An Overview of Robust Subspace Recovery

Gilad Lerman and Tyler Maunu

Abstract—This paper will serve as an introduction to the body of work on robust subspace recovery. Robust subspace recovery involves finding an underlying low-dimensional subspace in a dataset that is possibly corrupted with outliers. While this problem is easy to state, it has been difficult to develop optimal algorithms due to its underlying nonconvexity. This work emphasizes advantages and disadvantages of proposed approaches and unsolved problems in the area.

Index Terms—dimension reduction, subspace recovery, robustness, big data, unsupervised learning

I. INTRODUCTION: WHAT IS ROBUST SUBSPACE RECOVERY?

The purpose of this work is to survey and discuss the existing literature related to the problem of *robust subspace recovery* (RSR). By "robust", we mean that the methods we consider should not be too sensitive to corruptions in a dataset. These ideas trace their roots back quite far in the statistical literature [44, 71]. The basic motivation behind the development of robust procedures is that real data often does not subscribe to the clean assumptions required by many classical statistical procedures. Quoting Huber [44], "robustness signifies insensitivity to small deviations from the assumptions". The body of work considered in this survey tackles the question of robustness in a certain challenging and nonconvex statistical problem.

RSR involves finding a low-dimensional subspace structure in a corrupted, potentially high-dimensional dataset. Since the set of all subspaces of a fixed dimension is nonconvex, the RSR problem itself is inherently nonconvex. This has made the problem challenging to solve and has, in part, led to the variety of works outlined here.

At this point, it is essential that we clearly specify the problem, since there are many works in related but different areas. Indeed, the literature is confusing to navigate because this problem has also been coined robust principal component analysis (RPCA). As a classical statistical method, principal component analysis (PCA) attempts to model data by a subspace that captures the directions of maximum variance, but it is notoriously sensitive to corrupted data. Many researchers have proposed robust estimators, but the estimators mostly fall into two camps: *outlier-robust* methods and *sparse-corruption* methods. We hope to make this distinction clear, so as to avoid confusion between the two competing bodies of literature. The RSR problem is related to the former, while it has become common to use RPCA to refer to the latter.

For this discussion, assume we are given a dataset $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, with corresponding data matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$. In the literature, RPCA or *sparse-corruption* methods have focused on procomposition of a matrix $\boldsymbol{X}$ into low-rank and sparse components, $\boldsymbol{X} = \boldsymbol{L} + \boldsymbol{S}$, where $\boldsymbol{L}$ is low-rank and $\boldsymbol{S}$ is sparse (elementwise) [14, 124]. Here, the goal
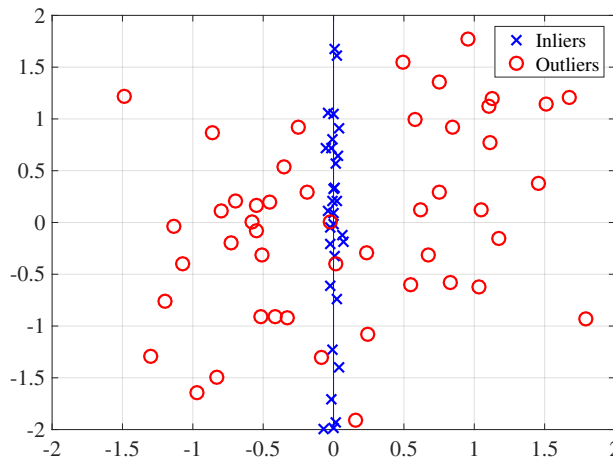

Fig. 1: Demonstration of an inlier-outlier dataset in $\mathbb{R}^2$ with an underlying one-dimensional subspace (the y-axis). The inliers are denoted by "x" and the outliers are denoted by "o".

is to recover the full low rank matrix $\boldsymbol{L}$ from the corrupted observations. A comprehensive review of this topic is given in [105].

On the other hand, the best way of thinking about RSR datasets is through partitioning $\mathcal{X}$ into inlier and outlier components, $\mathcal{X} = \mathcal{X}_{\text{in}} \cup \mathcal{X}_{\text{out}}$, where the inliers lie on or near a low-dimensional subspace, and the outliers are somehow distributed in the ambient space. We call such a dataset an inlier-outlier dataset. For clarity and so the reader may visualize the case we are talking about, we have displayed an artificial inlier-outlier dataset in Figure 1. The RSR problem asks to recover the underlying low-dimensional subspace. This problem is sometimes written as $\boldsymbol{X} = \boldsymbol{L} + \boldsymbol{C}$, where the columns of $\boldsymbol{L}$ span the underlying subspace and the non-zero columns of $\boldsymbol{C}$ correspond to outliers. Similar to the formulations of RPCA, some works have enforced column-sparsity of $\boldsymbol{C}$. However, calling $\boldsymbol{C}$ column-sparse in general is misleading, since many works on RSR consider very high percentages of outliers, in which case most of the columns of $\boldsymbol{C}$ are non-zero. This notation is also somewhat problematic, since the actual goal of RSR is to recover the underlying low-dimensional subspace, rather than the full low-rank matrix $\boldsymbol{L}$. Estimation of the subspace itself gives a more flexible output, while there is some freedom in choosing low-rank matrices $\boldsymbol{L}$ and corruption matrices $\boldsymbol{C}$ corresponding to a given subspace.

It is also important to note that the second case (column-sparse-corruption) is not just a special case of the first (elementwise-sparse-corruption). First of all, as mentioned above, many works on outlier-robust methods have considered cases with high percentages of outliers and, in some cases, have considered models where algorithms can tolerate arbi-

trary percentages of outliers. In this case, the corruption matrix can become quite dense. Second of all, the theoretical results for most sparse-corruption based methods have assumed that the corruptions are uniformly distributed across the elements of the data matrix. A matrix with column-sparse corruptions would have positions that are highly correlated and thus none of the current theoretical results for RPCA apply to RSR.

A. Roadmap

Here we briefly give an overview of the structure of this survey paper. We first give the basic formulations and algorithmic approaches for robust subspace recovery in §II. Then, in §III, we discuss and compare the various recovery guarantees for RSR algorithms, and we include a detailed discussion on well-defined data models. We display the computational complexity and memory requirements for the competing RSR algorithms in §IV. Empirical comparisons of the various RSR algorithms are discussed in §V, where we consider how one should measure the performance of an RSR algorithm, give comprehensive comparisons using various simulated datasets, discuss experiments that have been done on real data, and propose the creation of a substantial database for testing the applicability of RSR algorithms. The influence of RSR methods on other areas is discussed in §VI. Finally, in §VII, we finish with an outline of what remains to be done for RSR algorithms, and where we believe the field should go next.

B. Notation

In general, bold capital letters denote matrices and bold lower case letters denote vectors. For two sets A and B, $A \setminus B$ denotes the relative complement of B in A. The $(D-1)$-dimensional unit sphere in $\mathbb{R}^D$ is denoted by S^{D-1}. The Grassmannian $G(D, d)$ is the set of d-dimensional linear subspaces in $\mathbb{R}^D$, which we also refer to as d-subspaces. For a subspace $L \in G(D, d)$, its orthogonal complement is denoted by $L^\perp \in G(D, D - d)$. The matrix $\boldsymbol{I}_d$ denotes the $d \times d$ identity matrix, and, where it is not ambiguous, we just write $\boldsymbol{I}$. The set of semi-orthogonal matrices $O(D, d)$ is defined as $O(D, d) = \{\boldsymbol{U} \in \mathbb{R}^{D \times d} : \boldsymbol{U}^T \boldsymbol{U} = \boldsymbol{I}_d\}$. The norm $\|\cdot\|$ is used to refer to the Euclidean norm, and $\#(\cdot)$ denotes the number of elements in a set. The matrix $\boldsymbol{P}_L$ denotes the orthoprojector onto the subspace L, while $\boldsymbol{Q}_L$ is the orthoprojector onto $L^\perp$: $\boldsymbol{Q}_L = \boldsymbol{I} - \boldsymbol{P}_L$. Throughout the paper, we assume an inliers-outliers dataset $\mathcal{X} = \mathcal{X}_{\text{in}} \cup \mathcal{X}_{\text{out}}$ with N points and define $N_{\text{in}} = \#(\mathcal{X}_{\text{in}})$ and $N_{\text{out}} = \#(\mathcal{X}_{\text{out}}) = N - N_{\text{in}}$. As mentioned earlier, we denote the data points of $\mathcal{X}$ by $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^D$ and their corresponding data matrix by $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N] \in R^{D \times N}$. The data matrices for $\mathcal{X}_{\text{in}}$ and $\mathcal{X}_{\text{out}}$ are $\boldsymbol{X}_{\text{in}} \in \mathbb{R}^{D \times N_{\text{in}}}$ and $\boldsymbol{X}_{\text{out}} \in \mathbb{R}^{D \times N_{\text{out}}}$, respectively. We use "w.h.p." to denote "with high probability", which refers to probabilities that have orders $1 - O(N^{-a})$, for some absolute constant $a > 0$. Similarly, we use "w.o.p." to denote "with overwhelming probability", which refers to probabilities that scale at least like $1 - O(e^{-bN^a})$, for an absolute constant $a > 0$, and a constant $b > 0$ that is independent of N, but may depend on D, d, and the fraction of outliers. In many of the nonconvex optimization problems considered here, the

minimizer or maximizer may not be unique in general. Thus, we write "$\in$ argmax" or "$\in$ argmin" to denote that the estimator is contained in the set of maximizers of minimizers, respectively.

