Robust Subspace Segmentation by Low-Rank Representation

Guangcan Liu ROTH@SJTU.EDU.CN
Shanghai Jiao Tong University, P.R. China

Zhouchen Lin ZHOULIN@MICROSOFT.COM
Microsoft Research Asia

Yong Yu YYU@APEX.SJTU.EDU.CN
Shanghai Jiao Tong University, P.R. China

Abstract

We propose low-rank representation (LRR) to segment data drawn from a union of multiple linear (or affine) subspaces. Given a set of data vectors, LRR seeks the lowest-rank representation among all the candidates that represent all vectors as the linear combination of the bases in a dictionary. Unlike the well-known sparse representation (SR), which computes the sparsest representation of each data vector individually, LRR aims at finding the lowest-rank representation of a collection of vectors jointly. LRR better captures the global structure of data, giving a more effective tool for robust subspace segmentation from corrupted data. Both theoretical and experimental results show that LRR is a promising tool for subspace segmentation.

1. Introduction

In scientific data analysis and system engineering, one usually needs a parametric model to characterize a given set of data. To this end, linear models such as the *linear subspaces*[1] are possibly the most common choice, mainly because they are easy to compute and are also often effective in real applications. So the subspaces have been gaining much attention in recent years. For example, the hotly discussed matrix competition (Candès & Recht, 2009; Keshavan et al., 2009; Candès et al., 2009) problem is essentially based on the hypothesis that the data is drawn from a low-rank subspace. However, a given data set is seldom well described by a *single* subspace. A more reasonable model is to consider data as lying near several subspaces, leading to the challenging problem of *subspace segmentation*. Here, the goal is to segment (or cluster) data into clusters with each cluster corresponding to a subspace. Subspace segmentation is an important data clustering problem as it arises in numerous research areas, including machine learning (Lu & Vidal, 2006), computer vision (Ho et al., 2003), image processing (Fischler & Bolles, 1981) and system identification.

Previous Work. According to their mechanisms of representing the subspaces, existing works can be roughly divided into four main categories: mixture of Gaussian, factorization, algebraic and compressed sensing.

In statistical learning, mixed data are typically modeled as a set of independent samples drawn from a mixture of probabilistic distributions. As a single subspace can be well modeled by a Gaussian distribution, it is straightforward to assume that each probabilistic distribution is Gaussian, so known as the mixture of Gaussian model. Then the problem of segmenting the data is converted to a model estimation problem. The estimation can be performed either by using the Expectation Maximization (EM) algorithm to find a maximum likelihood estimate, as done in (Gruber & Weiss, 2004), or by iteratively finding a min-max estimate, as adopted by K-subspaces (Ho et al., 2003) and Random Sample Con-

[1] There is no loss of generality in assuming that the subspaces are linear, i.e., they all contain the origin. For the affine subspaces that do not contain the origin, we can always increase the dimension of the ambient space by one and identify each affine subspace with the linear subspace that it spans. So we always use the term "subspace" to denote "linear subspace" and "affine subspace" in this work.

Appearing in *Proceedings of the 26th International Conference on Machine Learning*, Haifa, Israel, 2010.

sensus (RANSAC) (Fischler & Bolles, 1981). These methods are sensitive to the noise and outliers. So several efforts have been made for improving their robustness, e.g., the Median K-flats (Zhang et al., 2009) for K-subspaces, the work of (Yang et al., 2006) for RANSAC, and (Ma et al., 2008a; Wright et al., 2008a) use a coding length to characterize a mixture of Gaussian, which may have some robustness. Nevertheless, the problem is still not well solved due to the optimization difficulty, which is a bottleneck for these methods to achieve robustness.

Factorization based methods (Costeira & Kanade, 1998; Gruber & Weiss, 2004) seek to represent the given data matrix as a product of two matrices, so that the support pattern of one of the factors reveals the grouping of the points. These methods aim at modifying popular factor analysis algorithms (often based on alternating minimization or EM-style algorithms) to produce such factorizations. Nevertheless, these methods are sensitive to noise and outliers, and it is not easy to modify them to be robust because they usually need iterative optimization algorithms to obtain the factorizations.

Generalized Principal Component Analysis (GPCA) (Ma et al., 2008b) presents an algebraic way to model the data drawn from a union of multiple subspaces. By describing a subspace containing a data point by using the gradient of a polynomial at that point, subspace segmentation is then equivalent to fitting the data with polynomials. GPCA can guarantee the success of the segmentation under certain conditions, and it does not impose any restriction on the subspaces. However, this method is sensitive to noise and outliers due to the difficulty of estimating the polynomials from real data, which also causes the high computation cost of GPCA. Recently, Robust Algebraic Segmentation (RAS)(Rao et al., 2009b) is proposed to resolve the robustness issue of GPCA. However, the computation difficulty for fitting polynomials is unfathomed. So RAS can make sense only when the data dimension is low and the number of subspaces is small.

