Coherence Pursuit: Fast, Simple, and Robust Subspace Recovery

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

Abstract

A remarkably simple, yet powerful, algorithm termed Coherence Pursuit for robust Principal Component Analysis (PCA) is presented. In the proposed approach, an outlier is set apart from an inlier by comparing their coherence with the rest of the data points. As inliers lie in a low dimensional subspace, they are likely to have strong mutual coherence provided there are enough inliers. By contrast, outliers do not typically admit low dimensional structures, wherefore an outlier is unlikely to bear strong resemblance with a large number of data points. As Coherence Pursuit only involves one simple matrix multiplication, it is significantly faster than the state-of-the-art robust PCA algorithms. We provide a mathematical analysis of the proposed algorithm under a random model for the distribution of the inliers and outliers. It is shown that the proposed method can recover the correct subspace even if the data is predominantly outliers. To the best of our knowledge, this is the first provable robust PCA algorithm that is simultaneously non-iterative, can tolerate a large number of outliers and is robust to linearly dependent outliers.

1. Introduction

Standard tools such as Principal Component Analysis (PCA) have been instrumental in reducing dimensionality by finding linear projections of high-dimensional data along the directions where the data is most spread out to minimize information loss. These techniques are widely applicable in a broad range of data analysis problems, including computer vision, image processing, machine learning and bioinformatics (Basri & Jacobs, 2003; Costeira & Kanade, 1998; Hosseini et al., 2014).

Given a data matrix $\mathbf{D} \in \mathbb{R}^{m\times n}$, PCA finds an r-dimensional subspace by solving

$$\min_{\hat{\mathbf{U}}} \|\mathbf{D} - \hat{\mathbf{U}}\hat{\mathbf{U}}^T\mathbf{D}\|_F \quad \text{subject to} \quad \hat{\mathbf{U}}^T\hat{\mathbf{U}} = \mathbf{I}, \qquad (1)$$

where $\hat{\mathbf{U}} \in \mathbb{R}^{m\times r}$ is an orthonormal basis for the r-dimensional subspace, $\mathbf{I}$ denotes the identity matrix and $\|.\|_F$ the Frobenius norm. Despite its notable impact on exploratory data analysis and multivariate analyses, PCA is notoriously sensitive to outliers that prevail much of the real world data since the solution to (1) can arbitrarily deviate from the true subspace in presence of a small number of outlying data points that do not conform with the low dimensional model (Hauberg et al., 2016; Lerman & Zhang, 2014; Maronna, 2005; McCoy et al., 2011; Xu et al., 2010a; Zhang, 2012) .

As a result, much research work was devoted to investigate PCA algorithms that are robust to outliers. The corrupted data can be expressed as

$$\mathbf{D} = \mathbf{L} + \mathbf{C}\,, \qquad (2)$$

where $\mathbf{L}$ is a low rank matrix whose columns span a low-dimensional subspace, and the matrix $\mathbf{C}$ models the data corruption, and is referred to as the outlier matrix. There are two main models for the outlier matrix that were considered in the literature, and these two models are mostly incomparable in theory, practice and analysis techniques. The first corruption model is the element-wise model in which $\mathbf{C}$ is a sparse matrix with arbitrary support, whose entries can have arbitrarily large magnitudes (Chandrasekaran et al., 2011; Candès et al., 2011; Netrapalli et al., 2014; Yi et al., 2016). In view of the arbitrary support of $\mathbf{C}$, any of the columns of $\mathbf{L}$ may be affected by the non-zero elements of $\mathbf{C}$. We do not consider this model in this paper. The second model, which is the focus of our paper, is a column-wise model wherein only a fraction of the columns of $\mathbf{C}$ are non-zero, wherefore a portion of the columns of $\mathbf{L}$ (the so-called inliers) remain unaffected by $\mathbf{C}$ (Lerman & Maunu, 2014; Xu et al., 2010b; Chen et al., 2011; Rahmani & Atia, 2017). Next, we formally describe the data model adopted in this paper, which only focuses on the column-wise outlier model.

Data Model 1. *The given data matrix* $\mathbf{D}$ *satisfies the following.*

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

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

1. The matrix $\mathbf{D}$ *can be expressed as*

$$\mathbf{D} = \mathbf{L} + \mathbf{C} = [\mathbf{A} \ \ \mathbf{B}] \, \mathbf{T} \,, \tag{3}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n_1}$*,* $\mathbf{B} \in \mathbb{R}^{m \times n_2}$*, and* $\mathbf{T}$ *is an arbitrary permutation matrix.*
2. The columns of $\mathbf{A}$ *lie in an* r*-dimensional subspace* $\mathcal{U}$*, namely, the column space of* $\mathbf{L}$*. The columns of* $\mathbf{B}$ *do not lie entirely in* $\mathcal{U}$*, i.e., the columns of* $\mathbf{A}$ *are the inliers and the columns of* $\mathbf{B}$ *are the outliers.*

The column-wise model for robust PCA has direct bearing on a host of applications in signal processing and machine learning, which spurred enormous progress in dealing with subspace recovery in the presence of outliers. This work is motivated by some of the limitations of existing techniques, which we further detail in Section 2 on related work. The vast majority of existing approaches to robust PCA have *high computational complexity*, which makes them unsuitable in high dimensional settings. For instance, many of the existing iterative techniques incur a long run time as they require a large number of iterations, each with a Singular Value Decomposition (SVD) operation. Also, most iterative solvers have *no provable guarantees* for exact subspace recovery. Moreover, many of the existing robust PCA algorithms *cannot tolerate a large number of outliers* as they rely on sparse outlier models, while others *cannot handle linearly dependent outliers*. In this paper, we present a new provable non-iterative robust PCA algorithm, dubbed Coherence Pursuit (CoP), which involves one simple matrix multiplication, and thereby achieves remarkable speedups over the state-of-the-art algorithms. CoP can tolerate a large number of outliers – even if the ratio of inliers to outliers $\frac{n_1}{n_2}$ approaches zero – and is robust to linearly dependent outliers and to additive noise.