II. BASIC FORMULATIONS FOR ROBUST SUBSPACE RECOVERY

In this section, we hope to motivate a few basic strategies for subspace recovery in order to give a better understanding of the problem. For the rest of this survey, we assume a linear subspace setting. That is, the subspace on or around which the inliers lie is linear. Here, we have an inlier-outlier data matrix, $\boldsymbol{X} \in \mathbb{R}^{D \times N}$, and we wish to recover a linear subspace $L \in G(D, d)$. We may interchangeably search for a matrix $\boldsymbol{U} \in O(D, d)$ whose columns span $L \in G(D, d)$. The case of affine subspaces is discussed in §VII. After briefly reviewing PCA in §II-A and discussing the difficulties of developing an outlier-robust version of PCA in §II-B, we discuss the various approaches of RSR algorithms in the following categories

- §II-C Projection Pursuit
- §II-D Least Absolute Deviations
- §II-E L_1-PCA
- §II-F Robust Covariances
- §II-G Other Energy Minimizers
- §II-H Filtering Outliers
- §II-I Exhaustive Subspace Search

At last, in §II-J we discuss some related parallel works to RSR.

A. Review of Subspace Modeling by PCA

Classically, subspace modeling has been formulated using principal component analysis (PCA), which finds the orthogonal directions of maximum variance. Using the notation in §I-B, the PCA d-subspace of the dataset $\{\boldsymbol{x}_i\}_{i=1}^N \subset \mathbb{R}^D$ is defined as

$$L_{PCA} \in \text{argmax}_{L \in G(D,d)} \sum_{i=1}^N \|\boldsymbol{P}_L \boldsymbol{x}_i\|^2. \tag{1}$$

This subspace has a direct and simple numerical solution. Indeed, it is the span of the top d eigenvectors of the scaled sample covariance, $\boldsymbol{X} \boldsymbol{X}^T$, or equivalently, the top d left singular vectors of $\boldsymbol{X}$. This solution is unique when the dth and $(d+1)$st eigenvalues of $\boldsymbol{X} \boldsymbol{X}^T$ are not equal. Otherwise, all d-subspaces of a larger subspace of $\mathbb{R}^D$ are the global minimizers, and there are no other local minimizers. The PCA minimization is very nice compared to many other nonconvex optimization formulations due to this direct solution.

The equivalent formulation for this problem over $O(D, d)$ is

$$\boldsymbol{U}_{PCA} \in \text{argmax}_{\boldsymbol{U} \in O(D,d)} \sum_{i=1}^N \|\boldsymbol{U} \boldsymbol{U}^T \boldsymbol{x}_i\|^2. \tag{2}$$

Another equivalent formulation of (1) immediately follows from the identity $\|\boldsymbol{x}_i\|^2 = \|\boldsymbol{P}_L \boldsymbol{x}_i\|^2 + \|\boldsymbol{Q}_L \boldsymbol{x}_i\|^2$:

$$L_{PCA} \in \text{argmin}_{L \in G(D,d)} \sum_{i=1}^N \|\boldsymbol{Q}_L \boldsymbol{x}_i\|^2. \tag{3}$$

This formulation can be interpreted as minimizing the variance orthogonal to a subspace. In simple geometric terms, it minimizes the sum of squared orthogonal distances between the data points and the subspace L. Indeed, the function $\|\boldsymbol{Q}_L \boldsymbol{x}_i\|$ in (3) is just the orthogonal distance between the point, $\boldsymbol{x}_i$, and the subspace L. Notice that the choice of the squared Euclidean norm can be motivated by maximum likelihood estimation of the PCA subspace under a Gaussian generative model, analogous to the least squares estimator in ordinary least squares regression.

B. Difficulties of Developing Outlier-Robust PCA

Beyond PCA, which has a direct solution, the problem of robustly estimating a subspace becomes hard. Indeed, issues range from the proper definition of a robust estimator to the actual calculation of these estimators.

As an example, consider the following program to robustly find an underlying subspace. In a noiseless inlier-outlier dataset, one may replace the least squares formulation of PCA in (3) with the following ℓ_0-type formulation:

$$\hat{L} \in \operatorname{argmin}_{L \in G(D,d)} \#(\mathcal{X} \setminus L). \tag{4}$$

In the case of noisy inliers, one may try to find

$$\hat{L} \in \operatorname{argmin}_{L \in G(D,d)} \#\{\boldsymbol{x} \in \mathcal{X} : \|\boldsymbol{Q}_L \boldsymbol{x}\| > \epsilon\}, \tag{5}$$

where $\epsilon > 0$ is somehow tied to the magnitude of the noise. There is no easy way of even approximating the solution to (4) or (5) in general. Further, when real data is noisy, there is no obvious way to choose the parameter ϵ in (5). As we will discuss later, relaxing (4) to an ℓ_1 formulation still results in an NP-hard problem. This stands in contrast to the ℓ_0 to ℓ_1 relaxation in settings like regression or compressed sensing, where one gets a convex program that can be solved using a variety of methods. Also, the solutions of (4) and (5) may not be unique, whereas our initial formulation of the RSR problem assumes a unique underlying subspace. This issue, which is evident in non-convex programs for RSR, will be later addressed in §III-A.

It is also unclear that the formulations in (4) and (5) are the most natural ones. Indeed, in real situations, data is quite messy and never lies exactly on a subspace, and so one must consider (5) in general. However, there are various scenarios where (5) may not give a useful estimate. For example, (5) may not perform well when the noise is not uniform around the subspace or when the outliers lie around a union of nearby subspaces and ϵ is overestimated, as we demonstrate later in Figure 2e.

C. Projection Pursuit

A body of works on robust subspace recovery includes projection pursuit based methods [1, 18, 34, 44, 51, 57, 70, 74], which can be motivated in the following way. One can attempt to find a direction (component) maximizing a robust scale function $\rho : \mathbb{R}^N \to [0, \infty)$ with respect to the data as follows:

$$\boldsymbol{v}_1 \in \operatorname{argmax}_{\boldsymbol{v} \in S^{D-1}} \rho(\boldsymbol{v}^T \boldsymbol{X}). \tag{6}$$

One typically finds all d components in a sequential manner, which we explain after discussing the notion of a robust scale function and attempts to solve (6).

When using the non-robust scale function $\rho(\boldsymbol{y}) = \|\boldsymbol{y}\|_2^2$, $\boldsymbol{v}_1$ is the top principal component, which is also expressed by (2) when $d = 1$. A robust version of the top principal component can be developed by choosing a proper scale function, such as a trimmed variance, $\rho(\boldsymbol{y}) = \|\boldsymbol{y}\|_1$, or a Huber-type scale function. When $d = 1$, using $\rho(\boldsymbol{y}) = \|\boldsymbol{y}\|_1$ results in the maximization variants of both least absolute deviations and L_1-PCA, which will be presented later in (12) and (26) respectively. One can attempt to optimize the nonconvex objective (6) in many ways. In general, exhaustively searching for this maximizer results in a non-polynomial time algorithm. Instead, most algorithms resort to finding a local maximum of (6) or some sort of approximate global maximum. Past works have used iterative reweighting schemes [70], bit-flipping [51], and convex relaxation [74].

One can estimate a set of components in a sequential manner in the following way. After finding $\boldsymbol{v}_1$ by (6), each sequential component $\boldsymbol{v}_j$, $1 < j \leq d$, is found by solving the same problem with the added constraint of orthogonality with the previously found vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{j-1}$. That is, $\boldsymbol{v}_j$ is found as

$$\boldsymbol{v}_j \in \operatorname*{argmax}_{\boldsymbol{v} \in S^{D-1}, \boldsymbol{v} \perp \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{j-1}} \rho(\boldsymbol{v}^T \boldsymbol{X}). \tag{7}$$

Note that this is equivalent to solving (6) after the columns of $\boldsymbol{X}$ are projected onto the orthogonal complement of $\operatorname{Sp}(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{i-1})$. One can also try to find a maximizer of the joint energy $\sum_j \rho(\boldsymbol{v}_j^T \boldsymbol{X})$ such that the set of components, $\boldsymbol{v}_j \in S^{D-1}$, $j = 1, \ldots, d$, are pairwise orthogonal [74]. McCoy and Tropp [74] develop the Maximum Mean Absolute Deviation Rounding (MDR) algorithm, which finds an approximate global maximizer for the joint problem

$$\operatorname*{argmax}_{\substack{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d \in S^{D-1} \\ \boldsymbol{v}_j \perp \boldsymbol{v}_k, j \neq k}} \sum_{j=1}^{d} \|\boldsymbol{v}_j^T \boldsymbol{X}\|_1. \tag{8}$$

We note that (8) is also known as the maximization variant or L_1-PCA, which we discuss further in §II-E.

