Subspace Clustering with Applications to Dynamical Vision (CS 229 Final Project)

Adel Javanmard Mahdi Soltanolkotabi

December 10, 2010

1. INTRODUCTION

Data that arises from engineering applications often contains some type of low dimensional structure that enables intelligent representation and processing. This leads to a very challenging problem: discovering compact representations of high-dimensional data. A very common approach to address this problem is modeling data as a mixture of multiple linear (or affine) subspaces. Given a set of data points drawn from a union of subspaces, subspace clustering refers to the problem of finding the number of subspaces, their dimension and the segmentation of data. In this sense, subspace clustering can be thought of as a generalization of the Principal Component Analysis (PCA) method. In PCA, the underlying assumption is that the data is drawn from a single subspace while in subspace segmentation, data is assumed to be concentrated around multiple low rank subspaces.

A number of methods have been developed to address the subspace clustering problem, including algebraic methods, spectral clustering-based methods and statistical methods. On the heels of compressed sensing and low rank matrix recovery[1, 2, 3], a new class of algorithms have very recently emerged. These algorithms try to represent each data point, in a union of subspaces, as a linear combination of all other points. By enforcing a low dimensional structure on this representation (e.g. l1-norm for sparsity or nuclear norm for low-rankness), an affinity matrix is built which is subsequently used for subspace segmentation. This new approach resolves the exponential complexity issues of some of the previous methods and is more robust to noisy data. Motivated by this approach, we propose a new algorithm, called *Nuclear-* ℓ_1 for the segmentation problem. Moreover, we present an iterative scheme as an alternative method for handling large scale problems.

Subspace segmentation has many applications in computer vision, image processing, and system theory. In this project we will mainly focus on motion segmentation, and use the Nuclear-ℓ_1 algorithm for this problem. Based on the simulation results on the Hopkins155 motion database, the performance of Nuclear-ℓ_1 is quite competitive.

2. SUBSPACE SEPARATION

2.1 Noiseless Scenario

Let X be a matrix in $\mathbb{R}^{n\times N}$ with columns drawn from a union of k subspaces $\{S_i\}_{i=1}^k$, of unknown dimensions, embedded in a larger space with dimension n (the data has an ambient dimension n). More specifically $X = [X_1, \cdots, X_k]P$, where $X_i \in \mathbb{R}^{n\times N_i}$ is the set of N_i samples drawn from the i^{th} subspace S_i and P is a permutation marix. Also $N = \sum_{i=1}^k N_i$ denotes the total number of samples. Assuming that the subspaces are of low dimension and independent [1], the goal is to segment all data points into their respective subspaces (see Fig.1. for an illustration).

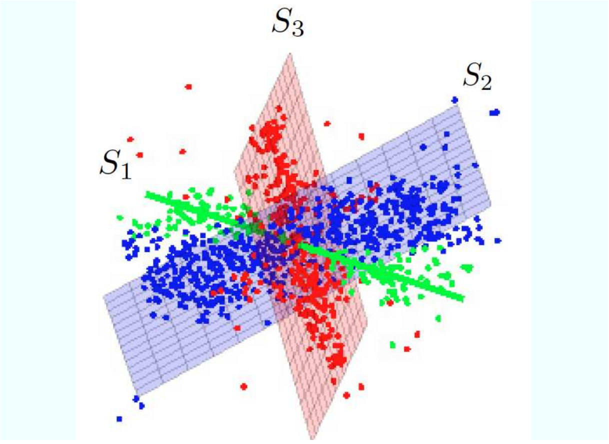


Figure 1: A set of points concentrated around the union of subspaces S_1, S_2, and S_3.

There are two algorithms that use low dimensional representation for subspace clustering. Both of these algorithms have provable recovery gaurantees when the subspaces are independent [4, 6]. We will briefly describe them in the next two subsections. Based on these two algorithms we will propose a new algorithm (Nuclear-ℓ_1) which is subsequently described.

2.1.1 *Low Rank Representation (LRR) [4]*

In this setting the assumption is that each data vector can be represented by a linear combination of a small number of vectors in a *dictionary* $A = [a_1, a_2, \cdots, a_m]$, i.e., $X = AZ$, where $Z \in \mathbb{R}^{N\times N}$ is the coefficient matrix. A natural choice for the dictionary is $A = X$, i.e. one tries to write each point as a linear combination of all the data points. This is a reasonable assumption if the sample points are sufficiently dense in their respective subspaces. In this sense each X_i is self-expressive : $X_i = X_i Z_i$ for some $Z_i \in \mathbb{R}^{N_i\times N_i}$. Hence, the permuted block-diagonal matrix $Z = diag(Z_1, Z_2, \cdots, Z_k)P$ satisfies $X = XZ$.

