Robust Subspace Segmentation with Block-diagonal Prior

Jiashi Feng[1], Zhouchen Lin[2]*, Huan Xu[3], Shuicheng Yan[1]
[1]Department of ECE, National University of Singapore, Singapore
[2]Key Lab. of Machine Perception, School of EECS, Peking University, China
[3]Department of ME, National University of Singapore, Singapore
[1]{a0066331,eleyans}@nus.edu.sg, [2]zlin@pku.edu.cn, [3]mpexuh@nus.edu.sg

Abstract

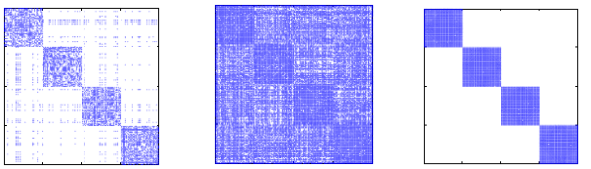
The subspace segmentation problem is addressed in this paper by effectively constructing an exactly block-diagonal *sample affinity matrix. The block-diagonal structure is heavily desired for accurate sample clustering but is rather difficult to obtain. Most current state-of-the-art subspace segmentation methods (such as SSC [4] and LRR [12]) resort to alternative structural priors (such as sparseness and low-rankness) to construct the affinity matrix. In this work, we directly pursue the block-diagonal structure by proposing a graph Laplacian constraint based formulation, and then develop an efficient stochastic subgradient algorithm for optimization. Moreover, two new subspace segmentation methods, the block-diagonal SSC and LRR, are devised in this work. To the best of our knowledge, this is the first research attempt to explicitly pursue such a block-diagonal structure. Extensive experiments on face clustering, motion segmentation and graph construction for semi-supervised learning clearly demonstrate the superiority of our novelly proposed subspace segmentation methods.*

1. Introduction

High-dimensional vision data, such as face images and rigid object motion trajectories, are generally distributed in a union of multiple low-dimensional subspaces [8, 21, 25]. To find such a low-dimensional structure, we usually need to cluster the data into multiple groups and meanwhile fit each group by a subspace. This introduces the important *subspace segmentation* problem defined as follows.

Definition 1 (Subspace Segmentation [22])**.** *Given a set of sample vectors* $X = [X_1, \ldots, X_k] = [x_1, \ldots, x_n] \in \mathbb{R}^{d\times n}$ *drawn from a union of* k *subspaces* $\{\mathcal{S}_i\}_{i=1}^k$*. Let* X_i *be a collection of* n_i *samples drawn from the subspace* $\mathcal{S}_i$*,* $n = \sum_{i=1}^k n_i$*. The task of subspace segmentation is to segment the samples according to the underlying subspaces they are drawn from.*

*Corresponding author.



(a) Sparse (b) Low-rank (c) Block-diagonal

Figure 1. An illustrative comparison on the constructed sample affinity matrices for synthetic noisy samples from 4 subspaces, using different priors: (a) sparse, (b) low-rank, and (c) our proposed block-diagonal. The block-diagonal affinity matrix characterizes the sample clusters and subspace segmentation more accurately.

Recently, many spectral clustering based solutions to the subspace segmentation problem have been proposed [4, 12, 13]. These methods use local or global information around each sample to build a sample affinity matrix. The segmentation of the samples is then obtained by applying spectral clustering [15] on the affinity matrix.

In particular, Sparse Subspace Clustering (SSC) [4] and Low-Rank Representation (LRR) [12], as two examples of the state-of-the-art subspace segmentation methods, construct the affinity matrix through finding a sparse or low-rank linear representation of each sample with respect to the whole sample collection. The obtained representation coefficients are then used directly to build the affinity matrix. These methods are able to generate a block-diagonal affinity matrix under restrictive conditions. However, the block-diagonal structure obtained by those methods is fragile and will be destroyed when the signal noise ratio is small, the different subspaces are too close, or the subspaces are not independent. Hence the subspace segmentation performance may be degraded severely [19].

In this work, we propose to explicitly pursue such a block-diagonal structured affinity matrix for subspace segmentation. We impose an explicit fixed rank constraint on the graph Laplacian, which can equivalently constrain the

number of connected components in the constructed affinity matrix. Thus a block-diagonal affinity matrix can be effectively obtained, even under the adversarial scenarios, such as small signal noise ratio, improper localization of different subspaces, *etc*. An illustrative example is given in Figure 1. It demonstrates that compared with sparse and low-rank prior, our proposed method yields more accurate affinity matrices. To solve the induced optimization problem efficiently, we propose a stochastic sub-gradient descent method along with a projection operation to guarantee the affinity matrix is block-diagonal in the iteration.

We take the state-of-the-art subspace segmentation techniques, SSC and LRR, as illustrating examples for our proposed method. We specifically show how the proposed method is able to generate an exactly block-diagonal affinity matrix for them and improve their performances on subspace segmentation tasks. It is worth noting that our proposed method is by no means restricted to these two cases. In fact, the proposed method is quite general and can be applied for other affinity matrix construction methods straightforwardly. To verify the effectiveness of the proposed method, extensive subspace segmentation experiments are conducted, including synthetic data clustering, face images clustering and motion trajectories segmentation. We also perform semi-supervised learning experiments for digit and face recognition to further demonstrate the superiority of the proposed method, compared with other popular graph construction methods.

2. Related Work

During the past decades, a number of subspace segmentation or grouping methods have been developed [17, 4, 12, 23, 13, 5]. Since in this work we focus on the spectral clustering based methods, in the following we will review the related work along this direction in details. For other methods, a good review can be found in [22].

