Efficient Subspace Segmentation via Quadratic Programming

Shusen Wang[1], **Xiaotong Yuan**[2], **Tiansheng Yao**[1], **Shuicheng Yan**[2], **Jialie Shen**[3]
[1]College of Computer Science and Technology, Zhejiang University, China
[2]Department of Electrical and Computer Engineering, National University of Singapore, Singapore
[3]School of Information Systems, Singapore Management University, Singapore
wssatzju@gmail.com, eleyuanx@nus.edu.sg, tsyaoo@gmail.com, eleyans@nus.edu.sg, jlshen@smu.edu.sg

Abstract

We explore in this paper efficient algorithmic solutions to robust subspace segmentation. We propose the SSQP, namely *Subspace Segmentation via Quadratic Programming*, to partition data drawn from multiple subspaces into multiple clusters. The basic idea of SSQP is to express each datum as the linear combination of other data regularized by an overall term targeting zero reconstruction coefficients over vectors from different subspaces. The derived coefficient matrix by solving a quadratic programming problem is taken as an affinity matrix, upon which spectral clustering is applied to obtain the ultimate segmentation result. Similar to sparse subspace clustering (SCC) and low-rank representation (LRR), SSQP is robust to data noises as validated by experiments on toy data. Experiments on Hopkins 155 database show that SSQP can achieve competitive accuracy as SCC and LRR in segmenting affine subspaces, while experimental results on the Extended Yale Face Database B demonstrate SSQP's superiority over SCC and LRR. Beyond segmentation accuracy, all experiments show that SSQP is much faster than both SSC and LRR in the practice of subspace segmentation.

Introduction

Subspace segmentation is an important clustering problem with numerous applications in machine learning and computer vision literature, e.g. image compression (Hong et al. 2006), motion segmentation (Tron and Vidal 2007; Rao et al. 2008), and image clustering under varying illuminations (Ho et al. 2003). The problem is formally defined as follows:

Definition 1. *(Subspace Segmentation) Given sufficient data* $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N] \in \mathbb{R}^{D \times N}$ *drawn from* n *(*n *is known or unknown) linear subspaces* $\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_n \subset \mathbb{R}^D$ *of arbitrary dimensions, the task is to segment the data according to the underlying subspaces they are drawn from.*

Prior work

Subspace segmentation problem has been widely studied and many algorithms have been proposed in the past two decades. These algorithms can be roughly divided into five categories: matrix factorization based methods, algebraic methods, iterative methods, statistical methods, and spectral clustering based methods. We give a brief review of these categories of methods in this subsection. For more complete descriptions, we refer the readers to a recent survey (Vidal 2010) and the references therein.

Matrix factorization based methods (Boult and Brown 1991; Costeira and Kanade 1998) factorize the data matrix as the product of an orthogonal bases matrix and another low rank representation matrix. These methods seek to reveal the clusters from the structure of the factorization results, and work well if the subspaces are linearly independent and the data are noise free. These methods are however not robust to noises and outliers, and easy to fail if the independent subspace assumption is violated.

Generalized PCA (GPCA) (Vidal, Ma, and Sastry 2005) is a popular algebraic method for modeling and segmenting mixed data using a collection of subspaces. GPCA is motivated by the fact that one can fit a union of n subspaces with a set of polynomials of degree n, whose derivatives at a point $\mathbf{x}$ give a vector normal to the subspace containing point $\mathbf{x}$. Thus the subspace segmentation is equivalent to fitting the data with polynomials and feature clustering via polynomial differentiation (Vidal and Hartley 2004). However, this method is also sensitive to noises and outliers, and its practical performance is often not satisfactory.

Iterative methods, such as K-subspace clustering (Ho et al. 2003; Agarwal and Mustafa 2004), iteratively assign data to the nearest subspaces followed by updating the subspaces. The advantages of K-subspace lie in its simplicity and finite termination. The disadvantages are that the algorithmic convergence is local and these methods are sensitive to outliers.

Random Sample Consensus (RANSAC) (Fischler and Bolles 1981) and Agglomerative Lossy Compression (ALC) (Ma et al. 2007) are two statistical approaches by making explicit assumptions on the distribution of the data. Both of them can handle noises and outliers, and they need not to know the number of subspaces beforehand. However, for both RANSAC and ALC, determining the number of subspaces depends on one parameter, which needs to be carefully tuned for obtaining an right subspace number. Another major disadvantage of RANSAC is that the dimensions of the subspaces must be equal, which limits the application of RANSAC to some extent. The most critical problem with ALC is the lack of theoretic guarantee for the optimality of the agglomerative procedure.

Copyright © 2011, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

SSC and LRR Sparse subspace clustering (SSC) (Elhamifar and Vidal 2009) and low-rank representation (LRR) (Liu, Lin, and Yu 2010) are two spectral clustering based methods, and they are the most effective approaches to subspace segmentation so far (Liu, Lin, and Yu 2010; Vidal 2010). LRR may achieve even better performance than SSC in presence of heavy noises.

