

Efficient l_q norm based sparse subspace clustering via smooth IRLS and ADMM

Shenfen Kuang^{1,2} · HongYang Chao¹ · Jun Yang¹

Received: 24 February 2016 / Revised: 26 September 2016 / Accepted: 24 October 2016 /
Published online: 9 November 2016
© Springer Science+Business Media New York 2016

Abstract Recently, sparse subspace clustering, as a subspace learning technique, has been successfully applied to several computer vision applications, e.g. face clustering and motion segmentation. The main idea of sparse subspace clustering is to learn an effective sparse representation that are used to construct an affinity matrix for spectral clustering. While most of existing sparse subspace clustering algorithms and its extensions seek the forms of convex relaxation, the use of non-convex and non-smooth l_q ($0 < q < 1$) norm has demonstrated better recovery performance. In this paper we propose an l_q norm based Sparse Subspace Clustering method (lqSSC), which is motivated by the recent work that l_q norm can enhance the sparsity and make better approximation to l_0 than l_1 . However, the optimization of l_q norm with multiple constraints is much difficult. To solve this non-convex problem, we make use of the Alternating Direction Method of Multipliers (ADMM) for solving the l_q norm optimization, updating the variables in an alternating minimization way. ADMM splits the unconstrained optimization into multiple terms, such that the l_q norm term can be solved via Smooth Iterative Reweighted Least Square (SIRLS), which converges with guarantee. Different from traditional IRLS algorithms, the proposed algorithm is based on gradient descent with adaptive weight, making it well suit for general sparse subspace clustering problem. Experiments on computer vision tasks (synthetic data, face clustering and motion segmentation) demonstrate that the proposed approach achieves considerable improvement of clustering accuracy than the convex based subspace clustering methods.

✉ HongYang Chao
isschhy@mail.sysu.edu.cn

Shenfen Kuang
shfkuang@gmail.com

Jun Yang
yangj95@mail2.sysu.edu.cn

¹ School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China

² School of Mathematics and Statistics, Shaoguan University, Shaoguan, China

Keywords Sparse subspace clustering · Smooth IRLS · ADMM · l_q minimization · Sparse representation

1 Introduction

The high dimensional data, such as images, video, medical data, are ubiquitous in many real world applications. However, the high dimensional data not only increase the computational cost and memory requirement, but also cause the curse of dimensionality. In practice, the high dimensional data often lie in low dimensional subspace, namely, their intrinsic dimension is often much lower than the dimension of ambient space. For instance, in motion segmentation problem, feature point trajectories extracted from a video sequence with multiple moving objects lie in an affine subspace with dimension at most 4 [29]. Face images of different subjects under varying illumination lie in a linear subspace with dimension at most 9 [1]. Background modelling from surveillance video lie in a low rank subspace [18]. Complex patient data, created by bio-medicine, often lie in a low dimensional subspace of patient groups [19]. All of these applications motivate the technique for finding low-dimensional representations of the high-dimensional data. Subspace clustering, which refers to the problem of segment data into the corresponding subspaces, is proposed to address those problems.

Inspired by the success of compressed sensing and sparse coding, many subspace learning methods based on sparsity [35, 37] are proposed. Among them, Sparse Subspace Clustering (SSC) [10, 27] is the representative state-of-the-art subspace learning method, and has received tremendous attentions due to its simplicity and elegant formulation. The key for SSC is to design a good sparse representation model that reveals the real subspace structure of high-dimensional data. SSC tends to separate the high dimensional data into their underlying subspace, such that each data can be linearly represented by a few atoms of their corresponding subspace. It's a basis problem abstracted from machine learning, signal processing and computer vision. Many recently interesting applications motivated the development of SSC, e.g., Visual analytics for concept exploration in subspaces of patient groups [19], robust visual tracking via sparsity-induced subspace learning [28], dimensionality reduction subspace clustering through random projection [17], to name just a few. As a spectral based clustering method, it has been studied widely. In contrast with most classical subspace learning approaches, SSC can deal with multiple subspaces, of varying dimensions, data nuisances and robust to noise and outliers. SSC takes full advantage of the notion of self-expressiveness property [9], that is, each data can be expressed as a linear combination of a few points from its own subspace. Under the sparse assumptions, SSC can discover the low dimensional subspace structure.

In general, SSC leverages the sparsest representation under convex l_1 regularizer to learn the affinity matrix, then subspace segmentation is performed by spectral clustering algorithm. There are many state-of-the-art methods being proposed recently, compared to SSC, the difference mainly comes from the regularization. Least Squares Regression (LSR) [24] methods encourage a grouping effect which tends to group highly correlated data together. The Subspace Segmentation via Quadratic Programming (SSQP) [31] models subspace clustering with a convex quadratic optimization, Low Rank Representation (LRR) [22, 30] seeks the lowest rank representation among all data, and surrogate the rank function via nuclear norm minimization. Among the most existing algorithms for subspace clustering, they all implicitly seek the block diagonal structure of affinity matrix to ensure the performance. There are

many extensions to improve the performance by adapting this idea, such as [23] enhances subspace clustering by manifold regularization. Feng et al. [11] adapts block-diagonal prior to subspace segmentation. However, Most of existing subspace clustering are all based on the relax convex minimization, e.g. l_1 [34] or quadratic programmin [26] g. However, convex relaxation with l_1 norm, though is known to be the best convex surrogate of l_0 , is equal or closely approximated to l_0 only under some strict condition, such as RIP conditions [6]. In reality many applications violate this, and that the performance will degrade unless RIP satisfies. On the other hand, quadratic regularizations does not lead to sparse solution. Hence, to overcome this issue making by l_1 or quadratic programming, Thresholding Subspace Clustering (TSC) [16], SSC with Orthogonal Matching Pursuit (SSC-OMP) [8] are two heuristic approaches trying to find the sparse representation, but such greedy based methods are often sensitive to the noise and outliers.

Most of the methods mentioned above adapt regularization techniques to encourage sparsity, in that sparsity is a nature property of most signals(e.g. image data), it's key for the success of sparse subspace clustering. Many recently research [4, 6, 7, 21] indicated that enhancing sparsity will better uncover the structure of data and better recovery the subspace structure. The recent literature [5, 14] shown that in many situations, adapting l_q ($0 < q < 1$) norm approximate to l_0 will outperform l_1 minimization, in the sense that it often leads to more sparsity results. However, due to its non-convexity and non-smoothness, it's often difficult to solve. l_q norm instead of l_1 norm minimization make many problems more challenge. Many effective methods [12, 20] have been proposed for l_q ($1 < q < 1$) problem recently, yet most of the methods are based on iterative reweighted least square that transforms the origin problem to weighted l_1 [33] or l_2 [12] minimization. It has been proved that the algorithm has a linear rate of convergence for l_1 norm and super-linear for l_q ($0 < q < 1$) norm, under the RIP [32] condition. Although the l_q minimization problem is more difficult to solve than l_1 minimization problem, there are some novel methods being proposed recently for l_q norm problem, see, e.g., [4, 6, 21].

In many situations l_1 minimization does not always yield sufficiently sparse solution, l_q minimization is adopted as an alternative to l_0 minimization, the solution is often sparser than l_1 [20]. More specifically, recent work [14] shows that l_q norm minimization improves the performance on face recognition problem. [5] proposed locality constrained- l_q sparse subspace clustering, where l_q norm is employed to enhance sparsity and, for each data point, choosing k nearest neighbours instead of entire data points to lower the computational cost. Those works received state-of-the-art results by employing l_q norm minimization instead of l_1 minimization. Motivated by the recent advances of l_q ($0 < q < 1$) minimization, we investigate l_q norm regularization for SSC problem, demonstrating that it can further improve the clustering accuracy. However, existing l_q norm minimization for SSC only address the noise(e.g. Gaussian noise) case, in many real world applications the data are often corrupted by outliers and noise simultaneously [15]. Such that in [5] finding k neighbours for each data may fail. Moreover, traditional IRLS algorithms can only handle with the linear constraints, but does not take the measurement noise and outlier into consideration. It's difficult to address the l_q norm regularization via IRLS for SSC under noise and outliers. Moreover, some particular SSC applications often require extra constraints, such as affine constraint for motion segmentation problem, which is computational intractability. Hence, traditional iterative reweighted algorithm does not suitable for solving SSC problem. In this paper, we propose lqSSC that unify ADMM and IRLS to solve l_q norm based minimization efficiently, making it suitable for general sparse subspace clustering problem. Different from traditional IRLS algorithm that decomposes l_q norm with adapted reweighted l_1 or l_2

norm, our smooth IRLS algorithm is based on gradient descent with adaptive weight [25]. The advantage of gradient descent for l_q norm minimization is two fold. Firstly, easy to implementation and general for different kinds of problems. Secondly, though we discuss l_q norm here, the technique is suitable and will benefit other non-convex regularization problems. For instance, Logarithm penalty, SCAD, LSP, and capped norm, all of which share many similar properties with l_q norm, Zhang [36] unified this work.