Recently, the work of (Rao et al., 2009a) and Sparse Subspace Clustering (SSC) (Elhamifar & Vidal, 2009) introduced compressed sensing techniques to subspace segmentation. SSC uses the sparsest representation produced by ℓ_1-minimization (Wright et al., 2008b; Eldar & Mishali, 2008) to define the affinity matrix of an undirected graph. Then subspace segmentation is performed by spectral clustering algorithms such as the Normalized Cuts (NCut) (Shi & Malik, 2000). Under the assumption that the subspaces are independent, SSC shows that the sparsest representation is also "block-sparse". Namely, the within-cluster affinities are sparse (but nonzero) and the between-cluster affinities are all zeros. This implies that the graph is well defined and easy to segment. However, as SR finds the sparsest representation of each data vector *individually*, there is therefore no global constraint on its solution. So the method may be inaccurate at capturing the global structures of data. This drawback can largely depress the performance when the data is grossly corrupted. When no extra "clean" data are available, SR approaches may not be robust to noise and outliers in the subspace segmentation problem [2].

Our Contributions. In this work, we propose a novel method, called the *low-rank representation* (LRR), to address the subspace segmentation problem. As a compressed sensing technique, LRR also represents a data vector as a linear combination of the other vectors. Given a set of data vectors drawn from a union of multiple subspaces, unlike SR, LRR finds the *lowest-rank* representation of all data *jointly*. The lowest-rank representation can be used to define the affinities of an undirected graph, and then the final segmentation results can be obtained by spectral clustering. Comparing to SR, LRR is better at capturing global structures of data. So it better fits the subspace segmentation problem. When the subspaces are independent, we prove that there is a lowest-rank representation that reveals the membership of the samples: the within-cluster affinities are dense, and the between-cluster affinities are all zeros. We show that this solution solves a nuclear norm minimization problem, and give an efficient algorithm for solving this problem. For corrupted data, since the corruption will largely increase the rank as shown in matrix competition lectures (e.g., (Candès & Recht, 2009; Keshavan et al., 2009; Candès et al., 2009)), the lowest-rank criterion can enforce to correct corruptions and LRR could be robust to noise and outliers. In summary, the contributions of this work include:

- We introduce a simple and efficient new subspace segmentation algorithm that outperforms state-of-the-art algorithms in handling corrupted data.
- Our work extends the recovery of corrupted data from single subspace(Candès et al., 2009) to multiple subspaces. Comparing to (Eldar & Mishali, 2008), which requires the bases of subspaces to be known to handle the corrupted data from multiple subspaces, our method is *autonomous*, i.e., no extra clean data is required.

[2] SR can be robust to noise and outliers when extra clean data is available, e.g., a well defined dictionary as adopted by (Wright et al., 2008b).

2. Problem Formulation

More precisely, this paper addresses the following problem.

Problem 2.1 *Given a set of sufficiently dense data vectors $X = [x_1, x_2, \cdots, x_n]$ (each column is a sample) drawn from a union of k subspaces $\{\mathcal{S}_i\}_{i=1}^k$ of unknown dimensions, in a D-dimensional Euclidean space, segment all data vectors into their respective subspaces.*

We consider the problem under two assumptions of increasing practicality and difficulty.

Assumption 1. The subspaces are *low-rank* and *independent*[3], and the data is noiseless.

Assumption 2. A fraction of the data vectors are corrupted by noise or contaminated by outliers, or to be more precise, the data contains *sparse* and *properly bounded* errors.

The independence assumption is mild, because this is usually true especially when the subspaces are low-rank. What is critical is that the real data may not *strictly* follow subspace structures, i.e., there exist errors in data. In case of there is no extra clean data available, this problem is rather challenging. When the clean data is sufficient to represent the subspaces (i.e., the errors are sparse), and the errors do not "confuse" different subspaces (i.e., the errors are properly bounded), it will be shown that LRR is rather robust.

