Provable Self-Representation Based Outlier Detection in a Union of Subspaces

Chong You, Daniel P. Robinson, René Vidal
Johns Hopkins University, Baltimore, MD, 21218, USA

Abstract

Many computer vision tasks involve processing large amounts of data contaminated by outliers, which need to be detected and rejected. While outlier detection methods based on robust statistics have existed for decades, only recently have methods based on sparse and low-rank representation been developed along with guarantees of correct outlier detection when the inliers lie in one or more low-dimensional subspaces. This paper proposes a new outlier detection method that combines tools from sparse representation with random walks on a graph. By exploiting the property that data points can be expressed as sparse linear combinations of each other, we obtain an asymmetric affinity matrix among data points, which we use to construct a weighted directed graph. By defining a suitable Markov Chain from this graph, we establish a connection between inliers/outliers and essential/inessential states of the Markov chain, which allows us to detect outliers by using random walks. We provide a theoretical analysis that justifies the correctness of our method under geometric and connectivity assumptions. Experimental results on image databases demonstrate its superiority with respect to state-of-the-art sparse and low-rank outlier detection methods.

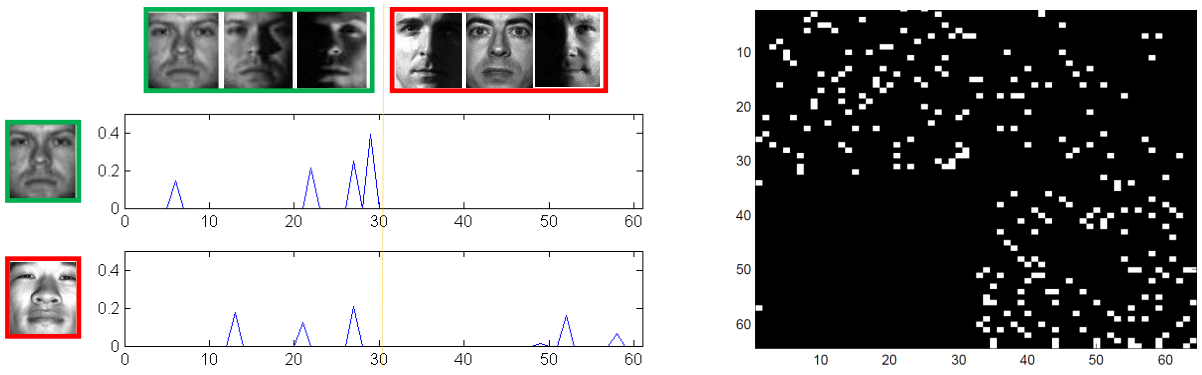
1. Introduction

In many applications in computer vision, including motion estimation and segmentation [18] and face recognition [2], high-dimensional datasets can be well approximated by a union of low-dimensional subspaces. Such applications have motivated a lot of research on the problems of learning one or more subspaces from data, a.k.a. subspace learning and subspace clustering, respectively. In practice, datasets are often contaminated by points that do not lie in the subspaces, i.e. outliers. In such situations, it is often essential to detect and reject these outliers before any subsequent processing/analysis is performed.

Prior work. We address the problem of outlier detection in the setting when the inlier data are assumed to lie close to a union of unknown low-dimensional subspaces (low relative to the dimension of the ambient space). A traditional method for solving this problem is RANSAC [12], which is based on randomly selecting a subset of points, fitting a subspace to them, and counting the number of points that are well fit by this subspace; this process is repeated for sufficiently many trials and the best fit is chosen. RANSAC is intrinsically combinatorial and the number of trials needed to find a good estimate of the subspace grows exponentially with the subspace dimension. Consequently, the methods of choice have been to robustly learn the subspaces by penalizing the sum of *unsquared* distances (in lieu of *squared* distances used in classical methods such as PCA) of points to the closest subspace [9, 21, 54, 53]. Such a penalty is robust to outliers because it reduces the contributions from large residuals arising from outliers. However, the optimization problem is usually nonconvex and a good initialization is extremely important for finding the optimal solution.

The groundbreaking work of Wright et al. [47] and Candès et al. [4] on using convex optimization techniques to solve the PCA problem with robustness to corrupted entries has led to many recent methods for PCA with robustness to outliers [48, 28, 23, 52, 20]. For example, Outlier Pursuit [48] uses the nuclear norm $\|\cdot\|_*$ to seek low-rank solutions by solving the problem $\min_L \|X-L\|_{2,1}+\lambda\|L\|_*$ for some $\lambda > 0$. A prominent advantage of convex optimization techniques is that they are guaranteed to correctly identify outliers under certain conditions. Very recently, several nonconvex outlier detection methods have also been developed with guaranteed correctness [19, 6]. Nonetheless, these methods typically model a *unique* inlier subspace, e.g., by a low rank matrix L in Outlier Pursuit, and therefore cannot deal with multiple inlier subspaces since the union of multiple subspaces could be high-dimensional.

Another class of methods with theoretical guarantees for correctness utilizes the fact that outliers are expected to have low similarities with other data points. In [5, 1], a multi-way similarity is introduced that is defined from the polar curvature, which has the advantage of exploiting the subspace structure. However, the number of combinations in multi-way similarity can be prohibitively large. Some recent works have explored using inner products between data points for outlier detection [16, 35]. Although computationally very efficient, these methods require the inliers to be well distributed and densely sampled within the subspaces.



(a) Two exemplar representation vectors (b) Representation matrix R

Figure 1. An illustration of a self-representation matrix R in the presence of outliers. The first 32 columns of the data matrix X correspond to 32 images of one individual under different illuminations from the Extended Yale B database, and the next 32 images are randomly chosen from all other individuals; three examples from each category are shown near the top of 1(a). We also show a typical representation vector for an inlier and an outlier image in 1(a), and the complete representation matrix R in 1(b), where white and black denote $r_{ij} \neq 0$ and $r_{ij} = 0$. Notice that inliers use only other inliers in their representation, while outliers use both inliers and outliers in their representations.

