arXiv:1507.01238v2 [cs.CV] 31 Aug 2015

Sparse Subspace Clustering by Orthogonal Matching Pursuit

Chong You and René Vidal
Center for Imaging Science, Johns Hopkins University, Baltimore, MD, 21218, USA
{cyou,rvidal}@cis.jhu.edu

Abstract

*Subspace clustering methods based on ℓ_1, ℓ_2 or nuclear norm regularization have become very popular due to their simplicity, theoretical guarantees and empirical success. However, the choice of the regularizer can greatly impact both theory and practice. For instance, ℓ_1 regularization is guaranteed to give the correct clustering under broad conditions (*e.g.*, arbitrary subspaces and corrupted data), but requires solving a large scale convex optimization problem. On the other hand, ℓ_2 and nuclear norm regularization provide efficient closed form solutions, but require very strong assumptions to guarantee the correct clustering (*e.g.*, independent subspaces and uncorrupted data). This paper proposes a new subspace clustering method based on orthogonal matching pursuit that is computationally efficient and guaranteed to provide the correct clustering for arbitrary subspaces. Experiments on synthetic data verify our theoretical analysis, and applications in handwritten digit and face clustering show that our approach achieves the best trade off between accuracy and efficiency. Moreover, our approach is the only one that can handle $100,000$ points.*

1. Introduction

In many computer vision applications, such as motion segmentation [14, 29, 52, 40], hand written digit clustering [57] and face clustering [5, 31], data from different classes can be well approximated by a union of low-dimensional subspaces. In these scenarios, the task is to partition the data according to the membership of data points to subspaces.

More formally, given a set of points $\{\boldsymbol{x}_j \in \mathbb{R}^D\}_{j=1}^N$ lying in an unknown number n of subspaces $\{S_i\}_{i=1}^n$ of unknown dimensions $\{d_i\}_{i=1}^n$, *subspace clustering* is the problem of clustering the data into several groups so that each group contains data points from the same subspace. This problem has received great attention in the past decade (see [47]) and many subspace clustering algorithms have been developed, including iterative methods [8, 46, 1, 2, 36, 56, 57], algebraic methods [6, 14, 24, 50, 49, 51, 28, 16, 38], statistical methods [44, 32, 43, 27, 55, 37, 3], and spectral clustering based methods [6, 54, 21, 26, 12, 18, 19, 20, 34, 57, 48].

Sparse and Low Rank Methods. Among existing spectral subspace clustering techniques, methods based on applying spectral clustering to an affinity matrix obtained by ℓ_1, ℓ_2 or nuclear norm regularization have become extremely popular due to their simplicity, theoretical correctness, and empirical success. These methods are based on the so-called *self-expressiveness property* of data lying in a union of subspaces, originally proposed in [18]. This property states that each point in a union of subspaces can be written as a linear combination of other data points in the subspaces. That is,

$$\boldsymbol{x}_j = X\boldsymbol{c}_j, \;\; c_{jj} = 0, \;\; \text{or} \;\; X = XC, \;\; \text{diag}(C) = \boldsymbol{0}, \;\; (1)$$

where $X = \big[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\big] \in \mathbb{R}^{D\times N}$ is the data matrix and $C = \big[\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N\big] \in \mathbb{R}^{N\times N}$ is the matrix of coefficients.

While the solution of (1) need not be unique, there exist solutions C whose entries are such that $c_{ij} \neq 0$ if $\boldsymbol{x}_i$ is in the same subspace as $\boldsymbol{x}_j$. For example, a point $\boldsymbol{x}_j \in S_i$ can always be written as a linear combination of d_i points in S_i. Such solutions are called *subspace preserving* since they preserve the clustering of the subspaces. Given a subspace preserving C, one can build an affinity matrix W between every pair of points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ as $w_{ij} = |c_{ij}| + |c_{ji}|$, and use spectral clustering [53] to cluster the data.

To find a subspace preserving C, existing methods regularize C with a norm $\|\cdot\|$, and solve a problem of the form:

$$C^* = \arg\min_C \|C\| \;\; \text{s.t.} \;\; X = XC, \;\; \text{diag}(C) = \boldsymbol{0}. \;\; (2)$$

The sparse subspace clustering (SSC) algorithm [18, 20], for instance, uses the ℓ_1 norm of C to encourage sparsity. It has been shown that the nonzero entries of each column $\boldsymbol{c}_j$ correspond to points from the same subspace as $\boldsymbol{x}_j$ provided that the subspaces are independent [18], or disjoint and sufficiently separated and the data is well spread out within each subspace [19, 20]. The work of [41] further extended the applicability of SSC to the case of arbitrary subspaces (*i.e.* the requirement of disjointness is relaxed) as well as to the case where the data is contaminated by outliers.

Other self-expressiveness based methods use different regularizations on the coefficient matrix C. Least squares

regression (LSR) [35] uses ℓ_2 regularization on C, while low rank representation (LRR) [34, 33] and low rank subspace clustering (LRSC) [22, 48] use nuclear norm minimization to encourage C to be low-rank. The advantage of LSR, LRR and LRSC over SSC is that the solution for C can be computed in closed form from the SVD of the data matrix X, thus they are computationally more attractive. However, the resulting C is subspace preserving only when subspaces are independent and the data is uncorrupted.

In conclusion, there is a need for developing new methods that both guarantee the correctness of clustering under broad conditions and are computationally efficient.

Paper Contributions. This work proposes a new self-expressiveness based subspace clustering method that uses orthogonal matching pursuit (OMP) to find a sparse representation in lieu of the ℓ_1-based basis pursuit (BP) method. Our method is termed SSC-OMP, for its kinship to the original SSC, which is referred to as SSC-BP in this paper.

The main contributions of this paper are to find theoretical conditions for the correctness of SSC-OMP and to demonstrate its efficiency. More specifically, we show that

1. SSC-OMP gives a subspace-preserving C for independent subspaces, or for arbitrary subspaces that are sufficiently separated when the data is well distributed.
2. SSC-OMP gives subspace-preserving C if the subspaces and data are drawn uniformly at random, and the dimensions of the subspaces are sufficiently small.
3. SSC-OMP is orders of magnitude faster than the original SSC-BP, and can handle up to $100,000$ data points.

It is worth noting that the idea of using OMP for SSC had already been considered in a recent work [17]. The core contribution of our work is to provide much weaker conditions for the correctness of clustering as well as a much more detailed experimental evaluation and comparison.

Paper Outline. The remainder of the paper is organized as follows. In Section 2 we formulate the problem to be studied and review the OMP algorithm. In Section 3 we analyze SSC-OMP for a deterministic model where the subspaces and the data are fixed, as well as for a model where both the subspaces and the data points are drawn at random. In Section 4 we compare our results with those in prior work. In Section 5 we present experiments that verify our theoretical analysis and demonstrate the performance of SSC-OMP for vision problems. Finally, we draw conclusions in Section 6.

2. SSC by Orthogonal Matching Pursuit

Let $\mathcal{X} = \{\boldsymbol{x}_j\}_{j=1}^N$ be a set of points in $\mathbb{R}^D$ drawn from n subspaces $\{S_i\}_{i=1}^n$ of dimensions $\{d_i\}_{i=1}^n$. The goal of subspace clustering is to partition $\mathcal{X}$ into n groups corresponding to the data points in each one of the subspaces. Throughout the paper we will assume that all data points have been normalized to unit ℓ_2 norm, and that no point is in the intersection of any two subspaces. We will denote the matrix whose columns are the data points as $X \in \mathbb{R}^{D\times N}$.

The SSC algorithm approaches the subspace clustering problem by finding a sparse representation of each point in terms of other data points. Since each point in S_i can be expressed in terms of at most $d_i \ll N$ other points in S_i, such a sparse representation always exists. In principle, we can find it by solving the following optimization problem:

$$\boldsymbol{c}_j^* = \arg\min_{\boldsymbol{c}_j} \|\boldsymbol{c}_j\|_0 \ \text{s.t.}\ \boldsymbol{x}_j = X\boldsymbol{c}_j, c_{jj} = 0, \quad (3)$$

where $\|\boldsymbol{c}\|_0$ counts the number of nonzero entries in $\boldsymbol{c}$. This problem has been studied extensively in the compressed sensing community, see, *e.g.*, the tutorials [9, 11]. Since this problem is NP hard, the SSC algorithm in [18] relaxes this problem and solves the following ℓ_1 minimization problem

$$\boldsymbol{c}_j^* = \arg\min_{\boldsymbol{c}_j} \|\boldsymbol{c}_j\|_1 \ \text{s.t.}\ \boldsymbol{x}_j = X\boldsymbol{c}_j, \ \ c_{jj} = 0. \quad (4)$$

This problem is often termed as basis pursuit (BP), hence we name the SSC algorithm in [18] as SSC-BP.

It is shown in [18, 19, 20] that both (3) and (4) provide the correct clustering when the subspaces are either independent or disjoint, and the data are noise free and well distributed. By correct clustering we mean that each sparse representation $\boldsymbol{c}_j$ is *subspace preserving*, as defined next.

