

## ROBUST SUBSPACE CLUSTERING

BY MAHDI SOLTANOLKOTABI<sup>1</sup>, EHSAN ELHAMIFAR  
AND EMMANUEL J. CANDÈS<sup>2</sup>

*Stanford University, University of California, Berkeley and Stanford University*

Subspace clustering refers to the task of finding a multi-subspace representation that best fits a collection of points taken from a high-dimensional space. This paper introduces an algorithm inspired by sparse subspace clustering (SSC) [In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2009) 2790–2797*] to cluster noisy data, and develops some novel theory demonstrating its correctness. In particular, the theory uses ideas from geometric functional analysis to show that the algorithm can accurately recover the underlying subspaces under minimal requirements on their orientation, and on the number of samples per subspace. Synthetic as well as real data experiments complement our theoretical study, illustrating our approach and demonstrating its effectiveness.

### 1. Introduction.

1.1. *Motivation.* In many problems across science and engineering, a fundamental step is to find a lower-dimensional subspace which best fits a collection of points taken from a high-dimensional space; this is classically achieved via Principal Component Analysis (PCA). Such a procedure makes perfect sense as long as the data points are distributed around a lower-dimensional subspace, or expressed differently, as long as the data matrix with points as column vectors has approximately low rank. A more general model might sometimes be useful when the data come from a mixture model in which points do not lie around a single lower-dimensional subspace but rather around a union of low-dimensional subspaces. For instance, consider an experiment in which gene expression data are gathered on many cancer cell lines with unknown subsets belonging to different tumor types. One can imagine that the expressions from each cancer type may span a distinct lower-dimensional subspace. If the cancer labels were known in advance, one would apply PCA separately to each group but we here consider

---

Received February 2013; revised December 2013.

<sup>1</sup>Supported by a Benchmark Stanford Graduate Fellowship.

<sup>2</sup>Supported in part by AFOSR under Grant FA9550-09-1-0643 and by ONR under Grant N00014-09-1-0258 and by a gift from the Broadcom Foundation.

*MSC2010 subject classifications.* 62-07.

*Key words and phrases.* Subspace clustering, spectral clustering, LASSO, Dantzig selector,  $\ell_1$  minimization, multiple hypothesis testing, true and false discoveries, geometric functional analysis, nonasymptotic random matrix theory.

the case where the observations are unlabeled. Thus, the goal in such an example would be to separate gene expression patterns into different cancer types if possible. Finding the components of the mixture and assigning each point to a fitted subspace is called subspace clustering. Even when the mixture model holds, the full data matrix may not have low rank at all, a situation which is very different from that where PCA is applicable.

In recent years, numerous algorithms have been developed for subspace clustering and applied to various problems in computer vision/machine learning [53] and data mining [43]. At the time of this writing, subspace clustering techniques are certainly gaining momentum as they begin to be used in fields as diverse as identification and classification of diseases [37], network topology inference [23], security and privacy in recommender systems [56], system identification [6], hyperspectral imaging [18], identification of switched linear systems [35, 41], and music analysis [29] to name just a few. In spite of all these interesting works, tractable subspace clustering algorithms either lack a theoretical justification, or are guaranteed to work under restrictive conditions rarely met in practice. (We note that although novel and often efficient clustering techniques come about all the time, establishing rigorous theory for such techniques has proven to be quite difficult. In the context of subspace clustering, Section 5 offers a partial survey of the existing literature.) Furthermore, proposed algorithms are not always computationally tractable. Thus, one important issue is whether tractable algorithms that can (provably) work in less than ideal situations—that is, under severe noise conditions and relatively few samples per subspace—exist.

Elhamifar and Vidal [20] have introduced an approach to subspace clustering, which relies on ideas from the sparsity and compressed sensing literature, please see also the longer version [22] which was submitted while this manuscript was under preparation. *Sparse subspace clustering* (SSC) [20, 22] is computationally efficient since it amounts to solving a sequence of  $\ell_1$  minimization problems and is, therefore, tractable. Now the methodology in [20] is mainly geared toward noiseless situations where the points lie exactly on lower-dimensional planes, and theoretical performance guarantees in such circumstances are given under restrictive assumptions. Continuing on this line of work, [46] showed that good theoretical performance could be achieved under broad circumstances. However, the model supporting the theory in [46] is still noise free.

This paper considers the subspace clustering problem in the presence of noise. We introduce a tractable clustering algorithm, which is a natural extension of SSC, and develop rigorous theory about its performance; see the results from Section 3.1. In a nutshell, we propose a statistical mixture model to represent data lying near a union of subspaces, and prove that in this model, the algorithm is effective in separating points from different subspaces as long as there are sufficiently many samples from each subspace and that the subspaces are not too close to each other. In this theory, the performance of the algorithm is explained in terms

of interpretable and intuitive parameters such as (1) the values of the principal angles between subspaces, (2) the number of points per subspace, (3) the noise level and so on. In terms of these parameters, our theoretical results indicate that the performance of the algorithm is in some sense near the limit of what can be achieved by any algorithm, regardless of tractability.

1.2. *Problem formulation and model.* We assume we are given data points lying near a union of unknown linear subspaces; there are  $L$  subspaces  $S_1, S_2, \dots, S_L$  of  $\mathbb{R}^n$  of dimensions  $d_1, d_2, \dots, d_L$ . These together with their number are completely unknown to us. We are given a point set  $\mathcal{Y} \subset \mathbb{R}^n$  of cardinality  $N$ , which may be partitioned as  $\mathcal{Y} = \mathcal{Y}_1 \cup \mathcal{Y}_2 \cup \dots \cup \mathcal{Y}_L$ ; for each  $\ell \in \{1, \dots, L\}$ ,  $\mathcal{Y}_\ell$  is a collection of  $N_\ell$  vectors that are “close” to subspace  $S_\ell$ . The goal is to approximate the underlying subspaces using the point set  $\mathcal{Y}$ . One approach is first to assign each data point to a cluster, and then estimate the subspaces representing each of the groups with PCA.

Our statistical model assumes that each point  $\mathbf{y} \in \mathcal{Y}$  is of the form

$$(1.1) \quad \mathbf{y} = \mathbf{x} + \mathbf{z},$$

where  $\mathbf{x}$  belongs to one of the subspaces and  $\mathbf{z}$  is an independent stochastic noise term. We suppose that the inverse signal-to-noise ratio (SNR) defined as  $\mathbb{E} \|\mathbf{z}\|_2^2 / \|\mathbf{x}\|_{\ell_2}^2$  is bounded above. Each observation is thus the superposition of a noiseless sample taken from one of the subspaces and of a stochastic perturbation whose Euclidean norm is about  $\sigma$  times the signal strength so that  $\mathbb{E} \|\mathbf{z}\|_{\ell_2}^2 = \sigma^2 \|\mathbf{x}\|_{\ell_2}^2$ . All the way through, we assume that

$$(1.2) \quad \sigma < \sigma^* \quad \text{and} \quad \max_{\ell} d_{\ell} < c_0 \frac{n}{(\log N)^2},$$

where  $\sigma^* < 1$  and  $c_0$  are fixed numerical constants. To remove any ambiguity,  $\sigma$  is the noise level and  $\sigma^*$  the maximum value it can take on. The second assumption is here to avoid unnecessarily complicated expressions later on. While more substantial, the first is not too restrictive since it just says that the signal  $\mathbf{x}$  and the noise  $\mathbf{z}$  may have about the same magnitude. (With an arbitrary perturbation of Euclidean norm equal to two, one can move from any point  $\mathbf{x}$  on the unit sphere to just about any other point.)

This is arguably the simplest model providing a good starting point for a theoretical investigation. For the noiseless samples  $\mathbf{x}$ , we consider the intuitive *semi-random model* introduced in [46], which assumes that the subspaces are fixed with points distributed uniformly at random on each subspace. One can think of this as a mixture model where each component in the mixture is a lower-dimensional subspace. (One can extend the methods to affine subspace clustering as briefly explained in Section 2.)

1.3. *What makes clustering hard?* Two important parameters fundamentally affect the performance of subspace clustering algorithms: (1) the distance between subspaces and (2) the number of samples on each subspace.

1.3.1. *Distance/affinity between subspaces.* Intuitively, any subspace clustering algorithm operating on noisy data will have difficulty segmenting observations when the subspaces are close to each other. We of course need to quantify closeness, and Definition 1.2 captures a notion of distance or similarity/affinity between subspaces.

DEFINITION 1.1. The principal angles  $\theta^{(1)}, \dots, \theta^{(d \wedge d')}$  between two subspaces  $S$  and  $S'$  of dimensions  $d$  and  $d'$ , are recursively defined by

$$\cos(\theta^{(i)}) = \max_{\mathbf{u}_i \in S} \max_{\mathbf{v}_i \in S'} \frac{\mathbf{u}_i^T \mathbf{v}_i}{\|\mathbf{u}_i\|_{\ell_2} \|\mathbf{v}_i\|_{\ell_2}}$$

with the orthogonality constraints  $\mathbf{u}_i^T \mathbf{u}_j = 0, \mathbf{v}_i^T \mathbf{v}_j = 0, j = 1, \dots, i - 1$ .

Alternatively, if the columns of  $\mathbf{U}$  and  $\mathbf{V}$  are orthobases for  $S$  and  $S'$ , then the cosine of the principal angles are the singular values of  $\mathbf{U}^T \mathbf{V}$ .

DEFINITION 1.2. The normalized affinity between two subspaces is defined by

$$\text{aff}(S, S') = \sqrt{\frac{\cos^2 \theta^{(1)} + \dots + \cos^2 \theta^{(d \wedge d')}}{d \wedge d'}}.$$

The affinity is a measure of correlation between subspaces. It is low when the principal angles are nearly right angles (it vanishes when the two subspaces are orthogonal) and high when the principal angles are small (it takes on its maximum value equal to one when one subspace is contained in the other). Hence, when the affinity is high, clustering is hard whereas it becomes easier as the affinity decreases. Ideally, we would like our algorithm to be able to handle higher affinity values—as close as possible to the maximum possible value.

There is a statistical description of the affinity which goes as follows: sample independently two unit-normed vectors  $\mathbf{x}$  and  $\mathbf{y}$  uniformly at random from  $S$  and  $S'$ . Then

$$\mathbb{E}\{(\mathbf{x}^T \mathbf{y})^2\} \propto \{\text{aff}(S, S')\}^2,$$

where the constant of proportionality is  $d \vee d'$ . Having said this, there are of course other ways of measuring the affinity between subspaces; for instance, by taking the cosine of the first principal angle. We prefer the definition above as it offers the flexibility of allowing for some principal angles to be small or zero. As an example, suppose we have a pair of subspaces with a nontrivial intersection. Then  $|\cos \theta^{(1)}| = 1$  regardless of the dimension of the intersection whereas the value of the affinity would depend upon this dimension.

1.3.2. *Sampling density.* Another important factor affecting the performance of subspace clustering algorithms has to do with the distribution of points on each subspace. In the model we study here, this essentially reduces to the number of points that lie on each subspace.<sup>3</sup>

DEFINITION 1.3. The sampling density  $\rho$  of a subspace is defined as the number of samples on that subspace per dimension. In our multi-subspace model, the density of  $S_\ell$  is, therefore,  $\rho_\ell = N_\ell/d_\ell$ .<sup>4</sup>

One expects the clustering problem to become easier as the sampling density increases. Obviously, if the sampling density of a subspace  $S$  is smaller than one, then any algorithm will fail in identifying that subspace correctly as there are not sufficiently many points to identify all the directions spanned by  $S$ . Hence, we would like a clustering algorithm to be able to operate at values of the sampling density as low as possible, that is, as close to one as possible.

**2. Robust subspace clustering: Methods and concepts.** This section introduces our methodology through heuristic arguments confirmed by numerical experiments while proven theoretical guarantees about the first step of algorithm follow in Section 3. From now on, we arrange the  $N$  observed data points as columns of a matrix  $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N] \in \mathbb{R}^{n \times N}$ . With obvious notation,  $\mathbf{Y} = \mathbf{X} + \mathbf{Z}$ .

