Innovation Pursuit: A New Approach to the Subspace Clustering Problem

Mostafa Rahmani [1] **George Atia** [1]

Abstract

This paper presents a new scalable approach, termed Innovation Pursuit (iPursuit), to the problem of subspace clustering. iPursuit rests on a new geometrical idea whereby each subspace is identified based on its novelty with respect to the other subspaces. The subspaces are identified consecutively by solving a series of simple linear optimization problems, each searching for a direction of innovation in the span of the data. A detailed mathematical analysis is provided establishing sufficient conditions for the proposed approach to correctly cluster the data points. Moreover, the proposed direction search approach can be integrated with spectral clustering to yield a new variant of spectral-clustering-based algorithms. Remarkably, the proposed approach can provably yield exact clustering even when the subspaces have significant intersections. The numerical simulations demonstrate that iPursuit can often outperform the state-of-the-art subspace clustering algorithms – more so for subspaces with significant intersections – along with substantial reductions in computational complexity.

1. Introduction

Principal Component Analysis (PCA) is a popular and efficient procedure to approximate the data with a single low dimensional subspace (Lerman et al., 2015). Nonetheless, in numerous contemporary applications the data points may originate from multiple independent sources, in which case a union of subspaces can better model the data (Vidal, 2011). The problem of subspace clustering is concerned with learning these low-dimensional subspaces and clustering the data points to their respective subspaces, generally without prior knowledge about the number of subspaces and their dimensions, nor the membership of the data points to these subspaces. Subspace clustering naturally arises in many machine learning and data analysis problems, including computer vision (e.g. motion segmentation (Vidal et al., 2008), face clustering (Ho et al., 2003)), image processing (Yang et al., 2008) and system identification (Vidal et al., 2003). Numerous approaches for subspace clustering have been studied in prior work, including statistical-based approaches (Yang et al., 2006; Rao et al., 2010), spectral clustering (Soltanolkotabi et al., 2012; Von Luxburg, 2007; Dyer et al., 2013; Elhamifar & Vidal, 2013; Heckel & Bölcskei, 2013; Liu et al., 2013; Chen & Lerman, 2009), the algebraic-geometric approach (Vidal et al., 2005) and iterative methods (Bradley & Mangasarian, 2000). We refer the reader to (Vidal, 2011) for a comprehensive survey on the topic.

This paper aims to advance the state-of-the-art research on subspace clustering on several fronts. First, the proposed approach – termed iPursuit – rests on a novel geometrical idea whereby one subspace is identified at a time based on its novelty with respect to (w.r.t.) the other subspaces. Second, the proposed method is a provable and scalable subspace clustering algorithm – the computational complexity of iPursuit only scales linearly in the number of subspaces and quadratically in their dimensions (c.f. Section 2.5). In contrast to the spectral-clustering-based algorithms such as (Dyer et al., 2013; Elhamifar & Vidal, 2013; Liu et al., 2013), which need to solve an M_2^2-dimensional optimization problem to build the similarity matrix (where M_2 is the number of data points), the proposed method requires solving few M_2-dimensional linear optimization problems. This feature makes iPursuit remarkably faster than the state-of-the-art algorithms. Third, innovation pursuit in the data span enables superior performance when the subspaces have considerable intersections in comparison to the state-of-the-art subspace clustering algorithms.

1.1. Notation and definitions

Given a matrix $\mathbf{A}$, $\|\mathbf{A}\|$ denotes its spectral norm. For a vector $\mathbf{a}$, $\|\mathbf{a}\|$ denotes its ℓ_2-norm and $\|\mathbf{a}\|_1$ its ℓ_1-norm. Given matrices $\{\mathbf{A}_i\}_{i=1}^n$ with equal number of rows, we use the union symbol $\cup$ to define the matrix $\overset{n}{\underset{i=1}{\cup}}\mathbf{A}_i := [\mathbf{A}_1 \;\; \mathbf{A}_2 \;\; ... \;\; \mathbf{A}_n]$ as the concatenation of the matrices $\{\mathbf{A}_i\}_{i=1}^n$. For a matrix $\mathbf{D}$, we overload the set member-

[1] University of Central Florida, Orlando, Florida, USA. Correspondence to: Mostafa Rahmani <mostafa@knights.ucf.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

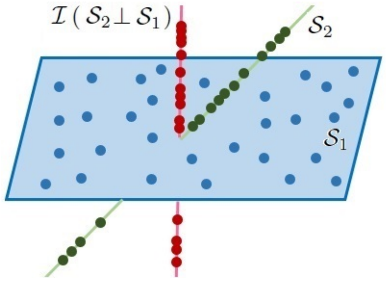

Figure 1. The subspace $\mathcal{I}(\mathcal{S}_2 \perp \mathcal{S}_1)$ is the innovation subspace of $\mathcal{S}_2$ over $\mathcal{S}_1$. The subspace $\mathcal{I}(\mathcal{S}_2 \perp \mathcal{S}_1)$ is orthogonal to $\mathcal{S}_1$.

ship operator by using the notation $\mathbf{d} \in \mathbf{D}$ to signify that $\mathbf{d}$ is a column of $\mathbf{D}$. A collection of subspaces $\{\mathcal{G}_i\}_{i=1}^n$ is said to be independent if $\dim\left(\overset{n}{\underset{i=1}{\oplus}} \mathcal{G}_i\right) = \sum_{i=1}^n \dim(\mathcal{G}_i)$, where $\oplus$ denotes the direct sum operator and $\dim(\mathcal{G}_i)$ is the dimension of $\mathcal{G}_i$. Given a vector $\mathbf{a}$, $|\mathbf{a}|$ is the vector of absolute values of the elements of $\mathbf{a}$. For a real number a, $\text{sgn}(a)$ denotes the sign of a. The complement of a set $\mathcal{L}$ is denoted $\mathcal{L}^c$. For any positive integer n, the index set $\{1, \ldots, n\}$ is denoted $[n]$.

Consider two subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$, such that $\mathcal{S}_2 \not\subseteq \mathcal{S}_1$ and $\mathcal{S}_1 \not\subseteq \mathcal{S}_2$. This means that each of the subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$ carries some innovation w.r.t. the other. As such, corresponding to each subspace we define an innovation subspace capturing its novelty (innovation) w.r.t. the other subspaces, defined formally as follows.

Definition 1. *Assume that $\mathbf{V}_1$ and $\mathbf{V}_2$ are two orthonormal bases for $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. We define the innovation subspace of $\mathcal{S}_2$ over $\mathcal{S}_1$, denoted $\mathcal{I}(\mathcal{S}_2 \perp \mathcal{S}_1)$, as the subspace spanned by $\left(\mathbf{I} - \mathbf{V}_1\mathbf{V}_1^T\right)\mathbf{V}_2$. In other words, $\mathcal{I}(\mathcal{S}_2 \perp \mathcal{S}_1)$ is the complement of $\mathcal{S}_1$ in the subspace $\mathcal{S}_1 \oplus \mathcal{S}_2$.*

Similarly, we can also define $\mathcal{I}(\mathcal{S}_1 \perp \mathcal{S}_2)$ as the innovation subspace of $\mathcal{S}_1$ over $\mathcal{S}_2$. Fig. 1 illustrates a scenario in which the data lies in a union of a two-dimensional and a one-dimensional subspace. Note that the innovation subspace of $\mathcal{S}_2$ over $\mathcal{S}_1$ is orthogonal to $\mathcal{S}_1$ and is the complement of $\mathcal{S}_1$ in $\mathcal{S}_1 \oplus \mathcal{S}_2$.