1.1 Contribution

1. We make use of $l_q(0 < q < 1)$ norm for the sparse representation problem. Although it's non-convexity and non-smoothness, we introduce a smooth regularization and develop an Smooth Iterative Reweighted Least Square(SIRLS) algorithm, which is fast, easy to parallel and guarantee to converge.
2. We propose an l_q norm based model for sparse subspace clustering under ADMM framework, such that l_q norm terms can be solved via smooth IRLS. Experimental results show that the proposed approach achieves considerable improvements in comparison to convex models. Though our model focuses on l_q norm minimization, it can easily generalizes to other kinds of non-convex non-smooth regularizations(e.g. Log penalty, SCAD, LSP, Capped norm).

Notation and abbreviation

1. SSC: Sparse Subspace Clustering
2. IRLS: Iterative Reweighted Least Square
3. ADMM: Alternating Direction Method of Multipliers

for a vector $x \in R^d$, and a matrix $C \in R^{d \times n}$, we discuss the case $0 < p, q < 1$ by default, and define (p is defined similarly)

$$\begin{aligned} \|x\|_q &= \left(\sum_{i=1}^d x_i^q \right)^{\frac{1}{q}} \\ \|x\|_{q,\epsilon} &= \left(\sum_{i=1}^d (x_i^2 + \epsilon^2)^{\frac{q}{2}} \right)^{\frac{1}{q}} \\ \|C\|_q &= \sum_{i=1}^d \sum_{j=1}^n C_{i,j}^q \\ \|C\|_{q,\epsilon} &= \sum_{i=1}^d \sum_{j=1}^n (C_{i,j}^2 + \epsilon^2)^{\frac{q}{2}} \end{aligned}$$

2 Sparse subspace clustering model

We first formulate the basis sparse subspace clustering problem as follow: Given a data matrix $Y = [y_1, y_2, \dots, y_n] \in R^{d \times n}$, each data point is represented as a column vector and $\{y_i\}_{i=1}^n$ is drawn from a union of k linear subspaces $\{S_i\}_{i=1}^k$ and the corresponding dimension of each subspaces is $\{d_i\}_{i=1}^k$. The main task is to segment the data points to their underlying subspace and recovery the low dimensional subspace. It's known to be

NP-hard unless additional assumptions. A key technique for the SSC problem is sparse representation, which can be formulated as

$$\min \|c_i\|_0 \quad s.t. \quad y_i = Y_{-i}c_i \tag{1}$$

where $Y_{-i} = [y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_n]$ is a matrix excluding the i -th column of matrix Y . $\|c_i\|_0$ is the l_0 norm of vector c , that is, the number of non-zero elements. The problem (1) is known to be non-convex and NP-hard, but recent research in the emerging theory of sparse coding and compressed sensing [32] reveal that, if the solution c_i sought is sparse enough, the solution of the (1) is equal to (2)

$$\min \|c\|_1 \quad s.t. \quad y_i = Y_{-i}c_i \tag{2}$$

It has been shown that under some suitable conditions, c_i can be stably or exactly recovered. One of the commonly used framework for sparse representation is restricted isometry property (RIP) [32]. A vector x is k -sparse if for any $m \times n$ matrix A and any integer k , $1 \leq k \leq n$, the k -restricted isometry constant δ_k is defined as the smallest constant such that

$$(1 - \delta_k)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_k)\|x\|_2^2 \tag{3}$$

for all k -sparse vector x , the sufficient conditions include $\delta_{2k} < \sqrt{2} - 1$, $\delta_{2k} < \sqrt{2}/2$ and so on. However, it is shown in [32] that exactly recover x is not always possible if $\delta_{2k} \geq \sqrt{2}/2$. Therefore, l_1 norm may degrade the performance in such case. To illustrate this situation clearly, we now consider the following toy example, let

$$Y = \left\{ \begin{array}{cccc} 1 & 0 & 1/4 & 1/8 \\ 0 & 1 & 1/4 & 1/8 \end{array} \right\}, \min \|c\| \quad s.t. \quad y_4 = Y_{-4}c$$

It's obvious to see that when the objective function is employed with l_1 norm, the solution is $c_{l_1} = [1/8, 1/8, 0, 0]^T$, but the l_0 norm solution is $c_{l_0} = [0, 0, 1/2, 0]^T$, $\|c_{l_1}\| = 1/4 < \|c_{l_0}\| = 1$. However, the sparsity of l_1 is 2 and l_0 is 1, such that l_1 may result in inferior solution. This implies that l_1 is not a good approximation to l_0 . In order to obtain a solution more approximate to l_0 , we study l_q ($0 < q < 1$) minimization, which can be represented as follows

$$\min \|c_i\|_q^q \quad s.t. \quad y_i = Y_{-i}c_i \tag{4}$$

In a general situation, we consider the data that are contaminated with sparse outlying entries and noise, let

$$y_i = Y_{-i}c_i + e_i + z_i \tag{5}$$

be the data that is obtained by corrupting and error free $Y_{-i}c_i$, which perfectly lies in a subspace. e_i is the vector of sparse outlier that has only a small number of non-zero elements, which means that e_i is a sparse vector. z_i is the noise that is bounded as $\|z_i\|_2 < s$ for some $s > 0$, which means that s_i is a vector with small perturbation. We can formulate the objective function as

$$\min \|c_i\|_q^q + \|e_i\|_p^p + \|z_i\|_F^2 \tag{6}$$

with $0 < p, q \leq 1$. In order to deal with affine subspaces, we also add affine constraints on the coefficient c_i , that is $1^T c = 1$. We reformulate and unify it with noise and outliers in a matrix form

$$\begin{aligned} \min_{C,E,Z} & \|C\|_q + \lambda_e \|E\|_p + \lambda_z \|Z\|_F^2 \\ s.t. & \quad Y = YC + E + Z, \text{diag}(C) = 0, C^T \mathbf{1} = \mathbf{1} \end{aligned} \tag{7}$$

when $p = q = 1$, the model be the general SSC model that equivalents to [10]. Now we consider the case with $0 < p, q < 1$, which will form the l_q norm sparse subspace clustering(lqSSC for short) problem this paper focuses on. Both of the coefficient matrix C and outlier E are modelled with non-convex norm. Clearly, this problem is in general

intractable, for the reason that l_p, l_q norm are all non-convexity and non-smoothness. In order to address this problem, we propose a new approach by unifying ADMM and IRLS algorithm, which details in the following section.

3 Proposed method

In this section, we make use of ADMM for solving l_q norm minimization problem. We begin by introducing a general framework of ADMM, then deduce the smooth IRLS algorithm.

Alternating Direction Augmented Lagrangian(ADMM for short) [3], a method for solving the constrained optimization problems, has recently attracted much attention as it is well suited for some important classes of problems with special structure. In ADMM framework, l_q norm problem can be split by introducing an auxiliary variable that decouples the variables, such that the l_q norm term can be addressed by iterative reweighted algorithm. We consider the general l_q norm problem

$$L(x) = f(x) + g(x) \tag{8}$$

where $f(x) = \|x\|_q^q$ is non-smooth and non-convex, $g(x)$ is a smooth and convex function. Generally, all the constrained optimization can be relaxed to unconstrained optimization via Lagrangian multiplier. Hence it can always be abstracted and converted the unconstrained optimization with only two terms, one is the l_q norm regularizer term which we denote by $f(x)$, and the other term be the remain part with $g(x)$. Thus the complications resulting from the coupling of the origin problems of the augmented Lagrangian are eliminated. We introduce its augmented Lagrangian function

$$\min L(x, z, w) = f(x) + g(z) + w^T(x - z) + \frac{\rho}{2}\|x - z\|_2^2 \tag{9}$$

where w is the Lagrange multiplier dual variable, ρ is a positive parameter associated with the augmentation which improves the numerical stability of the algorithm. The augmented Lagrangian (9) can be equivalently formed as

$$\min L(x, z, w) = f(x) + g(z) + \frac{\rho}{2}\|x - z + \frac{1}{\rho}w\|_2^2 \tag{10}$$