3. Subspace Segmentation via LRR

3.1. Low-Rank Representation

Consider a set of data vectors $X = [x_1, x_2, \cdots, x_n]$ (each column is a sample) in R^D, each of which can be represented by the linear combination of the basis in a "dictionary" $A = [a_1, a_2, \cdots, a_m]$:

$$X = AZ, \tag{1}$$

where $Z = [z_1, z_2, \cdots, z_n]$ is the coefficient matrix with each z_i being the *representation* of x_i. The dictionary is often overcomplete and hence there can be multiple feasible solutions to problem (1). It has been observed (e.g., by (Elhamifar & Vidal, 2009)) that sparse representations using an appropriate dictionaries A may reveal the clustering of the points x_i. However, as mentioned above, sparse representation may not capture the global structures of the data X. As we will see, *low rankness* may be a more appropriate criterion. That is, we look for a representation Z by solving the problem

$$\min_Z \ \mathrm{rank}(Z), \quad \text{s.t.}, \ X = AZ. \tag{2}$$

We call the optimal solutions Z^* of the above problem the "lowest-rank representations" of data X with respect to a dictionary A. The above optimization problem is difficult to solve due to the discrete nature of the rank function. Fortunately, as suggested by matrix completion methods (e.g., (Candès & Recht, 2009; Keshavan et al., 2009; Candès et al., 2009)) the following convex optimization provides a good surrogate for problem (2):

$$\min_Z ||Z||_*, \quad \text{s.t.}, \ X = AZ. \tag{3}$$

Here, $||\cdot||_*$ denotes the *nuclear norm* (Fazel, 2002) of a matrix, i.e., the sum of the singular values of the matrix.

3.2. The Basic Messages

Let $X = [x_1, x_2, \cdots, x_n]$ be a set of data vectors drawn from a union of k subspaces $\{\mathcal{S}_i\}_{i=1}^k$. Let $\{d_i\}_{i=1}^k$ be the unknown dimensions of the k subspaces, and X_i be collection of n_i data vectors drawn from the i-th subspace $\mathcal{S}_i$. Without loss of generality, we assume that $X = [X_1, X_2, \cdots, X_k]$ (i.e., the indices have been rearranged to satisfy the true segmentation of the data).

In order to segment the data into their respective subspaces, we need to compute an affinity matrix that encodes the pairwise affinities between data vectors. So we use the data X itself as the dictionary, i.e., problem (3) becomes

$$\min_Z ||Z||_*, \quad \text{s.t.}, \ X = XZ. \tag{4}$$

Note here that there always exist feasible solutions even when the data sampling is insufficient, because a data vector can be used to represent itself in LRR. To be more precise, since the solution space is $I - null(X)$, there always exist nontrivial solutions when $rank(X) < n$. This is different from SR, which is prone to obtain a "trivial" solution if a data vector is used to represent itself.

Theorem 3.1 *Assume that the data sampling is sufficient such that $n_i > rank(X_i) = d_i$. If the subspaces are independent then there exists an optimal solution*

[3] The subspaces are independent if and only if $\sum_{i=1}^k \mathcal{S}_i = \bigoplus_{i=1}^k \mathcal{S}_i$, where $\bigoplus$ is the direct sum.

Z^* to problem (4) that is block-diagonal:

$$Z^* = \begin{bmatrix} Z_1^* & 0 & 0 & 0 \\ 0 & Z_2^* & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & Z_k^* \end{bmatrix}_{n\times n},$$

with Z_i^* being an $n_i \times n_i$ matrix with $rank(Z_i^*) = d_i$, $\forall\, i$.

The proof is based on the following well-known lemma, whose proof is provided in the appendix for completeness:

Lemma 3.1 *Let A and D be square matrices. Then for any matrices B and C of compatible dimension,*

$$\left\| \begin{pmatrix} A & B \\ C & D \end{pmatrix} \right\|_* \geq \left\| \begin{pmatrix} A & 0 \\ 0 & D \end{pmatrix} \right\|_* = \|A\|_* + \|D\|_*.$$

The above lemma allows us to lower-bound the objective value at any solution Z by the value of the block-diagonal restriction of Z. This leads to a simple proof of Theorem 3.1:

Proof (of Theorem 3.1). Let Z be any optimizer to (4). Form a block-diagonal matrix W by setting

$$W_{ij} = \begin{cases} Z_{ij}, & x_i \text{ and } x_j \text{ belong to the same subspace,} \\ 0, & \text{otherwise.} \end{cases}$$

Write $Q = Z - W$. For any matrix M, let $[M]_j$ denote its j-th column. Let x_j belong to the l-th subspace; i.e., $[XZ]_j \in S_l$. Then by construction $[XW]_j \in S_l$, and $[XQ]_j \in \oplus_{i \neq l} S_i$. But $[XQ]_j = [XZ]_j - [XW]_j \in S_l$. By independence, $S_l \cap \oplus_{i\neq l} S_i = \{0\}$, and so $[XQ]_j = 0$.