Definition 1 (Subspace-preserving representation)**.** *A representation $\boldsymbol{c} \in \mathbb{R}^N$ of a point $\boldsymbol{x} \in S_i$ in terms of the dictionary $X = \big[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\big]$ is called subspace preserving if its nonzero entries correspond to points in S_i,* i.e.

$$\forall j = 1, \ldots, N, \quad c_j \neq 0 \implies \boldsymbol{x}_j \in S_i. \quad (5)$$

In practice, however, solving N ℓ_1-minimization problems over N variables may be prohibitive when N is large. This motivates us to propose an alternative subspace clustering approach based on the following optimization problem:

$$\boldsymbol{c}_j^* = \arg\min_{\boldsymbol{c}_j} \|\boldsymbol{x}_j - X\boldsymbol{c}_j\|^2 \ \text{s.t.}\ \|\boldsymbol{c}_j\|_0 \leq k, \ c_{jj} = 0. \quad (6)$$

In particular, we propose to solve this problem approximately using the orthogonal matching pursuit (OMP) algorithm proposed in [39]. OMP (see Algorithm 1) solves the problem $\min_{\boldsymbol{c}} \|A\boldsymbol{c} - \boldsymbol{b}\|_2^2$ s.t. $\|\boldsymbol{c}\|_0 \leq k$ greedily by selecting one column of $A = \big[\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N\big]$ at a time and computing the coefficients for the support set of columns until k columns are selected. For subspace clustering purposes, we compute each column of C, $\boldsymbol{c}_j$, as $\text{OMP}(X_{-j}, \boldsymbol{x}_j)$, where $\boldsymbol{c}^* = \text{OMP}(A, b)$ denotes the output of Algorithm 1[1]

[1] If $\arg\max$ in step 3 of the algorithm gives multiple items, pick one of them in a deterministic way, *e.g.*, pick the one with the smallest value.

and X_{-j} is the data matrix with the j-th column removed. Given C, the segmentation of the data is found by applying spectral clustering to the affinity matrix $W = |C| + |C^\top|$ as done in SSC-BP.

Algorithm 1 Orthogonal Matching Pursuit

Input: $A = [\boldsymbol{a}_1, \dots, \boldsymbol{a}_M] \in \mathbb{R}^{m \times M}$, $\boldsymbol{b} \in \mathbb{R}^m$, $k_{\max}$ and ϵ.

1: Initialize $k = 0$, residual $\boldsymbol{q}_0 = \boldsymbol{b}$, support set $T_0 = \emptyset$.
2: **while** $k < k_{\max}$ and $\|\boldsymbol{q}_k\|_2 > \epsilon$ **do**
3: $T_{k+1} = T_k \bigcup \{i^*\}$, where $i^* = \arg\max_{i=1,\cdots,M} |\boldsymbol{a}_i^T \boldsymbol{q}_k|$.
4: $\boldsymbol{q}_{k+1} = (I - P_{T_{k+1}})\boldsymbol{b}$, where $P_{T_{k+1}}$ is the projection onto the space spanned by columns indexed by T_{k+1}.
5: $k \leftarrow k + 1$.
6: **end while**

Output: $\boldsymbol{c}^* = \arg\min_{\boldsymbol{c}:\text{Supp}(\boldsymbol{c}) \subseteq T_k} \|\boldsymbol{b} - A\boldsymbol{c}\|_2$.

3. Conditions for Correctness of SSC-OMP

OMP has been shown to be effective for sparse recovery, with the advantage over BP that it admits simple, fast implementations. However, note that existing conditions for the correctness of OMP for sparse recovery are way too strong for the subspace clustering problem. In particular, note that the matrix X need not satisfy the mutual incoherence [45] or restricted isometry properties [15]. For, two points in a subspace could be arbitrarily close to each other, which would violate mutual incoherence. More importantly, these conditions are not applicable here because our goal is not to recover a "unique" sparse solution. In fact, the sparse representation is not unique since any d_i linearly independent points from S_i can be used to represent a point $\boldsymbol{x}_j \in S_i$. Therefore, there is a need to find conditions under which the output of OMP (which need not coincide with the solution of (6) or (3)) is guaranteed to be subspace preserving.

This section is devoted to studying sufficient conditions under which SSC-OMP gives a subspace preserving representation. Since we will be mostly concerned with the case of noiseless data, our results assume that the termination parameters of Algorithm 1 are $\epsilon = 0$ and k_{max} large enough (*e.g.*, $k_{max} = M$). To make our results consistent with state of the art results, we first study the case where the subspaces are deterministic, including both independent subspaces as well as arbitrary subspaces. We then study the case where both the subspaces and the data points are drawn at random.

3.1. Independent Deterministic Subspace Model

We first consider the case where the subspaces are fixed, the data points are fixed, and the subspaces are independent.

Definition 2. *A collection of subspaces $\{S_i\}_{i=1}^n$ is called independent if* $\dim\left(\sum_i S_i\right) = \sum_i \dim(S_i)$*, where $\sum_i S_i$ is defined as the subspace $\{\sum_i \boldsymbol{x}_i : \boldsymbol{x}_i \in S_i\}$.*

Notice that two subspaces are *independent* if and only if they are *disjoint*, *i.e.*, if they intersect only at the origin. However, pairwise disjoint subspaces need not be independent, *e.g.*, three lines in $\mathbb{R}^2$ are disjoint but not independent. Notice also that any subset of a set of independent subspaces is also independent. Therefore, any two subspaces in a set of independent subspaces are independent and hence disjoint. Finally, notice that if $\{S_i\}_{i=1}^n$ are independent, then S_i and $S_{(-i)} := \sum_{m \neq i} S_m$ are independent.

Now, to establish conditions for correctness of OMP for independent subspaces, it is important to note that when solving the problem OMP$(X_{-j}, \boldsymbol{x}_j)$ our goal is to select other points in the same subspace as $\boldsymbol{x}_j$. The process for selecting these points occurs in step 3 of Algorithm 1, where the dot products between all points $\boldsymbol{x}_m$, $m \neq j$, and the current residual $\boldsymbol{q}_k$ are computed and the point with the highest product is chosen. Since in the first iteration the residual is $\boldsymbol{q}_0 = \boldsymbol{x}_j$, we could immediately choose a point $\boldsymbol{x}_m$ in another subspace whenever the dot product of $\boldsymbol{x}_j$ with a point in another subspace is larger than the dot product of $\boldsymbol{x}_j$ with points in its own subspace. What is interesting about the proof of Theorem 1, which is provided in the appendix, is that, even though OMP may select points in the wrong subspaces as iterations proceed, the final coefficient c_{mj} associated to a point $\boldsymbol{x}_m$ in other subspaces will be zero. Therefore, OMP is guaranteed to find a subspace-preserving representation, as stated in the next theorem.

Theorem 1. *If the subspaces are independent, OMP gives a subspace-preserving representation of each data point.*

3.2. Arbitrary Deterministic Subspace Model

We will now consider a more general class of subspaces, which need not be independent or disjoint, and investigate conditions under which OMP gives a subspace-preserving representation. As described before, a sufficient condition for the output of OMP$(X_{-j}, \boldsymbol{x}_j)$ to be subspace preserving is that, in step 3 of Algorithm 1, the point that maximizes the dot product lies in the same subspace as $\boldsymbol{x}_j$. Since $\boldsymbol{q}_0 = \boldsymbol{x}_j$ and $\boldsymbol{q}_1$ is equal to $\boldsymbol{x}_j$ minus the projection of $\boldsymbol{x}_j$ onto the subspace spanned by the selected point, say $\boldsymbol{x}_m$, it follows that if $\boldsymbol{x}_j, \boldsymbol{x}_m \in S_i$ then $\boldsymbol{q}_1 \in S_i$. By a simple induction argument, it follows that if all the selected points are in S_i, then so are the residuals $\{\boldsymbol{q}_k\}$. This suggests that the condition for correctness of OMP for $\boldsymbol{x}_j$ must depend on the dot products between the data points and a subset of the set of residuals (the subset contained in the same subspace as $\boldsymbol{x}_j$). This motivates the following definition and lemma.

Definition 3. *Let $Q(A, \boldsymbol{b})$ be the set of all residual vectors in step 4 of OMP$(A, \boldsymbol{b})$. Assume that $\boldsymbol{x}_j \in S_i$ and let $X^i \in \mathbb{R}^{D \times N_i}$ be the submatrix of X containing the points in the i-th subspace. Also, let $X^i_{-j} \in \mathbb{R}^{D \times (N_i - 1)}$ be the matrix X^i except for point $\boldsymbol{x}_j$. The set of OMP residual directions*

associated with matrix X^i_{-j} and point $\boldsymbol{x}_j \in S_i$ is defined as:

$$\mathcal{W}^i_j := \{\boldsymbol{w} = \frac{\boldsymbol{q}}{\|\boldsymbol{q}\|_2} : \boldsymbol{q} \in Q(X^i_{-j}, \boldsymbol{x}_j), \boldsymbol{q} \neq \boldsymbol{0}\}. \quad (7)$$

The set of OMP residual directions associated with the data matrix X^i is defined as $\mathcal{W}^i := \cup_{j:\boldsymbol{x}_j \in S_i} \mathcal{W}^i_j$.