Overview of our method and contributions. In this work, we address the problem of outlier detection by using data self-representation. The proposed approach builds on the self-expressiveness property of data in a union of low-dimensional subspaces, originally introduced in [10], which states that a point in a subspace can always be expressed as a linear combination of other points in the subspace. In particular, if the columns of $X = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N]$ lie in multiple subspaces, then for all $j = 1, \ldots, N$, there exists a vector $\boldsymbol{r}_j \in I\!R^N$ such that $\boldsymbol{x}_j = X\boldsymbol{r}_j$ and the nonzero entries of $\boldsymbol{r}_j$ correspond to points in the same subspace as $\boldsymbol{x}_j$. If the subspace dimensions are small, $\boldsymbol{r}_j$ can be taken to be sparse and be computed by solving the ℓ_1 minimization problem

$$\min_{\boldsymbol{r}_j} \|\boldsymbol{r}_j\|_1 + \frac{\gamma}{2}\|\boldsymbol{x}_j - X\boldsymbol{r}_j\|_2^2 \quad \text{s.t. } r_{jj} = 0 \qquad (1)$$

for some $\gamma > 0$. In [10], an *undirected* graph is constructed from $R = [\boldsymbol{r}_1, \cdots, \boldsymbol{r}_N]$ in which each vertex corresponds to a data point, and vertices corresponding to $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are connected if either r_{ij} or r_{ji} is nonzero. Such a graph can be used to segment the data into their respective subspaces by applying spectral clustering [41] to the graph's Laplacian.

Consider now the case where X contains outliers to the subspaces. Figure 1 illustrates an example representation matrix R computed from (1) for data drawn from a single subspace (face images from one individual) plus outliers (other images). In this case, the representation R is such that inliers express themselves as linear combinations of a few other inliers, while outliers express themselves as linear combinations of both inliers and outliers. Motivated by this observation, we use a *directed* graph to model data relations: a directed edge from $\boldsymbol{x}_j$ to $\boldsymbol{x}_i$ indicates that $\boldsymbol{x}_j$ uses $\boldsymbol{x}_i$ in its representation (i.e. $r_{ij} \neq 0$). Then a random walk on the representation graph initialized at an outlier will not return to the set of outliers since once the random walk reaches an inlier it cannot return to the outliers. Therefore, we design a random walk process and identify outliers as those whose probabilities tend to zero. Our work makes the following contributions with respect to the state of the art:

1. Our method can detect outliers using the probability distribution of a *random walk* on a graph constructed from *data self-representation*.
2. Our *data self-representation* allows our method to handle multiple inlier subspaces. Knowledge of the number of subspaces and their dimensions is not required, and the subspaces may have a nontrivial intersection.
3. Our method can explore contextual information by using a *random walk*, i.e., the "outlierness" of a particular point depends on the "outlierness" of its neighbors.
4. Our analysis shows that our method correctly identifies outliers under suitable assumptions on the data distribution and connectivity of the representation graph.
5. Experiments on real image databases illustrate the effectiveness of our method.

2. Related work

Outlier detection by self-representation. Prior work has explored using data self-representation as a tool for outlier detection in a union of subspaces. Specifically, motivated by the observation that outliers do not have *sparse* representations, [37, 8] declare a point $\boldsymbol{x}_j$ as an outlier if $\|\boldsymbol{r}_j\|_1$ is above a threshold. However, this ℓ_1-thresholding strategy is not robust to outliers that are close to each other since their representation vectors may have small ℓ_1-norms. The LRR [25] solves for a low-rank self-representation matrix R in lieu of a sparse representation and penalizes the sum of unsquared self-representation errors $\|\boldsymbol{x}_j - X\boldsymbol{r}_j\|_2$, which

makes it more robust to outliers. However, LRR requires the subspaces to be independent and the sum of the union of subspaces to be low-dimensional [26].

Outlier detection by maximum consensus. In a diverse range of contexts such as maximum consensus [55, 7] and robust linear regression [29, 42], people have studied problems of the form

$$\min_{\mathbf{b}} \sum_{i=1}^{N} \mathbb{I}(|\boldsymbol{x}_i^\top \mathbf{b} - y_i| \geq \epsilon), \tag{2}$$

in which $\mathbb{I}(\cdot)$ is the indicator function. Note that if we set $y_i = 1$ for all i, then (2) can be interpreted as detecting outliers in data X where the inliers lie close to an *affine* hyperplane. A problem closely related to (2) is

$$\min_{\mathbf{b}} \sum_{i=1}^{N} \mathbb{I}(|\boldsymbol{x}_i^\top \mathbf{b}| \geq \epsilon) \;\text{ s.t. }\; \mathbf{b} \neq 0, \tag{3}$$

which appears in many applications (e.g. see [34]). In particular, (3) can be used to learn a *linear* hyperplane from data corrupted by outliers. To detect outliers in a general low-dimensional subspace, one can apply (2) and (3) recursively to find a basis for the orthogonal complement of the subspace [39]. However, such an approach is limited because there can be only one inlier subspace and the dimension of that subspace must be known in advance.

Outlier detection by random walk. Perhaps the most well-known random walk based algorithm is PageRank [3]. Originally introduced to determine the authority of website pages from web graphs, PageRank and its variants have been used in different contexts for ranking the centrality of the vertices in a graph. In particular, [30, 31] propose the OutRank, which ranks the "outlierness" of points in a dataset by applying PageRank to an undirected graph in which the weight of an edge is the cosine similarity or RBF similarity between the two connected data points. Then, points that have low centrality are regarded as outliers. The outliers returned by OutRank are those that have low similarity to other data points. Therefore, OutRank does not work if points in a subspace are not dense enough.

3. Outlier detection by self-representation

In this section, we present our data self-representation based outlier detection method. We first describe the data self-representation and its associated properties for inliers and outliers. We then design a random walk algorithm on the representation graph whose limiting behavior allows us to identify the sets of inliers and outliers.