Sparse Subspace Clustering (SSC) [4] expresses each sample as a linear combination of all other samples in the collection, where the combination coefficients are required to be sparse. Afterwards, Liu *et al.* [12] propose the Low Rank Representation (LRR) for subspace segmentation. LRR enforces the constructed affinity matrix to be low-rank, which captures the global prior that the union of the underlying subspaces is still low-dimensional. This prior endows LRR with strong robustness to gross corruptions in the samples. Another block-diagonal inducing prior is introduced by Wang *et al.* in [23]. However they prove that only under the condition that the subspaces are orthogonal to each other, the sample affinity matrix is exactly block-diagonal. This condition is rather restrictive and does not apply to realistic data. Recently, Lu *et al.* [13] propose a least square regression based method for the affinity matrix construction. It is claimed that grouping effect brought by least square regression for the samples from the same subspace is able to improve the performance of subspace segmentation.

Though the existing methods have achieved great success for the subspace segmentation tasks, none of them is able to produce an exactly block-diagonal matrix for realistic samples. Thus their performances are sensitive to the noise, corruption and improper localization of the underlying subspaces.

3. Problem Formulation

In this work, we focus on applying spectral clustering on an affinity matrix, which is constructed based on global information, for subspace segmentation. For affinity matrix construction, each sample $\mathbf{x}_i \in \mathbb{R}^d$ is approximated by a linear combination of the reference samples $X\mathbf{z}_i$. Here $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ is the collection of the reference samples. The absolute value of the element $|\mathbf{z}_i(j)|$ in the coefficient vector $\mathbf{z}_i$ is then directly used as the affinity value between $\mathbf{x}_i$ and the j-th reference sample $\mathbf{x}_j$ in X. Denote $S(\mathbf{z}_i)$ as the index set of related samples for $\mathbf{z}_i$: $S(\mathbf{z}_i) = \{j|\mathbf{z}_i(j) \neq 0, \forall j = 1, \ldots, n\}$. To accurately segment the samples into the corresponding subspaces, it is required that $S(\mathbf{z}_i) \cap S(\mathbf{z}_k) = \emptyset$ if the i-th and k-th samples are from different subspaces. Namely, for any two samples from different subspaces, their affinity values should be zero. Thus, after proper row/column permutation of the symmetric affinity matrix $W = (|Z| + |Z^\top|)/2$, where $Z = [\mathbf{z}_1, \ldots, \mathbf{z}_n] \in \mathbb{R}^{n\times n}$ and $|Z|$ takes absolute values of the elements in Z, we obtain the following block-diagonal matrix:

$$\widetilde{W} = \text{blockdiag}\left(W_1, W_2, \ldots, W_k\right).$$

Each block W_i in the above block-diagonal affinity matrix $\widetilde{W}$ corresponds to a single subspace $\mathcal{S}_i$. The sample segmentation can be obtained immediately. Though several existing subspace segmentation methods are able to obtain such block-diagonal affinity matrices, the required conditions are too restrictive for realistic data, as discussed previously. For example, when the signal noise ratio is small, the samples will be perturbed by the noise and deviate from their corresponding subspaces. A certain sample may be closer to one sample from a different subspace and the affinity value between these two samples is non-zero. Such an incorrect affinity may affect the following spectral clustering process and thus degrade the performance of subspace segmentation. In contrast, if a block-diagonal prior is imposed, such an incorrect affinity may be corrected by the requirement of building an exactly block-diagonal matrix. The performance of the spectral clustering for subspace segmentation will be improved.

3.1. Laplacian Constraint for block-diagonality

Before introducing the proposed Laplacian constraint for block-diagonal matrix pursuit, we first recall the definition of the Laplacian matrix and the relationship between the spectral property of the Laplacian matrix and the structure of the affinity matrix.

Definition 2 (Laplacian Matirx)**.** *Consider an affinity matrix $W \in \mathbb{R}^{n\times n}$ of n samples with weights $W(j,j')$. The Laplacian matrix $L_W \in \mathbb{R}^{n\times n}$ is defined as: $L_W(j,j') = -W(j,j')$, if $j \neq j'$; $\sum_{\ell\neq j} W(j,\ell)$ otherwise.*

The following well known theorem [14] relates the rank of the Laplacian matrix to the number of blocks in the corresponding affinity matrix.

Theorem 1 ([14])**.** *Let W be an affinity matrix. Then the multiplicity k of the eigenvalue 0 of the corresponding Laplacian L_W equals the number of connected components (blocks) in W.*

Based on the above theorem, we can enforce a general square matrix to be k-block-diagonal by imposing the following Laplacian Constraint (LC): $\mathrm{rank}(L_W) = n - k$. Then we can define a set of k-block-diagonal matrix (k-BDMS) as:

$$\mathcal{K} = \left\{ Z | \mathrm{rank}(L_W) = n-k, W = \frac{1}{2}\left(|Z| + |Z^\top|\right) \right\}.$$

In the above constraint set, the parameter k is the required number of clusters in the subspace segmentation problems (see Definition 1). After building k block diagonal affinity matrix, the samples are readily segmented into k clusters. The value of k is usually specified by users.