2. Proposed Approach

In this section, the main geometrical idea underlying iPursuit is first presented. This idea is based on a non-convex ℓ_0-norm minimization problem searching for a direction of innovation in the span of the data. Then, we provide a convex relaxation to a linear optimization problem, whose solution is shown to yield the correct subspaces under mild sufficient conditions. Due to space limitations, the proofs of all the theoretical results are deferred to an extended version of this paper (Rahmani & Atia, 2015). It is assumed that the given data matrix follows the following data model.

Data Model 1. The data matrix $\mathbf{D} \in \mathbb{R}^{M_1 \times M_2}$ can be represented as $\mathbf{D} = [\mathbf{D}_1 \ ... \ \mathbf{D}_N]\mathbf{T}$, where $\mathbf{T}$ is an arbitrary permutation matrix. The columns of $\mathbf{D}_i \in \mathbb{R}^{M_1 \times n_i}$ lie in $\mathcal{S}_i$, where $\mathcal{S}_i$ is an r_i-dimensional linear subspace, for $1 \leq i \leq N$, and, $\sum_{i=1}^N n_i = M_2$. Define $\mathbf{V}_i$ as an orthonormal basis for $\mathcal{S}_i$. In addition, define $\mathcal{D}$ as the space spanned by the data, i.e., $\mathcal{D} = \overset{N}{\underset{i=1}{\oplus}} \mathcal{S}_i$. Moreover, it is assumed that every subspace in the set of subspaces $\{\mathcal{S}_i\}_{i=1}^N$ has an innovation over the other subspaces, to say that, for $1 \leq i \leq N$, the subspace $\mathcal{S}_i$ does not completely lie in $\overset{N}{\underset{\substack{k=1 \\ k \neq i}}{\oplus}} \mathcal{S}_k$. In addition, the columns of $\mathbf{D}$ are normalized, i.e., each column has an ℓ_2-norm equal to one.

2.1. iPursuit: Basic idea

iPursuit is a multi-step algorithm that identifies one subspace at a time. In each step, the data is clustered into two subspaces. One subspace is the identified subspace and the other one is the direct sum of the other subspaces. The data points of the identified subspace are removed and the algorithm is applied to the remaining data to find the next subspace. Accordingly, each step of the algorithm can be interpreted as a subspace clustering problem with two subspaces. Therefore, for ease of exposition we first investigate the two-subspace scenario then extend the result to multiple (more than two) subspaces. Thus, in this subsection, it is assumed that the data follows Data model 1 with $N = 2$.

To gain some intuition, we first consider a simple example before stating our main result. Suppose that $\mathcal{S}_1$ and $\mathcal{S}_2$ are not orthogonal and assume that $n_2 < n_1$. The non-orthogonality of $\mathcal{S}_1$ and $\mathcal{S}_2$ is not a requirement, but is merely used herein to simplify the exposition of the basic idea underlying the proposed approach. Define $\mathbf{c}^*$ as the optimal point of the following optimization problem

$$\min_{\hat{\mathbf{c}}} \ \ \|\hat{\mathbf{c}}^T\mathbf{D}\|_0 \quad \text{subject to} \quad \hat{\mathbf{c}} \in \mathcal{D} \quad \text{and} \quad \|\hat{\mathbf{c}}\| = 1, \tag{1}$$

where $\|.\|_0$ is the ℓ_0-norm. The first constraint limits the search space to the span of the data, and the equality constraint $\|\hat{\mathbf{c}}\| = 1$ is used to avoid the trivial $\hat{\mathbf{c}} = 0$ solution. Assume that the columns of $\mathbf{D}_1$ and $\mathbf{D}_2$ are uniformly distributed in $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. Accordingly, with high probability (whp) the data is not aligned along any specific direction in $\mathcal{S}_1$ and $\mathcal{S}_2$.

The ℓ_0-norm minimization problem (1) searches for a nonzero vector in $\mathcal{D}$ that is orthogonal to the maximum number of data points. Since $\mathbf{c}^*$ has to lie in $\mathcal{D}$, we claim that the optimal point of (1) lies in $\mathcal{I}(\mathcal{S}_2 \perp \mathcal{S}_1)$ whp given the assumption that the number of data points in $\mathcal{S}_1$ is greater than the number of data points in $\mathcal{S}_2$. To clarify, consider the following cases:

I. If $\mathbf{c}^* \in \mathcal{S}_1$, then it would not be orthogonal to the majority of the data points in $\mathcal{S}_1$ given that the columns of $\mathbf{D}_1$ are uniformly distributed in $\mathcal{S}_1$. In addition, it cannot be orthogonal to most of the data points in $\mathcal{S}_2$ since $\mathcal{S}_1$ and $\mathcal{S}_2$ are not orthogonal. Since the optimal vector should be orthogonal to the maximum number of data points, it is highly likely that $\mathbf{c}^* \notin \mathcal{S}_1$. Similarly, it is highly unlikely that the optimal point lies in $\mathcal{S}_2$. **II.** If $\mathbf{c}^* \in \mathcal{I}(\mathcal{S}_1 \perp \mathcal{S}_2)$, then according to Definition 1, it is orthogonal to the data points in $\mathbf{D}_2$. However, since it was assumed that $n_2 < n_1$, the cost function of (1) can be decreased if $\mathbf{c}^*$ lies in $\mathcal{I}(\mathcal{S}_2 \perp \mathcal{S}_1)$ (which is orthogonal to $\mathcal{S}_1$). **III.** If $\mathbf{c}^*$ does not lie in any of the subspaces $\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{I}(\mathcal{S}_2 \perp \mathcal{S}_1)$ and $\mathcal{I}(\mathcal{S}_2 \perp \mathcal{S}_1)$, then it is neither orthogonal to $\mathcal{S}_1$ nor to $\mathcal{S}_2$. Now, since the data points are distributed uniformly in the subspaces, we see that $\mathbf{c}^*$ would not be orthogonal to the maximum number of data points.

Hence, it is highly likely that $\mathbf{c}^*$ lies in $\mathcal{I}(\mathcal{S}_2 \perp \mathcal{S}_1)$. It follows that the columns of $\mathbf{D}$ corresponding to the nonzero elements of $(\mathbf{c}^*)^T\mathbf{D}$ lie in $\mathcal{S}_2$. The following lemma ensures that these columns span $\mathcal{S}_2$.

Lemma 1. *The columns of* $\mathbf{D}$ *corresponding to the nonzero elements of* $(\mathbf{c}^*)^T\mathbf{D}$ *span* $\mathcal{S}_2$ *if the following conditions are satisfied:* ***(i)*** $\mathbf{c}^* \in \mathcal{I}(\mathcal{S}_2 \perp \mathcal{S}_1)$ *,* ***(ii)*** $\mathbf{D}_2$ *cannot follow Data model 1 with* $N > 1$*, that is, the data points in* $\mathbf{D}_2$ *do not lie in the union of lower dimensional subspaces within* $\mathcal{S}_2$ *each with innovation w.r.t. the other subspaces.*

It is important to note that the conditions of Lemma 1 are by no means restrictive. Specifically, if the requirement ***(ii)*** of Lemma 1 is not satisfied, then the problem can be viewed as a subspace clustering problem with more than two subspaces. In Section 2.4, we will investigate the clustering problem with more than two subspaces.

