Cascaded Low Rank and Sparse Representation on Grassmann Manifolds

Boyue Wang²,³, Yongli Hu¹,²,³, *, Junbin Gao⁴, Yanfeng Sun¹,²,³ and Baocai Yin⁵,¹,²,³

¹Beijing Advanced Innovation Center for Future Internet Technology, China
²Beijing Key Laboratory of Multimedia and Intelligent Software Technology, China
³Faculty of Information Technology, Beijing University of Technology, China
⁴The University of Sydney Business School, University of Sydney, NSW 2006, Australia
⁵Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, China

Abstract

Inspired by low rank representation and sparse subspace clustering acquiring success, ones attempt to simultaneously perform low rank and sparse constraints on the affinity matrix to improve the performance. However, it is just a trade-off between these two constraints. In this paper, we propose a novel Cascaded Low Rank and Sparse Representation (CLRSR) method for subspace clustering, which seeks the sparse expression on the former learned low rank latent representation. By this cascaded way, the sparse and low rank properties of the data are revealed adequately. Additionally, we extend CLRSR onto Grassmann manifolds to deal with multi-dimension data such as imageset or videos. An effective solution and its convergence analysis are also provided. The experimental results demonstrate the proposed method has excellent performance compared with state-of-the-art clustering methods.

1 Introduction

Unsupervised clustering is a fundamental topic in computer vision and artificial intelligence areas, which aims to group data with different patterns into different clusters by exploring the intrinsic membership of data. Recently, benefiting from the efficiency of analyzing complex relationship and structures hidden in data, spectral clustering method is considered having better prospects. Generally, in spectral clustering method, an affinity matrix is firstly learned from the given data, and then the affinity matrix is fed to a standard clustering algorithm such as NCut or K-means to obtain the final clustering results. The data representation and its affinity matrix have heavy impacts on the clustering results, so numerous efforts have been made to exploit better affinity matrices to reveal the intrinsic structure embedded in data. Sparse Subspace Clustering (SSC) [Elhamifar and Vidal, 2011] and Low Rank Representation (LRR) [Liu et al., 2013] are two representative methods. They respectively use sparse and low rank constraints on the affinity matrix of self-expression model.

*Corresponding author: Yongli Hu (huyongli@bjut.edu.cn)

In the view of subspace, SSC is regarded to reveal the “local” structure of the data, as the sparsity of a datum representation is resulted by the hypothesis that only the data representation coefficients corresponding to the data samples belonging to the same subspace are nonzero. LRR tries to formulate the “global” structure of data space via low rank constraint which is a holistic constraint to reflect the relationship among data. In the subspace clustering, the ideal affinity matrix need have diagonal block shape and each block can represent a subspace [Feng et al., 2014; Wu et al., 2016]. According to the conclusion from [Luxburg, 2007], the number of blocks is determined by the rank of the Laplacian matrix which is constructed by the affinity matrix; thus, LRR essentially describes the relationship among subspaces. Due to the above advantages, SSC and LRR and their varieties achieve state-of-the-art clustering performance.

With the success of SSC and LRR mentioned above, it is intuitive to combine the global and local structure constraints together to obtain further improvements. Several such combined models have been proposed in recent researches, such as Low Rank Sparse Subspace Clustering [Zhuang et al., 2012; Patel et al., 2015] and Laplacian regularized LRR [Yin et al., 2016]. These combined models which simultaneously enforce sparse and low rank constraints on the same affinity matrix have achieved some gains, but this combining strategy is not considered the best way to reveal the global and local structures of data. It is just a trade-off between the two penalties used in the current methods in general. However, the sparse and low rank properties of a matrix usually cannot be satisfied simultaneously, for example, the full rank matrix of identity matrix or diagonal matrix. From this observation, to fully reveal the global and local structures of the data, we propose a cascaded version of low rank and sparse subspace clustering methods, namely Cascaded Low Rank and Sparse Representation (CLRSR). In the proposed method, the low rank and sparse properties of data are successively mined by a global to local manner in two cascaded steps: First, the global structure of data is represented by LRR; Then the sparse representation of the former learned low rank representation is exploited to capture the local structure of data.

