Robust Subspace Structure Recovery and Subspace Segmentation via Low-Rank Representation

Abhinav Kumar
Indian Institute of Technology
Kanpur, India
Roll no: 16907018
Email: abhikmr@iitk.ac.in

Kaushal Kishore
Indian Institute of Technology
Kanpur, India
Roll no: 160327
Email: kaushalk@iitk.ac.in

Term Paper Mode: Mixed

***Abstract*—Our main focus in this term paper is the problem of subspace clustering. Consider a union of multiple subspaces and then suppose we draw a set of data samples from it approximately. Then we want to cluster these data samples into their respective subspaces and also remove outliers if any. In this term paper we discuss the Low-Rank Representation approach [1] to tackle this problem. In this approach, given a dictionary matrix containing bases, we want to find the lowest-rank representation of the data samples as a linear combination of these given bases. We formulate a convex optimization problem to solve this problem for three assumptions: when we have clean data then LRR can exactly recover the subspae structure of data samples; for data with outliers, LRR can detect outliers and exactly recover row space of original data under certain conditions; for data with noise corruptions, LRR can still recover row space of original data approximately.**

I. Introduction

In signal processing and pattern analysis, there is often a structure to the data which enables us to represent and process it intelligently. Of all the choices, the linear subspaces are the most common choice for this representation in real life applications due to less computational efficiency and easier representation. For this, considering data to have been approximately drawn from multiple low-rank subspaces is more reasonable than considering only a single subspace as is done in Principle Component Analysis (PCA) methods. The approach for considering data drawn from multiple subspaces is more general and leads to the problem of subspace segmentation otherwise known as clustering. The problem of subspace segmentation is to segment data into different clusters where each cluster is associated with a different subspace. The main issue we want to address is the handling of errors like noise, corruptions and outliers in the data. We therefore consider the problem of subspace clustering which aims to deal with above mentioned irregularities in the data.

Problem 1.1 (Subspace Clustering): Given that our data is drawn approximately from a union of linear subspaces, we want to segment all the data samples into their respective original subspaces and we want to handle any errors that may be present in the data.

Here, the error we want to handle is basically the amount with which the data is deviated from the underlying true subspace. It can be anything ranging from but not limited to noise, outliers, corruptions and missed entries. In this term paper, we consider mainly on sample-specific corruptions along with some attention to noisy data and random corruptions that may exist in the data. We handle the outliers, which is a data sample belonging to a far away subspace, and heavily corrupted sample, which actually belongs to the same subspace, similarly as they are not much different practically.

The technique we will use to solve this problem is lowest-rank representation (LRR). It works as follows: from all the possible linear representations of the data represented as a linear combination of the bases of a dictionary, the LRR aims to find the representation which has the lowest-rank and represents all of the data jointly. The LRR aims at solving a nuclear norm regularized convex optimization problem whose computational complexity is in polynomial time.

For a chosen dictionary matrix, we show that LRR solves the subspace clustering problem as follows: for clean data, we show that LRR exactly recovers the row space of the data; for data with outliers, LRR can detect outliers and recover exact row space of the data under certain conditions; for data arbitrarily corrupted with noise, LRR can recover the row space approximately with some theoretical guarantees.

II. Problem Statement

More precisely, we look at the following problem from [1] in this term paper:

Problem 2.1 (Subspace Clustering): Consider a union of k subspaces of unknown dimensions $\mathcal{X}_u = \{\mathcal{X}_i\}_{i=1}^k$ and k being an unknown as well. Let S_0 have n d-dimensional samples drawn from $\mathcal{X}_u$ and let its skinny SVD be $U_0\Sigma_0V_0$. Let the observations be generated as

$$S = S_0 + E_0 \tag{1}$$

where E_0 is the noise. Now, we want to recover $V_0V_0^\top$ termed as the Shape Interaction Matrix (SIM) [2] or to recover the row space of S_0. SIM is an orthogonal operator that projects the columns of S_0 to the subspace spanned by columns of V_0 and it also preserves the original subspace structure of S_0. The recovery of row space guarantees high segmentation accuracy as will be discussed later. It also naturally ensures error correction. Therefore, the problem of subspace clustering

can be reduced to recovery of row space that we can get from $V_0 V_0^\top$.

