Robust Subspace Clustering via Half-Quadratic Minimization

Yingya Zhang, Zhenan Sun, Ran He, and Tieniu Tan
Center for Research on Intelligent Perception and Computing,
National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, Beijing, China
{zhangyingya, znsun, rhe, tnt}@nlpr.ia.ac.cn

Abstract

Subspace clustering has important and wide applications in computer vision and pattern recognition. It is a challenging task to learn low-dimensional subspace structures due to the possible errors (e.g., noise and corruptions) existing in high-dimensional data. Recent subspace clustering methods usually assume a sparse representation of corrupted errors and correct the errors iteratively. However large corruptions in real-world applications can not be well addressed by these methods. A novel optimization model for robust subspace clustering is proposed in this paper. The objective function of our model mainly includes two parts. The first part aims to achieve a sparse representation of each high-dimensional data point with other data points. The second part aims to maximize the correntropy between a given data point and its low-dimensional representation with other points. Correntropy is a robust measure so that the influence of large corruptions on subspace clustering can be greatly suppressed. An extension of our method with explicit introduction of representation error terms into the model is also proposed. Half-quadratic minimization is provided as an efficient solution to the proposed robust subspace clustering formulations. Experimental results on Hopkins 155 dataset and Extended Yale Database B demonstrate that our method outperforms state-of-the-art subspace clustering methods.

1. Introduction

It is desirable to achieve a low-dimensional representation of the complex and redundant high-dimensional data in the era of big data. The conventional solution is to project the data points into a single low-dimensional subspace [2][8][9]. However, the data points may be drawn from a union of multiple subspaces in practical applications. Therefore the problem of *subspace clustering (or segmentation)* is proposed to divide high-dimensional data points into multiple subspaces, and find a low-dimensional subspace into which each group of data points can fit simultaneously [24].

Subspace clustering has attracted a great attention due to its promising applications in computer vision and machine learning. The large number of subspace clustering methods proposed in the literature can be classified into four categories: iterative methods, algebraic methods, statistical methods, and spectral clustering-based methods [24].

Iterative methods, such as K-subspaces [23], first assign data to pre-defined multiple subspaces, then update the subspaces and reassign each data point to the nearest subspace. Repeating these two steps iteratively to convergence, we can obtain the segmentation result. The disadvantage of these methods is that they need to know the number of the subspaces and their dimensions in advance. Generalized Principal Component Analysis (GPCA) [25] uses an algebraic way to model and segment the data. This method fits the data with a polynomial, and the gradient of the polynomial at a point gives the normal vector that the point belongs to. However, this method is sensitive to noise and outliers, and as the data dimension increases, its computational complexity grows exponentially. Statistical approaches, such as Mixture of Probabilistic PCA (MPPCA) [21] and Multi-Stage Learning (MSL) [20], assume that the data are drawn from a mixture of probabilistic distributions. Then the Expectation Maximization (EM) technique is used to estimate the subspaces and cluster data iteratively. Although these methods have achieved good performance under constrained conditions, they are sensitive to noise and outliers in real-world applications.

Recently spectral clustering-based methods have drawn much attention, which assume that a data point can be represented as a combination of other data points in the same subspace. Then representational coefficients are used to construct the affinity matrix, and the spectral clustering algorithms are applied to obtain correct segmentation. Elhamifar and Vidal [5] introduced the sparse representation technique in Compressed Sensing (CS) literature to subspace clustering and proposed the Sparse Subspace Clustering (SSC) algorithm. SSC aims to find the sparsest representation of each data point by using the l_1 norm regularization on the coefficient matrix. Low-Rank Representation (LRR) [14][13], in another way, seeks to find the lowest-rank representation of all data points by using trace norm, which can capture the global structures of the data. In [17], Lu *et al*. first proved the Enforced Block Diagonal (EBD) conditions, and then proposed the Least Square Regression (LSR) method based on l_2 norm regularization.

The main difference among these methods is the regularization of the coefficient matrix. However, the main challenge of subspace clustering is to handle the errors (*e.g.* random noise and large corruptions) existing in data [13], which may lead to poor subspace clustering results due to the large weight of errors in optimization. Despite the variety of the regularization, most of these methods assume that the errors have a sparse representation, and correct the errors iteratively. However, large corruptions in real-world problems can not be well addressed by these error correction methods.

This paper aims to propose a novel subspace clustering method which is robust against large corruptions. Our basic idea is to minimize the influence of error data points on subspace clustering based on a robust measure of the similarity between data points and their subspace representations, correntropy [16]. The optimization model of the proposed robust subspace clustering method aims to achieve sparse representation coefficients and minimal reconstruction errors simultaneously. An efficient iterative solution to the proposed problem based on half-quadratic (HQ) optimization is provided. In each iteration, the complex optimization problems are simplified to quadratic problems that have a closed form solution in half-quadratic optimization. The performance of our method is evaluated and compared with state-of-the-art subspace clustering methods on motion segmentation and face clustering problems.

2. Related work

To better illustrate the main idea of our method in the context of sparse subspace clustering (SSC) [5], the algorithm of SSC is described as follows.

