and } D = A + E, \quad (44)$$

where, following [15, 7, 2], the ℓ_1 penalty on the matrix of outliers is motivated by the assumption that the outliers are sparse. As in the case of noise, the major difference of this formulation with respect to (16) and (18) is that, rather than using a corrupted dictionary, we search simultaneously for a clean dictionary A, the sparse coefficients C and the outliers E. Also, notice that the ℓ_1 norm of the matrix of coefficients is replaced by the nuclear norm and that the $\ell_{2,1}$ norm of the matrix of errors is replaced by the ℓ_1 norm.

4.1. Iterative Thresholding Approach

As in Section 3, we begin by considering a relaxed version of the problem in (44) in which the constraints are added to the cost function as penalties, *i.e.*,

$$\min_{A,C,E} \|C\|_* + \gamma \|E\|_1 + \frac{\tau}{2} \|A - AC\|_F^2 + \frac{\alpha}{2} \|D - A - E\|_F^2. \quad (45)$$

Notice that the second and fourth terms do not depend on C. Moreover, the first and third term are the same as those considered in Lemma 1. Therefore, the optimal solution for C is given by $\hat{C} = V_1 (1 - \frac{1}{\tau} \Lambda_1^{-2}) V_1^T$, which is obtained as a function of the SVD of $A = U_1 \Lambda_1 V_1^T + U_2 \Lambda_2 V_2^T$. After replacing the optimal C into (45) we obtain

$$\min_{A,E} \phi(A) + \gamma \|E\|_1 + \frac{\alpha}{2} \|D - A - E\|_F^2, \quad (46)$$

where $\phi(A)$ is defined in (21). Notice that if A is given, the optimal solution for E satisfies $\gamma \text{sign}(E) - \alpha(D - A - E) = 0$. This equation can be solved in closed form by using the shrinkage-thresholding operator

$$\widehat{E} = \mathcal{S}_{\frac{\gamma}{\alpha}} (D - A). \quad (47)$$

Finally, the derivative of the cost with respect to A should be zero, *i.e.*, $\tau A (I - C)(I - C)^T - \alpha (D - A) = 0$. Following (32) we obtain

$$D - \widehat{E} = [U_1\ U_2] \begin{bmatrix} \Lambda_1 + \frac{1}{\tau\alpha} \Lambda_1^{-3} & 0 \\ 0 & \Lambda_2 + \frac{\tau}{\alpha} \Lambda_2 \end{bmatrix} [V_1\ V_2]^T. \quad (48)$$

Therefore, if $\hat{E}$ was known, we could use Lemma 2 to compute $\widehat{A}$ and $\widehat{C}$ from the SVD of $D - \widehat{E}$. Conversely, if $\widehat{A}$ was known, we could compute $\widehat{E}$ from (47). This leads to an iterative thresholding algorithm that, starting from $E_0 = 0$, alternates between applying polynomial thresholding to $D - E_k$ to obtain $A_{k+1} = \mathcal{P}_{\alpha,\tau}(D - E_k)$ and applying shrinkage-thresholding to $D - A_{k+1}$ to obtain $E_{k+1} = \mathcal{S}_{\gamma/\alpha}(D - A_{k+1})$. When $\tau \to \infty$ the polynomial thresholding operator $\mathcal{P}_{\alpha,\tau}$ is simply replaced by the hard thresholding operator $\mathcal{H}_{\sqrt{\frac{2}{\alpha}}}$. However, notice that, as argued in [11], such iterative procedures have slow convergence compared to the ALM method.

4.2. Augmented Lagrange Multiplier Approach

In this section, we propose an alternative solution to problem (44) that results in a small variant of [11]. We start by considering the augmented Lagrangian formulation

$$\|C\|_* + \frac{\alpha}{2} \|D - A - E\|_F^2 + < Y, D - A - E > + \gamma \|E\|_1 \quad (49)$$

subject to $A = AC$. If we compute the first order derivatives of the above cost with respect to C and A and use the constraint $A = AC$, Lemma 3 tells us that $A = U_1 \Lambda_1 V_1^T$ and $C = V_1 V_1^T$, where V_1 corresponds to the singular values of $D - E + \alpha^{-1} Y$ larger than $\sqrt{2/\alpha}$. Given A and C the solution for E is the usual shrinkage thresholding $E = \mathcal{S}_{\gamma/\alpha}(D - A + \alpha^{-1} Y)$. We thus obtain the following:

$$(U, S, V) = \text{svd}(D - E_k + \alpha_k^{-1} Y_k) \quad (50)$$

$$A_{k+1} = U \mathcal{H}_{\sqrt{\frac{2}{\alpha}}}(S) V^T \quad (51)$$

$$E_{k+1} = \mathcal{S}_{\gamma \alpha_k^{-1}}(D - A_{k+1} + \alpha_k^{-1} Y_k) \quad (52)$$

$$Y_{k+1} = Y_k + \alpha_k (D - A_{k+1} - E_{k+1}) \quad (53)$$

$$\alpha_{k+1} = \rho \alpha_k \quad (54)$$

This method is identical to the ALM method in [11], except that the shrinkage-thresholding operator $\mathcal{S}$ is replaced by the hard thresholding operator $\mathcal{H}$. A similar algorithm based on the polynomial operator $\mathcal{P}$ can be derived when we include the constraint $A = AC$ in the cost function.