We will discuss the task of recovering row space under the following three assumptions:

1) Clean data, $E_0 = 0$.
2) Only a fraction of samples of all data are highly corrupted which is to say E_0 has sparse columns.
3) A fraction of all samples are highly corrupted and the rest have small Gaussian noise. E_0 has sparse columns and small Gaussian noise everywhere.

III. Matrix Recovery by Low-Rank Representation

A. Low-Rank Representation

For recovering the low-rank S_0 from the observations i.e., from $S = S_0 + E_0$ we consider the following regularized rank minimization problem:

$$\min_{D,E} rank(D) + \lambda \|E\|_\ell, \quad \text{s.t.} \quad S = D + E, \tag{2}$$

where $\lambda > 0$ and $\|.\|_\ell$ can be any norm for modelling different noises for regularization. The solution to (2), the low-rank D^* recovers the data S_0.

A more general form of this problem is to consider a dictionary matrix A which linearly spans the data space $\mathcal{X}_u$. Then the problem becomes:

$$\min_{C,E} rank(C) + \lambda \|E\|_\ell, \quad \text{s.t.} \quad S = AC + E. \tag{3}$$

The solution to (3), C^* is the lowest-rank representation of S w.r.t. the dictionary A. The original data is recovered as AC^* or by $S - E^*$. Also, $rank(AC^*) \leq rank(C^*)$ and therefore AC^* is also a low-rank recovery for S_0. For $A = I$, (3) becomes same as (2).

B. Low-Rank Representation Analysis

First, we consider the clean data problem:

$$\min_{C} rank(C), \quad \text{s.t.} \quad S = AC \tag{4}$$

to which the solutions aren't unique and we instead consider nuclear norm ℓ_* to get the convex optimization problem as follows:

$$\min_{C} \|C\|_*, \quad \text{s.t.} \quad S = AC \tag{5}$$

and the solution to (5) also solves (4).

Now we analyse some properties of (5) which build the foundations of LRR.

1) Uniqueness of Minimizer: The solutions to (5) can be many due to non-strict convexity of nuclear norm. But the minimizer to (5) is uniquely available in closed form by using the following theorem.

Theorem 3.1 Assuming $A \neq 0$ and $S \in span(A)$ i.e., $S = AC$ has feasible solution(s) then

$$C^* = A^\dagger S \tag{6}$$

minimizes the problem uniquely and $A^\dagger = A^\top (AA^\top)^{-1}$ is the pseudo-inverse of A.

The corollary mentioned below follows from the above theorem and it shows that solutions to (5) are also the solutions to (4).

Corollary 3.1: Let $S = AC$ have feasible solutions and $A \neq 0$. Let C^* minimize (5), then it implies that $rank(C^*) = rank(S)$ and therefore C^* is also a minimum rank solution for (4).

2) Block-Diagonal Property of the Minimizer:: The true segmentation can be achieved using lowest-rank representation if we choose the dictionary matrix appropriately. Assume (i) a collection of k subspaces $\{\mathcal{X}_1, \ldots \mathcal{X}_k\}$ with every $\mathcal{X}_i$ having a dimension/rank of $r_i > 0$, (ii) the dictionary distributed as $A = [A_1, \ldots A_k]$ and (iii) the data distributed as $S = [S_1, \ldots, S_2]$. Now, consider the following theorem.

Theorem 3.2: Assume that S_i consists of n_i samples from the subspace $\mathcal{X}_i$ and let A_i consist of m_i samples from $\mathcal{X}_i$ such that these samples are sufficient so that $rank(A_i) = r_i$ i.e., A_i consists of bases that span $\mathcal{X}_i$. Given that the subspaces are independent, the minimizer to (5) is block diagonal given as:

$$C^* = \begin{bmatrix} C_1^* & 0 & 0 & 0 \\ 0 & C_2^* & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & C_k^* \end{bmatrix}$$

where C_i^* is a coefficient matrix of size $m_i \times n_i$ and $rank(C_i^*) = rank(S_i), \forall i$.