2.1. Notations

Given a $D \times N$ data matrix X consisting of N vectors $\{\boldsymbol{x}_i \in \mathbb{R}^D\}_{i=1}^N$, which are drawn from a union of linear or affine subspaces $\{S_i\}_{i=1}^K$ of unknown dimensions $d_k = \dim(S_k)$, $0 < d_k < D$, the task of subspace clustering is to find the number of subspaces K, their dimensions $\{d_k\}_{k=1}^K$ and segment the data vectors $\boldsymbol{x}_i$ into these subspaces. $X_{\hat{i}} \in \mathbb{R}^{D \times N}$ stands for the matrix obtained from X by replacing its i-th column $\boldsymbol{x}_i$ with the vector $\mathbf{0} \in \mathbb{R}^D$ of all zeros. Given a vector $\boldsymbol{z} \in \mathbb{R}^p$, let $diag(\boldsymbol{z})$ be the diagonal $p \times p$ matrix whose i-th main diagonal element is the i-th entry of $\boldsymbol{z}$. The matrix I stands for the identity matrix and $\mathbf{1}$ for a vector of all 1s. The j-th entry of the data point $\boldsymbol{x}_i$ is denoted by $(\boldsymbol{x}_i)_j$.

2.2. Sparse Subspace Clustering

The idea of Sparse Subspace Clustering comes from the assumption that each data point sampled from a union of subspaces can be written as a linear combination of other points in the dataset, which is defined as self-expressiveness property in [5]. To be more precise, for a data point $\boldsymbol{x}_i$, we want to find a coefficient vector $\boldsymbol{c}_i$ that satisfies,

$$\boldsymbol{x}_i = X_{\hat{i}}\boldsymbol{c}_i \tag{1}$$

In practice, this is an ill-posed problem since the solution of (1) is not unique in general. A possible technique of this problem is to find the sparsest representation of $\boldsymbol{x}_i$. That is, for data point $\boldsymbol{x}_i$, one can minimize the number of nonzero entries of the coefficient vector $\boldsymbol{c}_i$,

$$\min_{\boldsymbol{c}_i} \|\boldsymbol{c}_i\|_1 \quad s.t. \quad \boldsymbol{x}_i = X_{\hat{i}}\boldsymbol{c}_i \tag{2}$$

where $\|\boldsymbol{c}_i\|_1 = \sum_{j=1}^N |\boldsymbol{c}_{ij}|$, which is the relaxation of the l_0 norm.

Considering the case that the data points may be contaminated with some dense noise, a natural way is to extend the original problem (2) to the following form by relaxing the equality constraint and adding a penalty to cost,

$$\min_{\boldsymbol{c}_i} \|\boldsymbol{c}_i\|_1 + \gamma \|\boldsymbol{x}_i - X_{\hat{i}}\boldsymbol{c}_i\|_2^2 \tag{3}$$

The l_2 norm term $\|\boldsymbol{x}_i - X_{\hat{i}}\boldsymbol{c}_i\|_2^2$ measures the fidelity of sparse representation.

In more practical and general scenarios, the data points may be contaminated by both large corruptions and small dense noise. SSC can be extended as the following model using the technique in robust face recognition [26],

$$\min_{\boldsymbol{c}_i, \boldsymbol{e}_i} \|\boldsymbol{c}_i\|_1 + \lambda\|\boldsymbol{e}_i\|_1 + \gamma\|\boldsymbol{x}_i - X_{\hat{i}}\boldsymbol{c}_i - \boldsymbol{e}_i\|_2^2 \tag{4}$$

where the term $\boldsymbol{e}_i \in \mathbb{R}^D$ models large corruptions which have sparse nonzero entries. This is the error correction method in sparse representation [11].

3. Proposed method

3.1. Correntropy

In order to deal with non-Gaussian noise and impulsive noise in signal processing, the concept of Correntropy was proposed in [16]. According to Renyi's quadratic entropy, the correntropy of two arbitrary variables $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined as

$$V_\sigma(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{E}[k_\sigma(\boldsymbol{x} - \boldsymbol{y})] \tag{5}$$

where $k_\sigma(.)$ is a kernel function. And it is used to measure the generalized similarity between $\boldsymbol{x}$ and $\boldsymbol{y}$.

In practice, sometimes the joint probability density function may be unknown, and only a finite number of data $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$ are available. Therefore, a sample estimator of correntropy is introduced,

$$V_\sigma(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{N} \sum_{i=1}^{N} k_\sigma(x_i - y_i) \tag{6}$$

In this paper, we use the Gaussian kernel $g(x) = \exp(-x^2/\sigma^2)$ where σ is the kernel size.

Liu *et al.* [16] further proposed a metric for any two vectors in the sample space which is named as Correntropy Induced Metric (CIM) on the basis of (6). It is defined as follows,

$$\begin{aligned} CIM(\boldsymbol{x}, \boldsymbol{y}) &= \{k(0) - V(\boldsymbol{x}, \boldsymbol{y})\}^{1/2} \\ &= (g(0) - \frac{1}{N} \sum_{i=1}^{N} g(x_i - y_i))^{1/2} \end{aligned} \tag{7}$$

It should be noted that CIM is a decreasing function of correntropy so maximization of correntropy is equally minimization of CIM. Compared to the global metric mean square error (MSE), the correntropy is a local metric. That means, the value of correntropy is mainly decided by the kernel function along the line $\boldsymbol{x} = \boldsymbol{y}$ [16]. Moreover, it becomes practical to choose an appropriate kernel size for correntropy because of the close relationship between correntropy and M-estimators [16].