3.2. LC for block-diagonal SSC

Sparse Subspace Clustering (SSC) solves the subspace segmentation problem through pursuing a sparse representation coefficient matrix [4]. Its objective function is:

$$\min_Z \|Z\|_1 + \frac{\lambda}{2}\|X - XZ\|_F^2, \text{ s.t. } \mathrm{diag}(Z) = 0.$$

After obtaining the coefficient matrix Z, a similarity matrix can be constructed as $W = (|Z| + |Z^\top|)/2$. Under certain conditions, *e.g.*, noiseless data, SSC only selects the samples from the same subspace as the target sample in its linear representation. And thus the resultant similarity matrix is block-diagonal. However, for the realistic data, it is difficult to satisfy the conditions. In order to obtain a block-diagonal affinity matrix, we impose the LC for SSC and propose the following block-diagonal SSC (BD-SSC) objective function:

$$\begin{aligned} &\min_Z f_1(Z) = \|Z\|_1 + \frac{\lambda}{2}\|X - XZ\|_F^2, \\ &\text{s.t. } \mathrm{diag}(Z) = 0, Z \in \mathcal{K}. \end{aligned} \tag{1}$$

As explained above, $Z \in \mathcal{K}$ enforces the obtained matrix Z to form a block-diagonal affinity matrix.

3.3. LC for block-diagonal LRR

Low-Rank Representation (LRR) seeks a low-rank representation coefficient matrix of a given set of samples, w.r.t. the basis composed by the samples themselves [12]. The rationale of the low-rank regularization is utilizing the global prior information that the union of the underlying low-dimensional subspaces is still low-dimensional. Formally, LRR solves the following optimization problem[1]:

$$\min_Z \|Z\|_* + \frac{\lambda}{2}\|X - XZ\|_F^2.$$

Under the conditions that the samples are sufficient, noiseless and the subspaces are independent, the obtained similarity matrix will be exactly block-diagonal. However, for realistic data, such as face images or images of other objects, such requirements are too restrictive. The obtained matrix W is usually not block-diagonal, which deteriorates the performance of subspace segmentation. To enforce a k-block-diagonal structure on the obtained similarity matrix W, we also introduce the aforementioned LC constraint to LRR and propose the following block-diagonal LRR (BD-LRR):

$$\min_{Z,E} f_*(Z) = \|Z\|_* + \frac{\lambda}{2}\|X - XZ\|_F^2, \text{ s.t. } Z \in \mathcal{K}. \tag{2}$$

The optimization problems for SSC and LRR, as shown in Eqn. (1) and Eqn. (2) respectively, are different and generally different optimization methods are proposed to solve them individually. In this work, we aim to develop a general optimization method to solve these two problems in a unified framework. The details of the optimization are provided in the following section.

4. Optimization with Laplacian Constraint

We employ the efficient stochastic sub-gradient descent (SSGD) method to solve the optimization problems in Eqn. (1) and Eqn. (2), which both involve a highly non-convex k-BDMS constraint. In each iteration, the affinity matrix Z moves along the negative sub-gradient direction to decrease the construction residue. Then Z is instantly projected back onto the constraint set $\mathcal{K}$ via $\Pi_{\mathcal{K}}(\cdot)$, to ensure it preserves the block-diagonal structure.

The sub-gradients are calculated as follows. For the objective function $f_1(Z)$ in Eqn. (1), the sub-gradient is $\mathcal{G}(f_1(Z)) = \lambda X^\top X(Z - I) + \partial\|Z\|_1/\partial Z$. Here

[1] In the original LRR, $\ell_{2,1}$-norm of the residual is minimized. However, for most cases, we find minimizing a Frobenius norm of the residual also works well in practice, which is much more efficient than minimizing the $\ell_{2,1}$-norm.

$\partial\|Z\|_1/\partial Z_{ij} = \text{sign}(Z_{ij})$ if $Z_{ij} \neq 0$ and $\theta \in [-1, 1]$ otherwise. Since the diagonal elements of the affinity matrix Z are not necessarily to be updated, we manually fix the diagonal elements of $\mathcal{G}(f_1(Z))$ to be zeros. For the objective function $f_*(Z)$ in Eqn. (2), the sub-gradient is $\mathcal{G}(f_*(Z)) = \lambda X^\top X(Z - I) + \partial\|Z\|_*/\partial Z$. Here let $U\Sigma V^\top$ be a compact SVD of Z, and then $\partial\|Z\|_*/\partial Z = UV^\top + H$[2] [24], where H is a set as zero matrix in this work. The details of the SSGD algorithm are given in Alg. 1, and the details of computing the projection $\Pi_{\mathcal{K}}(\cdot)$ are given in Alg. 2.

Algorithm 1: Basic SSGD

input : Data matrix X, objective function f, number of blocks k, maximal iteration number T, trade-off parameter λ.

Initialize $Z^{(0)} = 0, t = 0, p = \text{rank}(X)$.
Stepsize $\eta = 1.5\sqrt{np}/(1.5\lambda n\|XX^\top\|_2 + \sqrt{p})\sqrt{T}$
while $t \leq T$ **do**
  Generate an $n \times p$ probing matrix Y by randomly selecting p column vectors from $\{\sqrt{n}\mathbf{e}_1, \ldots, \sqrt{n}\mathbf{e}_n\}$;
  Calculate sub-gradient: $g^{(t)} \leftarrow \mathcal{G}(f(Z^{(t)}))YY^\top$;
  Projection: $Z^{(t+1)} \leftarrow \Pi_{\mathcal{K}}(Z^{(t)} - \eta g^{(t)})$ via Alg. 2;
  $t \leftarrow t + 1$;
end

output: Coefficient matrix Z.

4.1. Unbiased Sub-gradient Estimation

One of the most important ingredients for SSGD to be convergent and effective is an unbiased estimator for a sub-gradient of the objective function. $\tilde{g}^{(t)}$ is called as an unbiased estimator of the sub-gradient $g^{(t)}$ if $\mathbb{E}(\tilde{g}^{(t)}) = g^{(t)}$. In this work, we adopt the following randomized sparsification method to obtain an unbiased estimator of the sub-gradient and meanwhile reduce the cost of computing the sub-gradient [7]. In particular, we use a probing matrix to sparsify the sub-gradient estimation. A probing matrix is defined as follows.