2.1. *The normalized model.* In practice, one may want to normalize the columns of the data matrix so that for all  $i$ ,  $\|\mathbf{y}_i\|_{\ell_2} = 1$  [R-code snippet for renormalizing a data point  $y$  is:  $y \leftarrow y / \text{sqrt}(\text{sum}(y \wedge 2))$ ]. Since with our SNR assumption, we have  $\|\mathbf{y}\|_{\ell_2} \approx \|\mathbf{x}\|_{\ell_2} \sqrt{1 + \sigma^2}$  before normalization, then after normalization:

$$\mathbf{y} \approx \frac{1}{\sqrt{1 + \sigma^2}}(\mathbf{x} + \mathbf{z}),$$

where  $\mathbf{x}$  is unit-normed, and  $\mathbf{z}$  has i.i.d. random Gaussian entries with variance  $\sigma^2/n$ .

For ease of presentation, we work—in this section and in the proofs—with a model  $\mathbf{y} = \mathbf{x} + \mathbf{z}$  in which  $\|\mathbf{x}\|_{\ell_2} = 1$  instead of  $\|\mathbf{y}\|_{\ell_2} = 1$  (the numerical Section 6 is the exception). The normalized model with  $\|\mathbf{x}\|_{\ell_2} = 1$  and  $\mathbf{z}$  i.i.d.  $\mathcal{N}(0, \sigma^2/n)$  is nearly the same as before. In particular, all of our methods and theoretical results in Section 3 hold with both models in which either  $\|\mathbf{x}\|_{\ell_2} = 1$  or  $\|\mathbf{y}\|_{\ell_2} = 1$ .

<sup>3</sup>In a general deterministic model, where the points have arbitrary orientations on each subspace, we can imagine that the clustering problem becomes harder as the points align along an even lower-dimensional structure.

<sup>4</sup>Throughout, we take  $\rho_\ell \leq e^{d_\ell/2}$ . Our results hold for all other values by substituting  $\rho_\ell$  with  $\rho_\ell \wedge e^{d_\ell/2}$  in all the expressions.

2.2. *The SSC scheme.* We describe the approach in [20], which follows a three-step procedure:

- I. Compute a similarity<sup>5</sup> matrix  $\mathbf{W}$  encoding similarities between sample pairs as to construct a weighted graph  $\mathcal{G}$ .
- II. Construct clusters by applying spectral clustering techniques (e.g., [40]) to  $\mathcal{G}$ .
- III. Apply PCA to each of the clusters.

The novelty in [20] concerns step I, the construction of the affinity matrix. Interestingly, similar ideas were introduced earlier in the statistics literature for the purpose of graphical model selection [38]. Now the work [20] of interest here is mainly concerned with the noiseless situation in which  $\mathbf{Y} = \mathbf{X}$  and the idea is then to express each column  $\mathbf{x}_i$  of  $\mathbf{X}$  as a sparse linear combination of all the other columns. The reason is that under any reasonable condition, one expects that the *sparsest* representation of  $\mathbf{x}_i$  would only select vectors from the subspace in which  $\mathbf{x}_i$  happens to lie in. Applying the  $\ell_1$  norm as the convex surrogate of sparsity leads to the following sequence of optimization problems:

$$(2.1) \quad \min_{\boldsymbol{\beta} \in \mathbb{R}^N} \|\boldsymbol{\beta}\|_{\ell_1} \quad \text{subject to } \mathbf{x}_i = \mathbf{X}\boldsymbol{\beta} \quad \text{and} \quad \beta_i = 0.$$

Here,  $\beta_i$  denotes the  $i$ th element of  $\boldsymbol{\beta}$  and the constraint  $\beta_i = 0$  removes the trivial solution that decomposes a point as a linear combination of itself. Collecting the outcome of these  $N$  optimization problems as columns of a matrix  $\mathbf{B}$ , [20] sets the  $N \times N$  similarity matrix  $\mathbf{W}$  to be  $W_{ij} = |B_{ij}| + |B_{ji}|$ . [This algorithm clusters linear subspaces but can also cluster affine subspaces by adding the constraint  $\boldsymbol{\beta}^T \mathbf{1} = 1$  to (2.1).]

The issue here is that we only have access to the noisy data  $\mathbf{Y}$ ; that is, we do not see the matrix  $\mathbf{X}$  of covariates but rather a corrupted version  $\mathbf{Y}$ . This makes the problem challenging, as unlike conventional sparse recovery problems where only the response vector  $\mathbf{x}_i$  is corrupted, here both the covariates (columns of  $\mathbf{X}$ ) and the response vector are corrupted. In particular, it may not be advisable to use (2.1) with  $\mathbf{y}_i$  and  $\mathbf{Y}$  in place of  $\mathbf{x}_i$  and  $\mathbf{X}$  as, strictly speaking, sparse representations no longer exist. Observe that the expression  $\mathbf{x}_i = \mathbf{X}\boldsymbol{\beta}$  can be rewritten as  $\mathbf{y}_i = \mathbf{Y}\boldsymbol{\beta} + (\mathbf{z}_i - \mathbf{Z}\boldsymbol{\beta})$ . Viewing  $(\mathbf{z}_i - \mathbf{Z}\boldsymbol{\beta})$  as a perturbation, it is natural to use ideas from sparse regression to obtain an estimate  $\hat{\boldsymbol{\beta}}$ , which is then used to construct the similarity matrix. In this paper, we follow the same three-step procedure and shall focus on the first step in Algorithm 1; that is, on the construction of reliable similarity measures between pairs of points. Since we have noisy data, we shall not use (2.1) here. Also, we add denoising to step III, check the output of Algorithm 1. We would like to emphasize early on that the theoretical analysis provided in this

---

<sup>5</sup>We use the terminology similarity graph or matrix instead of affinity matrix as not to overload the word “affinity.”

---

**Algorithm 1** Robust SSC procedure

---

**Input:** A data set  $\mathcal{Y}$  arranged as columns of  $\mathbf{Y} \in \mathbb{R}^{n \times N}$ .

1. For each  $i \in \{1, \dots, N\}$ , produce a sparse coefficient sequence  $\{\hat{\beta}_i\}$  by regressing the  $i$ th vector  $\mathbf{y}_i$  onto the other columns of  $\mathbf{Y}$ . Collect these as columns of a matrix  $\mathbf{B}$ .
2. Form the similarity graph  $\mathcal{G}$  with nodes representing the  $N$  data points and edge weights given by  $W_{ij} = |B_{ij}| + |B_{ji}|$ .
3. Sort the eigenvalues  $\delta_1 \geq \delta_2 \geq \dots \geq \delta_N$  of the normalized Laplacian of  $\mathcal{G}$  in descending order, and set

$$\hat{L} = N - \arg \max_{i=1, \dots, N-1} (\delta_i - \delta_{i+1}).$$

4. Apply a spectral clustering technique to the similarity graph using  $\hat{L}$  as the estimated number of clusters to obtain the partition  $\mathcal{Y}_1, \dots, \mathcal{Y}_{\hat{L}}$ .
5. Use PCA to find the best subspace fits ( $\{S_\ell\}_1^{\hat{L}}$ ) to each of the partitions ( $\{\mathcal{Y}_\ell\}_1^{\hat{L}}$ ) and denoise  $\mathbf{Y}$  as to obtain clean data points  $\hat{\mathbf{X}}$ .

**Output:** Subspaces  $\{S_\ell\}_1^{\hat{L}}$  and cleaned data points  $\hat{\mathbf{X}}$ .

---

paper only concerns the first step—the sparse regression part—of the algorithm. We do not provide any guarantees for the spectral clustering step.

2.3. *Performance metrics for similarity measures.* Given the general structure of the method, we are interested in sparse regression techniques, which tend to select points in the same clusters (share the same underlying subspace) over those that do not share this property. Expressed differently, the hope is that whenever  $B_{ij} \neq 0$ ,  $\mathbf{y}_i$  and  $\mathbf{y}_j$  originate from the same subspace. We introduce metrics to quantify performance.

DEFINITION 2.1 (False discoveries). Fix  $i$  and  $j \in \{1, \dots, N\}$  and let  $\mathbf{B}$  be the outcome of step 1 in Algorithm 1. Then we say that  $(i, j)$  obeying  $B_{ij} \neq 0$  is a false discovery if  $\mathbf{y}_i$  and  $\mathbf{y}_j$  do not originate from the same subspace.

DEFINITION 2.2 (True discoveries). In the same situation,  $(i, j)$  obeying  $B_{ij} \neq 0$  is a true discovery if  $\mathbf{y}_j$  and  $\mathbf{y}_i$  originate from the same cluster/subspace.

When there are no false discoveries, we shall say that the *subspace detection property* holds. In this case, the matrix  $\mathbf{B}$  is block diagonal after applying a permutation which makes sure that columns in the same subspace are contiguous. In some cases, the sparse regression method may select vectors from other subspaces and this property will not hold. However, it might still be possible to detect and construct reliable clusters by applying steps 2–5 in Algorithm 1.

2.4. *LASSO with data-driven regularization.* A natural sparse regression strategy is the LASSO:

$$(2.2) \quad \min_{\beta \in \mathbb{R}^N} \frac{1}{2} \|\mathbf{y}_i - \mathbf{Y}\beta\|_{\ell_2}^2 + \lambda \|\beta\|_{\ell_1} \quad \text{subject to } \beta_i = 0.$$

Whether such a methodology should succeed is unclear as we are not under a traditional model for both the response  $\mathbf{y}_i$  and the covariates  $\mathbf{Y}$  are noisy; see [45] for a discussion of sparse regression under matrix uncertainty and what can go wrong. The main contribution of this paper is to show that if one selects  $\lambda$  in a data-driven fashion, then compelling practical and theoretical performance can be achieved.

2.4.1. *About as many true discoveries as dimension.* The nature of the problem is such that we wish to make few false discoveries (and not link too many pairs belonging to different subspaces) and so we would like to choose  $\lambda$  large. At the same time, we wish to make many true discoveries, whence a natural trade off. The reason why we need many true discoveries is that spectral clustering needs to assign points to the same cluster when they indeed lie near the same subspace. If the matrix  $\mathbf{B}$  is too sparse, this will not happen.

We now introduce a principle for selecting the regularization parameter; our exposition here is informal and we refer to Section 3 and the supplemental article [47] for precise statements and proofs. Suppose we have noiseless data so that  $\mathbf{Y} = \mathbf{X}$ , and thus solve (2.1) with equality constraints. Under our model, assuming there are no false discoveries, the optimal solution is guaranteed to have exactly  $d$ —the dimension of the subspace the sample under study belongs to—nonzero coefficients with probability one. That is to say, when the point lies in a  $d$ -dimensional space, we find  $d$  “neighbors.”

The selection rule we shall analyze in this paper is to take  $\lambda$  as large as possible (as to prevent false discoveries) while making sure that the number of true discoveries is also on the order of the dimension  $d$ , typically in the range  $[0.5d, 0.8d]$ . We can say this differently. Imagine that all the points lie in the same subspace of dimension  $d$  so that every discovery is true. Then we wish to select  $\lambda$  in such a way that the number of discoveries is a significant fraction of  $d$ , the number one would get with noiseless data. Which value of  $\lambda$  achieves this goal? We will see in Section 2.4.2 that the answer is around  $1/\sqrt{d}$ . To put this in context, this means that we wish to select a regularization parameter which depends upon the dimension  $d$  of the subspace our point comes from. (We are aware that the dependence on  $d$  is unusual as in sparse regression the regularization parameter usually does not depend upon the sparsity of the solution.) In turn, this immediately raises another question: since  $d$  is unknown, how can we proceed? In Section 2.4.4, we will see that it is possible to guess the dimension and construct fairly reliable estimates.