D. Least Absolute Deviations

A popular approach to RSR is to replace the least squares formulation in (3) with least absolute deviations:

$$\hat{L} \in \operatorname{argmin}_{L \in G(D,d)} \sum_{i=1}^{N} \|\boldsymbol{Q}_L \boldsymbol{x}_i\|. \tag{9}$$

This problem has been considered for many reasons, such as its nice interpretation as a geometric median subspace. Indeed, the minimizer of (9) can heuristically be motivated by the geometric median, which solves the least absolute deviations analog for estimating the center of a dataset [65]. Despite being an appealing formulation, (9) is NP-hard to even approximately minimize to an error of order $\Omega(1/\operatorname{poly}(D))$ [19].

One of the attractive features of using the least absolute deviations formulation is that it is rotationally invariant with respect to choice of basis [24]. We clarify this notion

of invariance as follows. A subspace in $G(D, d)$ can be represented by an orthonormal system of vectors spanning this subspace. The latter vectors can be identified with the columns of an element of $O(D, d)$. Right multiplication of this element in $O(D, d)$ by an element of $O(d, d)$ results in another semi-orthogonal matrix whose columns still span the same subspace. Therefore, $G(D, d)$ is identified with equivalence classes of $O(D, d)$, and the equivalence relation is obtained by the right action of $O(d, d)$. The cost in (9) is the same for any choice of coordinates within an equivalence class. Indeed, if $L = \mathrm{Sp}(\boldsymbol{U}_1) = \mathrm{Sp}(\boldsymbol{U}_2)$ for two different matrices $\boldsymbol{U}_1, \boldsymbol{U}_2 \in O(D, d)$, where $\boldsymbol{U}_1 = \boldsymbol{U}_2 \boldsymbol{R}$ for some $\boldsymbol{R} \in O(d, d)$, then

$$\sum_{i=1}^{N} \|\boldsymbol{Q}_L \boldsymbol{x}_i\| = \sum_{i=1}^{N} \|(\boldsymbol{I} - \boldsymbol{U}_1 \boldsymbol{U}_1^T) \boldsymbol{x}_i\| \qquad (10)$$
$$= \sum_{i=1}^{N} \|(\boldsymbol{I} - \boldsymbol{U}_2 \boldsymbol{U}_2^T) \boldsymbol{x}_i\|.$$

This rotational invariance is an essential feature of estimation over the Grassmannian, and not all problem formulations have this (see, e.g., the later formulation for L_1-PCA in (26), which is not rotationally invariant).

Some motivation for this formulation of RSR can also come from relaxing an ℓ_0 problem to an ℓ_1 problem, mirroring ideas in compressed sensing. This involes rewriting the function $\#(\cdot)$ in (4) as the ℓ_0-norm of the vector of distances between the data points and a subspace. One can then relax the ℓ_0-norm to an ℓ_1-norm and arrive at (9).

A recent work that was motivated by the sparse formulation in (4) was originally discussed by [96] and further analyzed by Tsakiris and Vidal [103]. However, their formulation is really just least absolute deviations in disguise. Indeed, they iteratively try to find a hyperplane that approximately contains as many points as possible by solving for its normal vector, $\boldsymbol{b}$, as follows:

$$\min_{\boldsymbol{b} \in S^{D-1}} \|\boldsymbol{X}^T \boldsymbol{b}\|_1. \qquad (11)$$

We note that this is equivalent to (9) with $d = D - 1$ because $|\boldsymbol{x}_i^T \boldsymbol{b}| = \|\boldsymbol{Q}_{\mathrm{Sp}(\boldsymbol{b})^\perp} \boldsymbol{x}_i\|_2$, where $\mathrm{Sp}(\boldsymbol{b})^\perp \in G(D, D-1)$.

We observe that a "least absolute deviation" formulation of (1) is

$$\hat{L} \in \mathrm{argmax}_{L \in G(D, d)} \sum_{i=1}^{N} \|\boldsymbol{P}_L \boldsymbol{x}_i\|. \qquad (12)$$

Even though the solutions of (9) and (12) may not necessarily be the same, many of the methods developed for (9) can be adapted to (12). When $d = 1$, the projection pursuit procedure in (6) with $\rho(\boldsymbol{y}) = \|\boldsymbol{y}\|_1$ coincides with (12). An approximate polynomial-time solution of (12) for any fixed d, within a large absolute factor, was suggested in [76].

We claim that the least absolute deviations formulation is very amenable to the use of iteratively reweighted least squares (IRLS). It is easiest to explain this claim with the following straightforward argument of Lerman and Maunu

[53] for approximating (9) (this can also be adapted to (12)). They suggest the iterative procedure

$$L^{k+1} \in \mathrm{argmin}_{L \in G(D, d)} \sum_{i=1}^{N} w_i^k \|\boldsymbol{Q}_L \boldsymbol{x}_i\|^2, \qquad (13)$$

where $w_i^k = 1/\|\boldsymbol{Q}_{L^k} \boldsymbol{x}_i\|$. The formulation (13) is a weighted PCA problem, which has a direct solution via the SVD of the matrix whose columns are $\{\sqrt{w_i^k} \boldsymbol{x}_i\}_{i=1}^{N}$. Other IRLS approaches for the least absolute deviations problem appear in [56, 119].

Many methods have been developed to approximate the solution of (9). We distinguish below between convex relaxations and direct nonconvex strategies.

1) Convex Relaxations: The first relaxation of the least absolute deviations problem was concurrently considered by [74] and [111]. These works propose the following optimization problem for noiseless RSR

$$\min_{\boldsymbol{L}, \boldsymbol{C} \in \mathbb{R}^{D \times N}} \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{C}\|_{1,2}, \ \text{s.t.} \ \boldsymbol{L} + \boldsymbol{C} = \boldsymbol{X}. \qquad (14)$$

Here, $\|\cdot\|_*$ denotes the nuclear norm of a matrix and $\|\cdot\|_{1,2}$ denotes the sum of the column norms of a matrix. The reason why (14) relaxes (9) is discussed in the next paragraph. For the noisy case, they consider the problem

$$\min_{\boldsymbol{L}, \boldsymbol{C} \in \mathbb{R}^{D \times N}} \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{C}\|_{1,2}, \ \text{s.t.} \ \|\boldsymbol{X} - (\boldsymbol{L} + \boldsymbol{C})\|_F \leq \epsilon, \quad (15)$$

where ϵ is an estimated small noise level. In both algorithms, the parameter λ is chosen to be $3/(7\sqrt{N_{\text{out}}})$ [111]. Since N_{out} is not known, one must guess an upper bound on the number of outliers. In practice, with sufficiently small percentages of outliers, the authors argue that one can overestimate N_{out} and still have good performance, because the algorithm will first remove all outliers and then remove some inliers. The resulting set of inliers can then still recover the correct column space of $\boldsymbol{L}$. We have found that choosing $\lambda = 3/(7\sqrt{N_{\text{out}}})$ does not perform well in the settings we test. We instead choose $\lambda = 0.8\sqrt{D/N}$, which seems to work better (this choice was also used in [119]).

To show that this is a convex relaxation of (9), one can replace $\|\boldsymbol{L}\|_*$ with $\mathrm{rank}(\boldsymbol{L})$. Then, a simple geometric argument shows that $\boldsymbol{C} = \boldsymbol{Q}_{\mathrm{Sp}(\boldsymbol{L})} \boldsymbol{X}$, and so $\|\boldsymbol{C}\|_{1,2}$ then measures the deviation of the columns of $\boldsymbol{X}$ from the column span of $\boldsymbol{L}$. In other words, $\|\boldsymbol{C}\|_{1,2}$ is the sum of orthogonal distances between points and the span of $\boldsymbol{L}$.

Since (14) and (15) form a convex programs, it is possible to optimize them using a range of algorithms. Xu et al. [111] advocate using a proximal gradient algorithm. They refer to the problems in (14) and (15) as "outlier pursuit", for which we use the acronym OP.