Different from the traditional augmented Lagrangian methods that attempt to alternatively minimize x and z , ADMM algorithm minimizes the augmented Lagrangian by iteratively updating the primal and dual variables. Given the current iterates $(x^{(j)}, z^{(j)}, w^{(j)})$, it generates a new iterate $(x^{(j+1)}, z^{(j+1)}, w^{(j+1)})$ with respect to x , then respect to z , and finally performing a multiplier update. The $(j + 1)$ -th iteration can be represented as follows

$$x^{(j+1)} = \arg \min_x J(x) = f(x) + \frac{\rho}{2}\|x - z^{(j)} + \frac{1}{\rho}w^{(j)}\|_2^2 \tag{11}$$

$$z^{(j+1)} = \arg \min_z g(z) + \frac{\rho}{2}\|z - x^{(j)} - \frac{1}{\rho}w^{(j)}\|_2^2 \tag{12}$$

$$w^{(j+1)} = w^{(j)} + \rho(x^{(j+1)} - z^{(j+1)}) \tag{13}$$

Even $0 < q < 1$, the above iterate equations has the same iteration procedure as ADMM algorithm for the l_1 based models. However, in x -step it is no longer restricted to convex function. Since when $0 < q < 1$, a possible local minimizer may be trapped in after a few iterations. As the problem is non-convex, solving this non-convex problem directly may be converged to one of local minimizers. To overcome this problem, we introduce a sequential

minimization strategy, which improves the global convergence performance by employing smooth IRLS.

As $z^{(k+1)}$ is smooth and convex, it can be updated by any standard methods, such as Newton’s method or conjugate gradient method. We focus on x^{k+1} and solve it with an iterative reweighted algorithm. Due to the singularity of the gradient of the associated functional above because of the sparsity of solution x , we introduce the smooth regularized version of x^{k+1} . To make to problem easy to tackle, it can be converted with a smooth parameter ϵ

$$\|x\|_{q,\epsilon}^q = \sum_{i=1}^d (x_i^2 + \epsilon^2)^{\frac{q}{2}}$$

$\epsilon > 0$ is a smooth parameter which will decrease to zero in order to make $\|x\|_q^q$ differential. We now consider the optimization

$$x^{(j+1)} = \arg \min_x J(x, \epsilon) = \|x\|_{q,\epsilon}^q + \frac{\rho}{2} \|x - z^{(j)} + \frac{1}{\rho} w^{(j)}\|_2^2 \tag{14}$$

It’s well known that when $q = 1, \epsilon = 0$, (14) can be solved with l_1 shrinkage operator, that is, for $i=1$ to d ,

$$x_i^{(j+1)} = \begin{cases} z^{(j)} - \frac{1}{\rho} w^{(j)} - \frac{1}{\rho} x_i^{(j)} \geq \frac{1}{\rho} \\ 0 & |x_i^{(j)}| < \frac{1}{\rho} \\ z^{(j)} - \frac{1}{\rho} w^{(j)} + \frac{1}{\rho} x_i^{(j)} \leq -\frac{1}{\rho} \end{cases} \tag{15}$$

but when $0 < q < 1$, it can’t be solved directly as there not exist closed form solutions. Now we develop an efficient iterative reweighted algorithm to approximate this problem, because the function $J(x, \epsilon)$ in (14) are differential, derivative $J(x, \epsilon)$ with x and let it to 0, we get

$$\sum_{i=1}^d \frac{qx_i}{(\epsilon^2 + x_i^2)^{1-\frac{q}{2}}} + \rho \left(x - z^{(j)} + \frac{w^{(j)}}{\rho} \right) = 0 \tag{16}$$

Inspired by the IRLS algorithm, firstly, let

$$W^{(k)} = \text{diag}(x_i^{(k)}), \quad x_i^{(k)} = (\epsilon_k^2 + (x_i^{(k)})^2)^{\frac{q}{2}-1} \tag{17}$$

the (16) can be reformulated as follows

$$qW^{(k)}x + \rho \left(x - z^{(j)} + \frac{w^{(j)}}{\rho} \right) = 0 \tag{18}$$

x -update minimization of (18) can be carried out by solving linear equations or conjugate gradient method. Note that $z^{(j)}$ and $w^{(j)}$ are fixed in (18) of the Algorithm 1 and it updates only on the ADMM iterations. In order to achieve a more sparser solution, the terminate condition needs to be carefully designed. Our algorithm utilizes an alternating method for choosing weights and minimizers based on the (18). For $x \in R^d$, define the non-increasing rearrangement $r(x)$ of the absolute values of the entries of x . Thus $r(x)_m$ is the m^{th} -largest of the set $\{|x_j|, j = 1, \dots, d\}$, and a vector x is m -sparse if and only if $r(x)_{m+1} = 0$. We decrease ϵ with $\epsilon^{(k+1)} = \min\{\epsilon^{(k)}, r(x^{(k+1)})_{m+1}/d\}$, it can serve as a posteriori check on whether or not it have converged to a sparse solution. If $\epsilon^{(k+1)} > \epsilon$ and $\epsilon^{(k+1)} = r(x^{(k+1)})_{m+1} = \epsilon$, then if we let $x_{[m]}$ to be a m sparse vector, we have $\|x^{(k+1)} - x_{[m]}\| \leq \epsilon/n$. In particular if $\epsilon^{(k+1)} = 0$, then $x^{(k+1)}$ is m -sparse solution.

Algorithm 1 Smooth Iterative Reweighted Least Square (SIRLS) Algorithm For (14)

Input:

initializes $x_0, \epsilon_0, \gamma > 0, \epsilon > 0, k = 0;$
 set $W^{(0)} = \text{diag}(x_0)$, sparsity m .

Output:

m -sparse approximated solution $x^{(k+1)}$.

1: **repeat**

2: Solve $x^{(k+1)}$ by

$$qW^{(k)}x^{(k+1)} + \rho(x^{(k+1)} - z^{(j)} + \frac{w^{(j)}}{\rho}) = 0;$$

3: Update $\epsilon_{k+1} = \min\{\epsilon_k, r(x^{(k+1)})_{m+1}/n\};$

4: Update $W^{(k+1)} = \text{diag}(x_i^{(k+1)});$

5: $k = k + 1;$

6: **until** $\|x^{(k)} - x^{(k-1)}\|_2 < \gamma$ or $\epsilon_k < \epsilon;$

3.1 Convergence analysis

The convergence analysis of ADMM has been studied thoroughly in many literature like [3] and we omit it here. We mainly focus on the convergence of smooth IRLS illustrating in Algorithm 1, which is the main contribution of this paper. As (14) is not convex, the stationary points can't be ensure to converge to global minimization, like the EM-type algorithm. It depends on a good initial value W . Therefore, the algorithm may be in local optimization if the initiation is not properly designed. To overcome the limitation of local minimization, the algorithm can be repeated a few times in a sequence. For example, we first choose an random initiation for the algorithm with $0 < q_1 < 1$, then run it again with the initiation of the previous result but choose $0 < q_2 < q_1$. Conducting Algorithm 1 for a few times the global optimization can often be achieved. This may not take much time in that it allows warm start by the previous results as an initialization. If the smooth IRLS of (14) find a good approximation, the ADMM procedure may converge to the global solution with high probability.

We now give the convergence analysis of the smooth IRLS algorithm. Firstly, $J(x, \epsilon) \geq J(x)$ with a given $\epsilon > 0$, where the equality holds if and only if $\epsilon = 0$. Secondly, it's easy to find that $J(x)$ is majorized by $J(x, \epsilon)$. Decreasing $J(x, \epsilon)$ tends to decrease $J(x)$. Given any $\gamma > 0$, there exist $\epsilon > 0$, such that $J(x, \epsilon) - J(x) < \gamma$. Suppose x_ϵ^*, x^* is the optimal solution to (14) and (11), then we have

$$0 < J(x_\epsilon^*) - J(x^*) \leq J(x_\epsilon^*, \epsilon) - J(x^*, \epsilon) + \gamma \leq \gamma$$

Based on the above results, we have the following convergence theorem of the smooth IRLS algorithm.

Theorem 1 *The sequence $\{x^{(k)}\}$ generated in Algorithm 1 satisfies the following properties:*

- (1) $J(x, \epsilon)$ in non-increasing, that is, $J(x^{(k+1)}, \epsilon_{k+1}) \leq J(x^{(k)}, \epsilon_k)$.
- (2) The sequence $\{x^{(k)}\}$ is bounded.
- (3) the sequence converges to a critical point, that is, there exist a point x^{ϵ^*} satisfies, (let $u = z^{(j)} - w^{(j)}/\rho$)

$$qW^{\epsilon^*}x^{\epsilon^*} + \rho(x^{\epsilon^*} - u) = 0 \tag{19}$$

we give the proof in [Appendix](#).