1.1. Notation and definitions

Bold-face upper-case letters are used to denote matrices and bold-face lower-case letters are used to denote vectors. Given a matrix $\mathbf{A}$, $\|\mathbf{A}\|$ denotes its spectral norm and $\|\mathbf{A}\|_*$ its nuclear norm. For a vector $\mathbf{a}$, $\|\mathbf{a}\|_p$ denotes its ℓ_p-norm and $\mathbf{a}(i)$ its i^{th} element. Given two matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ with an equal number of rows, the matrix

$$\mathbf{A}_3 = [\mathbf{A}_1 \ \ \mathbf{A}_2]$$

is the matrix formed by concatenating their columns. For a matrix $\mathbf{A}$, $\mathbf{a}_i$ denotes its i^{th} column, and $\mathbf{A}_{-i}$ is equal to $\mathbf{A}$ with the i^{th} column removed. The function orth$(\cdot)$ returns an orthonormal basis for the range of its matrix argument.

Definition 1. *The mutual coherence of two non-zero vectors* $\mathbf{v}_1 \in \mathbb{R}^{m \times 1}$ *and* $\mathbf{v}_2 \in \mathbb{R}^{m \times 1}$ *is defined as*

$$\frac{|\mathbf{v}_1^T \mathbf{v}_2|}{\|\mathbf{v}_1\| \|\mathbf{v}_2\|} .$$

2. Related Work

Some of the earliest approaches to robust PCA relied on robust estimation of the data covariance matrix, such as S-estimators, the minimum covariance determinant, the minimum volume ellipsoid, and the Stahel-Donoho estimator (Huber, 2011). This is a class of iterative approaches that compute a full SVD or eigenvalue decomposition in each iteration and generally have no explicit performance guarantees. The performance of these approaches greatly degrades for $\frac{n_1}{n_2} \leq 0.5$.

To enhance robustness to outliers, another approach is to replace the Frobenius norm in (1) with other norms (Lerman et al., 2011). For example, (Ke & Kanade, 2005) uses an ℓ_1-norm relaxation commonly used for sparse vector estimation, yielding robustness to outliers (Candes & Tao, 2005; Candès et al., 2006; 2011). However, the approach presented in (Ke & Kanade, 2005) has no provable guarantees and requires $\mathbf{C}$ to be column sparse, i.e., a very small portion of the columns of $\mathbf{C}$ can be non-zero. The work in (Ding et al., 2006) replaces the ℓ_1-norm in (Ke & Kanade, 2005) with the $\ell_{1,2}$-norm. While the algorithm in (Ding et al., 2006) can handle a large number of outliers, the complexity of each iteration is $\mathcal{O}(nm^2)$ and its iterative solver has no performance guarantees. Recently, the idea of using a robust norm was revisited in (Lerman et al., 2012; Zhang & Lerman, 2014). Therein, the non-convex constraint set is relaxed to a larger convex set and exact subspace recovery is guaranteed under certain conditions. The algorithm presented in (Lerman et al., 2012) obtains the column space of $\mathbf{L}$ and (Zhang & Lerman, 2014) finds its complement. However, the iterative solver of (Lerman et al., 2012) computes a full SVD of an $m \times m$ weighted covariance matrix in each iteration. Thus, the overall complexity of the solver of (Lerman et al., 2012) is roughly $\mathcal{O}(m^3 + nm^2)$ per iteration, where the second term is the complexity of computing the weighted covariance matrix. Similarly, the solver of (Zhang & Lerman, 2014) has $\mathcal{O}(nm^2 + m^3)$ complexity per iteration. In (Tsakiris & Vidal, 2015), the complement of the column space of $\mathbf{L}$ is recovered via a series of linear optimization problems, each obtaining one direction in the complement space. This method is sensitive to linearly dependent outliers and requires the columns of $\mathbf{L}$ not to exhibit a clustering structure, which in fact prevails much of the real world data. Also, the approach presented in (Tsakiris & Vidal, 2015) requires solving $m-r$ linear optimization problems consecutively resulting in high computational complexity and long run time for high dimensional data.

Robust PCA using convex rank minimization was first analyzed in (Chandrasekaran et al., 2011; Candès et al., 2011) for the element-wise corruption model. In (Xu et al., 2010b), the algorithm analyzed in (Chandrasekaran et al., 2011; Candès et al., 2011) was extended to the column-wise corruption model where it was shown that the optimal point of

$$\begin{aligned} \min_{\hat{\mathbf{L}},\hat{\mathbf{C}}} \quad & \|\hat{\mathbf{L}}\|_* + \lambda\|\hat{\mathbf{C}}\|_{1,2} \\ \text{subject to} \quad & \hat{\mathbf{L}} + \hat{\mathbf{C}} = \mathbf{D} \end{aligned} \tag{4}$$

yields the exact subspace and correctly identifies the outliers provided that $\mathbf{C}$ is sufficiently column-sparse. The solver of (4) requires too many iterations, each computing the SVD of an $m \times n$ dimensional matrix. Also, the algorithm can only tolerate a small number of outliers – the ratio $\frac{n_2}{n_1}$ should be roughly less than 0.05.

A different approach to outlier detection was proposed in (Soltanolkotabi & Candes, 2012; Elhamifar & Vidal, 2013), the idea being that outliers do not typically follow low dimensional structures, thereupon few outliers cannot form a linearly dependent set. While this approach can recover the correct subspace even if a remarkable portion of the data is outliers, it cannot detect linearly dependent outliers and has $\mathcal{O}(n^3)$ complexity per iteration (Elhamifar & Vidal, 2013). In the outlier detection algorithm presented in (Heckel & Bölcskei, 2013), a data point is identified as an outlier if the maximum value of its mutual coherences with the other data points falls below a predefined threshold. However, this approach is unable to detect outliers that lie in close neighborhoods. For instance, any repeated outliers will be falsely detected as inliers since their mutual coherence is 1.