Lemma 1. *OMP gives a subspace-preserving representation for point $\boldsymbol{x}_j \in S_i$ if*

$$\forall \boldsymbol{w} \in \mathcal{W}^i_j \quad \max_{k:\boldsymbol{x}_k \notin S_i} |\boldsymbol{w}^\top \boldsymbol{x}_k| < \max_{\substack{k:\boldsymbol{x}_k \in S_i \\ k \neq j}} |\boldsymbol{w}^\top \boldsymbol{x}_k|. \quad (8)$$

Proof. [Sketch only] By using an induction argument, it is easy to see that the condition in (8) implies that the sequence of residuals of OMP$(X_{-j}, \boldsymbol{x}_j)$ is the same as that of the fictitious problem OMP$(X^i_{-j}, \boldsymbol{x}_j)$. Hence, the output of OMP$(X_{-j}, \boldsymbol{x}_j)$ is the same as that of OMP$(X^i_{-j}, \boldsymbol{x}_j)$, which is, by construction, subspace-preserving. □

Intuitively, Lemma 1 tells us that if the dot product between the residual directions for subspace i and the data points in all other subspaces is smaller than the dot product between the residual directions for subspace i and all points in subspace i other than $\boldsymbol{x}_j \in S_i$, then OMP gives a subspace preserving representation. While such a condition is very intuitive from the perspective of OMP, it is not as intuitive from the perspective of subspace clustering as it does not rely on the geometry of the problem. Specifically, it does not directly depend on the relative configuration of the subspaces or the distribution of the data in the subspaces. In what follows, we try to derive conditions on the subspaces and the data that guarantee that the condition in (8) holds. Before doing so, we will need some additional definitions.

Definition 4. *The coherence between two sets of points $\mathcal{X}$ and $\mathcal{Y}$ is defined as $\mu(\mathcal{X}, \mathcal{Y}) = \max_{\boldsymbol{x} \in \mathcal{X}, \boldsymbol{y} \in \mathcal{Y}} |\langle \boldsymbol{x}, \boldsymbol{y} \rangle|$.*

The coherence measures the degree of "similarity" between two sets of points. In our case, we can see that the left hand side of (8) is bounded above by the coherence between the sets $\mathcal{W}^i_j$ and $\cup_{k \neq i} \mathcal{X}^k$. As per (8), this coherence should be small, which implies that data points from different subspaces should be sufficiently separated (in angle).

Definition 5. *The inradius $r(\mathcal{P})$ of a convex body $\mathcal{P}$ is the radius of the largest Euclidean ball inscribed in $\mathcal{P}$.*

As shown in Lemma 2, the right hand side of (8) is bounded below by $r(\mathcal{P}^i_{-j})$, where $\mathcal{P}^i_{-j} := \text{conv}\left(\pm \mathcal{X}^i_{-j}\right)$ is the symmetrized convex hull of the points in the i-th subspace other than $\boldsymbol{x}_j$, *i.e.*, $\mathcal{X}^i_{-j}$. Therefore, (8) suggests that the inradius $r_i := \min_j r(\mathcal{P}^i_{-j})$ should be large, which implies that the points in S_i should be well-distributed.

Lemma 2. *Let $\boldsymbol{x}_j \in S_i$. Then, for all $\boldsymbol{w} \in \mathcal{W}^i_j$, we have:*

$$\max_{k:\boldsymbol{x}_k \notin S_i} |\boldsymbol{w}^\top \boldsymbol{x}_k| \leq \max_{k:k \neq i} \mu(\mathcal{W}^i, \mathcal{X}^k) \leq \max_{k:k \neq i} \mu(\mathcal{X}^i, \mathcal{X}^k)/r_i;$$
$$\max_{k:\boldsymbol{x}_k \in S_i, k \neq j} |\boldsymbol{w}^\top \boldsymbol{x}_k| \geq r(\mathcal{P}^i_{-j}) \geq r_i. \quad (9)$$

The proof can be found in the appendix. It allows us to make the condition of Lemma 1 more interpretable, as stated in the following theorem.

Theorem 2. *The output of OMP is subspace preserving if*

$$\forall i = 1, \ldots, n, \quad \max_{k:k \neq i} \mu(\mathcal{W}^i, \mathcal{X}^k) < r_i. \quad (10)$$

Corollary 1. *The output of OMP is subspace preserving if*

$$\forall i = 1, \ldots, n, \quad \max_{k:k \neq i} \mu(\mathcal{X}^i, \mathcal{X}^k) < r_i^2. \quad (11)$$

Note that points in $\mathcal{W}^i$ are all in subspace S_i, as step 4 of OMP$(A := X^i_{-j}, \boldsymbol{b} := \boldsymbol{x}_j)$ has $\boldsymbol{b}$ and $P_{T_{k+1}}\boldsymbol{b}$ both in S_i. The conditions (10) and (11) thus show that for each subspace S_i, a set of points (*i.e.*, $\mathcal{X}^i$ or $\mathcal{W}^i$) in S_i should have low coherence with all points from other subspaces, and that points in $\mathcal{X}^i$ should be uniformly located in S_i to have a large inradius. This is in agreement with the intuition that points from different subspaces should be well separated, and points within a subspace should be well distributed.

For a comparison of Corollary 1 and Theorem 2, note that due to Lemma 2 condition (10) is tighter than condition (11), making Theorem 2 preferable. Yet Corollary 1 has the advantage that both sides of condition (11) depend directly on the data points in $\mathcal{X}$, while condition (10) depends on the residual points in $\mathcal{W}^i$, making it algorithm-specific.

Another important thing to notice is that conditions (10) and (11) can be satisfied even if the subspaces are *neither independent nor disjoint*. For example, consider the case where $S_i \cap S_k \neq \boldsymbol{0}$. Then, the coherence $\mu(\mathcal{W}^i, \mathcal{X}^k)$ could still be bounded away from $+1$ as long as no points in $\mathcal{W}^i$ and $\mathcal{X}^k$ are near the intersection of S_i and S_k. Actually, even this is too strong of an assumption since the intersection is a subspace that contains multiple points. Thus even if $\boldsymbol{x} \in \mathcal{X}^k$, $\boldsymbol{y} \in \mathcal{W}^i$ are both very close to the intersection, it is still possible that they are far apart from each other. The same argument also works for condition (11). Admittedly, under specific distributions of points, it is possible that there exists $\boldsymbol{x} \in \mathcal{X}^k$ and $\boldsymbol{y} \in \mathcal{W}^i$ that are arbitrarily close to each other when they are near the intersection. However, this worst case scenario is unlikely to happen if we consider a random model, as discussed next.

3.3. Arbitrary Random Subspace Model

This section considers the fully random model in [41], where all the subspaces are distributed uniformly at random in the ambient space of dimension D and all the points are

uniformly distributed on the unit sphere of each subspace. Theorem 3 shows that the sufficient condition in (10) holds true with high probability given some conditions on the subspace dimension, the ambient space dimension, the number of subspaces and the number of data points.

Theorem 3. *Assume a random model where all subspaces are of equal dimension d and the number of data points in each subspace is $\rho d+1$, where $\rho > 1$ is the "density", so the total number data points in all subspaces is $N = n(\rho d+1)$. The output of OMP is subspace preserving with probability $p > 1 - \frac{2d}{N} - Ne^{-\sqrt{\rho} d}$ if*

$$d < \frac{c^2(\rho) \log \rho}{12} \frac{D}{\log N}, \tag{12}$$

where $c(\rho) > 0$ is a constant that depends only on ρ.

The above result shows that our sufficient conditions in Theorem 2 is satisfied with high probability under some conditions. One interpretation of the condition in (12) is that the dimension d of the subspaces should be small relative to the ambient dimension D. It also shows that as the number of subspaces n increases, the factor $\log N$ also increases, making the condition more difficult to be satisfied. In terms of the density ρ, one can show that there exists a ρ_0 such that $c(\rho) = 1/\sqrt{8}$ when $\rho > \rho_0$. Then, it is easy to see that when $\rho > \rho_0$, the term that depends on ρ is $\frac{\log \rho}{\log N} = \frac{\log \rho}{\log(n(\rho d+1))}$, which is a monotonically increasing function of ρ. This makes the condition easier to be satisfied as the density of points in the subspaces increases. Moreover, the probability of success is $1 - \frac{2d}{N} - Ne^{-\sqrt{\rho} d}$, which is also an increasing function of ρ when ρ is greater than a threshold. As a consequence, as the density of the points increases, the condition in Theorem 3 becomes easier to satisfy and the probability of success also increases.

4. Relationships with Other Methods

In this section we compare our results for SSC-OMP with those for other methods of the general form in (2). These methods include SSC-BP [19, 20, 41], which uses the ℓ_1 norm as a regularizer, LRR [33] and LRSC [48], which use the nuclear norm, and LSR [35] which uses the ℓ_2 norm. We also compare our results to those of [17] for SSC-OMP. The comparison is in terms of whether the solutions given by these alternative algorithms are subspace-preserving.

Independent Subspaces. Independence is a strong assumption on the union of subspaces. Under this assumption, a subspace has a trivial intersection with not only every other subspace, but also the union of all other subspaces. This case turns out to be especially easy for a large category of self-expressive subspace clustering methods [35], and SSC-BP, LRR, LRSC and LSR are all able to give subspace-preserving representations. Thus, in this easy case, the proposed method is as good as state of the art methods.