Definition 3 (Probing Matrix [7])**.** *A random $n \times p$ matrix Y with $p < n$ is a probing matrix if $\mathbb{E}[YY^\top] = I_{n\times n}$ where $I_{n\times n}$ is the $n \times n$ identity matrix and the expectation is over the choice of Y.*

In this work, we randomly sample p vectors from the scaled standard basis $\{\sqrt{n}\mathbf{e}_1, \ldots, \sqrt{n}\mathbf{e}_n\}$ to form the probing matrix: $Y = [\sqrt{n}\mathbf{e}_{(1)}, \ldots, \sqrt{n}\mathbf{e}_{(p)}]/\sqrt{p}$. It is easy to verify that $\mathbb{E}[YY^\top] = I$. Then the sub-gradient in each descent step is calculated as $g^{(t)} = \mathcal{G}(f(Z^{(t)}))YY^\top$, where $\mathcal{G}(\cdot)$ is the sub-gradient defined as above.

[2] H should satisfy: $U^\top H = 0, HV = 0, \|H\|_2 \leq 1$.

To see how this technique reduces the computation cost, we take BD-LRR as an example (the explanation also applies for BD-SSC). In Step 5 of Alg. 1, the gradient is $\mathcal{G}(f(Z))YY^\top$, where computing $\mathcal{G}(f(Z))$ contains calculating $XX^\top Z$ plus an SVD on Z. For the $XX^\top Z$ part, after multiplying with $YY^\top$, it becomes $XX^\top(ZY)Y^\top$. ZY actually samples p out of n columns of Z. The size of Z is reduced from n^2 to np. The complexity of computations involving Z is thus reduced. The multiplication with $Y^\top$ is padding the left matrix with zeros and quite fast. As for SVD of Z, we use the implementation in Lemma 2.4 of [7]. The computational cost is reduced from $O(n^2r)$ to $O(npr)$. Here $r = \text{rank}(Z)$. This technique reduces the complexity from $O(n^3)$ to $O(pn^2)$ in the sub-gradient calculation. If $p \ll n$, the efficiency enhancement is significant.

4.2. Solving Projection to k-BDMS

After a step of sub-gradient descent, the variable matrix Z may move out of the constraint set and no longer possesses a k-block-diagonal structure. Thus, we need to project the matrix Z back to the k-BDMS constraint set. The projection operator for a matrix Z_0 is defined as:

$$\Pi_{\mathcal{K}}(Z_0) = \arg\min_{Z\in\mathcal{K}} \|Z - Z_0\|_F^2. \tag{3}$$

The projection essentially finds a matrix in the set $\mathcal{K}$ which is closest to Z_0 in terms of the Euclidean distance. The involved optimization problem can be explicitly written as:

$$\min_Z \frac{1}{2}\|Z - Z_0\|_F^2, \text{ s.t. } Z \in \mathcal{K}.$$

The above optimization problem is severely complicated by the two imposed constraints (recall definition of $\mathcal{K}$), both of which contain complicated transformation on the variable Z. However, after inspecting the problem, we find that if the two constraints are decoupled thus we do not need to simultaneously deal with the two constraints, the problem will be significantly simplified. Therefore, we introduce an auxiliary variable $\widetilde{Z}$ to replace the Laplacian matrix L_W. Then the objective function can be written equivalently as:

$$\min_{Z,\widetilde{Z}} \frac{1}{2}\|Z - Z_0\|_F^2,$$
$$\text{s.t. } \text{rank}(\widetilde{Z}) = n - k, W = \frac{1}{2}\left(|Z| + |Z^\top|\right), \widetilde{Z} = L_W.$$

We further rewrite the constraint $\widetilde{Z} = L_W$ as a penalty term via Augmented Lagrangian Multiplier (ALM) [11]:

$$\min_{Z,\widetilde{Z}} \frac{1}{2}\|Z - Z_0\|_F^2 + \langle J, \widetilde{Z} - L_W\rangle + \frac{\beta}{2}\|\widetilde{Z} - L_W\|_F^2,$$
$$\text{s.t. } \text{rank}(\widetilde{Z}) = n - k, W = \frac{1}{2}(|Z| + |Z^\top|).$$

Here J is the Lagrangian multiplier and β is an increasing weight parameter for the term of enforcing $\widetilde{Z} = L_W$. Now the two constraints are decoupled and we can alternatively optimize Z and $\widetilde{Z}$. In particular, each optimization problem only involves one constraint.

Algorithm 2: Projection to k-BDMS

input : Target matrix Z_0, number of blocks k
output: A block-diagonal matrix Z.