Hence, $XQ = 0$, and W is feasible for (4). By Lemma 3.1, $\|Z\|_* \geq \|W\|_*$, and so W is optimal for (4). We can write W as

$$W = \begin{bmatrix} W_1 & 0 & 0 & 0 \\ 0 & W_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & W_k \end{bmatrix} \in \mathbb{R}^{n\times n},$$

where $W_i \in \mathbb{R}^{n_i \times n_i}$. For each i, let $\mathtt{P}_i \in \mathbb{R}^{n_i\times n_i}$ be the projection onto the null space of X_i. Then $X_i(\mathtt{I}-\mathtt{P}_i)W_i = X_iW_i = X_i$. So, if we set $Z_i^* = (\mathtt{I}-\mathtt{P}_i)W_i$, then

$$Z^* = \begin{bmatrix} Z_1^* & 0 & 0 & 0 \\ 0 & Z_2^* & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & Z_k^* \end{bmatrix} \in \mathbb{R}^{n\times n}$$

is again feasible for (4). Now, $\|Z^*\|_* = \sum_i \|Z_i^*\|_* = \sum_i \|(\mathtt{I}-\mathtt{P}_i)W_i\|_* \leq \sum_i \|W_i\|_* = \|W\|_*$, where we have used (e.g., (Horn & Johnson, 1991) Corollary 3.5.10) to show that $\|(\mathtt{I}-\mathtt{P}_i)W_i\|_* \leq \|(\mathtt{I}-\mathtt{P}_i)\|\|W_i\|_* \leq \|W_i\|_*$. Hence, Z^* is again optimal for (4). Moreover, for each i, $\mathrm{rank}(Z_i^*) \leq \mathrm{rank}(\mathtt{I}-\mathtt{P}_i) = d_i$. Since $X_i = X_iZ_i^*$, $\mathrm{rank}(Z_i^*) \geq d_i$, and so we conclude that $\mathrm{rank}(Z_i^*) = d_i$ for each i. ■

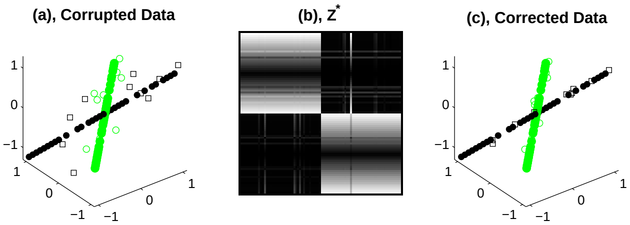

Figure 1. Illustrating that LRR can automatically correct the corruptions in data: (a) The data a part of which is corrupted. (b) The optimal solution Z^* of problem (5) with $\lambda = 0.05$ (c) The corrected data obtained by XZ^*.

Theorem 3.1 does not guarantee that an arbitrary optimal solution to problem (4) is block-diagonal. The difficulty is essentially that the minimizer is non-unique. In theory, this could be resolved, e.g., by choosing a minimizer to (4) of smallest Frobenius norm (which is again a convex problem). However, in our simulations we have observed that the solution obtained is always block-diagonal, and so we do not pursue this here.

3.3. Robustness to Noise and Outliers

In real applications, our observations are often noisy, or even grossly corrupted, and observations may be missing. For small noise (e.g., Gaussian) a reasonable strategy is simply to relax the equality constraint in (4), similar to (Candes & Plan, 2009). If we imagine instead that a fraction of the data vectors are grossly corrupted, a more reasonable objective might be

$$\min_{Z,E} ||Z||_* + \lambda ||E||_{2,1}, \quad \text{s.t., } X = XZ + E, \tag{5}$$

where $||E||_{2,1} = \sum_{j=1}^n \sqrt{\sum_{i=1}^n ([E]_{ij})^2}$ is called as the $\ell_{2,1}$-norm, and the parameter $\lambda > 0$ is used to balance the effects of the two parts, which could be chosen according to properties of the two norms, or tuned empirically. Since $\ell_{2,1}$-norm encourages the columns of E to be zero, the underlying assumption here is that the corruptions are "sample-specific", i.e., some data vectors are corrupted and the others are clean.

After obtaining an optimal solution (Z^*, E^*), we could recover the original data by using $X - E^*$ (or XZ^*). To illustrate how the corruptions are corrected, we refer to an example as shown in Fig.1. There are about 80 data vectors sampled from two one-dimensional subspaces embedded in R^3, and about 25% data vectors are corrupted by large Gaussian errors. The results in Fig.1 show that LRR can well handle the corruptions. One message from these results is that LRR is unlikely to cause "positive-false" to change the clean data. So the possibly existed corruptions could be cor-