Arbitrary Subspaces. To the best of our knowledge, when the subspaces are not independent, there is no guarantee of correctness for LRR, LRSC and LSR. For SSC-BP, as shown in [41], the representation is subspace-preserving if

$$\forall i = 1, \cdots, n, \quad \max_{k:k \neq i} \mu(\mathcal{V}^i, \mathcal{X}^k) < r_i, \tag{13}$$

where $\mathcal{V}^i$ is a set of N_i *dual directions* associated with X^i. When comparing (13) with our result in condition (10), we can see that the right hand sides are the same. However, the left hand sides are not directly comparable, as there is no general relationship between the sets $\mathcal{V}^i$ and $\mathcal{W}^i$. Nonetheless, notice that the number of points in these two sets are not the same since $\text{card}(\mathcal{V}^i) = N_i$ and $\text{card}(\mathcal{W}^i) = N_i d_i$. Therefore, if we assume that the points in $\mathcal{V}^i$ and $\mathcal{W}^i$ are distributed randomly, then $\mu(\mathcal{W}^i, \mathcal{X}^k)$ is expected to be larger than $\mu(\mathcal{V}^i, \mathcal{X}^k)$, making the condition for SSC-OMP less likely to be satisfied than that for SSC-BP. Now, when comparing (13) with our condition in (11), we see that the left hand sides are comparable under a random model where both $\mathcal{V}^i$ and $\mathcal{X}^i$ contain N_i points. However, the right hand side is r_i^2, which is less than or equal to r_i since the data are normalized and $r_i \leq 1$. This again makes the condition for SSC-OMP more difficult to hold than that for SSC-BP.

Random Subspaces. For the random model, [41] shows that SSC-BP gives a subspace-preserving representation with probability $p > 1 - \frac{2}{N} - Ne^{\sqrt{\rho} d}$ if

$$d < \frac{c^2(\rho) \log \rho}{12} \frac{D}{\log N}. \tag{14}$$

If we compare this result with that of Theorem 3, we can see that the condition under which both methods succeed with high probability (*i.e.* a probability that converges to 1 as the number of points grows to infinity) is exactly the same. The difference between them is that SSC-BP has a higher probability of success than SSC-OMP when $d > 1$. However, the difference in probability goes to zero as the density ρ goes to infinity. This means that the performance difference vanishes as the scale of the problem increases.

Other Results for SSC-OMP. Finally, we compare our results with those in [17] for SSC-OMP. Define the principal angles between two subspaces as:

$$\theta^*_{i,k} = \min_{\substack{\boldsymbol{x} \in S_i \\ \|\boldsymbol{x}\|_2 = 1}} \min_{\substack{\boldsymbol{y} \in S_k \\ \|\boldsymbol{y}\|_2 = 1}} \arccos \langle \boldsymbol{x}, \boldsymbol{y} \rangle. \tag{15}$$

It is shown in [17] that the output of SSC-OMP is subspace-preserving if for all $i = 1, \ldots, n$,

$$\max_{k:k \neq i} \mu(\mathcal{X}^i, \mathcal{X}^k) < r_i - \frac{2\sqrt{1 - (r_i)^2}}{\sqrt[4]{12}} \max_{k:k \neq i} \cos \theta^*_{i,k}. \tag{16}$$

The merit of this result is that it introduces the subspace angles in the condition, and satisfies the intuition that the

algorithm is more likely to work if subspaces are far apart from each other. However, as we proof in the appendix, this result is weaker than our result in Theorem 2.

5. Experiments

In this section, we first verify our theoretical results for SSC-OMP and compare them with those for SSC-BP by doing experiments on synthetic data using the random model. Specifically, we show that even if the subspaces are not independent, the sparse representation is subspace-preserving with a probability that grows with the number of data points. Second, we test the performance of the proposed method on handwritten digit clustering and face image clustering tasks, and conclude that SSC-OMP achieves the best trade off between accuracy and efficiency.

Methods. We compare the performance of state of the art spectral subspace clustering methods, including LRSC [48], SSC-BP [20], LSR [35], and spectral curvature clustering (SCC) [12]. For all methods, we use the normalized spectral clustering in [53] with 20 random initialization in k-means, except for SCC which has its own spectral clustering step.

Solvers. OMP is coded according to Algorithm 1 with ϵ set to 10^{-3}. $k_{\max}$ is set to the true subspace dimension in the synthetic experiments, 10 for digit clustering and 5 for face clustering. For LSR, we use the model "LSR2" in [35] with regularization parameter $\lambda = 60$ for digit clustering and $\lambda = 0.3$ for face clustering. For LRSC, we use model "P_3" in [48], with parameters $\tau = \alpha = 0.1$ for digit clustering and $\tau = \alpha = 150$ for face clustering. For SCC, we use dimension $d = 8$ for digit clustering and 5 for face clustering, and set all other parameters to their default values in their code. For SSC-BP, we use ℓ_1-Magic for the synthetic experiments, and the alternating direction method of multipliers (ADMM) in [20] for digit and face clustering. ℓ_1-Magic is based on interior-point methods and gives relatively precise solutions, so it is suitable to study the subspace-preserving property of the method. On the other hand, ADMM [7] can be very slow to converge to high accuracy but can get to modest accuracy in a few tens of iterations, which is usually enough for subspace clustering problems. We use the noisy variation of SSC-BP in [20] for digit clustering with $\lambda_z = 80/\mu_z$, and the sparse outlying entries variation of SSC-BP in [20] for face clustering with $\lambda_e = 30/\mu_e$.

Pre/post-processing. The codes provided by the authors of different methods usually do different preprocessing on the data and/or post-processing on the coefficient matrix C. For a fair and easy comparison, we allow the standard pre/post-processing specified below and apply them whenever they improve the clustering accuracy. For preprocessing, we allow normalizing the original data points to unit ℓ_2 norm, and for post-processing, we allow normalizing the coefficient vectors to unit ℓ_∞ norm. For synthetic experiment we do not use any pre/post-processing. In digit clustering, preprocessing is applied to SSC-BP and SCC, and post-processing is used for SSC-OMP and SSC-BP. For face clustering, preprocessing is applied to SSC-OMP, LSR, LRSC and SCC, while post-processing is used for SSC-BP and LRSC.

Metrics. We use two metrics to evaluate the degree to which the subspace-preserving property is satisfied. The first one is a direct measure of whether the solution is subspace-preserving or not. However, for comparing with state of the art methods whose output is generally not subspace-preserving, the second one measures how close the coefficients are from being subspace-preserving.

– Percentage of subspace-preserving representations ($p\%$): this is the percentage of points whose representations are subspace-preserving. Due to imprecision of solvers, coefficients with absolute value less than 10^{-3} are considered zero. A subspace-preserving solution gives $p = 100$.

– Subspace-preserving representation error ($e\%$) [20]: for each $\boldsymbol{c}_j$ in (1), we compute the fraction of its ℓ_1 norm that comes from other subspaces and then average over all j, *i.e.*, $e = \frac{1}{N}\sum_j(1-\sum_i(\boldsymbol{w}_{ij}\cdot|\boldsymbol{c}_{ij}|)/\|\boldsymbol{c}_j\|_1)$, where $w_{ij} \in \{0,1\}$ is the true affinity. A subspace-preserving C gives $e = 0$.

Now, the performance of subspace clustering depends on not only the subspace-preserving property, but also the connectivity of the similarity graph, *i.e.*, whether the data points in each cluster form a connected component of the graph.

– Connectivity (c): For an undirected graph with weights $W \in \mathbb{R}^{N\times N}$ and degree matrix $D = \text{diag}(W \cdot \mathbf{1})$, we use the second smallest eigenvalue λ_2 of the normalized Laplacian $L = I - D^{-1/2}WD^{-1/2}$ to measure the connectivity of the graph. λ_2 is in the range $[0, \frac{n-1}{n}]$ and is zero if and only if the graph is not connected [23, 13]. In our case, we compute the algebraic connectivity for each cluster, λ_2^i, and take the minimum $c = \min_i \lambda_2^i$ as the connectivity measure.

Finally, we use the following two metrics to evaluate the performance of subspace clustering methods.