2.1. Motivation and summary of contributions

This work is motivated by the limitations of prior work on robust PCA as summarized below.

Complex iterations. Most of the state-of-the-art robust PCA algorithms require a large number of iterations each with high computational complexity. For instance, many of these algorithms require the computation of the SVD of an $m \times n$, or $m \times m$, or $n \times n$ matrix in each iteration (Hardt & Moitra, 2012; Xu et al., 2010b; Lerman et al., 2012), which leads to long run time.

Guarantees. While the optimal points of the optimization problems underlying many of the existing robust subspace recovery techniques yield the exact subspace, there are no such guarantees for their corresponding iterative solvers. Examples include the optimal points of the optimization problems presented in (Xu et al., 2010b; Ding et al., 2006).

Robustness issues. Most existing algorithms are tailored to one specific class of outlier models. For example, algorithms based on sparse outlier models utilize sparsity promoting norms, thus can only handle a small number of outliers. On the other hand, algorithms such as (Heckel & Bölcskei, 2013; Soltanolkotabi & Candes, 2012) can handle a large number of outliers, albeit they fail to locate outliers with high similarity or linearly dependent outliers. Spherical PCA (SPCA) is a non-iterative robust PCA algorithm that is also scalable (Maronna et al., 2006). In this algorithm, all the columns of $\mathbf{D}$ are first projected onto the unit sphere $\mathbb{S}^{m-1}$, then the subspace is identified as the span of the principal directions of the normalized data. However, in the presence of outliers, the recovered subspace is never equal to the true subspace and it significantly deviates from the underlying subspace when outliers abound.

To the best of our knowledge, CoP is the first algorithm that addresses these concerns all at once. In the proposed method, we distinguish outliers from inliers by comparing their degree of coherence with the rest of the data. The advantages of the proposed algorithm are summarized below.

- Coherence Pursuit (CoP) is a considerably simple non-iterative algorithm which roughly involves one matrix multiplication to compute the Gram matrix. It also has provable performance guarantees (c.f. Section 4).
- CoP can tolerate a large number of outliers. It is shown that exact subspace recovery is guaranteed with high probability even if $\frac{n_1}{n_2}$ goes to zero provided that $\frac{n_1}{n_2}\frac{m}{r}$ is sufficiently large.
- CoP is robust to linearly dependent outliers since it measures the coherence of a data point with respect to all the other data points.

Algorithm 1 CoP: Proposed Robust PCA Algorithm

Initialization: Set $p = 1$ or $p = 2$.

1. Data Normalization: Define matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ as $\mathbf{x}_i = \mathbf{d}_i / \|\mathbf{d}_i\|_2$.

2. Mutual Coherence Measurement
2.1 Define $\mathbf{G} = \mathbf{X}^T\mathbf{X}$ and set its diagonal elements to zero.
2.2 Define vector $\mathbf{p} \in \mathbb{R}^{n \times 1}$ as $\mathbf{p}(i) = \|\mathbf{g}_i\|_p, i = 1, \dots, n$.

3. Subspace Identification: Construct matrix $\mathbf{Y}$ from the columns of $\mathbf{X}$ corresponding to the largest elements of $\mathbf{p}$ such that they span an r-dimensional subspace.

Output: The columns of $\mathbf{Y}$ are a basis for the column space of $\mathbf{L}$.

3. Proposed Method

In this section, we present the Coherence Pursuit algorithm and provide some insight into its characteristics. The main theoretical results are provided in Section 4. The table of

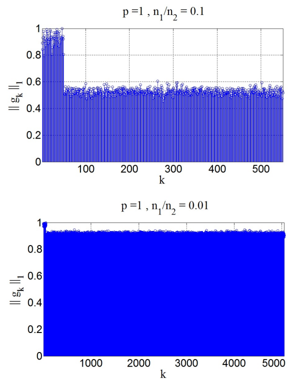

Figure 1. The values of vector $\mathbf{p}$ for different values of p and $\frac{n_1}{n_2}$.

Algorithm 1 presents the proposed method along with the definitions of the used symbols.

Coherence: The inliers lie in a low dimensional subspace $\mathcal{U}$. In addition, for most practical purposes the inliers are highly coherent with each other in the sense of having large values of the mutual coherence in Definition 1. By contrast, the outlying columns do not typically follow low dimensional structures and do not bear strong resemblance with the rest of the data. As such, in CoP all mutual coherences between the columns of $\mathbf{D}$ are first computed. Then, the column space of $\mathbf{A}$ is obtained as the span of those columns that have strong mutual coherence with the rest of the data.

For instance, assume that the distributions of the inliers and the outliers follow the following assumption.

Assumption 1. *The subspace $\mathcal{U}$ is a random r-dimensional subspace in $\mathbb{R}^m$. The columns of $\mathbf{A}$ are drawn uniformly at random from the intersection of $\mathbb{S}^{m-1}$ and $\mathcal{U}$. The columns of $\mathbf{B}$ are drawn uniformly at random from $\mathbb{S}^{m-1}$. To simplify the exposition and notation, it is assumed that $\mathbf{T}$ in (3) is equal to the identity matrix without any loss of generality, i.e, $\mathbf{D} = [\mathbf{A} \;\; \mathbf{B}]$.*

In addition, suppose that $\mathbf{v}_1$ and $\mathbf{v}_2$ are two inliers and $\mathbf{u}_1$ and $\mathbf{u}_1$ are two outliers. It can be shown that

$$\mathbb{E}(\mathbf{v}_1^T \mathbf{v}_2)^2 = 1/r$$

while

$$\mathbb{E}(\mathbf{u}_1^T \mathbf{u}_2)^2 = 1/m \;\;, \;\; \mathbb{E}(\mathbf{u}_1^T \mathbf{v}_1)^2 = 1/m \;.$$

Accordingly, if $m \gg r$, the inliers exhibit much stronger mutual coherences.