The motivations of these two methods are similar: they express each datum $\mathbf{x}_i \in \mathcal{S}_\alpha$ as the linear combination of all other data $\mathbf{x}_i = \sum_{j \neq i} z_{ij}\mathbf{x}_j$, and implicitly enforce those coefficients z_{ij} to be zero for all $\mathbf{x}_j \notin \mathcal{S}_\alpha$ under certain assumptions. That is to learn a coefficient matrix $\mathbf{Z} \in \mathbb{R}^{N\times N}$ such that $z_{ij} = 0$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to different subspaces. Both SSC and LRR solve the subspace segmentation problem by optimizing certain convex criteria to learn such a coefficient matrix $\mathbf{Z}$. Formally, SSC solves Problem (1),

$$\begin{array}{ll} \min_{\mathbf{Z}} & \|\mathbf{XZ} - \mathbf{X}\|_F^2 + \lambda\|\mathbf{Z}\|_1 \\ \text{s.t.} & \mathsf{diag}(\mathbf{Z}) = \mathbf{0}\,, \end{array} \tag{1}$$

and LRR solves Problem (2),

$$\min_{\mathbf{Z}} \quad \|\mathbf{XZ} - \mathbf{X}\|_1 + \lambda\|\mathbf{Z}\|_*, \tag{2}$$

where $\|\mathbf{X}\|_F = \sqrt{\sum_i \sum_j \mathbf{x}_{ij}^2}$ is the Frobenius norm of X, $\|\mathbf{X}\|_1 = \sum_i \sum_j |\mathbf{x}_{ij}|$ denotes ℓ_1-norm, $\mathsf{diag}(Z)$ is the diagonal vector of matrix Z, and the nuclear norm $\|\mathbf{X}\|_*$ is defined to be the sum of all singular values of $\mathbf{X}$.

SSC and LRR are characterized as two-step algorithms. In the first step, they both obtain the coefficient matrix $\mathbf{Z}$ via convex optimization. The difference between these two methods lies in the regularization on $\mathbf{Z}$: SSC enforces $\mathbf{Z}$ to be sparse by imposing an ℓ_1-norm regularization on $\mathbf{Z}$, while LRR encourages $\mathbf{Z}$ to be of low-rank by the nuclear norm regularization. The second step is generally to take the matrix $|\mathbf{Z} + \mathbf{Z}^T|/2$, which is symmetric and entrywise nonnegative, as the affinity matrix upon which spectral clustering is applied to get the ultimate segmentation results. Both SSC and LRR are robust against considerable noises and outliers.

Despite the good performance and robustness, SSC and LRR both suffer from heavy computational burdens when calculating $\mathbf{Z}$. In the optimization step, SSC solves the constrained lasso regression problem, which is non-smooth and demands much more computational cost each iteration than general smooth problems. The LRR solves an even more complicated convex optimization problem of minimizing a weighted sum of nuclear norm and ℓ_1-norm (or $\ell_{2,1}$-norm). Though the augmented lagrange multiplier method (ALM) (Lin et al. 2009) is an efficient approach to solve this kind of problems, LRR still requires one singular value decomposition operation each iteration and often iterates hundreds of times before convergence. When applied to large scale problems, optimizing Problem (1) and Problem (2) is quite time consuming. For this reason, we seek to propose a new solution which is computationally cheap while its performance is competitive with SSC and LRR.

Our contributions

In this paper, we propose a novel subspace segmentation method, *Subspace Segmentation via Quadratic Programming* (SSQP), which is efficient while only requires to solve a convex quadratic programming problem for optimization. In analog to SSC and LRR, our motivation is to find an optimization formulation for learning an affinity matrix $\mathbf{Z}$ such that $z_{ij} = 0$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ lie in different subspaces. With the affinity matrix $\mathbf{Z}$ learned, spectral clustering can be employed to segment the data into clusters targeting the consistency with the underlying subspaces they are drawn from.

The major contribution of SSQP is the new regularization item $\|\mathbf{Z}^T\mathbf{Z}\|_1$. Such a regularizer can be theoretically proven to enforce $\mathbf{Z}$ to have diagonal structure under certain assumptions, as will described shortly in Theorem 2. Different from SSC and LRR that require time consuming sparse or low rank optimization techniques, SSQP solves a quadratic program with box constraints, for which several off-the-shelf scalable solvers are publicly available.

We compare SSQP with SSC and LRR in both segmentation error rate and computation time. Experiments on Hopkins 155 Database (Tron and Vidal 2007) and the Extended Yale Face Database B (Georghiades, Belhumeur, and Kriegman 2001) show that SSQP method can achieve competitive segmentation accuracy as SSC and LRR, while SSQP is much more efficient in computation than SSC and LRR.

Subspace Segmentation via Quadratic Programming

In this section, we first formally describe SSQP as a quadratic program whose solution $\mathbf{Z}^*$ can be further used as the affinity matrix for spectral clustering. Then we prove that $\mathbf{Z}^*$ is a product of a block diagonal matrix and a permutation matrix if the subspaces are orthogonal or the weighting parameter λ is large enough. We then present a procedure for solving this quadratic programming problem.