3.1. Data self-representation

Given an unlabeled dataset $X = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N]$ containing inliers and outliers, the first step of our algorithm is to construct the data self-representation matrix denoted by $R = [\boldsymbol{r}_1, \cdots, \boldsymbol{r}_N]$. As briefly discussed in the introduction (see also Figure 1), a self-representation matrix R computed from (1) is observed to have different properties for inliers and outliers. Specifically, inliers usually use only other inliers for self-representation, i.e. for an inlier $\boldsymbol{x}_j$, the representation is such that $r_{ij} \neq 0$ only if $\boldsymbol{x}_i$ is also an inlier, where r_{ij} is the (i,j)-th entry of R. This property is expected to hold if the inliers lie in a union of low dimensional subspaces, as evidenced from the works [11, 37, 51, 45, 43]. As an intuitive explanation, if the inliers lie in a low dimensional subspace, then any inlier has a *sparse* representation using other points in this subspace. Thus such a representation can be found by using sparsity-inducing regularization as seen in (1). In contrast, outliers are generally randomly distributed in the ambient space, so that a self-representation usually contains both inliers and outliers.

Since the representation R computed from (1) is sparse, there are potentially connectivity issues in the representation graph, i.e. an inlier that is not well-connected to other inliers may be detected as an outlier, and an outlier that is not well connected may be detected as an inlier. To address the connectivity issue, we compute the data self-representation matrix R by the elastic net problem [56, 49]:

$$\min_{\boldsymbol{r}_j} \lambda\|\boldsymbol{r}_j\|_1 + \frac{1-\lambda}{2}\|\boldsymbol{r}_j\|_2^2 + \frac{\gamma}{2}\|\boldsymbol{x}_j - X\boldsymbol{r}_j\|_2^2 \;\text{ s.t. } r_{jj} = 0, \tag{4}$$

in which $\lambda \in [0,1]$ controls the balance between sparseness (via ℓ_1 regularization) and connectivity (via ℓ_2 regularization). Specifically, if λ is chosen close to 1, we can still expect that the computed representation for an inlier will only use inliers. The ℓ_2 regularization has been introduced to promote more connections between data points, i.e. if $\lambda \in [0,1)$, then one expects more nonzero entries in R. A detailed discussion of the representation computed from (4) and the connectivity issue is provided in Section 4.

3.2. Representation graph and random walk

We use a directed graph G, which we call a *representation graph*, to capture the behavior of inliers and outliers from the representation matrix R. The vertices of G correspond to the data points X, and the edges are given by the (weighted) adjacency matrix $A := |R|^\top \in \mathbb{R}^{N\times N}$ with the absolute value taken elementwise, i.e., the weight of the edge from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ is given by $a_{ij} = |r_{ji}|$. In the representation graph, we expect that vertices corresponding to inliers will have edges that only lead to inliers, while vertices that are outliers will have edges that lead to both inliers and outliers. In other words, we do not expect to have any edges that lead from an inlier to an outlier.

Using the previous paragraph as motivation, we design a random walk procedure to identify the outliers. A random walk on the representation graph G is a discrete time

Markov chain, for which the transition probability from $\boldsymbol{x}_i$ at a given time to $\boldsymbol{x}_j$ at the next time is given by $p_{ij} := a_{ij}/d_i$ with $d_i := \sum_j a_{ij}$. By this definition, if the starting point of a random walk is an inlier then it will never escape the set of inliers as there is no edge going from any inlier to any outlier. In contrast, a random walk starting from an outlier will likely end up in an inlier state since once it enters any inlier it will never return to an outlier state. Thus, by using different data points to initialize random walks, outliers can be identified by observing the final probability distribution of the state of the random walks (see Figure 2).

If $P \in \mathbb{R}^{N\times N}$ is the transition matrix with entries p_{ij}, then P is related to the representation matrix R by

$$p_{ij} = |r_{ji}|/\|\boldsymbol{r}_i\|_1 \quad \text{for all} \quad \{i,j\} \subset \{1,2,\cdots N\}. \tag{5}$$

We define $\boldsymbol{\pi}^{(t)} = [\pi_1^{(t)}, \ldots, \pi_N^{(t)}]$ to be the state probability distribution at time t, then the state transition is given by $\boldsymbol{\pi}^{(t+1)} = \boldsymbol{\pi}^{(t)} P$. Thus, a t-step transition is $\boldsymbol{\pi}^{(t)} = \boldsymbol{\pi}^{(0)} P^t$ with $\boldsymbol{\pi}^{(0)}$ the chosen initial state probability distribution.

3.3. Main algorithm: Outlier detection by R-graph

We propose to perform outlier detection by using random walks on the representation graph G. We set the initial probability distribution as $\boldsymbol{\pi}^{(0)} = [1/N, \cdots, 1/N]$, and then compute the t-step transition $\boldsymbol{\pi}^{(t)} = \boldsymbol{\pi}^{(0)} P^t$. This can be interpreted as initializing a random walk from each of the N data points, and then finding the sum of probability distributions of all random walks after t steps. It is expected that all random walks—starting from either an inlier or an outlier—will eventually have high probabilities for the inlier states and low probabilities for the outlier states.

We note that the $\boldsymbol{\pi}^{(t)}$ defined as above need not converge, as shown by the 2-dimensional example $P = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$. Instead, we choose to use the T-step Cesàro mean, given by

$$\bar{\boldsymbol{\pi}}^{(T)} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\pi}^{(0)} P^t \equiv \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\pi}^{(t)}, \tag{6}$$

which is the average of the first T t-step probability distributions (see Figure 2). The sequence $\{\bar{\boldsymbol{\pi}}^{(T)}\}$ has the benefit that it always converges, and its limit is the same as that of $\boldsymbol{\pi}^{(t)}$ whenever the latter exists. In the next section, we give a more detailed discussion of this choice, its properties for outlier detection, and its convergence behavior.

Our complete algorithm is stated as Algorithm 1.

Algorithm 1 Outlier detection by representation graph

Input: Data $X = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N]$, #iterations T, threshold ϵ.

1: Use X to solve for $R = [\boldsymbol{r}_1, \cdots, \boldsymbol{r}_N]$ using (4).
2: Compute P from R using (5).
3: Initialize $t = 0$, $\boldsymbol{\pi} = [1/N, \cdots, 1/N]$, and $\bar{\boldsymbol{\pi}} = \mathbf{0}$.
4: **for** $t = 1, 2, \ldots T$ **do**
5: Compute $\boldsymbol{\pi} \leftarrow \boldsymbol{\pi} \cdot P$, and then set $\bar{\boldsymbol{\pi}} \leftarrow \bar{\boldsymbol{\pi}} + \boldsymbol{\pi}$.
6: **end for**
7: $\bar{\boldsymbol{\pi}} \leftarrow \bar{\boldsymbol{\pi}}/T$.