3.1. Large number of outliers

Unlike most robust PCA algorithms which require n_2 to be much smaller than n_1, the proposed method tolerates a large number of outliers. For instance, consider a setting in which $m = 400$, $r = 5$, $n_1 = 50$, and the distributions of inliers and outliers follow Assumption 1. Fig. 1 shows the vector $\mathbf{p}$ (c.f. Algorithm 1) for different values of p and n_2. In all the figures, the maximum element is scaled to 1. One can observe that even if $n_1/n_2 = 0.01$, the proposed technique can recover the exact subspace since there is a clear gap between the values of $\mathbf{p}$ corresponding to outliers and inliers, more so for $p = 2$.

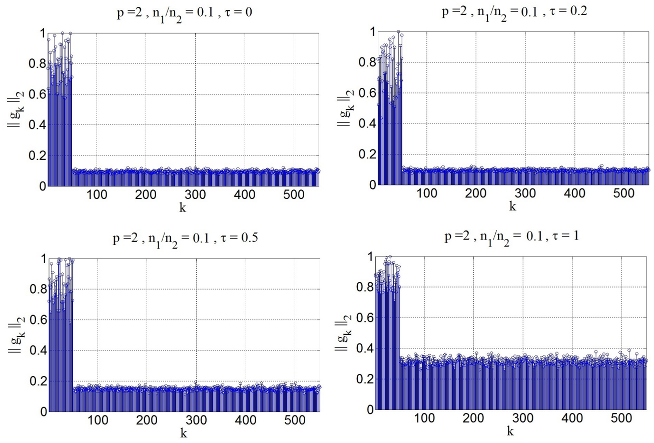

Figure 2. The elements of vector $\mathbf{p}$ with different values of parameter τ.

3.2. Robustness to noise

In the presence of additive noise, we model the data as

$$\mathbf{D} = [\mathbf{A} \;\; \mathbf{B}] \, \mathbf{T} + \mathbf{E} \,, \tag{5}$$

where $\mathbf{E}$ represents the noise component.

The high coherence between the inliers (columns of $\mathbf{A}$) and the low coherence of the outliers (columns of $\mathbf{B}$) with each other and with the inliers result in the large gap between the elements of $\mathbf{p}$ observed in Fig. 1 even when $n_1/n_2 < 0.01$. This gap gives the algorithm tolerance to high levels of noise. For example, assume $r = 5$, $n_1 = 50$, $n_2 = 500$ and the distribution of the inliers and outliers follow Assumption 1. Define the parameter τ as

$$\tau = \frac{\mathbb{E}\|\mathbf{e}\|_2}{\mathbb{E}\|\mathbf{a}\|_2} \tag{6}$$

where $\mathbf{a}$ and $\mathbf{e}$ are arbitrary columns of $\mathbf{A}$ and $\mathbf{E}$, respectively. Fig. 2 shows the entries of $\mathbf{p}$ for different values of τ. As shown, the elements corresponding to inliers are clearly separated from the ones corresponding to outliers even at very low signal to noise ratio, e.g. $\tau = 0.5$ and $\tau = 1$. The mathematical analysis of CoP with noisy data confirms that the proposed method remains robust to high levels of noise even with data that is predominantly outliers (Rahmani & Atia, 2016).

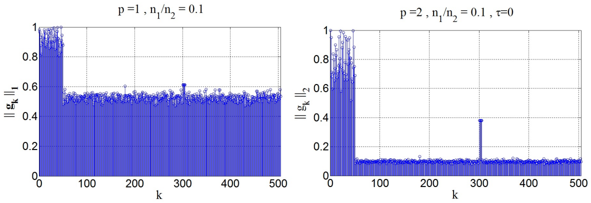

Figure 3. The values of vector $\mathbf{p}$ when the 301th to 305th columns are repeated outliers.

3.3. Linearly dependent outliers

Many of the existing robust PCA algorithms cannot detect linearly dependent outliers (Hardt & Moitra, 2012; Soltanolkotabi & Candes, 2012). For instance, in the outlier detection algorithm presented in (Soltanolkotabi & Candes, 2012) a data column is identified as an outlier if it does not admit a sparse representation in the rest of the data. As such, if some outlying columns are repeated, the algorithm in (Soltanolkotabi & Candes, 2012) fails to detect them.

At a fundamental level, the proposed approach affords a more comprehensive definition of an outlying column, namely, a data column is identified as an outlier if it has weak total coherency with the rest of the data. This global view of a data point w.r.t. the rest of the data allows the algorithm to identify outliers that are linearly dependent with few other data points.

For illustration, assume the given data follows Data model 1, $r = 5$, $n_1 = 50$, $n_2 = 500$, where the 301th to 305th columns are repeated outliers. Fig. 3 shows the elements of vector $\mathbf{p}$ with $p = 1$ and $p = 2$. All the elements corresponding to inliers are clearly greater than the elements corresponding to outliers and the algorithm correctly identifies the subspace. Fig. 3 suggests that CoP with $p = 1$ is better at handling repeated outliers since the entries of $\mathbf{G}$ with large absolute magnitudes are more gracefully amplified by the ℓ_1-norm.

3.4. Subspace identification

In the third step of Algorithm 1, we sample the columns of $\mathbf{X}$ with the largest coherence values which span an r-dimensional space. In this section, we present some tips for efficient implementation of this step. One way is to start sampling the columns with the highest coherence values and stop when the rank of the sampled columns is equal to r. However, if the columns of $\mathbf{L}$ admit a clustering structure and their distribution is highly non-uniform, this method will sample many redundant columns, which can in turn increase the run time. In this section, we propose two low-complexity techniques to accelerate the subspace identification step.

1. In many applications, we may have an upper bound on n_2/n. For instance, suppose we know that up to 40 percent of the data could be outliers. In this case, we simply remove 40 percent of the columns corresponding to the smallest values of $\mathbf{p}$ and obtain the subspace using the remaining data points.