Another observation is that most of the current subspace clustering methods take the data as vectors with Euclidean...
distance as the metric. However, for many image sets, video
data or other multi-dimension data, the simple vectorized
data form and its metric are not applicable since it has been
proven that many high-dimension data are embedded in low-
dimension manifolds with nonlinear metric [Wang et al.,
2008]. So it is rational to model such multi-dimension data as
manifolds. For this purpose, we adopt the Grassmann mani-

2 Relevant Concepts

The contributions of this work are summarized as follows:

- Proposing a novel cascaded low rank and sparse rep-
  resentation model to fully capture the global and local
  structures underlying the observed data;

- Extending the CLRSR model onto Grassmann mani-
  folds which favors the representation of multi-dimension
  data; and

- Giving a practical solution to the optimization problem
  on manifold space for the proposed G-CLRSR model.

2.2 Grassmann Manifolds

Definition 1 (Grassmann Manifolds) [Absil et al., 2008]
The Grassmann manifold, denoted by $\mathcal{G}(p,n)$, consists of
all the $p$-dimensional subspaces in $n$-dimensional Euclidean
space $\mathbb{R}^n$ ($0 \leq p \leq n$).

To intuitively represent Grassmann manifolds, the popular
way is to construct Grassmann manifolds by

$$\mathcal{G}(p,n) = \{X \in \mathbb{R}^{n \times p} : X^T X = I_p\} / \mathcal{O}(p).$$

(2)

From this, we can see that a Grassmannian point actually is an
equivalent class (a subset) in which any two thin-tall orthogonal
matrices can be converted to each other applying a $p \times p$

2.2 Grassmann Manifolds

Definition 1 (Grassmann Manifolds) [Absil et al., 2008]
The Grassmann manifold, denoted by $\mathcal{G}(p,n)$, consists of
all the $p$-dimensional subspaces in $n$-dimensional Euclidean
space $\mathbb{R}^n$ ($0 \leq p \leq n$).

To intuitively represent Grassmann manifolds, the popular
way is to construct Grassmann manifolds by

$$\mathcal{G}(p,n) = \{X \in \mathbb{R}^{n \times p} : X^T X = I_p\} / \mathcal{O}(p).$$

(2)

From this, we can see that a Grassmannian point actually is an
equivalent class (a subset) in which any two thin-tall orthogonal
matrices can be converted to each other applying a $p \times p$

2.2 Grassmann Manifolds

Definition 1 (Grassmann Manifolds) [Absil et al., 2008]
The Grassmann manifold, denoted by $\mathcal{G}(p,n)$, consists of
all the $p$-dimensional subspaces in $n$-dimensional Euclidean
space $\mathbb{R}^n$ ($0 \leq p \leq n$).

To intuitively represent Grassmann manifolds, the popular
way is to construct Grassmann manifolds by

$$\mathcal{G}(p,n) = \{X \in \mathbb{R}^{n \times p} : X^T X = I_p\} / \mathcal{O}(p).$$

(2)

From this, we can see that a Grassmannian point actually is an
equivalent class (a subset) in which any two thin-tall orthogonal
matrices can be converted to each other applying a $p \times p$

2.2 Grassmann Manifolds

Definition 1 (Grassmann Manifolds) [Absil et al., 2008]
The Grassmann manifold, denoted by $\mathcal{G}(p,n)$, consists of
all the $p$-dimensional subspaces in $n$-dimensional Euclidean
space $\mathbb{R}^n$ ($0 \leq p \leq n$).

To intuitively represent Grassmann manifolds, the popular
way is to construct Grassmann manifolds by

$$\mathcal{G}(p,n) = \{X \in \mathbb{R}^{n \times p} : X^T X = I_p\} / \mathcal{O}(p).$$

(2)

From this, we can see that a Grassmannian point actually is an
equivalent class (a subset) in which any two thin-tall orthogonal
matrices can be converted to each other applying a $p \times p$

2.2 Grassmann Manifolds

Definition 1 (Grassmann Manifolds) [Absil et al., 2008]
The Grassmann manifold, denoted by $\mathcal{G}(p,n)$, consists of
all the $p$-dimensional subspaces in $n$-dimensional Euclidean
space $\mathbb{R}^n$ ($0 \leq p \leq n$).