2.4.2. *Data-dependent regularization.* We now discuss values of  $\lambda$  obeying the demands formulated in the previous section. Our arguments are informal and we refer the reader to Section 3 for rigorous statements and to the supplemental article [47]. First, it simplifies the discussion to assume that we have no noise (the noisy case assuming  $\sigma \ll 1$  is similar). Following our earlier discussion, imagine we have a vector  $\mathbf{x} \in \mathbb{R}^n$  lying in the  $d$ -dimensional span of the columns of an  $n \times N$  matrix  $\mathbf{X}$ . We are interested in values of  $\lambda$  so that the minimizer  $\hat{\boldsymbol{\beta}}$  of the LASSO functional

$$K(\boldsymbol{\beta}, \lambda) = \frac{1}{2} \|\mathbf{x} - \mathbf{X}\boldsymbol{\beta}\|_{\ell_2}^2 + \lambda \|\boldsymbol{\beta}\|_{\ell_1}$$

has a number of nonzero components in the range  $[0.5d, 0.8d]$ , say. Now let  $\hat{\boldsymbol{\beta}}_{\text{eq}}$  be the solution of the problem with equality constraints, or equivalently of the problem above with  $\lambda \rightarrow 0^+$ . Then

$$(2.3) \quad \frac{1}{2} \|\mathbf{x} - \mathbf{X}\hat{\boldsymbol{\beta}}\|_{\ell_2}^2 \leq K(\hat{\boldsymbol{\beta}}, \lambda) \leq K(\hat{\boldsymbol{\beta}}_{\text{eq}}, \lambda) = \lambda \|\hat{\boldsymbol{\beta}}_{\text{eq}}\|_{\ell_1}.$$

We make two observations: the first is that if  $\hat{\boldsymbol{\beta}}$  has a number of nonzero components in the range  $[0.5d, 0.8d]$ , then  $\|\mathbf{x} - \mathbf{X}\hat{\boldsymbol{\beta}}\|_{\ell_2}^2$  has to be greater than or equal to a fixed numerical constant. The reason is that we cannot approximate to arbitrary accuracy a generic vector living in a  $d$ -dimensional subspace as a linear combination of about  $d/2$  elements from that subspace. The second observation is that  $\|\hat{\boldsymbol{\beta}}_{\text{eq}}\|_{\ell_1}$  is on the order of  $\sqrt{d}$ , which is a fairly intuitive scaling (we have  $d$  coordinates, each of size about  $1/\sqrt{d}$ ). This holds with the proviso that the algorithm operates correctly in the noiseless setting and does not select columns from other subspaces. Then (2.3) implies that  $\lambda$  has to scale at least like  $1/\sqrt{d}$ . On the other hand,  $\hat{\boldsymbol{\beta}} = \mathbf{0}$  if  $\lambda \geq \|\mathbf{X}^T \mathbf{x}\|_{\ell_\infty}$ . Now the informed reader knows that  $\|\mathbf{X}^T \mathbf{x}\|_{\ell_\infty}$  scales at most like  $\sqrt{(\log N)/d}$  so that choosing  $\lambda$  around this value yields no discovery (one can refine this argument to show that  $\lambda$  cannot be higher than a constant times  $1/\sqrt{d}$  as we would otherwise have a solution that is too sparse). Hence,  $\lambda$  is around  $1/\sqrt{d}$ .

It might be possible to compute a precise relationship between  $\lambda$  and the expected number of true discoveries in an asymptotic regime in which the number of points and the dimension of the subspace both increase to infinity in a fixed ratio by adapting ideas from [8, 9]. We will not do so here as this is beyond the scope of this paper. Rather, we investigate this relationship by means of a numerical study.

Here, we fix a single subspace in  $\mathbb{R}^n$  with  $n = 2000$ . We use a sampling density equal to  $\rho = 5$  and vary the dimension  $d \in \{10, 20, 50, 100, 150, 200\}$  of the subspace as well as the noise level  $\sigma \in \{0.25, 0.5\}$ . For each data point, we solve (2.2) for different values of  $\lambda$  around the heuristic  $\lambda_o = 1/\sqrt{d}$ , namely,  $\lambda \in [0.1\lambda_o, 2\lambda_o]$ . In our experiments, we declare a discovery if an entry in the optimal solution exceeds  $10^{-3}$ . Figure 1(a) and (b) shows the number of discoveries per subspace dimension (the number of discoveries divided by  $d$ ). One can clearly see that the curves corresponding to various subspace dimensions stack up on top of each other,

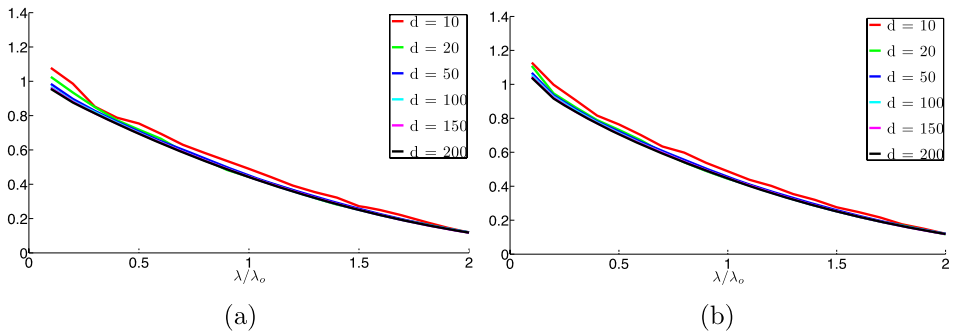


FIG. 1. Average number of true discoveries normalized by subspace dimension for values of  $\lambda$  in an interval including the heuristic  $\lambda_o = 1/\sqrt{d}$ . (a)  $\sigma = 0.25$ . (b)  $\sigma = 0.5$ .

thereby confirming that a value of  $\lambda$  on the order of  $1/\sqrt{d}$  yields a fixed fraction of true discoveries. Further inspection also reveals that the fraction of true discoveries is around 50% near  $\lambda = \lambda_o$ , and around 75% near  $\lambda = \lambda_o/2$ . We have observed empirically that increasing  $\rho$  typically yields a slight increase in the fraction of true discoveries (unless, of course,  $\rho$  is exponentially large in  $d$ ).

**2.4.3. The false-true discovery trade off.** We now show empirically that in our model choosing  $\lambda$  around  $1/\sqrt{d}$  typically yields very few false discoveries as well as many true discoveries; this holds with the proviso that the subspaces are of course not very close to each other.

In this simulation, 22 subspaces of varying dimensions in  $\mathbb{R}^n$  with  $n = 2000$  have been independently selected uniformly at random; there are 5, 4, 3, 4, 4 and 2 subspaces of respective dimensions 200, 150, 100, 50, 20 and 10. This is a challenging regime since the sum of the subspace dimensions equals 2200 and exceeds the ambient dimension (the clean data matrix  $\mathbf{X}$  has full rank). We use a sampling density equal to  $\rho = 5$  for each subspace and set the noise level to  $\sigma = 0.3$ . To evaluate the performance of the optimization problem (2.2), we proceed by selecting a subset of columns as follows: for each dimension, we take 100 cases at random belonging to subspaces of that dimension. Hence, the total number of test cases is  $m = 600$  so that we only solve  $m$  optimization problems (2.2) out of the total  $N$  possible cases. Below,  $\beta^{(i)}$  is the solution to (2.2) and  $\beta_S^{(i)}$  its restriction to columns with indices in the same subspace. Hence, a nonvanishing entry in  $\beta_S^{(i)}$  is a true discovery, and likewise, a nonvanishing entry in  $\beta_{S^c}^{(i)}$  is false. For each data point, we sweep the tuning parameter  $\lambda$  in (2.2) around the heuristic  $\lambda_o = 1/\sqrt{d}$  and work with  $\lambda \in [0.05\lambda_o, 2.5\lambda_o]$ . In our experiments, a discovery is a value obeying  $|B_{ij}| > 10^{-3}$ .

In analogy with the signal detection literature, we view the empirical averages of  $\|\beta_{S^c}^{(i)}\|_{\ell_0}/(n-d)$  and  $\|\beta_S^{(i)}\|_{\ell_0}/d$  as False Positive Rate (FPR) and True Positive Rate (TPR). On the one hand, Figure 2(a) and (b) shows that for values around

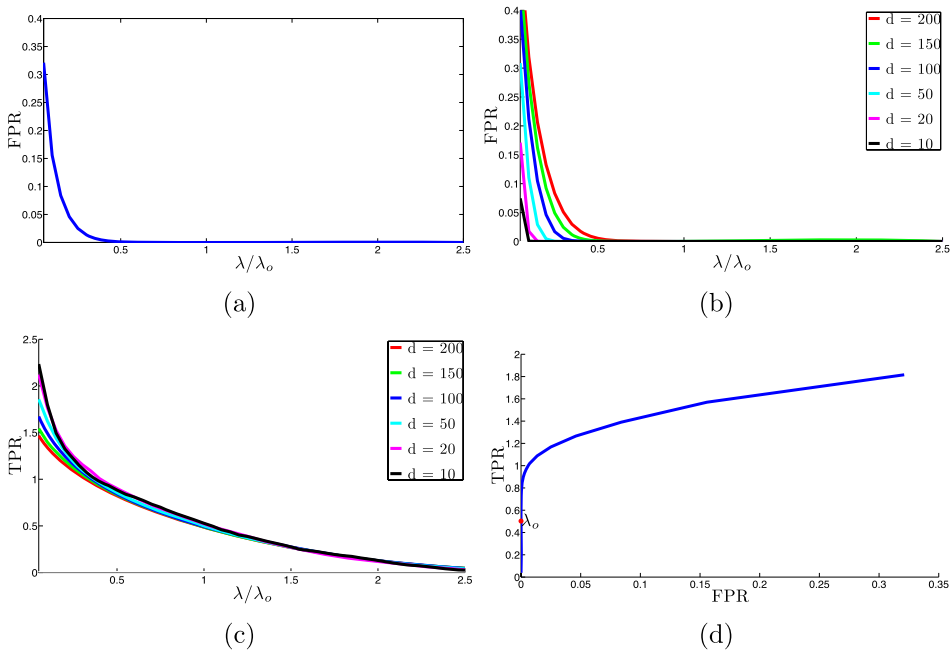


FIG. 2. Performance of LASSO for values of  $\lambda$  in an interval including the heuristic  $\lambda_o = 1/\sqrt{d}$ . (a) Average number of false discoveries normalized by  $(n - d)$  (FPR) on all  $m$  sampled data points. (b) FPR for different subspace dimensions. Each curve represents the average FPR over those samples originating from subspaces of the same dimension. (c) Average number of true discoveries per dimension for various dimensions (TPR). (d) TPR vs. FPR (ROC curve). The point corresponding to  $\lambda = \lambda_o$  is marked as a red dot.

$\lambda = \lambda_o$ , the FPR is zero (so there are no false discoveries). On the other hand, Figure 2(c) shows that the TPR curves corresponding to different dimensions are very close to each other and resemble those in Figure 2(c) in which all the points belong to the same cluster with no opportunity of making a false discovery. Hence, taking  $\lambda$  near  $1/\sqrt{d}$  gives a performance close to what can be achieved in a noiseless situation. That is to say, we have no false discovery and a number of true discoveries about  $d/2$  if we choose  $\lambda = \lambda_o$ . Figure 2(d) plots TPR versus FPR [a.k.a. the Receiver Operating Characteristic (ROC) curve] and indicates that  $\lambda = \lambda_o$  (marked by a red dot) is an attractive trade-off as it provides no false discoveries and sufficiently many true discoveries.

2.4.4. A two-step procedure. Returning to the selection of the regularization parameter, we would like to use  $\lambda$  on the order of  $1/\sqrt{d}$ . However, we do not know  $d$  and proceed by substituting an estimate. In the next section, we will see that we are able to quantify theoretically the performance of the following proposal: (1) run a hard constrained version of the LASSO and use an estimate  $\hat{d}$  of dimension

---

**Algorithm 2** Two-step procedure with data-driven regularization

---

**for**  $i = 1, \dots, N$  **do**

1. Solve

$$(2.4) \quad \boldsymbol{\beta}^* = \arg \min_{\boldsymbol{\beta} \in \mathbb{R}^N} \|\boldsymbol{\beta}\|_{\ell_1} \quad \text{subject to } \|\mathbf{y}_i - \mathbf{Y}\boldsymbol{\beta}\|_{\ell_2} \leq \tau \quad \text{and} \quad \boldsymbol{\beta}_i = 0.$$

2. Set  $\lambda = f(\|\boldsymbol{\beta}^*\|_{\ell_1})$ .

3. Solve

$$\hat{\boldsymbol{\beta}} = \arg \min_{\boldsymbol{\beta} \in \mathbb{R}^N} \frac{1}{2} \|\mathbf{y}_i - \mathbf{Y}\boldsymbol{\beta}\|_{\ell_2}^2 + \lambda \|\boldsymbol{\beta}\|_{\ell_1} \quad \text{subject to } \boldsymbol{\beta}_i = 0.$$

4. Set  $\mathbf{B}_i = \hat{\boldsymbol{\beta}}$ .

**end for**

---

based on the  $\ell_1$  norm of the fitted coefficient sequence; (2) impute a value for  $\lambda$  constructed from  $\hat{d}$ . The two-step procedure is explained in Algorithm 2. Again, our exposition is informal here and we refer to Section 3 for precise statements.