[1] The subspaces are independent if and only if $\sum_{i=1}^k S_i = \oplus_{i=1}^k S_i$

The idea of LRR method is to enforce a low-rank structure on Z as a surrogate for “block-diagaonal-ness” of Z by solving the problem:

$$\begin{aligned} \min_{Z} \quad & \operatorname{rank}(Z) \\ \text{s.t.,} \quad & X = AZ \end{aligned} \tag{2.1}$$

After finding Z one can perform subspace segmentation by spectral clustering algorithms such as normalized cut [7]. It is well known that (2.1) is NP-hard. A common relaxation to rank minimization problem is to replace the rank function with the nuclear norm, resulting in the following problem:

$$\begin{aligned} \min_{Z} \quad & ||Z||_* \\ \text{s.t.,} \quad & X = AZ \end{aligned} \tag{2.2}$$

It has been shown in [4] that this problem effectively recovers the block diagonal structure when the subspaces are independent. The details of LRR is shown in Algorithm 1.

```
Algorithm 1: Subspace Segmentation by LRR
Input: data matrix $X$, number of subspaces $k$.
1. Solve the optimization problem to get $Z^*$.
2. Construct an undirected graph by using $|Z^*| + |(Z^*)^T|$ as the affinity matrix of the graph.
3. Use Normalized Cut to segment the vertices of the graph into $k$ clusters.
```

2.1.2 Sparse Reconstruction Method (SR) [6]

Sparse Reconstruction Method (SR) enforces sparsity on the coefficient matrix Z by minimizing its ℓ_1-norm. Formally, it solves the problem:

$$\begin{aligned} \min_{Z} \quad & ||Z||_{\ell_1} \\ \text{s.t.,} \quad & X = AZ, \\ & [Z_{ii}] = 0. \end{aligned} \tag{2.3}$$

Note that without the constraint $[Z_{ii}] = 0$, the method is prone to return the trivial solution $Z = I$. After finding the affinity matrix Z, similar to LRR, the algorithm uses spectral clustering algorithms for subspace clustering. It has been shown in [6] that this problem also recovers the block diagonal structure when the subspaces are independent.

2.1.3 Proposed Method (Nuclear - ℓ_1 Algorithm)

The LRR algorithm is pretty robust to (gross) noise; however, it shows poor performance if the independence assumption about the subspaces is not valid. On the other hand, SR looks for the sparsest coefficient matrix Z and is more likely to work when the subspaces are not independent.

Motivated by this observation, we propose the nuclear-ℓ_1 algorithm as a unifying approach that incorporates LRR and SR methods. This algorithm obtains the coefficient matrix Z by solving

$$\begin{aligned} \min_{Z} \quad & ||Z||_{\ell_1} + \theta ||Z||_* \\ \text{s.t.,} \quad & X = AZ \end{aligned} \tag{2.4}$$

Note that by setting $\theta = 0$, the nuclear-ℓ_1 algorithm reduces to SR. Also, in the limit $\theta \to \infty$, nuclear-ℓ_1 reduces to LRR. Therefore, it can be thought of a generalization of both LRR and SR methods and is likely to inherit the benefits of each of them simultaneously. Some of the useful properties of Nuclear-ℓ_1 norm and its applications/proof guarantees in data modeling in general (and more specifically subspace extraction) is discussed in a paper by the second author [11].

2.2 Noisy Scenario

According to [5], the main challenge of subspace segmentation is to handle noisy in data, i.e., to handle the data that may not strictly follow subspace structures. The authors in [4] claim that the LRR method has a better performance in the presence of noise compared with previous algorithms. As the experimental reports in [5] demonstrate, the SR based segmentation method are quite competitive in real-world applications. However, since SR finds the sparsest representation of each data vector individually, it is not guaranteed to capture the global structure of the data. This drawback can adversely affect the performance when the data is grossly corrupted.