To intuitively represent Grassmann manifolds, the popular
way is to construct Grassmann manifolds by

$$\mathcal{G}(p,n) = \{X \in \mathbb{R}^{n \times p} : X^T X = I_p\} / \mathcal{O}(p).$$

(2)

From this, we can see that a Grassmannian point actually is an
equivalent class (a subset) in which any two thin-tall orthogonal
matrices can be converted to each other applying a $p \times p$

2.2 Grassmann Manifolds

Definition 1 (Grassmann Manifolds) [Absil et al., 2008]
The Grassmann manifold, denoted by $\mathcal{G}(p,n)$, consists of
all the $p$-dimensional subspaces in $n$-dimensional Euclidean
space $\mathbb{R}^n$ ($0 \leq p \leq n$).

To intuitively represent Grassmann manifolds, the popular
way is to construct Grassmann manifolds by

$$\mathcal{G}(p,n) = \{X \in \mathbb{R}^{n \times p} : X^T X = I_p\} / \mathcal{O}(p).$$

(2)

From this, we can see that a Grassmannian point actually is an
equivalent class (a subset) in which any two thin-tall orthogonal
matrices can be converted to each other applying a $p \times p$

2.2 Grassmann Manifolds

Definition 1 (Grassmann Manifolds) [Absil et al., 2008]
The Grassmann manifold, denoted by $\mathcal{G}(p,n)$, consists of
all the $p$-dimensional subspaces in $n$-dimensional Euclidean
space $\mathbb{R}^n$ ($0 \leq p \leq n$).

To intuitively represent Grassmann manifolds, the popular
way is to construct Grassmann manifolds by

$$\mathcal{G}(p,n) = \{X \in \mathbb{R}^{n \times p} : X^T X = I_p\} / \mathcal{O}(p).$$

(2)

From this, we can see that a Grassmannian point actually is an
equivalent class (a subset) in which any two thin-tall orthogonal
matrices can be converted to each other applying a $p \times p$

2.2 Grassmann Manifolds

Definition 1 (Grassmann Manifolds) [Absil et al., 2008]
The Grassmann manifold, denoted by $\mathcal{G}(p,n)$, consists of
all the $p$-dimensional subspaces in $n$-dimensional Euclidean
space $\mathbb{R}^n$ ($0 \leq p \leq n$).

To intuitively represent Grassmann manifolds, the popular
way is to construct Grassmann manifolds by

$$\mathcal{G}(p,n) = \{X \in \mathbb{R}^{n \times p} : X^T X = I_p\} / \mathcal{O}(p).$$

(2)

From this, we can see that a Grassmannian point actually is an
equivalent class (a subset) in which any two thin-tall orthogonal
matrices can be converted to each other applying a $p \times p$

2.2 Grassmann Manifolds

Definition 1 (Grassmann Manifolds) [Absil et al., 2008]
The Grassmann manifold, denoted by $\mathcal{G}(p,n)$, consists of
all the $p$-dimensional subspaces in $n$-dimensional Euclidean
space $\mathbb{R}^n$ ($0 \leq p \leq n$).

To intuitively represent Grassmann manifolds, the popular
way is to construct Grassmann manifolds by

$$\mathcal{G}(p,n) = \{X \in \mathbb{R}^{n \times p} : X^T X = I_p\} / \mathcal{O}(p).$$

(2)

From this, we can see that a Grassmannian point actually is an
equivalent class (a subset) in which any two thin-tall orthogonal
matrices can be converted to each other applying a $p \times p$

2.2 Grassmann Manifolds

Definition 1 (Grassmann Manifolds) [Absil et al., 2008]
The Grassmann manifold, denoted by $\mathcal{G}(p,n)$, consists of
all the $p$-dimensional subspaces in $n$-dimensional Euclidean
space $\mathbb{R}^n$ ($0 \leq p \leq n$).

To intuitively represent Grassmann manifolds, the popular
way is to construct Grassmann manifolds by

$$\mathcal{G}(p,n) = \{X \in \mathbb{R}^{n \times p} : X^T X = I_p\} / \mathcal{O}(p).$$

(2)

From this, we can see that a Grassmannian point actually is an
equivalent class (a subset) in which any two thin-tall orthogonal
matrices can be converted to each other applying a $p \times p$

2.2 Grassmann Manifolds

Definition 1 (Grassmann Manifolds) [Absil et al., 2008]
The Grassmann manifold, denoted by $\mathcal{G}(p,n)$, consists of
all the $p$-dimensional subspaces in $n$-dimensional Euclidean
space $\mathbb{R}^n$ ($0 \leq p \leq n$).