Algorithm 1 Solving Problem (5) by Inexact ALM

```
Input: data matrix $X$, parameter $\lambda$
Initialize: $Z = J = 0, E = 0, Y_1 = 0, Y_2 = 0, \mu = 10^{-6}, max_u = 10^{10}, \rho = 1.1, \varepsilon = 10^{-8}$.
while not converged do
    1. fix the others and update $J$ by
        $J = \arg\min \frac{1}{\mu}||J||_* + \frac{1}{2}||J - (Z + Y_2/\mu)||_F^2$
    2. fix the others and update $Z$ by
        $Z = (I + X^t X)^{-1}(X^t X - X^t E + J + (X^t Y_1 - Y_2)/\mu)$
    3. fix the others and update $E$ by
        $E = \arg\min \frac{\lambda}{\mu}||E||_{2,1} + \frac{1}{2}||E - (X - XZ + Y_1/\mu)||_F^2$
    4. update the multipliers
        $Y_1 = Y_1 + \mu(X - XZ - E)$
        $Y_2 = Y_2 + \mu(Z - J)$
    5. update the parameter $\mu$ by $\mu = \min(\rho\mu, max_u)$
    6. check the convergence conditions
        $||X - XZ - E||_\infty < \varepsilon$ and $||Z - J||_\infty < \varepsilon$.
end while
```

rected in such a way: Let's rearrange the data vectors into $X = [X_l, X_c]$, where X_l is the clean data without corruptions and X_c is the corrupted data. In the case that the remainder clean data X_l is still sufficient to represent the subspaces, and the corruptions are properly bounded, it shall automatically correct the corruptions so as to obtain the lowest-rank representation.

3.4. Solving the Optimization Problem

Since the problem (5) can fall back to the problem (4) by setting the parameter λ to be relatively large, here we just present how to solve problem (5). We first convert it to the following equivalent problem:

$$\begin{aligned} \min_{Z,E,J} \; & ||J||_* + \lambda||E||_{2,1}, \\ \text{s.t.}, \; & X = XZ + E, \\ & Z = J, \end{aligned} \tag{6}$$

which can be solved by solving the following Augmented Lagrange Multiplier (ALM) problem:

$$\begin{aligned} \min_{Z,E,J,Y_1,Y_2} & ||J||_* + \lambda||E||_{2,1} + \\ \text{tr}\left[Y_1^t(X - XZ - E)\right] & + \text{tr}\left[Y_2^t(Z - J)\right] + \\ \frac{\mu}{2}(||X - XZ - E||_F^2 & + ||Z - J||_F^2), \end{aligned} \tag{7}$$

Algorithm 2 Subspace Segmentation by LRR

```
Input: data matrix $X$, number of subspaces $k$
1. obtain the lowest-rank representation by solving problem (5)
2. construct an undirected graph by using the lowest-rank representation to define the affinity matrix of the graph
3. use NCut to segment the vertices of the graph into $k$ clusters
```

where Y_1 and Y_2 are Lagrange multipliers and $\mu > 0$ is a penalty parameter. The above problem can by solved by either *exact* or *inexact* ALM algorithms (Lin et al., 2009). For efficiency, we choose the inexact ALM, which we outline in Algorithm 1. Its convergence properties could be proved similarly as those in (Lin et al., 2009). Notice that although steps 1 and 3 of the algorithm are convex problems, they both have closed-form solutions. Step 1 is solved via the singular value thresholding operator (Cai et al., 2008), while Step 3 is solved via the following lemma:

Lemma 3.2 *Let $Q = [q_1, q_2, \cdots, q_i, \cdots]$ be a given matrix and $||\cdot||_F$ be the Frobenius norm. If the optimal solution of*

$$\min_W \lambda||W||_{2,1} + \frac{1}{2}||W - Q||_F^2$$

is W^*, then the i-th column of W^* is

$$W^*(:, i) = \begin{cases} \frac{||q_i|| - \lambda}{||q_i||} q_i, & \textit{if } \lambda < ||q_i||, \\ 0, & \textit{otherwise.} \end{cases}$$

3.5. The Complete Segmentation Algorithm

After solving problem (5), as in (Elhamifar & Vidal, 2009), we utilize the lowest-rank representation (denoted by Z^*) to define the affinity matrix of an undirected graph. The data vectors correspond to the vertices and the affinity between x_i and x_j is computed by $|[Z^*]_{ij}| + |[Z^*]_{ji}|$. We then could use the spectral clustering algorithms such as Normalized Cuts (Shi & Malik, 2000) to produce the final segmentation results. Integrating LRR with spectral clustering has some advantages. First, since LRR may fail to obtain a block-diagonal representation in complex applications, the spectral clustering could ensure the robustness of the segmentation. Second, it is convenient to integrate the lowest-rank representation with other information by defining such an undirected graph. For example, in some specific applications such as image segmentation, ones may want to enforce that only the neighbor samples can be connected by edges. Algorithm 2 summarizes the whole segmentation algorithm of LRR.