Quadratic programming formulation

The basic idea of SSQP is to express each datum $\mathbf{x}_i \in \mathcal{S}_\alpha$ as the linear combination $\mathbf{x}_i = \sum_{j\neq i} z_{ij}\mathbf{x}_j$ along with an overall regularization term on $\mathbf{Z}$ to enforce the coefficients of vectors $\mathbf{x}_j \notin \mathcal{S}_\alpha$ to be zero under certain assumptions. To take the possible noises into consideration, we express $\mathbf{X}$ to be $\mathbf{X} = \mathbf{XZ} + \mathbf{Y}$, where $\mathbf{Y}$ is the noise term. We take the Frobenius norm on $\mathbf{Y}$ as the loss function to penalize misfit and use $\|\mathbf{Z}^T\mathbf{Z}\|_1$ to enforce the block diagonal structure of $\mathbf{Z}^*$. The final formulation is given by the following constrained optimization problem:

$$\begin{array}{ll} \min_{\mathbf{Z}} & f(\mathbf{Z}) = \|\mathbf{XZ} - \mathbf{X}\|_F^2 + \lambda\|\mathbf{Z}^T\mathbf{Z}\|_1 \\ \text{s.t.} & \mathbf{Z} \geq \mathbf{0}; \\ & \mathsf{diag}(\mathbf{Z}) = \mathbf{0}\,. \end{array} \tag{3}$$

Note that $\|\mathbf{Z}^T\mathbf{Z}\|_1$ is equivalent to $\mathbf{e}^T\mathbf{Z}^T\mathbf{Z}\mathbf{e}$, where $\mathbf{e}$ is an all-one vector. Thus Problem (3) is a linear constrained quadratic programming, which can be solved by several off-the-shelf solvers efficiently.

We now give some justifications on the regularization and constraints on Z. First, as shown later in Theorem 2, the regularization term $\|\mathbf{Z}^T\mathbf{Z}\|_1$ with $\mathbf{Z}$ being nonnegative can enforce z_{ij} to be zero if $\mathbf{x}_i$ and $\mathbf{x}_j$ lie in different subspaces.

Secondly, minimizing $\|\mathbf{Z}^T\mathbf{Z}\|_1$ may potentially enforce the sparsity of $\mathbf{Z}^*$. Since $\mathbf{Z}$ is nonnegative, $\|\mathbf{Z}^T\mathbf{Z}\|_1 =$

$\sum_{i,j} \mathbf{z}_i^T \mathbf{z}_j$, minimizing which encourages $\mathbf{z}_i$ and $\mathbf{z}_j$ to be both sparse such that the inner product of $\mathbf{z}_i$ and $\mathbf{z}_j$ tends to be zero. Though there is currently no theoretic proof that the regularization $\|\mathbf{Z}^T\mathbf{Z}\|_1$ can guarantee the sparsity of the solution $\mathbf{Z}^*$, our off-line experiments empirically show that $\mathbf{Z}^*$ is very sparse (even "sparser" than the solution of SSC).

Finally, the nonnegativity of the coefficient matrix $\mathbf{Z}$ provides better interpretations for the consequent clustering process. The nonnegative constraint requires the data to be the nonnegative linear combination of other data, which depicts a "positive" correlation between any two data vectors. It is thereby more interpretable when applied to problems such as face clustering where two faces are clustered into the same group only if the two faces are positively correlated.

Block diagonal property

The following theorem indicates the block diagonal structure of the solution to Problem (3) under the orthogonal linear subspaces assumption. We will also show that without the orthogonal subspace assumption, the solution may still enjoys the block diagonal property when λ is large enough.

Theorem 2. *Let $\mathbf{X} \in \mathbb{R}^{D\times N}$ be a matrix whose columns are drawn from a union of n orthogonal linear subspaces $\{\mathcal{S}_1, \mathcal{S}_2, \cdots \mathcal{S}_n\}$. Let $\mathbf{\Gamma}$ be a permutation matrix such that $\tilde{\mathbf{X}} = \mathbf{X}\mathbf{\Gamma} = [\tilde{\mathbf{X}}_1, \tilde{\mathbf{X}}_2, \cdots, \tilde{\mathbf{X}}_n]$, where $\tilde{\mathbf{X}}_i$ is an $D \times N_i$ matrix whose columns lie in the same subspace $\mathcal{S}_i$, and $N_1 + \cdots + N_n = N$. The solution to the optimization Problem* (3) *satisfies the property that $\mathbf{\Gamma}^{-1}\mathbf{Z}^*\mathbf{\Gamma}$ is block diagonal*

$$\tilde{\mathbf{Z}}^* = \mathbf{\Gamma}^{-1}\mathbf{Z}^*\mathbf{\Gamma} = \begin{pmatrix} \tilde{\mathbf{Z}}_1^* & & & \mathbf{0} \\ & \tilde{\mathbf{Z}}_2^* & & \\ & & \ddots & \\ \mathbf{0} & & & \tilde{\mathbf{Z}}_n^* \end{pmatrix}_{N\times N}$$

where submatrix $\tilde{\mathbf{Z}}_i^* \in \mathbb{R}^{N_i \times N_i}$.