Remark 1. *At a high level, the innovation search problem (1) finds the most sparse vector in the row space of* $\mathbf{D}$*. Interestingly, finding the most sparse vector in a linear subspace has bearing on, and has been effectively used in, other machine learning problems, including dictionary learning and spectral estimation (Qu et al., 2014).*

2.2. Convex relaxation

The ℓ_1-norm is known to provide an efficient convex relaxation of the ℓ_0-norm. Thus, we relax the non-convex cost function and rewrite (1) as

$$\min_{\hat{\mathbf{c}}} \ \|\hat{\mathbf{c}}^T\mathbf{D}\|_1 \quad \text{subject to} \quad \hat{\mathbf{c}} \in \mathcal{D} \quad \text{and} \quad \|\hat{\mathbf{c}}\| = 1\,. \tag{2}$$

Since the feasible set of (2) is non-convex, we further substitute the equality constraint with a linear constraint to consider the following convex program

$$\text{(IP)} \quad \min_{\hat{\mathbf{c}}} \ \|\hat{\mathbf{c}}^T\mathbf{D}\|_1 \quad \text{s. t.} \quad \hat{\mathbf{c}} \in \mathcal{D} \quad \text{and} \quad \hat{\mathbf{c}}^T\mathbf{q} = 1. \tag{3}$$

(IP) is the core program of iPursuit to find a direction of innovation. The vector $\mathbf{q}$ is a unit ℓ_2-norm vector that should not be orthogonal to $\mathcal{D}$. In Section 2.6, we present a methodology to obtain a good choice for the vector $\mathbf{q}$ from the given data. However, our investigations have shown that iPursuit performs well generally even when $\mathbf{q}$ is chosen as a random vector in $\mathcal{D}$.

2.3. Segmentation of two subspaces: Performance guarantees

Suppose that $\mathbf{D}$ follows Data model 1 with $N = 2$, i.e., the data lies in a union of two subspaces. In order to show that the optimal point of (IP) yields correct clustering, it suffices to show that the optimal point lies in $\mathcal{I}(\mathcal{S}_2 \perp \mathcal{S}_1)$ given that condition ***(ii)*** of Lemma 1 is satisfied for $\mathbf{D}_2$ (or lies in $\mathcal{I}(\mathcal{S}_1 \perp \mathcal{S}_2)$ given that the condition is satisfied for $\mathbf{D}_1$). The following theorem provides sufficient conditions for the optimal point of (3) to lie in $\mathcal{I}(\mathcal{S}_2 \perp \mathcal{S}_1)$ provided that

$$\inf_{\substack{\mathbf{c} \in \mathcal{I}(\mathcal{S}_2 \perp \mathcal{S}_1) \\ \mathbf{c}^T\mathbf{q}=1}} \|\mathbf{c}^T\mathbf{D}\|_1 < \inf_{\substack{\mathbf{c} \in \mathcal{I}(\mathcal{S}_1 \perp \mathcal{S}_2) \\ \mathbf{c}^T\mathbf{q}=1}} \|\mathbf{c}^T\mathbf{D}\|_1. \tag{4}$$

If the inequality in (4) is reversed, then similar sufficient conditions can be established for the optimal point of (3) to lie in $\mathcal{I}(\mathcal{S}_1 \perp \mathcal{S}_2)$. Hence, assumption (4) does not lead to any loss of generality. The subspaces $\mathcal{I}(\mathcal{S}_2 \perp \mathcal{S}_1)$ and $\mathcal{I}(\mathcal{S}_1 \perp \mathcal{S}_2)$ are orthogonal to $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. Thus, (4) is equivalent to

$$\inf_{\substack{\mathbf{c} \in \mathcal{I}(\mathcal{S}_2 \perp \mathcal{S}_1) \\ \mathbf{c}^T\mathbf{q}=1}} \|\mathbf{c}^T\mathbf{D}_2\|_1 < \inf_{\substack{\mathbf{c} \in \mathcal{I}(\mathcal{S}_1 \perp \mathcal{S}_2) \\ \mathbf{c}^T\mathbf{q}=1}} \|\mathbf{c}^T\mathbf{D}_1\|_1. \tag{5}$$

Henceforth, "innovation subspace" refers to $\mathcal{I}(\mathcal{S}_2 \perp \mathcal{S}_1)$ whenever the two-subspace scenario is considered and (5) is satisfied. Theorem 2 stated next provides sufficient conditions for the optimal point of (IP) in (3) to lie in $\mathcal{I}(\mathcal{S}_2 \perp \mathcal{S}_1)$. These conditions are characterized in terms of the optimal solution to an oracle optimization problem (OP), wherein the feasible set of (IP) is replaced by the innovation subspace. Define $\mathbf{c}_2$ as the optimal point of the following optimization problem

$$\text{(OP)} \quad \begin{aligned} &\min_{\hat{\mathbf{c}}} \quad \|\hat{\mathbf{c}}^T\mathbf{D}_2\|_1 \\ &\text{subject to} \quad \hat{\mathbf{c}} \in \mathcal{I}(\mathcal{S}_2 \perp \mathcal{S}_1) \ \text{ and } \ \hat{\mathbf{c}}^T\mathbf{q} = 1. \end{aligned} \tag{6}$$

Theorem 2 establishes that $\mathbf{c}_2$ is the optimal point of (3). Before we state the theorem, we define the index set $\mathcal{L}_0 := \{i \in [n_2] : \mathbf{c}_2^T\mathbf{d}_i = 0, \mathbf{d}_i \in \mathbf{D}_2\}$, with cardinality $n_0 = |\mathcal{L}_0|$ and a complement set $\mathcal{L}_0^c$, comprising the indices of the columns of $\mathbf{D}_2$ orthogonal to $\mathbf{c}_2$.

Theorem 2. *Suppose the data matrix $\mathbf{D}$ follows Data model 1 with $N = 2$. Also, assume that condition (5) and the requirement of Lemma 1 for $\mathbf{D}_2$ are satisfied (condition **(ii)** of Lemma 1). Let $\mathbf{c}_2$ be the optimal point of the oracle (OP) in (6) and define $\alpha = \sum_{\substack{\mathbf{d}_i \in \mathbf{D}_2 \\ i \in \mathcal{L}_0^c}} \operatorname{sgn}(\mathbf{c}_2^T \mathbf{d}_i)\mathbf{d}_i$. Also, let $\mathbf{P}_2$ denote an orthonormal basis for $\mathcal{I}\left(\mathcal{S}_2 \perp \mathcal{S}_1\right)$ and assume that $\mathbf{q}$ is a unit ℓ_2-norm vector in $\mathcal{D}$ that is not orthogonal to $\mathcal{I}\left(\mathcal{S}_2 \perp \mathcal{S}_1\right)$. If*

$$\frac{1}{2} \inf_{\substack{\delta \in \mathcal{S}_1 \\ \|\delta\|=1}} \sum_{\mathbf{d}_i \in \mathbf{D}_1} \left|\delta^T \mathbf{d}_i\right| > \|\mathbf{V}_1^T \mathbf{V}_2\| \Big(\|\alpha\| + n_0 \Big) \,, \ \textit{and}$$
$$\frac{\|\mathbf{q}^T \mathbf{P}_2\|}{2\|\mathbf{q}^T \mathbf{V}_1\|} \left(\inf_{\substack{\delta \in \mathcal{S}_1 \\ \|\delta\|=1}} \sum_{\mathbf{d}_i \in \mathbf{D}_1} \left|\delta^T \mathbf{d}_i\right| \right) > \|\mathbf{V}_2^T \mathbf{P}_2\| \Big(\|\alpha\| + n_0 \Big), \tag{7}$$

then $\mathbf{c}_2 \in \mathcal{I}\left(\mathcal{S}_2 \perp \mathcal{S}_1\right)$ is the optimal point of (IP) in (3), and iPursuit clusters the data correctly.