Initialization: $Z^{(0)} = Z_0$; $t = 0$; $\rho = 1.1$; $\beta^{(0)} = 1 \times 10^{-4}$
while *Not converged* **do**
  Update $Z^{(t+1)}$ by solving the problem in (4);
  Update $\widetilde{Z}^{(t+1)}$ by solving the problem in (5);
  Update $Y^{(t+1)} = Y^{(t)} + \beta^{(t)}(\widetilde{Z} - L_W)$;
  Update $\beta^{(t+1)} = \rho\beta^{(t)}$;
  $t \leftarrow t + 1$.
end

Fixing the variable $\widetilde{Z}$ and multiplier J and removing the term not containing Z, we have the following problem to solve Z:

$$\min_{Z} \frac{1}{2}\|Z - Z_0\|_F^2 - \langle J, L_W\rangle + \frac{\beta}{2}\|\widetilde{Z} - L_W\|_F^2,$$
$$\text{s.t. } W = \frac{1}{2}(|Z| + |Z^\top|). \tag{4}$$

Note that except for the term $\|Z - Z_0\|_F^2$, all other terms in the above problem only contain the absolute value of Z. Thus the elements of the solution Z must have the same sign as the ones in Z_0. Otherwise, we can always change the sign of Z's elements, decreasing the value of $\|Z - Z_0\|_F^2$ while not changing the values of other terms. Therefore, based on this observation, the solution can be written as $Z = \hat{Z} \otimes \text{sign}(Z_0)$, where $\otimes$ denotes the element-wise multiplication and $\hat{Z}$ is the solution to the following problem,

$$\min_{\hat{Z}} \frac{1}{2}\|\hat{Z} - |Z_0|\|_F^2 - \langle J, L_W\rangle + \frac{\beta}{2}\|\widetilde{Z} - L_W\|_F^2,$$
$$\text{s.t. } W = \frac{1}{2}(\hat{Z} + \hat{Z}^\top), \hat{Z} \geq 0.$$

The above objective function can be efficiently solved by any off-the-shelf quadratic programming solver. After solving out $\hat{Z}$, we can recover the solution $Z = \hat{Z} \otimes \text{sign}(Z_0)$.

Now we turn to solving $\widetilde{Z}$. Similarly, fixing the variable Z and multiplier J, we update $\widetilde{Z}$ via solving:

$$\min_{\widetilde{Z}} \langle J, \widetilde{Z}\rangle + \frac{\beta}{2}\|\widetilde{Z} - L_W\|_F^2, \text{ s.t. } \text{rank}(\widetilde{Z}) = n - k. \tag{5}$$

It is equivalent to:

$$\min_{\widetilde{Z}} \|\widetilde{Z} - (L_W - 1/\beta J)\|_F^2, \text{ s.t. } \text{rank}(\widetilde{Z}) = n - k.$$

This problem admits a closed-form solution according to the Eckart-Young theorem [3]. More specifically, the closed-form solution can be obtained by performing SVD on $L_W - \frac{1}{\beta}J = U\Sigma V^\top$ and selecting the top $n - k$ singular vectors: $\widetilde{Z} = U_{1:(n-k)}[\Sigma]_{1:(n-k)}V_{1:(n-k)}^\top$. The details of optimizing Z and $\widetilde{Z}$ are presented in Alg. 2. In the implementation, when $\|\widetilde{Z}^{(t)} - L_{W^{(t)}}\| < 1 \times 10^{-6}$ or $\beta^{(t)} \max(\|\widetilde{Z}^{(t+1)} - \widetilde{Z}^{(t)}\|, \|Z^{(t+1)} - Z^{(t)}\|) < 1 \times 10^{-4}$, the optimization is stopped.

4.3. Notes on the Convergence

The optimization problems in Eqn. (1) and Eqn. (2) are heavily non-convex due to the k-BDMS constraint. Fortunately, we can prove the solution of Alg. 1 converges to the global optimum. Here we briefly explain it theoretically and the experimental validation is deferred to the next section. The convergence argument for Alg. 1 is built on the results in [1]. In particular, for affine problems with non-convex constraint, gradient descent with projection will converge to the global optimum if the Scalable Restricted Isometry Property (SRIP) holds. As for the problem (3) (which Alg. 2 solves), though it is not convex, we can also obtain the optimum. In practice, we only need $Z^{(t+1)}$ to be the optimum to (3) when $Z_0 = Z^{(t)} - \eta g^{(t)}$. To this end, we control η to be small. Then $Z^{(t)}$ must be close to projection of Z_0 in $\mathcal{K}$. We start the ALM iteration from $Z^{(t)}$, and ALM converges to its KKT point $Z^{(t+1)}$ [11], which must be thes optimum since $Z^{(t)}$ is close to Z_0.

5. Experiments

5.1. Synthetic Data

Data generation We generate 5 sets of synthetic samples under different noise levels. The samples are generated following the scheme in [12]. We construct $k = 4$ independent subspaces of 3 dimensional, whose ambient dimension is equal to 30, *i.e.*, $\{\mathcal{S}_i\}_{i=1}^4 \subset \mathbb{R}^{30}$. From each subspace $\mathcal{S}_i$, 50 samples are drawn. Then 30% of the samples are randomly chosen and corrupted by adding Gaussian noise with zero mean and variance $\sigma\|\mathbf{x}\|_2$. Here $\|\mathbf{x}\|_2$ denotes the ℓ_2-norm of the corresponding sample, and σ can be seen as the signal-noise ratio. We evaluate the performance of SSC, LRR, BD-SSC and BD-LRR under 5 different signal-noise ratios of $\sigma \in \{0, 0.1, 0.2, 0.3, 0.4\}$ respectively. Here $\sigma = 0$ means there is no noise corrupting the samples. The symmetric affinity matrix is obtained as $W = (|Z| + |Z^\top|)/2$, where matrix Z is the output of the above methods.