3.2. Proposed formulations

A main problem of the existing subspace clustering methods is how to measure the reconstruction fidelity of outlier data points using a robust function. So correntropy is introduced as a robust measure in regularization terms of subspace clustering. A novel formulation of robust subspace clustering is proposed as follows:

$$\min_{\boldsymbol{c}_i} \sum_{j=1}^{N} \phi_r((\boldsymbol{c}_i)_j) + \gamma \sum_{j=1}^{D} \phi_c((\boldsymbol{x}_i - X_{\hat{i}} \boldsymbol{c}_i)_j) \tag{8}$$

where $\phi_c(x) = (1 - \exp(-x^2/\sigma^2))$ and $\phi_r(x) = \sqrt{x^2 + \alpha}$, which is called l_1-l_2 loss function in M-estimation. γ is a positive scalar.

To be more specific, the first term in (8) controls the sparsity of the coefficient vector $\boldsymbol{c}_i$. Although l_1 loss is convex, it is non-differentiable at zero-point. The optimization methods for l_1 regularization often oscillate around the true optimum and slowly converge towards the optimum. Compared to l_1 loss, l_1-l_2 loss around the zero-point is differentiable and can be efficiently solved. Specially, when $\alpha \to 0$, the l_1-l_2 loss turns to l_1 loss. And the second term is the squared CIM, which is a robust measure of the reconstruction fidelity of high-dimensional data points with other data. The parameter γ is used to keep a balance between reconstruction fidelity and coefficient sparsity.

Because of the property of correntropy, the problem (8) treats the representation of each entry of $\boldsymbol{x}_i$ differently. For example, if there exists noise or corruptions in $\boldsymbol{x}_i$, those entries in $\boldsymbol{x}_i$ only have limited influence to the correntropy. So the data is weighted in a robust way depending on the residuals. Compared with most existing subspace clustering methods, our method can achieve a more robust solution because the subspace clustering optimization is mainly determined by the uncorrupted data entries and the errors caused by noise or corruptions in data are greatly suppressed.

In practice, the errors in data are usually unpredictable. An extension of our model by explicitly modeling error terms is proposed as follows:

$$\begin{aligned} \min_{\boldsymbol{c}_i, \boldsymbol{e}_i} \sum_{j=1}^{N} \phi_r((\boldsymbol{c}_i)_j) &+ \lambda \sum_{j=1}^{D} \phi_s((\boldsymbol{e}_i)_j) \\ &+ \gamma \sum_{j=1}^{D} \phi_c((\boldsymbol{x}_i - X_{\hat{i}} \boldsymbol{c}_i - \boldsymbol{e}_i)_j) \end{aligned} \tag{9}$$

where $\phi_s(.)$ is also the l_1-l_2 loss function.

Here the term $\boldsymbol{e}_i$ is used to model the errors existing in data. And the error term $\boldsymbol{e}_i$ is assumed to have a sparse representation, which is expressed by the l_1-l_2 loss function. The third term of the objective function is a robust reconstruction fidelity measure of high-dimensional data using subspace representation and residual errors based on correntropy. Compared with the first formulation of robust subspace clustering defined in (8), the second formulation (9) is more suitable for the applications with large corruptions since its optimization model explicitly has error terms.

4. Solution of the proposed formulations

An iterative regularization method based on half-quadratic optimization is firstly proposed to solve the proposed formulations in (8) and (9). And then the convergence of the solution is analyzed. Finally the whole procedure of our subspace clustering algorithm is given. Since the optimization problem in (8) is a special case of (9), only the solution to (9) is provided here.

4.1. The half-quadratic approach

Since the problems (8) and (9) are not convex, it is difficult to optimize them directly. Fortunately, the half-quadratic technique [19] can be utilized to optimize the nonconvex function by minimizing its augmented function alternately.

Table 1. Loss functions and their minimizer functions

Functions $\phi(.)$	Minimizer functions $\delta(.)$
$\sqrt{\alpha + x^2}$	$1/\sqrt{\alpha + x^2}$
$1 - \exp(-\frac{x^2}{\sigma^2})$	$\exp(-\frac{x^2}{\sigma^2})$

According to the conjugate function theory [3] and HQ theory [19] [7], we have:

Lemma 1. *Suppose that $\phi(x)$ is a function that satisfies some conditions listed in* [19]*, then for a fixed x, there exists a dual potential function $\psi(.)$, such that*

$$\phi(x) = \inf_{s \in R} \{sx^2 + \psi(s)\} \tag{10}$$

where s is an auxiliary variable which is determined by the minimizer function $\delta(.)$ with respect to $\phi(x)$ (two specific functions and their minimizer functions are listed in Table 1).

According to Lemma 1, the augmented cost-function $\mathcal{J}$ of (9) reads

$$\begin{aligned}\mathcal{J}(\boldsymbol{c}_i, \boldsymbol{e}_i, \boldsymbol{r}, \boldsymbol{s}, \boldsymbol{q}) &= \boldsymbol{c}_i{}^T R \boldsymbol{c}_i + \lambda \boldsymbol{e}_i{}^T S \boldsymbol{e}_i \\ &+ \gamma(\boldsymbol{x}_i - X_{\hat{i}} \boldsymbol{c}_i - \boldsymbol{e}_i)^T Q (\boldsymbol{x}_i - X_{\hat{i}} \boldsymbol{c}_i - \boldsymbol{e}_i) \\ &+ \sum_{j=1}^{N} \psi_r(r_j) + \sum_{j=1}^{D} \psi_s(s_j) + \sum_{j=1}^{D} \psi_c(q_j)\end{aligned} \tag{11}$$

where $\boldsymbol{r} \in \mathbb{R}^N$, $\boldsymbol{s} \in \mathbb{R}^D$, $\boldsymbol{q} \in \mathbb{R}^D$ are auxiliary vectors, and $R = diag(\boldsymbol{r})$, $S = diag(\boldsymbol{s})$, $Q = diag(\boldsymbol{q})$.