To intuitively represent Grassmann manifolds, the popular
way is to construct Grassmann manifolds by

$$\mathcal{G}(p,n) = \{X \in \mathbb{R}^{n \times p} : X^T X = I_p\} / \mathcal{O}(p).$$

(2)

From this, we can see that a Grassmannian point actually is an
equivalent class (a subset) in which any two thin-tall orthogonal
matrices can be converted to each other applying a $p \times p$
There exist different types of geometries on Grassmann manifold, please refer to [Absil et al., 2008].

3 Cascaded Low Rank and Sparse Representation on Grassmann Manifolds

In this Section, we will elaborate a cascaded version of low rank and sparse representation model for clustering.

3.1 CLRSR on Grassmann Manifolds

To explore the global structure, we enforce the low rank constraint on the self-expression coefficients matrix of data. Then to fulfill the local properties, the low rank representation of data is cascaded with the sparse representation. So we have

\[
\min_{Z, E_1, E_2} \text{rank}(Z) + \lambda_1 \|E_1\|_F^2 + \lambda_2 \|C\|_1 + \lambda_3 \|E_2\|_F^2, \\
\text{s.t. } Y = YZ + E_1, \quad Z = ZC + E_2, \quad \text{diag}(C) = 0,
\]

(5)

where the singular value thresholding operator is defined as \( S(x) = \text{sign}(x) \max(\|x\|; 0) \) where \( \lambda \) is a positive number.

Following the notation used in [Wang et al., 2014], we firstly simplify the tensorial error term \( E \) in formula (9) to avoid a complex computation between tensor and matrix,

\[
\|E\|_F^2 = \sum_{i=1}^N \|E(:,:,i)\|_F^2,
\]

where \( E(:,:,i) = X_iX_i^T - \sum_{j=1}^N z_{ij}(X_jX_j^T) \) is the \( i \)-th front slice of \( E \). For convenience, we denote \( \Delta_{ij} = \text{tr}((X_jX_j^T)(X_iX_i^T)) \) and the \( N \times N \) symmetric matrix \( \Delta = (\Delta_{ij})_{i,j=1}^N \). After a sequence of simple mathematical manipulations, we can show that problem (9) is equivalent to

\[
\min_{Z, C, E} \|Z\|_* + \lambda_1 \|E_1\|_F^2 + \lambda_2 \|C\|_1 + \lambda_3 \|E_2\|_F^2, \\
\text{s.t. } Z = ZC + E, \quad \text{diag}(C) = 0,
\]

(6)

To handle multi-dimension data, we consider the generalization of model (6) onto Grassmann manifolds. Given a set of Grassmannian points \( X = \{X_1, X_2, \ldots, X_N\} \) where \( X_i \in \mathcal{G}(p, d) \), the parallelized model is given by

\[
\min_{Z, C, E} \|Z\|_* + \lambda_1 \sum_{i=1}^N \|X_i \odot (\bigodot_{j=1}^N z_{ijj} \odot X_j)\|_G + \lambda_2 \|C\|_1 + \lambda_3 \|E\|_F^2,
\]

\[
\text{s.t. } Z = ZC + E, \quad \text{diag}(C) = 0,
\]

(7)

where we use abstract symbols \( \odot, \bigodot_{j=1}^N \) and \( \odot \) to simulate the “linear” operations to be defined on manifolds, i.e., addition, subtraction and scalar-multiplication. \( \|X_i \odot (\bigodot_{j=1}^N z_{ijj} \odot X_j)\|_G \) measures the error between the Grassmannian point \( X_i \) and its “reconstruction” \( \bigodot_{j=1}^N z_{ijj} \odot X_j \). According to the property of Grassmann manifolds in (3), it is easy to use the embedding distance to replace the manifold distance as,

\[
\|X_i \odot (\bigodot_{j=1}^N z_{ijj} \odot X_j)\|_G = d^2_G(X_i, \bigodot_{j=1}^N z_{ijj} \odot X_j) = \|X - X' \times_3 Z\|_F^2,
\]