Assume that a percentage of the data vectors are grossly corrupted and the others are contaminated by small noise. Therefore, $X = X^0 + E$, where X is the observed data and E is the noisy part. In order to recover the low rank matrix Z from the observed data, LRR solves the following problem:

$$\begin{aligned} \min_{Z,E} \quad & ||Z||_* + \lambda ||E||_{2,1} \\ \text{s.t.,} \quad & X = AZ + E \end{aligned} \tag{2.5}$$

where $||.||_{2,1}$ denotes the ℓ_2/ℓ_1-norm[2].

Similarly, the nuclear-ℓ_1 algorithm is modified in noisy scenarios by introducing a regularization term as follows.

$$\begin{aligned} \min_{Z,E} \quad & ||Z||_{\ell_1} + \theta ||Z||_* + \lambda ||E||_{2,1} \\ \text{s.t.,} \quad & X = AZ + E \end{aligned} \tag{2.6}$$

2.2.1 Iterative Method for solving Large Scale Problems using Nuclear-ℓ_1

The problem (2.6) is a convex optimization problem and can be solved using semidefinite programming solvers. These solvers are based on interior-point methods, and are problematic when the size of the matrix is large because they need to solve huge systems of linear equations to compute the Newton direction. In fact, they can only handle $n \times n$ matrices with $n \leq 100$. Therefore, we need to find another way to solve the above problem which is scalable to large matrices. To this end, we convert (2.6) to the following equivalent problem:

$$\begin{aligned} \min_{Z,E,J_1,J_2} \quad & ||J_1||_* + \theta ||J_2||_{\ell_1} + \lambda ||E||_{2,1} \\ \text{s.t.,} \quad & X = AZ + E, \\ & Z = J_1 \\ & J_1 = J_2 \end{aligned} \tag{2.7}$$

Using the Augmented Lagrange Multiplier (ALM) method, an iterative scheme can be proposed to solve this problem. Consider the following augmented Lagrange function:

$$\begin{gathered} ||J_1||_* + \theta ||J_2||_{\ell_1} + \lambda ||E||_{2,1} \\ + tr\left(Y_1^T(X - AZ - E)\right) + tr\left(Y_2^T(Z - J_1)\right) + tr\left(Y_3^T(J_1 - J_2)\right) \\ + \frac{\mu}{2}(||X - AZ - E||_F^2 + ||Z - J_1||_F^2 + ||J_1 - J_2||_F^2). \end{gathered}$$

[2] For matrix $A = [A_1, A_2, \cdots, A_n]$, where A_i denotes the i^{th} column, $||A||_{2,1}$ is given by $\sum_{i=1}^{n} ||A_i||_2$

The above problem is unconstrained and can be minimized with respect to J_1, J_2, Z, and E, respectively, by fixing the other variables. This results in the following update rules:

(i) Fix the others and update J_1 by

$$\begin{aligned} J_1 &= \arg\ \min ||J_1||_* + tr\left(Y_2^T(Z-J_1)\right) + tr\left(Y_3^T(J_1-J_2)\right) \\ &\quad + \frac{\mu}{2}||Z-J_1||_F^2 \\ &= \arg\ \min ||J_1||_* + \frac{\mu}{2}||J_1 - Z + (Y_3-Y_2)/\mu||_F^2 \\ &= \arg\ \min \frac{1}{\mu}||J_1||_* + \frac{1}{2}||J_1 - Z + (Y_3-Y_2)/\mu||_F^2. \end{aligned}$$

(ii) Fix the others and update J_2 by

$$\begin{aligned} J_2 &= \arg\ \min\ \theta||J_2||_{\ell_1} + tr\left(Y_3^T(J_1-J_2)\right) + \frac{\mu}{2}||J_1-J_2||_F^2 \\ &= \arg\ \min\ \theta||J_2||_{\ell_1} + \frac{\mu}{2}||J_1 - J_2 + Y_3/\mu||_F^2 \\ &= \arg\ \min\ \frac{\theta}{\mu}||J_2||_{\ell_1} + \frac{1}{2}||J_1 - J_2 + Y_3/\mu||_F^2. \end{aligned}$$

(iii) Fix the others and update Z by

$$Z = (I + A^TA)^{-1}(A^T(X-E) + J_1 + (A^TY_1 - Y_2)/\mu).$$

(iv) Fix the others and update E by

$$\begin{aligned} E &= \arg\ \min\ \lambda||E||_{2,1} + tr\left(Y_1^T(X-AZ-E)\right) \\ &\quad + \frac{\mu}{2}||X-AZ-E||_F^2 \\ &= \arg\ \min \frac{\lambda}{\mu}||E||_{2,1} + \frac{1}{2}||E - (X - AZ + Y_1/\mu)||_F^2. \end{aligned}$$