In what follows, we provide a detailed discussion of the significance of the sufficient conditions (7) of Theorem 2, which reveal some interesting facts about the properties of iPursuit.

1. *Data distribution:* The LHS of (7) is known as the permeance statistic, an efficient measure of how well the data points are distributed in a subspace (Lerman et al., 2015). As such, the sufficient conditions (7) imply that the distribution and the number of the data points within the subspaces are important performance factors for iPursuit. For a set of data points $\mathbf{D}_i$ in a subspace $\mathcal{S}_i$, the permeance statistic is defined as $\mathcal{P}(\mathbf{D}_i, \mathcal{S}_i) = \inf_{\substack{\mathbf{u} \in \mathcal{S}_i \\ \|\mathbf{u}\|=1}} \sum_{\mathbf{d}_i \in \mathbf{D}_i} \left|\mathbf{u}^T \mathbf{d}_i\right|$. From this definition, we see that the permeance statistic is fairly small if a set of data points are aligned along a given direction (i.e. not well distribued). In addition, having n_0 on the RHS reveals that the distribution of the data points within $\mathcal{S}_2$ also matters since $\mathbf{c}_2$ cannot be simultaneously orthogonal to a large number of columns of $\mathbf{D}_2$ if the data does not align along specific directions. Hence, according to (7), iPursuit yields correct clustering if the data is well distributed within the subspaces.

2. *The coherency of $\mathbf{q}$ with $\mathcal{I}\left(\mathcal{S}_2 \perp \mathcal{S}_1\right)$:* For the optimal point of (IP) to lie in $\mathcal{I}\left(\mathcal{S}_2 \perp \mathcal{S}_1\right)$, the vector $\mathbf{q}$ should not be too coherent with $\mathcal{S}_1$. This can be seen by observing that if $\mathbf{q}$ has a small projection on $\mathcal{I}\left(\mathcal{S}_2 \perp \mathcal{S}_1\right)$ – in which case it would be more coherent with $\mathcal{S}_1$ – the Euclidean norm of any feasible point of (3) lying in $\mathcal{I}\left(\mathcal{S}_2 \perp \mathcal{S}_1\right)$ will have to be large to satisfy the equality constraint in (IP). Such points are less likely to be optimal in the sense of attaining the minimum of the objective function in (3). The aforementioned intuition is affirmed by the analysis. Indeed, the factor $\|\mathbf{q}^T \mathbf{P}_2\| / \|\mathbf{q}^T \mathbf{V}_1\|$ in the sufficient conditions of Theorem 2 and Lemma 3 indicates that the coherency of $\mathbf{q}$ with the innovation subspace is an important performance factor for iPursuit. The coherence property could have a more serious effect on the performance of the algorithm for non-independent subspaces, especially when the dimension of their intersection is significant. For instance, consider the scenario where the vector $\mathbf{q}$ is chosen randomly from $\mathcal{D}$, and define y as the dimension of the intersection of $\mathcal{S}_1$ and $\mathcal{S}_2$. It follows that $\mathcal{I}\left(\mathcal{S}_2 \perp \mathcal{S}_1\right)$ has dimension $r_2 - y$. Thus, $\frac{\mathbb{E}[\|\mathbf{q}^T \mathbf{P}_2\|]}{\mathbb{E}[\|\mathbf{q}^T \mathbf{V}_1\|]} = \frac{r_2 - y}{r_1}$. Therefore, a randomly chosen vector $\mathbf{q}$ is likely to have a small projection on the innovation subspace when y is large. As such, in dealing with subspaces with significant intersection, it may not be favorable to choose the vector $\mathbf{q}$ at random. In Section 2.6 and section 2.7, we develop a simple technique to learn a good choice for $\mathbf{q}$ from the given data. It makes iPursuit remarkably powerful in dealing with subspaces with intersection as shown in the numerical results section.

Remark 2. *We conjecture that if the ratios $\{n_i/r_i\}_{i=1}^N$ are sufficiently large, the optimal point of (2) always lies in an innovation subspace; our investigations have shown that the optimal point of (IP) always lies in an innovation subspace provided that $\mathbf{q}$ is sufficiently coherent with the innovation subspace regardless how large the intersections between the subspaces are. This is particularly compelling if the subspaces are too close, in which case spectral-clustering-based methods can seldom construct a correct similarity matrix (which leads to large clustering errors). By contrast, in such cases iPursuit can yield accurate clustering provided that the constraint vector is sufficiently coherent with the innovation subspace. In Section 2.6, we present a method to identify a coherent constraint vector.*

Now, we demonstrate that the sufficient conditions (7) are not restrictive. The following lemma simplifies the conditions when the data points are randomly distributed in the subspaces. In this setting, we show that the conditions are naturally satisfied.

Lemma 3. *Assume that the distribution of the columns of $\mathbf{D}_1$ in $\mathcal{S}_1$ and $\mathbf{D}_2$ in $\mathcal{S}_2$ is uniformly random and consider the same setup of Theorem 2. If*

$$\begin{aligned} &\sqrt{\frac{2}{\pi}}\frac{n_1}{r_1} - 2\sqrt{n_1} - t_1\sqrt{\frac{n_1}{r_1 - 1}} \\ &\qquad\qquad > 2\|\mathbf{V}_1^T \mathbf{V}_2\| \Big(t_2\sqrt{n_2 - n_0} + n_0 \Big), \\ &\frac{\|\mathbf{q}^T \mathbf{P}_2\|}{\|\mathbf{q}^T \mathbf{V}_1\|} \left(\sqrt{\frac{2}{\pi}}\frac{n_1}{r_1} - 2\sqrt{n_1} - t_1\sqrt{\frac{n_1}{r_1 - 1}} \right) \\ &\qquad\qquad > 2\|\mathbf{V}_2^T \mathbf{P}_2\| \Big(t_2\sqrt{n_2 - n_0} + n_0 \Big) \,, \end{aligned} \tag{8}$$

then the optimal point of (3) lies in $\mathcal{I}\left(\mathcal{S}_2 \perp \mathcal{S}_1\right)$ with probability at least $1 - \exp\left(-\frac{r_2}{2}(t_2^2 - \log(t_2^2) - 1)\right) - \exp\left(-\frac{t_1^2}{2}\right)$, for all $t_2 > 1$, $t_1 \geq 0$.

When the data points are not aligned along any specific directions, $\mathbf{c}_2$ can only be simultaneously orthogonal to a small number of columns of $\mathbf{D}_2$. Thus, n_0 will be much smaller than n_2. The LHS of (8) has order n_1 and the RHS has order $\sqrt{n_2} + n_0$ (which is much smaller than n_2). Therefore, the sufficient conditions are naturally satisfied when the data is well distributed within the subspaces.