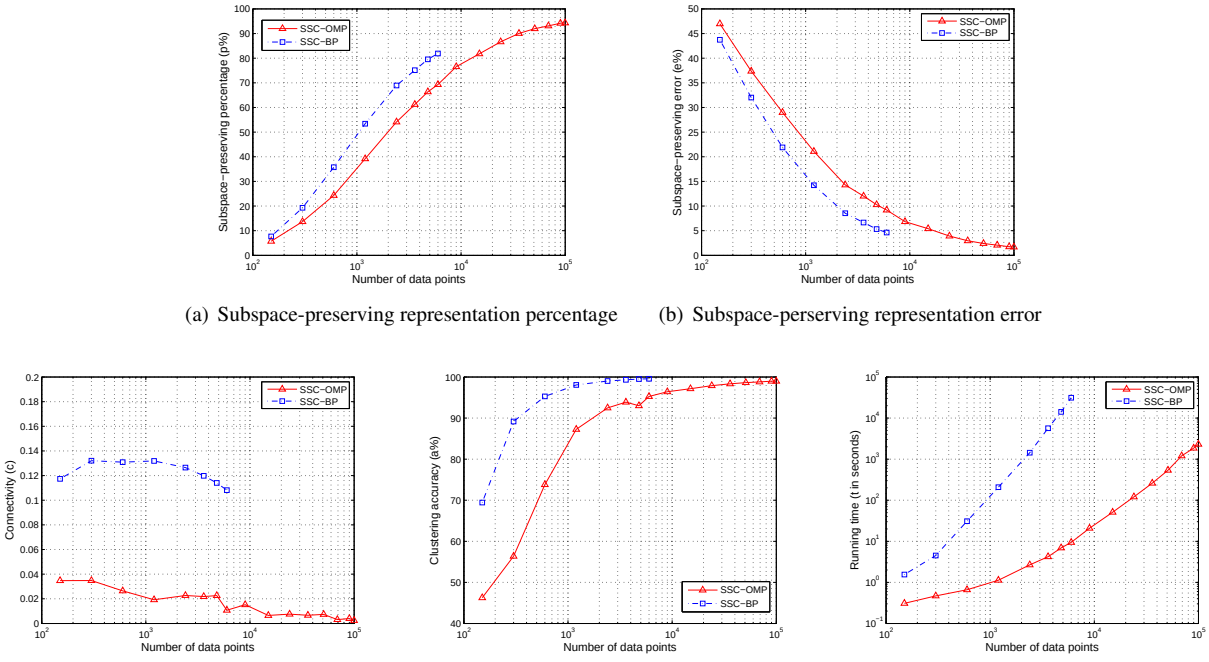
– Clustering accuracy ($a\%$): this is the percentage of correctly labeled data points. It is computed by matching the estimated and true labels as $a = \max_\pi \frac{1}{N}\sum_{ij} L^{est}_{\pi(i)j} L^{true}_{ij}$, where π is a permutation of the n groups, and $L_{ij} = 1$ if point j belongs to cluster i and $L_{ij} = 0$ otherwise.

– Running time (t): for each clustering task using ®Matlab.

The reported numbers in all the experiments of this section are averages over 20 trials.

5.1. Synthetic Experiments

We randomly generate $n = 5$ subspaces each of dimension $d = 6$ in an ambient space of dimension $D = 9$. Each subspace contains $N_i = \rho d$ sample points randomly generated in the unit sphere, where ρ is varied from 5 to 3,333, so that the number of points varies from 150 to 99,990. Due to the computational complexity, SSC-BP is run for $\rho \leq 200$.



(a) Subspace-preserving representation percentage (b) Subspace-perserving representation error

(c) Connectivity (d) Clustering accuracy (e) Computational time

Figure 1. Performance of SSC-OMP and SSC-BP on synthetic data. The data are drawn from 5 subspaces of dimension 6 in ambient dimension 9. Each subspace contains the same number of points and the overall number of points is varied from 150 to 10^5 and is shown in log scale. For SSC-BP, however, the maximum number of points tested is $6,000$ due to time limit. Notice that the bottom right figure also uses log scale in y-axis.

The subspace-preserving representation percentage and error are plotted in Figure 1(a) and 1(b). Observe that the probability that SSC-OMP gives a subspace-preserving solution grows as the number of data point increases. When comparing with SSC-BP, we can see that SSC-OMP is outperformed. This matches our analysis that the condition for SSC-OMP to give a subspace-preserving representation is stronger (*i.e.*, is more difficult to be satisfied).

From a subspace clustering perspective, we are more interested in how well the method performs in terms of clustering accuracy, as well as how efficient the method is in terms of running time. These results are plotted in Figure 1(d) and 1(e), together with the connectivity 1(c). We first observe that SSC-OMP does not have as good a connectivity as SSC-BP. This could be partly due to the fact that it has fewer correct connections in the first place as shown by the subspace-preserving percentage. For clustering accuracy, SSC-OMP is also outperformed by SSC-BP. This comes at no surprise as the sparse representations produced by SSC-OMP are not as subspace-preserving or as well connected as those of SSC-BP. However, we observe that as the number of data points increases, the difference in clustering accuracy also decreases, and SSC-OMP seems to achieve arbitrarily good clustering accuracy for large N. Also, it is evident from the bottom right figure that SSC-OMP is significantly faster: the time it takes to cluster $100,000$ points is about one tenth of that needed by SSC-BP for $6,000$ points. We conclude that as N increases, the difference in clustering accuracy between SSC-OMP and SSC-BP reduces, yet SSC-OMP is significantly faster, which makes SSC-OMP preferable for large-scale problems.

5.2. Clustering Images of Handwritten Digits

In this experiment, we evaluate the performance of different subspace clustering methods on clustering images of handwritten digits. We use the MNIST dataset [30], which contains grey scale images of handwritten digits $0-9$.

In each experiment, $50-600$ images for each of the 10 digits are randomly picked. For each image, we compute a set of feature vectors using a scattering convolution network [10]. The feature is a concatenation of coefficients in each layer of the network, and is translation invariant and deformation stable, making them suitable for subspace clustering. Each feature vector is of size $3,472$. The features for all images are then projected to dimension 500 using PCA. The subspace clustering techniques are then applied to the projected features. The results are reported in Table 1.

The numbers show that both SSC-OMP and SSC-BP give a much smaller subspace-preserving representation error than all other methods, with SSC-BP being better than

Table 1. Performance of subspace clustering methods on the MNIST dataset. The data consists of a randomly chosen number $N \in \{50, 100, 200, 400, 600\}$ of images for each of the 10 digits (*i.e.*, 0-9), with features extracted from a scattering network and projected to dimension 500 using PCA.

No. points	500	1000	2000	4000	6000
e%: subspace-preserving representation error					
SSC-OMP	42.13	38.73	36.20	34.22	33.22
SSC-BP	29.56	24.88	21.07	17.80	16.08
LSR	78.24	79.68	80.83	81.75	82.18
LRSC	81.33	81.99	82.67	83.15	83.27
SCC	89.89	89.87	89.85	89.81	89.81
c%: connectivity					
SSC-OMP	.113	.099	.093	.089	.078
SSC-BP	.096	.080	.064	.052	.045
LSR	.448	.463	.480	.500	.510
LRSC	.500	.512	.525	.537	.541
SCC	.952	.993	.992	.992	.991
a%: average clustering accuracy					
SSC-OMP	83.64	86.67	90.60	91.22	91.25
SSC-BP	83.01	84.06	85.58	86.00	85.60
LSR	75.84	78.42	78.09	79.06	79.91
LRSC	75.02	79.76	79.44	78.46	79.88
SCC	53.45	61.47	66.43	71.46	70.60
t(sec.): running time					
SSC-OMP	2.7	11.4	93.8	410.4	760.9
SSC-BP	20.1	97.9	635.2	4533	13605
LSR	1.7	5.9	42.4	136.1	327.6
LRSC	1.9	6.4	43.0	145.6	312.9
SCC	31.2	48.5	101.3	235.2	366.8

Table 2. Performance of subspace clustering methods on EYaleB dataset. 'NA' denotes running error in the process of the code. The data consists of face images under 64 different illumination conditions of a randomly picked $L = \{2, 10, 20, 30, 38\}$ individuals. Images are downsampled to size 48×42 as features.

No. subjects	2	10	20	30	38
e%: subspace-preserving representation error					
SSC-OMP	4.14	13.62	16.80	18.66	20.13
SSC-BP	2.70	10.33	12.67	13.74	14.64
LSR	22.77	67.07	79.52	84.94	87.57
LRSC	26.87	69.76	80.58	85.56	88.02
SCC	48.70	NA	NA	96.57	97.25
c%: connectivity					
SSC-OMP	.080	.037	.019	.003	.000
SSC-BP	.073	.048	.041	.038	.035
LSR	.454	.387	.355	.334	.327
LRSC	.476	.427	.383	.345	.320
SCC	.962	NA	NA	.543	.463
a%: average clustering accuracy					
SSC-OMP	99.18	86.09	81.55	78.27	77.59
SSC-BP	99.45	91.85	79.80	76.10	68.97
LSR	96.77	62.89	67.17	67.79	63.96
LRSC	94.32	66.98	66.34	67.49	66.78
SCC	78.91	NA	NA	14.15	12.80
t(sec.): running time					
SSC-OMP	0.6	8.3	31.1	63.7	108.6
SSC-BP	49.1	228.2	554.6	1240	1851
LSR	0.1	0.8	3.1	8.3	15.9
LRSC	1.1	1.9	6.3	14.8	26.5
SCC	50.0	NA	NA	520.3	750.7

SSC-OMP. This is consistent with our theoretical analysis as there is no guarantee that LSR or LRSC give a subspace-preserving representation for non-independent subspaces, and SSC-BP has a higher probability of giving a subspace-preserving representation than SSC-OMP.

In terms of clustering accuracy, SSC-OMP is better than SSC-BP, which in turn outperforms LSR and LRSC. The good performance of SSC-OMP can be explained by observing the subspace-preserving representation error and the connectivity measure as a trade-off: SSC-BP is subspace-preserving but the affinity graph is not well connected, while LSR, LRSC and SCC have densely connected but not subspace-preserving affinities. In fact, for OMP we use $k_{\max} = 10$, which gives 10 connections, and SSC-BP might be sparser thus less connected, giving rise to an inferior performance compared with SSC-OMP. Finally, SCC does not work as well as other methods.