Output: An indicator of outliers: $\boldsymbol{x}_j$ is an outlier if $\bar{\boldsymbol{\pi}}_j \le \epsilon$.

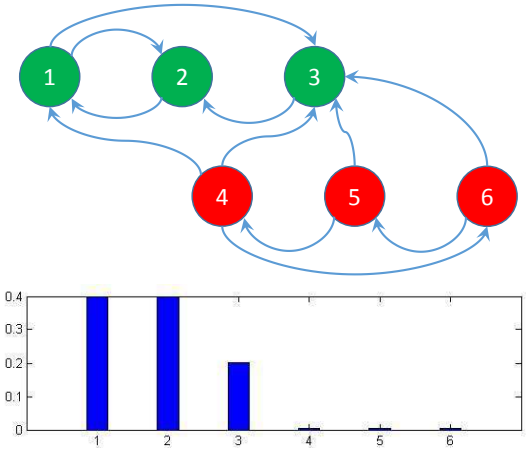


Figure 2. Illustration of random walks on a representation graph. Top: green balls represent inliers and red balls represent outliers, and arrows represent edges among nodes. Notice that there is no edge going from inliers to outliers. A random walk starting from any point will end up at only inlier points. Bottom: bar plot of $\bar{\boldsymbol{\pi}}^{(100)}$ with the ith bar corresponding to the ith entry in $\bar{\boldsymbol{\pi}}^{(100)}$. The use of thresholding on this probability distribution will correctly distinguish outliers from inliers.

4. Theoretical guarantees for correctness

Let us first formally define the problem of outlier detection when data is drawn from a union of subspaces.

Problem 1 (Outlier detection in a union of subspaces) *Given data $X = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N] \in \mathbb{R}^{D\times N}$ whose columns contain inliers that are drawn from an unknown number of unknown subspaces $\{\mathcal{S}_\ell\}_{\ell=1}^n$, and outliers that are outside of $\cup_{\ell=1}^n \mathcal{S}_\ell$, the goal is to identify the set of outliers.*

Recall that motivation for our method is that ideally there will be no edge going from an inlier to an outlier in the representation graph. This motivates us to assume that a random walk starting at any inlier will eventually return to itself, i.e. inliers are *essential states* of the Markov chain, while outliers are those that have a chance of never coming back to itself, i.e. outliers are *inessential states*. Formally, we work with a (time homogeneous) Markov chain with state space $\Omega = \{1, \cdots, N\}$, in which each state j corresponds to data $\boldsymbol{x}_j$, and the transition probability P is given by (5). Given $\{i,j\} \subset \Omega$, we say that j is accessible from i, denoted as $i \to j$, if there exists some $t > 0$ such that the (i,j)-th entry of P^t is positive. Intuitively, $i \to j$ if a random walk can move from i to j in finitely many steps.

Definition 1 (Essential and inessential state [22]) *A state $i \in \Omega$ is essential if for all j such that $i \to j$ it is also true that $j \to i$. A state is inessential if it is not essential.*

Our aim in this section is to establish that if inliers connect to themselves, i.e. they are *subspace-preserving* (Section 4.1), and the representation R satisfies certain connectivity conditions (Section 4.2), then inliers are essential states of the Markov chain and outliers are inessential states. Subsequently, in Section 4.3 we show that the Cesàro mean (6) identifies essential and inessential states, thus establishing the correctness of Algorithm 1 for outlier detection.

4.1. Subspace-preserving representation

We first establish that inliers express themselves with only other inliers when they lie in a union of low dimensional subspaces. This property is well-studied in the subspace clustering literature. We will borrow terminologies and results from prior work and modify them for our current task of outlier detection.

Definition 2 (Subspace-preserving representation [40]) *If $\boldsymbol{x}_j \in \mathcal{S}_\ell$ is an inlier, then the representation $\boldsymbol{r}_j \in I\!R^N$ is called subspace-preserving if the nonzero entries of $\boldsymbol{r}_j$ correspond to points in $\mathcal{S}_\ell$, i.e. $r_{ij} \neq 0$ only if $\boldsymbol{x}_i \in \mathcal{S}_\ell$. The representation matrix $R = [\boldsymbol{r}_1, \cdots, \boldsymbol{r}_N] \in I\!R^{N \times N}$ is called subspace-preserving if $\boldsymbol{r}_j$ is subspace-preserving for every inlier $\boldsymbol{x}_j$.*

A representation matrix R is subspace-preserving if each inlier uses points in its own subspace for representation. Given X, a subspace-preserving representation R can be obtained by solving (4) when certain geometric conditions hold. The following result is modified from [49]. It assumes that columns of X are normalized to have unit ℓ_2-norm.

Theorem 1 *Let $\boldsymbol{x}_j \in \mathcal{S}_\ell$ be an inlier. Define the oracle point of $\boldsymbol{x}_j$ to be $\boldsymbol{\delta}_j := \gamma \cdot (\boldsymbol{x}_j - X^\ell_{-j} \cdot \boldsymbol{r}^\ell_j)$, where X^ℓ_{-j} is the matrix containing all points in $\mathcal{S}_\ell$ except $\boldsymbol{x}_j$ and*

$$\boldsymbol{r}^\ell_j := \arg\min_{\boldsymbol{r}} \lambda \|\boldsymbol{r}\|_1 + \frac{1-\lambda}{2}\|\boldsymbol{r}\|_2^2 + \frac{\gamma}{2}\|\boldsymbol{x}_j - X^\ell_{-j}\boldsymbol{r}\|_2^2.$$

The solution $\boldsymbol{r}_j$ to (4) is subspace-preserving if

$$\max_{k \neq j, \boldsymbol{x}_k \in \mathcal{S}_\ell} |\langle \boldsymbol{x}_k, \bar{\boldsymbol{\delta}}_j \rangle| - \max_{k: \boldsymbol{x}_k \notin \mathcal{S}_\ell} |\langle \boldsymbol{x}_k, \bar{\boldsymbol{\delta}}_j \rangle| > \frac{1-\lambda}{\lambda}, \quad (7)$$

where $\bar{\boldsymbol{\delta}}_j := \boldsymbol{\delta}_j / \|\boldsymbol{\delta}_j\|_2$.