C. Low-Rank Matrix Recovery by Convex Optimization

Consider (3), now by using corollary 3.1 we can replace the rank function with the nuclear norm. Moreover, we use $\ell_{2,1}$ norm defined as ($\|M\|_{2,1} = \sum_i \|[M]_{:,i}\|_2$) to characterize E as the error term for sample specific corruptions and outliers. Then a low-rank recovery of S_0 can be obtained using the solution to following convex optimization problem:

$$\min_{C,E} \|C\|_* + \lambda \|E\|_{2,1}, \quad \text{s.t.} \quad S = AC + E \tag{7}$$

On obtaining solutions (C^*, E^*) from (7), the low-rank recovery to S_0 can be obtained by AC^* or $(S - E^*)$.

To solve the convex problem in (7), we use Augmented Lagrange Multiplier (ALM) method. To use ALM we convert the problem in (7) to the following form:

$$\min_{C,E,J} \|J\|_* + \lambda \|E\|_{2,1}, \quad \text{s.t.} \quad S = AC + E, C = J$$

Now, the ALM method will minimize the following:

$$\mathcal{L} = \|J\|_* + \lambda \|E\|_{2,1} + tr(G_1^\top (S - AC - E)) + tr(G_2^\top (C - J)) + \frac{\mu}{2}(\|S - AC - E\|_F^2 + \|C - J\|_F^2)$$

where ℓ_F is the matrix Frobenious norm. The above is an augmented Lagrange function which is an unconstrained minimization problem. Hence, it can be minimized w.r.t. J, C and E respectively by fixing the other variables and then updating G_1 and G_2 which are the Lagrange multipliers and

Algorithm 1 Inexact ALM to solve problem in (7)

Input: data matrix S and parameter λ

Initialize: $C = J = 0, E = 0, G_1 = 0, G_2 = 0, \mu = 10^{-6}, \mu_{max} = 10^6, \rho = 1.1$ and $\epsilon = 10^{-8}$

while not converged **do**

1) fix the other variables and update J by

$$J = \arg\min \frac{1}{\mu}\|J\|_* + \frac{1}{2}\|J - (C + G_2/\mu)\|_F^2$$

2) fix the others and update C by

$$C = (\mathrm{I} + A^T A)^{-1}(A^T(S - E) + J + (A^T G_1 - G_2)/\mu)$$

3) fix the others and update E by

$$E = \arg\min \frac{\lambda}{\mu}\|E\|_{2,1} + \frac{1}{2}\|E - (S - AC + G_1/\mu)\|_F^2$$

4) update the multipliers

$$G_1 = G_1 + \mu(S - AC - E)$$
$$G_2 = G_2 + \mu(C - J)$$

5) update the parameter μ by $\mu = \min(\rho\mu, \mu_{\max})$

6) check the convergence conditions

$$\|S - AC - E\|_\infty < \epsilon \quad \text{and} \quad \|C - J\|_\infty < \epsilon$$

end while

the parameter $\mu > 0$ enforces penalty. The steps of inexact ALM method can be found in Algorithm 1.

Step 1 and Step 2 of Algorithm 1 are convex in nature and their solutions are present in closed form. We solve Step 1 using the theorem by Cai et al, using the Singular Value Thresholding (SVT) operator [3], which states that for any matrix M with SVD $M = U\Sigma V^*$, $\mathcal{D}_\tau(M) = \arg\min_M \left[\frac{1}{2}\|M - Y\|_F^2 + \tau\|M\|_*\right]$ where $\mathcal{D}_\tau(M) := U\mathcal{D}_\tau(\Sigma)V^*, \mathcal{D}_\tau(\Sigma) = diag(\{\sigma_i - \tau\}_+)$ and $t_+ = \max(0, t)$

To solve Step 3, consider the following lemma:

Lemma 3.1: [4] For a given matrix Q if the optimal solution to

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

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

$$[W^*]_{:,i} = \begin{cases} \frac{\|Q_{:,i}\|_2 - \alpha}{\|Q_{:,i}\|_2}, & \text{if } \|Q_{:,i}\|_2 > \alpha; \\ 0, & \text{otherwise.} \end{cases}$$

1) Convergence Properties: According to [5] we need to consider the following two conditions, (i) the dictionary matrix A if of full column rank and (ii) error ϵ_k in each iteration is monotonically decreasing. These two conditions are sufficient for convergence of Algorithm 1. The error ϵ_k is given by

$$\epsilon_k = \|(C_k, J_k) - \arg\min_{C,J} \mathcal{L}\|_F^2$$

where (C_k, J_k) are the solutions at k-th iteration.

2) Computational Complexity: Assuming the size of A and S both to be $d \times n$ we can see that the most of the computation is being done in Step 1 of Algorithm 1 where we need to compute SVD of an $n \times n$ matrix. Therefore if the size of data samples n is large this computation can be time consuming. Consider the following theorem that follows from Theorem 3.1 and helps in reducing computational cost for LRR.

Theorem 3.3: From the LRR problem in (7) it is easy to see that its solution (C^*, E^*) follows

$$C^* \in span(A^\top).$$

The above theorem implies that the optimal solution C^* of (7) lies in the row space spanned by A. Hence, C^* can be factorized as $C^* = P^*\tilde{C}^*$ where, by orthogonalizing columns of $A^\top$ we can compute P^* in advance and thus (7) can be written equivalently in a simpler form as:

$$\min_{\tilde{C},E} \|\tilde{C}\|_* + \lambda\|E\|_{2,1}, \quad \text{s.t.} \quad S = B\tilde{C} + E$$

where $B = AP^*$. From the solution $(\tilde{C}^*, E^*)$ we can obtain the optimal solution to (7) which will be $(P^*\tilde{C}^*, E^*)$. To obtain the optimal solution to (7) we only had to work with $\tilde{C}$ which has at most $r_A (= rank(A))$ rows, the complexity of solving the above problem using Algorithm 1 is $O(dnr_A + nr_A^2 + r_A^3)$. Therefore, for a low-rank dictionary A, LRR can be scalable. If we use $A = S$ then the computational complexity becomes at most $O(d^2n + d^3), d \leq n$. For low data dimension d, this is fast as well.

The total complexity of Algorithm 1 after considering orthogonalization cost and iterations required for convergence, let n_s, we get

$$O(d^2n) + O(n_s(dnr_A + nr_A^2 + r_A^3))$$

The choice of ρ affects n_S, if ρ is large then n_s is small and vice versa.

IV. Subspace Clustering by LRR

Here, we apply LRR to Problem 2.1 and present both theoretical and experimental results.

A. Exactness to Clean Data

For clean data i.e., $E_0 = 0$ and $S = S_0$ we can exactly recover the row space of S_0 through $V_0V_0^\top$ by solving the nuclear norm minimization problem

$$\min_C \|C\|_*, \quad \text{s.t.} \quad S = SC \tag{8}$$

where $A = S$ is the chosen dictionary matrix in (5). We have the following theorem from [6] by using Theorem 3.1.

Theorem 4.1: We assume skinny SVD of S to be $U\Sigma V^\top$. Then the minimizer to (8) is unique and defined as

$$C^* = VV^\top$$

which directly implies that C^* recovers $V_0V_0^\top$ exactly for $E_0 = 0$.

Theorem 4.1 shows the connection between LRR and PCA in [2]. It will be shown below that LRR can exactly recover the row space of S_0 even when the data has outliers whereas this is where PCA fails.

B. Robusteness to Outliers and Sample-Specific Corruptions

In our second assumption we consider a fraction of data to be away from the underlying subspaces. Which means that E_0 has sparse columns and therefore we can use $\ell_{2,1}$ norm to characterize the error E_0. We take the convex optimization problem in (7) and choose $A = S$ as a dictionary matrix to get the following:

$$\min_{C,E} \|C\|_* + \lambda\|E\|_{2,1}, \quad \text{s.t.} \quad S = SC + E \tag{9}$$

We show that $A = S$ is actually a good choice for a dictionary matrix in the next subsections.