Since the auxiliary vectors are only determined by their minimizer functions, the analytic forms of $\psi(.)$ in (11) can be eliminated when the auxiliary vectors are fixed. And λ is a positive constant scalar, so we can rewrite the function (11) as follows,

$$\begin{aligned}\mathcal{J}(\boldsymbol{c}_i, \boldsymbol{e}_i, \boldsymbol{r}, \boldsymbol{s}, \boldsymbol{q}) &= \boldsymbol{c}_i{}^T R \boldsymbol{c}_i + \boldsymbol{e}_i{}^T (\lambda S) \boldsymbol{e}_i \\ &+ \gamma(\boldsymbol{x}_i - X_{\hat{i}} \boldsymbol{c}_i - \boldsymbol{e}_i)^T Q (\boldsymbol{x}_i - X_{\hat{i}} \boldsymbol{c}_i - \boldsymbol{e}_i)\end{aligned} \tag{12}$$

or

$$\mathcal{J}(\boldsymbol{w}_i, \boldsymbol{p}, \boldsymbol{q}) = \boldsymbol{w}_i^T P \boldsymbol{w}_i + \gamma(\boldsymbol{x}_i - Y\boldsymbol{w}_i)^T Q (\boldsymbol{x}_i - Y\boldsymbol{w}_i) \tag{13}$$

where $\boldsymbol{w}_i = [\boldsymbol{c}_i^T, \boldsymbol{e}_i^T]^T \in \mathbb{R}^{(N+D)}$, $Y = [X_{\hat{i}}, I] \in \mathbb{R}^{D \times (N+D)}$, $\boldsymbol{p} = [\boldsymbol{r}^T, \lambda \boldsymbol{s}^T]^T$ and $P = diag(\boldsymbol{p})$.

Based on the HQ optimization theory, the function $\mathcal{J}(\boldsymbol{w}_i, \boldsymbol{p}, \boldsymbol{q})$ can be alternately minimized as follows,

$$p_j^t = \begin{cases} 1/\sqrt{(\boldsymbol{w}_i)_j^2 + \alpha} & j \le N \\ \lambda/\sqrt{(\boldsymbol{w}_i)_j^2 + \alpha} & \text{otherwise} \end{cases} \tag{14}$$

$$q_k^t = \exp(-(\boldsymbol{x}_i - Y\boldsymbol{w}_i)_k^2/\sigma^2) \tag{15}$$

$$\boldsymbol{w}_i^t = \arg\min_{\boldsymbol{w}_i} \mathcal{J}(\boldsymbol{w}_i, \boldsymbol{p}^t, \boldsymbol{q}^t) \tag{16}$$

where t is the iteration number.

The partial derivative of $\mathcal{J}(\boldsymbol{w}_i, \boldsymbol{p}^t, \boldsymbol{q}^t)$ with respect to $\boldsymbol{w}_i$ is

$$\frac{\partial J(\boldsymbol{w}_i, \boldsymbol{p}^t, \boldsymbol{q}^t)}{\partial \boldsymbol{w}_i} = 2P\boldsymbol{w}_i + 2\gamma Y^T Q Y \boldsymbol{w}_i - 2\gamma Y^T Q \boldsymbol{x}_i \tag{17}$$

By setting the derivative to zero, we obtain the closed solution of (16)

$$\boldsymbol{w}_i^* = \gamma(P + \gamma Y^T Q Y)^{-1} Y^T Q \boldsymbol{x}_i \tag{18}$$

Like any other kernel methods, the selection of kernel size σ will affect the performance of the proposed method, therefore σ should be carefully selected to guarantee a non-increasing function for the objective function. The kernel size σ of this paper is computed following the method in [10]

$$\sigma^2 = \frac{1}{2D} \|\boldsymbol{x}_i - Y\boldsymbol{w}_i\|_2^2 \tag{19}$$

The complete algorithm is summarized in Algorithm 1.

Algorithm 1 Solving Problem (9) via HQ Minimization

```
Input: A data point $\boldsymbol{x}_i \in \mathbb{R}^D$, the matrix $X \in \mathbb{R}^{D \times N}$.
Output: $\boldsymbol{c}_i \in \mathbb{R}^N$ and $\boldsymbol{e}_i \in \mathbb{R}^D$.
$\boldsymbol{w}_i \leftarrow 0$, $Y \leftarrow [X_{\hat{i}}, I]$ and $t \leftarrow 1$.
repeat
    $p_j^t = \begin{cases} 1/\sqrt{(\boldsymbol{w}_i)_j^2 + \alpha} & j \le N \\ \lambda/\sqrt{(\boldsymbol{w}_i)_j^2 + \alpha} & \text{otherwise} \end{cases}$;
    $q_k^t = \exp(-(\boldsymbol{x}_i - Y\boldsymbol{w}_i)_k^2/\sigma^2)$;
    $\boldsymbol{w}_i^t = \gamma(P + \gamma Y^T Q Y)^{-1} Y^T Q \boldsymbol{x}_i$;
    $\sigma^2 = \frac{1}{2D} \|\boldsymbol{x}_i - Y\boldsymbol{w}_i\|_2^2$;
    $t = t + 1$;
until Converges
$\boldsymbol{c}_i = \boldsymbol{w}_i(1 : N)$ and $\boldsymbol{e}_i = \boldsymbol{w}_i(N + 1 : N + D)$.
```