Proof. Let $\tilde{\mathbf{Z}} = \mathbf{\Gamma}^{-1} Z \mathbf{\Gamma}$. From the properties of permutation matrices, we have $\mathbf{\Gamma}^{-1} = \mathbf{\Gamma}^T$ and $\mathbf{\Gamma}\mathbf{e} = \mathbf{e}$, thus

$$\begin{aligned} f(\mathbf{Z}) &= \|\mathbf{X}\mathbf{Z} - \mathbf{X}\|_F^2 + \lambda \mathbf{e}^T \mathbf{Z}^T \mathbf{Z} \mathbf{e} \\ &= \|\tilde{\mathbf{X}}\tilde{\mathbf{Z}} - \tilde{\mathbf{X}}\|_F^2 + \lambda \mathbf{e}^T \tilde{\mathbf{Z}}^T \tilde{\mathbf{Z}} \mathbf{e} \,. \end{aligned} \tag{4}$$

Hence if we take $\tilde{\mathbf{X}}$ as the the data matrix instead of $\mathbf{X}$, then the solution to Problem (3) is $\tilde{\mathbf{Z}}^*$.

Therefore we assume, without loss of generality, the columns of $\mathbf{X}$ are in general position: $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_n]$, where all the columns of submatrix $\mathbf{X}_\alpha$ lie in the same subspace $\mathcal{S}_\alpha$.

Assume $\mathbf{Z}^*$ is the optimal solution, and we decompose $\mathbf{Z}^*$ to be the sum of two matrices

$$\begin{aligned} \mathbf{Z}^* &= \mathbf{Z}^D + \mathbf{Z}^C \\ &= \begin{pmatrix} \mathbf{Z}_{11}^* & & & \mathbf{0} \\ & \mathbf{Z}_{22}^* & & \\ & & \ddots & \\ \mathbf{0} & & & \mathbf{Z}_{nn}^* \end{pmatrix} \\ &+ \begin{pmatrix} \mathbf{0} & \mathbf{Z}_{12}^* & \cdots & \mathbf{Z}_{1n}^* \\ \mathbf{Z}_{21}^* & \mathbf{0} & \cdots & \mathbf{Z}_{2n}^* \\ \vdots & & & \vdots \\ \mathbf{Z}_{n1}^* & \mathbf{Z}_{n2}^* & \cdots & \mathbf{0} \end{pmatrix} \end{aligned} \tag{5}$$

where $\mathbf{Z}_{i,j}^* \in \mathbb{R}^{N_i \times N_j}$. Note that both $\mathbf{Z}^D$ and $\mathbf{Z}^C$ are nonnegative.

According to the decomposition of $\mathbf{Z}^*$, any column of $\mathbf{Z}^*$ can be written as $\mathbf{z}_i^* = \mathbf{z}_i^D + \mathbf{z}_i^C$, with $\mathbf{z}_i^D$ and $\mathbf{z}_i^C$ supported on disjointed subset of indices. We can write $\|\mathbf{X}\mathbf{Z}^* - \mathbf{X}\|_F^2$ as

$$\begin{aligned} & \|\mathbf{X}\mathbf{Z}^* - \mathbf{X}\|_F^2 \\ = \; & \textstyle\sum_{i=1}^N \|\mathbf{X}\mathbf{z}_i^* - \mathbf{x}_i\|_2^2 \\ = \; & \textstyle\sum_{i=1}^N \|\mathbf{X}\mathbf{z}_i^D - \mathbf{x}_i + \mathbf{X}\mathbf{z}_i^C\|_2^2 \\ = \; & \textstyle\sum_{i=1}^N \|\mathbf{X}\mathbf{z}_i^D - \mathbf{x}_i\|_2^2 + \sum_{i=1}^N \|\mathbf{X}\,\mathbf{z}_i^C\|_2^2 \\ & + 2\textstyle\sum_{i=1}^N \cos\theta_i \cdot \|\mathbf{X}\mathbf{z}_i^D - \mathbf{x}_i\|_2 \cdot \|\mathbf{X}\mathbf{z}_i^C\|_2 \end{aligned} \tag{6}$$

where θ_i is the angle between vector $\mathbf{X}\mathbf{z}_i^D - \mathbf{x}_i$ and $\mathbf{X}\mathbf{z}_i^C$.

Since the matrix $\mathbf{X} = [\mathbf{X}_1, \cdots \mathbf{X}_n]$ is well arranged, any column $\mathbf{x}_i \in \mathbf{X}_\alpha$ and $\mathbf{x}_j \in \mathbf{X}_\beta$ lie in different subspaces if $\alpha \neq \beta$. Let $\mathbf{x}_i \in \mathcal{S}_\alpha$, according to the definition of $\mathbf{z}_i^D$ and $\mathbf{z}_i^C$, we have $\mathbf{X}\mathbf{z}_i^D \in \mathcal{S}_\alpha$ and $\mathbf{X}\mathbf{z}_i^C \notin \mathcal{S}_\alpha$. Based on the orthogonal subspace assumption, we have $(\mathbf{X}\mathbf{z}_i^D - \mathbf{x}_i) \perp \mathbf{X}\mathbf{z}_i^C$ and $\theta_i = \pi/2$, thus

$$\begin{aligned} \|\mathbf{X}\mathbf{Z}^* - \mathbf{X}\|_F^2 &= \|\mathbf{X}\mathbf{Z}^D - \mathbf{X}\|_F^2 + \|\mathbf{X}\mathbf{Z}^C\|_F^2 \\ &\geq \|\mathbf{X}\mathbf{Z}^D - \mathbf{X}\|_F^2 \,. \end{aligned} \tag{7}$$