Finally, if (11) converge, the stationary points can be achieved via alternating minimization of (12), (11), (13).

4 l_q norm based sparse subspace clustering

In this section, we will unify smooth IRLS and ADMM to solve the l_q norm based sparse subspace clustering. SSC problem is equality constrained minimization problem as (7) shown. We first reformulate the problem (7) into unconstrained optimization, and then decompose the optimization with separately iteration. Note that using the equality constraint in (7), we can eliminate Z from the optimization program and equivalently solve

$$\begin{aligned} \min_{C,E} & \|C\|_q + \lambda_e \|E\|_p + \frac{\lambda_z}{2} \|Y - YC - E\|_F^2 \\ \text{s.t.} & \quad C^T \mathbf{1} = \mathbf{1}, \text{diag}(C) = 0 \end{aligned} \tag{20}$$

To solve optimization problem (20), we first introduce an auxiliary variable matrix A to augment the constraints into the objective function, and iteratively minimize the Lagrangian with respect to the primal variables and maximize the dual variables. Convert the problem (20) to the following equivalent formulation

$$\begin{aligned} \min_{C,E} & \|C\|_q + \lambda_e \|E\|_p + \frac{\lambda_z}{2} \|Y - YC - E\|_F^2 \\ \text{s.t.} & \quad A^T \mathbf{1} = \mathbf{1}, A = C - \text{diag}(C) \end{aligned} \tag{21}$$

The solution of 21 and (20) is equivalent. The purpose of auxiliary variable is to split the optimization into multiple terms like the discussion in previous section. Next, by introducing the parameter ρ , the objective function of (21) has two extra penalty terms corresponding to the constraints, and consider the following optimization program

$$\begin{aligned} L(C, A, E, \Delta, \rho) &= \|C\|_q + \lambda_e \|E\|_p + \frac{\lambda_z}{2} \|Y - YC - E\|_F^2 \\ &+ \langle \delta, A^T \mathbf{1} - \mathbf{1} \rangle + \langle \Delta, A - C + \text{diag}(C) \rangle \\ &> + \frac{\rho}{2} \|A^T \mathbf{1} - \mathbf{1}\|_2^2 + \frac{\rho}{2} \|A - C + \text{diag}(C)\|_F^2 \end{aligned} \tag{22}$$

where $\langle A, B \rangle = \text{trace}(A^T B)$. The optimization of (22) can be solved iteratively by updating C, A, E, Δ, ρ while keeping other variables fixed, The update procedure can be computed as follows

- Update $C^{(k+1)}$ by minimizing (22) with respect to C , while $A^{(k)}, E^{(k)}, \delta^{(k)}, \Delta^{(k)}$ are fixed. Note that,

$$C^{(k+1)} = \|C\|_q + \frac{\rho}{2} \|C - \text{diag}(C) - A^{(k)} + \frac{\Delta^{(k)}}{\rho}\|_F^2 \tag{23}$$

the optimization of (23) is the matrix form of (11). To solve this problem, we first smooth the objective function with ϵ , and consider the following function

$$\begin{aligned} \min_J F(J, \epsilon) &= \|J\|_{q,\epsilon} + \frac{\rho}{2} \|J - A^{(k)} + \frac{\Delta^{(k)}}{\rho}\|_F^2 \\ &= \sum_{i=1}^n \|J_i\|_{q,\epsilon} + \frac{\rho}{2} \|J_i - A_i^{(k)} + \frac{\Delta_i^{(k)}}{\rho}\|_2^2 \end{aligned} \tag{24}$$

We can minimize $F(J_i, \epsilon) (i = 1, 2, ..n)$ via smooth IRLS independently using (Algorithm 1), then we have $C^{(k+1)} = J^{(k+1)} - \text{diag}(J^{(k+1)})$.

- Update $E^{(k+1)}$ by

$$E^{(k+1)} = \min_E \|E\|_q + \frac{\lambda_z}{2\lambda_e} \|Y - YC^{(k+1)} - E\|_F^2 \tag{25}$$

The solution has the same form as (14), and we use the smooth IRLS depicted in Algorithm 1 to obtain the solution.

- Update $A^{(k+1)}$ by minimizing $L(C, A, E, \delta, \rho)$ with respect to A , while $C^{(k+1)}, E^{(k)}, \delta^{(k)}, \rho^{(k)}$ are fixed. Let the derivative of L with A to zero, we get

$$(\lambda_z Y^T Y + \rho I + \rho \mathbf{1}\mathbf{1}^T)A^{(k+1)} = \lambda_z Y^T (Y - E^{(k+1)}) + \rho(\mathbf{1}\mathbf{1}^T + C^{(k+1)}) - \mathbf{1}\delta^{(k)T} - \Delta^{(k)} \tag{26}$$

We solve the above equation directly since the coefficient of the left-hand side is positive definite matrix. We obtain $A^{(k+1)}$ via the linear equations, or solve it with conjugate gradient method.

- Update the Lagrangian multipliers with $C^{(k+1)}, E^{(k+1)}, A^{(k+1)}$ fixed, perform a gradient ascent update with the step size of ρ .

$$\begin{aligned} \delta^{(k+1)} &= \delta^{(k)} + \rho(A^{(k+1)T} \mathbf{1} - \mathbf{1}) \\ \Delta^{(k+1)} &= \Delta^{(k)} + \rho(A^{(k+1)} - C^{(k+1)}) \end{aligned} \tag{27}$$

Iteratively updated the procedure above until convergence is achieved to the setting of maximum iteration, or convergence is achieved when $\|A^{(k)} - A^{(k-1)}\|_\infty \leq \eta, \|A^{(k)T} \mathbf{1} - \mathbf{1}\|_\infty \leq \eta, \|A^{(k)} - C^{(k)}\|_\infty \leq \eta$, and $\|E^{(k)} - E^{(k-1)}\|_\infty \leq \eta$, where η denotes the error tolerance for the primal and dual residuals. In this paper we choose $\eta = 10^{-6}$ for all experiments. The detail process is illustrated in Algorithm 2.

Algorithm 2 l_q norm based Sparse Subspace Clustering (lqSSC) with IRLS and ADMM

Input:

- Input data matrix Y ;
- Set maxIter, $k = 0, \eta$;
- Initialization $C^{(0)}, A^{(0)}, E^{(0)}$;

Output:

Optimal sparse affinity matrix $C^{(k)}$.

- 1: **repeat**
 - 2: Update $C^{(k+1)}$ with (24) by Algorithm 1;
 - 3: Update $A^{(k+1)}$ with (26) by linear equations solution or conjugate gradient methods;
 - 4: Update $E^{(k+1)}$ with (25) by Algorithm 1;
 - 5: Update $\delta^{(k+1)} = \delta^{(k)} + \rho(A^{(k+1)T} \mathbf{1} - \mathbf{1})$;
 - 6: Update $\Delta^{(k+1)} = \Delta^{(k)} + \rho(A^{(k+1)} - C^{(k+1)})$;
 - 7: $k = k + 1$;
 - 8: **until** $\|A^{(k)} - A^{(k-1)}\|_\infty \leq \eta, \|A^{(k)T} \mathbf{1} - \mathbf{1}\|_\infty \leq \eta, \|A^{(k)} - C^{(k)}\|_\infty \leq \eta$, and $\|E^{(k)} - E^{(k-1)}\|_\infty \leq \eta$.
-

After getting the affinity matrix C , spectral clustering or Ncut methods is performed on the affinity matrix, and then the segmentation of the data in the low-dimensional space and the recovered sparse subspace are obtained. The clustering process is summarized in Algorithm 3.

Algorithm 3 Spectral Clustering for lqSSC

Input:

- Input data matrix Y .

Output:

Subspace $\{S_i\}_{i=1}^k$ and each data correspond to one of the subspaces.