2. The second technique is an adaptive sampling method presented in the table of Algorithm 2. First, the data is projected onto a random kr-dimensional subspace to reduce the computational complexity for some integer $k > 1$. According to the analysis presented in (Rahmani & Atia, 2017; Li & Haupt, 2015), even $k = 2$ is sufficient to preserve the rank of $\mathbf{A}$ and the structure of the outliers $\mathbf{B}$, i.e., the rank of $\mathbf{\Phi A}$ is equal to r and the columns of $\mathbf{\Phi B}$ do not lie in the column space of $\mathbf{\Phi A}$, where $\mathbf{\Phi}$ is the projection matrix. The parameter ν that thresholds the ℓ_2-norms of the columns of the projected data is chosen based on the noise level (if the data is noise free, $\nu = 0$). In Algorithm 2, the data is projected onto the span of the sampled columns (step 2.3). Thus, a newly sampled column brings innovation with respect to the previously sampled columns. Therefore, redundant columns are not sampled.

Algorithm 2 Adaptive Column Sampling for the Subspace Identification Step (step 3) of CoP

```
Initialization: Set $k$ equal to an integer greater than 1, a threshold $\nu$ greater than or equal to 0, and $\mathbf{F}$ an empty matrix.
1. Data Projection: Define $\mathbf{X}_\phi \in \mathbb{R}^{kr \times n}$ as $\mathbf{X}_\phi = \mathbf{\Phi X}$, where $\mathbf{\Phi} \in \mathbb{R}^{kr \times m}$ projects the columns of $\mathbf{X}$ onto a random $kr$-dimensional subspace.
2. Column Sampling
For $i$ from 1 to $r$ do
    2.1 Set equal to zero the elements of $\mathbf{p}$ corresponding to columns of $\mathbf{X}_\phi$ with $\ell_2$-norms less than or equal to $\nu$.
    2.2 Define $j := \arg\max_k \mathbf{p}(k)$, update $\mathbf{F} = \text{orth}\Big( [\mathbf{F} \;\; \mathbf{x}_j] \Big)$, and set $\mathbf{p}(j) = 0$.
    2.3 Update $\mathbf{X}_\phi = \mathbf{X}_\phi - \mathbf{F}\mathbf{F}^T \mathbf{X}_\phi$.
End For
Output Construct $\mathbf{Y}$ using the columns of $\mathbf{X}$ that correspond to the columns that formed $\mathbf{F}$.
```

Remark 1. *Suppose we run Algorithm 2 h times (each time the sampled columns are removed from the data and newly sampled columns are added to $\mathbf{Y}$). If the given data is noisy, the first r singular values of $\mathbf{Y}$ are the dominant ones and the rest correspond to the noise component. If we increase h, the span of the dominant singular vectors will be closer to the column space of $\mathbf{A}$. However, if h is chosen unreasonably high, the sampler may also sample outliers.*

3.5. Computational complexity

The main computational complexity is in the second step of Algorithm 2 which is of order $\mathcal{O}(mn^2)$. If we utilize Algorithm 2 as the third step of Algorithm 1, the overall complexity is of order $\mathcal{O}(mn^2 + r^3 n)$. However, since the algorithm roughly involves only one matrix multiplication, it is very fast and very simple for hardware implementation (c.f. Section 5.2 on run time). Moreover, if we utilize the randomized designs presented in (Rahmani & Atia, 2017; Li & Haupt, 2015), the overall complexity can be reduced to $\mathcal{O}(r^4)$.

4. Theoretical Investigations

In this section, we first establish sufficient conditions to ensure that the expected value of the elements of the vector $\mathbf{p}$ corresponding to the inliers are much greater than the elements corresponding to the outliers, in which case the algorithm is highly likely to yield exact subspace recovery. Subsequently, we provide two theorems establishing performance guarantees for the proposed approach for $p = 1$ and $p = 2$.

The following lemmas establish sufficient conditions for the expected value of the elements of $\mathbf{p}$ corresponding to inliers to be at least twice as large as those corresponding to outliers. Due to space limitations, the proofs of all lemmas and theorems are deferred to an extended version of this paper (Rahmani & Atia, 2016).

Lemma 1. *Suppose Assumption 1 holds, the i^{th} column is an inlier and the $(n_1 + j)^{th}$ column is an outlier. If*

$$\frac{n_1}{\sqrt{r}}\left(\sqrt{\frac{2}{\pi}} - \sqrt{\frac{4r^2}{m}}\right) + \sqrt{\frac{4}{m}} > \frac{5\, n_2}{4\sqrt{m}} + \sqrt{\frac{2}{\pi r}}\,, \tag{7}$$

then

$$\mathbb{E}\, \|\mathbf{g}_i\|_1 > 2\, \mathbb{E}\, \|\mathbf{g}_{n_1+j}\|_1$$

recalling that $\mathbf{g}_i$ is the i^{th} column of matrix $\mathbf{G}$.

Lemma 2. *Suppose Assumption 1 holds, the i^{th} column is an inlier and the $(n_1 + j)^{th}$ column is an outlier. If*

$$\frac{n_1}{r}\left(1 - \frac{2r^2}{m}\right) > \frac{n_2}{m} + \frac{1}{r} \tag{8}$$

then

$$\mathbb{E}\, \|\mathbf{g}_i\|_2^2 > 2\, \mathbb{E}\, \|\mathbf{g}_{n_1+j}\|_2^2\,.$$

Remark 2. *The sufficient conditions provided in Lemma 1 and Lemma 2 reveal three important points.*

I) The important performance factors are the ratios $\frac{n_1}{r}$ and $\frac{n_2}{m}$. The intuition is that as $\frac{n_1}{r}$ increases, the density of the inliers in the subspace increases, and consequently their mutual coherence also increases. Similarly, if $\frac{n_2}{m}$ increases, the mutual coherence between the outliers increases. Thus, the main requirement is that $\frac{n_1}{r}$ should be sufficiently larger than $\frac{n_2}{m}$.

II) In real applications, $r \ll m$ and $n_1 > n_2$, hence the sufficient conditions are easily satisfied. This fact is observed in Fig. 1 which shows that Coherence Pursuit can recover the correct subspace even if $n_1/n_2 = 0.01$.