Based on the nonnegativity of $\mathbf{Z}^*$, $\mathbf{Z}^C$, and $\mathbf{Z}^D$, we have

$$\begin{aligned} \|\mathbf{Z}^{*T}\mathbf{Z}^*\|_1 &= \textstyle\sum_{i,j} |\mathbf{z}_i^{*T}\mathbf{z}_j^*| = \sum_{i,j} \mathbf{z}_i^{*T}\mathbf{z}_j^* \\ &= \textstyle\sum_{i,j} (\mathbf{z}_i^D + \mathbf{z}_i^C)^T (\mathbf{z}_j^D + \mathbf{z}_j^C) \\ &\geq \textstyle\sum_{i,j} \mathbf{z}_i^{D^T}\mathbf{z}_j^D + \sum_{i,j} \mathbf{z}_i^{C^T}\mathbf{z}_j^C \\ &= \|\mathbf{Z}^{D^T}\mathbf{Z}^D\|_1 + \|\mathbf{Z}^{C^T}\mathbf{Z}^C\|_1 \\ &\geq \|\mathbf{Z}^{D^T}\mathbf{Z}^D\|_1 \,. \end{aligned} \tag{8}$$

From inequalities (7) and (8) we have $f(\mathbf{Z}^*) \geq f(\mathbf{Z}^D)$. Because $\mathbf{Z}^*$ is the optimal, we have $f(\mathbf{Z}^*) = f(\mathbf{Z}^D)$ and $\mathbf{Z}^C = \mathbf{0}$, thus $\mathbf{Z}^* = \mathbf{Z}^D$. Hence the optimal solution to Problem (3) is block diagonal and Theorem 2 holds. □

Though Theorem 2 guarantees the block diagonal structure of the solution to Problem (3), its orthogonal subspace assumption is often too strong to be satisfied in practice. On the other hand, without the orthogonal subspace assumption, we may still claim that: when the weighting parameter λ in the objective function is large enough, the optimal solution to Problem (3) may also possibly be block diagonal. From (6) and (8), the objective function $f(\mathbf{Z}^*) \geq f(\mathbf{Z}^D) + \lambda\|\mathbf{Z}^{C^T}\mathbf{Z}^C\|_1 + \|\mathbf{X}\mathbf{Z}^C\|_F^2 + 2\sum_{i=1}^N (\cos\theta_i \cdot \|\mathbf{X}\mathbf{z}_i^D - \mathbf{x}_i\|_2 \cdot \|\mathbf{X}\mathbf{z}_i^C\|_2)$. If λ is large enough such that $\lambda\|\mathbf{Z}^{C^T}\mathbf{Z}^C\|_1 + \|\mathbf{X}\mathbf{Z}^C\|_F^2 - 2\sum_{i=1}^N (|\cos\theta_i| \cdot \|\mathbf{X}\mathbf{z}_i^D - \mathbf{x}_i\|_2 \cdot \|\mathbf{X}\mathbf{z}_i^C\|_2) = 0$ (or ≥ 0) holds, then we could come to the conclusion that $\mathbf{Z}^*$ must be block diagonal.

It is known that SSC and LRR work under the assumption that the linear subspaces are independent. Our SSQP method makes stronger assumption that the subspaces are orthogonal or λ in Problem (3) is large enough. However, the experiment results on toy data show that SSQP also performs well when subspaces are simply independent and λ is properly set. Furthermore, all these three methods show reasonably good performances in practical problems where independent subspace assumption is possibly violated.

Algorithm 1 Spectral Projected Gradient Method

Input: data matrix $\mathbf{X}$, parameter λ,
Output: $\mathbf{Z}$.
$\mathbf{Z} \leftarrow \mathbf{Z}_0$ (initial guess of $\mathbf{Z}$),
$\sigma \leftarrow 1$,
repeat
  $\mathbf{D} \leftarrow \mathcal{P}_{\mathcal{C}}(\mathbf{Z} - \sigma \nabla f(\mathbf{Z})) - \mathbf{Z}$,
  Compute the step length ρ using line search,
  $\mathbf{Z}_{new} \leftarrow \mathbf{Z} + \rho \mathbf{D}$,
  $\mathbf{s} \leftarrow \mathsf{vec}(\mathbf{Z}_{new} - \mathbf{Z})$,
  $\mathbf{y} \leftarrow \mathsf{vec}(\nabla f(\mathbf{Z}_{new}) - \nabla f(\mathbf{Z}))$,
  $\sigma \leftarrow \mathbf{y}^T\mathbf{y}/\mathbf{s}^T\mathbf{y}$,
until converged

Optimization

SSQP need to solve a convex quadratic programming problem with box constraints, which can be effectively solved by many existing solvers. The gradient projection methods are simple but efficient, and they are most efficient when the constraints are simple in form especially when there are only bounds on variables – that coincides with the formulation of Problem (3).

We choose to use Spectral Project Gradient method (SPG) (Birgin, Martínez, and Raydan 1999), as shown in Algorithm 1, for its simplicity and efficiency. To apply SPG, we need to calculate the gradient of objective function and the projection operator.