(8)

where \( X' = \{X_1X_1^T, X_2X_2^T, \ldots, X_NX_N^T\} \) is a 3-order tensor and \( \times_3 \) is the mode-3 tensor-vector multiplication [Kolda and Bader, 2009]. This error measurement not only avoids using Log map operator but also has simple computation with F-norm. Finally, the CLRSR model on Grassmann manifolds can be defined as follows,

\[
\min_{Z, E, C, E} \|Z\|_* + \lambda_1 \|E\|_F^2 + \lambda_2 \|C\|_1 + \lambda_3 \|E\|_F^2, \\
\text{s.t. } X' = X \times_3 Z + E, \quad Z = ZC + E, \quad \text{diag}(C) = 0,
\]

(9)

where the reconstructed error \( E \) is also a 3-order tensor. This model is named as G-CLRSR.

3.2 Optimization

We use Alternating Direction Method [Lin et al., 2011] to solve such a convex problem. For this purpose, we introduce two auxiliary variables \( J = Z \) and \( Q = C \) to separate variables \( Z \) and \( Q \), and reformulate the optimization problem as

\[
\min_{J, Z, Q, N} \|J\|_* + \lambda_1 \|J\|_* + \lambda_2 \|Q\|_1 + \lambda_3 \|Z - ZC\|_F^2, \\
\text{s.t. } J = Z, \quad Q = C,
\]

(10)

which can be solved by minimizing its ALM formula

\[
\mathcal{L} = \|J\|_* + \lambda_1 \|J\|_* + \lambda_2 \|Q\|_1 + \lambda_3 \|Z - ZC\|_F^2 + \lambda_2 \|Q\|_1 + \frac{\mu}{2} \|J - Z - \frac{A}{\mu} \|_F^2 + \|Q - C - \frac{B}{\mu} \|_F^2
\]

(12)

where \( A \) and \( B \) are the Lagrange multipliers, and \( \mu \) is the penalty parameter. The above ALM is appealing on if we can find closed form solutions to the following subproblems (13), (14), (15) and (16).

Updating \( J \). Fixing other variables, we update \( J \) by,

\[
J^{k+1} = \arg \min_{J} \mathcal{L}(J^k, Z^k, Q^k, C^k) = \arg \min_{J} \|J\|_* + \frac{\mu}{2} \|J - (Z + A)\|_F^2
\]

(13)

A closed-form solution is suggested as the following theorem [Cai et al., 2010],

Lemma 1 Given that \( UDV^T = \text{SVD}(Z + \frac{A}{\mu}) \) as defined above, the solution is given by

\[
J^* = US_{\mu^{-1}}(D)V^T
\]

where \( D \) is the diagonal matrix of singular values. The singular value thresholding operator is defined as \( S_{\tau}(x) = \text{sign}(x) \cdot \max(|x| - \tau, 0) \) where \( \tau \) is a positive number.
Updating $Z$. To update $Z$, we solve the following problem by fixing other variables,

$$ Z^{k+1} = \arg \min_{Z} \mathcal{L}(J^{k+1}, Z^{k}, Q^{k}, C^{k}) $$

$$ = \arg \min_{Z} \lambda_1 \left(-2 \text{tr}(Z \Delta) + \text{tr}(Z \Delta Z^T)\right) $$

$$ + \lambda_3 \|Z - ZC\|_F^2 + \frac{\mu}{2} \|J - Z - \frac{A}{\mu}\|_F^2. $$

This is a quadratic optimization problem with respect to $Z$. We set its derivation as zero and get the closed solution as,

$$ Z^* = (2 \lambda_1 \Delta - A + \mu J)(2 \lambda_1 \Delta + 2 \lambda_3 - 2 \lambda_3 C - 2 \lambda_3 C^T - 2 \lambda_3 CC^T + \mu I)^{-1}. $$

Updating $Q$. By fixing other variables, we update $Q$ through,

$$ Q^{k+1} = \arg \min_{Q} \mathcal{L}(J^{k+1}, Z^{k+1}, Q^{k}, C^{k}) $$

$$ = \arg \min_{Q} \lambda_2 \|Q\|_1 + \frac{\mu}{2} \|Q - (C + \frac{B}{\mu})\|_F^2. $$