- 1: Solve C with Algorithm 2;
 - 2: Construct the affinity matrix with $W = C + C'$, where $C = [C_1, C_2, \dots, C_n]$;
 - 3: Construct Laplacian matrix $L = D^{-1/2} W D^{-1/2}$, where $D = \text{diag}(d_i)$ with $d_i = \sum_{j=1}^n W_{ij}$;
 - 4: Solve the eigenvector problems: $Lx = \lambda x$ to obtain the k eigenvectors $V = [V_1, V_2, \dots, V_k]$, which correspond to the k smallest eigenvalues;
 - 5: Use k -means to to the row of V and get the clustering results.
-

5 Experiments

In this section, we evaluate our proposed method lqSSC on both synthetic and real datasets to demonstrate the efficacy. We first investigate the performance of lqSSC with random generate synthetic data of independent subspace, then carry out face clustering application on ORL database, Extend Yale B datasets¹ and motion segmentation on Hopkins 155 dataset.² All the experiments are done on an Kylin Ubuntu 15.04 system with 2.6 GHz Intel Core i5 processor and 8 GB memory using Matlab. Subspace clustering error (or accuracy), as a measure of performance of different algorithms, is defined as

$$\text{subspace clustering error} = \frac{\text{\#of misclassified points}}{\text{total \# of points}}$$

We empirically evaluate our method in comparison to four recently proposed methods which both have theoretical guarantees and satisfactory results: Sparse Subspace Clustering (SSC),³ Low-Rank Representation method (LRR),⁴ Subspace Segmentation via Quadratic Programming (SSQP),⁵ Least Square Regression (LSR).⁶ All of those comparison methods are based on spectral clustering, and can be downloaded at the author's website.

To have a fair comparison, firstly, the affinity matrix of different algorithms are learned by manually tuning the parameters to achieve the best results, then all the methods are performed the same spectral clustering method. We use the spectral clustering code provided by SSC for the benchmark. We also assume that the number of subspace of all data are provided in advance for all algorithms. In the following experiments, our algorithm utilizes $q = 0.6$ and $q = 0.3$ for the experimental comparison.

5.1 Parameter setting

For the proposed method, we follow the same experimental setting in [10] for all of the following experiments. In Algorithm 1, we initialize $W = \text{diag}(c_i)$ with any one of the solution $y_i = Y_{-i}c_i$, $\epsilon_0 = 1$, $\epsilon = \mu = 10^{-4}$, the sparsity m in Algorithm 1 is chosen with cross validation schema. In Algorithm 2, we set $\lambda_e = 2 / \min_i \max_{j \neq i} \|y_j\|_1$, $\lambda_z = 2 / \min_i \max_{j \neq i} |y'_i y_j|$.

5.2 Synthetic data

In this section, we generate a synthetic data to study the performance of the proposed method as well as the compared methods. We consider k independent subspace and the angle between each subspace is small. We randomly sample $n = 200$ data vectors from each subspace. The dimension of each subspace is $d = 5$, the first subspace is generated with Matlab command $D = \text{rand}(n, n)$, $[U, S, V] = \text{svd}(D)$, $U_1 = U(:, 1 : d)$. The bases of other subspaces can be computed by $U^{(i+1)} = T \times U^{(i)}$, $2 \leq i \leq k$, where $T \in R^{n \times n}$ is a random rotational matrix and $U^{(i)} \in R^{n \times d}$ is a random matrix with orthogonal columns.

¹<http://people.cs.uchicago.edu/~xiaofei/>

²<http://vision.jhu.edu/data>

³<http://www.eecs.berkeley.edu/~ehsan.elhamifar/code.htm>

⁴<https://sites.google.com/site/guangcanliu/>

⁵<http://wangshusen.github.io/publications.html>

⁶<https://sites.google.com/site/canyilu/>

Table 1 Clustering error (%) of synthetic data, the subspace is setted to 3, 5, 10

Algorithms	K=3		K=5		K=10	
	median	average	median	average	median	average
SSC	1.04	0.19	2.53	1.32	17.34	2.48
LRR	3.12	0.22	17.31	3.85	42.41	9.51
LSR	6.21	0.28	11.24	2.75	31.45	6.6
SSQP	1.01	0.13	3.19	1.22	16.28	4.34
lqSSC	0.82	0.18	2.11	1.12	13.29	2.33

Each of the algorithms are run 10 times. We set $q = 0.3$ for our proposed method

Then we generate the subspace with $S_i = U_i \times rand(d, n)$, such that the data matrix can be represented by $Y = [S_1, S_2, \dots, S_k]$. We randomly add the Gaussian noise to some of elements of the data. In our experiments we add 10 % of elements with noise, which follows standard Gaussian distribution $e \sim N(0, 1)$. We run each algorithm with the number of subspaces $k = 3, 5, 10$ independently and record the mean and median error or accuracy. The segmentation errors are shown in Table 1. Observe that when the number of subspace is small, all algorithms show satisfactory results, but when the subspace increase to 10, the clustering error increase quickly. The clustering performance of LRR and LSR deteriorate quickly, yet our proposed algorithm(lqSSC) show stable results, reducing clustering error compared to SSC.

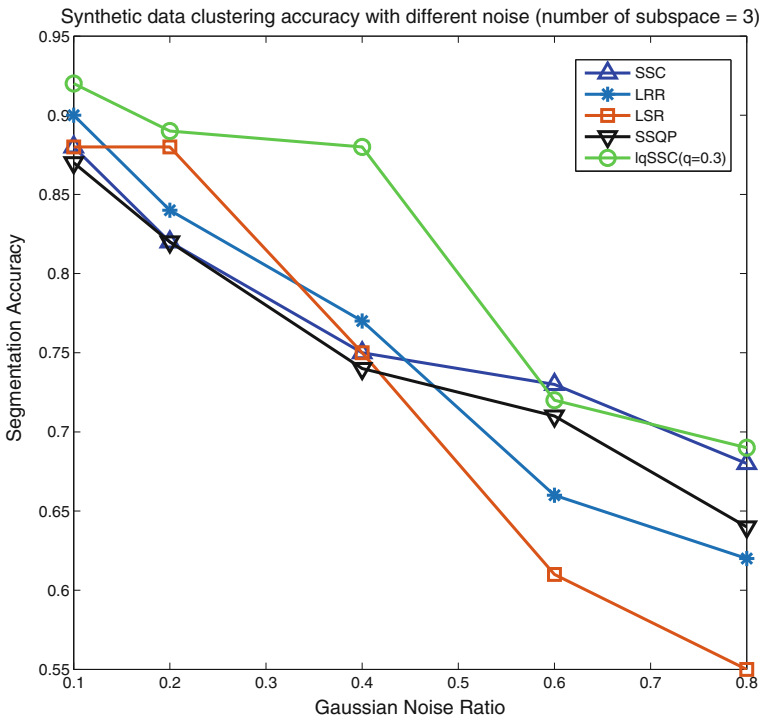


Fig. 1 Clustering error on synthetic data added with different noise ratio. The number of subspace is setted to 3, and the results are averaged from 10 runs



Fig. 2 Some examples of the images of 4 class in ORL face datasets

We then add noise to the data with different ratio that the level is from 10 to 80 %. The clustering results are shown in Fig. 1. We can see that lqSSC achieve lower clustering error compared to other methods. Moreover, our algorithm is robust to small noise ratio. When the noise ratio increase, the clustering error is comparative with SSC.

5.3 Human face clustering

Face clustering aims to group the face image data under different illumination, occlusion or noise ratio according to their subjects. The reason for studying subspace clustering to this problem origin from the point of view that the vectorized images of a given face image taken under varying illumination conditions lie approximately in a 9-dimensional linear subspace [2]. Hence the subject of face images, with varying illumination and occlusion can be well approximated by a union of low-dimensional linear subspace, each subspace contain the images corresponding to a given person. We apply the subspace clustering methods to the following two widely study datasets:

- Extended Yale Face Database B (Fig. 2), which contains 192×168 pixel images of 38 subjects of persons, each taken under 64 different illumination conditions. The dataset consist of 192×168 pixel cropped face images. In order to reduce the computational requirement, we resize the images to 48×42 pixels. The datasets has 38 individuals and around 64 near frontal images under different illuminations per individual. In our experiments we use the subjects $L = 3, 5, 10$ for clustering comparison, and the subject L is randomly chosen from the 38 subjects. We first vectorize the images and normalize it to unit vector to form the data matrix Y , each algorithm runs 10 times and record the average and median errors. We report the results in Table 2.
- The ORL face data sets (Fig. 3) contain ten different images of each of 40 distinct subjects. For some subjects, the images were taken at different time, with varying lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details



Fig. 3 Some examples of the images of 4 class in Extended Yale Database B

Table 2 The median and average error (%) of different algorithms on ORL face datasets, which apply PCA separately to the images from each subject

Data	K=5		K=10		K=20	
	median	average	median	average	median	average
LSR	23.93	19.81	32.8	26.21	67.33	56.38
LRR	13.17	14.21	27.38	16.32	42.33	38.19
SSQP	15.23	6.82	36.27	15.32	76.19	56.33
SSC	11.14	4.2	21.32	11.45	51.66	36.85
lqSSC(0.6)	10.92	3.68	21.11	11.92	45.51	29.23
lqSSC(0.3)	10.88	4.32	10.11	13.32	41.1	28.12

Boldface indicates the best performing algorithm in each experiment

(glasses / no glasses). All the images were taken in a relative dark background with the faces in an upright, frontal position (with tolerance for a little small movement). In our experiments we use $L = 5, 10, 20$ subjects for clustering comparisons. This datasets contains a small number of samples in each subjects and some of those are occluded by the glass, which makes it suitable for our experiments. We test whether the algorithms are robust to the case when the dimensions are far larger than the number of samples. We run the algorithm the same as the above setting. We report the clustering results in Table 3.