Considering the running time of the methods, SSC-BP requires much more computations, especially when the number of points is large. Though SSC-OMP is an iterative method, its computation time is about twice that of LSR and LRSC, which have closed form solutions. This again qualifies the proposed method for large scale problems.

5.3. Clustering Face Images with Varying Lighting

In this experiment, we evaluate the performance of different subspace clustering methods on the Extended Yale B dataset [25], which contains frontal face images of 38 individuals under 64 different illumination conditions, each of the size 192×168. In this case, the data points are the original face images downsampled to 48×42 pixels. In each experiment, we randomly pick $n \in \{2, 10, 20, 30, 38\}$ individuals and take all the images (under different illuminations) of them as the data to be clustered.

The clustering performance of different methods is reported in Table 2. In terms of subspace-preserving recovery, we can observe a slightly better performance of SSC-BP over SSC-OMP in all cases. The other three methods have very large subspace-preserving representation errors especially when the number of subjects is $n \geq 10$. In terms of clustering accuracy, all methods do fairly well when the number of clusters is 2 except for SCC, which is far worse than others. As the number of subjects increases from 10 to 38, LSR and LRSC can only maintain an accuracy of about

60% and SCC is even worse, but SSC-OMP and SSC-BP maintain a reasonably good performance, although the accuracy also degrades gradually. We can see that SSC-BP performs slightly better when the number of subjects is 2 or 10, but SSC-OMP performs better when $n > 10$.

6. Conclusion

We proposed a new sparse subspace clustering algorithm called SSC-OMP. We derived theoretical conditions under which SSC-OMP is guaranteed to give the correct clustering. Our conditions are broader than those of state of the art methods based on ℓ_2 or nuclear norm regularization, and slightly weaker than those of SSC-BP. Experiments on synthetic and real world datasets showed that SSC-OMP is much more accurate than state of the art methods based on ℓ_2 or nuclear norm regularization and about twice as slow. On the other hand, SSC-OMP is less accurate than SSC-BP, but orders of magnitude faster. Moreover, our experiments demonstrated for the first time subspace clustering experiments on $100,000$ points, while state of the art methods had only been tested on at most $10,000$ data points. Overall, SSC-OMP provides the best trade-off of accuracy versus computation for large scale subspace clustering problems.

Appendices

In the appendices, we provide proofs for the theoretical results in the paper. We also provide supplementary experimental results for the synthetic experiment and the face clustering experiment.

A. Proof of Theorem 1

In Theorem 1 at the end of section 3.1, we claim that the SSC-OMP gives subspace preserving representations if subspaces are independent. Here we provide the proof.

Theorem. *If the subspaces are independent, OMP gives a subspace-preserving representation of every data point.*

Proof. Consider a data point $\boldsymbol{x}_j \in S_i$. We need to show that the output of OMP$(X_{-j}, \boldsymbol{x}_j)$ is subspace-preserving. As an assumption, the termination parameters in OMP are set to be $\epsilon = 0$ and $k_{\max} = N - 1$ (*i.e.*, the total number of points in the dictionary X_{-j}).

Assume the OMP algorithm terminates at iteration $k := k^*$. It would have the two cases of either $\boldsymbol{q}_{k^*} = \mathbf{0}$ or $k^* = N - 1$. Denote $T^* := T_{k^*}$. If $\boldsymbol{q}_{k^*} = \mathbf{0}$, then $P_{T^*} \cdot \boldsymbol{x}_j = \boldsymbol{x}_j$ by line 4 of Algorithm 1, so $\boldsymbol{x}_j$ is in the range of matrix X_{T^*} which denotes the columns of X_{-j} indexed by T^*. If $k^* = N - 1$, then the support set T^* indexes all the data points in the dictionary X_{-j}, including the points X^i_{-j}. In this case, since $\boldsymbol{x}_j \in S_i$ is also in the range of X^i_{-j}, it has that $\boldsymbol{x}_j$ is in the range of X_{T^*}. Thus, in both cases, $\boldsymbol{x}_j$ is in the span of columns indexed by the support set T^*.

As a consequence, the final output of OMP, given by

$$\boldsymbol{c}^* = \underset{\boldsymbol{c}:\text{Supp}(\boldsymbol{c})\subseteq T^*}{\arg\min} \|\boldsymbol{x}_j - X_{-j}\boldsymbol{c}\|_2,$$

would satisfy that $\boldsymbol{x}_j = X_{-j} \cdot \boldsymbol{c}^*$. We rewrite it as

$$\boldsymbol{x}_j - \sum_{\substack{m:\boldsymbol{x}_m \in S_i \\ m \in T^*}} \boldsymbol{x}_m \cdot c^*_m = \sum_{\substack{m:\boldsymbol{x}_m \notin S_i \\ m \in T^*}} \boldsymbol{x}_m \cdot c^*_m. \tag{17}$$

Observe that the left hand side of 17 is in subspace S_i while the right hand side is in subspace $S_{-i} := \sum_{m \neq i} S_m$. By the assumption that the set of all subspaces is independent, we know S_i and S_{-i} are also independent, so they intersect only at the origin. As a consequence, we have

$$\sum_{m:\boldsymbol{x}_m \notin S_i} \boldsymbol{x}_m \cdot c^*_m = 0. \tag{18}$$

Finally, note that in the OMP algorithm, the columns of X_{-j} indexed by T_k for any k are always linearly independent. This is evident from step 4 of Algorithm 1, as the residual vector $\boldsymbol{q}_k$ is orthogonal to every column of X_{-j} indexed by T_k, thus when choosing a new entry to be added to T_k in step 3 of Algorithm 1, points that are not linearly independent with the points indexed by T_k would have zero inner product with $\boldsymbol{q}_k$, so would not be picked. Combining this fact with Equation (18), we know $c^*_m = 0$ if $\boldsymbol{x}_m \notin S_i$. Thus the solution c^* is subspace-preserving. □

B. Proof of Lemma 2

In this section, we prove Lemma 2 in Section 3.2.

Lemma. *Let $\boldsymbol{x}_j \in S_i$. Then, for all $\boldsymbol{w} \in \mathcal{W}^i_j$, we have:*

$$\max_{k:\boldsymbol{x}_k \notin S_i} |\boldsymbol{w}^\top \boldsymbol{x}_k| \leq \max_{k:k\neq i} \mu(\mathcal{W}^i, \mathcal{X}^k) \leq \max_{k:k\neq i} \mu(\mathcal{X}^i, \mathcal{X}^k)/r_i;$$
$$\max_{k:\boldsymbol{x}_k \in S_i, k\neq j} |\boldsymbol{w}^\top \boldsymbol{x}_k| \geq r(\mathcal{P}^i_{-j}) \geq r_i. \tag{19}$$

Proof. Two of the inequalities need proofs while the other two follow directly from definitions.

For the first one, we prove that $\max_{k:k\neq i} \mu(\mathcal{W}^i, \mathcal{X}^k) \leq \max_{k:k\neq i} \mu(\mathcal{X}^i, \mathcal{X}^k)/r_i$. To do this, it suffices to show that for any $k \neq i$, $\mu(\mathcal{W}^i, \mathcal{X}^k) \leq \mu(\mathcal{X}^i, \mathcal{X}^k)/r_i$. Notice that any point $\hat{\boldsymbol{w}}$ in $\mathcal{W}^i$ is in the subspace S_i, so it could be written as a linear combination of the points in $\mathcal{X}^i$, *i.e.* $\hat{\boldsymbol{w}} = X^i \cdot \boldsymbol{c}$ for some $\boldsymbol{c}$. Specifically, we pick a $\boldsymbol{c}$ that is given by the following optimization program:

$$\hat{\boldsymbol{c}} = \underset{\boldsymbol{c}}{\arg\min} \|\boldsymbol{c}\|_1 \text{ s.t. } \hat{\boldsymbol{w}} = X^i \cdot \boldsymbol{c}. \tag{20}$$

Given this, observe that we can write

$$\|X^{k\top}\hat{\boldsymbol{w}}\|_\infty = \|X^{k\top}\cdot X^i\cdot\hat{\boldsymbol{c}}\|_\infty \le \mu(\mathcal{X}^i,\mathcal{X}^k)\cdot\|\hat{\boldsymbol{c}}\|_1,$$

in which we use the fact that $|\langle \boldsymbol{x},\boldsymbol{y}\rangle| \le \|\boldsymbol{x}\|_\infty\|\boldsymbol{y}\|_1, \forall \boldsymbol{x},\boldsymbol{y}$.