Gradient of objective function. Let $\mathbf{E}$ be an all-one matrix. Because $\mathbf{Z}$ is element-wise nonnegative, the objective function can be re-expressed as the following smooth function

$$\begin{aligned} f(\mathbf{Z}) &= \|\mathbf{XZ} - \mathbf{X}\|_F^2 + \lambda \|\mathbf{Z}^T\mathbf{Z}\|_1 \\ &= \mathrm{tr}[(\mathbf{XZ} - \mathbf{X})^T(\mathbf{XZ} - \mathbf{X})] + \lambda \mathbf{e}^T\mathbf{Z}^T\mathbf{Ze} \\ &= \mathrm{tr}(\mathbf{Z}^T\mathbf{X}^T\mathbf{XZ}) - 2\mathrm{tr}(\mathbf{X}^T\mathbf{XZ}) \\ &\quad + \mathrm{tr}(\mathbf{X}^T\mathbf{X}) + \lambda \mathrm{tr}(\mathbf{Z}^T\mathbf{ZE}). \end{aligned} \tag{9}$$

The gradient of the objective function is then given by

$$\nabla_{\mathbf{Z}} f(\mathbf{Z}) = 2\mathbf{X}^T\mathbf{XZ} - 2\mathbf{X}^T\mathbf{X} + 2\lambda \mathbf{ZE}. \tag{10}$$

Projection onto the feasible set. Let $\mathcal{C} = \{\mathbf{Z} \in \mathbb{R}^{N\times N} \mid z_{ii} = 0 \text{ and } z_{ij} \geq 0 \ \forall\, i, j\}$ be the feasible set. We define the projection operator $\mathcal{P}_{\mathcal{C}} : \mathbb{R}^{N\times N} \mapsto \mathcal{C}$ as:

$$[\mathcal{P}_{\mathcal{C}}(\mathbf{Z})]_{ij} = \begin{cases} z_{ij} & \text{if } i \neq j \text{ and } z_{ij} \geq 0; \\ 0 & \text{if } i = j \text{ or } z_{ij} < 0\,. \end{cases} \tag{11}$$

It's easy to prove that $\mathcal{P}_{\mathcal{C}}(\mathbf{Z}) = \mathrm{argmin}_{\mathbf{Y}\in\mathcal{C}} \|\mathbf{Z} - \mathbf{Y}\|_F$.

Implementation details. Several criteria can be used to examine the convergence, e.g. $\|\mathbf{Z}_{new} - \mathbf{Z}\|_F^2 < \tau_1$ or $\sigma < \tau_2$, where τ_1 and τ_2 are pre-defined convergence tolerance. Empirically speaking, the convergence tolerance τ and the weighting parameter λ should be set small if the data are "clean" and they should be relatively large if the data are contaminated with heavy data noises.

Algorithm 2 Subspace Segmentation via Quadratic Programming (SSQP)

Input: data matrix $\mathbf{X}$.
Output: segmentation result.

1. Call Algorithm 1 to solve Problem (3), return $\mathbf{Z}$,
2. Adjacency matrix: $\mathbf{W} \leftarrow (\mathbf{Z} + \mathbf{Z}^T)/2$,
3. Compute graph Laplacian matrix $\mathbf{L}$ using $\mathbf{W}$,
4. Segmentation result $\leftarrow SpectralClustering(\mathbf{L})$.

Solving the subspace segmentation problem

Theorem 2 guarantees that under the orthogonal subspace assumption or λ is large enough, it is enforced to have that $z_{ij}^* = 0$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are from two different subspaces. With the affinity matrix $\mathbf{Z}^*$ learned, the ultimate segmentation results can be obtained by applying spectral clustering on $\mathbf{Z}^*$. The entire procedure of SSQP is depicted in Algorithm 2.

Experiments

In this section, we carry out three sets of experiments to evaluate the performance of SSQP in comparison with SSC and LRR. The source codes of SSC[1] and LRR[2] are released by (Elhamifar and Vidal 2009) and (Liu, Lin, and Yu 2010) respectively.

We first generate some toy data to evaluate the algorithmic robustness to data noises. Then we apply SSQP for motion segmentation and face clustering under varying illuminations. All experiments were performed on a PC with a quad-core 2.83HZ CPU and 8GB RAM.

Toy data

We construct three independent but not orthogonal linear subspaces $\{\mathcal{S}_k\}_{k=1}^3 \subseteq \mathbb{R}^{200}$, and randomly sample 200 data vectors from each subspace. The bases of each subspace are computed by $\mathbf{U}^{(k+1)} = \mathbf{TU}^{(k)}, 1 \leq k \leq 2$, where $\mathbf{T} \in \mathbb{R}^{200\times 200}$ is a random rotational matrix and $\mathbf{U}^{(k)} \in \mathbb{R}^{200\times 5}$ is random matrix with orthogonal columns. The data matrix $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3]$ is randomly sampled from the three subspaces by $\mathbf{X}_k = \mathbf{U}^{(k)}\mathbf{Q}^{(k)}$, where $\mathbf{Q}^{(k)} \in \mathbb{R}^{5\times 200}$ is a random matrix with entries uniformly distributed $q_{ij}^{(k)} \sim \mathcal{U}(0, 1)$. We add Gaussian noise of $\mathcal{N}(0, 0.3)$ to a fraction of the entries of matrix $\mathbf{X}$.