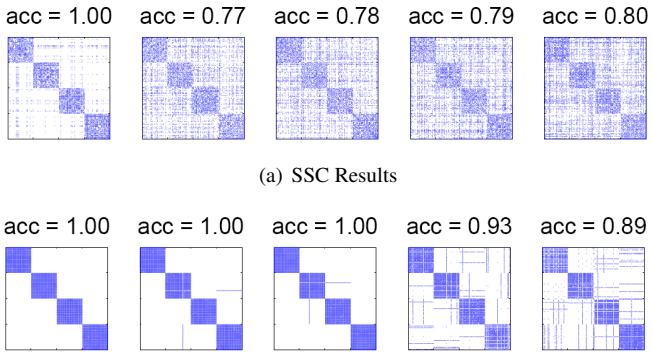
Implementation details For both BD-SSC and BD-LRR, the trade-off parameter λ is fixed as 10 and the maximal iteration number is set as $T = 600$. The variable Z in Alg. 1 is initialized as an all-zero matrix throughout the experiments. For SVD, we conduct partial SVD on $L_w - Y/\beta$

in (5) up to k smallest singular values $U_k\Sigma_k V_k$ and take $\widetilde{Z} = L_w - Y/\beta - U_k\Sigma_k V_k$ (amount to top $n-k$ singular values). Since usually $k \ll n-k$, this method reduces the cost of computing the partial SVD greatly. BD-SSC takes around 20 seconds to build an affinity matrix for $2,000$ samples on the MATLAB platform on a PC with Quad CPU of 2.83GHz and 8GB RAM, and BD-LRR costs more time, around 70 seconds, due to computing SVD. In comparison, SSC and LRR only costs 5.5 and 2.3 seconds respectively. Computational cost is a bottleneck of the proposed algorithm for large-scale applications. We will also investigate how to scale the algorithm to large scale applications in future. A promising speeding up technique is the divide-and-conquer distributed optimization, considering its success in accelerating LRR in [20]. We experimentally find the optimization converges though the problem is highly non-convex. In particular, we plot the objective value convergence curves for BD-SSC and BD-LRR in the supplementary material.

Results Figure 2 and Figure 3 show the obtained affinity matrices, whose rows and columns are permuted according to the ground-truth labels of the samples. The corresponding segmentation accuracies are also shown in the figures. We can observe that for both SSC and LRR, when the noise level $\sigma \geq 0.1$, some samples are selecting the samples from other subspaces in the representation. The block-diagonal structure of the affinity matrix is destroyed, and the segmentation accuracy decreases severely. For example, for the noiseless case ($\sigma = 0$), all of the methods can achieve perfect segmentation with an accuracy equal to 1. However, when the noise level σ increases to 0.2, the accuracy of SSC and LRR dramatically drops to 0.79 and 0.83 respectively. Due to the introduced block-diagonal constraint, BD-SSC and BD-LRR methods are able to produce the block-diagonal affinity matrix and the accuracy remains 1. The results well demonstrate that the proposed method has enhanced the robustness to noises and is able to achieve much better segmentation performance, compared with state-of-the-art SSC and LRR. Note that the sample noise introduces some errors into the affinity matrices from BD-SSC and BD-LRR, but the obtained affinity matrix is exactly block-diagonal if shown after proper permutation.

5.2. Face Clustering

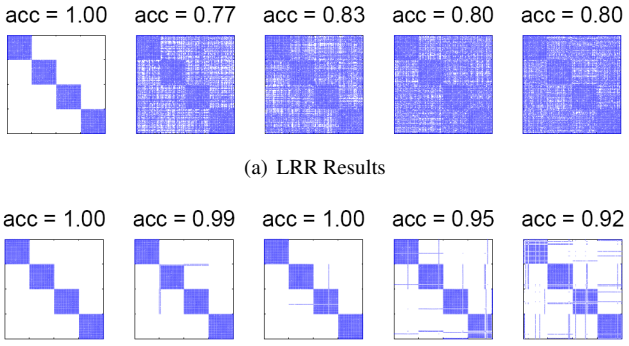
We evaluate the performance of the proposed method, as well as other state-of-the-art methods, for face clustering on the Extended Yale Database B [6]. The dataset contains face images for 38 subjects. For each subject, 64 frontal face images are taken under different illuminations. In this experiment, we use the first $c = \{2, 3, 5, 8, 10\}$ subject classes for face clustering. To reduce the computational cost and the memory requirements of all the methods, each image is resized from 192×168 pixels to 48×42 pixels. The formed



(a) SSC Results

(b) BD-SSC Results

Figure 2. The affinity matrix obtained from SSC and BD-SSC under different noise levels, with $k = 4$ subspaces and 50 samples from each subspace. From left to right, the noise level is $\sigma = 0, 0.1, 0.2, 0.3, 0.4$ respectively. The segmentation accuracy is shown on the top of each sub-figure.



(a) LRR Results

(b) BD-LRR Results

Figure 3. The affinity matrix obtained from LRR and BD-LRR under different noise levels, with $k = 4$ subspaces and 50 samples from each subspace. From left to right, the noise level is $\sigma = 0, 0.1, 0.2, 0.3, 0.4$ respectively. The sample segmentation accuracy is shown on the top of each sub-figure.

$2,016$-dimensional vectorized images are further projected to $9 \times c$-dimensional subspace via standard PCA. Here 9-dimension for each subject is obtained through observing the singular value curve of the matrix of its face image vectors. For SSC and LRR, we use the default parameter settings as provided by the authors. Namely, $\lambda = 20$ for SSC and $\lambda = 0.18$ for LRR. For BD-SSC and BD-LRR we set the value of the trade-off parameter λ the same as SSC and LRR respectively. The iteration number is set as $T = 1,000$ and parameter β in the learning rate estimation is set as $\beta = 1$. Thus the learning rate for BD-SSC is $\eta = 7.6 \times 10^{-4}$ and for BD-LRR $\eta = 3.4 \times 10^{-6}$.

Note that for the SSC subspace segmentation, the authors provide an additional heuristic thresholding operation as post processing to sparsify the obtained affinity matrix [4] for better results. For LRR, there is also a heuristic post

Table 1. Face clustering error (%) on Extended Yale dataset B. The results for $2, 3, 5, 8, 10$ subject classes are shown.