4.2. Convergence analysis

According to the properties of HQ [10][19], $J(\boldsymbol{w}_i^{t+1}, \boldsymbol{p}^{t+1}, \boldsymbol{q}^{t+1}) \le J(\boldsymbol{w}_i^t, \boldsymbol{p}^{t+1}, \boldsymbol{q}^{t+1}) \le J(\boldsymbol{w}_i^t, \boldsymbol{p}^t, \boldsymbol{q}^t)$. The cost function is non-increasing at each alternating minimization step. And according to the property of correntropy [16], the objective function $J(\boldsymbol{w}_i^t, \boldsymbol{p}^t, \boldsymbol{q}^t)$ is bounded and hence the objective function in (9) should be decreased following Algorithm 1 step by step until it converges. Since the loss function is non-convex, Algorithm 1 may obtain a local minimization. Algorithm 1 can be efficiently implemented using parallel computing because the representation of each data point is computed independently.

4.3. Subspace clustering

For each data point $\boldsymbol{x}_i$, we solve the optimization problem (8) or (9) by the Algorithm 1. Then we obtain the coefficient matrix $C = [\boldsymbol{c}_1, ..., \boldsymbol{c}_N]$, and the affinity matrix is defined as $W = |C| + |C|^T$. Similar to SSC, we apply the spectral clustering algorithm of Ng *et al.* [18] to the affinity matrix and get the ultimate clustering results. The whole procedure of our subspace clustering method can be summarized in Algorithm 2.

Algorithm 2 Subspace Clustering via Half-Quadratic minimization (SCHQ)

```
Input: Data matrix $X = [x_1, ..., x_N] \in \mathbb{R}^{D \times N}$.
Output: Segmentation of the data.
1: For each data point $x_i$, solve the problem (8) or (9) by
   the Algorithm 1. Obtain the coefficients matrix $C$.
2: Define the affinity matrix $W = |C| + |C|^T$.
3: Apply the spectral clustering algorithm [18] to the affin-
   ity matrix.
```

5. Experiments

Two real-world applications of subspace clustering, motion segmentation and face clustering, are used to evaluate the proposed SCHQ (Subspace Clustering based on Half-Quadratic Minimization) method and compare it with state-of-the-art subspace clustering methods, e.g., Local Subspace Analysis (LSA) [27], Spectral curvature clustering (SCC) [4], LRR [13], LSR [17], Low-Rank Subspace Clustering (LRSC) [6], and SSC [5].

5.1. Datasets

The Hopkins 155 dataset [22], which is available online at `http://www.vision.jhu.edu/data/hopkins155/`, is used for motion segmentation experiments. This dataset consists of 120 2-motion and 35 3-motion video sequences[1] and each motion corresponds to a single subspace. The feature trajectories in the video sequences are automatically extracted with a tracker, and outliers in the dataset have been removed manually.

The Extended Yale Dataset B [12] is adopted as the benchmark for the face clustering problem. The dataset consists of frontal face images of 38 subjects taken under varying light conditions, and each image is cropped into 192×148 pixels. The face clustering experiments only use the facial images of the first 10 subjects for testing following other subspace clustering experiments in the literature. And each face image is resized to 48×42 pixels and stacked into a $2016D$-vector.

5.2. Settings

The proposed subspace clustering method (SCHQ) is implemented based on the Algorithm 2. Since the motion segmentation experiment is a problem of clustering slightly corrupted data points lying in a union of affine subspaces, the formulation of (8) is used following the similar technique in [5]. The sum of the coefficients is enforced to be 1, *e.g.*, $\mathbf{1}^T \boldsymbol{c}_i = 1$. The optimization procedure is presented in Appendix. The formulation (9) is used for face clustering problem because there are sharp intensity variations among intra-class face images in the Extended Yale Dataset B.

The source code of state-of-the-art subspace clustering methods downloaded or provided by the authors is implemented for comparison. And we have tried our best to use the same algorithm settings described in the publications of compared methods. More specifically, the same setting of SSC [5] is used in the experiments, i.e., the noisy variation is used for motion segmentation and sparse outlying entries variation is used for face clustering. Both two versions of LSR described in [17], LSR1 and LSR2, are used in the experiments. Since a post-processing step is used in [13] to the coefficients matrix obtained by LRR, we report both the results of LRR and LRR-H (with post-processing). It should be noted the latest release of the algorithm [13] is implemented in this paper, which is different to the version in submission stage. The method in [6] is used for LRSC, i.e., Lemma 1 for motion segmentation and an ALM variant for face clustering. Because the authors do not provide the clustering algorithm and it is also based on low-rank representation, we use the same clustering algorithm as LRR-H to achieve the best performance in comparison.