2.4. Clustering multiple subspaces

Suppose that $\mathbf{D}$ follows Data Model 1 with $N = m$ where $m > 2$. Similar to (5), without loss of generality assume that

$$\inf_{\substack{\mathbf{c} \in \mathcal{I}\left(\mathcal{S}_m \perp \overset{m}{\underset{\substack{k=1 \\ k \neq m}}{\oplus}} \mathcal{S}_k\right) \\ \mathbf{c}^T \mathbf{q}_m = 1}} \|\mathbf{c}^T \mathbf{D}_m\|_1 < \inf_{\substack{\mathbf{c} \in \mathcal{I}\left(\mathcal{S}_j \perp \overset{m}{\underset{\substack{k=1 \\ k \neq j}}{\oplus}} \mathcal{S}_k\right) \\ \mathbf{c}^T \mathbf{q}_m = 1}} \|\mathbf{c}^T \mathbf{D}_j\|_1 \quad (9)$$

for all $1 \leq j \leq m - 1$, where $\mathbf{q}_m$ is a unit ℓ_2-norm in $\oplus_{k=1}^{m} \mathcal{S}_k$. Given (9), we expect the optimal point of (IP) with $\mathbf{q} = \mathbf{q}_m$ to lie in the innovation subspace of $\mathcal{S}_m$ over $\overset{m}{\underset{\substack{k=1 \\ k \neq m}}{\oplus}} \mathcal{S}_k$, i.e., $\mathcal{I}\big(\mathcal{S}_m \perp \overset{m}{\underset{\substack{k=1 \\ k \neq m}}{\oplus}} \mathcal{S}_k\big)$, in which case $\mathcal{S}_m$ will be identified. Accordingly, in this step the clustering problem separates $(\mathbf{D}_m, \mathcal{S}_m)$ and $(\overset{m-1}{\underset{i=1}{\cup}} \mathbf{D}_i \, , \overset{m-1}{\underset{i=1}{\oplus}} \mathcal{S}_i)$. Theorem 2 can be used to derive sufficient conditions for the optimal point to lie in $\mathcal{I}\big(\mathcal{S}_m \perp \overset{m}{\underset{\substack{k=1 \\ k \neq m}}{\oplus}} \mathcal{S}_k\big)$ by substituting $(\mathbf{D}_2, \mathcal{S}_2)$ with $(\mathbf{D}_m, \mathcal{S}_m)$ and $(\mathbf{D}_1, \mathcal{S}_1)$ with $(\overset{m-1}{\underset{i=1}{\cup}} \mathbf{D}_i \, , \overset{m-1}{\underset{i=1}{\oplus}} \mathcal{S}_i)$. Consequently, we can also establish sufficient conditions for exact subspace segmentation. Due to space limitations, we defer the analysis to (Rahmani & Atia, 2015).

2.5. Complexity analysis

Define $\mathbf{U}$ as an orthonormal basis for $\mathcal{D}$. Thus, the optimization problem (3) is equivalent to $\min\limits_{\mathbf{a}} \|\mathbf{a}^T \mathbf{U}^T \mathbf{D}\|_1$ subject to $\mathbf{a}^T \mathbf{U}^T \mathbf{q} = 1$. Further, define $\mathbf{f} = \mathbf{U}^T \mathbf{q}$. This optimization problem can be efficiently solved using the Alternating Direction Method of Multipliers (ADMM) (Boyd et al., 2011). Due to space constraints, we defer the details of the iterative solver to (Rahmani & Atia, 2015). The complexity of the initialization step of the solver is $\mathcal{O}(r^3)$ plus the complexity of obtaining $\mathbf{U}$. Obtaining an appropriate $\mathbf{U}$ has $O(r^2 M_2)$ complexity by applying the clustering algorithm to a random subset of the rows of $\mathbf{D}$ (with the rank of sampled rows equal to r). In addition, the complexity of each iteration of the solver is $\mathcal{O}(r M_2)$. Thus, the overall complexity is less than $\mathcal{O}((r^3 + r^2 M_2) N)$ since the number of data points remaining keeps decreasing over the iterations. In most cases, $r \ll M_2$, hence the overall complexity is roughly $\mathcal{O}(r^2 M_2 N)$.

Table 1. Run time of different algorithms ($M_1 = 50$, $N = 3$, $\{r_i\}_{i=1}^{3} = 10$, $\{n_i\}_{i=1}^{3} = M_2/3$)

M_2	SSC	LRR	iPursuit	SSC-OMP
300	1.1 s	0.9 s	0.14 s	1.6 s
3000	209 s	26 s	0.35 s	56 s
15000	> 2 h	2800	2.78 s	5340 s
30000	> 2 h	> 2 h	10.6 s	> 2 h

Remark 3. *The proposed method brings about substantial speedups over existing algorithms due to the following: i) unlike existing multi-step algorithms (such as RANSAC) which have exponential complexity in the number and dimension of subspaces, the complexity of iPursuit is linear in the number of subspaces and quadratic in their dimension. In addition, while iPursuit has linear complexity in M_2, spectral-clustering-based algorithms have complexity $\mathcal{O}(M_2^2 N)$ for their spectral clustering step plus the complexity of obtaining the similarity matrix; ii) more importantly, the solver of the proposed optimization problem has $\mathcal{O}(r M_2)$ complexity per iteration, while the other operations – whose complexity are $\mathcal{O}(r^2 M_2)$ and $\mathcal{O}(r^3)$ – sit outside of the iterative solver. This feature makes the proposed method notably faster than most of the existing algorithms which solve high-dimensional optimization problems. For instance, solving the optimization problem of the SSC algorithm has roughly $\mathcal{O}(M_2^3 + r M_2)$ complexity per iteration (Elhamifar & Vidal, 2013).*
For instance, suppose $M_1 = 100$, the data lies in a union of three 10-dimensional subspaces and $n_i = M_2/3$. Table 1 compares the running time of the algorithms. One can observe that iPursuit is remarkably faster. More running time comparisons are available in (Rahmani & Atia, 2015).

2.6. How to choose the vector q?

Our investigations have shown that iPursuit performs very well when the subspaces are independent or have small intersections *even if* $\mathbf{q}$ *is chosen randomly*. However, in the more challenging scenarios in which the dimensions of the intersections between the subspaces are significant, randomly choosing the vector $\mathbf{q}$ could be unfavorable since the dimension of the innovation subspace decreases as the dimension of the intersection increases. This motivates the methodology described next that aims to search for a "good" vector $\mathbf{q}$. Consider the optimization problem

$$\min_{\hat{\mathbf{q}}} \ \|\hat{\mathbf{q}}^T \mathbf{D}\|_2 \quad \text{subject to} \quad \hat{\mathbf{q}} \in \mathcal{D} \quad \text{and} \quad \|\hat{\mathbf{q}}\| = 1, \quad (10)$$

which searches for a non-zero vector in $\mathcal{D}$ with small projections on the columns of $\mathbf{D}$. It is straightforward to show that the optimal point of (10) is the singular vector corresponding to the least non-zero singular value of $\mathbf{D}$. When the subspaces are close to each other, it is not hard to see that the least singular vector is highly coherent with the innovation subspace, thus can be a good candidate for the vector $\mathbf{q}$. For subspaces with remarkable intersections, this choice of $\mathbf{q}$ brings about substantial improvement in performance compared to using a randomly generated $\mathbf{q}$ (cf. Section 3). Clearly, when the data is noisy, we utilize the least dominant singular vecor. In addition, when the singular values of the noisy data decay rapidly, it may be hard to accurately estimate the rank of $\mathbf{D}$, which may lead to an unfavorable use of a singular vector corresponding to noise as the constraint vector. Alternatively, we can choose the data point closest to the least dominant singular vector as our vector $\mathbf{q}$. This technique makes the proposed method robust to the presence of noise (cf. Section 2.7).