# Classes	SSC	BD-SSC	LRR	BD-LRR
2	9.37	**3.90**	8.59	**3.91**
3	20.13	**17.70**	13.54	**10.02**
5	30.00	**27.50**	15.00	**12.97**
8	36.33	**33.20**	31.64	**27.70**
10	43.59	**39.53**	35.16	**30.84**

processing to scale the obtained affinity matrix [12]. In this experiment, we do not adopt such additional post processing so that other factors can be prevented from damaging the fairness of the comparison. Table 1 lists the face clustering error rate of each method. From the results, we can observe that the proposed methods, BD-SSC and BD-LRR, both outperform their counterpart methods with a margin of 3 to 6 percentages. This demonstrates that the introduced block-diagonal constraint significantly enhances the accuracy and robustness during the affinity matrix construction.

5.3. Motion Segmentation

Motion segmentation refers to clustering the motion trajectories of multiple rigidly moving objects into spatialtemporal regions such that each region corresponds to a single moving object [21]. The coordinates of the points in trajectories of one moving object form a low dimensional subspace. Thus the motion segmentation problem can be solved via performing subspace segmentation on the trajectory spatial coordinates. We use the benchmark dataset Hopkins155 [21] for evaluation. The dataset consists of 156 video sequences of two and three motions, which are divided into three categories: checkerboard, traffic, and articulated sequences. The trajectories are extracted automatically with a tracker, and outliers are manually removed.

The segmentation error rates including their maximal, mean, median and the standard deviation values, for SSC, LRR, BD-SSC and BD-LRR, are shown in Table 2. For all the methods, no pre-/post-processing is performed on the data. From the results, we can observe that BD-SSC and BD-LRR can reduce 0.21% and 1.48% segmentation error rates over SSC and LRR respectively. It is worth noting that the segmentation error rates achieved by SSC and LRR are already quite low, and such improvement is indeed significant. As shown in [4] and [12], the heuristic post-processing for SSC and LRR can improve the motion segmentation performance significantly. In Table 3, we present the results obtained by also applying the thresholding post-processing [4] to SSC and BD-SSC and the scaling post-processing [12] to LRR and BD-LRR. From the results, it can be seen that the proposed method is able to further reduce the segmentation error for SSC and LRR by 0.31% and 0.77% respectively. To the best of our knowledge, our proposed BD-LRR achieves the best performance, 0.97% error rate, for the total 156 sequences on this dataset [4]. Pham *et al.* [16] have reported 0.13% error rate on this dataset. However, their method relies on extra spatial information [16]. In contrast, our method does not need such information.

Table 2. Segmentation errors (%) on Hopkins 155 dataset. The max value (Max), mean value (Mean), median value (Med) and the standard deviation (Std) of the error rates are reported on the total 156 motion sequences.

	SSC	BD-SSC	LRR	BD-LRR
Max	42.34	42.34	41.18	**38.97**
Mean	3.11	**2.90**	4.83	**3.35**
Med	0	0	0.52	**0.37**
Std	7.78	7.48	9.35	8.84

Table 3. Segmentation errors (%) on Hopkins 155 dataset, with post processing. The max value (Max), mean value (Mean), median value (Med) and the standard deviation (Std) of the error rates are reported on the total 156 motion sequences.

	SSC	BD-SSC	LRR	BD-LRR
Max	47.20	43.07	43.38	**14.93**
Mean	1.99	**1.68**	1.74	**0.97**
Med	0	0	0	0
Std	6.97	5.97	5.51	2.46

5.4. Application for Semi-supervised Learning

The proposed BD-SSC and BD-LRR can be directly applied in semi-supervised learning tasks, being used to construct the affinity graph for the (labeled and unlabeled) training samples. Note that graph based semi-supervised learning methods [26] generally include two steps: graph construction and label propagation on the graph. This work focuses on the first step. We mainly compare with other graph construction methods and fix the label propagation method used in the second step in the experiments. Supervised learning methods do not utilize the unlabeled training data and thus may perform worse, thus we do not report their performance here.

In the experiments, two datasets are used for evaluation. One is the USPS handwritten digit dataset [9], which includes 10 classes of digits (0-9) and $11,000$ samples in total. Following the experimental setting in [2], we randomly select 200 samples for each digit character in the experiments, namely $2,000$ samples in total. The other dataset is the Extended Yale Database B as introduced in the previous subsection. Here all the face images of 38 subjects are used. After constructing the affinity matrix, we apply the algorithm of random walk [10] to propagate the class labels from labeled samples to unlabeled samples. On each dataset, four types of popular affinity matrices/graphs are constructed as baselines, including the LLE-graph [18], kNN-graph, ℓ_1-graph constructed using SSC [2] and LRR-graph [12]. For LLE-graph and kNN graph construction, the neighbor size is set as 1% of the total number of samples considering a sparse graph usually achieves better performance [2]. For the ℓ_1-graph construction, the regularization parameter λ is set as 5×10^{-3} and 3×10^{-2} for Yale-B and

Table 4. USPS digit recognition error rate (%) of semi-supervised learning methods on different graphs, using different numbers of labeled samples per class.

# Labeled	LLE	kNN	ℓ_1	BD-SSC	LRR	BD-LRR
10	35.8	21.4	16.1	12.4	13.4	**11.2**
20	24.8	17.0	11.4	**8.9**	11.8	9.4
30	18.0	15.2	11.5	**10.1**	11.2	10.7
40	15.7	11.5	8.2	**6.4**	10.9	6.9

Table 5. Face recognition error rate (%) on Extended Yale dataset B of semi-supervised learning methods on different graphs, using different numbers of labeled samples per class.