Here, $\mu > 0$ is a penalty parameter. Note that although the steps (i), (ii), and (iv) are convex problems, they all have simple closed form solutions.
Let $\mathcal{S}_\tau : \mathbb{R} \to \mathbb{R}$ denote the shrinkage operator $\mathcal{S}_\tau[x] = sgn(x) \cdot \max(|x| - \tau, 0)$, and extend it to matrices by applying it to each element. Furthermore, let $\mathcal{D}_\tau(X)$ denote the singular value thresholding operator for matrices given by $\mathcal{D}_\tau(X) = U\mathcal{S}_\tau(\Sigma)V^*$, where $X = U\Sigma V^*$ is any singular value decomposition. It is not hard to see that the solutions to step (i) and (ii) are respectively given by

$$\begin{aligned} J_1 &= \mathcal{D}_{1/\mu}(Z + (Y_2 - Y_3)/\mu), \\ J_2 &= \mathcal{S}_{\theta/\mu}(J_1 + Y_3/\mu). \end{aligned}$$

The solution to step (iv) is given by virtue of lemma 3.3 in [8]; Define $Q := X - AZ + Y_1/\mu$ and let q_i be the i^{th} column of Q. Then, the i^{th} column of the solution E, denoted by E_i is given by

$$E_i = \max(||q_i||_2 - \frac{\lambda}{\mu}, 0)\frac{q_i}{||q_i||_2}.$$

The inexact ALM method is outlined in algorithm 2.

3. EXPERIMENT ON SYNTHESIZED DATA

In this section we present the result of performing Nuclear-ℓ_1 LRR and SR on synthesized data. We perform experiments in both the noisy and noiseless case. Our criteria for evaluating the performance are:

$$precision = \frac{||M^0 o Z^*||_0}{||Z^*||_0}, \quad recall = \frac{||M^0 o Z^*||_0}{||M^0||_0}$$

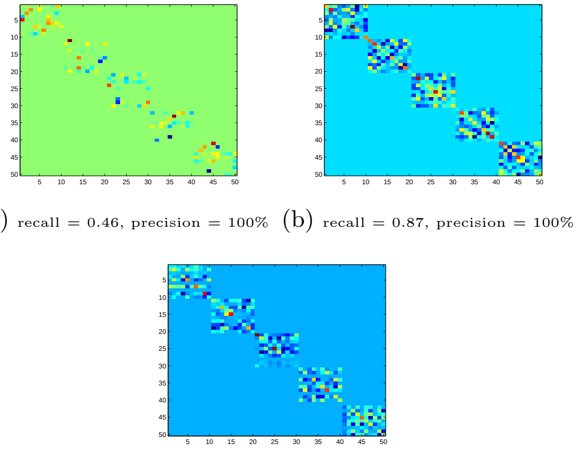
where o is the Hadamard product and M^0 is a mask matrix that has 1 on the blocks and zeros everywhere else.

```
**Algorithm 2:** Solving Problem (2.7) by ALM
**Input:** data matrix $X$, regularization parameters $\lambda$, $\theta$.
**Initialize:** $Z = J_1 = J_2 = 0,\ E = 0,\ Y_1 = Y_2 = Y_3 = 0,$
$\mu = 10^{-6}, U = 10^6,\ \rho = 1.1,$ and $\epsilon = 10^{-8}$.
**While** $||X - AZ - E||_\infty \geq \epsilon$ or $||Z - J1||_\infty \geq \epsilon$
or $||J_1 - J2||_\infty \geq \epsilon$ **do**
  1. Fix the others and update $J_1$ by
       $J_1 = \mathcal{D}_{1/\mu}(Z + (Y_2 - Y_3)/\mu)$.
  2. Fix the others and update $J_2$ by
       $J_2 = \mathcal{S}_{\theta/\mu}(J_1 + Y_3/\mu)$.
  3. Fix the others and update $Z$ by
       $Z = (I + A^TA)^{-1}(A^T(X - E) + J_1 + (A^TY_1 - Y_2)/\mu)$
  4. Fix the others and update $E$.
       Define $Q := X - AZ + Y_1/\mu$ and let $q_i$ be the $i^{th}$ column
       of $Q$. Then, the $i^{th}$ column of $E$, denoted by $E_i$ updates
       $E_i = \max(||q_i||_2 - \frac{\lambda}{\mu}, 0)\frac{q_i}{||q_i||_2}$
  5. Update the multipliers
       $Y_1 = Y_1 + \mu(X - AZ - E)$,
       $Y_2 = Y_2 + \mu(Z - J_1)$,
       $Y_3 = Y_3 + \mu(J_1 - J_2)$.
  6. Update $\mu$ by $\mu = \min(\rho\mu, U)$.
**end While.**
```