To understand the rationale behind this, imagine we have noiseless data—that is,  $\mathbf{Y} = \mathbf{X}$ —and are solving (2.1), which simply is our first step (2.4) with the proviso that  $\tau = 0$ . When there are no false discoveries, one can show that the  $\ell_1$  norm of  $\boldsymbol{\beta}^*$  is roughly of size  $\sqrt{d}$  as shown in Lemma A.2 from the supplemental article [47]. This suggests using a multiple of  $\|\boldsymbol{\beta}^*\|_{\ell_1}$  as a proxy for  $\sqrt{d}$ . To drive this point home, take a look at Figure 3(a) which solves (2.4) with the same data as in the previous example and  $\tau = 2\sigma$ . The plot reveals that the values of  $\|\boldsymbol{\beta}^*\|_{\ell_1}$  fluctuate around  $\sqrt{d}$ . This is shown more clearly in Figure 3(b), which shows that  $\|\boldsymbol{\beta}^*\|_{\ell_1}$  is concentrated around  $\frac{1}{4}\sqrt{d}$  with, as expected, higher volatility at lower values of dimension.

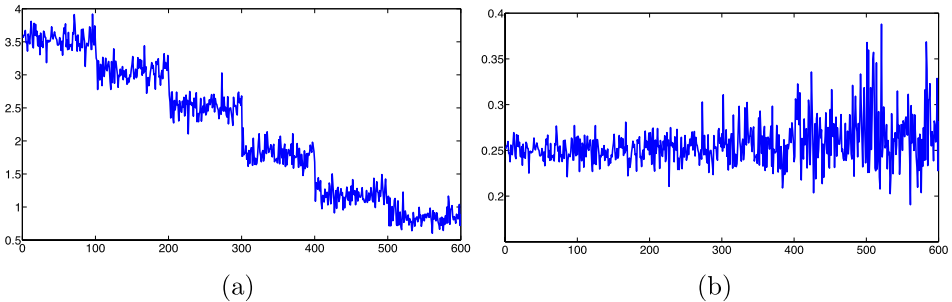


FIG. 3. Optimal values of (2.4) for 600 samples using  $\tau = 2\sigma$ . The first 100 values correspond to points originating from subspaces of dimension  $d = 200$ , the next 100 from those of dimension  $d = 150$ , and so on through  $d \in \{100, 50, 20, 10\}$ . (a) Value of  $\|\boldsymbol{\beta}^*\|_{\ell_1}$ . (b) Value of  $\|\boldsymbol{\beta}^*\|_{\ell_1} / \sqrt{d}$ .

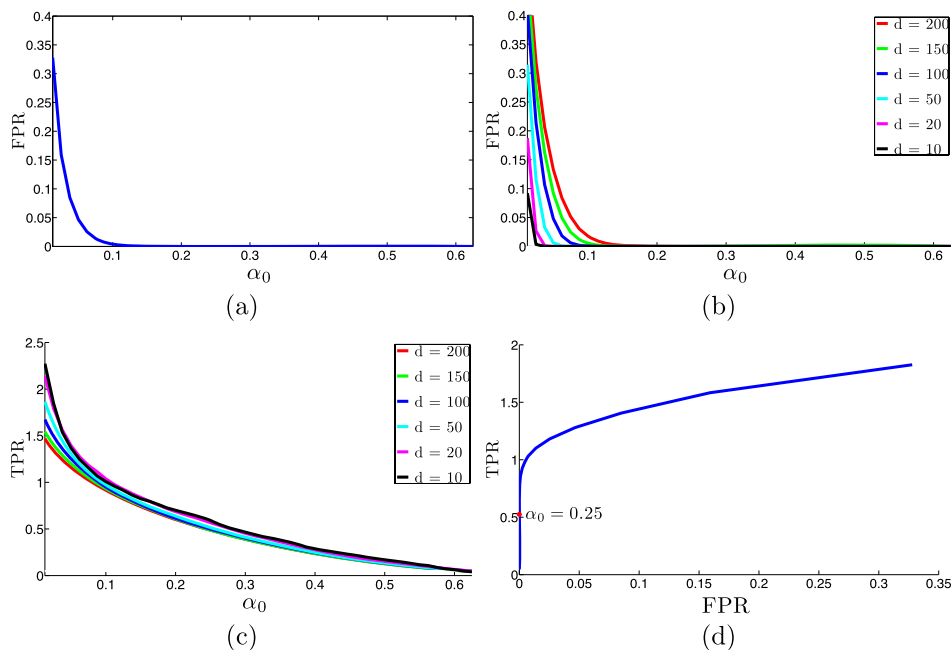


FIG. 4. Performance of the two-step procedure using  $\tau = 2\sigma$  and  $f(t) = \alpha_0 t^{-1}$  for values of  $\alpha_0$  around the heuristic  $\alpha_0 = 0.25$ . (a) False positive rate (FPR). (b) FPR for various subspace dimensions. (c) True positive rate (TPR). (d) TPR vs. FPR.

Under suitable assumptions, we shall see in Section 3 that with noisy data, there are simple rules for selecting  $\tau$  that guarantee, with high probability, that there are no false discoveries. To be concrete, one can take  $\tau = 2\sigma$  and  $f(t) \propto t^{-1}$ . Returning to our running example, we have  $\|\beta^*\|_{\ell_1} \approx \frac{1}{4}\sqrt{d}$ . Plugging this into  $\lambda = 1/\sqrt{d}$  suggests taking  $f(t) \approx 0.25t^{-1}$ . The plots in Figure 4 demonstrate that this is indeed effective. Experiments in Section 6 indicate that this is a good choice on real data as well.

The two-step procedure requires solving two LASSO problems for each data point and is useful when there are subspaces of large dimensions (in the hundreds, say) and some others of low-dimensions (three or four, say). In some applications such as motion segmentation in computer vision, the dimensions of the subspaces are all equal and known in advance [51]. In this case, one can forgo the two-step procedure and simply set  $\lambda = 1/\sqrt{d}$ .

**3. Theoretical results.** This section presents our main theoretical results concerning the performance of the two-step procedure (Algorithm 2). We defer the proof of these results to the supplemental article [47]. We make two assumptions:

- *Affinity condition.* We say that a subspace  $S_\ell$  obeys the *affinity condition* if

$$(3.1) \quad \max_{k: k \neq \ell} \text{aff}(S_\ell, S_k) \leq \kappa_0 / \log N,$$

where  $\kappa_0$  a fixed numerical constant.

- *Sampling condition.* We say that subspace  $S_\ell$  obeys the *sampling condition* if

$$(3.2) \quad \rho_\ell \geq \rho^*,$$

where  $\rho^*$  is a fixed numerical constant.

The careful reader might argue that we should require smaller affinity values as the noise level increases. The reason why  $\sigma$  does not appear in (3.1) is that we assumed a bounded noise level. For higher values of  $\sigma$ , the affinity condition would read as in (3.1) with a right-hand side equal to

$$\kappa = \frac{\kappa_0}{\log N} - \sigma \sqrt{\frac{d_\ell}{2n \log N}}.$$

3.1. *Main results.* From here on, we use  $d(i)$  to refer to the dimension of the subspace the vector  $\mathbf{y}_i$  originates from.  $N(i)$  and  $\rho(i)$  are used in a similar fashion for the number and density of points on this subspace.

**THEOREM 3.1 (No false discoveries).** *Assume that the subspace attached to the  $i$ th column obeys the affinity and sampling conditions and that the noise level  $\sigma$  is bounded as in (1.2), where  $\sigma^*$  is a sufficiently small numerical constant. In Algorithm 2, take  $\tau = 2\sigma$  and  $f(t)$  obeying  $f(t) \geq 0.707\sigma t^{-1}$ . Then with high probability,<sup>6</sup> there is no false discovery in the  $i$ th column of  $\mathbf{B}$ .*

**THEOREM 3.2 (Many true discoveries).** *Consider the same setup as in Theorem 3.1 with  $f(\cdot)$  also obeying  $f(t) \leq \alpha_0 t^{-1}$  for some numerical constant  $\alpha_0$ . Then with high probability,<sup>7</sup> there are at least*

$$(3.3) \quad c_1 \frac{d(i)}{\log \rho(i)}$$

*true discoveries in the  $i$ th column ( $c_1$  is a positive numerical constant).*

The above results indicate that the first step of the algorithm works correctly in fairly broad conditions. To give an example, assume two subspaces of dimension  $d$  overlap in a smaller subspace of dimension  $s$  but are orthogonal to each other

<sup>6</sup>Probability at least  $1 - 2e^{-\gamma_1 n} - 6e^{-\gamma_2 d(i)} - e^{-\sqrt{N(i)d(i)}} - \frac{23}{N^2}$ , for fixed numerical constants  $\gamma_1, \gamma_2$ .

<sup>7</sup>Probability at least  $1 - 2e^{-\gamma_1 n} - 6e^{-\gamma_2 d(i)} - e^{-\sqrt{N(i)d(i)}} - \frac{23}{N^2}$ , for fixed numerical constants  $\gamma_1, \gamma_2$ .

in the remaining directions (equivalently, the first  $s$  principal angles are 0 and the rest are  $\pi/2$ ). In this case, the affinity between the two subspaces is equal to  $\sqrt{s/d}$  and (3.1) allows  $s$  to grow almost linearly in the dimension of the subspaces. Hence, subspaces can have intersections of large dimensions. In contrast, previous work with perfectly noiseless data [21] would impose to have a first principal angle obeying  $|\cos \theta^{(1)}| \leq 1/\sqrt{d}$  so that the subspaces are practically orthogonal to each other. Whereas our result shows that we can have an average of the cosines practically constant, the condition in [21] asks that the maximum cosine be very small.

In the noiseless case, [46] showed that when the sampling condition holds and

$$\max_{k:k \neq \ell} \text{aff}(S_\ell, S_k) \leq \kappa_0 \frac{\sqrt{\log \rho_\ell}}{\log N}$$

(albeit with slightly different values  $\kappa_0$  and  $\rho^*$ ), then applying the noiseless version (2.1) of the algorithm also yields no false discoveries. Hence, with the proviso that the noise level is not too large, conditions under which the algorithm is provably correct are essentially the same.

Earlier, we argued that we would like to have, if possible, an algorithm provably working at (1) high values of the affinity parameters and (2) low values of the sampling density as these are the conditions under which the clustering problem is challenging. (Another property on the wish list is the ability to operate properly with high noise or low SNR and this is discussed next.) In this context, since the affinity is at most one, our results state that the affinity can be within a log factor from this maximum possible value. The number of samples needed per subspace is minimal as well. That is, as long as the density of points on each subspace is larger than a constant  $\rho > \rho^*$ , the algorithm succeeds.<sup>8</sup>

We would like to have a procedure capable of making no false discoveries and many true discoveries at the same time. Now in the noiseless case, whenever there are no false discoveries, the  $i$ th column contains exactly  $d(i)$  true discoveries. Theorem 3.2 states that as long as the noise level  $\sigma$  is less than a fixed numerical constant, the number of true discoveries is roughly on the same order as in the noiseless case. In other words, a noise level of this magnitude does not fundamentally affect the performance of the algorithm. This holds even when there is great variation in the dimensions of the subspaces, and is possible because  $\lambda$  is appropriately tuned in an adaptive fashion.

The number of true discoveries is shown to scale at least like dimension over the log of the density. This may suggest that the number of true discoveries decreases (albeit very slowly) as the sampling density increases. This behavior is to be expected: when the sampling density becomes exponentially large (in terms of the

---

<sup>8</sup>This is with the proviso that the density does not grow exponentially in the dimension of the subspace. This is not a restrictive assumption as having exponentially many points from the same subspace makes the problem especially easy.

dimension of the subspace) the number of true discoveries become small since we need fewer columns to synthesize a point. In fact, the  $d/\log \rho$  behavior seems to be the correct scaling. Indeed, when the density is low and  $\rho$  takes on a small value, (3.3) asserts that we make on the order of  $d$  discoveries, which is tight. Imagine now that we are in the high-density regime and  $\rho$  is exponential in  $d$ . Then as the points gets tightly packed, we expect to have only one discovery in accordance with (3.3).