Two other algorithms, the Geometric Median Subspace (GMS) [119] and REAPER [56], are also convex relaxations of the robust energy in (9). GMS seeks a relaxed orthoprojection onto the orthogonal complement of an underlying subspace

through robustly estimating the inverse covariance matrix. The GMS estimator is constructed through the convex relaxation

$$\hat{Q} = \operatorname{argmin}_{Q \in \mathbb{H}} \sum_{i=1}^{N} \|Q x_i\|, \quad (16)$$

$$\mathbb{H} = \{Q \in \mathbb{R}^{D \times D} : Q = Q^T, \operatorname{Tr}(Q) = 1\}.$$

The underlying subspace is then estimated from the bottom eigenvectors of $\hat{Q}$. On the other hand, REAPER solves a tighter convex relaxation designed to robustly estimate the orthoprojector onto the underlying subspace L_*. The convex program is

$$\hat{P} = \operatorname{argmin}_{P \in \mathbb{G}} \sum_{i=1}^{N} \|(I - P) x_i\|, \quad (17)$$

$$\mathbb{G} = \{P \in \mathbb{R}^{D \times D} : 0 \preceq P \preceq I, \operatorname{Tr}(P) = d\}.$$

Here, the estimated subspace is calculated as the top eigenvectors of $\hat{P}$. Note that $\mathbb{G}$ is the convex hull of the set of orthoprojectors of rank d [56]. Thus, by identifying subspaces with their orthoprojectors, we see that (17) is the tightest convex relaxation of (9). We remark that the minimizer of (17) does not change if the constraint $\preceq I$ is removed (see proof of Lemma 14 in [119]). Therefore, one may note that (16) is obtained from (17) by setting $P = I - Q$ and dropping the constraint $P \succeq 0$. Indeed, after doing this, any fixed value of $\operatorname{Tr}(Q)$ yields the same subspace. Both of these algorithms employ IRLS procedures to efficiently solve their respective optimization problems.

2) Nonconvex Optimization: An alternative to convex relaxation of (9) is to attempt to directly minimize this energy function. The advantage of doing this is that one can obtain faster algorithms for special settings. However, these algorithms are typically hard to theoretically justify, despite their impressive practical performance. Only recently have theoretical results shown the strength of these methods in certain regimes.

Ding et al. [24] considered direct optimization of the nonconvex program in (9), which they incorrectly assumed was convex. To do this, they use a form of the power method (see the method of orthogonal iteration in §8.2.1 of [40]). This algorithm is referred to as Rotational Invariant L_1-norm PCA (R1PCA). This method is somewhat problematic since the optimization technique they use is tied to convex methods and may lead to poor solutions in the nonconvex case.

Direct optimization of (9) on $G(D, d)$ was later considered in the sequence of works by Lerman and Maunu [53] and Maunu et al. [73]. In [53], the authors directly use IRLS on (9). The resulting method is called the Fast Median Subspace algorithm (FMS). In the next work [73], they use a geodesic gradient descent method to minimize (9) over $G(D, d)$ by drawing on ideas from [28]. In practice, FMS seems to perform better than GGD, but existing theoretical guarantees for GGD are stronger.

Another work that attempts to approximately minimize the least absolute deviations energy is given in [19]. Their algorithm, called ConstApprox, also accounts for sparse inputs, which yields reduced computational complexity for sparse matrices. The approximation method can return a $(1 + \epsilon)$

approximation to the minimum value of the program in (9) for sufficiently large ϵ. On the other hand, they show that the approximation problem becomes NP-hard when $\epsilon = \Omega(1/\operatorname{poly}(D))$.

Another nonconvex optimization method closely tied to (9) and the outlier pursuit relaxation came in Cherapanamjeri et al. [17], where the authors coin their algorithm Thresholding based Outlier Robust PCA (TORP). The authors use a nonconvex thresholding based algorithm, which iterates between fitting a PCA subspace and filtering points that are either far from the subspace or highly incoherent. The definition of incoherence is later given in §III-C1. A disadvantage of this method is that it requires the user to input the percentage of outliers, which is not known in practice. As in OP, one can overestimate the percentage of outliers and still have accurate recovery when the percentage of outliers is sufficiently small.

Tsakiris and Vidal [103] proposed the Dual Principal Component Pursuit (DPCP) algorithm that sequentially fits nested hyperplanes by finding stationary points in the program (11). For solving (11), they follow an algorithm of [96], which uses an alternating sequence of convex relaxation followed by a nonconvex projection. More precisely, the sequence $(\tilde{b}^k)_{k \geq 1}$ is defined by the following program:

$$b^{k+1} \in \operatorname{argmin}_{n^T \tilde{b}^k = 1} \|X n\|_1, \quad \tilde{b}^{k+1} = \frac{b^{k+1}}{\|b^{k+1}\|}. \quad (18)$$

Notice that the minimization in (18) just involves solving a linear program at each iteration. After one hyperplane is found (i.e., the $(D - 1)$-subspace perpendicular to the limit of this sequence), the DPCP procedure searches for a hyperplane of this hyperplane, which results in a $(D - 2)$-subspace. This procedure is repeated until one is left with a d-subspace.

E. L_1-PCA

There are two different variants of L_1-PCA that we discuss here: the minimization and maximization based formulations [10, 68]. It seems that the minimization based variant is more closely tied to the RPCA problem reviewed in [105], while the maximization variant seems to be tied to joint projection pursuit and thus is more closely related to RSR.

The minimization formulation of L_1-PCA forms the following analog of (3):

$$U_{L_1-\min} \in \operatorname*{argmin}_{\substack{U \in O(D,d) \\ \{y_i\}_{i=1}^N \subset \mathbb{R}^d}} \sum_{i=1}^{N} \|x_i - U y_i\|_1. \quad (19)$$

One can also write (19) as

$$U_{L_1-\min} \in \operatorname*{argmin}_{\substack{U \in O(D,d) \\ Y \in \mathbb{R}^{d \times N}}} \|X - UY\|_{1,1}, \quad (20)$$

where the $\|\cdot\|_{1,1}$-norm sums the absolute values of the matrix elements. This formulation is equivalent to PCA when one uses the squared L_2-norm instead of the L_1-norm. Indeed, one can write the PCA minimization as

$$\min_{U \in O(D,d), Y \in \mathbb{R}^{d \times N}} \|X - UY\|_{2,2}, \quad (21)$$

where $\|\cdot\|_{2,2}$ corresponds to the Frobenius norm. Notice that the minimization in (19) is rotationally invariant in the sense of §II-D since we can write

$$U_{L_1-\min} \in \operatorname*{argmin}_{\substack{U\in O(D,d)\\ \{y_i\}_{i=1}^N\subset\mathbb{R}^d}} \sum_{i=1}^N \|x_i - URR^Ty_i\|_1, \quad (22)$$

and cast the optimization over the variables $U' = UR$ and $z_i = R^Ty_i$.

A perhaps simpler equivalent formulation of the minimization in (19) is given by

$$\hat{L} \in \operatorname*{argmin}_{\operatorname{rank}(L)\le d}\|X - L\|_{1,1}. \quad (23)$$

The subspace estimate can then be found from the span of L.

We remark that (23) can be viewed a non-convex relaxation of the following problem

$$\hat{L} \in \operatorname*{argmin}_{\operatorname{rank}(L)\le d}\|X - L\|_{0,0}, \quad (24)$$

where the $\|\cdot\|_{0,0}$ is just the number of non-zero entries of a matrix. The later problem is in fact the RPCA problem, where one seeks low-rank approximation to a matrix with sparse corruptions. Attempts to find approximate solutions for this problem are discussed in [105].

The nonconvex and nonsmooth minimization problem in (19) was originally considered in Baccini et al. [7], where the authors show that this choice of norm is equivalent to finding the maximum likelihood estimate (MLE) subspace under a Laplacian noise assumption (rather than Gaussian for PCA). Further convex relaxation algorithms were developed by [48] and later by a more recent surge of work (see Yu et al. [117] and Brooks et al. [10] for some examples). Brooks et al. [10] give a nonconvex, polynomial time algorithm for the special case of $d = D-1$. Gillis and Vavasis [35] showed that this minimization problem is NP-hard for $d < D - 1$. Song et al. [95] study approximate minimization of this quantity, where they derive a polynomial time algorithm to approximate the minimizer up to a given threshold.

We emphasize that while the minimization variant of L_1-PCA is a natural robust extension of PCA, it may not be ideal for solving the RSR problem discussed in this paper. Indeed, the formulation in (23) and its MLE interpretation seem to be more robust to elementwise corruption than to outliers.

Unlike least absolute deviations, the minimization variant of L_1-PCA does not have a simple IRLS formulation to take advantage of. Indeed, the elementwise weighting procedure presents some issues. For example, similar to the idea summarized in (13), one could try to apply the following IRLS procedure to approximate (23):

$$L^{k+1} \in \operatorname*{argmin}_{\operatorname{rank}(L)\le d} \sum_{i,j} w_{ij}^k(X_{ij} - L_{ij})^2, \quad (25)$$

where $w_{ij}^k = 1/|X_{ij} - L_{ij}^k|$. However, this least squares problem has no straightforward solution at each iteration [97]. One could use a strategy like the alternating least squares algorithm presented by De La Torre and Black [22] for solving (25) with different robust weights w_{ij}^k. However, there would be no guarantee of globally minimizing the least squares problem at each iteration.