For computational efficiency, firstly, we pre-process the face datasets by performing Principal Component Analysis (PCA) separately on each subject. PCA reduces the dimensionality of the data by reserving about 98 % energy. Tables 2 and 3 report the clustering results on both datasets. The table shows the average and median clustering errors of different methods. Note that when the subjects is small, all the algorithms work well. As the subjects increase to 20, LSR and SSQP fail, they show inferior results. Note that SSC performs better than LSR, LRR and SSQP, implying that ADMM for the l_1 norm minimization is well suited for the subspace clustering. Our methods with $q = 0.3$ case, by replacing the l_1 with l_q norm minimization, improve the clustering results further. The clustering results of our method validate and coincide with recent work of l_q norm minimization. This indicate that enhancing the sparsity by l_q norm indeed improve the subspace clustering results.

Table 3 The median and average error (%) of different algorithms on Extend Yale B face data, which apply PCA separately to the images from each subject

Data	K=3		K=5		K=10	
	median	average	median	average	median	average
LRR	7.78	4.34	19.92	5.2	32.87	25.66
LSR	28.87	18.21	32.55	11.88	52.56	30.88
SSQP	19.96	5.53	28.65	9.33	45.56	43.12
SSC	3.63	1.2	4.63	2.45	11.91	9.85
lqSSC(0.6)	3.52	1.28	4.29	1.88	10.21	8.42
lqSSC(0.3)	2.77	1.12	3.22	2.81	9.91	8.2

Boldface indicates the best performing algorithm in each experiment

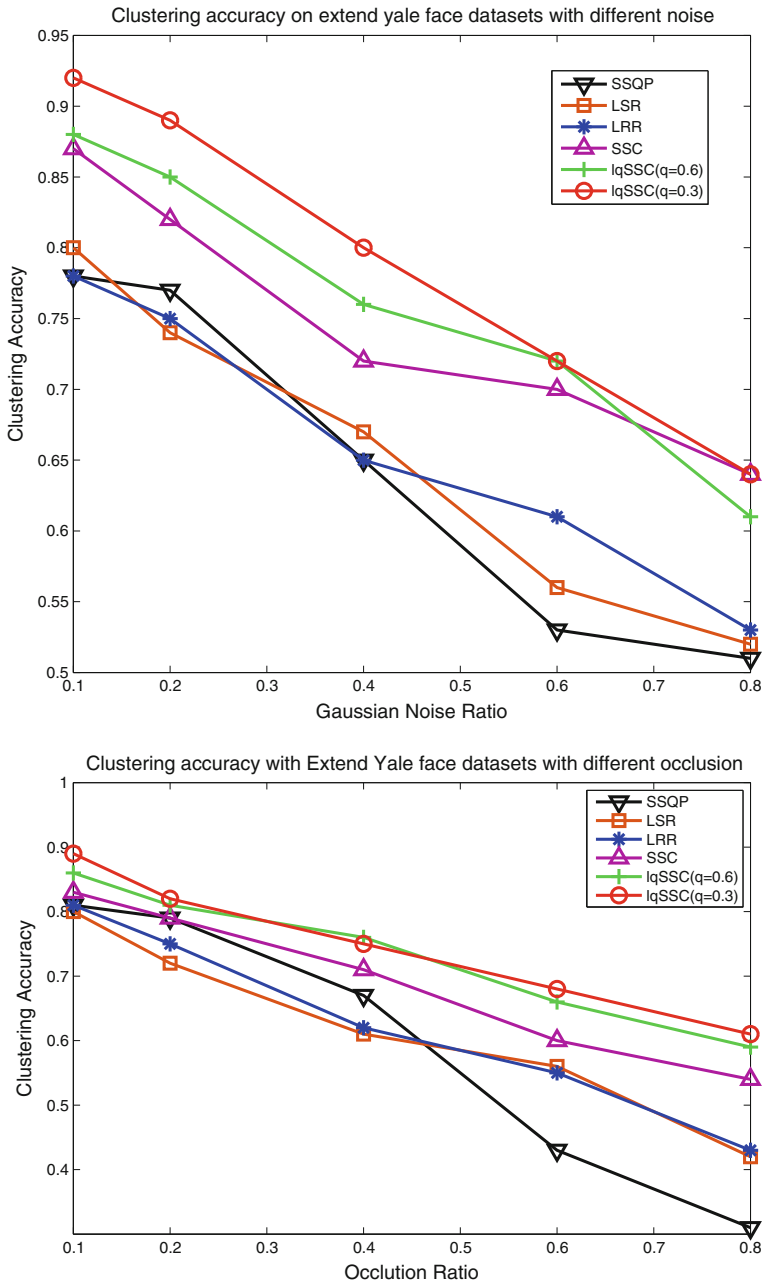


Fig. 4 Face clustering accuracy with different ratio of noise and occlusion on Extend Yale B face data, respectively. The Gaussian noise ratio levels from 10 to 80 %. The occlusion ratio is defined as the occlusion part of pixels divided by the total pixels of image

We also test the clustering accuracy with different noise and occlusions. We run all of the algorithms 10 times independently, but just record the best clustering accuracy, which is shown in Fig. 4. Our algorithm is robust to data corruption, such as noise and occlusions. We can see that our method outperforms the benchmark methods.

We then test the performance of face recovery, namely, recovery the clean face from the noise face. This experiment tests whether the algorithms can successfully recover the low-dimensional subspace when data contaminated by noise or occlusion. We now perform our proposed method to recovery the subspace of the face in the Extend Yale B data. The result is shown in Fig. 5. We can see that although the image contaminated by different ratio of illumination, Gaussian noise and occlusion, our algorithm is able to recovery the low-dimensional face subspace with quite promising results.

5.4 Motion segmentation

Motion segmentation refers to the problem of grouping the motion trajectories of multiple rigidly moving objects into spatial temporal part, such that each part corresponds to a single moving object [29]. Given a video sequence of multiple moving objects, let $\{x_{fp} \in R^2, p = 1, \dots, P, f = 1, \dots, F\}$, denotes the x-coordinate and y-coordinate in the P points and F

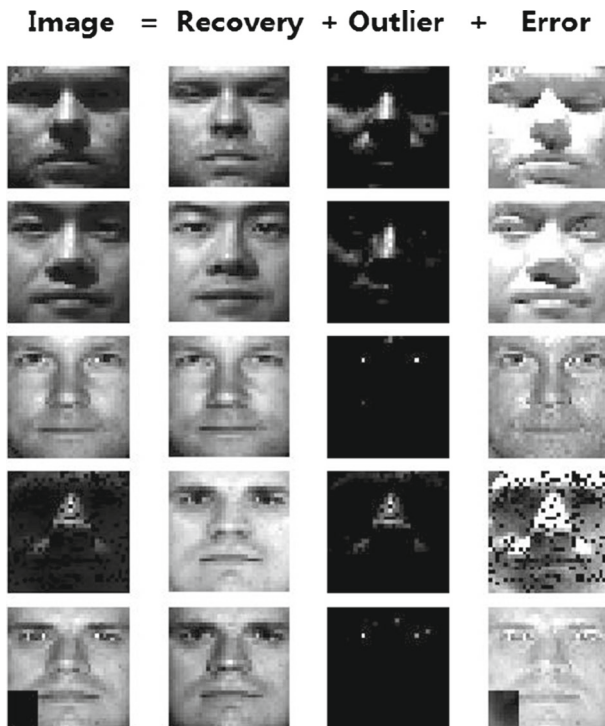


Fig. 5 Face recovery under different degree of corruption and noise. The four column corresponds to origin images, recovery images, outliers and sparse error. Each row corresponds to origin face and recovered results of the corresponding face



Fig. 6 Example frames videos in the Hopkins 155 datasets

frames of the sequences. Each data point y_i , called feature trajectory, corresponds to a $2F$ -dimensional vector that consists of stacking the feature points x_{fi} in the video as

$$y_i = [x_{1i}^T \ x_{2i}^T \ \dots \ x_{Fi}^T]^T \in R^{2F} \tag{28}$$

Under the affine projection model, all feature trajectories, formed by $y_i (i = 1, 2, \dots, N)$, lie in a union of low-dimensional subspace of dimension at most 4. As a result, the motion segmentation problem can be modelled as a subspace clustering problem on the trajectory spatial coordinates.