We provide an outline of the proof in [50]. Note that the oracle point $\boldsymbol{\delta}_j$ lies in $\mathcal{S}_\ell$ and that its definition only depends on points in $\mathcal{S}_\ell$. The first term in condition (7) captures the distribution of points in $\mathcal{S}_\ell$ near $\bar{\boldsymbol{\delta}}_j$, and is expected to be large if the neighborhood of $\bar{\boldsymbol{\delta}}_j$ is well-covered by points from $\mathcal{S}_\ell$. The second term characterizes the similarity between the oracle point $\bar{\boldsymbol{\delta}}_j$ and all other data points, which includes the outliers and the inliers from other subspaces. The condition requires the former to be larger than the latter by a margin of $\frac{1-\lambda}{\lambda}$, which is close to zero if λ is close to 1. Overall, condition (7) requires that points in $\mathcal{S}_\ell$ are dense around $\bar{\boldsymbol{\delta}}_j$, which is itself in $\mathcal{S}_\ell$, and that outliers and inliers from other subspaces do not lie close to $\bar{\boldsymbol{\delta}}_j$.

Even if (7) holds for all j so that the representation R is subspace-preserving, we cannot automatically establish an equivalence between inliers/outliers and essential/inessential states because of potential complications related to the graph's *connectivity*. This is addressed next.

4.2. Connectivity considerations

In the context of sparse subspace clustering, the well-known connectivity issue [32, 46, 27, 49, 44] refers to the problem that points in the same subspace may not be well-connected in the representation graph, which may cause oversegmentation of the true clusters. Thus, one has to make the assumption that each true cluster is connected to guarantee correct clustering. For the outlier detection problem, it may happen that an inlier is inessential and thus classified as an outlier when the inliers are not well-connected; similarly, an outlier may be essential and thus classified as an inlier if it is not connected to at least one inlier. In fact, the situation is even more involved since the representation graph is directed and inliers and outliers behave differently.

Suppose, as a first example, that there exists an inlier that is never used to express any other inliers. This is equivalent to saying that there is no edge going into this point from any other inliers. Note that the subspace-preserving property can still hold if this inlier expresses itself using other inliers. Yet, since a random walk leaving this point would never return it can not be identified as an inlier. To avoid such cases, we need the following assumption.

Assumption 1 *For any inlier subspace $\mathcal{S}_\ell$, the vertices $\{\boldsymbol{x}_j \in \mathcal{S}_\ell\}$ of the representation graph are strongly connected, i.e. there is a path in each direction between each pair of vertices.*

Assumption 1 requires good connectivity between points from the same inlier subspace. We also need good connectivity between outliers and inliers. Consider the example when there is a subset of outliers for which all of their outgoing edges lead only to points within that same subset. In this case, the subset of points can not be detected as outliers since their representation pattern is the same as for the inliers. The next assumption rules out this case.

Assumption 2 *For each subset of outliers there exists an edge in the representation graph that goes from a point in this subset to an inlier or to an outlier outside this subset.*

4.3. Main theorem: guaranteed outlier detection

We can now establish guaranteed outlier detection by our representation graph based method stated as Algorithm 1.

Theorem 2 *If the representation R is subspace-preserving and satisfies Assumptions 1 and 2, then Algorithm 1 with $T = \infty$ and $\epsilon = 0$ correctly identifies outliers.*

Theorem 2 is a direct consequence of the following two facts (see [50] for the proof).

Lemma 1 *If the representation R is subspace-preserving and Assumptions 1 and 2 hold, then inliers and outliers correspond to essential and inessential states, respectively.*

Lemma 2 *For any probability transition matrix P, the averaged probability distribution in (6) satisfies $\lim_{T\to\infty} \bar{\boldsymbol{\pi}}^{(T)} = \boldsymbol{\pi}$, where $\boldsymbol{\pi}$ is such that $\pi_j = 0$ if and only if state j is inessential.*

Theorem 2 shows that Problem 1 is solved by Algorithm 1 if the data X satisfies the geometric conditions in (7) and the representation graph satisfies the required connectivity assumptions.

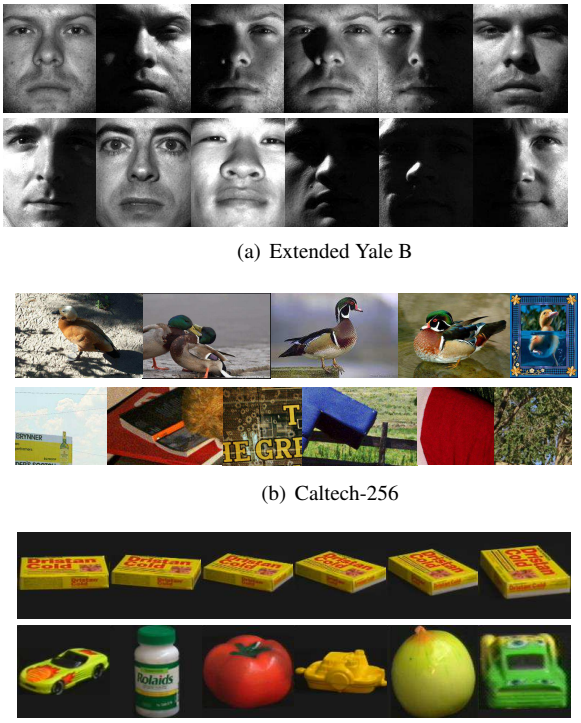
We note that the random walk by the Cesàro mean adopted here is different from the popular random walk with restart as adopted by PageRank, for example. The benefit of PageRank is that the random walk converges to the unique stationary distribution. However, it is not clear whether this stationary distribution identifies the outliers. In fact, all states in the random walk of PageRank are essential, so that outliers do not converge to zero probabilities. In contrast, the random walk in our method does not necessarily have a unique stationary distribution, but the Cesàro mean does converge to one of the stationary distributions, which we have shown can be used to identify outliers. A detailed discussion is provided in [50].