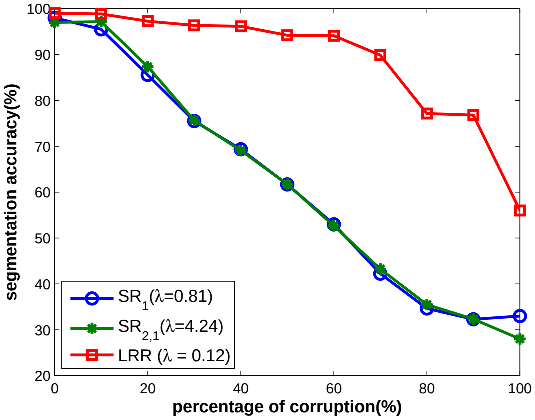

Figure 2. The segmentation accuracy (averaged from 20 random trials) across the entire range of corruption for various methods. LRR (red curve) clearly outperforms both of SR$_1$ and SR$_{2,1}$, performing nearly perfectly up to 60 percent corruptions. The parameter λ is tuned to be the best at 20% percentage of corruption for various methods.

Table 1. Segmentation errors (%) on Hopkins155. For LRR, the parameter is set as $\lambda = 2.4$. The parameters of the other methods have been also tuned to be the best.

	GPCA	LSA	RANSAC	SSC	LRR
Max	55.67	38.37	41.31	37.44	**32.50**
Mean	30.51	8.77	7.81	3.66	**3.13**
Std.	11.79	9.80	9.72	7.21	**5.96**

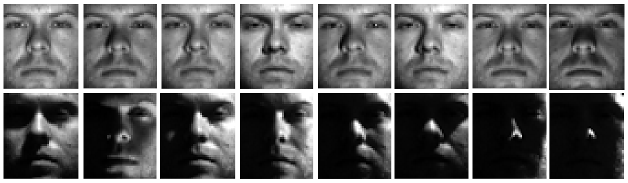

Figure 3. Some examples of the images of a class in Extended Yale Database B.

4. Experiments

4.1. Toy Data

For this experiment, we compare the robustness of LRR and SR under the context of subspace segmentation. For fair of comparison, we implement two versions of SR. The first one is the standard version used by most previous methods:

$$SR_1: \quad \min_{Z,E} ||Z||_1 + \lambda ||E||_1, \quad \text{s.t., } X = XZ + E, [Z]_{ii} = 0, \tag{8}$$

where $||Z||_1 = \sum_{i,j=1}^{n} |[Z]_{ij}|$ is called as the ℓ_1-norm. The second one is modified for better fitting the corruptions:

$$SR_{2,1}: \quad \min_{Z,E} ||Z||_1 + \lambda ||E||_{2,1}, \quad \text{s.t., } X = XZ + E, [Z]_{ii} = 0. \tag{9}$$

We construct 5 independent subspaces $\{\mathcal{S}_i\}_{i=1}^5 \subset R^{100}$ whose bases $\{U_i\}_{i=1}^5$ are computed by $U_{i+1} = TU_i, 1 \leq i \leq 4$, where T is a random rotation and U_1 is a random orthogonal matrix of dimension 100×4. So, each subspace has a dimension of 4. We sample 20 data vectors from each subspace by $X_i = U_i Q_i, 1 \leq i \leq 5$ with Q_i being a 4×20 iid $\mathcal{N}(0, 1)$ matrix. Some data vectors are randomly chosen to corrupt, e.g., for a data vector x chosen to corrupt, its observed vector is computed by adding Gaussian noise with zero mean and variance $0.3||x||$ ($||x||$ mostly ranges from 0.1 to 1.7 in this experiment).

After obtaining the coefficient matrix, we use the algorithm presented in Section 3.5 to segment the data into 5 clusters and observe the *segmentation accuracy* [4] of each method. Although the $\ell_{2,1}$-norm is better than ℓ_1-norm to fit the corruptions in this experiment, both of SR$_{2,1}$ and SR$_1$ are significantly outperformed by our LRR, as shown Fig.2. This illustrate the superiority of the lowest-rank criterion. As the corruption is unnecessary to decrease the sparsity, SR cannot handle well the corrupted data in unsupervised environment. Whereas, LRR is good at handling such corrupted data as analyzed in Section 3.3.

4.2. Slightly Corrupted Data

In this section, we evaluate LRR on the Hopkins155 motion database (Tron & Vidal, 2007). The database consists of 156 sequences each of which has 39∼550 data vectors drawn from two or three motions (a motion corresponds to a subspace). Each sequence is a sole clustering task and so there are 156 clustering tasks in total. As the data dimension must be bounded above by 12 (Elhamifar & Vidal, 2009), we use a preprocessing step to project the data to be 12-dimensional by PCA. The outliers in the data have been manually removed and the PCA can also reduce some noise. So it could be regarded that this database only contains slight corruptions.