The maximization formulation of L_1-PCA is given by

$$U_{L_1-\max} \in \operatorname*{argmax}_{U\in O(D,d)} \sum_{i=1}^N \|U^Tx_i\|_1. \quad (26)$$

Note that (19) and (26) are the L_1-PCA versions of (3) and (1), respectively. However, while (3) and (1) are equivalent, (19) and (26) are not. The L_1-PCA version in (26) is actually a special case of joint energy projection pursuit. If one considers the joint projection pursuit energy from §II-C, $\sum_{j=1}^d \rho(v_j^Tx_i)$, with $\rho(x) = \|x\|_1$ over orthonormal sets $\{v_1,\ldots,v_d\}$, one arrives at precisely the formulation in (26). Therefore, like projection pursuit, the formulation in (26) addresses the RSR problem. It thus has different characteristics than the formulation in (19), which is tied to the RPCA problem. We remark that there is no straightforward maximum likelihood interpretation of (26), unlike (19).

Notice that the formulation in (26) is not rotation invariant with respect to choice of basis, unlike the formulation in (19). Indeed, if $R \in O(d,d)$, then unlike the Euclidean norm, $\|U^Tx_i\|_1 \ne \|R^TU^Tx_i\|_1$ in general. Thus, this formulation is not truly over $G(D,d)$. If instead we wish to formulate (26) over $G(D,d)$, we should try to solve

$$U'_{L_1-\max} \in \operatorname*{argmax}_{U\in O(D,d)} \sum_{i=1}^N \|UU^Tx_i\|_1. \quad (27)$$

Indeed, since $\|UU^Tx_i\|_1 = \|URR^TU^Tx_i\|_1$, we have rotation invariance with respect to choice of basis. We are not aware of work focusing on the maximization in (27).

For both of the maximization problems in (26) and (27), one could come up with IRLS formulations as was done for (19) in (25). However, the same issues arise as before since it is not an easy task to solve the least squares portion of the algorithm.

For large N and D, the maximization problem in (26) is NP-hard [74]. Nevertheless, Kwak [51] first developed an algorithm that sequentially outputs local maxima of the one-dimensional version of (26). Later, exact algorithms were developed by Markopoulos et al. [67] for sufficiently small N and D. An approximate polynomial-time solution of (26), within a large absolute factor, was suggested in [76]. Their work improves over an earlier $O(\log(N))$ approximation factor in [74]. A review of algorithms and methods for the L_1-maximization problem in (26) appears in [68].

F. Robust Covariances

Another line of thought has considered robustly estimating the underlying covariance matrix of a dataset [26, 27, 63, 66, 69, 71, 78, 98, 104, 106, 118, 119], which can then be used to locate underlying subspaces. The simplest setting assumes that the population mean, μ, is 0. After calculating the robust covariance estimator, one can find the robust principal subspace from its top eigenvectors. The direct synthesis of these ideas with the problem of subspace recovery can be seen in [118, 119].

One example is the Maronna M-estimator [69]. It minimizes a certain robust energy that is a maximum likelihood covari-

ance estimator under an elliptical distribution with heavy tails. More precisely, it is the minimizer of

$$\frac{1}{N}\sum_{i=1}^{N}\rho(\boldsymbol{x}_i^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x}_i) + \frac{1}{2}\log\det(\boldsymbol{\Sigma}) \qquad (28)$$

over all positive definite $\boldsymbol{\Sigma}$, where ρ is a function that satisfies certain conditions. Similarly, the Tyler M-estimator (TME) [104] minimizes the energy

$$\frac{1}{N}\sum_{i=1}^{N}\log(\boldsymbol{x}_i^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x}_i) + \frac{1}{D}\log\det(\boldsymbol{\Sigma}), \qquad (29)$$

among all $\boldsymbol{\Sigma}$ positive definite with trace 1. A more in depth discussion of these energy functions and their robustness is given in Appendix A.

The advantage of (28) and (29) is that their formulations are geodesically convex [6, 110, 118, 123]. Both estimators can be iteratively computed by an IRLS procedure. When $D > N$, these estimators are undefined [69, 104], and even when $D \le N$ they may be ill-conditioned. It is thus common to regularize them [82, 100].

Perhaps the simplest robust covariance estimator is the spherical sample covariance [53, 63, 71], which can be estimated as

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_{i=1}^{N}\frac{\boldsymbol{x}_i\boldsymbol{x}_i^T}{\|\boldsymbol{x}_i\|^2}. \qquad (30)$$

Spherical PCA (SPCA) computes the principal subspace of this estimator, which is the PCA subspace of the normalized dataset $\{\boldsymbol{x}_i/\|\boldsymbol{x}_i\|\}_{i=1}^{N}$ [63].

In a more general setting, both the mean, $\boldsymbol{\mu}$, and the covariance, $\boldsymbol{\Sigma}$, are unknown. If one only cares about estimating the covariance, then one can calculate the estimators above on the set of differences between data points, $\boldsymbol{x}_i - \boldsymbol{x}_j$ for $i \ne j$, $i, j = 1, \ldots, N$. For example, the spatial Kendall's tau matrix [106] estimates the spherical covariance by

$$\hat{\boldsymbol{\Sigma}} = \frac{2}{N(N-1)}\sum_{i\ne j}\frac{(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}. \qquad (31)$$

Similarly, Dümbgen's M-estimator [27] computes TME on the set of differences between points, and Nordhausen and Tyler [78] apply this procedure, which they refer to as symmetrization, to other robust covariance estimators. These estimators can address RSR in the affine setting. Indeed, an affine subspace can be decomposed into a linear subspace plus an offset. The estimated linear subspace is the principal subspace of the "symmetrized" robust covariance estimator, which is expected to approximate the underlying linear component. On the other hand, the offset could be well-approximated by a robust point estimator, such as the geometric median. A benefit of symmetrization is that it avoids estimating the offset first and centering the data at this offset. With the latter procedure, small approximation error of the offset may result in large approximation error of the linear subspace component.

G. Other Energy Minimizers

The methods reviewed so far were formulated by energy minimization or by maximization of a utility function. Another example is given by Xu and Yuille [113], who tried to minimize a trimmed version of the PCA energy given by

$$\min_{L \in G(D,d)} \sum_{i=1}^{N} \begin{cases} \|\boldsymbol{x}_i - \boldsymbol{P}_L\boldsymbol{x}_i\|^2, & \|\boldsymbol{x}_i - \boldsymbol{P}_L\boldsymbol{x}_i\|^2 < \eta, \\ \eta, & \|\boldsymbol{x}_i - \boldsymbol{P}_L\boldsymbol{x}_i\|^2 \ge \eta. \end{cases} \quad (32)$$

The motivating idea is that trimming the energy would give robustness to outliers, while maintaining some desirable characteristics of PCA. An additional example of a method that aims to maximize a utility function appears below in (34).

H. Filtering Outliers

One way of attempting RSR is to first filter outliers and then fit a subspace to the data by using PCA. A simple filtering idea is to use affinities that express presence in an underlying subspace (or multiple underlying subspaces) to screen and remove outliers. The first recipe was suggested by Chen and Lerman [16] (see, in particular, §3.1). They form a symmetric weight matrix that aims to express the likelihood that pairs of points lie on an "underlying d-dimensional subspace", that is, a subspace that many other data points lie on. The degrees of the data points are then computed from this weight matrix, where a degree of a data point is the sum of weights in the corresponding row of the matrix. The outliers are identified as points with low degree, or in other words, points with low likelihood of being contained in a d-subspace. This idea can be used in the setting of robust subspace recovery and also in the setting of robust subspace clustering. In the latter setting, inliers lie on a union of subspaces and the goal is to recover these subspaces in the presence of outliers. A similar idea is suggested in [4] for the more general setting of robust manifold clustering, where inliers lie on a union of manifolds and the goal is to recover these manifolds in the presence of outliers.

Soltanolkotabi and Candès [94], whose ideas build on those in [30], also identify outliers according to low degrees of a weight matrix. Their expression for the degree of a data point $\boldsymbol{x}_j \in \mathbb{R}^D$ is the value of the following program:

$$\min_{\boldsymbol{r}_j \in \mathbb{R}^N} \|\boldsymbol{r}_j\|_1, \text{ s.t. } \boldsymbol{X}_{-j}\boldsymbol{r}_j = \boldsymbol{x}_j, \qquad (33)$$

where $\boldsymbol{X}_{-j}$ is the $D \times N$ data matrix with the jth column zeroed out. To relate this idea to the framework of [16], one can form the asymmetric $N \times N$ weight matrix $\boldsymbol{R}$, whose jth row is the vector $\boldsymbol{r}_j$ minimizing (33). Clearly, the jth degree of $\boldsymbol{R}$ (i.e., the sum of the weights in row j) is the minimal value in (33). You et al. [116] use a similar matrix $\boldsymbol{R}$, which is formed by elastic net minimization instead of pure ℓ_1-norm minimization, to create a random walk over the nodes of the graph. They iterate $\boldsymbol{R}$ in an interesting way to obtain a limiting vector that aims to be supported on the inliers of the robust subspace clustering problem. They refer to this method as Self-Representation Outlier Detection (SRO).