Then we use the data matrices to compare the performances of SSQP with LRR and SSC. We run each algorithm 10 times and record the average accuracies. All parameters are set best. The segmentation accuracies are shown in Figure 1 and the time elapsed is shown in Table 1.

The results show that all the three methods are robust to data noises. On segmentation accuracy, as can be seen from Figure 1 that SSQP and LRR are comparable, both better than SSC. On computational time, SSQP is considerably faster than LRR and SSC, as shown in Table 1. And this experiment empirically shows that SSQP converges faster in presence of data noises.

[1] `http://www.vision.jhu.edu/code`
[2] `http://apex.sjtu.edu.cn/apex_wiki/gcliu`

Table 1: Comparison of average elapsed time (seconds) on toy data experiment. The data matrices are noise free, 50% entries with noises and 100% with noises, respectively.

Method	SSC	LRR	SSQP
noise-free	884.1	426.3	36.2
50% noises	1114.6	362.8	4.1
100% noises	1169.6	359.0	3.4

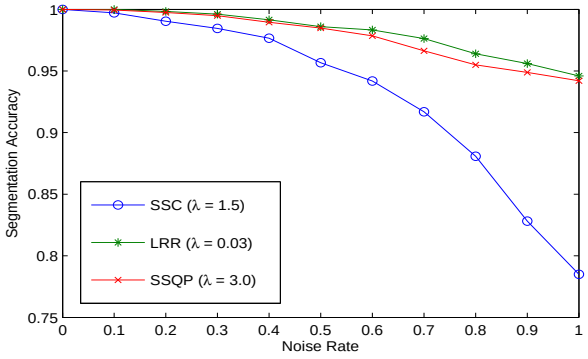


Figure 1: Error rate (%) in presence of noise on the toy data experiment.

Motion segmentation

We also apply SSQP to motion segmentation problem and evaluate its performance on the Hopkins 155 database[3] (Tron and Vidal 2007). We only compare our result with SSC and LRR, which are reported as the state-of-the-art by (Vidal 2010; Liu, Lin, and Yu 2010). For those results by GPCA, RANSAC, ALC and others, please refer to (Vidal 2010).

For the SSQP method, we project the data onto 8-dimension and 12-dimension subspaces by the principal component analysis (PCA), then perform Algorithms 1 to learn the coefficient matrix $\mathbf{Z}^*$ (the parameter is fixed for all sequences). Then we use $(\mathbf{Z}^* + (\mathbf{Z}^*)^T)/2$ as the affinity matrix and perform spectral clustering. We compare the clustering results with SSC (preprocessed by random projection) and LRR (without projection, which lead to better results because LRR is an implicit dimension reduction procedure). The results are listed in Table 2.

The Hopkins155 data are nearly noise-free, therefore SSC, LRR, and SSQP all achieve quite good performance. Note that here we apply different data preprocessing schemes to make sure that each individual method works best on this data set.

We also compare the elapsed time of different methods. For all of the three methods, data vectors are projected onto 12-dimension subspaces by PCA. SSQP takes 4627 seconds to go through all the 155 sequences, while the computational time is 13661 seconds for SSC and 14613 seconds for LRR.

[3] http://www.vision.jhu.edu/data/hopkins155

Table 2: Motion segmentation error rate (%). SSC uses Bernoulli and normal random projection, denoted as SSC-B and SSC-N, respectively. LRR does not use data projection. SSQP uses PCA to project the data onto 8-dimension and 12-dimension subspaces, denoted as PCA-8 and PCA-12.

Method	SSC		LRR	SSQP	
	SSC-B	SSC-N	No Proj	PCA-8	PCA-12
Checkboard with 2 motions: 78 sequences					
Mean	0.83	1.12	1.45	0.95	0.88
Traffic with 2 motions: 31 sequences					
Mean	0.23	0.02	1.57	0.14	1.66
Articulated with 2 motions: 11 sequences					
Mean	1.63	0.62	1.44	2.64	7.68
Checkboard with 3 motions: 26 sequences					
Mean	4.49	2.97	3.35	3.73	2.19
Traffic with 3 motions: 7 sequences					
Mean	0.61	0.58	8.26	0.18	0.58
Articulated with 3 motions: 2 sequences					
Mean	1.60	1.42	4.26	1.60	1.60
All 155 sequences					
Mean	1.45	1.24	2.13	1.35	1.50
Median	0	0	0	0	0

Since the data is very "clean", as aforementioned we set SSQP's $\lambda = 10^{-5}$, and thus the convergence tolerance is correspondingly very small. In this case, SSQP converges slower, but is still much faster than SSC and LRR.

Face clustering under varying illuminations

Given p subjects, each of which is photoed under q poses and m illumination, we need to segment the $p \times q \times m = N$ pictures into $p \times q = n$ clusters. Previous work shows that the set of images of the same face in fixed pose under varying lighting can be effectively approximated by low-rank linear subspaces (Lee, Ho, and Kriegman 2005). Thus we can use subspace segmentation to solve this problem.

We use the Extended Yale Face Database B (Yale B) (Georghiades, Belhumeur, and Kriegman 2001) to evaluate the SSC, LRR, and SSQP methods on face clustering under varying illuminations. For convenience, we use the cropped images (Lee, Ho, and Kriegman 2005). The Yale B contains 16128 images of 28 human subjects under 9 poses and 64 illumination conditions each, thus the task is to segment the 16128 images into $28 \times 9 = 252$ groups.