3.1 Experiment on Noiseless data

We construct 5 independent subspaces $\{S_i\}_{i=1}^5$ whose basis $\{U_i\}_{i=1}^5$ are 40×3 matrices with entries i.i.d. $\mathcal{N}(0, 1)$. Thus each subspace has rank 3 and the ambient dimension is 40. We construct a 40×50 data matrix $X = [X_1, \cdots, X_5]$ by sampling 10 data vectors from each subspace by $X_i = U_iC_i$, $1 \leq i \leq 5$ with C_i being a 3×10 i.i.d. $\mathcal{N}(0, 1)$ matrix.

The affinity matrix Z obtained by performing noiseless SR, LRR, and Nuclear-ℓ_1 on this data is shown in Fig. 2. As can be seen the off block diagonal terms are very small.



(a) recall = 0.46, precision = 100% (b) recall = 0.87, precision = 100%

(c) recall = 0.91, precision = 100

Figure 2: Comparison of the affinity matrices produced by (a) SR, (b) LRR, (c) Nuclear-ℓ_1 on noiseless synthesized data.

The value of recall and precision using each algorithm is presented in the caption of each subfigure. Notice that all algorithms have *precision* = 100%, this is in line with theory because with high probability the subspaces U_i will be independent in this case and thus we expect all algorithms to perform well.

3.2 Experiment on Noisy data

We sample 100 data vectors from 5 subspaces constructed in a similar way as in Example 3.0.1. This time, a percentage of the data are grossly corrupted by a multiplicative Gaussian noise $\mathcal{N}(0, 0.2)$ and the rest are contaminated by small additive Gaussian noise $\mathcal{N}(0, 0.01)$. We use the modified versions of LRR and Nuclear-ℓ_1 for noisy case to segment data. The parameter λ in LRR was set to 0.2. The parameters λ and θ in Nuclear-ℓ_1 were set to 0.2 and 0.1, respectively. The results are summarized in table 1. These results verify that Nuclear-ℓ_1 is more robust to gross noise.

Table 1: Experiment results on noisy synthesized data

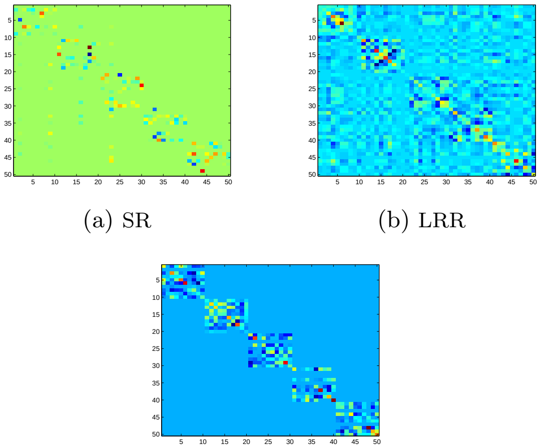
	LRR		Nuclear-ℓ_1	
Percentage of large noise	recall	precision	recall	precision
10 %	62%	92 %	64 %	99.84 %
20 %	60%	81 %	61 %	98.61 %
30 %	51%	70 %	55 %	97.35 %

3.3 Comparison of robustness to the independence assumption

We construct 5 independent subspaces $\{S_i\}_{i=1}^5$ whose basis $\{U_i\}_{i=1}^5$ are 10×3 matrices with entries i.i.d. $\mathcal{N}(0, 1)$. Thus each subspace has rank 3 and the ambient dimension is 10. We construct a 10×50 data matrix $X = [X_1, \cdots, X_5]$ by sampling 10 data vectors from each subspace by $X_i = U_i C_i$, $1 \leq i \leq 5$ with C_i being a 3×10 i.i.d. $\mathcal{N}(0, 1)$ matrix. The affinity matrix Z obtained by performing noiseless SR, LRR, and Nuclear-ℓ_1 on this data is shown in Fig. 3. Note that the subspaces are not independent in this case (5 subspaces of dimension 3 in $\mathbb{R}^{10}$). As can be seen in Fig. 3, LRR has a lot of nonzero coefficients in the off block-diagonal part, were as the non-zero coefficients in Z using SR and Nuclear-ℓ_1 is almost concentrated in the block diagonals, verifying our previous intuition that SR and Nuclear-ℓ_1 should perform better in this setting.