III) In high dimensional settings, $r \ll m$. Therefore, $\frac{m}{\sqrt{m}}$ could be much greater than $\frac{r}{\sqrt{r}}$. Accordingly, the conditions in Lemma 1 are stronger than the conditions of Lemma 2, suggesting that with $p = 2$ Coherence Pursuit can tolerate a larger number of outliers than with $p = 1$. This is confirmed by comparing the plots in the last row of Fig. 1.

The following theorems show that the same set of factors are important to guarantee that the proposed algorithm recovers the exact subspace with high probability.

Theorem 3. *If Assumption 1 is true and*

$$\frac{n_1}{\sqrt{r}}\left(\sqrt{\frac{2}{\pi}} - \frac{r + 2\sqrt{\beta\kappa\, r}}{\sqrt{m}}\right) - 2\sqrt{n_1} - \sqrt{\frac{2n_1 \log\frac{n_1}{\delta}}{r-1}} > \frac{n_2}{\sqrt{m}} + 2\sqrt{n_2} + \sqrt{\frac{2n_2 \log\frac{n_2}{\delta}}{m-1}} + \frac{1}{\sqrt{r}}\,, \tag{9}$$

then Algorithm 1 with $p = 1$ recovers the exact subspace with probability at least $1 - 3\delta$, where $\beta = \max(8\log n_2/\delta, 8\pi)$ and $\kappa = \frac{m}{m-1}$.

Theorem 4. *If Assumption 1 is true and*

$$n_1\left(\frac{1}{r} - \frac{r + 4\zeta\kappa + 4\sqrt{\zeta r \kappa}}{m}\right) - \eta_1 > 2\eta_2 + \frac{1}{r}\,, \tag{10}$$

then Algorithm 1 with $p = 2$ recovers the correct subspace with probability at least $1 - 4\delta$, where

$$\eta_1 = \max\left(\frac{4}{3}\log\frac{2rn_1}{\delta}, \sqrt{4\frac{n_1}{r}\log\frac{2rn_1}{\delta}}\right)\,,$$

$$\eta_2 = \max\left(\frac{4}{3}\log\frac{2mn_2}{\delta}, \sqrt{4\frac{n_2}{m}\log\frac{2mn_2}{\delta}}\right)\,,$$

$\zeta = \max(8\pi, 8\log\frac{n_2}{\delta})$, and $\kappa = \frac{m}{m-1}$.

Remark 3. *The dominant factors of the LHS and the RHS of (10) are $\frac{n_1}{r}\left(1 - \frac{r^2}{m}\right)$ and $\sqrt{4\frac{n_2}{m}\log\frac{2mn_2}{\delta}}$, respectively. As in Lemma 2, we see the factor $\frac{n_2}{m}$, but under the square root. Thus, the requirement of Theorem 4 is less stringent than that of Lemma 2. This is because Theorem 4 guarantees that the elements corresponding to inliers are greater than those corresponding to outliers with high probability, but does not guarantee a large gap between the values as in Lemma 2.*

Algorithm 3 Subspace Clustering Error Correction Method

```
Input: The matrices $\{\hat{\mathbf{D}}^i\}_{i=1}^L$ represent the clustered data (the output of a subspace clustering algorithm) and $L$ is the number of clusters.

Error Correction
For $k$ from 1 to $t$ do
  1 Apply the robust PCA algorithm to the matrices $\{\hat{\mathbf{D}}^i\}_{i=1}^L$. Define the orthonormal matrices $\{\hat{\mathbf{U}}^i\}_{i=1}^L$ as the learned bases for the inliers of $\{\hat{\mathbf{D}}^i\}_{i=1}^L$, respectively.
  2 Update the data clustering with respect to the obtained bases $\{\hat{\mathbf{U}}^i\}_{i=1}^L$ (the matrices $\{\hat{\mathbf{D}}^i\}_{i=1}^L$ are updated), i.e., a data point $\mathbf{d}$ is assigned to the $i^{\text{th}}$ cluster if $i = \arg\max_k \|\mathbf{x}^T \hat{\mathbf{U}}^k\|_2$.
End For

Output: The matrices $\{\hat{\mathbf{D}}^i\}_{i=1}^L$ represent the clustered data and the matrices $\{\hat{\mathbf{U}}^i\}_{i=1}^L$ are the orthonormal bases for the identified subspaces.
```

5. Numerical Simulations

In this section, the performance of the proposed method is investigated with both synthetic and real data. We compare the performance of CoP with the state-of-the-art robust PCA algorithms including FMS (Lerman & Maunu, 2014), GMS (Zhang & Lerman, 2014), R1-PCA (Ding et al., 2006), OP (Xu et al., 2010b), and SPCA (Maronna et al., 2006).

5.1. Phase transition plot

Our analysis has shown that CoP yields exact subspace recovery with high probability if n_1/r is sufficiently greater than n_2/m. In this experiment, we investigate the phase transition of CoP in the n_1/r and n_2/m plane. Suppose $m = 100$, $r = 10$, and the distributions of inliers/outliers follow Assumption 1. Define $\mathbf{U}$ and $\hat{\mathbf{U}}$ as the exact and recovered orthonormal bases for the span of inliers, respectively. A trial is considered successful if

$$\left(\|\mathbf{U} - \hat{\mathbf{U}}\hat{\mathbf{U}}^T\mathbf{U}\|_F / \|\mathbf{U}\|_F \right) \leq 10^{-5} \,.$$

In this simulation, we construct the matrix $\mathbf{Y}$ using 20 columns of $\mathbf{X}$ corresponding to the largest 20 elements of the vector $\mathbf{p}$. Fig. 4 shows the phase transition plot. White indicates correct subspace recovery and black designates incorrect recovery. As shown, if n_2/m increases, we need higher values of n_1/r. However, one can observe that with $n_1/r > 4$, the algorithm can yield exact recovery even if $n_2/m > 30$.