Theorem 3.2 establishes that there are many true discoveries. This would not be useful for clustering purposes if there were only a handful of very large true discoveries and all the others of negligible magnitude. The reason is that the similarity matrix  $\mathbf{W}$  would then be close to a sparse matrix and we would run the risk of splitting true clusters. Our proofs show that this does not happen although we do not present an argument for lack of space. Rather, we demonstrate this property empirically. On our running example, Figure 5(a) and (b) shows that the histograms of appropriately normalized true discovery values resemble a bell-shaped curve. Note that each true discovery corresponds to a nonzero coefficient which can take on either a positive or negative value.

As stated numerous times, our theoretical analysis only concerns the first step of the algorithm. We now wish to explain how these theoretical results relate to complete guarantees for clustering. First, Theorem 3.1 states that clusters that should be disconnected from each other are, in fact, disconnected so that the algorithm does not group together points from different subspaces. To guarantee perfect clustering, it is then sufficient to show that each restriction of the similarity graph to a subspace is connected. Due to the nature of the random model under study, a subgraph resembles an Erdős–Rényi graph with the probability of having an edge roughly proportional to the number of true discoveries. As long as there are sufficiently many true discoveries (as shown in Theorem 3.2), such a graph is well connected—in fact, it has very good expansion properties. Proving that each subgraph is indeed connected is a problem we regard as interesting, the main challenge being caused by the dependencies the algorithm generates. Second, a more quantitative characterization of the expansion or connectedness of each subgraph via

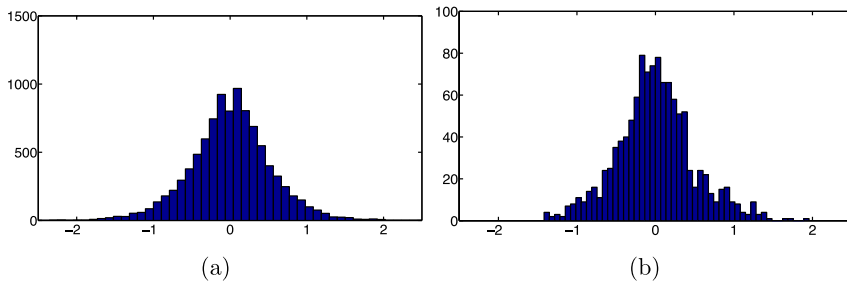


FIG. 5. Histograms of the true discovery values from the two step procedure with  $\alpha_0 = 0.25$  (multiplied by  $\sqrt{d}$ ). (a)  $d = 200$ . (b)  $d = 20$ .



Cheeger’s constant or the eigenvalue gap may ultimately demonstrate that the algorithm succeeds even in the presence of few false discoveries with small values of  $W_{ij}$ ; please see [27] and references therein.

Finally, we would like to comment on the fact that our main results hold when  $\lambda$  belongs to a fairly broad range of values. First, when all the subspaces have small dimensions, one can choose the same value of  $\lambda$  for all the data points since  $1/\sqrt{d}$  is essentially constant. Hence, when we know a priori that we are in such a situation, there may be no need for the two-step procedure. (We would still recommend the conservative two-step procedure because of its superior empirical performance on real data.) Second, the proofs also reveal that if we have knowledge of the dimension of the largest subspace  $d_{\max}$ , the first theorem holds with a fixed value of  $\lambda$  proportional to  $\sigma/\sqrt{d_{\max}}$ . Third, when the subspaces themselves are drawn at random, the first theorem holds with a fixed value of  $\lambda$  proportional to  $\sigma(\log N)/\sqrt{n}$ . (Both these statements follow by plugging these values of  $\lambda$  in the proofs of the supplemental article [47] and we omit the calculations.) We merely mention these variants to give a sense of what our theorems can also give. As explained earlier, we recommend the more conservative two-step procedure with the proxy for  $1/\sqrt{d}$ . The reason is that using a higher value of  $\lambda$  allows for a larger value of  $\kappa_0$  in (3.1), which says that the subspaces can be even closer. In other words, we can function in a more challenging regime. To drive this point home, consider the noiseless problem. When the subspaces are close, the equality constrained  $\ell_1$  problem may yield some false discoveries. However, if we use the LASSO version—even though the data is noiseless—we may end up with no false discoveries while maintaining sufficiently many true discoveries.

**4. The bias-corrected Dantzig selector.** One can think of other ways of performing the first step in Algorithm 1 and this section discusses another approach based on a modification of the Dantzig selector, a popular sparse regression technique [15]. Unlike the two-step procedure, we do not claim any theoretical guarantees for this method and shall only explore its properties on real and simulated data.

Applied directly to our problem, the Dantzig selector takes the form

$$(4.1) \quad \min_{\beta \in \mathbb{R}^N} \|\beta\|_{\ell_1} \quad \text{subject to} \quad \|\mathbf{Y}_{(-i)}^T(\mathbf{y}_i - \mathbf{Y}\beta)\|_{\ell_\infty} \leq \lambda \quad \text{and} \quad \beta_i = 0,$$

where  $\mathbf{Y}_{(-i)}$  is  $\mathbf{Y}$  with the  $i$ th column deleted. However, this is hardly suitable since the design matrix  $\mathbf{Y}$  is corrupted. Interestingly, recent work [44, 45] has studied the problem of estimating a sparse vector from the standard linear model under uncertainty in the design matrix. The setup in these papers is close to our problem and we propose a modified Dantzig selection procedure inspired but not identical to the methods set forth in [44, 45].

4.1. *The correction.* If we had clean data, we would solve (2.1); this is (4.1) with  $\mathbf{Y} = \mathbf{X}$  and  $\lambda = 0$ . Let  $\boldsymbol{\beta}^I$  be the solution to this ideal noiseless problem. Applied to our problem, the main idea in [44, 45] would be to find a formulation that resembles (4.1) with the property that  $\boldsymbol{\beta}^I$  is feasible. Since  $\mathbf{x}_i = \mathbf{X}_{(-i)}\boldsymbol{\beta}^I_{(-i)}$ , observe that we have the following decomposition:

$$\begin{aligned} \mathbf{Y}_{(-i)}^T(\mathbf{y}_i - \mathbf{Y}\boldsymbol{\beta}^I) &= (\mathbf{X}_{(-i)} + \mathbf{Z}_{(-i)})^T(\mathbf{z}_i - \mathbf{Z}\boldsymbol{\beta}^I) \\ &= \mathbf{X}_{(-i)}^T(\mathbf{z}_i - \mathbf{Z}\boldsymbol{\beta}^I) + \mathbf{Z}_{(-i)}^T\mathbf{z}_i - \mathbf{Z}_{(-i)}^T\mathbf{Z}\boldsymbol{\beta}^I. \end{aligned}$$

Then the conditional mean is given by

$$\mathbb{E}[\mathbf{Y}_{(-i)}^T(\mathbf{y}_i - \mathbf{Y}\boldsymbol{\beta}^I)|\mathbf{X}] = -\mathbb{E}\mathbf{Z}_{(-i)}^T\mathbf{Z}_{(-i)}\boldsymbol{\beta}^I_{(-i)} = -\sigma^2\boldsymbol{\beta}^I_{(-i)}.$$

In other words,

$$\sigma^2\boldsymbol{\beta}^I_{(-i)} + \mathbf{Y}_{(-i)}^T(\mathbf{y}_i - \mathbf{Y}\boldsymbol{\beta}^I) = \boldsymbol{\xi},$$

where  $\boldsymbol{\xi}$  has mean zero. In Section 4.2, we compute the variance of the  $j$ th component  $\xi_j$ , given by

$$(4.2) \quad \mathbb{E}\xi_j^2 = \frac{\sigma^2}{n}(1 + \|\boldsymbol{\beta}^I\|_{\ell_2}^2) + \frac{\sigma^4}{n}(1 + (\beta_j^I)^2 + \|\boldsymbol{\beta}^I\|_{\ell_2}^2).$$

Owing to our Gaussian assumptions,  $|\xi_j|$  shall be smaller than 3 or 4 times this standard deviation, say, with high probability.

Hence, we may want to consider a procedure of the form

$$(4.3) \quad \begin{aligned} \min_{\boldsymbol{\beta} \in \mathbb{R}^N} \|\boldsymbol{\beta}\|_{\ell_1} \quad &\text{subject to } \|\mathbf{Y}_{(-i)}^T(\mathbf{y}_i - \mathbf{Y}\boldsymbol{\beta}) + \sigma^2\boldsymbol{\beta}_{(-i)}\|_{\ell_\infty} \leq \lambda \quad \text{and} \\ \boldsymbol{\beta}_i &= 0. \end{aligned}$$

It follows that if we take  $\lambda$  to be a reasonable multiple of (4.2), then  $\boldsymbol{\beta}^I$  would obey the constraint in (4.3) with high probability. Hence, we would need to approximate the variance (4.2). Numerical simulations together with asymptotic calculations presented in the supplemental article [47] give that  $\|\boldsymbol{\beta}^I\|_{\ell_2} \leq 1$  with very high probability. Thus, neglecting the term in  $(\beta_j^I)^2$ ,

$$\mathbb{E}\xi_j^2 \approx \frac{\sigma^2}{n}(1 + \sigma^2)(1 + \|\boldsymbol{\beta}^I\|_{\ell_2}^2) \leq 2\frac{\sigma^2}{n}(1 + \sigma^2).$$

This suggests taking  $\lambda$  to be a multiple of  $\sqrt{2/n}\sigma\sqrt{1 + \sigma^2}$ . This is interesting because the parameter  $\lambda$  does not depend on the dimension of the underlying subspace. We shall refer to (4.3) as the *bias-corrected Dantzig selector*, which resembles the proposal in [44, 45] for which the constraint is a bit more complicated and of the form  $\|\mathbf{Y}_{(-i)}^T(\mathbf{y}_i - \mathbf{Y}\boldsymbol{\beta}) + \mathbf{D}_{(-i)}\boldsymbol{\beta}\|_{\ell_\infty} \leq \mu\|\boldsymbol{\beta}\|_{\ell_1} + \lambda$ .

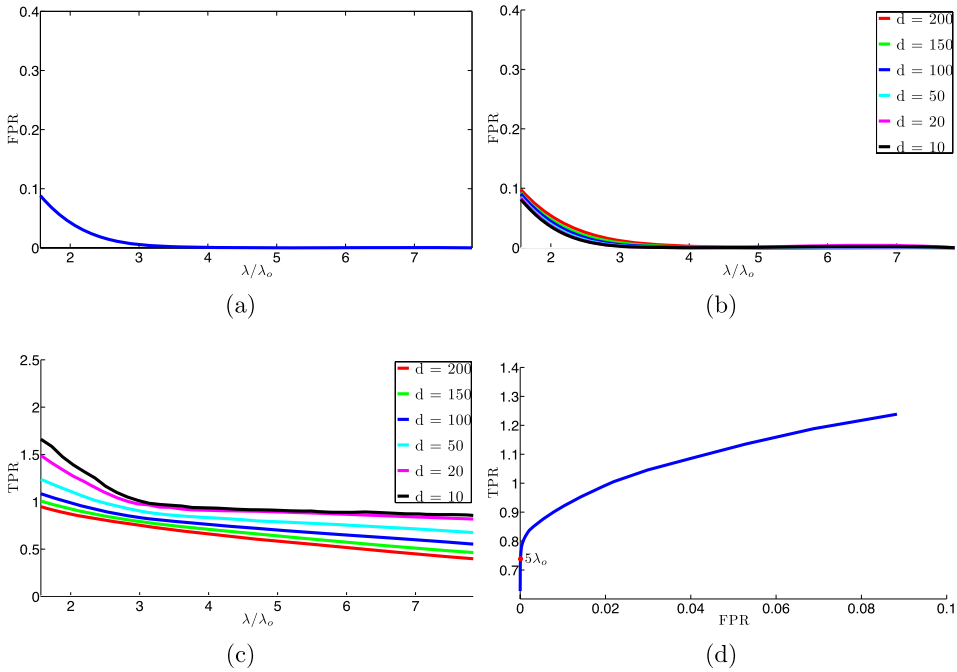


FIG. 6. Performance of the bias-corrected Dantzig selector for values of  $\lambda$  that are multiples of the heuristic  $\lambda_o = \sqrt{2/n}\sigma\sqrt{1 + \sigma^2}$ . (a) False positive rate (FPR). (b) FPR for different subspace dimensions. (c) True positive rate (TPR). (d) TPR vs. FPR.

To get a sense about the validity of this proposal, we test it on our running example by varying  $\lambda \in [\lambda_o, 8\lambda_o]$  around the heuristic  $\lambda_o = \sqrt{2/n}\sigma\sqrt{1 + \sigma^2}$ . Figure 6 shows that good results are achieved around factors in the range [4, 6].