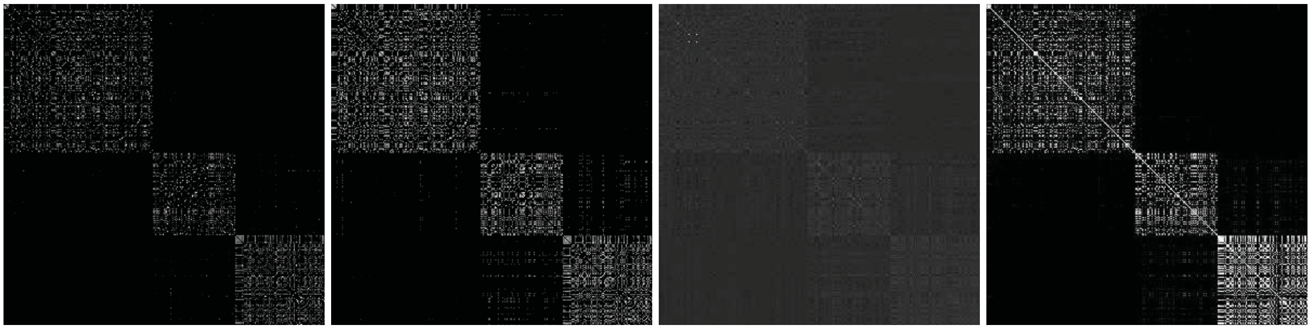
5.3. Motion segmentation

Motion segmentation problem is an important step in video sequences analysis [5]. Given a set of feature points tracked through the video sequences, the task of motion segmentation is to separate the trajectories of those feature points according to the motions belong to. As pointed out in [5], the set of all feature trajectories lie in a union of affine subspaces, which means the problem of motion segmentation can be reduced to segmenting the set of data points into a union of subspaces.

Preprocessing steps like PCA are usually used to reduce the dimension of the data [5] [17] so the structure of the data may be damaged. Because the purpose of this paper is to study the robustness of subspace clustering, all tested methods are directly applied to the original $2F$-dimensional data.

The results of the clustering using different methods are shown in Table 2. As we can see, our method performs the best for the 2-motion case. And our result is just slightly worse than LRR-H in the 3-motion case. In summary, our

[1] Actually, there are total 156 video sequences in the dataset with one sequence of 5 motions. However, only the results of 155 video sequences are reported following the experimental settings of other methods.



(a) SCHQ (b) SSC (c) LRR (d) LRR-H

Figure 1. Sparse coefficients for the 3-motion sequence cars10.

Table 2. Clustering error (%) on Hopkins 155 dataset with the original $2F$-dimensional data

Algorithm	LSA	SCC	LRR	LRR-H	LSR1	LSR2	LSRC	SSC	SCHQ
2 Motions									
Mean	3.36	2.24	3.30	1.33	1.80	2.08	2.46	1.52	**1.08**
Median	0.50	**0.00**	0.34	**0.00**	0.11	0.10	**0.00**	**0.00**	**0.00**
3 Motions									
Mean	8.48	6.69	7.39	**2.51**	4.14	4.85	6.03	4.40	2.73
Median	1.94	0.40	2.80	**0.00**	1.60	1.83	2.20	0.56	0.43
All									
Mean	4.52	3.25	4.22	1.60	2.33	2.72	3.27	2.18	**1.45**
Median	0.57	**0.00**	0.53	**0.00**	0.31	0.31	**0.00**	**0.00**	**0.00**

method achieves the best overall performance for all 155 video sequences. In order to demonstrate the superiority of our method in constructing sparse coefficients, the coefficients matrix C of the 3-motion sequence cars10 obtained by the proposed SCHQ, SSC, LRR and LRR-H are shown in Fig. 1. And the clustering error rates of these methods are 0.67%, 4.04%, 10.44%, and 5.39%, respectively. Note that the smaller number of non-zeros entries lying outside the diagonal blocks, the better result in subspace clustering. Obviously, the matrix obtained by our method is much more clear than others outside the diagonal blocks. This may be the reason why our method works.

Specially, we can also see that the post-processing of the coefficient matrix makes the matrix more 'bright' and 'clean' for LRR, which helps improve the clustering performance (LRR-H). The mean error rate of the method proposed in [15] (a variation of LRR) is 2.95%, while the mean error rate decreases significantly to 0.85% after the post-processing step. To the best of our knowledge, this is the best performance on the Hopkins 155 dataset under the same setting. The results also demonstrate the importance of post-processing in LRR method.

5.4. Face clustering

The face clustering problem refers to clustering face images of multiple subjects taken under fixed pose and varying illumination according to their subjects [5]. It has been proved in [1] that, under the Lambertian assumption, images of a object obtained with varying illumination lie close to a $9D$ linear subspace. Therefore, it implies that the set of face images of different subjects with varying illumination can be approximated by a union of 9-dimensional linear subspaces.

In order to evaluate our method's effectiveness in dealing with large corruptions, we use the Extended Yale Dataset B. For the purpose of studying the effect of subject number in face clustering, we test all the methods on the first 5 and 10 subjects respectively. Consistently, we apply the clustering methods to the original data as we did in the motion segmentation problem.

The clustering results of different methods are shown in Table 3. As we can see, as the number of subject increases, the clustering errors of all methods increase in different degrees. It may be the result of the expansion of the dictionary X. Obviously, our method performs the best both on 5 and 10 subjects clustering problems. And the advantage of our method is much more significantly on the 10-subject case which implies that our method is more robust to the errors caused by large corruptions or noise.