5. Experiments

Subspace clustering with noise. We apply our framework to the Hopkins155 motion segmentation database [18], which is available online at http://www.vision.jhu.edu/data/hopkins155. The database consists of 120 sequences of 2 motions and 35 sequences with 3 motions. For each sequence, point trajectories are extracted automatically

with a tracker and the outliers are manually removed. The task is to cluster the trajectories according to the different motions. Since the trajectories associated with each motion live in an affine subspace, the motion segmentation problem is equivalent to the problem of clustering affine subspaces. We extend our method in Lemma 1 to affine subspaces by enforcing the coefficients to add up to 1, *i.e.*, $\mathbf{1}^T C = \mathbf{1}^T$. The parameter τ is chosen as $\tau = 426$.

State-of-the-art motion segmentation algorithms use a number of pre-processing steps before subspace clustering. Notably, PCA is often used to project the trajectories onto a low-dimensional space. Since our method searches for a low-rank approximation of the data, it is not clear if applying PCA as a preprocessing step is beneficial to our framework. We thus compare our method without preprocessing.

We compare our method to several existing methods in the literature: Shape Interaction Matrix (SIM) [5], Local Subspace Affinity (LSA) [23], Sparse Subspace Clustering (SSC) [7], and the Low-Rank Representation (LRR) [13]. Table 1 shows the classification errors. We obtain an accuracy similar to that of the top-performing algorithms. However, our method is significantly faster (0.4 secs/sequence).

Subspace clustering with outliers. We apply our method in Sect. 4.2 on 12 sequences from [21], 9 with two motions and 3 with three motions, where 4%–35% of the point trajectories are corrupted with outliers. Table 2 compares our method against ℓ_1-based ALC [15] and SSC [7]. These results indicate the robustness of our method to outliers. In contrast to ALC, we do not need to use ℓ_1 minimization to correct the trajectories and then apply the segmentation algorithm. The resulting sparse coefficients are used directly to build the similarity graph and do the spectral clustering. As one can see, we also obtain very competitive results. Finally, as one can evince from the simplicity of our algorithm, the computational time is identical to that of [11].

6. Conclusions

We presented a general framework for subspace estimation and clustering in the presence of noise/outliers. Our key contribution was to show that, with noisy data, this problem can be solved in closed form. Our algorithm amounts to an SVD of the data matrix and a polynomial thresholding of its singular values. Our algorithm also gives a clean dictionary with respect to which the corrupted data can be expressed. For data corrupted by outliers, we proposed to minimize a non-convex cost via an augmented Lagrange multipliers method. We tested our algorithm on two motion segmentation databases. Our approach obtained an accuracy comparable to the state of the art, but significantly faster.

Acknowledgements. Work funded by ONR N00014-09-1-1067, ONR N00014-09-10084, ONR YIP N00014-09-10839, NSF 0931805 and Google Research Award 113095.

Table 1. Errors on Hopkins155 data without pre/post-processing.

Method	LSA	SIM	SSC	LRR	OUR
Average	8.99%	5.25%	3.89%	**3.16%**	3.28%

Table 2. Errors on 12 motion sequences with corrupted trajectories.

Method	$\ell^1 + \text{ALC}_5$	$\ell^1 + \text{ALC}_{\text{sp}}$	SSC	OUR
Average	4.15%	3.02%	**1.05%**	1.22%

References

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

[2] E. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis. *Journal of the ACM*, (Submitted) 2010.

[3] E. Candès and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

[4] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky. Rank-Sparsity Incoherence for Matrix Decomposition. *SIAM Journal on Optimization*, 2009.

[5] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *IJCV*, 29(3):159–179, 1998.

[6] D. L. Donoho. For most large underdetermined systems of linear equations the minimal ℓ^1-norm solution is also the sparsest solution. *Comm. on Pure and Applied Mathematics*, 59(6):797–829, 2006.

[7] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *IEEE CVPR*, 2009.

[8] E. Elhamifar and R. Vidal. Clustering disjoint subspaces via sparse representation. In *IEEE ICASSP*, 2010.

[9] A. Gruber and Y. Weiss. Multibody factorization with uncertainty and missing data using the EM algorithm. In *IEEE CVPR*, 2004.

[10] I. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.

[11] Z. Lin, M. Chen, L. Wu, and Y. Ma. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. *http://arxiv.org/abs/1009.5055*, 2009.

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

[13] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *ICML*, 2010.

[14] N. Ozay, M. Sznaier, C. Lagoa, and O. Camps. GPCA with denoising: A moments-based convex approach. In *IEEE CVPR*, 2010.

[15] S. Rao, R. Tron, R. Vidal, and Y. Ma. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *IEEE PAMI*, 2009.

[16] B. Recht, M. Fazel, and P. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

[17] M. Tipping and C. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 1999.

[18] R. Tron and R. Vidal. A benchmark for the comparison of 3-D motion segmentation algorithms. In *IEEE CVPR*, 2007.

[19] R. Vidal. Subspace clustering. *Signal Processing Magazine*, 28(2):52–68, 2011.

[20] R. Vidal, Y. Ma, and S. Sastry. Generalized Principal Component Analysis (GPCA). *IEEE PAMI*, 27(12):1–15, 2005.

[21] R. Vidal, R. Tron, and R. Hartley. Multiframe motion segmentation with missing data using PowerFactorization and GPCA. *IJCV*, 79(1):85–105, 2008.

[22] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17, 2007.

[23] J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *ECCV*, 2006.