In our synthetic simulations, both the two-step procedure and the corrected Dantzig selector seem to be working well in the sense that they yield many true discoveries while making very few false discoveries, if any. Comparing Figure 6(b) and (c) with those from Section 2 show that the corrected Dantzig selector has more true discoveries for subspaces of small dimensions (they are essentially the same for subspaces of large dimensions); that is, the two-step procedure is more conservative when it comes to subspaces of smaller dimensions. As explained earlier, this is due to our conservative choice of  $\lambda$  resulting in a TPR about half of what is obtained in a noiseless setting. Having said this, it is important to keep in mind that in these simulations the planes are drawn at random and as a result, they are sort of far from each other. This is why a less conservative procedure can still achieve a low FPR. When subspaces of smaller dimensions are closer to each other or when the statistical model does not hold exactly as in real data scenarios, a conservative procedure may be more effective. In fact, experiments on real data in Section 6 confirm this and show that for the corrected Dantzig selector, one needs to choose values much larger than  $\lambda_o$  to yield good results.

4.2. *Variance calculation.* By definition,

$$\begin{aligned} \xi_j &= \langle \mathbf{x}_j, \mathbf{z}_i - \mathbf{Z}\boldsymbol{\beta}^I \rangle + \langle \mathbf{z}_j, \mathbf{z}_i \rangle - (\mathbf{z}_j^T \mathbf{z}_j - \sigma^2)\beta_j^I - \sum_{k:k \neq i,j} \mathbf{z}_j^T \mathbf{z}_k \beta_k^I \\ &:= I_1 + I_2 + I_3 + I_4. \end{aligned}$$

A simple calculation shows that for  $\ell_1 \neq \ell_2$ ,  $\text{Cov}(I_{\ell_1}, I_{\ell_2}) = 0$  so that

$$\mathbb{E} \xi_j^2 = \sum_{\ell=1}^4 \text{Var}(I_\ell).$$

We compute

$$\begin{aligned} \text{Var}(I_1) &= \frac{\sigma^2}{n} (1 + \|\boldsymbol{\beta}^I\|_{\ell_2}^2), & \text{Var}(I_3) &= \frac{\sigma^4}{n} 2(\beta_j^I)^2, \\ \text{Var}(I_2) &= \frac{\sigma^4}{n}, & \text{Var}(I_4) &= \frac{\sigma^4}{n} [\|\boldsymbol{\beta}^I\|_{\ell_2}^2 - (\beta_j^I)^2] \end{aligned}$$

and (4.2) follows.

**5. Comparisons with other works.** We now briefly comment on other approaches to subspace clustering. Since this paper is theoretical in nature, we shall focus on comparing theoretical properties and refer to [22, 53] for a detailed comparison about empirical performance. Three themes will help in organizing our discussion.

- *Tractability.* Is the proposed method or algorithm computationally tractable?
- *Robustness.* Is the algorithm provably robust to noise and other imperfections?
- *Efficiency.* Is the algorithm correctly operating near the limits we have identified above? In our model, how many points do we need per subspace? How large can the affinity between subspaces be?

One can broadly classify existing subspace clustering techniques into four categories, namely, algebraic, iterative, statistical and spectral clustering-based methods.

Methods inspired from algebraic geometry have been introduced for clustering purposes. In this area, a mathematically intriguing approach is the *generalized principal component analysis* (GPCA) presented in [54]. Unfortunately, this algorithm is not tractable in the dimension of the subspaces, meaning that a polynomial-time algorithm does not exist. Another feature is that GPCA is not robust to noise although some heuristics have been developed to address this issue; see, for example, [36]. As far as the dependence upon key parameters is concerned, GPCA is essentially optimal. An interesting approach to make GPCA robust is based on semidefinite programming [42]. However, this novel formulation is still intractable in the dimension of the subspaces and it is not clear how the performance of the algorithm depends upon the parameters of interest.

A representative example of an iterative method—the term is taken from the tutorial [53]—is the  $K$ -subspace algorithm [52], a procedure which can be viewed as a generalization of  $K$ -means. Here, the subspace clustering problem is formulated as a nonconvex optimization problem over the choice of bases for each subspace as well as a set of variables indicating the correct segmentation. A cost function is then iteratively optimized over the basis and the segmentation variables. Each iteration is computationally tractable. However, due to the nonconvex nature of the problem, the convergence of the sequence of iterates is only guaranteed to a local minimum. As a consequence, the dependence upon the key parameters is not well understood. Furthermore, the algorithm can be sensitive to noise and outliers. Other examples of iterative methods may be found in [1, 14, 33, 57].

Statistical methods typically model the subspace clustering problem as a mixture of degenerate Gaussian observations. Two such approaches are *mixtures of probabilistic PCA* (MPPCA) [50] and *agglomerative lossy compression* (ALC) [34]. MPPCA seeks to compute a maximum-likelihood estimate of the parameters of the mixture model by using an expected-maximization (EM) style algorithm. ALC searches for a segmentation of the data by minimizing the code length necessary (with a code based on Gaussian mixtures) to fit the points up to a given distortion. Once more, due to the nonconvex nature of these formulations, the dependence upon the key parameters and the noise level is not understood.

Many other methods apply spectral clustering to a specially constructed graph [2, 4, 13, 16, 17, 25, 55, 58]. They share the same difficulties as stated above and [53] discusses advantages and drawbacks. An approach of this kind is termed *Sparse Curvature Clustering* (SCC) [16, 17]; please also see [3, 4]. This approach is not tractable in the dimension of the subspaces as it requires building a tensor with  $N^{(d+2)}$  entries and involves computations with this tensor. Some theoretical guarantees for this algorithm are given in [16] although its limits of performance and robustness to noise are not fully understood. An approach similar to SCC is called *low-rank representation* (LRR) [31]. The LRR algorithm is tractable but its robustness to noise and its dependence upon key parameters is not understood. The work in [30] formulates the robust subspace clustering problem as a nonconvex geometric minimization problem over the Grassmanian. Because of the nonconvexity, this formulation may not be tractable. On the positive side, this algorithm is provably robust and can accommodate noise levels up to  $\mathcal{O}(1/(Ld^{3/2}))$ . However, the density  $\rho$  required for favorable properties to hold is an unknown function of the dimensions of the subspaces (e.g.,  $\rho$  could depend on  $d$  in a super polynomial fashion). Also, the bound on the noise level seems to decrease as the dimension  $d$  and number of subspaces  $L$  increases. In contrast, our theory requires  $\rho \geq \rho^*$  where  $\rho^*$  is a fixed numerical constant. While this manuscript was under preparation, we learned of [23] which establishes robustness to sparse outliers but with a dependence on the key parameters that is super-polynomial in the dimension of the subspaces demanding  $\rho \geq C_0 d^{\log n}$ . (Numerical simulations in [23] seem to indicate that  $\rho$  cannot be a constant.)

We note that the papers [32, 44, 45] also address regression under corrupted covariates. However, there are three key differences between these studies and our work. First, our results show that LASSO without any change is robust to corrupted covariates whereas these works require modifications to either LASSO or the Dantzig selector. Second, the modeling assumptions for the uncorrupted covariates are significantly different. These papers assume that  $\mathbf{X}$  has i.i.d. rows and obeys the *restricted eigenvalue condition* (REC) whereas we have columns sampled from a mixture model so that the design matrices do not have much in common. Last, for clustering and classification purposes, we need to verify that the support of the solution is correct whereas these works establish closeness to an oracle solution in an  $\ell_2$  sense. In short, our work is far closer to multiple hypothesis testing.

Finally, in the data mining literature subspace clustering is sometimes used to describe a different—although related—problem; see [26, 28, 39].

**6. Numerical experiments.** In this section, we perform numerical experiments corroborating our main results and suggesting their applications to temporal segmentation of motion capture data. In this application, we are given sensor measurements at multiple joints of the human body captured at different time instants. The goal is to segment the sensory data so that each cluster corresponds to the same activity. Here, each data point corresponds to a vector whose elements are the sensor measurements of different joints at a fixed time instant.

We use the Carnegie Mellon Motion Capture dataset (available at <http://mocap.cs.cmu.edu>), which contains 149 subjects performing several activities (data are provided in [59]). The motion capture system uses 42 markers per subject. We consider the data from subject 86 in the dataset, consisting of 15 different trials, where each trial comprises multiple activities. We use trials 2 and 5, which feature more activities (8 activities for trial 2 and 7 activities for trial 5) and are, therefore, harder examples relative to the other trials. Figure 7 shows a few snapshots of each activity (walking, squatting, punching, standing, running, jumping, arms-up and

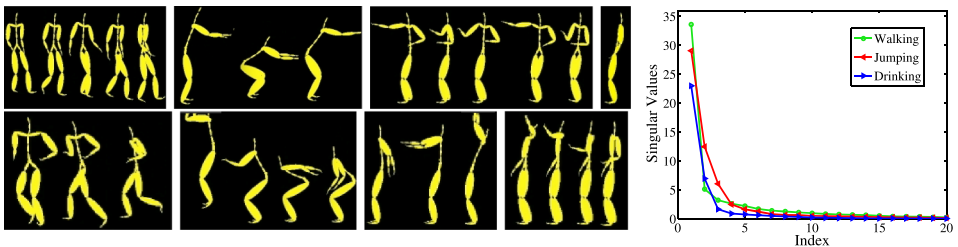


FIG. 7. Left: eight activities performed by subject 86 in the CMU motion capture dataset: walking, squatting, punching, standing, running, jumping, arms-up and drinking. Right: singular values of the data from three activities (walking, jumping, drinking) show that the data from each activity lie approximately in a low-dimensional subspace.

drinking) from trial 2. The right plot in Figure 7 shows the singular values of three of the activities in this trial. Notice that all the curves have a low-dimensional knee, showing that the data from each activity lie in a low-dimensional subspace of the ambient space ( $n = 42$  for all the motion capture data).

We compare three different algorithms: a baseline algorithm, the two-step procedure and the bias-corrected Dantzig selector. We evaluate these algorithms based on the *clustering error*. That is, we assume knowledge of the number of subspaces and apply spectral clustering to the similarity matrix built by the algorithm. After the spectral clustering step, the clustering error is simply the ratio of misclassified points to the total number of points. We report our results on half of the examples—downsampling the video by a factor 2 keeping every other frame—as to make the problem more challenging. (As a side note, it is always desirable to have methods that work well on a smaller number of examples as one can use split-sample strategies for tuning purposes.)<sup>9</sup>

As a baseline for comparison, we apply spectral clustering to a standard similarity graph built by connecting each data point to its  $K$ -nearest neighbors. For pairs of data points,  $\mathbf{y}_i$  and  $\mathbf{y}_j$ , that are connected in the  $K$ -nearest neighbor graph, we define the similarities between them by  $W_{ij} = \exp(-\|\mathbf{y}_i - \mathbf{y}_j\|_2^2/t)$ , where  $t > 0$  is a tuning parameter (a.k.a. temperature). For pairs of data points,  $\mathbf{y}_i$  and  $\mathbf{y}_j$ , that are not connected in the  $K$ -nearest neighbor graph, we set  $W_{ij} = 0$ . Thus, pairs of neighboring data points that have small Euclidean distances from each other are considered to be more similar, since they have high similarity  $W_{ij}$ . We then apply spectral clustering to the similarity graph and measure the clustering error. For each value of  $K$ , we record the minimum clustering error over different choices of the temperature parameter  $t > 0$  as shown in Figure 8(a) and (b). The minimum clustering error for trials 2 and 5 are 17.06% and 12.47%.

For solving the LASSO problems in the two-step procedure, we developed a computational routine made publicly available [60] based on TFOCS [10] solving

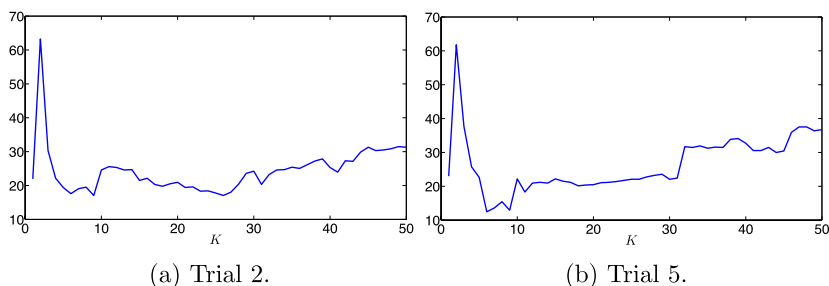


FIG. 8. Minimum clustering error (%) for each  $K$  in the baseline algorithm.