In this experiment, we use only 5 subsets of the 252 groups for convenience. Rather than choosing "good" data set elaborately to facilitate SSQP's high performance, we simply use $[1, 2]$, $[1, 2, 3]$, $\cdots$, $[1, 2, 3, 4, 5, 6]$ for fair of comparison, where each number from 1 to 6 is the index of a group (among the 252 groups), each group contains 64 images. For computation efficiency, we resize the images to smaller 32×32 images and then use PCA projection. Following the previous work (Lee, Ho, and Kriegman 2005), we assume all 64 images in one group lie in a rank r subspace. From the linear independent assumption, we assume different groups lie in independent subspaces. Therefore, we

Table 3: Face clustering error rates (%) and elapsed time (seconds) on the Extended Yale Face Database B. (Note that n denotes the number of groups in each subset).

	Error Rate			Time		
n	SSC	LRR	SSQP	SSC	LRR	SSQP
2	1.56	3.12	0.78	29.0	4.6	0.1
3	1.56	10.41	2.08	57.5	9.0	0.3
4	2.73	7.81	1.95	87.0	21.3	0.6
5	2.81	10.31	2.19	116.4	36.8	0.9
6	2.86	10.16	2.86	148.7	63.9	3.1

project the subset $[1, 2]$ onto $2r$-dimension subspace, $[1, 2, 3]$ onto $3r$, and so on. By searching within $\{1, 2, \cdots, 10\}$, we choose to use $r = 6$ which is good for all of the three methods. Table 3 lists the error rates and running time of SSC, LRR, and SSQP.

In this experiment, SSQP achieves the best performance among the three methods, which could possibly be explained by the nonnegativity of the coefficient matrix enforced by SSQP. Rather than affine subspaces, all face vectors lies in a polyhedral cone in the nonnegative orthant, thus the coefficients of the linear combination should be nonnegative. SSQP addresses the face clustering problem by expressing each face as the nonnegative linear combination of other faces. If two faces are segmented into the same cluster, they are positively correlated. Therefore SSQP is more interpretable and effective in this dataset.

Conclusions

We propose in this paper the SSQP method to efficiently and accurately solve the subspace segmentation problem. SSQP is formulated as a convex quadratic program which can be optimized with standard QP solvers, e.g., SPG used in our implementation. Its time efficiency and robustness to noises have been verified by extensive experiments on toy and real-world data sets, with comparison to the state-of-the-art subspace segmentation methods SSC and LRR. In the setting where the data vectors are drawn from a cone in the nonnegative orthant, such as the face clustering under varying illuminations problem, SSQP is particularly more effective then both SSC and LRR.

Acknowledgments

We would like to acknowledge the support of "NExT Research Center" funded by MDA, Singapore, under the research grant, WBS:R-252-300-001-490.

References

Agarwal, P. K., and Mustafa, N. H. 2004. k-means projective clustering. In *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 155–165.

Birgin, E. G.; Martínez, J. M.; and Raydan, M. 1999. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization* 10:1196–1211.

Boult, T., and Brown, L. 1991. Factorization-based segmentation of motions. In *Proceedings of the IEEE Workshop on Visual Motion*, 179 –186.

Costeira, J., and Kanade, T. 1998. A multibody factorization method for independently moving objects. *International Journal of Computer Vision* 19(3):159–179.

Elhamifar, E., and Vidal, R. 2009. Sparse subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Fischler, M. A., and Bolles, R. C. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of ACM* 24:381–395.

Georghiades, A.; Belhumeur, P.; and Kriegman, D. 2001. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Conference on Computer Vision and Pattern Recognition* 23(6):643–660.

Ho, J.; Yang, M.-H.; Lim, J.; Lee, K.-C.; and Kriegman, D. 2003. Clustering appearances of objects under varying illumination conditions. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Hong, W.; Wright, J.; Huang, K.; and Ma, Y. 2006. Multiscale hybrid linear models for lossy image representation. *IEEE Transactions on Image Processing* 15(12):3655–3671.

Lee, K.; Ho, J.; and Kriegman, D. 2005. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(5):684–698.

Lin, Z.; Chen, M.; Wu, L.; and Ma, Y. 2009. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *UIUC Technical Report, UILU-ENG-09-2215*.

Liu, G.; Lin, Z.; and Yu, Y. 2010. Robust subspace segmentation by low-rank representation. In *Proceedings of the 27th International Conference on Machine Learning*.

Ma, Y.; Derksen, H.; Hong, W.; and Wright, J. 2007. Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(9):1546–1562.

Rao, S.; Tron, R.; Ma, Y.; and Vidal, R. 2008. Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Tron, R., and Vidal, R. 2007. A benchmark for the comparison of 3-d motion segmentation algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Vidal, R., and Hartley, R. 2004. Motion segmentation with missing data using power factorization and gpca. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Vidal, R.; Ma, Y.; and Sastry, S. 2005. Generalized principal component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(12):1–15.

Vidal, R. 2010. A tutorial on subspace clustering. *To appear in the IEEE Signal Processing Magazine*.