5. Experiments

We use several image databases (see Figure 3) to evaluate our outlier detection method (Algorithm 1). For computing the representation $\boldsymbol{r}_j$ in (4), we use the solver in [17] with $\lambda = 0.95$ and $\gamma = \alpha \cdot \frac{\lambda}{\max_{i:i\neq j} |\boldsymbol{x}_j^\top \boldsymbol{x}_i|}$, where α is a parameter tuned to each dataset. In particular, the solution to (4) is nonzero if and only if $\alpha > 1$. The number of iterations T is set to be $1{,}000$.

5.1. Experimental setup

Databases. We construct outlier detection tasks from three publicly available databases. The Extended Yale B [14] dataset contains frontal face images of 38 individuals each under 64 different illumination conditions. The face images are of size 192×168, for which we downsample to 48×42. The Caltech-256 [15] is a database that contains images from 256 categories that have more than 80 images each. There is also an additional "clutter" category in this database that contains 827 images of different varieties, which are used as outliers. The Coil-100 dataset [33] contains 7,200 images of 100 different objects. Each object has 72 images taken at pose intervals of 5 degrees, with the images being of size 32×32. For the Extended Yale B and Coil-100 datasets we use raw pixel intensity as the feature representation. Images in Caltech-256 are represented by a 4,096-dimensional feature vector extracted from the last fully connected layer of the 16-layer VGG network [36].



(a) Extended Yale B

(b) Caltech-256

(c) Coil-100

Figure 3. Examples of data used for outlier detection. For each database, the top row shows examples of the inlier set and the bottom row shows examples from the outlier set.

Baselines. We compare with 6 other representative methods that are designed for detecting outliers in one or multiple subspaces: CoP [35], OutlierPursuit [48], REAPER [20], DPCP [39], LRR [25] and ℓ_1-thresholding [37]. We also compare with a graph based method: OutRank [30, 31]. We implement the inexact ALM [24] for solving the optimization in OutlierPursuit. For LRR, we use the code available online at https://sites.google.com/site/guangcanliu/. For DPCP, we use the code provided by the authors. All other methods are implemented according to the description in their respective papers.

Evaluation metric. Each outlier detection method gener-

Table 1. Results on the Extended Yale B database. Inliers are taken to be the images of either one or three randomly chosen subjects, and outliers are randomly chosen from the other subjects (at most one from each subject). For R-graph we set $\alpha = 5$ in the definition of γ.

	OutRank	CoP	REAPER	OutlierPursuit	LRR	DPCP	ℓ_1-thresholding	R-graph (ours)
*Inliers: all images from **one** subject Outliers: 35%, taken from other subjects*								
AUC	0.536	0.556	0.964	0.972	0.857	0.952	0.844	**0.986**
F1	0.552	0.563	0.911	0.918	0.797	0.885	0.763	**0.951**
*Inliers: all images from **three** subjects Outliers: 15%, taken from other subjects*								
AUC	0.519	0.529	0.932	0.968	0.807	0.888	0.848	**0.985**
F1	0.288	0.292	0.758	0.856	0.509	0.653	0.545	**0.878**

ates a numerical value for each data point that indicates its "outlierness", and a threshold value is required for determining inliers and outliers. A Receiver Operating Characteristic (ROC) curve plots the true positive rate and false positive rate for all threshold values. We use the area under the curve (AUC) as a metric of performance in terms of the ROC. The AUC is always between 0 and 1, with a perfect model having an AUC of 1 and a model that guesses randomly having an AUC of approximately 0.5.

As a second metric, we provide the F1-score, which is the harmonic mean of precision and recall. The F1-score is dependent upon the threshold, and we report the largest F1-score across all thresholds. An F1-score of 1 means there exists a threshold that gives both precision and recall equal to 1, i.e. a perfect separation of inliers and outliers.

The reported numbers for all experiments discussed in this section are the averages over 50 trials.

5.2. Outliers in face images

Suppose we are given a set of images of one or more individuals but that the data set is also corrupted by face images of a variety of other individuals. The task is to detect and remove those outlying face images. It is known that images of a face under different lighting conditions lie approximately in a low dimensional subspace. Thus, this task can be modeled as the problem of outlier detection in one subspace or in a union of subspaces.

We use the extended Yale B database. In the first experiment, we randomly choose a single individual from the 38 subjects and use all 64 images of this subject as the inliers. We then choose images from the remaining 37 subjects as outliers with at most one image from each subject. The overall data set has 25% outliers. The average AUC and F1 measures over 50 trials are reported in Table 1. For a fair comparison, we fine-tuned the parameters for all methods.

Comparing to state of the art. We see that our representation graph based method R-graph outperforms the other methods. Besides our method, the REAPER, Outlier Pursuit and DPCP algorithms all perform well. These three methods learn a single subspace and treat those that do not fit the subspace as outliers, thus making them well suited for this data (the images of one individual can be well-approximated by a single low dimensional subspace).

The LRR and ℓ_1-thresholding methods use data self-representation, which is also the case for our method. However, LRR does not give good outlier detection results, probably because its algorithm for solving the LRR model is not guaranteed to converge to a global optimum. The ℓ_1-thresholding also does not give good results, showing that the magnitude of the representation vector is not a robust measure for classifying outliers. By considering the connection patterns in the representation graph, our method achieves significantly better results.

The performance of OutRank and CoP is significantly worse than that of the other methods. This poor performance can be explained by the use of a coherence-based distance, which fails to capture similarity between data points when the data lie in subspaces. For example, it can be argued that the coherence between two faces with the same illumination condition can be higher than two images of the same face under different illumination conditions.

Dealing with multiple inlier groups. In order to test the ability of the methods to deal with multiple inlier groups, we designed a second experiment in which inliers are taken to be images of 3 randomly chosen subjects, and outliers are randomly drawn from other subjects as before. For all methods, we use the same parameters as in the previous experiment to test the robustness to parameter tuning. The results of this experiment are reported in Table 1.