4. MOTION SEGMENTATION

One of the main problems in dynamic vision is the analysis of dynamic scenes. In these scenes, in addition to the motion of the camera, there are multiple moving object in the scene (usually independent of each other). Thus in order to analyze dynamic video scenes an initial step is *motion segmentation*. i.e. given multiple image frames of a dynamic scene taken by a (possibly moving) camera, the goal is to cluster the trajectory of feature points (on the moving objects) according to the different motions these trajectories belong to. In literature many different camera models have been proposed. In this project we will focus on the *affine* camera model.



(a) SR

(b) LRR

(c) Nuclear-ℓ_1

Figure 3: Comparison of the affinity matrices produced by SR, LRR, Nuclear-ℓ_1 without the independence assumption

4.1 Formulation of Motion Segmentation as Subspace Separation

Consider a feature point y, in 3-D space ($y \in \mathbb{R}^3$). and it's projection on the 2-D image plane ($x \in \mathbb{R}^2$). Under the affine camera model, these two quantities are related by a linear transformation:

$$x = A_f \begin{bmatrix} y \\ 1 \end{bmatrix} \tag{4.1}$$

where $A_f \in \mathbb{R}^{2\times4}$ is known as the *affine motion matrix*. More specifically as described in [5],

$$A_f = K \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

Here $K \in \mathbb{R}^{2\times3}$ is the camera calibration matrix and $(R, \mathbf{t}) \in SE(3)$ is the relative orientation of the image plane with respect to the world coordinates. Suppose we have access to the trajectories of P feature points of a rigid object, obtained from F 2-D image frames taken by a moving camera (denoted by $\{y_{fp}\}_{f=1...F}^{p=1...P}$, where y_{fp} is the projection of the p-th point onto the image plane in the f-th frame). The linear constrains of (4.1) can be lumped together in the form:

$$X = \mathbf{A_f} Y$$

where

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1P} \\ \vdots & \vdots & \ddots & \vdots \\ x_{F1} & x_{F2} & \cdots & x_{FP} \end{bmatrix}$$

$$A_f = \begin{bmatrix} A_1 \\ \vdots \\ A_F \end{bmatrix}; Y = \begin{bmatrix} y_1 & \cdots & y_P \\ 1 & \cdots & 1 \end{bmatrix}$$

where A_f is the affine motion matrix at frame f. These linear constraints relates the 3-D coordinates to the tracked feature points. Notice that

$$rank(X) = rank(A_f Y) \leq min(rank(A_f), rank(Y)) \leq 4$$

This latter fact shows that the trajectories of feature points from a single rigid motion will all lie in a linear subspace of $\mathbb{R}^{2F}$ of dimension at most four. In case there are k moving objects the trajectory of the feature points will lie in a union of k linear subspaces in $\mathbb{R}^{2F}$. Thus the problem of motion segmentation (separating feature points based on their movement) becomes equivalent to the separation of data drawn from multiple subspaces, based on the subspace they belong to.

5. EXPERIMENTS ON HOPKINS 155

In this section we test Nuclear-ℓ_1 on real motion segmentation tasks. Some previous state of-the-art methods are also included for comparison.

We evaluate Nuclear-ℓ_1, LRR and SR on the Hopkins 155 motion database [9]. The database consists of 155 sequences of two and three motions which can be divided into three main categories: checkerboard, traffic, and articulated sequences. The trajectories are extracted automatically with a tracker, and outliers are manually removed. Therefore, the trajectories are corrupted by noise, but do not have missing entries or outliers.

We consider two settings on this database. The first one is to compare all algorithms under the same circumstances. i.e. all algorithms use the raw data without any special preprocessing and we use the same spectral partitioning technique on all of them. In the second setting, different algorithms use specific preprocessing and postprocessing techniques to enhance their performance. Table 2 and 3 shows that Nuclear-ℓ_1 is competitive in both settings. Another interesting result is that the new class of algorithms (SR, LRR, Nuclear-ℓ_1) outperform more classical approaches in motion segmentation.