We know that the problem in (15) has a closed-form solution, given by [Beck and Teboulle, 2009],

**Lemma 2** Let matrix $M = C + \frac{B}{\mu}$ and a positive number $\tau$, a closed-form global optimal solution of above problem is

$$ Q^* = \mathcal{S}_{\tau\mu}(M) $$

where the singular value thresholding operator is also defined as $\mathcal{S}_x = \text{sign}(x) \cdot \max(|x| - \tau, 0)$.

Updating $C$. The final self-expression $C$ is updated by solving the following problem,

$$ C^{k+1} = \arg \min_{C} \mathcal{L}(J^{k+1}, Z^{k+1}, Q^{k+1}, C^{k}) $$

$$ = \arg \min_{C} \lambda_3 \|Z - ZC\|_F^2 + \frac{\mu}{2} \|Q - C - \frac{B}{\mu}\|_F^2. $$

Similar to update the subproblem $Z$, we also set the derivation of (16) to zero and get the closed-form solution,

$$ C^* = (2 \lambda_3 Z^T Z + \mu I)^{-1}(2 \lambda_3 Z^T Z - B + \mu Q) $$

Updating $A$ and $B$. After we solving the above subproblems on $J$, $Z$, $Q$ and $C$, respectively, we can easily update these two linear Lagrangian multipliers by

$$ \begin{cases} A^{k+1} = A^k + \mu_k (Z^{k+1} - J^{k+1}) \\ B^{k+1} = B^k + \mu_k (C^{k+1} - Q^{k+1}) \end{cases} $$

Adapting penalty parameter $\mu$. For the penalty parameter $\mu > 0$, we could update it by

$$ \mu^{k+1} = \min(\rho \mu^k, \mu_{\text{max}}) $$

where $\mu_{\text{max}}$ is the pre-defined upper bound of $\mu^k$.

Once obtaining the coefficient matrix $C$, any clustering algorithm, i.e., NCut used in this paper, can be performed on affinity matrix $W = (|C| + |C|^T)/2$ to receive final clustering results. We summarize pseudo code in Algorithm 1.

**Algorithm 1** The G-CLRSR Optimizing Problem.

**Input:** Grassmannian points $\{X_i\}_{i=1}^N$, $X_i \in \mathcal{G}(p, d)$, the balancing penalty parameters $\lambda_1$, $\lambda_2$ and $\lambda_3$.

**Output:** The coefficient matrix $C$.

1: Initialize: $J = Z = N = Q = 0$, $A = B = 0$, $\rho = 1.1$, $\epsilon = 10^{-8}$, $\mu_0 = 10^{-8}$ and $\mu_{\text{max}} = 10^6$.

2: ## Calculating $\Delta$

3: for $i=1:N$ do

4: for $j=1:N$ do

5: $\Delta_{ij} \leftarrow \text{tr}((X_j X_j^T)(X_i^T X_i))$

6: end for

7: end for

8: ## Optimizing variables

9: while not converged do

10: Updating variable $J$, $Z$, $Q$, $C$ via (13-16), respectively.

11: Updating multipliers $A$ and $B$ via (17).

12: Updating parameter $\mu$ by $\mu = \min(\rho \mu, \mu_{\text{max}})$.

13: Checking the convergence conditions:

$$ \|Z - J\|_{\infty} < \epsilon \text{ and } \|C - Q\|_{\infty} < \epsilon. $$

14: end while

**Figure 2:** Convergence curve of G-CLRSR. The value of Termination Criterion is $\min(\|Z - J\|_{\infty}, \|Q - C\|_{\infty})$ in each iteration.

### 3.3 Complexity and Convergence Analysis

As shown in Algorithm 1, the major time-consuming components lie in steps 5, 10, 11, and 13. (i) The complexity of step 5, the computation of the symmetric matrix $\Delta$, is $O(p^2 d + p^3)$ as its main complexity comes from 4 matrices multiplication; (ii) Step 10 takes a SVD operation for updating $J$, leading to a complexity of $O(N^3)$; and (iii) Both steps 11 and 13 involve matrix multiplication and inversion operations, thus the complexity is $O(N^3)$. Overall, the total complexity is $O(N^2 p^2 d + d^3 + N^3)$ for each iteration. Under the condition $p \ll d$, the total complexity is basically $O(N^2 p^2 d + k N^3)$ where $k$ denotes the number of iterations.