We can see that Outlier Pursuit and our R-graph are the two best methods. Although Outlier Pursuit only models a single low dimensional subspace, it can still deal with this data since the union of the three subspaces corresponding to the three subjects in the inlier set is still low dimensional and can be treated as a single low dimensional subspace. However, we postulate that Outlier Pursuit will eventually fail as we increase the number of inlier groups, since the union of low dimensional subspaces will no longer be low rank. Our method does not have this limitation.

Similar to Outlier Pursuit, both REAPER and DPCP can, in principle, handle multiple inlier groups by fitting a single subspace to their union. However, REAPER and DPCP require as input the dimension of the union of the inlier subspaces, which can be hard to estimate in practice. Indeed, in Table 1, we observe that the performances of REAPER and DPCP are less competitive in comparison to Outlier Pursuit

Table 2. Results on the Caltech-256 database. Inliers are taken to be images of one, three, or five randomly chosen categories, and outliers are randomly chosen from category 257-clutter. For R-graph we set $\alpha = 20$ in the definition of γ.

	OutRank	CoP	REAPER	OutlierPursuit	LRR	DPCP	ℓ_1-thresholding	R-graph (ours)
Inliers: ***one*** *category of images* *Outliers:* 50%								
AUC	0.897	0.905	0.816	0.837	0.907	0.783	0.772	**0.948**
F1	0.866	0.880	0.808	0.823	0.893	0.785	0.772	**0.914**
Inliers: ***three*** *categories of images* *Outliers:* 50%								
AUC	0.574	0.676	0.796	0.788	0.479	0.798	0.810	**0.929**
F1	0.682	0.718	0.784	0.779	0.671	0.777	0.782	**0.880**
Inliers: ***five*** *categories of images* *Outliers:* 50%								
AUC	0.407	0.487	0.657	0.629	0.337	0.676	0.774	**0.913**
F1	0.667	0.672	0.716	0.711	0.667	0.715	0.762	**0.858**

Table 3. Results on the Coil-100 database. Inliers are taken to be images of one, four, or seven randomly chosen categories, and outliers are randomly chosen from other categories (at most one from each category). For R-graph we set $\alpha = 10$ in the definition of γ.

	OutRank	CoP	REAPER	OutlierPursuit	LRR	DPCP	ℓ_1-thresholding	R-graph (ours)
Inliers: all images from ***one*** *category* *Outliers:* 50%								
AUC	0.836	0.843	0.900	0.908	0.847	0.900	0.991	**0.997**
F1	0.862	0.866	0.892	0.902	0.872	0.882	0.978	**0.990**
Inliers: all images from ***four*** *categories* *Outliers:* 25%								
AUC	0.613	0.628	0.877	0.837	0.687	0.859	0.992	**0.996**
F1	0.491	0.500	0.703	0.686	0.541	0.684	0.941	**0.970**
Inliers: all images from ***seven*** *categories* *Outliers:* 15%								
AUC	0.570	0.580	0.824	0.822	0.628	0.804	0.991	**0.996**
F1	0.342	0.346	0.541	0.528	0.366	0.511	0.897	**0.955**

and our R-graph for the three subspace case.

5.3. Outliers in images of objects

We test the ability of the methods to identify one or several object categories that frequently appear in a set of images amidst outliers that consist of objects that rarely occur. For Caltech-256, images in $n \in \{1, 3, 5\}$ randomly chosen categories are used as inliers in three different experiments. From each category, we use the first 150 images if the category has more than 150 images. We then randomly pick a certain number of images from the "clutter" category as outliers such that there are 50% outliers in each experiment. For Coil-100, we randomly pick $n \in \{1, 4, 7\}$ categories as inliers and pick at most one image from each of the remaining categories as outliers.

The results are reported in Table 2 and Table 3. We see that our R-graph method achieves the best performance. The two geometric distance based methods, OutRank and CoP, achieve good results when there is one inlier category, but deteriorate when the number of inlier categories increases. The performance of REAPER, Outlier Pursuit and DPCP are similar to each other and worse than our method. This may be because they all try to fit a linear subspace to the data, while the data in these two databases may be better modeled by a nonlinear manifold. The ℓ_1-thresholding and the representation graph method are all based on data self-expression, and seem to be more powerful for this data.

6. Conclusion

We presented an outlier detection method that combined data self-representation and random walks on a representation graph. Unlike many prior methods for robust PCA, our method is able to deal with multiple subspaces and does not require the number of subspaces or their dimensions to be known. Our analysis showed that the method is guaranteed to identify outliers when certain geometric conditions are satisfied and two connectivity assumptions hold. In our experiments on face image and object image databases, our method achieves the state-of-the-art performance.

Acknowledgment

This work was supported by NSF BIGDATA grant 1447822. The authors also thank Manolis Tsakiris, Conner Lane and Chun-Guang Li for helpful comments.

References

[1] E. Arias-Castro, G. Chen, and G. Lerman. Spectral clustering based on local linear approximations. *Electron. J. Statist.*, 5:1537–1587, 2011. 1

[2] R. Basri and D. Jacobs. Lambertian reflection and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):218–233, 2003. 1

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30:107–117, 1998. 3

[4] E. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis. *Journal of the ACM*, 58, 2011. 1

[5] G. Chen and G. Lerman. Spectral curvature clustering (SCC). *International Journal of Computer Vision*, 81(3):317–330, 2009. 1

[6] Y. Cherapanamjeri, P. Jain, and P. Netrapalli. Thresholding based efficient outlier robust pca. *arXiv preprint arXiv:1702.05571*, 2017. 1

[7] T.-J. Chin, Y. Heng Kee, A. Eriksson, and F. Neumann. Guaranteed outlier removal with mixed integer linear programs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5858–5866, 2016. 3

[8] Y. Cong, J. Yuan, and J. Liu. Sparse reconstruction cost for abnormal event detection. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pages 3449–3456, 2011. 2

[9] C. Ding, D. Zhou, X. He, and H. Zha. R_1-pca: rotational invariant l_1-norm principal component analysis for robust subspace factorization. In *Proceedings of the 23rd international conference on Machine learning*, pages 281–288. ACM, 2006. 1

[10] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2790–2797, 2009. 2

[11] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013. 3

[12] M. A. Fischler and R. C. Bolles. RANSAC random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 26:381–395, 1981. 1

[13] R. G. Gallager. *Stochastic processes: theory for applications*. Cambridge University Press, 2013.