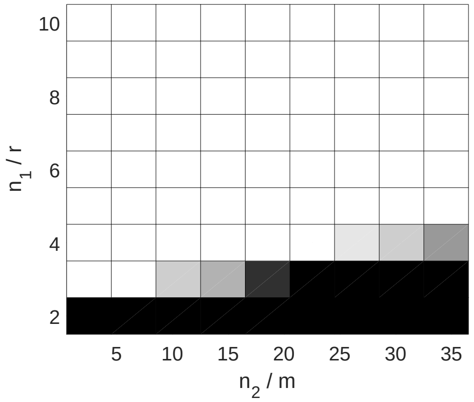

Figure 4. The phase transition plot of CoP versus n_1/r and n_2/m.

5.2. Running time

In this section, we compare the run time of CoP to the existing approaches. Table 1 presents the run time in seconds for different data sizes. In all experiments, $n_1 = n/5$ and $n_2 = 4n/5$. One can observe that CoP is remarkably faster given its simplicity (single step algorithm).

Table 1. Running time of the algorithms

$m = n$	CoP	FMS	OP	R1-PCA
1000	0.02	1	45	1.45
2000	0.7	5.6	133	10.3
5000	5.6	60	811	83.3
10000	27	401	3547	598

5.3. Subspace recovery in presence of outliers

In this experiment, we assess the robustness of CoP to outliers in comparison to existing approaches. It is assumed that $m = 50$, $r = 10$, $n_1 = 50$ and the distribution of inliers/outliers follows Assumption 1. Define $\mathbf{U}$ and $\hat{\mathbf{U}}$ as before, and the recovery error as

$$\text{Log-Recovery Error} = \log_{10} \left(\|\mathbf{U} - \hat{\mathbf{U}}\hat{\mathbf{U}}^T\mathbf{U}\|_F / \|\mathbf{U}\|_F \right) .$$

In this simulation, we use 30 columns to form the matrix $\mathbf{Y}$. Fig. 5 shows the recovery error versus n_2/n_1 for different values of n_2. In addition to its remarkable simplicity, CoP gives the highest accuracy and yields exact subspace recovery even if the data is overwhelmingly outliers.

5.4. Clustering error correction

In this section, we consider a very challenging robust subspace recovery problem with real data. We use the robust PCA algorithm as a subroutine for error correction in a subspace clustering problem. In this experiment, the outliers

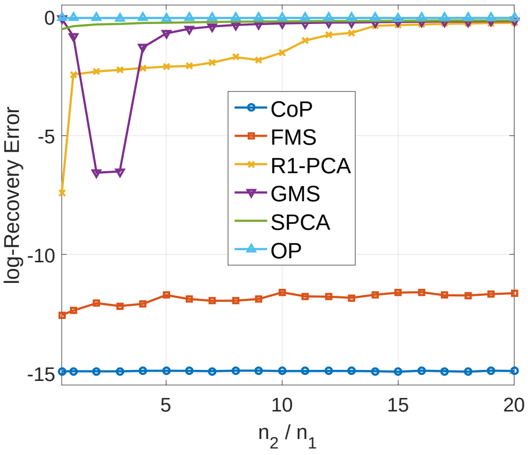

Figure 5. The subspace recovery error versus n_2/n_1.

can be close to the inliers and can also be linearly dependent.

The subspace clustering problem is a general form of PCA in which the data points lie in a union of an unknown number of unknown linear subspaces (Vidal, 2011; Rahmani & Atia, 2015). A subspace clustering algorithm identifies the subspaces and clusters the data points with respect to the subspaces. The performance of the subspace clustering algorithms – especially the ones with scalable computational complexity – degrades in presence of noise or when the subspaces are closer to each other. Without loss of generality, suppose $\mathbf{D} = [\mathbf{D}^1 \; ... \; \mathbf{D}^L]$ is the given data where the columns of $\{\mathbf{D}^i\}_{i=1}^L$ lie in the linear subspaces $\{\mathcal{S}_i\}_{i=1}^L$, respectively, and L is the number of subspaces. Define $\{\hat{\mathbf{D}}^i\}_{i=1}^L$ as the output of some clustering algorithm (the clustered data). Define the clustering error as the ratio of misclassified points to the total number of data points. With the clustering error, some of the columns of $\hat{\mathbf{D}}^i$ believed to lie in $\mathcal{S}_i$ may actually belong to some other subspace. Such columns can be viewed as outliers in the matrix $\hat{\mathbf{D}}^i$. Accordingly, the robust PCA algorithm can be utilized to correct the clustering error. We present Algorithm 3 as an error correction algorithm which can be applied to the output of any subspace clustering algorithm to reduce the clustering error. In each iteration, Algorithm 3 applies the robust PCA algorithm to the clustered data to obtain a set of bases for the subspaces. Subsequently, the obtained clustering is updated based on the obtained bases.

In this experiment, we imagine a subspace clustering algorithm with 20 percent clustering error and apply Algorithm 3 to the output of the algorithm to correct the errors. We use the Hopkins155 dataset which contains video sequences of 2 or 3 motions (Tron & Vidal, 2007). The data is generated by extracting and tracking a set of feature points through the frames. In motion segmentation, each motion corresponds to one subspace. Thus, the problem here is to cluster data lying in two or three subspaces (Vidal, 2011). We use the traffic data sequences, which include 8 scenarios with two motions and 8 scenarios with three motions.

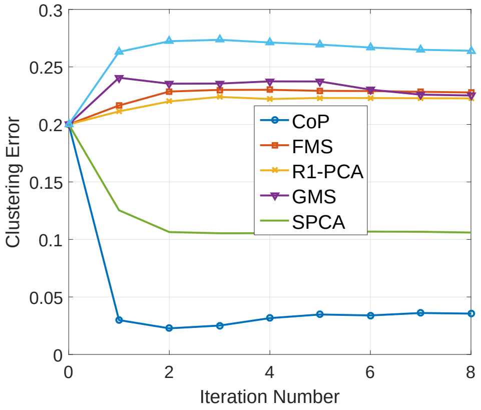

Figure 6. The clustering error after each iteration of Algorithm 2.