1) Exactness to Outliers: If a data sample is away from the underlying subspace then it is considered as an outlier. In this case, in $S = S_0 + E_0$ S_0 is the data actually sampled from the underlying subspaces and E_0 will contain the outlier samples. We consider another constraint over S_0 to assess this setting precisely which is

$$\mathcal{P}_{\mathcal{I}_0}(S_0) = 0 \tag{10}$$

where $\mathcal{I}_0$ represents the column support or the indices of outliers in E_0 and $\mathcal{P}_{\mathcal{I}}$ is a projection matrix where for any matrix M, $\mathcal{P}_{\mathcal{I}}(M)$ is obtained by setting $[M]_{:,i} = 0 \quad \forall \quad i \in \mathcal{I}$. Let n be the total number of data samples in S, then define $\gamma = \mid \mathcal{I}_0 \mid /n$ as the fraction of outliers and define r_0 to be the rank of S_0. Now the following theorem states that LRR detects indices of outliers and exactly recovers the row space of S_0.

Theorem 4.2 [7]: There exists an optimal $\gamma^* > 0$ such that using the parameter $\lambda = \frac{3}{7\|S\|\sqrt{\gamma^* n}}$ in LRR always results in success for $\gamma \leq \gamma^*$. The success here means that for any minimizer (C^*, E^*) to (9) produces

$$U^*(U^*)^\top = V_0 V_0^\top \quad \text{and} \quad \mathcal{I}^* = \mathcal{I}_0 \tag{11}$$

where $\mathcal{I}^*$ represent indices/column supports of E^* and U^* represent the column space of C^*.

The condition of parameter setting of λ for $\gamma \leq \gamma^*$ is only a sufficient condition for success of LRR and in practice, for $\gamma > \gamma^*$ it is possible to find values of λ that may achieve success of LRR.

2) Robustness to Sample-Specific Corruptions: : The other way to look at the the data sample that is far away from the underlying subspaces is that it does belongs to the subspace but is grossly corrupted. These are usually sample-specific corruptions. This is again modelled as E_0 having sparse column supports and we can still use (9) formulation but we cannot consider the setting in (10). This implies that the exact recovery of row space $V_0 V_0^\top$ may not be possible by using LRR. The result $\mathcal{I}^* = \mathcal{I}_0$ still holds in practice as E_0 can still detect indices of these sample-specific corruptions.

In LRR, a data sample that is corrupted enough to be considered as an outlier will be considered so as this is a reasonable manipulation. For example, if an image of a face is corrupted to look like like a non-face, we can treat it as an outlier.

C. Robustness in the Presence of Noise, Outliers and Sample-Specific Corruptions

For noisy data, we cannot have sparse column support for E_0 but we can still use the problem formulation in (9) and $\ell_{2,1}$ norm can also handle signals with approximately sparse column support. As all of the data is noisy, its not likely to recover the exact row space $V_0 V_0^\top$ but we can try to get near exact recovery. Consider the following theorem based on triangle inequality of matrix norms.

Theorem 4.3: For $d \times n$ data matrix S and S_0 with rank r_0, any solution (C^*, E^*) to (9) with $\lambda > 0$ we have

$$\|C^* - V_0 V_0^\top\|_F \leq min(d, n) + r_0$$

D. Algorithms for Subspace Segmentation, Model Estimation and Outlier Detection

1) Segmentation with Given Subspace Number: On solving (9) we get the solution (C^*, E^*) and the column space of C^* is characterized by $U^*(U^*)^\top$ which is utilized for subspace segmentation. An affinity matrix W can be defined by assuming skinny SVD of C^* to be $U^*\Sigma(V^*)^\top$ as follows:

$$[W]_{ij} = ([\tilde{U}\tilde{U}^\top]_{ij})^2 \tag{12}$$

where $\tilde{U} = U^*(\Sigma^*)^{1/2}$ and it has normalized rows. The goal of multiplying $(\Sigma^*)^{1/2}$ is to assign a weight to columns of U^*. Squaring is done to ensure that the value of the affinity matrix are positive. For case with clean data $\Sigma^* = I$ and hence there is no effect of this method. We then use Normalized Cuts (NCuts) [8] for spectral clustering and segmenting the data samples into given k subspaces. In NCut, a diagonal matrix where the diagonal is obtained by summing the affinity matrix row-wise and taking the inverse square root of the resulting elements. This diagonal matrix is then used with the affinity matrix to produce a Laplacian matrix. SVD is performed over this Laplacian matrix and the obtained row space is multiplied with the diagonal matrix calculated earlier. Finally, KMeans clustering algorithm is applied to obtain the labels. Steps for doing segmentation using LRR can be found in Algorithm 2.

2) Estimating the Number of Subspaces k: Estimating the number of spaces is a challenging problem but we can make use of block-diagonal structure of the affinity matrix and resolve this problem of estimating number of clusters. For a strict block-diagonal affinity matrix W we can do the following to estimate the number of subspaces k.

We first compute the normalized Laplacian (L) of the affinity matrix W. Then we count the number of zero singular values of L. For a more practical case i.e., when the affinity matrix is near block-diagonal, we can count the number of singular values less than a threshold to estimate the number of subspaces. Below is a soft threshold technique to estimate the number of subspaces $\hat{k}$.

$$\hat{k} = n - int(\Sigma_{i=1}^{n} f_\tau(\sigma_i)) \tag{13}$$

where n is the total number of data samples. The singular values of Laplacian matrix L are given by $\{\sigma_i\}_i^n$ and $int(.)$

Algorithm 2 Subspace Segmentation

Input: data matrix S, number of subspaces K

1) calculate solution to (9)
2) calculate skinny SVD

$$C^* = U^* \Sigma^* (V^*)^\top$$

3) construct affinity matrix W by using (12)
4) perform NCut by using W and segment sample data into k clusters

Algorithm 3 Subspace Number k Estimation

Input: data matrix S

1) construct affinity matrix W by using (12)
2) compute the Laplacian matrix as

$$L = I - D^{-1/2} W D^{-1/2}$$
$$\text{where} \quad D = diag(\Sigma_j [W]_{1j}, \ldots, \Sigma_j [W]_{nj})$$

3) use (13) to estimate subspace number k

outputs the closest integer to the real value input. The soft thresholding operator $f_\tau(.)$ is given by

$$f_\tau(\sigma) = \begin{cases} 1, & if \sigma \geq \tau, \\ \log_2(1 + \frac{\sigma^2}{\tau^2}), & otherwise, \end{cases}$$
$$\text{where } \tau \text{ is a parameter and} \quad 0 < \tau < 1$$

Steps for estimating number of subspaces k can be found in Algorithm 3.

3) Outlier Detection: As we have already seen in Theorem 4.2, we can utilize E^* to detect indices of the outliers. We only need to find nonzero columns of E^* given that a fraction of data is clean i.e., there is no noise over all data. Here as well we can use thresholding technique when E^* has approximate sparse columns only. We can decide whether or not the i-th data vector of S is an outlier by using the following threshold

$$\|[E^*]_{:,i}\|_2 > \delta \tag{14}$$
$$\text{where} \quad \delta > 0$$

and δ is a parameter. If for $[E^*]_{:,i}$ the inequality (14) is true then that data vector of S is an outlier otherwise not.

V. Experiments and Numerical Analysis

A. Motion Segmentation on Hopkins155

1) Dataset: The Hopkins155 dataset [9] consists of 155 motion sequences each having some number of points tracked over different frames. The subspace clustering algorithm is applied on these sequences to segment the points correctly and the result is compared to the ground truths. Note that here we know the number of subspaces beforehand.