We use the Hopkins155 motion segmentation datasets [29] to evaluate the proposed method as well as that of state-of-the-art subspace clustering methods. Figure 6 shows some examples taken from the datasets. The datasets contain sequences with two and three motions, which can be roughly divided into three categories: Checker board sequences, which consists of 104 sequences of indoor scenes taken with a hand held camera under controlled conditions. The checker board pattern on the objects is used to assure a large number of tracked points. Traffic sequences, which consists of 38 sequences of outdoor traffic scenes taken by a moving hand held camera. Articulated/no-rigid sequences, which consists of 13 sequences displaying motions constrained by joints head and face motions, etc. The datasets consists of totally 155 video sequences, 120 of which contain two moving objects and 35 of which contain three moving objects, corresponding to 2 or 3 low dimensional subspaces of the ambient space. The point trajectories are provided in their respective datasets. Some examples of the sequence from the video are shown in Fig. 6.

In our experiment, the sequences of 2 motions have $N = 266$ feature trajectories and $F = 30$ frames, while the sequences of 3 motions have $N = 398$ feature trajectories and

Table 4 Clustering errors(%) of different algorithms on the Hopkins 155 datasets with 2F-dimensional data points

	Algorithms	SSQP	LRR	LSR	SSC	lqSSC(0.6)	lqSSC(0.3)
2 motions							
Mean		4.33	2.61	3.7	1.5	1.01	1.25
Median		0.5	0.00	0.35	0.00	0.00	0.00
3 motions							
Mean		5.37	3.82	4.9	2.31	1.32	1.01
Median		0.65	0.00	0.16	0.00	0.00	0.00

The results contain 2 motions (120 sequences) and 3 motions (35 sequences)

Table 5 Average running times (seconds) for 2 motions and 3 motions sequences

	SSQP	LRR	LSR	SSC	lqSSC(0.6)	lqSSC(0.3)
2 motions						
Check	0.31	0.93	0.15	0.28	0.26	0.29
Traffic	0.21	0.81	1.08	0.65	0.72	0.88
Articulated	0.01	0.09	0.01	0.07	0.05	0.09
3 motions						
Check	2.28	2.73	1.11	1.23	1.36	1.18
Traffic	1.23	3.09	1.98	2.06	2.42	2.88
Articulated	0.21	0.34	0.19	0.17	0.25	0.19

The time is combined with two terms, one is the average computation time of affinity matrix, the other is the average computation time of spectral clustering

$F = 29$ frames. We report the segmentation results in Table 4. All the methods show promising low segmentation clustering error. The reason behind this is that the feature trajectories is ideally lie in a low dimension subspace, such that the low dimensional subspace can be easily recovered. It is important to state that though all the methods achieve low error rate, our method outperform other methods and show a little improvement. SSC is state-of-the-art method for motion segmentation, by employing the l_q norm for the SSC, the segmentation is further improved. Powered by the l_q norm, it's not surprising that it achieves a lower segmentation error.

Finally, we test the running time of our proposed and several benchmark methods. The results are shown in Table 5. LSR achieves the lowest computational time, as it only involving the matrix computation. LRR involves more computational time due to the performing SVD at each iteration. Our algorithm iteratively solve the l_q norm minimization, it may need a little extra computational cost relative to l_1 norm based SSC, which has closed form solution. The extra running time in comparison to l_1 norm minimization is worth as it improves motion segmentation accuracy, yet the additional computational cost is inexpensive and tolerable. Though smooth IRLS iteratively find the sparse solution, it can easily parallel to accelerate the computational process. Note that the compared results of running time of our method does not employ any accelerate technique. The implementation of our method is merely written with pure Matlab code for fair of comparisons.

6 Conclusion

In this work, we study the l_q norm based sparse subspace clustering method, building upon the recent work that l_q norm can enhance sparsity compared to traditional convex regularizations. For better solving this problem, the ADMM based framework is employed to split the problem into multiple terms, such that the non-convex and non-smooth l_q norm term can be solved by the smooth IRLS algorithm. We analysis the mechanism of smooth IRLS algorithm for l_q minimization and its convergence property, then unify ADMM and smooth IRLS for solving l_q norm based SSC problem. Compared to traditional convex optimization, the proposed method achieves higher clustering accuracy. The benefits of our model is that, it can easily extend to many non-convex and non-smooth regularization problems, such as SCAD, LSP, Logarithm and Capped norm regularizations(see [36] and reference therein), which all provide natural procedures for sparse recovery, but have not

been well developed for subspace clustering problem. We believe that the study of unifying ADMM and smooth IRLS for l_q SSC will also suit for those non-convex non-smooth regularizations. As future work, we plan to study those regularizations for the SSC problem under the ADMM and IRLS framework. We believe that the proposed method will shed light on the further research of other non-convex regularizations. Though our algorithm achieves higher clustering accuracy, it results in additional computational time. We aim at designing and investigating more efficient algorithm to overcome this problem in future.

Appendix: Proof of the Theory 1

To prove Theorem 1, let $u = z^{(j)} - \frac{1}{\rho}w^{(j)}$, and u is a constant here. Firstly, for $j = 1, \dots, d$

$$y_j = x_j^{(k)2} + \epsilon_k^2, \quad z_j = x_j^{(k+1)2} + \epsilon_{k+1}^2$$

Then

$$\begin{aligned} & J(x^{(k)}, \epsilon_k) - J(x^{(k+1)}, \epsilon_{k+1}) \\ &= \sum_{j=1}^d (y_j^{q/2} - z_j^{q/2}) + \rho/2(\|x^{(k)} - u\|_2^2 - \|x^{(k+1)} - u\|_2^2) \end{aligned} \tag{29}$$

Note that $f(x) = x^{q/2}$ ($0 < q < 1$) is a concave function, for any $y, z \in R^1$

$$f(y) - f(z) \geq \frac{q}{2}y^{\frac{q}{2}-1}(y - z) \tag{30}$$

Using (30), we have

$$\begin{aligned} & J(x^{(k)}, \epsilon_k) - J(x^{(k+1)}, \epsilon_{k+1}) \\ &= \sum_{j=1}^d (y_j^{q/2} - z_j^{q/2}) + \frac{\rho}{2}(\|x^{(k)} - u\|_2^2 - \|x^{(k+1)} - u\|_2^2) \end{aligned} \tag{31}$$

By using the rule of (30), the left part of (31) can be transformed as

$$\begin{aligned} & \sum_{j=1}^d y_j^{q/2} - z_j^{q/2} \geq \frac{q}{2} \sum_{j=1}^d W_{jj}^k (x_j^{(k)2} - x_j^{(k+1)2}) \\ &= \frac{q}{2} \sum_{j=1}^d W_{jj}^k (x_j^{(k)} - x_j^{(k+1)})^2 + q \sum_{j=1}^d W_{jj}^{(k)} (x_j^{(k)} - x_j^{(k+1)})x_j^{(k+1)} \\ &= \frac{q}{2} \sum_{j=1}^d W_{jj}^k (x_j^{(k)} - x_j^{(k+1)})^2 + q(x^{(k)} - x^{(k+1)})'W^{(k)}x^{(k+1)} \end{aligned} \tag{32}$$

We now consider the (18), multiplying $(x^{(k)} - x^{(k+1)})'$ on both sides

$$q(x^{(k)} - x^{(k+1)})^T W^{(k)}x^{(k+1)} + \rho(x^{(k)} - x^{(k+1)})'(x^{(k+1)} - u) = 0 \tag{33}$$

Simplify (32) and get (33), then convert it as follows

$$\begin{aligned} & \sum_{j=1}^d y_j^{q/2} - z_j^{q/2} \\ & \geq \frac{q}{2} \sum_{j=1}^d W_{jj}^k (x_j^{(k)} - x_j^{(k+1)})^2 - \rho(x^{(k)} - x^{(k+1)})^T (x^{(k+1)} - u) \end{aligned} \tag{34}$$

Note that $f(x) = x^2$ is a concave function, then for all $y, z \in \mathbb{R}^d$

$$f(y) - f(z) \geq 2z(y - z)$$

By employing this inequality, we have

$$\|x^{(k)} - u\|_2^2 - \|x^{(k+1)} - u\|_2^2 \geq 2(x^{(k)} - x^{(k+1)})'(x^{(k+1)} - u) \tag{35}$$