[4] As clustering methods cannot predict the class label of each cluster, we use a postprocess step to assign each cluster a label: Given ground truth classification results, the label of a cluster is the index of the ground truth class that contributes the maximum number of samples to the cluster. And then we can obtain the segmentation accuracy by computing the percentage of correctly classified data vectors.

Table 2. Segmentation accuracy (%) on Extended Yale Database B. We have tuned the parameters of all methods to the best. The parameter of LRR is set to be $\lambda = 0.15$

	GPCA	LSA	RANSAC	SSC	LRR
Acc.	NA	31.72	NA	37.66	**62.53**

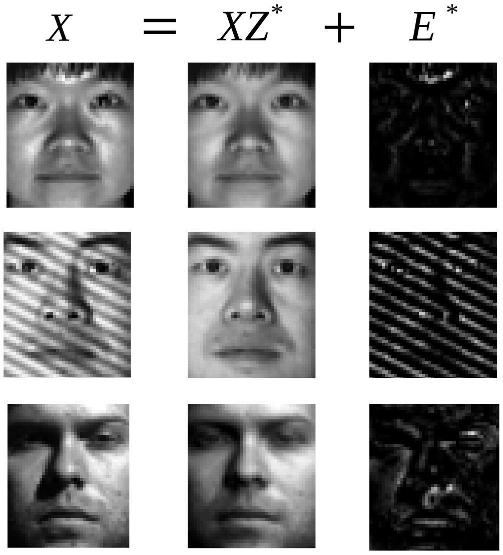

Figure 4. Some examples of using LRR to correct the corruptions in faces. Left: The original data (X); Middle: The corrected data (XZ^*); Right: The error (E^*).

In order to compare LRR with the state of the art, we also list the results of GPCA, Local Subspace Analysis (LSA) (Yan & Pollefeys, 2006), RANSAC and SSC[5]. We manually tune the parameters of each method and report their best results. Table 1 shows that LRR significantly outperforms GPCA, LSA and RANSAC, and also outperforms SSC. This confirms that the lowest-rank criterion is accurate for modeling the structures of subspaces.

4.3. Heavily Corrupted Data

We test LRR's ability to cope with the large corruptions in data, using a part of Extended Yale Database B (Lee et al., 2005), which consists of 640 frontal face images of 10 classes (there are 38 classes in the whole database and we use the first 10 classes for experiments). Each class contains about 64 images. We resize the images into 42×48 and use the raw pixel values to form data vectors of dimension 2016. As shown in Fig.3, more than half of the data vectors have been corrupted by "shadows" and noise. So the corruptions in this database is heavy.

Since the computation of GPCA and RANSAC is unbearable on this database, we only list the results of LSA, SSC and LRR. Table 2 shows that LRR distinctly outperforms the baselines. The advantages of LRR mainly comes from its ability of automatically correcting the corruptions in data, as shown in Fig.4.

5. Conclusion and Future Work

In this work we propose the low-rank representation (LRR) to recover the lowest-rank representation of a set of data vectors in a *joint* way, i.e., to recover the lowest-rank representation of matrix data. Comparing to the widely used SR, LRR is better at handling the global structures and correcting the corruptions in data automatically. Both theoretical and experimental results illustrate the effectiveness of LRR. However, there remain several directions for future work:

- It will be better to learn a compact dictionary for LRR, which is to recover the structure that generates the data.

- The subspace segmentation should not be the only application of LRR. For example, one may use LRR to do supervised classification in a similar way as (Wright et al., 2008b).

- LRR also gives a way to recover the corrupted data drawn from multiple subspaces. The theoretical conditions for the success of the recovery should be established.

Appendix

Proof (of Lemma 3.1). It is well-known (e.g., (Horn & Johnson, 1991) Theorem 3.4.1) that for any M,

$$\|M\|_* = \max_{\substack{U^tU=\mathrm{I}\\ V^tV=\mathrm{I}}} \mathrm{tr}\left[U^tMV\right].$$

Hence,

$$\left\|\begin{pmatrix} A & B \\ C & D \end{pmatrix}\right\|_* \geq$$

$$\max_{\substack{U_1^tU_1=\mathrm{I},\ U_2^tU_2=\mathrm{I}\\ V_1^tV_1=\mathrm{I},\ V_2^tV_2=\mathrm{I}}} \mathrm{tr}\left[\begin{pmatrix} U_1^t & 0 \\ 0 & U_2^t \end{pmatrix}\begin{pmatrix} A & B \\ C & D \end{pmatrix}\begin{pmatrix} V_1 & 0 \\ 0 & V_2 \end{pmatrix}\right]$$

$$= \max_{\substack{U_1^tU_1=\mathrm{I},\ U_2^tU_2=\mathrm{I}\\ V_1^tV_1=\mathrm{I},\ V_2^tV_2=\mathrm{I}}} \mathrm{tr}\left[U_1^tAV_1\right] + \mathrm{tr}\left[U_2^tDV_2\right]$$