To proceed, we need to provide a bound on $\|\hat{\boldsymbol{c}}\|_1$. As $\hat{\boldsymbol{c}}$ is defined by (20), it is shown that such a bound exists and is given by (see, e.g. lemma B.2 in [42])

$$\|\hat{\boldsymbol{c}}\|_1 \le \|\hat{\boldsymbol{w}}\|_2/r(\mathcal{P}^i) = 1/r(\mathcal{P}^i),$$

where $\mathcal{P}^i := \text{conv}(\pm\mathcal{X}^i)$, and we use the fact that every point in the set of residual directions $\mathcal{W}^i$ is defined to have unit norm. Now, by the definitions we can get $r(\mathcal{P}^i) \ge r(\mathcal{P}^i_{-j}) \ge r_i$, thus $\|\hat{\boldsymbol{c}}\|_1 \le 1/r(\mathcal{P}^i) \le 1/r_i$, which gives

$$\|X^{k\top}\hat{\boldsymbol{w}}\|_\infty \le \mu(\mathcal{X}^k,\mathcal{X}^i)/r_i. \tag{21}$$

Finally, since (21) holds for any $\hat{\boldsymbol{w}}$ in $\mathcal{W}^i$, the conclusion follows that $\mu(\mathcal{W}^i,\mathcal{X}^k) \le \mu(\mathcal{X}^i,\mathcal{X}^k)/r_i$.

For a second part, we prove that for all $\boldsymbol{w} \in \mathcal{W}^i_j$, $\max_{\{k:\boldsymbol{x}_k\in S_i, k\neq j\}} |\boldsymbol{w}^\top\boldsymbol{x}_k| \ge r(\mathcal{P}^i_{-j})$, or equivalently, $\|X^{i\top}_{-j}\cdot\boldsymbol{w}\|_\infty \ge r(\mathcal{P}^i_{-j})$. The proof relies on the result (see definition 7.2 in [41]) that for an arbitrary vector $\boldsymbol{y}\in S_i$,

$$\|X^{i\top}_{-j}\cdot\boldsymbol{y}\|_\infty \le 1 \Rightarrow \|\boldsymbol{y}\|_2 \le 1/r(\mathcal{P}^i_{-j}).$$

It then follows that if (by contradiction) $\|X^{i\top}_{-j}\cdot\boldsymbol{w}\|_\infty < r(\mathcal{P}^i_{-j})$, then $\|X^{i\top}_{-j}\cdot\boldsymbol{w}\|_\infty = r(\mathcal{P}^i_{-j}) - \epsilon > 0$ for some $\epsilon > 0$, and

$$\begin{aligned}
&\|X^{i\top}_{-j}\frac{\boldsymbol{w}}{r(\mathcal{P}^i_{-j})-\epsilon}\|_\infty = 1 \le 1\\
\Rightarrow&\|\frac{\boldsymbol{w}}{r(\mathcal{P}^i_{-j})-\epsilon}\|_2 \le 1/r(\mathcal{P}^i_{-j})\\
\Rightarrow&\|\boldsymbol{w}\|_2 \le (r(\mathcal{P}^i_{-j})-\epsilon)/r(\mathcal{P}^i_{-j}) < 1,
\end{aligned}$$

which is a contradiction of the fact that $\boldsymbol{w}$ is normalized. □

C. Proof of Theorem 3

Theorem. *Assume a random model where all subspaces are of equal dimension d and the number of data points in each subspace is $\rho d+1$, where $\rho > 1$ is the "density", so the total number data points in all subspaces is $N = n(\rho d+1)$. The output of OMP is subspace preserving with probability $p > 1 - \frac{2d}{N} - Ne^{-\sqrt{\rho}d}$ if*

$$d < \frac{c^2(\rho)\log\rho}{12}\frac{D}{\log N}, \tag{22}$$

where $c(\rho) > 0$ is a constant that depends only on ρ.

Proof. The proof goes by providing bounds for the left and right hand side of the inequality in Theorem 2, copied here for convenience of reference:

$$\forall i = 1,\dots,n, \quad \max_{k:k\neq i}\mu(\mathcal{W}^i,\mathcal{X}^k) < r_i. \tag{23}$$

We first give a bound on the inradius r_i. Denote

$$\bar{r} = \frac{c(\rho)\sqrt{\log\rho}}{\sqrt{2d}}, \bar{p}_r = Ne^{-\sqrt{\rho}d},$$

in which $c(\rho)$ is a numerical constant depending on ρ. [41] shows that as points in each subspace are independently distributed, it has

$$P(r_i \ge \bar{r} \text{ for all } i) \ge 1 - \bar{p}_r,$$

Now we give a bound on the coherence. From an upper bound on the area of spherical cap [4, 41], we have that if $\boldsymbol{x},\boldsymbol{y}\in\mathbb{R}^D$ are two random vectors that are distributed uniformly and independently on the unit sphere, then

$$P\{|\langle\boldsymbol{x},\boldsymbol{y}\rangle| \ge \sqrt{\frac{6\log N}{D}}\} \le \frac{2}{N^3}. \tag{24}$$

Under the random model, points $\boldsymbol{x}\in\mathcal{X}^k, \forall k$ are distributed uniformly at random on the unit sphere of $\mathbb{R}^D$ by assumption. Any residual point $\boldsymbol{w}\in\mathcal{W}^i, \forall i$ also has uniform distribution on the unit sphere as it depends only on points in $\mathcal{X}^i$, which are independent and uniformly distributed. Furthermore, any pair of points $\boldsymbol{x}\in\mathcal{X}^k$ and $\boldsymbol{w}\in\mathcal{W}^i$ are distributed independently because points in $\mathcal{X}^k$ and $\mathcal{X}^i$ are independent. Thus the result of Equation (24) is applicable here. Since there are at most $d\times N^2$ pairs of inner product in $\mu(\mathcal{W}^i,\mathcal{X}^k)$, by using union bound we can get

$$P(\mu(\mathcal{W}^i,\mathcal{X}^k) \le \bar{\mu}, \text{ for all } i,k) \ge 1 - \bar{p}_\mu dN^2,$$

in which we define

$$\bar{\mu} = \sqrt{\frac{6\log N}{D}}, \bar{p}_\mu = 2/N^3,$$

As a consequence, if the condition (22) holds then we have $\bar{r} < \bar{\mu}$. Applying again the union bound we get that condition (23) holds with probability $p > 1 - \bar{p}_\mu d - \bar{p}_r$. This finishes the proof. □

D. Comparison with prior work on SSC-OMP

In Theorem 2 we give a sufficient condition for guaranteeing subspace-preserving of the SSC-OMP:

$$\forall i = 1,\dots,n, \quad \max_{k:k\neq i}\mu(\mathcal{W}^i,\mathcal{X}^k) < r_i, \tag{25}$$

and in Corollary 1 a stronger sufficient condition:

$$\forall i = 1, \dots, n, \quad \max_{k:k\neq i} \mu(\mathcal{X}^i, \mathcal{X}^k) < r_i^2. \tag{26}$$

Prior to this work, [17] gives another sufficient condition for SSC-OMP giving subspace-preserving representation, namely,

$$\max_{k:k\neq i} \mu(\mathcal{X}^i, \mathcal{X}^k) < r_i - \frac{2\sqrt{1-(r_i)^2}}{\sqrt[4]{12}} \max_{k:k\neq i} \cos\theta^*_{i,k}, \tag{27}$$

in which the subspace angle is defined as

$$\theta^*_{i,k} = \min_{\substack{\boldsymbol{x}\in S_i \\ \|\boldsymbol{x}\|_2=1}} \min_{\substack{\boldsymbol{y}\in S_k \\ \|\boldsymbol{y}\|_2=1}} \arccos\langle \boldsymbol{x}, \boldsymbol{y}\rangle. \tag{28}$$

We claim that Theorem 2 in this work is a stronger result than that provided in the work [17], as the sufficient condition of Equation (27) implies Equation (25). Here we give a rigorous argument for this claim.

Notice that the inequality in (27) implies that $\forall k \neq i$,

$$\mu(\mathcal{X}^i, \mathcal{X}^k) < r_i - \sqrt{2-2r_i}\cos\theta^*_{i,k}, \tag{29}$$

see Lemma 1 in their paper. We show that condition (29) implies (25) when $r_i \leq 1/2$, and implies condition (26) when $r_i > 1/2$, which concludes that the their result is weaker than our result in (25).

Case 1. If $r_i \leq 1/2$, then $\sqrt{2-2r_i} \geq 1$, thus

$$\begin{aligned}(29) &\Rightarrow \mu(\mathcal{X}^i, \mathcal{X}^k) < r_i - \cos\theta^*_{i,k} \Rightarrow \cos\theta^*_{l,k} < r_i \\ &\Rightarrow \mu(\mathcal{X}^k, \mathcal{W}^i) < r_i \Leftrightarrow (25).\end{aligned}$$

Case 2. If $r_i > 1/2$, then

$$\begin{aligned}(29) &\Rightarrow \mu(\mathcal{X}^i, \mathcal{X}^k) < r_i - \sqrt{2-2r_i}\mu(\mathcal{X}^i, \mathcal{X}^k) \\ &\Rightarrow \mu(\mathcal{X}^i, \mathcal{X}^k) < r_i/(1+\sqrt{2-2r_i}) \\ &\Rightarrow \mu(\mathcal{X}^i, \mathcal{X}^k) < r_i/(1+(2-2r_i)) \\ &\Rightarrow \mu(\mathcal{X}^i, \mathcal{X}^k) < (r_i)^2 \Leftrightarrow (26) \Rightarrow (25).\end{aligned}$$

So the condition in Equation (25) is implied by (27).