To experimentally show the convergence behavior of the proposed method, we provide an intuitive curve illustration of the convergence with respect to the iteration number on all datasets, shown in Figure 2. Our proposed algorithm converges in about 230 steps, which reflects that our optimization method has a good convergence property.
4 Experiments

4.1 Datasets

Extended Yale B dataset is collected from 38 subjects and each subject has 64 front face images. We randomly choose 8 images from the same subject to construct an imageset. CMUPIE\(^1\) dataset contains totally 1632 front face images evenly from 68 subjects. Every 4 images from the same subject are chosen to construct an imageset. UCF sport dataset\(^2\) includes a total of 150 video clips with 13 actions, which has a wide range of scenes and viewpoints. Each clip has 22 to 144 frames.

SKIG gesture dataset\(^3\) contains 1080 RGB-D video clips with 10 gesture types of 6 subjects. All the gestures are performed under different backgrounds and illumination. Each sequence contains 63 to 605 frames.

Highway traffic dataset\(^4\) contains 253 video clips labeled with three levels. Each clip has 42 to 52 frames.

Each video clip is regarded as an imageset to construct a point on Grassmann manifolds.

4.2 Comparison Algorithms and Settings

For the comparison algorithms, we have the following (i) vector-form input methods, LRR [Liu et al., 2013], SSC [Vidal, 2011], LS3C [Patel et al., 2013] and CLRSR, (ii) Grassmann-form input methods, SCGSM [Turaga et al., 2011], SLRR [Yin et al., 2015], FGLRR [Wang et al., 2014], LapFGLRR [Wang et al., 2018] and a baseline G-LRSR.

To execute LRR, SSC, LS3C and LRSR on imageset and video data, we have to vectorize each imageset as inputs for them. However, in most experiments, we cannot simply take these long vectors into the algorithms due to the high dimension for a larger imageset. So, we reduce the dimension of these vectors to a low one which equals to the dimension of PCA components retaining 95% of its variance energy.

SLRR extends LRR onto Stiefel manifolds which has the same form with Grassmann manifolds. SCGSM and FGLRR can be considered as the Grassmann manifolds version of K-means and LRR models. In LapFGLRR, a local constraint among data, Laplacian regularization, is introduced into the FGLRR model. In G-LRSR, the LRSR model is embedded into Grassmann manifolds as a important baseline.

For a fair comparison, we execute the code of compared methods provided by authors. The parameters \(\lambda_1, \lambda_2, \lambda_3\) and \(\lambda_3\) in CLRSR and G-CLRSR, one parameter \(\lambda_1\) in LRR, SSC, FGLRR and SLRR, two parameters \(\lambda_1\) and \(\lambda_2\) in LapFGLRR, G-LRSR and LS3C, need to be tuned, respectively. We tune their values within the set of \(\{0.1, 0.2, \cdots, 1, 2, \cdots, 10\}\), and we report the best value in each experiment.

To make a comprehensive evaluation, we employ five clustering validation measurements, including ACC (accuracy), NMI (normalized mutual information), RI (rand index), Purity and FM (F-measure). For all the metrics, higher values indicate better performance.

\(^1\)http://www.cs.cmu.edu/afs/cs/project/PIE/MultiPie/MultiPie/Home.html
\(^2\)http://crcv.ucf.edu/data/
\(^3\)http://lshao.staff.shef.ac.uk/data/SheffieldKinectGesture.htm
\(^4\)http://www.svcl.ucsf.edu/projects/traffic/

4.3 Data Representation

An imageset \(F = \{f_1, \cdots, f_{2^d}\}; f_i \in R^d\), with \(f_i\) being the feature descriptors of \(i\)-th frame. In this paper, we just use the image intensity as the raw feature. The Grassmannian point can be represented by an orthogonal basis. Similar to the work in [Wang et al., 2014], the feature matrix \(F \in R^{d \times P}\) is decomposed by SVD as \(F = U \Sigma V^T\); then we pick up the first \(p\) \((p \leq P)\) singular-vectors of \(U\) to represent a point \(X = [u_1, u_2, \cdots, u_p]\) on Grassmann manifolds \(G(p, d)\).