2.7. Noisy data

In the presence of additive noise, we model the data as $\mathbf{D}_e = \mathbf{D} + \mathbf{E}$, where $\mathbf{D}_e$ is the given noisy data matrix, $\mathbf{D}$ is the clean data which follows Data model 1 and $\mathbf{E}$ represents the noise component. The rank of $\mathbf{D}$ is equal to r. Thus, the singular values of $\mathbf{D}_e$ can be divided into two subsets: the dominant singular values (the first r singular values) and the small singular values (or the singular values corresponding to the noise component). Estimating the number of dominant singular values is a fairly well-studied topic (Stoica & Selen, 2004).

Consider the optimization problem (IP) using $\mathbf{D}_e$, i.e.,

$$\min_{\hat{\mathbf{c}}} \ \|\hat{\mathbf{c}}^T \mathbf{D}_e\|_1 \quad \text{s.t.} \quad \hat{\mathbf{c}} \in \text{span}(\mathbf{D}_e) \ \text{and} \ \hat{\mathbf{c}}^T \mathbf{q} = 1. \quad (11)$$

Clearly, the optimal point of (11) is very close to the subspace spanned by the singular vectors corresponding to the small singular values. Thus, if $\mathbf{c}_e$ denotes the optimal solution of (11), then all the elements of $\mathbf{c}_e^T \mathbf{D}_e$ will be fairly small and the subspaces cannot be distinguished. However, the span of the dominant singular vectors is approximately equal to $\mathcal{D}$. Accordingly, we propose the following approximation to (IP),

$$\min_{\hat{\mathbf{c}}} \ \|\hat{\mathbf{c}}^T \mathbf{D}_e\|_1 \quad \text{s.t.} \quad \hat{\mathbf{c}} \in \text{span}(\mathbf{Q}) \ \text{and} \ \hat{\mathbf{c}}^T \mathbf{q} = 1 \quad (12)$$

where $\mathbf{Q}$ is an orthonormal basis for the span of the dominant singular vectors. The first constraint of (12) forces the optimal point to lie in $\text{span}(\mathbf{Q})$, which serves as a good approximation to $\text{span}(\mathbf{D})$. For instance, consider $\mathbf{D} = [\mathbf{D}_1 \ \mathbf{D}_2]$, where the columns of $\mathbf{D}_1 \in \mathbb{R}^{40 \times 100}$ lie in a 5-dimensional subspace $\mathcal{S}_1$, and the columns of $\mathbf{D}_2 \in \mathbb{R}^{40 \times 100}$ lie in another 5-dimensional subspace $\mathcal{S}_2$. Define $\mathbf{c}_e$ and $\mathbf{c}_r$ as the optimal points of (11) and (12), respectively. Fig. 2 shows $|\mathbf{c}_e^T \mathbf{D}_e|$ and $|\mathbf{c}_r^T \mathbf{D}_e|$ with the maximum element scaled to one. Clearly, $\mathbf{c}_r^T \mathbf{D}_e$ can be used to correctly cluster the data. In addition, when $\mathbf{D}$ is low rank, the subspace constraint in (12) can filter out a remarkable portion of the noise component.

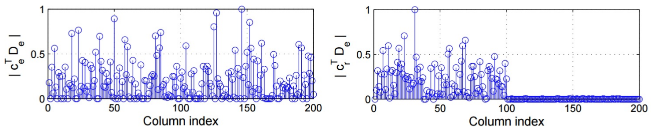

Figure 2. The left plot shows the output of (11), while the right plot shows the output of iPursuit when its search domain is restricted to the subspace spanned by the dominant singular vectors as per (12).

When the data is noisy and the singular values of $\mathbf{D}$ decay rapidly, it may be hard to accurately estimate r. If the dimension is incorrectly estimated, $\mathbf{Q}$ may contain some singular vectors corresponding to the noise component, wherefore the optimal point of (12) could end up lying close to a noise singular vector. In the sequel, we present two effective techniques to effectively avoid this undesirable scenario.

1. *Using a data point as a constraint vector:* A singular vector corresponding to the noise component is nearly orthogonal to the entire data, i.e., has small projection on all the data points. Thus, if the optimal vector is forced to have strong projection on a data point, it will be unlikely for the optimal direction to be close to a noise singular vector. Thus, we modify (12) as follows

$$\min_{\hat{\mathbf{c}}} \ \|\hat{\mathbf{c}}^T \mathbf{D}_e\|_1 \quad \text{s.t.} \quad \hat{\mathbf{c}} \in \text{span}(\mathbf{Q}) \ \text{and} \ \hat{\mathbf{c}}^T \mathbf{d}_{ek} = 1 \ , \quad (13)$$

where $\mathbf{d}_{ek}$ is the k^{th} column of $\mathbf{D}_e$. The modified constraint in (13) ensures that the optimal point is not orthogonal to $\mathbf{d}_{ek}$. If $\mathbf{d}_{ek}$ lies in the subspace $\mathcal{S}_i$, the optimal point of (13) will lie in the innovation subspace corresponding to $\mathcal{S}_i$ whp. In order to determine a good data point for the constraint vector, we leverage the principle presented in section 2.6. Specifically, we use the data point that is closest to the least dominant singular vector rather than the least dominant singular vector itself.

2. *Sparse representation of the optimal point:* When $\mathbf{D}$ is low rank, i.e., $r \ll \min(M_1, M_2)$, any direction in the span of the data – including the optimal direction sought by iPursuit – can be represented as a sparse combination of the data points. For such settings, we propose the alternative optimization problem

$$\begin{aligned} &\min_{\mathbf{a},\mathbf{z}} && \|\mathbf{a}^T \mathbf{Q}^T \mathbf{D}_e\|_1 + \gamma \|\mathbf{z}\|_1 \\ &\text{subject to} && \mathbf{a} = \mathbf{Q}^T \mathbf{D}_e \, \mathbf{z} \quad \text{and} \quad \mathbf{a}^T \mathbf{Q}^T \mathbf{d}_{ek} = 1 \ , \end{aligned} \quad (14)$$

where γ is a regularization parameter. Forcing a sparse representation in (14) for the optimal direction averts a solution that lies in close proximity with the small singular vectors, which are normally obtained through linear combinations of a large number of data points. This alternative formulation is particularly useful when the dimension of the data cannot be accurately estimated. When $\mathbf{D}$ is not a low rank matrix, we can set γ equal to zero. The table of Algorithm 1 details the proposed method for noisy data along with the used notation and definitions.

2.7.1. Error propagation

If κ (or c_i) and the threshold c_o in Algorithm 1 are chosen appropriately, the algorithm exhibits strong robustness in the presence of noise. Nonetheless, if the data is too noisy, an error incurred in one step of the algorithm may propagate and unfavorably affect the performance in subsequent steps. Two types of error could occur. The first type is that some data points are erroneously included in $\mathbf{G}_1$ or $\mathbf{G}_2$. An example is when $\mathcal{S}_m$ is the subspace to be identified in a given step of the algorithm (i.e., the optimal point of (13) lies in the innovation subspace corresponding to $\mathcal{S}_m$), but few data points from the other subspaces are erroneously included in $\mathbf{G}_1$. The second type of error is that some of the data points remain unidentified. For instance, $\mathcal{S}_m$ is to be identified in a given iteration, yet not all the data points belonging to $\mathcal{S}_m$ are identified. In an extended version of this work (Rahmani & Atia, 2015), we discuss the two main sources of error and present some techniques to effectively neutralize their impact on subsequent iterations. In addition, a brief discussion about handling the presence of outliers is provided.