$$= \max_{\substack{U_1^tU_1=\mathrm{I}\\ V_1^tV_1=\mathrm{I}}} \mathrm{tr}\left[U_1^tAV_1\right] + \max_{\substack{U_2^tU_2=\mathrm{I}\\ V_2^tV_2=\mathrm{I}}} \mathrm{tr}\left[U_2^tDV_2\right]$$

$$= \|A\|_* + \|D\|_*.$$

[5] The Matlab code of these baselines can be downloaded from `http://www.vision.jhu.edu/data/hopkins155/`. The Matlab code of our method is available at `www.apexlab.org/apex_wiki/gcliu.`

Acknowledgments

The authors thank John Wright for sharing his creative ideas, helping designing the experiments, and helping revising the paper. Yi Ma for giving us constructive instructions.

References

Cai, Jian-Feng, Candes, Emmanuel J., and Shen, Zuowei. A singular value thresholding algorithm for matrix completion. 2008.

Candes, Emmanuel J. and Plan, Yaniv. Matrix completion with noise. In *IEEE Proceeding*, 2009.

Candès, Emmanuel J. and Recht, Benjamin. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 2009.

Candès, Emmanuel J., Li, Xiaodong, Ma, Yi, and Wright, John. Robust principal component analysis?, 2009.

Costeira, Jo ao Paulo and Kanade, Takeo. A multibody factorization method for independently moving objects. *IJCV*, 1998.

Eldar, Yonina C. and Mishali, Moshe. Robust recovery of signals from a union of subspaces. *CoRR*, 2008.

Elhamifar, Ehsan and Vidal, René. Sparse subspace clustering. In *CVPR*, 2009.

Fazel, M. Matrix rank minimization with applications. *PhD thesis*, 2002.

Fischler, Martin A. and Bolles, Robert C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 1981.

Gruber, Amit and Weiss, Yair. Multibody factorization with uncertainty and missing data using the em algorithm. In *CVPR*, 2004.

Ho, Jeffrey, Yang, Ming-Hsuan, Lim, Jongwoo, Lee, Kuang-Chih, and Kriegman, David J. Clustering appearances of objects under varying illumination conditions. In *CVPR*, 2003.

Horn, R. and Johnson, C. *Topics in Matrix Analysis*. 1991.

Keshavan, Raghunandan, Montanari, Andrea, and Oh, Sewoong. Matrix completion from noisy entries. In *NIPS*, 2009.

Lee, K.C., Ho, J., and Kriegman, D. Acquiring linear subspaces for face recognition under variable lighting. *TPAMI*, 2005.

Lin, Zhouchen, Chen, Minming, Wu, Leqin, and Ma, Yi. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. Technical report, UIUC Technical Report UILU-ENG-09-2215, 2009.

Lu, Le and Vidal, René. Combined central and subspace clustering for computer vision applications. In *ICML*, 2006.

Ma, Yi, Derksen, H., Wei, Hong, and Wright, John. Segmentation of multivariate mixed data via lossy data coding and compression. *CVIU*, 2008a.

Ma, Yi, Yang, Allen, Derksen, Harm, and Fossum, Robert. Estimation of subspace arrangements with applications in modeling and segmenting mixed data. *SIAM Review*, 2008b.

Rao, Shankar, Tron, Roberto, Vidal, René, and Ma, Yi. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *TPAMI*, 2009a.

Rao, Shankar, Yang, Allen, Sastry, Shankar, and Ma, Yi. Robust algebraic segmentation of mixed rigid-body and planar motions in two views. *IJCV*, 2009b.

Shi, Jianbo and Malik, Jitendra. Normalized cuts and image segmentation. *TPAMI*, 2000.

Tron, Roberto and Vidal, René. A benchmark for the comparison of 3-d motion segmentation algorithms. In *CVPR*, 2007.

Wright, John, Tao, Yangyu, Lin, Zhouchen, Ma, Yi, and Shum, Heung-Yeung. Classification via minimum incremental coding length (MICL). In *NIPS*. 2008a.

Wright, John, Yang, Allen, Ganesh, A., Sastry, S. S., and Ma, Yi. Robust face recognition via sparse representation. *TPAMI*, 2008b.

Yan, Jingyu and Pollefeys, Marc. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *ECCV*, 2006.

Yang, Allen, Rao, Shankar, and Ma, Yi. Robust statistical estimation and segmentation of multiple subspaces. In *CVPR*, 2006.

Zhang, Teng, Szlam, Arthur, and Lerman, Gilad. Median k-flats for hybrid linear modeling with many outliers. *CoRR*, 2009.