<sup>9</sup>We have adopted this subsampling strategy to make our experiments reproducible. For tuning purposes, a random strategy may be preferable.

the optimization problems in parallel. For the corrected Dantzig selector, we use a homotopy solver in the spirit of [61].

For both the two-step procedure and the bias-corrected Dantzig selector, we normalize the data points as a preprocessing step. We work with a noise  $\sigma$  in the interval  $[0.001, 0.045]$ , and use  $f(t) = \alpha/t$  with values of  $\alpha$  around  $1/4$  (this is equivalent to varying  $\lambda$  around  $1/\lambda_o = 4\|\beta^*\|_{\ell_1}$ ) in the two-step procedure. For the bias-corrected Dantzig selector, we vary  $\lambda$  around  $\lambda_o = \sqrt{2/n}\sigma\sqrt{1+\sigma^2}$ . After building the similarity graph from the sparse regression output, we apply spectral clustering as explained earlier. Figures 9(a) and (b), 10(a) and (b) show the clustering error (on trial 5) and the red point indicates the location where the minimum clustering error is reached. Figure 9(a) and (b) shows that for the two-step procedure the value of the clustering error is not overly sensitive to the choice of  $\sigma$ —especially around  $\lambda = \lambda_o$ . Notice that the clustering error for the robust versions of SSC are significantly lower than the baseline algorithm for a wide range of parameter values. The reason the baseline algorithm performs poorly in this case

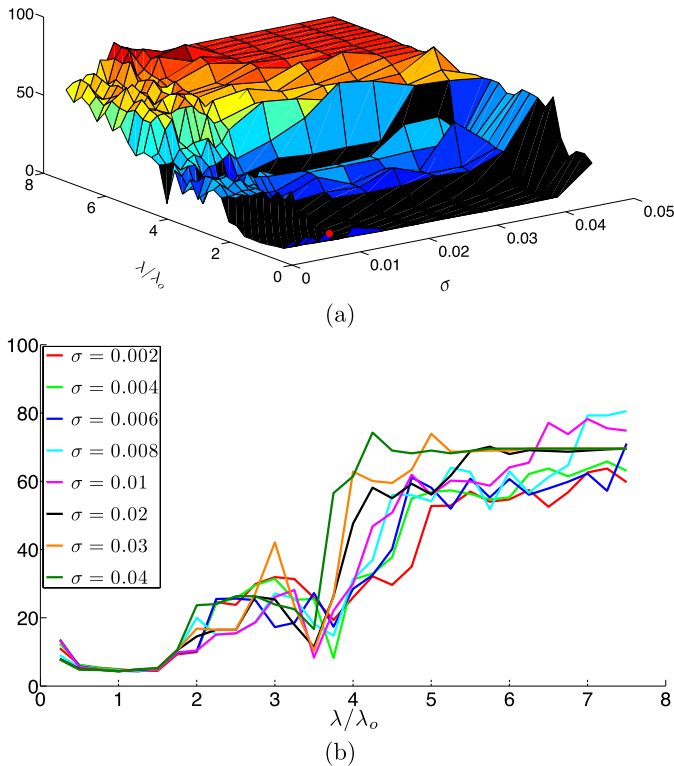


FIG. 9. Clustering error (%) for different values of  $\lambda$  and  $\sigma$  on trial 5 using the two-step procedure (a) 3D plot (minimum clustering error appears in red). (b) 2D cross sections.



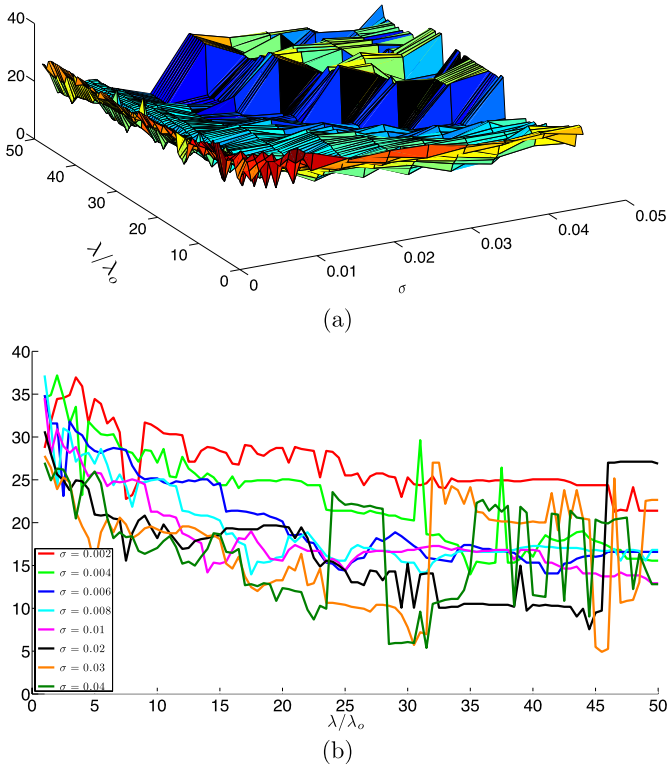


FIG. 10. Clustering error (%) for different values of  $\lambda$  and  $\sigma$  on trial 5 using the corrected Dantzig selector. (a) 3D plot (minimum clustering error appears in red). (b) 2D cross sections.

is that there are many points that are in small Euclidean distances from each other, but belong to different subspaces.

Finally a summary of the clustering errors of these algorithms on the two trials are reported in Table 1. Robust versions of SSC outperform the baseline algorithm. This shows that the multiple subspace model is better for clustering purposes. The two-step procedure seems to work slightly better than the corrected Dantzig selector for these two examples. Table 2 reports the optimal parameters that achieve the minimum clustering error for each algorithm. The table indicates that on real data,

TABLE 1  
Minimum clustering error

	Baseline algorithm	Two-step procedure	Corrected Dantzig selector
Trial 2	17.06%	<b>3.54%</b>	9.53%
Trial 5	12.47%	<b>4.35%</b>	4.92%

TABLE 2  
Optimal parameters

	Baseline algorithm	Two-step procedure	Corrected Dantzig selector
Trial 2	$K = 9, t = 0.0769$	$\sigma = 0.03, \lambda = 1.25\lambda_o$	$\sigma = 0.004, \lambda = 41.5\lambda_o$
Trial 5	$K = 6, t = 0.0455$	$\sigma = 0.01, \lambda = \lambda_o$	$\sigma = 0.03, \lambda = 45.5\lambda_o$

choosing  $\lambda$  close to  $\lambda_o$  also works very well. Also, one can see that in comparison with the synthetic simulations of Section 4, a more conservative choice of the regularization parameter  $\lambda$  is needed for the corrected Dantzig selector as  $\lambda$  needs to be chosen much higher than  $\lambda_o$  to achieve the best results. This may be attributed to the fact that the subspaces in this example are very close to each other and are not drawn at random as was the case with our synthetic data. To get a sense of the affinity values, we fit a subspace of dimension  $d_\ell$  to the  $N_\ell$  data points from the  $\ell$ th group, where  $d_\ell$  is chosen as the smallest nonnegative integer such that the partial sum of the  $d_\ell$  top singular values is at least 90% of the total sum. Figure 11 shows that the affinities are higher than 0.75 for both trials.

**7. Discussion and open problems.** In this paper, we have developed a tractable algorithm that can provably cluster data points in a fairly challenging regime in which subspaces can overlap along many dimensions and in which the number of points per subspace is rather limited. Our results about the performance of the robust SSC algorithm are expressed in terms of interpretable parameters. This is not a trivial achievement: one of the challenges of the theory for subspace clustering is precisely that performance depends on many different aspects of the problem such as the dimension of the ambient space, the number of subspaces, their dimensions, their relative orientations, the distribution of points around each subspace, the noise level and so on. Nevertheless, these results only offer a starting point as our work leaves open lots of questions, and at the same time, suggests topics for future research. Before presenting the proofs, we would like to close by listing a few questions colleagues may find of interest.

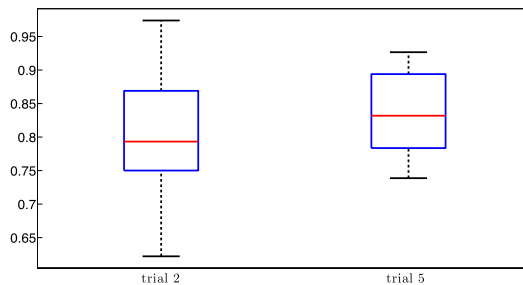


FIG. 11. Box plot of the affinities between subspaces for trials 2 and 5.

- We have shown that while having the affinities and sampling densities near what is information theoretically possible, robust versions of SSC that can accommodate noise levels  $\sigma$  of order one exist. It would be interesting to establish fundamental limits relating the key parameters to the maximum allowable noise level. What is the maximum allowable noise level for any algorithm regardless of tractability?
- It would be interesting to extend the results of this paper to a deterministic model where both the orientation of the subspaces and the noiseless samples are non-random. We leave this to a future publication.
- Our work in this paper concerns the construction of the similarity matrix and the correctness of sparse regression techniques. The full algorithm then applies clustering techniques to clean up errors introduced in the first step. It would be interesting to develop theoretical guarantees for this step as well. A potential approach is the interesting formulation developed in [7].
- We proposed a two-step procedure for robust subspace clustering. The first step is used to estimate the required regularization parameter for a LASSO problem. This is reminiscent of estimating noise in sparse regularization and covariance estimation. It would be interesting to design a joint optimization scheme to simultaneously optimize the regularization parameter and the regression coefficients. In recent years, there has been much progress on this issue in the sparse regression literature; see [11, 19, 24, 48, 49] and references therein. It is an open research direction to see whether any of these approaches can be applied to automatically learn the regularization parameter when both the response vector and covariates are corrupted and, in particular, for the purpose of robust subspace clustering.
- A natural direction is the development of clustering techniques that can provably operate with missing and/or sparsely corrupted entries (the work [46] only deals with grossly corrupted columns). The work in [23] provides one possible approach but requires a very high sampling density as we already mentioned. The paper [22] develops another heuristic approach without any theoretical justification.
- Our formulation uses a data-driven modeling approach by regressing each data point against all others. As noted by Bittorf et al. [12], this type of approach appears in a number of other factorization problems. In particular, [5] and recent variations [5, 12] use a convex formulation very similar to SSC for the purpose of nonnegative matrix factorizations. Exploring the connection between these factorization problems is an interesting research direction.
- One of the advantages of the suggested scheme is that it is highly parallelizable. When the algorithm is run sequentially, it would be interesting to see whether one can reuse computations to solve all the  $\ell_1$ -minimization problems more effectively.

**Acknowledgements.** We thank René Vidal for helpful discussions as well as Ery Arias-Castro, Rina Foygel and Lester Mackey for a careful reading of the manuscript and insightful comments. We also thank the Associate Editor and reviewers for constructive comments. Emmanuel J. Candès would like to thank Chiara Sabatti for invaluable feedback on an earlier version of the paper. He also thanks the organizers of the 41st annual Meeting of Dutch Statisticians and Probabilists held in November 2012 where these results were presented. A brief summary of this work was submitted in August 2012 and presented at the NIPS workshop on Deep Learning in December 2012.

## SUPPLEMENTARY MATERIAL

**Supplement: Proofs** (DOI: [10.1214/13-AOS1199SUPP](https://doi.org/10.1214/13-AOS1199SUPP); .pdf). We prove all of the results of this paper.