2.8. Subspace clustering using direction search and spectral clustering

In (Rahmani & Atia, 2017), we showed that the direction search optimization problem (13) can be utilized to find a neighborhood set for the k^{th} data point. We leveraged this feature to propose a new spectral-clustering-based subspace segmentation algorithm, dubbed Direction search based Subspace Clustering (DSC) (Rahmani & Atia, 2017). We showed that DSC often outperforms the existing spectral-clustering-based methods particularly for hard scenarios involving high levels of noise and close subspaces, and notably improves the state-of-the-art result for the challenging problem of face segmentation using subspace clustering.

3. Numerical Simulations

3.1. The coherency parameter

In this experiment, we examine the impact of the coherency of $\mathbf{q}$ with the innovation subspace. Here, it is assumed that the data follows Data Model 1 with $N = 2$ and $M_1 = 50$. The dimension of the subspaces is equal to 15 and the dimension of their intersection varies between 0 to 14. Each subspace contains 100 data points distributed uniformly at random within the subspace. Let $c_r = \|\mathbf{q}^T \mathbf{P}_2\| / \|\mathbf{q}^T \mathbf{V}_1\|$. Thus, c_r captures the coherency of $\mathbf{q}$ with the innovation subspace. Define $\hat{\mathbf{V}}_1$ and $\hat{\mathbf{V}}_2$ as orthonormal bases for the identified subspaces. A trial is considered successful if

$$\|(\mathbf{I} - \mathbf{V}_1 \mathbf{V}_1^T)\hat{\mathbf{V}}_1\|_F + \|(\mathbf{I} - \mathbf{V}_2 \mathbf{V}_2^T)\hat{\mathbf{V}}_2\|_F \leq 10^{-3} \,. \quad (15)$$

The left plot of Fig. 3 shows the phase transition in the plane of c_r and y, where y is the dimension of the intersection of the two subspaces. In this figure, white designates exact identification of the subspaces with probability almost equal to one. As shown, the probability of correct clustering increases if c_r is increased. Remarkably, the left plot of Fig. 3 shows that when c_r is sufficiently large, the algorithm yields exact segmentation even when $y = 14$.

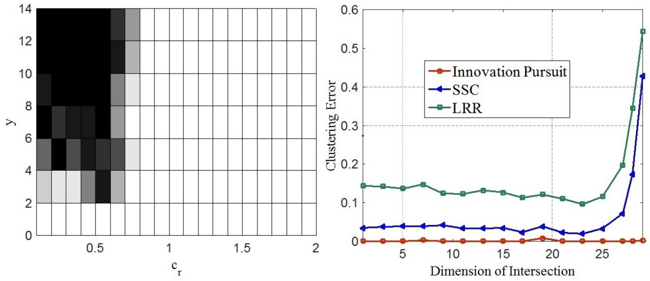

Figure 3. **Left**: Phase transition plot in the plane of the coherency parameter and the dimension of intersection. **Right**: The clustering error of iPursuit, SSC and LRR versus the dimension of the intersection.

3.2. Clustering data in union of multiple subspaces

In this simulation, we consider the subspace clustering problem with 15 30-dimensional subspaces $\{\mathcal{S}_i\}_{i=1}^{15}$ and $M_1 = 500$. Each subspace contains 90 data points and the distribution of the data within the subspaces is uniformly random. We compare the performance of the proposed approach to the state-of-the-art sparse subspace clustering (SSC) (Elhamifar & Vidal, 2013) algorithm and low rank representation (LRR) based clustering (Liu et al., 2013). The number of replicates used in spectral clustering for SSC and LRR is equal to 20. Define the clustering error as the ratio of misclassified points to the total number of data points. The right plot of Fig. 3 shows the clustering error versus the dimension of the intersection. The dimension of intersection varies between 1 and 29. Each point in the plot is obtained by averaging over 40 independent runs. iPursuit is shown to yield the best performance.

3.3. Noisy data

In this section, we study the performance of the proposed approach, SSC, LRR, SCC (Chen & Lerman, 2009), TSC (Heckel & Bölcskei, 2013) and SSC-OMP (Dyer et al.,

Table 2. CE (%) of algorithms on Hopkins155 dataset (Mean - Median).

N	SSC	LRR	iPursuit	SSC-OMP	TSC	K-flats	SCC
$N=2$	1.52 - 0	2.13 - 0	3.33 - 0.27	16.92 - 12.77 s	18.44 - 16.92	13.62 - 10.65	2.06 - 0
$N=3$	4.40 - 1.56 s	4.03 - 1.43	6.91 - 2.44	27.96 - 30.98	28.58 - 29.67	14.07 - 14.18	6.37 - 0.21

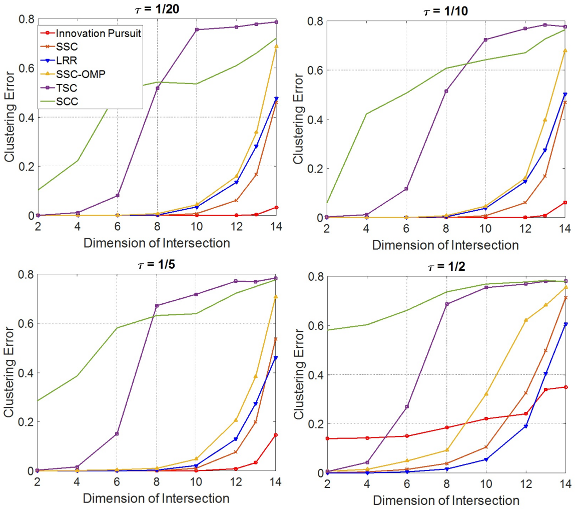

Figure 4. Performance of the algorithms versus the dimension of intersection for different noise levels.

2013) with different noise levels, and varying dimensions of the intersection between the subspaces, which gives rise to both low rank and high rank data matrices. It is assumed that $\mathbf{D}$ follows Data model 1 with $M_1 = 100$, $M_2 = 500$, $N = 6$ and $\{r_i\}_{i=1}^{6} = 15$. The dimension of the intersection between the subspaces varies from 0 to 14. Thus, the rank of $\mathbf{D}$ ranges from 20 to 90. The Noisy data is modeled as $\mathbf{D}_e = \mathbf{D} + \mathbf{E}$, with the elements of $\mathbf{E}$ sampled independently from a zero mean Gaussian distribution. Fig. 4 shows the performance of the different algorithms versus the dimension of the intersection for $\tau = \frac{\|\mathbf{E}\|_F}{\|\mathbf{D}\|_F}$ equal to $1/20$, $1/10$, $1/5$ and $1/2$. One can observe that even with $\tau = 1/5$, iPursuit significantly outperforms the other algorithms. In addition, when the data is very noisy, i.e., $\tau = 1/2$, it yields better performance when the dimension of the intersection is large. SSC, LRR, and SSC-OMP yield a better performance for lower dimension of intersection. This is explained by the fact that the rank of the data is high when the dimension of the intersection is low, and the subspace projection operation $\mathbf{Q}^T\mathbf{D}_e$ may not always filter out the additive noise effectively.