2) Parameters: We ran simulations of the subspace clustering algorithm on the Hopkins155 for various values of λ and calculated the mean error for each of them as plotted in Fig. 1. It can be noted that the optimum performance is achieved for $\lambda = 4.6$ with a mean error of 0.0172 over all the motion sequences. The error values at $\lambda = 4.6$ are summarized in Table I. As summarized by Liu et. al. [1], PCA had a mean error of 0.0456, SR had 0.0389 and LRR has 0.0171. It is evident that the LRR method outperforms them. The drawback of this LRR method is that we cannot estimate the optimum λ beforehand. However, heuristically it's value is kept high for low errors in the data and low for high errors in the data [1].

mean	std	max
0.0172	0.0489	0.3333

TABLE I: Errors of Motion Segmentation on Hopkins155 Dataset at optimum $\lambda = 4.6$

parameter τ	0.06	0.07	0.08	0.09	0.10	0.11
prediction rate	0.22	0.27	0.33	0.36	0.40	0.44
mean error	0.38	0.35	0.31	0.30	0.29	0.27

TABLE II: Subspace Number Estimation Results (on Hopkins155) at $\lambda = 4.0$

B. Face Segmentation

1) Dataset: Yale Face Dataset B [10] is used for face segmentation. It consists of human face images in various lighting setups.

2) Parameters: We ran simulations of the LRR subspace segmentation for different values of λ and calculated the corresponding accuracy. The optimum was obtained at $\lambda = 0.16$ with an accuracy of 0.63. The accuracy values for different values of λ are plotted in Fig. 2.

C. Choice of Norms

It is worth noting that, in the convergence criteria of algorithm 1 (LRR), Liu et. al. [1] used $\|M\|_\infty = \max_{i,j} \mid [M_{ij}] \mid$ for a matrix M of size $m \times n$ whereas the general infinity norm is defined as $\max_{1 \leq i \leq m} \sum_{j=1}^{n} \mid [M_{ij}] \mid$ which is the maximum of the absolute row sum of the matrix. We ran the LRR algorithm for the subspace segmentation problem with different norms to determine if the LRR algorithm had converged. The error and the number of iterations required were observed for different choices of the norm in the motion segmentation problem. While the results were almost similar for all the norms, the l_{-2} norm, which is the smallest singular value of the matrix, stood out as an exception. The number of iterations required were almost halved compared to when the infinity norm was used. This, however, was at a cost of an increased error. The results of using different norms are summarized in Table III.

The convergence error evolve very differently for the two choices of norms as shown in Fig. 3 It remains almost stable at a value for l_{-2} norm and suddenly decreases while it follows a very noisy path in the case of the infinity norm as used by Liu et. al. [1].

D. Estimation of Number of Subspaces

Algorithm 3 was applied on the Hopkins155 dataset. The infinity norm was used as usual. The simulations were run for different values of τ and the mean error and prediction rate was noted for each. The results are summarized in I. It was observed that the number of iterations required for the algorithm to converge was unaffected by the value of τ. 1 shows the evolution of the convergence error.

Norm	l_∞ (of [1])	l_∞	l_2	l_1	l_{-2}	l_{-1}
iterations	283	321	298	304	126	251
mean error	0.0172	0.0172	0.0172	0.0172	0.1110	0.0172
std error	0.0489	0.0489	0.0489	0.0489	0.1468	0.0489
max error	0.3333	0.3333	0.3333	0.3333	0.6179	0.3333

TABLE III: Results on Motion Segmentation on Hopkins155 Dataset ($\lambda = 4.0$)

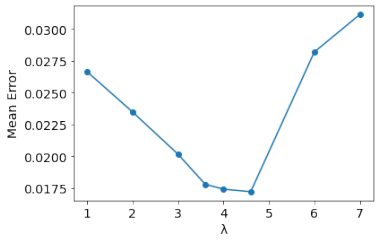


Fig. 1: Mean Error vs λ for Motion Segmentation on Hopkin155 Dataset

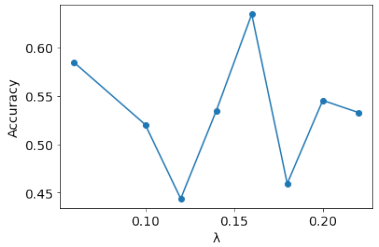
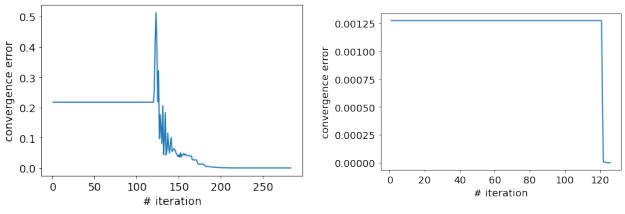