The recent Coherence Pursuit (CP) method by Rahmani and Atia [90] follows the initial framework in [16] of identifying outliers according to low degrees in a certain $N \times N$ symmetric

weight matrix. The weight matrix is denoted by $\boldsymbol{W}$, where the weight $\boldsymbol{W}_{ij}$ is the absolute value of the dot product or squared dot product of the normalized vectors $\boldsymbol{x}_i/\|\boldsymbol{x}_i\|$ and $\boldsymbol{x}_j/\|\boldsymbol{x}_j\|$. Among all the above methods that fall into the same framework (with possibly asymmetric weight matrices), this is the fastest to compute. However, it is somewhat simplistic when considering various outlier regimes. To speed up the algorithm, the authors mention using sketching to reduce computational complexity. In noisy settings, this algorithm struggles since it only takes the span of the top d points. Thus, Rahmani and Atia [90] propose a column sampling procedure, which iteratively projects the dataset and takes the most coherent point in an alternating fashion. This strategy is repeated until one recovers a sizeable set of points, and the underlying subspace is estimated from the recovered set of points using PCA. However, this method requires setting extra user-specified parameters, and in particular, requires an estimate of the noise level, which is not known in practice.

Xu et al. [112] developed the method of high-dimensional robust PCA (HR-PCA), which adaptively trims points to obtain a robust estimator. This method tries to maximize a robust variance estimator to capture subspace structures. Given a bound on the number of inliers, $\hat{t}$, the trimmed variance maximization is defined as

$$\hat{\boldsymbol{U}} \in \operatorname{argmax}_{\boldsymbol{U} \in O(D,d)} \max_{\substack{I \subset \{1,2,\ldots,N\} \\ \#(I)=\hat{t}}} \sum_{i \in I} \|\boldsymbol{U}^T \boldsymbol{x}_i\|^2. \quad (34)$$

The authors develop a randomized algorithm, where at each iteration, a point is removed with probability proportional to its variance in the current direction. The process is continued until one removes a prespecified number of points. The robust subspace can then be calculated from the remaining points. A deterministic version of this algorithm, called DHRPCA, was later developed in [31]. While the method can remove outliers with high influence on the PCA subspace, it is unintuitive as to why it should work in general settings with other more subtle types of outliers. Also, the algorithm requires the user to input the percentage of outliers, which is unknown in practice.

The idea of filtering outliers is also present in the work on adaptive compressive sampling (ACOS) [58]. Here, the authors subsample points and coordinates of the dataset, run outlier-pursuit or some other robust method, and filter outliers from the subsampled data. A subspace for the whole dataset can then be fit from the unfiltered points in full dimension.

The TORP algorithm [17], as discussed earlier in §II-D, can also be thought of as an outlier filtering method.

I. Exhaustive Subspace Search Methods

Another classical and simple way of robustly finding a subspace is to use RANSAC. In the celebrated paper, Fischler and Bolles [33] propose a general method where a subsample and estimator are iteratively improved over a dataset. Since this is such a common procedure, we review a RANSAC variant for RSR in more detail. The basic idea is to randomly sample $O(d)$ points and fit a d-subspace to them by using PCA. Then, one calculates the distances between all points and this subspace and labels inliers as those with distances less

than an input consensus threshold. If the set of inliers labelled in this way is sufficiently large (determined by comparison with an input consensus parameter), the algorithm returns this subspace. Otherwise, after a predetermined number of iterations, the algorithm outputs the model with highest consensus number.

Hardt and Moitra [41] proposed the RandomizedFind algorithm (RF), which is an exhaustive search method that is faster than RANSAC. For noiseless subspace recovery of a dataset $\mathcal{X} \subset \mathbb{R}^D$ with $N > D$ and where the inliers and outliers are in some general position as described in §III, they take random subsets, $\tilde{\mathcal{X}}$, of size D from $\mathcal{X}$ until one is found with $\operatorname{rank}(\tilde{\mathcal{X}}) < D$. Then this subset must contain at least $d + 1$ inliers and the indices of these inliers can be found by the non-zero elements of a vector in the kernel of $\tilde{\mathcal{X}}$. In order to deal with some noise, they propose replacing the condition $\operatorname{rank}(\tilde{\mathcal{X}}) < D$ with $\det(\tilde{\boldsymbol{X}}^T \tilde{\boldsymbol{X}}) < \delta$, where $\delta > 0$ is some small constant and $\tilde{\boldsymbol{X}}$ is the data matrix corresponding to $\tilde{\mathcal{X}}$. Finally, they also derive DeRandomizedFind (DRF), a deterministic polynomial time version of the RandomizedFind algorithm. Inspired by RF, Arias-Castro and Wang [2] studied a variant of RANSAC that subsamples $(d+1)$-subsets of points until a linearly dependent subset is found.

The scan statistic [36, 37] can also be used to exhaustively search for the underlying subspace in a structured way. This statistic measures the maximal number of occurrences in a sliding window of a fixed length. Arias-Castro et al. [3] proposed using the scan statistic in a multi-scale, multi-orientation fashion for the more general problem of robust manifold recovery. In this problem, inliers are uniformly sampled from a sufficiently smooth surface in $[0,1]^D$, outliers are uniformly distributed in $[0,1]^D$ and one needs to recover the underlying manifold.

J. Parallel Works

Here we discuss some different but related works to the RSR problem. Some of them have contributed to the development of RSR algorithms, while others have solved similar yet different problems.

One cannot consider RSR without acknowledging work done on robust orthogonal regression and its subsequent extension to RSR [70, 80, 83, 96, 109]. In this problem, one fits a $(D-1)$-dimensional subspace in $\mathbb{R}^D$, that is, an element of $G(D, D-1)$, using orthogonal distance as an error metric. The methods in this line of work use least absolute deviations to obtain robustness to corrupted data points.

Another body of related work, which was mentioned earlier is the RPCA problem [14, 15, 105]. A large variety of works have contributed to the study of this problem, such as robust energy minimization [22, 108], works on convex optimization [14, 15], online versions [32, 42, 87], nonconvex optimization [77, 108, 115], RANSAC methods [85], and many others [105]. Developments in RPCA and RSR seem to be somewhat complementary, and similar emergent themes can be seen in both.

Some other related problems, such as subspace clustering, synchronization, camera location estimation, and sparse vector estimation are discussed later in §VI.

III. Theoretical Recovery Guarantees

The theory behind algorithms for RSR has come in many forms, and it is hard to make sense of what the theory indicates about these algorithms. While there are many heuristic justifications for the methods discussed in the previous section, it is important to compare and contrast the various guarantees in order to gain an understanding of the most competitive methods. In this section, we attempt to distill the current recovery guarantees given for the RSR strategies. As a result, we hope to shed some light on where the field can go next. We leave the other important theoretical aspect of estimating the computational complexity of the algorithms, and in particular, rate of convergence of iterative schemes, to §IV.

In the following, we discuss exact recovery guarantees and near recovery guarantees. Exact recovery refers to a method's ability to exactly estimate the underlying subspace of a given noiseless inlier-outlier dataset. On the other hand, with noisy inliers-outliers datasets, one cannot hope to exactly estimate the underlying subspace. Instead, guarantees in the noisy case focus on near recovery, which means that the method finds a good approximation to the underlying subspace. Error of approximation in near recovery is typically bounded by a function of the noise level.

We explain the primary assumptions that seem to be shared among all works on RSR and the common RSR models in §III-A. We explain the theoretical work on RSR in §III-B-§III-G following the categories given in §II. We remark that §III-B also includes a discussion of the limitation of sequential methods. We conclude in §III-H with a comparison of the guarantees of these various methods for a specific statistical model of data.

A. Assumption on and Models of Data

The primary assumption for RSR is that inliers lie on or near a fixed underlying subspace, L_*, while outliers lie in the ambient space. For simplicity, we assume in most of the theoretical discussion the noiseless case, where the inliers lie exactly on the subspace. At times, we also comment on extensions to some noisy settings. We also assume in most of this paper that the dimension of L_*, d, is known. That is, we assume a noiseless (or sometimes noisy) RSR inlier-outlier dataset with known d, where one needs to recover (or nearly recover) the underlying d-subspace, L_*.

Here, we broadly describe the underlying statistical and combinatorial models involved in subspace recovery. An understanding of these models is essential in understanding the development of the field. We first describe several artificial examples in which the RSR problem is not well-defined and use them to motivate two basic principles for theoretical inlier-outlier datasets. These principles have to be followed in order to formulate well-defined theoretical data models.

In §III-A1, we lay out a principle for inlier distributions in the RSR problem. Then, §III-A2 gives a corresponding principle for outlier distributions. In §III-A3, we briefly mention the combination of these two principles to ensure well-defined models. Finally, in §III-A4, we carefully review specific theoretical data models that have been used for RSR in the context of these two principles.