The basic algorithm of Nuclear-ℓ_1 was able to achieve an error rate of 2.87%. This performance can be enhanced by some additional pre-post processing techniques as follows (these are the same "tricks" used in the LRR method);

First, we notice that the data base has low noise level, therefore to avoid overfitting, we randomly chose 10% of the entries and corrupt them by adding small Gaussian noise. Second, since the affinity matrix Z is asymmetric, we convert it into a Positive Semi-Definite (PSD) matrix Z_1 by solving

$$\begin{aligned} \min_{Z_1} \quad & \|Z_1\|_* + \alpha \|E\|_1 \\ s.t. \quad & Z = Z_1 + E, \\ & Z_1 \succeq 0 \end{aligned}$$

with α set to be 0.8. Third, inspired by [10], we decompose Z_1 into $Z_1 = QQ^T$ and define $L = (\tilde{Q}\tilde{Q}^T)^2$, where $\tilde{Q}$ is Q with normalized rows. Fourth, we use $L^{\beta}(\beta = 2.26)$ as the affinity matrix for spectral clustering. Finally, in the interest of fair comparison since there is a degenerated sequence in Hopkins 155 and previous results disregard the degenerated sequence so do we.

Table 2: Classification error on raw data

Algorithm	GPCA	LSA	RANSAC
mean	NA	8.99%	8.22%
Algorithm	SR	LRR	Nuclear-ℓ_1
mean	3.89%	3.16%	2.87%

Table 3: Classification error with algorithm specific pre-post processing

Algorithm	GPCA	LSA	SR	LRR	Nuclear-ℓ_1
mean	10.34%	4.94%	1.24%	0.87%	0.69%

To give a visual verification of our algorithm Fig. 4 shows the result of applying Nuclear-ℓ_1 to one video sequences in the Hopkins 155 data base.

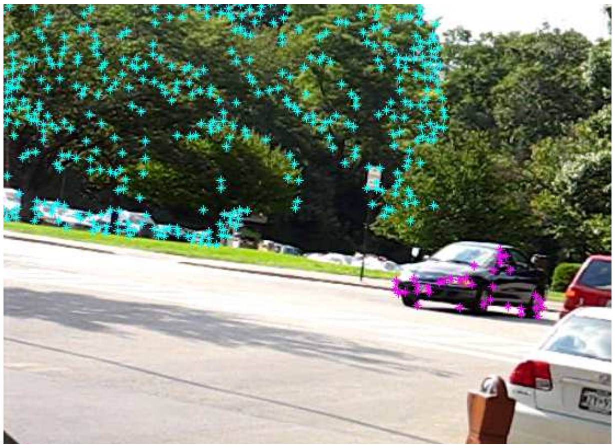


Figure 4: Visual performance verification of nuclear-ℓ_1 algorithm on an example frame from Hopkins 155.

6. REFERENCES

[1] E. Candes, J. Romberg, and T. Tao. *Stable signal recovery from incomplete and inaccurate measurements.* Communications on Pure and Applied Mathematics, 59(8):1207-1223, 2006

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

[3] E. J. Candes and B. Recht, *Exact matrix completion via convex optimization,* Foundations of Computational Mathematics, vol. 9, no. 6, pp. 717-772, 2009

[4] G. Liu et al., *Robust Recovery of Subspaces Structures by Low-Rank Representation*, arXiv 1010.2955v2

[5] S. Rao et al., *Motion Segmentation in the Presence of Outlying, Incomplete, or Corrupted Trajectories*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 32, no. 10, pp. 1832-1845, 2010.

[6] E. Elhamifar et al., *Sparse Subspace Clustering*, in IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2009, pp. 2790-2797.

[7] J. Shi et al., *Normalized Cuts and Image Segmentation*, IEEE Trans. on Pattern Analysis and Machine Intelligence, pp. 888-905, 2000.

[8] J. Yang et al., *A fast algorithm for edge-preserving variational multichannel image restoration*, SIAM Journal on Imaging Sciences, vol. 2, no. 2, pp. 569-592, 2009.

[9] R. Tron and R. Vidal, *A benchmark for the comparison of 3-d motion segmentation algorithms,* in IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1-8.

[10] F. Lauer and o. Christoph Schn *Spectral clustering of linear subspaces for motion segmentation,* in IEEE International Conference on Computer Vision, 2009.

[11] M. Soltanolkotabi, *The power of Nuclear-ℓ_1 relaxations in Modeling Data*, Stanford University Technical Report. under preperation.