3.4. Real data

We apply iPursuit to the problem of motion segmentation using the Hopkins155 (Tron & Vidal, 2007) dataset, which contains video sequences of 2 or 3 motions. In motion segmentation, each motion corresponds to one subspace. Thus, the problem here is to cluster data lying in two or three subspaces. Table 2 shows the clustering error (in percentage) for iPursuit, SSC, LRR, TSC, SSC-OMP and K-flats. We use the results reported in (Elhamifar & Vidal, 2013; Heckel & Bölcskei, 2013; Vidal, 2011; Park et al., 2014). For SSC-OMP and TSC, the number of parameters for motion segmentation are equal to 8 and 10. One can observe that iPursuit yields competitive results comparable to SSC, SCC, and LRR and outperforms TSC, SSC-OMP and K-flats.

Algorithm 1 Innovation pursuit (iPursuit) for noisy data

```
Initialization Set $\kappa$, $\hat{n}$ and $\hat{N}$ as integers greater than 1, and set $c_i$ and $c_o$ as positive real numbers less than 1.
While The number of identified subspaces is less than $\hat{N}$ or the number of the columns of $\mathbf{D}_e$ is less than $\hat{n}$.
    1. Obtaining the basis for the remaining Data: Construct $\mathbf{Q}$ as the orthonormal matrix formed by the dominant singular vectors of $\mathbf{D}_e$.
    2. Choosing the vector q: Set $\mathbf{q} =$ the column of $\mathbf{D}_e$ closest to the last column of $\mathbf{Q}$.
    3. Solve (14) and define $\mathbf{c}^* = \mathbf{Q}\mathbf{a}^*$, where $\mathbf{a}^*$ is the optimal point of (14) and define $\mathbf{h}_1 = \frac{|\mathbf{D}_e^T\mathbf{c}^*|}{\max(|\mathbf{D}_e^T\mathbf{c}^*|)}$.
    4. Finding a basis for the identified subspace: Construct $\mathbf{G}_1$ as the matrix consisting of the columns of $\mathbf{D}_e$ corresponding to the elements of $\mathbf{h}_1$ that are greater than $c_i$. Alternatively, construct $\mathbf{G}_1$ using the columns of $\mathbf{D}_e$ corresponding to the $\kappa$ largest elements of $\mathbf{h}_1$. Define $\mathbf{F}_1$ as an orthonormal basis for the dominant left singular vectors of $\mathbf{G}_1$.
    5. Finding a basis for the rest of the data:
       Define the vector $\mathbf{h}_2$ whose entries are equal to the $\ell_2$-norm of the columns of $(\mathbf{I} - \mathbf{F}_1\mathbf{F}_1^T)\mathbf{D}_e$. Scale $\mathbf{h}_2$ as $\mathbf{h}_2 := \mathbf{h}_2/\max(\mathbf{h}_2)$. Construct $\mathbf{G}_2$ as the columns of $\mathbf{D}_e$ corresponding to the elements of $\mathbf{h}_2$ greater than $c_o$. Define $\mathbf{F}_2$ as an orthonormal basis for the dominant left singular vectors of of $\mathbf{G}_2$.
    6. Find the data point belonging to the identified subspace: Assign $\mathbf{d}_{ei}$ to the identified subspace if $\|\mathbf{F}_1^T\mathbf{d}_{ei}\| \geq \|\mathbf{F}_2^T\mathbf{d}_{ei}\|$.
    7. Remove the data points belonging to the identified subspace: Update $\mathbf{D}_e$ by removing the columns corresponding to the identified subspace.
End While
```

Acknowledgment: This work was supported by NSF CAREER Award CCF-1552497 and NSF Grant CCF-1320547.

References

Boyd, Stephen, Parikh, Neal, Chu, Eric, Peleato, Borja, and Eckstein, Jonathan. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

Bradley, Paul S and Mangasarian, Olvi L. k-plane clustering. *Journal of Global Optimization*, 16(1):23–32, 2000.

Chen, Guangliang and Lerman, Gilad. Spectral curvature clustering (scc). *International Journal of Computer Vision*, 81(3):317–330, 2009.

Dyer, Eva L, Sankaranarayanan, Aswin C, and Baraniuk, Richard G. Greedy feature selection for subspace clustering. *The Journal of Machine Learning Research*, 14 (1):2487–2517, 2013.

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

Heckel, Reinhard and Bölcskei, Helmut. Robust subspace clustering via thresholding. *arXiv preprint arXiv:1307.4891*, 2013.

Ho, Jason, Yang, Ming-Hsuan, Lim, Jongwoo, Lee, Kuang-Chih, and Kriegman, David. Clustering appearances of objects under varying illumination conditions. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, 2003.

Lerman, Gilad, McCoy, Michael B, Tropp, Joel A, and Zhang, Teng. Robust computation of linear models by convex relaxation. *Foundations of Computational Mathematics*, 15(2):363–410, 2015.

Liu, Guangcan, Lin, Zhouchen, Yan, Shuicheng, Sun, Ju, Yu, Yong, and Ma, Yi. Robust recovery of subspace structures by low-rank representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1): 171–184, 2013.

Park, Dohyung, Caramanis, Constantine, and Sanghavi, Sujay. Greedy subspace clustering. In *Advances in Neural Information Processing Systems*, pp. 2753–2761, 2014.

Qu, Qing, Sun, Ju, and Wright, John. Finding a sparse vector in a subspace: Linear sparsity using alternating directions. In *Advances in Neural Information Processing Systems*, pp. 3401–3409, 2014.

Rahmani, Mostafa and Atia, George. Innovation pursuit: A new approach to subspace clustering. *arXiv preprint arXiv:1512.00907*, 2015.

Rahmani, Mostafa and Atia, George. A direction search and spectral clustering based approach to subspace clustering. *arXiv preprint*, 2017.

Rao, Shankar, Tron, Roberto, Vidal, Rene, and Ma, Yi. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(10):1832–1845, 2010.

Soltanolkotabi, Mahdi, Candes, Emmanuel J, et al. A geometric analysis of subspace clustering with outliers. *The Annals of Statistics*, 40(4):2195–2238, 2012.

Stoica, Petre and Selen, Yngve. Model-order selection: a review of information criterion rules. *Signal Processing Magazine, IEEE*, 21(4):36–47, 2004.

Tron, Roberto and Vidal, René. A benchmark for the comparison of 3-d motion segmentation algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2007.

Vidal, René, Soatto, Stefano, Ma, Yi, and Sastry, Shankar. An algebraic geometric approach to the identification of a class of linear hybrid systems. In *Proceedings of the 42nd IEEE Conference on Decision and Control (CDC)*, volume 1, pp. 167–172, 2003.

Vidal, Rene, Ma, Yi, and Sastry, Shankar. Generalized principal component analysis (GPCA). *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27 (12):1945–1959, 2005.

Vidal, René, Tron, Roberto, and Hartley, Richard. Multiframe motion segmentation with missing data using powerfactorization and GPCA. *International Journal of Computer Vision*, 79(1):85–105, 2008.

Vidal, Rene. Subspace clustering. *IEEE Signal Processing Magazine*, 2(28):52–68, 2011.

Von Luxburg, Ulrike. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

Yang, Allen Y, Rao, Shankar R, and Ma, Yi. Robust statistical estimation and segmentation of multiple subspaces. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, pp. 99–99. IEEE, 2006.

Yang, Allen Y, Wright, John, Ma, Yi, and Sastry, S Shankar. Unsupervised segmentation of natural images via lossy data compression. *Computer Vision and Image Understanding*, 110(2):212–225, 2008.