1) Restrictions on the Inliers and a First Principle: This section will develop a principle for inlier distributions that ensures the RSR problem is mathematically well-defined. We start with a somewhat extreme case, where the noiseless RSR problem is ill-defined. We assume no outliers and inliers lying at the origin, which is demonstrated in Figure 2a. In this case, any linear subspace contains the inliers, and it becomes impossible to designate any one subspace as "underlying".

Figure 2b illustrates another example where the inliers lie in a lower-dimensional subspace of L_* and the problem is ill-defined. In this example, L_* is a 2-subspace in $\mathbb{R}^3$, the inliers concentrate on a 1-subspace of L_* and the outliers concentrate on a 2-subspace that intersects L_* at this 1-subspace. The issue here is that the outlier subspace seems more natural for describing the data than the "underlying" subspace L_*. Indeed, more data points lie in this subspace than in L_*. There are two key points that one should take away from this artificial example. First, our setting assumes a fixed parameter d, which we have designated as $d = 2$ in this example. If instead d was unknown, one could argue that the underlying subspace is the 1-subspace at the intersection of the two 1-subspaces. Second, the issue in this example, and also in some following examples, could be resolved by exchanging the labels of inliers and outliers. However, this avoids the main issue we are trying to illustrate here. We are interested in outlining a well-defined mathematical setting with restrictions on the sets labeled as inliers and outliers. In particular, this example illustrates that some restrictions must be placed on the inlier dataset.

Restrictions on the distribution of outliers in Figure 2b could also make it well-defined. Instead, this section focuses on restrictions on the inliers that make the problem well-defined. We also comment that the notion of a subspace that describes the whole dataset better than L_* is not completely well-defined yet but is somewhat conveyed by this figure. We will discuss this issue more carefully when describing how to restrict outliers in §III-A2.

From the previous examples, we see that the inliers cannot be too concentrated around lower dimensional subspaces of L_* and must instead fill out L_* in order to have a mathematically well-defined setting. We refer to this as the principle of *permeance* of the inliers, since the inliers must permeate the underlying subspace. We will later demonstrate how different works formulate this principle in different ways. Figure 2c presents a cartoon of permeated inliers when $d = 2$ and $D = 3$. We remark that non-uniformity of sampling within L_*, and possibly some very low level of concentration on low-dimensional subspaces of L_*, can be tolerated.

2) Restrictions on the Outliers and a Second Principle: In a similar fashion to the previous discussion, some restrictions must also be placed on the outliers to prevent them from giving rise to a subspace that may describe the data better than the underlying subspace, L_*. For example, assume that the inliers permeate the underlying subspace to some degree and the outliers have a similar distribution to the inliers on another low-dimensional subspace. A special case of this more general example is demonstrated in Figure 2d, where
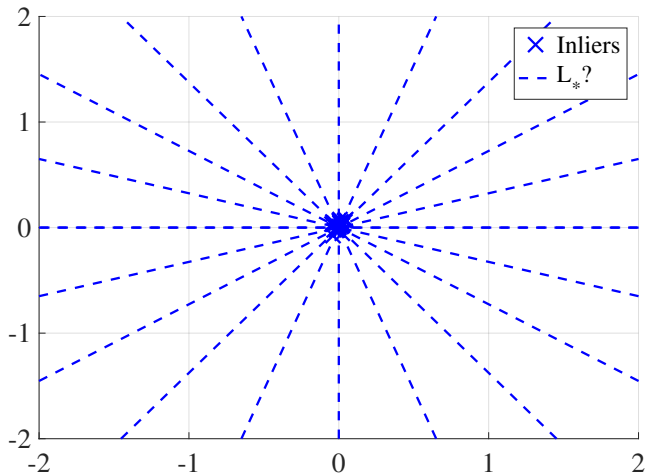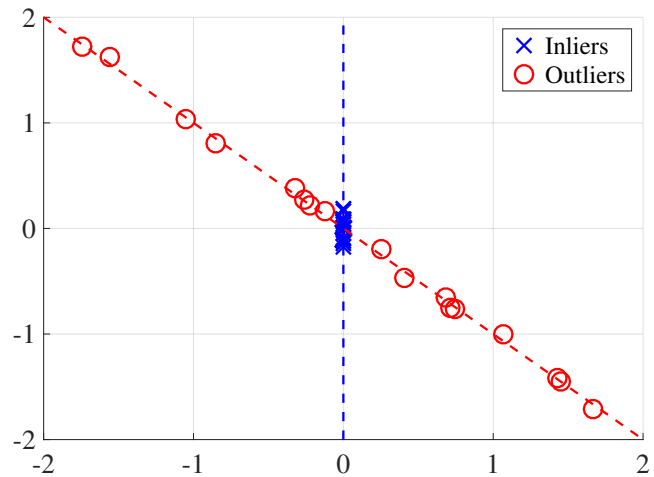
(a) The artificial data is composed of only inliers lying at the origin. Any line through the origin could be the underlying subspace.
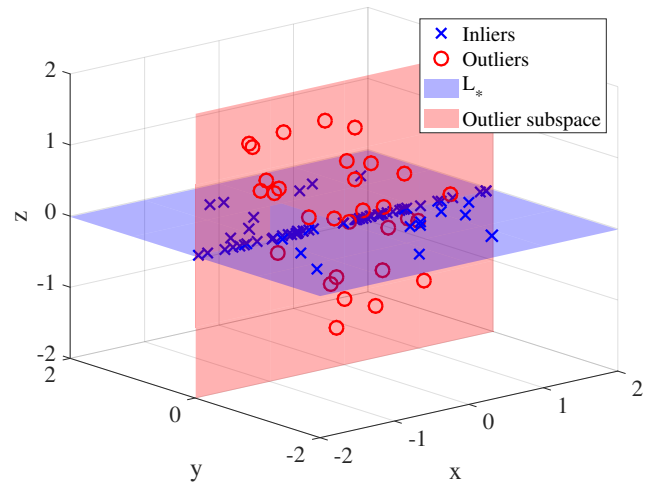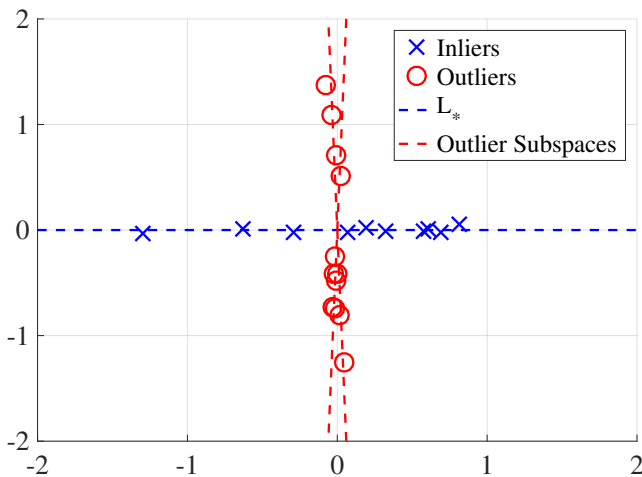
(b) L_* is the xy plane and the inliers concentrate around a line (the x axis). The outliers lie in the xz plane, and this subspace seems to capture more of the data points than L_*.
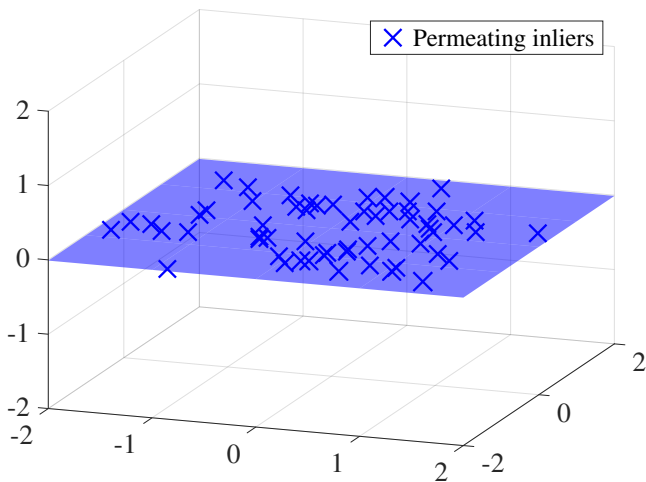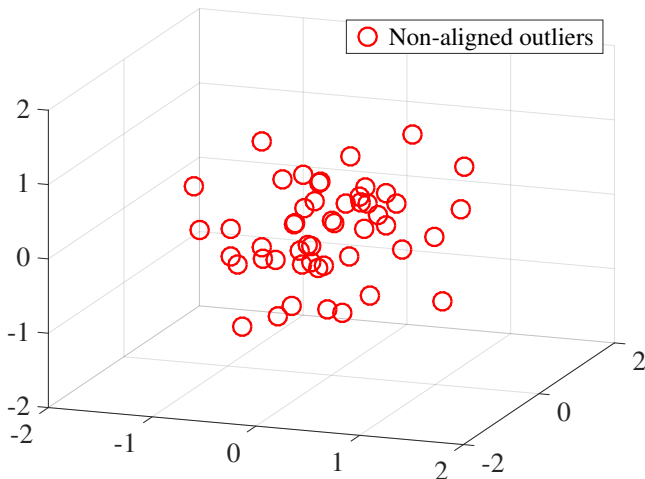
(c) A cartoon of permeating inliers.