Summarized (31), (34), (35)

$$J(x^{(k)}, \epsilon_k) - J(x^{(k+1)}, \epsilon_{k+1}) \geq 0$$

This prove that $J(x, \epsilon)$ is an decreasing sequence. Note that

$$0 < \|x^{(k)}\|_q^q \leq \|x^{(k)}\|_{q, \epsilon_k}^q \leq J(x^{(k)}, \epsilon_k) \leq J(x^{(0)}, \epsilon_0) \tag{36}$$

Thus the sequence $\{x^{(k)}\}$ is bounded. Furthermore, if $\epsilon > 0$, the boundedness of $\{x^{(k)}\}$ implies that there exists a subsequence $\{x^{(k_j)}\}$ converging to some point $x^{\epsilon*}$. Note that $\|x^{(k+1)} - x^{(k)}\|_2 \rightarrow 0$, thus the subsequence $x^{(k_j)}$ also converges to $x^{\epsilon*}$. Consider the subsequence in the (18)

$$qW^{(k_j)}x^{(k_j)} + \rho(x^{(k_j)} - u) = 0$$

Let $k_j \rightarrow \infty$, we get

$$qW^{\epsilon*}x^{\epsilon*} + \rho(x^{\epsilon*} - u) = 0 \tag{37}$$

Therefore, $x^{\epsilon*}$ is a critical point of (18).

Acknowledgments This work is partially supported by NSF of China under Grant 61672548, 61173081, and the Guangzhou Science and Tech-nology Program, China, under Grant 201510010165.

References

1. Basri R, Jacobs DW (2001) Lambertian reflectance and linear subspaces. International Conference on Computer Vision
2. Basri R, Jacobs DW (2003) Lambertian reflectance and linear subspaces. IEEE Trans Pattern Anal Mach Intell 25.2:218–233
3. Boyd S et al. (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. Found Trends Mach Learn 3.1:1–122
4. Candes EJ, Wakin MB, Boyd SP (2008) Enhancing sparsity by reweighted l1 minimization. J Fourier Anal Appl 14.5–6:877–905
5. Cheng W, Chow TWS, Zhao M (2016) Locality constrained-p sparse subspace clustering for image clustering[J]. Neurocomputing 205:22–31
6. Daubechies I et al. (2010) Iteratively reweighted least squares minimization for sparse recovery. Commun Pure Appl Math 63.1:1–38
7. Deng Y et al. (2013) Low-rank structure learning via nonconvex heuristic recovery. IEEE Trans Neural Netw Learn Syst 24.3:383–396
8. Dyer EL, Sankaranarayanan AC, Baraniuk RG (2013) Greedy feature selection for subspace clustering[J]. J Mach Learn Res 14(1):2487–2517
9. Dyer EL et al. (2015) Self-expressive decompositions for matrix approximation and clustering. arXiv preprint arXiv:1505.00824

10. Elhamifar E, Vidal R (2013) Sparse subspace clustering: algorithm, theory, and applications. *IEEE Trans Pattern Anal Mach Intell* 35.11:2765–2781
11. Feng J, Lin Z, Xu H et al. (2014) Robust subspace segmentation with block-diagonal prior[C]. In: *IEEE Conference on computer vision and pattern recognition (CVPR)*, 2014. IEEE, pp 3818–3825
12. Fornasier M, Rauhut H, Ward R (2011) Low-rank matrix recovery via iteratively reweighted least squares minimization. *SIAM J Optim* 21.4:1614–1640
13. Fornasier M et al. (2015) Conjugate gradient acceleration of iteratively re-weighted least squares methods. *arXiv preprint arXiv:1509.04063*
14. Guo S, Wang Z, Ruan Q (2013) Enhancing sparsity via l_p ($0 < p < 1$) minimization for robust face recognition. *Neurocomputing* 99:592–602
15. He R et al. (2015) Robust subspace clustering with complex noise. *IEEE Trans Image Process* 24.11:4001–4013
16. Heckel R, Bölcskei H (2013) Robust subspace clustering via thresholding[J]. *arXiv preprint arXiv:1307.4891*
17. Heckel R, Tschannen M, Bölcskei H (2015) Dimensionality-reduced subspace clustering[J]. *arXiv preprint arXiv:1507.07105*
18. Huang S-Y, Yeh Y-R, Eguchi S, Candes EJ, Li X, Ma Y, Wring J (2009) Robust principal component analysis. *J Assoc Comput Mach* 53(3):3179–213
19. Hund M, Böhm D, Sturm W et al. (2016) Visual analytics for concept exploration in subspaces of patient groups[J]. *Brain Inform*:1–15
20. Lai M-J, Wang J (2011) An unconstrained l_q minimization with l_q for sparse solution of underdetermined linear systems. *SIAM J Optim* 21.1:82–101
21. Lai M-J, Yangyang X, Yin W (2013) Improved iteratively reweighted least squares for unconstrained smoothed l_q minimization. *SIAM J Numer Anal* 51.2:927–957
22. Liu G, Lin Z, Yong Y (2010) Robust subspace segmentation by low-rank representation. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*
23. Liu J, Chen Y, Zhang J et al. (2014) Enhancing low-rank subspace clustering by manifold regularization[J]. *IEEE Trans Image Process* 23(9):4022–4030
24. Lu C (2012) Robust and efficient subspace segmentation via least squares regression, *ECCV*. pp 1–14
25. Lu C, Lin Z, Yan S (2015) Smoothed low rank and sparse matrix recovery by iteratively reweighted least squares minimization. *IEEE Trans Image Process* 24.2:646–654
26. Peng X, Yi Z, Tang H (2015) Robust subspace clustering via thresholding ridge regression. In: *AAAI Conference on artificial intelligence (aaai)*
27. Soltanolkotabi M, Elhamifar E, Candes EJ (2014) Robust subspace clustering. *Ann Stat* 42.2:669–699
28. Sui Y, Zhang S, Zhang L (2015) Robust visual tracking via sparsity-induced subspace learning[J]. *IEEE Trans Image Process* 24(12):4686–4700
29. Tron R, Vidal R (2007) A benchmark for the comparison of 3-d motion segmentation algorithms. In: *IEEE Conference on computer vision and pattern recognition, 2007. CVPR'07*. IEEE
30. Vidal R, Favaro P (2014) Low rank subspace clustering (LRSC). *Pattern Recog Lett* 43:47–61
31. Wang S, Yuan X, Yao T, Yan S (2011) Efficient subspace segmentation via quadratic programming. In: *Proc. Twenty-Fifth AAAI conf. artif. intell.*, pp 519–524
32. Wen J, Li D, Zhu F (2015) Stable recovery of sparse signals via l_p -minimization. *Appl Comput Harm Anal* 38.1:161–176
33. Xu J et al. (2015) Reweighted sparse subspace clustering. *Comput Vis Image Understand*
34. Yang AY et al. (2013) Fast-minimization algorithms for robust face recognition. *IEEE Trans Image Process* 22.8:3234–3246
35. You C, Vidal R (2015) Subspace-sparse representation. *arXiv preprint arXiv:1507.01307*
36. Zhang CH, Zhang T (2012) A general theory of concave regularization for high-dimensional sparse estimation problems[J]. *Stat Sci*:576–593
37. Zhang Y et al. (2013) Robust subspace clustering via half-quadratic minimization. In: *2013 IEEE International conference on computer vision (ICCV)*. IEEE



Shenfen Kuang is a Ph.D. Candidate in the school of data and computer science, Sun yat-sen university. His research interest is machine learning and computer vision, especially on low rank optimization and subspace learning. He received M.S. degree from Sun yat-sen university in 2010 . He is also a lecturer at the department of statistics of shaoguan university since 2010.



Hongyang Chao received the B.S. degree from Sun Yet-sen University, Guangzhou, China, and the Ph.D. degree from Sun Yet-sen University, Guangzhou, China, both in Computational Mathematics. From 1988 to 1994, she joined the Department of Computer Science of the Sun Yet-sen University at Guangzhou, China, where she was an Assistant Professor first and later Associate Professor. She presently is an Associate Dean and a full Professor in the School of Software. She has published extensively in the area of image/video processing and holds three US patents and 2 China patents in video coding. Her current research interests include the areas of image and video processing, image and video compression, massive multimedia data analysis and understanding, content based image (video) retrieval.



Jun Yang is a Ph.D. candidate in the School of Information Science and Technology, Sun Yat-sen University, China. He received his M.S. degree from Guangdong University of Technology in 2010. His current research interests include image processing, pattern recognition and machine learning.