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

[15] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. 2007. 6

[16] R. Heckel and H. Bölcskei. Robust subspace clustering via thresholding. *IEEE Transactions on Information Theory*, 61(11):6320–6342, 2015. 1

[17] B. Jin, D. Lorenz, and S. Schiffler. Elastic-net regulariztion: error estimates and active set methods. *Inverse Problems*, 25(11), 2009. 6

[18] K. Kanatani. Motion segmentation by subspace separation and model selection. In *IEEE International Conference on Computer Vision*, volume 2, pages 586–591, 2001. 1

[19] G. Lerman and T. Maunu. Fast, robust and non-convex subspace recovery. *arXiv preprint arXiv:1406.6145*, 2014. 1

[20] G. Lerman, M. B. McCoy, J. A. Tropp, and T. Zhang. Robust computation of linear models by convex relaxation. *Foundations of Computational Mathematics*, 15(2):363–410, 2015. 1, 6

[21] G. Lerman and T. Zhang. Robust recovery of multiple subspaces by geometric ℓ_p minimization. *Annals of Statistics*, 39(5):2686–2715, 2011. 1

[22] D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov chains and mixing times*. American Mathematical Soc., 2009. 5

[23] X. Li and J. Haupt. Identifying outliers in large matrices via randomized adaptive compressive sampling. *IEEE Transactions on Signal Processing*, 63(7):1792–1807, 2015. 1

[24] Z. Lin, M. Chen, L. Wu, and Y. Ma. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv:1009.5055v2*, 2011. 6

[25] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *International Conference on Machine Learning*, pages 663–670, 2010. 2, 6

[26] G. Liu, H. Xu, and S. Yan. Exact subspace segmentation and outlier detection by low-rank representation. In *AISTATS*, pages 703–711, 2012. 3

[27] C. Lu, Z. Lin, and S. Yan. Correlation adaptive subspace segmentation by trace lasso. In *IEEE International Conference on Computer Vision*, pages 1345–1352, 2013. 5

[28] M. McCoy, J. A. Tropp, et al. Two proposals for robust pca using semidefinite programming. *Electronic Journal of Statistics*, 5:1123–1160, 2011. 1

[29] K. Mitra, A. Veeraraghavan, and R. Chellappa. Analysis of sparse regularization based robust regression approaches. *IEEE Transactions on Signal Processing*, 61(5):1249–1257, 2013. 3

[30] H. Moonesignhe and P.-N. Tan. Outlier detection using random walks. In *2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*, pages 532–539. IEEE, 2006. 3, 6

[31] H. Moonesinghe and P.-N. Tan. Outrank: a graph-based outlier detection framework using random walk. *International Journal on Artificial Intelligence Tools*, 17(01):19–36, 2008. 3, 6

[32] B. Nasihatkon and R. Hartley. Graph connectivity in sparse subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 5

[33] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (COIL-100). *Technical Report CUCS-006-96*, 1996. 6

[34] Q. Qu, J. Sun, and J. Wright. Finding a sparse vector in a subspace: Linear sparsity using alternating directions. In *Advances in Neural Information Processing Systems*, pages 3401–3409, 2014. 3

[35] M. Rahmani and G. Atia. Coherence pursuit: Fast, simple, and robust principal component analysis. *arXiv preprint arXiv:1609.04789*, 2016. 1, 6

[36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 6

[37] M. Soltanolkotabi and E. J. Candès. A geometric analysis of subspace clustering with outliers. *Annals of Statistics*, 40(4):2195–2238, 2012. 2, 3, 6

[38] H. C. Tijms. *A first course in stochastic models*. John Wiley and Sons, 2003.

[39] M. Tsakiris and R. Vidal. Dual principal component pursuit. In *ICCV Workshop on Robust Subspace Learning and Computer Vision*, pages 10–18, 2015. 3, 6

[40] R. Vidal, Y. Ma, and S. Sastry. *Generalized Principal Component Analysis*. Springer Verlag, 2016. 5

[41] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007. 2

[42] Y. Wang, C. Dicle, M. Sznaier, and O. Camps. Self scaled regularized robust regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3261–3269, 2015. 3

[43] Y. Wang, Y. Wang, and A. Singh. A deterministic analysis of noisy sparse subspace clustering for dimensionality-reduced data. In *International Conference on Machine Learning*, pages 1422–1431, 2015. 3

[44] Y. Wang, Y.-X. Wang, and A. Singh. Graph connectivity in noisy sparse subspace clustering. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 538–546, 2016. 5

[45] Y.-X. Wang and H. Xu. Noisy sparse subspace clustering. *Journal of Machine Learning Research*, 17(12):1–41, 2016. 3

[46] Y.-X. Wang, H. Xu, and C. Leng. Provable subspace clustering: When LRR meets SSC. In *Neural Information Processing Systems*, 2013. 5

[47] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *NIPS*, 2009. 1

[48] H. Xu, C. Caramanis, and S. Sanghavi. Robust pca via outlier pursuit. In *Advances in Neural Information Processing Systems*, pages 2496–2504, 2010. 1, 6

[49] C. You, C.-G. Li, D. Robinson, and R. Vidal. Oracle based active set algorithm for scalable elastic net subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3928–3937, 2016. 3, 5

[50] C. You, D. Robinson, and R. Vidal. Provable self-representation based outlier detection in a union of subspaces. *Arxiv*, 2017. 5, 6

[51] C. You and R. Vidal. Geometric conditions for subspace-sparse recovery. In *International Conference on Machine learning*, pages 1585–1593, 2015. 3

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

[53] T. Zhang, A. Szlam, and G. Lerman. Median k-flats for hybrid linear modeling with many outliers. In *Workshop on Subspace Methods*, pages 234–241, 2009. 1

[54] T. Zhang, A. Szlam, Y. Wang, and G. Lerman. Hybrid linear modeling via local best-fit flats. *International Journal of Computer Vision*, 100(3):217–240, 2012. 1

[55] Y. Zheng, S. Sugimoto, and M. Okutomi. Deterministically maximizing feasible subsystem for robust model fitting with unit norm constraint. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1825–1832. IEEE, 2011. 3

[56] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005. 3