In Table 3, we also observe that the clustering errors of LSA, SCC, LSR1, and LSR2 are larger than those of other subspace clustering errors. This may be because LSA, SC-

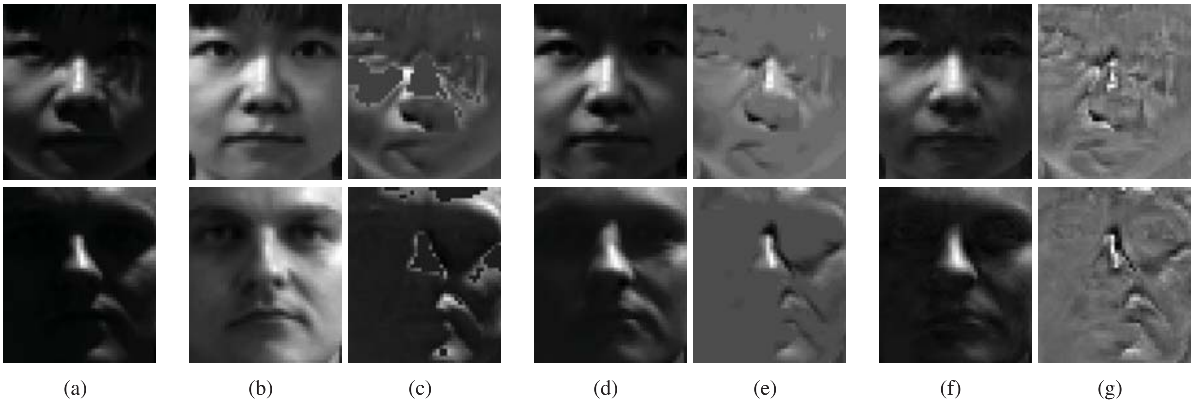


Figure 2. Examples of reconstruction images and errors of different methods (a) Original images (b) Reconstruction images by our method (c) Reconstruction errors by our method (d) Reconstruction images by SSC. (e) Reconstruction errors by SSC. (f) Reconstruction images by LRR-H. (g) Reconstruction errors by LRR-H.

Table 3. Clustering error (%) on the Extended Yale B dataset without pre-processing

Algorithm	LSA	SCC	LRR	LRR-H	LSR1	LSR2	LSRC	SSC	SCHQ
5 Subjects									
Error	54.06	60.56	14.69	1.88	13.75	5.31	3.75	**0.00**	**0.00**
10 Subjects									
Error	67.34	73.59	33.75	8.91	37.81	34.38	10.47	9.38	**2.03**

C, LSR1 and LSR2 are based on MSE. Since large errors will dominate the MSE, MSE based methods are prone to the presence of outliers that are significantly far away from the rest of the data points. Algorithmic robustness, which is derived from the statistical definition of a breakdown point, is the ability of an algorithm to tolerate a large number of outliers. From the viewpoint of robustness, these four methods may fail to deal with large outliers albeit they can work well on motion segmentation.

Some example reconstructed face images and sparse errors of different methods are shown in Fig. 3. We can see that even though the original face images are severely corrupted by shadow, our method can still recover the nearly perfect uncorrupted images while other methods all fail. It emphasizes the fact that our method can detect and correct the errors in data points simultaneously.

6. Conclusions

A novel optimization model for robust subspace clustering has been proposed in this paper. Correntropy has been demonstrated as a robust regularization term in subspace clustering. An efficient solution to the novel optimization problem is proposed based on half-quadratic minimization. Finally, experimental results on Hopkins 155 dataset and Extended Yale Database B without any-preprocessing have shown that our method have achieved the state-of-the-art performance. Our future work is to try other possible robust measures in subspace clustering since this paper has demonstrated the success of introduction of robust measures to the optimization problem of subspace clustering.

Acknowledgement

This work is funded by the National Basic Research Program of China (Grant No. 2012CB316300), the National Natural Science Foundation of China (Grant No. 61273272, 61103155) and the International S&T Cooperation Program of China (Grant No. 2010DFB14110).

A. Appendix

The following objective function is formulated if the constraint $\mathbf{1}^T \boldsymbol{c}_i = 1$ is integrated into problem (8):

$$\begin{aligned} \min_{\boldsymbol{c}_i} & \sum_{j=1}^{N} \phi_r((\boldsymbol{c}_i)_j) + \gamma \sum_{j=1}^{D} \phi_c((\boldsymbol{x}_i - X_{\hat{i}}\boldsymbol{c}_i)_j) \\ \text{s.t.} \quad & \mathbf{1}^T \boldsymbol{c}_i = 1 \end{aligned} \tag{20}$$

And the following Lagrangian function is obtained by introducing the Lagrangian multiplier λ to the function (20),

$$\begin{aligned} \mathcal{L}(\boldsymbol{c}_i) = & \sum\nolimits_{j=1}^{N} \phi_r((\boldsymbol{c}_i)_j) - \lambda(\mathbf{1}^T \boldsymbol{c}_i - 1) \\ & + \gamma \sum\nolimits_{j=1}^{D} \phi_c((\boldsymbol{x}_i - X_{\hat{i}}\boldsymbol{c}_i)_j) \end{aligned} \tag{21}$$

Note that λ is different from the penalty-term λ in (9).

The augmented cost-function $\mathcal{J}$ of (21) is as follows according to Section 4.1,

$$\begin{aligned}\mathcal{J}(\boldsymbol{c}_i, \boldsymbol{p}, \boldsymbol{q}) =&\boldsymbol{c}_i{}^T P\boldsymbol{c}_i - \lambda(\mathbf{1}^T\boldsymbol{c}_i - 1)\\ &+ \gamma(Y\boldsymbol{c}_i - X_{\hat{i}}\boldsymbol{c}_i)^T Q(Y\boldsymbol{c}_i - X_{\hat{i}}\boldsymbol{c}_i)\end{aligned} \tag{22}$$

where $Y = [\boldsymbol{x}_i, ..., \boldsymbol{x}_i] \in \mathbb{R}^{D\times N}$ and $Y\boldsymbol{c}_i = \boldsymbol{x}_i$.