## REFERENCES

- [1] AGARWAL, P. K. and MUSTAFA, N. H. (2004).  $k$ -means projective clustering. In *Proceedings of the Twenty-third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* 155–165.
- [2] ALDROUBI, A. and SEKMEN, A. (2012). Nearness to local subspace algorithm for subspace and motion segmentation. *Signal Process. Lett., IEEE* **19** 704–707.
- [3] ARIAS-CASTRO, E. (2011). Clustering based on pairwise distances when the data is of mixed dimensions. *IEEE Trans. Inform. Theory* **57** 1692–1706. [MR2815843](#)
- [4] ARIAS-CASTRO, E., CHEN, G. and LERMAN, G. (2011). Spectral clustering based on local linear approximations. *Electron. J. Stat.* **5** 1537–1587. [MR2861697](#)
- [5] ARORA, S., GE, R., KANNAN, R. and MOITRA, A. (2012). Computing a nonnegative matrix factorization—provably. In *STOC’12—Proceedings of the 2012 ACM Symposium on Theory of Computing* 145–161. ACM, New York. [MR2961503](#)
- [6] BAKO, L. (2011). Identification of switched linear systems via sparse optimization. *Automatica J. IFAC* **47** 668–677. [MR2878328](#)
- [7] BALCAN, M.-F., BLUM, A. and GUPTA, A. (2009). Approximate clustering without the approximation. In *Proceedings of the Twentieth Annual ACM–SIAM Symposium on Discrete Algorithms* 1068–1077. SIAM, Philadelphia, PA. [MR2807549](#)
- [8] BAYATI, M. and MONTANARI, A. (2011). The dynamics of message passing on dense graphs, with applications to compressed sensing. *IEEE Trans. Inform. Theory* **57** 764–785. [MR2810285](#)
- [9] BAYATI, M. and MONTANARI, A. (2012). The LASSO risk for Gaussian matrices. *IEEE Trans. Inform. Theory* **58** 1997–2017. [MR2951312](#)
- [10] BECKER, S. R., CANDÈS, E. J. and GRANT, M. C. (2011). Templates for convex cone problems with applications to sparse signal recovery. *Math. Program. Comput.* **3** 165–218. [MR2833262](#)
- [11] BELLONI, A., CHERNOZHUKOV, V. and WANG, L. (2011). Square-root lasso: Pivotal recovery of sparse signals via conic programming. *Biometrika* **98** 791–806. [MR2860324](#)
- [12] BITTORF, V., RECHT, B., RE, C. and TROPP, J. A. (2012). Factoring nonnegative matrices with linear programs. In *Proceedings of Natural Information Processing Systems Foundation NIPS*.
- [13] BOULT, T. E. and GOTTESFELD BROWN, L. (1991). Factorization-based segmentation of motions. In *Proceedings of the IEEE Workshop on Visual Motion* 179–186.

- [14] BRADLEY, P. S. and MANGASARIAN, O. L. (2000).  $k$ -plane clustering. *J. Global Optim.* **16** 23–32. [MR1770524](#)
- [15] CANDÈS, E. and TAO, T. (2007). The Dantzig selector: Statistical estimation when  $p$  is much larger than  $n$ . *Ann. Statist.* **35** 2313–2351. [MR2382644](#)
- [16] CHEN, G. and LERMAN, G. (2009). Foundations of a multi-way spectral clustering framework for hybrid linear modeling. *Found. Comput. Math.* **9** 517–558. [MR2534403](#)
- [17] CHEN, G. and LERMAN, G. (2009). Spectral curvature clustering (SCC). *Int. J. Comput. Vis.* **81** 317–330.
- [18] CHEN, Y., NASRABADI, N. M. and TRAN, T. D. (2011). Hyperspectral image classification using dictionary-based sparse representation. *IEEE Trans. Geosci. Remote Sens.* **99** 1–13.
- [19] DALALYAN, A. and CHEN, Y. (2012). Fused sparsity and robust estimation for linear models with unknown variance. In *Advances in Neural Information Processing Systems* **25** 1268–1276.
- [20] ELHAMIFAR, E. and VIDAL, R. (2009). Sparse subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR* 2790–2797.
- [21] ELHAMIFAR, E. and VIDAL, R. (2010). Clustering disjoint subspaces via sparse representation. In *IEEE International Conference on Acoustics Speech and Signal Processing, ICASSP* 1926–1929. IEEE Press, New York.
- [22] ELHAMIFAR, E. and VIDAL, R. (2013). Sparse subspace clustering: Algorithms, theory, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.* **35** 2765–2781.
- [23] ERIKSSON, B., BALZANO, L. and NOWAK, R. (2011). High-rank matrix completion and subspace clustering with missing data. Preprint. Available at [arXiv:1112.5629](#).
- [24] GIRAUD, C., HUET, S. and VERZELEN, N. (2012). High-dimensional regression with unknown variance. *Statist. Sci.* **27** 500–518. [MR3025131](#)
- [25] GOH, A. and VIDAL, R. (2007). Segmenting motions of different types by unsupervised manifold clustering. In *IEEE International Conference on Computer Vision and Pattern Recognition, CVPR* 1–6. IEEE Press, New York.
- [26] GUNNEMANN, S., MULLER, E., RAUBACH, S. and SEIDL, T. (2011). Flexible fault tolerant subspace clustering for data with missing values. In *IEEE International Conference on Data Mining, ICDM* 231–240.
- [27] KANNAN, R. and VEMPALA, S. (2008). Spectral algorithms. *Found. Trends Theor. Comput. Sci.* **4** 157–288 (2009). [MR2558901](#)
- [28] KELLER, F., MULLER, E. and BOHM, K. (2012). HICS: High contrast subspaces for density-based outlier ranking. In *IEEE International Conference on Data Engineering, ICDE* 1037–1048.
- [29] KOTROPOULOS, Y. P. C. and ARCE, G. R. (2011).  $\ell_1$ -graph based music structure analysis. In *International Society for Music Information Retrieval Conference, ISMIR*.
- [30] LERMAN, G. and ZHANG, T. (2011). Robust recovery of multiple subspaces by geometric  $l_p$  minimization. *Ann. Statist.* **39** 2686–2715. [MR2906883](#)
- [31] LIU, G., LIN, Z., YAN, S., SUN, J., YU, Y. and MA, Y. (2013). Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **35** 171–184.
- [32] LOH, P.-L. and WAINWRIGHT, M. J. (2012). High-dimensional regression with noisy and missing data: Provable guarantees with nonconvexity. *Ann. Statist.* **40** 1637–1664. [MR3015038](#)
- [33] LU, L. and VIDAL, R. (2006). Combined central and subspace clustering for computer vision applications. In *Proceedings of the 23rd International Conference on Machine Learning* 593–600. ACM, New York.
- [34] MA, Y., DERKSEN, H., HONG, W. and WRIGHT, J. (2007). Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE Trans. Pattern Anal. Mach. Intell.* **29** 1546–1562.

- [35] MA, Y. and VIDAL, R. (2005). Identification of deterministic switched arx systems via identification of algebraic varieties. In *Hybrid Systems: Computation and Control* 449–465.
- [36] MA, Y., YANG, A. Y., DERKSEN, H. and FOSSUM, R. (2008). Estimation of subspace arrangements with applications in modeling and segmenting mixed data. *SIAM Rev.* **50** 413–458. [MR2429444](#)
- [37] MCWILLIAMS, B. and MONTANA, G. (2014). Subspace clustering of high-dimensional data: A predictive approach. *Data Min. Knowl. Discov.* **28** 736–772. [MR3165525](#)
- [38] MEINSHAUSEN, N. and BÜHLMANN, P. (2006). High-dimensional graphs and variable selection with the lasso. *Ann. Statist.* **34** 1436–1462. [MR2278363](#)
- [39] MÜLLER, E., GUNNEMANN, S., ASSENT, I. and SEIDL, T. (2009). Evaluating clustering in subspace projections of high dimensional data. *Proc. VLDB Endow.* **2** 1270–1281.
- [40] NG, A. Y., JORDAN, M. I. and WEISS, Y. (2002). On spectral clustering: Analysis and an algorithm. *Adv. Neural Inf. Process. Syst.* **2** 849–856.
- [41] OZAY, N., SZNAIER, M. and LAGOA, C. (2010). Model (in) validation of switched arx systems with unknown switches and its application to activity monitoring. In *IEEE Conference on Decision and Control, CDC* 7624–7630.
- [42] OZAY, N., SZNAIER, M., LAGOA, C. and CAMPS, O. (2010). GPCA with denoising: A moments-based convex approach. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR* 3209–3216. IEEE Press, New York.
- [43] PARSONS, L., HAQUE, E. and LIU, H. (2004). Subspace clustering for high dimensional data: A review. *ACM SIGKDD Explor. Newsl.* **6** 90–105.
- [44] ROSENBAUM, M. and TSYBAKOV, A. B. (2013). Improved matrix uncertainty selector. In *From Probability to Statistics and Back: High-Dimensional Models and Processes—A Festschrift in Honor of Jon A. Wellner* 276–290. IMS, Beachwood, OH.
- [45] ROSENBAUM, M. and TSYBAKOV, A. B. (2010). Sparse recovery under matrix uncertainty. *Ann. Statist.* **38** 2620–2651. [MR2722451](#)
- [46] SOLTANOLKOTABI, M. and CANDÈS, E. J. (2012). A geometric analysis of subspace clustering with outliers. *Ann. Statist.* **40** 2195–2238. [MR3059081](#)
- [47] SOLTANOLKOTABI, M., ELHAMIFAR, E. and CANDÈS, E. J. (2014). Supplement to “Robust subspace clustering.” DOI:10.1214/13-AOS1199SUPP.
- [48] STÄDLER, N., BÜHLMANN, P. and VAN DE GEER, S. (2010).  $\ell_1$ -penalization for mixture regression models. *TEST* **19** 209–256. [MR2677722](#)
- [49] SUN, T. and ZHANG, C.-H. (2012). Scaled sparse linear regression. *Biometrika* **99** 879–898. [MR2999166](#)
- [50] TIPPING, M. E. and BISHOP, C. M. (1999). Probabilistic principal component analysis. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **61** 611–622. [MR1707864](#)
- [51] TOMASI, C. and KANADE, T. (1992). Shape and motion from image streams under orthography: A factorization method. *Int. J. Comput. Vis.* **9** 137–154.
- [52] TSENG, P. (2000). Nearest  $q$ -flat to  $m$  points. *J. Optim. Theory Appl.* **105** 249–252. [MR1757267](#)
- [53] VIDAL, R. (2011). Subspace clustering. *IEEE Signal Process. Mag.* **28** 52–68.
- [54] VIDAL, R., MA, Y. and SASTRY, S. (2005). Generalized principal component analysis (GPCA). *IEEE Trans. Pattern Anal. Mach. Intell.* **27** 1945–1959.
- [55] YAN, J. and POLLEFEYS, M. (2006). A general framework for motion segmentation: Independent, articulated, rigid, nonrigid, degenerate and nondegenerate. In *ECCV 2006* 94–106.
- [56] ZHANG, A., FAWAZ, N., IOANNIDIS, S. and MONTANARI, A. (2012). Guess who rated this movie: Identifying users through subspace clustering. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence* 944–953.
- [57] ZHANG, T., SZLAM, A. and LERMAN, G. (2009). Median  $k$ -flats for hybrid linear modeling with many outliers. In *IEEE International Conference on Computer Vision Workshops, ICCV* 234–241.

- [58] ZHANG, T., SZLAM, A., WANG, Y. and LERMAN, G. (2012). Hybrid linear modeling via local best-fit flats. *Int. J. Comput. Vis.* **100** 217–240. [MR2979307](#)
- [59] ZHOU, F., TORRE, F. and HODGINS, J. K. (2008). Aligned cluster analysis for temporal segmentation of human motion. In *IEEE International Conference on Automatic Face and Gesture Recognition, FG* 1–7.
- [60] [www.stanford.edu/~mahdisol/RSC](http://www.stanford.edu/~mahdisol/RSC).
- [61] [users.ece.gatech.edu/~sasif/homotopy](http://users.ece.gatech.edu/~sasif/homotopy).

M. SOLTANOLKOTABI  
DEPARTMENT OF ELECTRICAL ENGINEERING  
STANFORD UNIVERSITY  
350 SERRA MALL  
STANFORD, CALIFORNIA 94305  
USA  
E-MAIL: [mahdisol@stanford.edu](mailto:mahdisol@stanford.edu)

E. ELHAMIFAR  
EECS DEPARTMENT  
TRUST CENTER ROOM 337  
CORY HALL ENGINEERING DEPARTMENT  
UNIVERSITY OF CALIFORNIA  
BERKELEY, CALIFORNIA 94720-1774  
USA  
E-MAIL: [ehsan@eecs.berkeley.edu](mailto:ehsan@eecs.berkeley.edu)

E. J. CANDÈS  
DEPARTMENT OF STATISTICS  
STANFORD UNIVERSITY  
390 SERRA MALL  
STANFORD, CALIFORNIA 94305  
USA  
E-MAIL: [candes@stanford.edu](mailto:candes@stanford.edu)