Fig. 2: Segmentation Accuracy vs λ for Face Segmentation on Yale Face Dataset B



(a) Infinity Norm (b) l_{-2} Norm

Fig. 3: Comparison of the convergence error between different choice of norms

VI. Suggested Improvements

One of the enhancement of LRR is the Latent LRR or LatLRR [11] which outperforms many state-of-the-art algorithms. A *limitation* of LRR is that if the data is grossly corrupted or insufficient then choosing dictionary matrix as $A = S$ will decrease the performance of LRR significantly. LatLRR overcomes this limitation by considering hidden/unobserved data along with observed data and uses both to jointly weave subspace clustering and feature extraction together into a unified framework. Moreover, it is an unsupervised algorithm and thus is robust to corruptions when it comes to extracting features exactly.

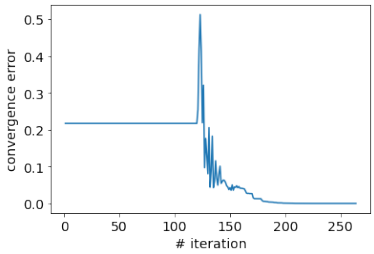


Fig. 4: Convergence of LRR when Estimating Subspace Number k

Another way to go is to consider Deep Dictionary Learning [12] which tries to learn the dictionary by extracting salient features using a stack of layers and a large number of weights and filters.

References

[1] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *CoRR*, vol. abs/1010.2955, 2010. [Online]. Available: http://arxiv.org/abs/1010.2955

[2] J. Costeira and T. Kanade, "A multi-body factorization method for independently moving objects," *International Journal of Computer Vision*, vol. 29, pp. 159–179, 09 1998.

[3] J.-F. Cai, E. J. Candes, and Z. Shen, "A singular value thresholding algorithm for matrix completion," 2008.

[4] J. Yang, W. Yin, Y. Zhang, and Y. Wang, "A fast algorithm for edge-preserving variational multichannel image restoration," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 569–592, 2009. [Online]. Available: https://doi.org/10.1137/080730421

[5] J. Eckstein and D. P. Bertsekas, "On the douglas–rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Program.*, vol. 55, no. 1–3, p. 293–318, Apr. 1992.

[6] S. Wei and Z. Lin, "Analysis and improvement of low rank representation for subspace segmentation," *CoRR*, vol. abs/1107.1561, 2011. [Online]. Available: http://arxiv.org/abs/1107.1561

[7] G. Liu, H. Xu, and S. Yan, "Exact subspace segmentation and outlier detection by low-rank representation," in *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, N. D. Lawrence and M. Girolami, Eds., vol. 22. La Palma, Canary Islands: PMLR, 21–23 Apr 2012, pp. 703–711. [Online]. Available: http://proceedings.mlr.press/v22/liu12a.html

[8] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[9] R. Tron and R. Vidal, "A benchmark for the comparison of 3-d motion segmentation algorithms," vol. 8, 07 2007, pp. 1–8.

[10] K.-C. Lee, J. Ho, and D. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 684–698, 2005.

[11] G. Liu and S. Yan, "Latent low-rank representation for subspace segmentation and feature extraction," in *2011 International Conference on Computer Vision*, 2011, pp. 1615–1622.

[12] S. Tariyal, A. Majumdar, R. Singh, and M. Vatsa, "Deep dictionary learning," *IEEE Access*, vol. 4, pp. 10 096–10 109, 2016.