E. Results for synthetic experiment

We provide tables detailing the results in the synthetic experiment in Section 5.1. Table 3 shows the result of up to 6000 data points, and Table 4 shows the rest part where number of point is from 6000 to around $100,000$.

Acknowledgment

The authors would like to thank the support of NSF BIGDATA grant 1447822.

References

[1] P. Agarwal and N. Mustafa. k-means projective clustering. In *ACM Symposium on Principles of database systems*, 2004.

[2] A. Aldroubi and K. Zaringhalam. Nonlinear least squares in $\mathbb{R}^N$. *Acta Applicandae Mathematicae*, 107(1-3):325–337, 2009.

[3] C. Archambeau, N. Delannay, and M. Verleysen. Mixtures of robust probabilistic principal component analyzers. *Neurocomputing*, 71(7–9):1274–1282, 2008.

[4] K. Ball. An elementary introduction to modern convex geometry. In *in Flavors of Geometry*, pages 1–58. Univ. Press, 1997.

[5] R. Basri and D. Jacobs. Lambertian reflection and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):218–233, 2003.

[6] T. Boult and L. Brown. Factorization-based segmentation of motions. In *IEEE Workshop on Motion Understanding*, pages 179–186, 1991.

[7] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010.

[8] P. S. Bradley and O. L. Mangasarian. k-plane clustering. *Journal of Global Optimization*, 16(1):23–32, 2000.

[9] A. Bruckstein, D. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1):34–81, Feb. 2009.

[10] J. Bruna and S. Mallat. Invariant scattering convolution networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1872–1886, 2013.

[11] E. Candès and M. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, Mar. 2008.

[12] G. Chen and G. Lerman. Spectral curvature clustering (SCC). *International Journal of Computer Vision*, 81(3):317–330, 2009.

[13] F. Chung. *Spectral graph theory*. Washington: Conference Board of the Mathematical Sciences, 1997.

[14] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998.

[15] M. A. Davenport and M. B. Wakin. Analysis of orthogonal matching pursuit using the restricted isometry property. *IEEE Transactions on Information Theory*, 56(9):4395–4401, 2010.

[16] H. Derksen. Hilbert series of subspace arrangements. *Journal of Pure and Applied Algebra*, 209(1):91–98, 2007.

[17] E. L. Dyer, A. C. Sankaranarayanan, and R. G. Baraniuk. Greedy feature selection for subspace clustering. *Journal of Machine Learning Research*, 14(1):2487–2517, 2013.

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

[19] E. Elhamifar and R. Vidal. Clustering disjoint subspaces via sparse representation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2010.

Table 3. Performance of OMP and BP based SSC on synthetic data. The data are drawn from 5 subspaces of dimension 6 in ambient dimension 9. Each subspace contains the same number of points and the overall number of points is varied from 150 to 10^5 for SSC-OMP and from 150 to 6000 for SSC-BP. See Figure 1 in the paper for the plots of the results and the discussions in section 5.1.

No. points	150	300	600	1200	2400	3600	4800	6000
p%: subspace-preserving representation percentage								
SSC-OMP	5.73	13.68	24.32	39.18	54.14	61.25	66.35	69.31
SSC-BP	7.67	19.28	35.77	53.41	68.93	75.12	79.55	81.89
e%: subspace-preserving representation error								
SSC-OMP	46.98	37.37	29.00	21.08	14.29	12.01	10.29	9.19
SSC-BP	43.73	32.01	21.90	14.22	8.55	6.66	5.34	4.64
c: connectivity								
SSC-OMP	.035	.035	.026	.019	.023	.022	.023	.011
SSC-BP	.117	.132	.131	.132	.126	.120	.114	.108
a%: average clustering accuracy								
SSC-OMP	46.27	56.37	73.80	87.25	92.47	93.87	93.00	95.25
SSC-BP	69.43	89.17	95.28	98.06	99.01	99.32	99.51	99.56
t(sec.): running time								
SSC-OMP	0.3	0.5	0.7	1.1	2.7	4.2	6.9	9.4
SSC-BP	1.6	4.5	30.8	207.9	1432	5658	14128	31305

Table 4. Performance of OMP and BP based SSC on synthetic data under random model (continuations).

No. points	9000	15000	24000	36000	51000	69000	90000	99990
p%: subspace-preserving recovery percentage								
SSC-OMP	76.54	81.79	86.66	90.04	91.97	93.12	94.16	94.32
e%: subspace-preserving representation error								
SSC-OMP	6.83	5.41	3.92	2.96	2.40	2.07	1.74	1.69
c: connectivity								
SSC-OMP	.015	.007	.008	.007	.007	.003	.004	.002
a%: average clustering accuracy								
SSC-OMP	96.40	97.15	97.89	98.34	98.63	98.81	98.97	98.98
t(sec.): running time								
SSC-OMP	20.9	51.0	120.7	259.8	538.4	1204	1850	2313

[20] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013.

[21] Z. Fan, J. Zhou, and Y. Wu. Multibody grouping by inference of multiple subspaces from high-dimensional data using oriented-frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):91–105, 2006.

[22] P. Favaro, R. Vidal, and A. Ravichandran. A closed form solution to robust subspace estimation and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[23] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czech. Math. J.*, 25:619–633, 1975.

[24] C. W. Gear. Multibody grouping from motion images. *International Journal of Computer Vision*, 29(2):133–150, 1998.

[25] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.

[26] A. Goh and R. Vidal. Segmenting motions of different types by unsupervised manifold clustering. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[27] A. Gruber and Y. Weiss. Multibody factorization with uncertainty and missing data using the em algorithm. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume I, pages 707–714, 2004.

[28] K. Huang, Y. Ma, and R. Vidal. Minimum effective dimension for mixtures of subspaces: A robust GPCA algorithm and its applications. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume II, pages 631–638, 2004.

[29] K. Kanatani. Motion segmentation by subspace separation and model selection. In *IEEE International Conference on Computer Vision*, volume 2, pages 586–591, 2001.

[30] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278 – 2324, 1998.

[31] K.-C. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, 27(5):684–698, 2005.

[32] A. Leonardis, H. Bischof, and J. Maver. Multiple eigenspaces. *Pattern Recognition*, 35(11):2613–2627, 2002.

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

[34] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *International Conference on Machine Learning*, 2010.

[35] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan. Robust and efficient subspace segmentation via least squares regression. In *Proceedings of European Conference on Computer Vision*, 2012.

[36] L. Lu and R. Vidal. Combined central and subspace clustering on computer vision applications. In *International Conference on Machine Learning*, pages 593–600, 2006.

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

[38] Y. Ma, A. Y. Yang, H. Derksen, and R. Fossum. Estimation of subspace arrangements with applications in modeling and segmenting mixed data. *SIAM Review*, 50(3):413–458, 2008.

[39] Y. Pati, R. Rezaiifar, and P. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with application to wavelet decomposition. In *Asilomar Conference on Signals, Systems and Computation*, 1993.

[40] S. Rao, R. Tron, R. Vidal, and Y. Ma. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1832–1845, 2010.

[41] M. Soltanolkotabi and E. J. Candès. A geometric analysis of subspace clustering with outliers. *Annals of Statistics*, 40(4):2195–2238, 2012.

[42] M. Soltanolkotabi, E. Elhamifar, and E. J. Candès. Robust subspace clustering. *Annals of Statistics*, 42(2):669–699, 2014.

[43] Y. Sugaya and K. Kanatani. Geometric structure of degeneracy for multi-body motion segmentation. In *Workshop on Statistical Methods in Video Processing*, 2004.

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

[45] J. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, Oct. 2004.

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

[47] R. Vidal. Subspace clustering. *IEEE Signal Processing Magazine*, 28(3):52–68, March 2011.

[48] R. Vidal and P. Favaro. Low rank subspace clustering (LRSC). *Pattern Recognition Letters*, 43:47–61, 2014.

[49] R. Vidal, Y. Ma, and J. Piazzi. A new GPCA algorithm for clustering subspaces by fitting, differentiating and dividing polynomials. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume I, pages 510–517, 2004.

[50] R. Vidal, Y. Ma, and S. Sastry. Generalized Principal Component Analysis (GPCA). In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume I, pages 621–628, 2003.

[51] R. Vidal, Y. Ma, and S. Sastry. Generalized Principal Component Analysis (GPCA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1–15, 2005.

[52] R. Vidal, R. Tron, and R. Hartley. Multiframe motion segmentation with missing data using PowerFactorization and GPCA. *International Journal of Computer Vision*, 79(1):85–105, August 2008.

[53] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17, 2007.

[54] J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *European Conference on Computer Vision*, pages 94–106, 2006.

[55] A. Y. Yang, S. Rao, and Y. Ma. Robust statistical estimation and segmentation of multiple subspaces. In *Workshop on 25 years of RANSAC*, 2006.

[56] T. Zhang, A. Szlam, and G. Lerman. Median k-flats for hybrid linear modeling with many outliers. In *Workshop on Subspace Methods*, 2009.

[57] T. Zhang, A. Szlam, Y. Wang, and G. Lerman. Hybrid linear modeling via local best-fit flats. *International Journal of Computer Vision*, 100(3):217–240, 2012.