4.4 Experimental Results

Image Set Data Clustering

Although facial recognition and clustering has been a mature application in computer vision, it often gets unfavorable experimental results since the varieties of facial expressions, poses or light conditions. Imageset data collect complementary information from multiple views of each object, which can naturally overcome the environment affection. Here, we regard each imageset as one point on Grassmann manifolds.

Table 1 reports all clustering results in terms of different validation criteria on two facial imageset datasets. From an overall perspective, manifold representation based methods obtain better performance than vector-form input clustering methods, i.e., LRR, SSC and LS3C, which demonstrates that manifold representation is a good tool for imageset data. Compared with SLRR, SCGSM and FGLRR, our proposed method simultaneously considers meaningful global and neighbor structures, which makes our proposed method more robust in clustering tasks. As for the important baselines, i.e., LapFGLRR and G-LRSR, they both involve global and local constraints on only one layer of representation, while our proposed approach can extract meaningful representation layer by layer. Through keeping the local neighbor relation on the base of global structure information, our proposed method receives the best experimental results.

Video Data Clustering

Action recognition is another open and hot issue in computer vision literature. Most action data are in the form of video clips, which is a set of sequential frame images and suitable to our manifold representation methods. So we select a human action and a gesture video datasets.

Table 2 lists the results of different clustering methods on video datasets. These datasets are much more challenging than the previous two facial imageset datasets, as the environment is more complex and varies dramatically. Obviously, for video datasets, manifold representation based methods reflect more impressive superior to vector-input based methods, which is further to verify the advantage of manifolds representation. What is more, LapFGLRR achieves the best performance on SKIG in terms of RI. However, the performance on the other datasets are not such promising.

In addition, we wish to evaluate our proposed method in a real and practical application; therefore, a natural scenario
dataset, Highway Traffic videos, is chosen as test data. Table 3 shows the clustering results of all the algorithms, our proposed method still get satisfied performance, i.e., almost 4 percent raising performance in ACC. For the general application of G-CLRSR, we can use it to learn the threshold of different level or part the period for the traffic jam on specific roads based on the historical traffic data.

<table>
<thead>
<tr>
<th>Method</th>
<th>ACC</th>
<th>NMI</th>
<th>RI</th>
<th>Purity</th>
<th>FM</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRR</td>
<td>67.59</td>
<td>35.75</td>
<td>68.50</td>
<td>69.96</td>
<td>50.60</td>
</tr>
<tr>
<td>SSC</td>
<td>60.47</td>
<td>57.10</td>
<td>73.36</td>
<td>81.42</td>
<td>46.24</td>
</tr>
<tr>
<td>LS3C</td>
<td>70.36</td>
<td>26.34</td>
<td>66.71</td>
<td>71.15</td>
<td>46.45</td>
</tr>
<tr>
<td>CLRSR</td>
<td>66.80</td>
<td>12.73</td>
<td>60.06</td>
<td>67.98</td>
<td>36.36</td>
</tr>
</tbody>
</table>

5 Conclusion

In this paper, we proposed a novel cascaded low rank and sparse representation clustering method and extended it onto Grassmann manifolds to handle multi-dimension or imageset data. Compared with existing works which integrate low rank and sparse constraints on the same data representation simultaneously, the proposed methods execute a sparse constraint on the low-rank representation matrix in a cascaded manner which contributes to obtaining an improved affinity matrix. In addition, an efficient alternative based algorithm is proposed to solve for the optimal solution. Intensive experiments were conducted on imagesets and video datasets to demonstrate the superior performance of our methods.

Acknowledgements

The research project is supported by the Australian Research Council (ARC) through the grant DP140102270 and also partially supported by National Natural Science Foundation of China under Grant No. 61390510, 61672071, 61772048, Beijing Natural Science Foundation No. 4172003, 4172004, 4184082, Beijing Municipal Science and Technology Project No. Z171100000517003, Z17110000517004, Z171100004417023, Z16110001116072.
References