The function (22) is simplified as follows,

$$\mathcal{J}(\boldsymbol{c}_i, \boldsymbol{p}, \boldsymbol{q}) = \boldsymbol{c}_i{}^T G\boldsymbol{c}_i - \lambda(\mathbf{1}^T\boldsymbol{c}_i - 1) \tag{23}$$

where $G = P + \gamma(Y - X_{\hat{i}})^T Q(Y - X_{\hat{i}})$.

And the objective function can be solved alternately as follows,

$$p_j^t = 1/\sqrt{(\boldsymbol{c}_i)_j^2 + \alpha} \tag{24}$$

$$q_k^t = \exp(-(\boldsymbol{x}_i - X_{\hat{i}}\boldsymbol{c}_i)_k^2/\sigma^2) \tag{25}$$

$$\boldsymbol{c}_i^t = \arg\min_{\boldsymbol{c}_i} \mathcal{J}(\boldsymbol{c}_i, \boldsymbol{p}^t, \boldsymbol{q}^t) \tag{26}$$

The partial derivatives of $\mathcal{J}(\boldsymbol{c}_i, \boldsymbol{p}^t, \boldsymbol{q}^t)$ with respect to $\boldsymbol{c}_i$ and λ are

$$\frac{\partial J(\boldsymbol{c}_i, \boldsymbol{p}^t, \boldsymbol{q}^t)}{\partial \boldsymbol{c}_i} = 2G\boldsymbol{c}_i - \lambda\mathbf{1} = 0 \tag{27}$$

$$\frac{\partial J(\boldsymbol{c}_i, \boldsymbol{p}^t, \boldsymbol{q}^t)}{\partial \lambda} = \mathbf{1}^T\boldsymbol{c}_i - 1 = 0 \tag{28}$$

Then the closed solution of (26) is obtained,

$$\boldsymbol{c}_i^* = \frac{\lambda}{2}G^{-1}\mathbf{1} \tag{29}$$

The parameter λ can be adjusted to ensure that the sum of the entries of $\boldsymbol{c}_i$ is equal to 1.

References

[1] R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Trans. PAMI*, 25(2):218–233, 2003.

[2] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Trans. PAMI*, 19(7):711–720, 1997.

[3] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[4] G. Chen and G. Lerman. Spectral curvature clustering (scc). *IJCV*, 81(3):317–330, 2009.

[5] E. Elhamifar and R. Vidal. Sparse subspace clustering: algorithm, theory, and applications. *IEEE Trans. PAMI*, 34(11):2765–2781, 2013.

[6] P. Favaro, R. Vidal, and A. Ravichandran. A closed form solution to robust subspace estimation and clustering. In *CVPR*, pages 1801–1807, 2011.

[7] D. Geman and G. Reynolds. Constrained restoration and the recovery of discontinuities. *IEEE Trans. PAMI*, 14(3):367–383, 1992.

[8] J. Gui, W. Jia, L. Zhu, S.-L. Wang, and D.-S. Huang. Locality preserving discriminant projections for face and palmprint recognition. *Neurocomputing*, 73(13):2696–2707, 2010.

[9] J. Gui, Z. Sun, W. Jia, R. Hu, Y. Lei, and S. Ji. Discriminant sparse neighborhood preserving embedding for face recognition. *Pattern Recognition*, 45(8):2884–2893, 2012.

[10] R. He, W.-S. Zheng, and B.-G. Hu. Maximum correntropy criterion for robust face recognition. *IEEE Trans. PAMI*, 33(8):1561–1576, 2011.

[11] R. He, W.-S. Zheng, T. Tan, and Z. Sun. Half-quadratic based iterative minimization for robust sparse representation. *IEEE Trans. PAMI, in press*, 2013.

[12] K.-C. Lee, J. Ho, and D. J. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Trans. PAMI*, 27(5):684–698, 2005.

[13] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. PAMI*, 35(1):171–184, 2013.

[14] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *ICML*, pages 663–670, 2010.

[15] G. Liu and S. Yan. Latent low-rank representation for subspace segmentation and feature extraction. In *ICCV*, pages 1615–1622, 2011.

[16] W. Liu, P. P. Pokharel, and J. C. Príncipe. Correntropy: properties and applications in non-gaussian signal processing. *IEEE Trans. Signal Processing*, 55(11):5286–5298, 2007.

[17] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan. Robust and efficient subspace segmentation via least squares regression. In *ECCV*, pages 347–360, 2012.

[18] A. Y. Ng, M. I. Jordan, Y. Weiss, et al. On spectral clustering: analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.

[19] M. Nikolova and M. K. Ng. Analysis of half-quadratic minimization methods for signal and image recovery. *SIAM Journal on Scientific computing*, 27(3):937–966, 2005.

[20] Y. Sugaya and K. Kanatani. Geometric structure of degeneracy for multi-body motion segmentation. In *Statistical Methods in Video Processing*, pages 13–25. 2004.

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

[22] R. Tron and R. Vidal. A benchmark for the comparison of 3-d motion segmentation algorithms. In *CVPR*, pages 1–8, 2007.

[23] P. Tseng. Nearest q-flat to m points. *Journal of Optimization Theory and Applications*, 105(1):249–252, 2000.

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

[25] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (gpca). *IEEE Trans. PAMI*, 27(12):1945–1959, 2005.

[26] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. PAMI*, 31(2):210–227, 2009.

[27] J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *ECCV*, pages 94–106, 2006.