When CoP is applied, 50 percent of the columns of $\mathbf{X}$ are used to form the matrix $\mathbf{Y}$. Fig. 6 shows the average clustering error (over all traffic data matrices) after each iteration of Algorithm 3 for different robust PCA algorithms. CoP clearly outperforms the other approaches. As a matter of fact, most of the robust PCA algorithms fail to obtain the correct subspaces and end up increasing the clustering error.

Acknowledgment

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

References

Basri, Ronen and Jacobs, David W. Lambertian reflectance and linear subspaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(2):218–233, 2003.

Candes, Emmanuel J and Tao, Terence. Decoding by linear programming. *Information Theory, IEEE Transactions on*, 51(12):4203–4215, 2005.

Candès, Emmanuel J, Romberg, Justin, and Tao, Terence. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *Information Theory, IEEE Transactions on*, 52(2):489–509, 2006.

Candès, Emmanuel J, Li, Xiaodong, Ma, Yi, and Wright, John. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

Chandrasekaran, Venkat, Sanghavi, Sujay, Parrilo, Pablo A, and Willsky, Alan S. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

Chen, Yudong, Xu, Huan, Caramanis, Constantine, and Sanghavi, Sujay. Robust matrix completion with corrupted columns. *arXiv preprint arXiv:1102.2254*, 2011.

Costeira, João Paulo and Kanade, Takeo. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998.

Ding, Chris, Zhou, Ding, He, Xiaofeng, and Zha, Hongyuan. R1-PCA: rotational invariant l 1-norm principal component analysis for robust subspace factorization. In *Proceedings of the 23rd international conference on Machine learning*, pp. 281–288. ACM, 2006.

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.

Hardt, Moritz and Moitra, Ankur. Algorithms and hardness for robust subspace recovery. *arXiv preprint arXiv:1211.1041*, 2012.

Hauberg, Soren, Feragen, Aasa, Enficiaud, Raffi, and Black, Michael. Scalable robust principal component analysis using grassmann averages. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11): 2298–2311, Nov 2016.

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

Hosseini, Mohammad-Parsa, Nazem-Zadeh, Mohammad R, Mahmoudi, Fariborz, Ying, Hao, and Soltanian-Zadeh, Hamid. Support vector machine with nonlinear-kernel optimization for lateralization of epileptogenic hippocampus in mr images. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 1047–1050. IEEE, 2014.

Huber, Peter J. *Robust statistics*. Springer, 2011.

Ke, Qifa and Kanade, Takeo. Robust l 1 norm factorization in the presence of outliers and missing data by alternative convex programming. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pp. 739–746. IEEE, 2005.

Lerman, Gilad and Maunu, Tyler. Fast, robust and non-convex subspace recovery. *arXiv preprint arXiv:1406.6145*, 2014.

Lerman, Gilad and Zhang, Teng. {l_p}-recovery of the most significant subspace among multiple subspaces with outliers. *Constructive Approximation*, 40(3):329–385, 2014.

Lerman, Gilad, Zhang, Teng, et al. Robust recovery of multiple subspaces by geometric lp minimization. *The Annals of Statistics*, 39(5):2686–2715, 2011.

Lerman, Gilad, McCoy, Michael, Tropp, Joel A, and Zhang, Teng. Robust computation of linear models, or how to find a needle in a haystack. Technical report, DTIC Document, 2012.

Li, Xingguo and Haupt, Jarvis. Identifying outliers in large matrices via randomized adaptive compressive sampling. *Signal Processing, IEEE Transactions on*, 63 (7):1792–1807, 2015.

Maronna, RARD, Martin, Douglas, and Yohai, Victor. *Robust statistics*. John Wiley & Sons, Chichester. ISBN, 2006.

Maronna, Ricardo. Principal components and orthogonal regression based on robust scales. *Technometrics*, 47(3): 264–273, 2005.

McCoy, Michael, Tropp, Joel A, et al. Two proposals for robust pca using semidefinite programming. *Electronic Journal of Statistics*, 5:1123–1160, 2011.

Netrapalli, Praneeth, Niranjan, UN, Sanghavi, Sujay, Anandkumar, Animashree, and Jain, Prateek. Nonconvex robust pca. In *Advances in Neural Information Processing Systems*, pp. 1107–1115, 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. Coherence pursuit: Fast, simple, and robust principal component analysis. *arXiv preprint arXiv:1609.04789*, 2016.

Rahmani, Mostafa and Atia, George. Randomized robust subspace recovery and outlier detection for high dimensional data matrices. *IEEE Transactions on Signal Processing*, 65(6):1580–1594, March 2017.

Soltanolkotabi, Mahdi and Candes, Emmanuel J. A geometric analysis of subspace clustering with outliers. *The Annals of Statistics*, pp. 2195–2238, 2012.

Tron, Roberto and Vidal, René. A benchmark for the comparison of 3-d motion segmentation algorithms. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–8. IEEE, 2007.

Tsakiris, Manolis C and Vidal, René. Dual principal component pursuit. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 10–18, 2015.

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

Xu, Huan, Caramanis, Constantine, and Mannor, Shie. Principal component analysis with contaminated data: The high dimensional case. *arXiv preprint arXiv:1002.4658*, 2010a.

Xu, Huan, Caramanis, Constantine, and Sanghavi, Sujay. Robust pca via outlier pursuit. In *Advances in Neural Information Processing Systems*, pp. 2496–2504, 2010b.

Yi, Xinyang, Park, Dohyung, Chen, Yudong, and Caramanis, Constantine. Fast algorithms for robust pca via gradient descent. *arXiv preprint arXiv:1605.07784*, 2016.

Zhang, Teng. Robust subspace recovery by geodesically convex optimization. *arXiv preprint arXiv:1206.1386*, 2012.

Zhang, Teng and Lerman, Gilad. A novel m-estimator for robust pca. *The Journal of Machine Learning Research*, 15(1):749–808, 2014.