# Labeled	LLE	kNN	ℓ_1	BD-SSC	LRR	BD-LRR
5	28.1	76.5	26.8	17.1	28.2	**15.1**
10	15.0	66.3	14.7	**7.6**	24.6	8.2
20	11.0	60.2	12.6	**4.2**	19.4	5.1
30	10.5	59.4	10.8	**2.6**	16.1	4.7

USPS datasets respectively. For LRR-graph, λ is fixed as 0.2 on the both datasets. The results on the two datasets are shown in Table 4 and Table 5 respectively.

From the results, it can be seen that ℓ_1-graph outperforms LLE-graph and kNN-graph significantly on USPS dataset benefitting from the sparsity prior. And on the Yale-B dataset, both ℓ_1-graph and LLE-graph perform quite well since LLE-graph utilizes the correlation of face images. On the both datasets, BD-SSC graph reduces the recognition error rates over the ℓ_1-graph by around 3% and 8% respectively. And BD-LRR graph reduces the recognition error for LRR-graph more significantly. Such improvement is mainly brought by the imposed block-diagonal constraint on the sample affinity matrix, which gives more accurate sample clustering. Since the face images are contaminated by the noises more seriously than the digit images, the performance improvement brought by BD-SSC and BD-LRR on theYale-B dataset is more significant than on the USPS dataset. This demonstrates that the proposed block-diagonal based graph achieves strong robustness to noises.

6. Conclusions

In this work, we proposed a graph Laplacian constraint based formulation to construct exactly block-diagonal affinity matrices. Two new subspace segmentation methods, *i.e.*, BD-SSC and BD-LRR, were devised based on the proposed formulation. Comprehensive experiments were conducted on segmenting synthetic subspaces, realistic faces and motion trajectories. And the proposed methods showed significantly better performance than current state-of-the-art methods. Moreover, BD-SSC and BD-LRR were also applied to construct the block-diagonal ℓ_1-graph and LRR graph for semi-supervised learning and achieved quite encouraging performance for face and digit recognition tasks.

Acknowledgement J. Feng and S. Yan are supported by the Singapore National Research Foundation under its International Research Centre @Singapore Funding Initiative and administered by the IDM Programme Office. Z. Lin is supported by NSF China (grant nos. 61272341, 61231002, 61121002). H. Xu is partially supported by the Ministry of Education of Singapore through AcRF Tier Two grant R-265-000-443-112.

References

[1] A. Beck and M. Teboulle. A linearly convergent algorithm for solving a class of nonconvex/affine feasibility problems. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Springer, 2011. 5

[2] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. Huang. Learning with ℓ^1-graph for image analysis. *TIP*, 2010. 7

[3] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1936. 5

[4] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *TPAMI*, 2013. 1, 2, 3, 6, 7

[5] Y. Fang, R. Wang, and B. Dai. Graph-oriented learning via automatic group sparsity for data analysis. In *ICDM*, 2012. 2

[6] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *TPAMI*, 2001. 6

[7] A. Haim, K. Satyen, K. Shiva, and S. Vikas. Efficient and practical stochastic subgradient descent for nuclear norm regularization. In *ICML*, 2012. 4

[8] J. Ho, M. Yang, J. Lim, K. Lee, and D. Kriegman. Clustering appearances of objects under varying illumination conditions. In *CVPR*, 2003. 1

[9] J. Hull. A database for handwritten text recognition research. *TPAMI*, 1994. 7

[10] M. S. T. Jaakkola and M. Szummer. Partially labeled classification with markov random walks. *NIPS*, 2002. 7

[11] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *Arxiv preprint arXiv:1009.5055*, 2010. 4, 5

[12] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *TPAMI*, 2013. 1, 2, 3, 5, 7

[13] C. Lu, H. Min, Z. Zhao, L. Zhu, D. Huang, and S. Yan. Robust and efficient subspace segmentation via least squares regression. In *ECCV*, 2012. 1, 2

[14] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 2007. 3

[15] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. *NIPS*, 2002. 1

[16] D. Pham, S. Budhaditya, D. Phung, and S. Venkatesh. Improved subspace clustering via exploitation of spatial constraints. In *CVPR*, 2012. 7

[17] S. Rao, R. Tron, R. Vidal, and Y. Ma. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *TPAMI*, 2010. 2

[18] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 2000. 7

[19] M. Soltanolkotabi and E. Candes. A geometric analysis of subspace clustering with outliers. *The Annals of Statistics*, 2012. 1

[20] A. Talwalkar, L. Mackey, Y. Mu, S.-F. Chang, and M. I. Jordan. Distributed low-rank subspace segmentation. In *ICCV*. IEEE, 2013. 6

[21] R. Tron and R. Vidal. A benchmark for the comparison of 3-d motion segmentation algorithms. In *CVPR*, 2007. 1, 7

[22] R. Vidal. Subspace clustering. *IEEE Signal Processing Magazine*, 2010. 1, 2

[23] S. Wang, X. Yuan, T. Yao, S. Yan, and J. Shen. Efficient subspace segmentation via quadratic programming. *AAAI*, 2011. 2

[24] G. Watson. Characterization of the subdifferential of some matrix norms. *Linear Algebra and its Applications*, 1992. 4

[25] A. Yang, J. Wright, Y. Ma, and S. Sastry. Unsupervised segmentation of natural images via lossy data compression. *CVIU*, 2008. 1

[26] X. Zhu. Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison*, 2006. 7