(d) An example where the outliers lie near a line that may describe the whole dataset better than the inliers.

(e) A demonstration of 10 inliers around a line and two lines containing 6 outliers each. For near recovery, the line in between the two outlier lines may better represent the whole dataset.

(f) A cartoon of outliers with restricted alignment. As clarified in this section, there is some flexibility in the notion of restricted alignment, and the strong restriction shown here is needed only in some regimes.

Fig. 2: Examples clarifying the two principles that ensure well-defined models.
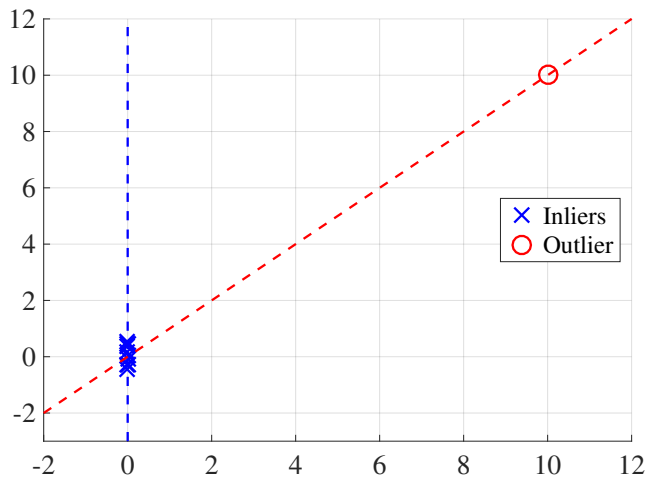
(a) An example of a dataset with inliers on a line and a single outlier of very large magnitude.

(b) The dataset of Figure 3a mapped onto the unit circle by normalizing each original data point by its Euclidean norm.
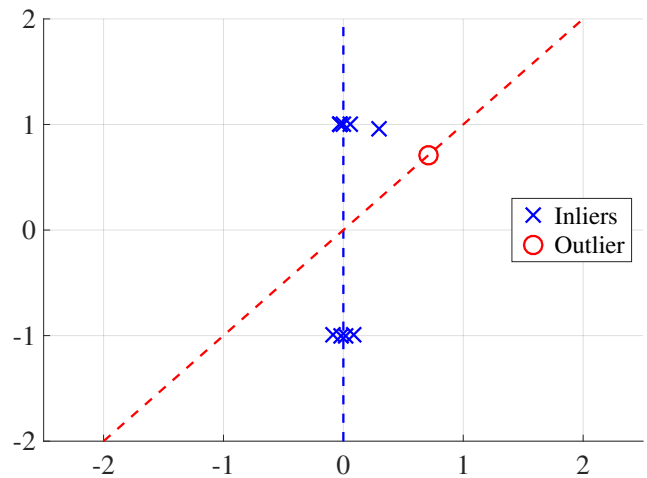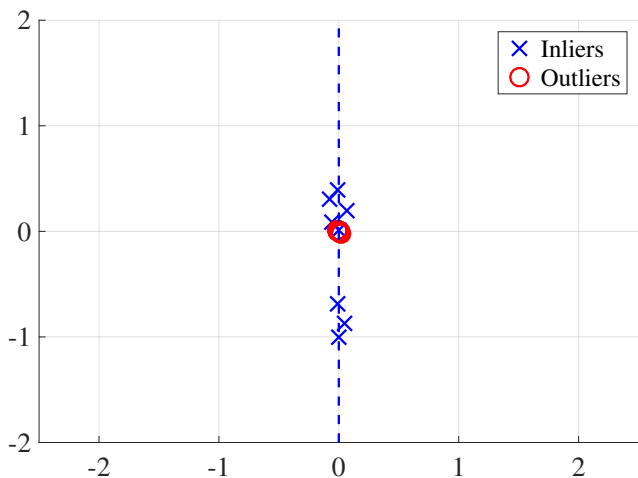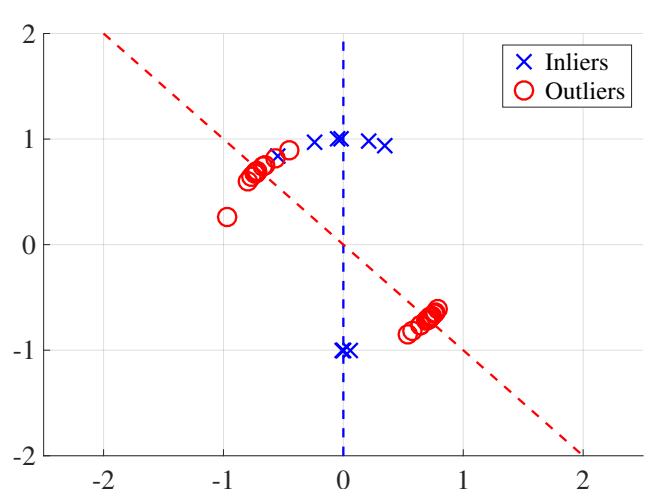
(c) An example of a dataset with outliers close to the origin, but near a different line than L_*. This line is unnoticeable since the magnitude of outliers is negligible.

(d) The dataset of Figure 3c mapped into the unit circle by normalizing each original data point by its Euclidean norm.

Fig. 3: Examples before and after mapping onto the unit circle by normalizing each data point by its Euclidean norm.

both subspaces are one-dimensional. One may claim that the outlier line describes the whole dataset better than the line that contains the inliers. As mentioned earlier, the notion that another subspace may fit the dataset better than the underlying subspace is not yet well-defined. First of all, for the noiseless case, the line L_* may still be more significant, in the sense that it contains more points. If on the other hand, the outliers in Figure 2d lie exactly on a line and not just near it, one could claim that the outlier line best represents the data. This debate boils down to two issues: 1) whether the number of inliers or outliers is large enough to determine which line represent better the data and 2) whether the larger relative magnitude of outliers contribute to their possible significance.

We start by focusing on the first issue, and we will discuss this second issue a bit later. We assume, in this noiseless version of the example, that the line with largest number of points best represents the data, and we will refer to this line as the "most significant".

The notion of most significant subspace is equivalent to the

subspace satisfying (4). However, as discussed in Section 1.1. of [55], this notion is problematic when the data points are even slightly noisy, where (4) needs to be replaced with (5). Figure 2e demonstrates such a problem in a simple case. Here, 10 inliers lie around the horizontal line, and 12 outliers lie around two other lines, each of which contains 6 points. Thus, while each of the outlier lines is less significant in terms of the number of points, the vertical line, which is close to the two outlier lines, has approximately 12 points near it and could be labelled as more significant. To avoid this problem, Lerman and Zhang [55], who have a model with several underlying d-subspaces, refer to a subspace as "most significant" if it contains more points than all other d-subspaces combined. We remark, though, that this notion applies to a very specific model and is not well-defined in general.

Assuming that this notion of most significant subspace is well-defined, the RSR problem can also be well-defined if one follows the principle of *restricted alignment* of the outliers. There are different ways of formulating this principle,

which affect the nature of the subsequent recovery guarantees. The examples in Figures 2b and 2e illustrate that one may need to exclude some sort of concentration of outliers around subspaces of dimensions at most d. This way, an outlier subspace cannot be the most significant subspace.

So far, we have ignored the effect of the relative magnitude of the outliers, although this can also influence the resulting conditions. In some works, restriction on alignment of outliers has to include some control on the ratio between the magnitude of outliers and inliers. If outliers have much larger magnitude than the inliers, they may have undue influence over a robust subspace criterion. Consequently, this sort of magnitude differential can make the problem ill-defined. However, it is possible to use "scale-invariant" methods to keep the problem well-posed in cases where there are no restrictions on the relative magnitude of outliers.

We demonstrate this issue with the special case of a dataset containing a single outlier of arbitrarily large magnitude and inliers lying on a one-dimensional underlying subspace in Figure 3a. The line through the large outlier might be viewed as the line that best represents the whole dataset since the distances of all inliers to this line are negligible. On the other hand, this outlier might be perceived as an adversarial one that should be excluded, especially since the rest of data points lie on another line. In this simple case, the outlier can be easily filtered out according to its large magnitude. There are also more general scale-invariant methods that give no weight to the magnitude of the data points, and thus one arbitrarily large outlier has little contribution when applying these methods.

We say that an RSR algorithm is scale-invariant if the output of the algorithm does not change after multiplying all the data points by different non-zero factors. A simple technique that results in scale-invariant algorithms is to initially normalize the data points by their Euclidean norms so that they lie on the sphere, S^{D-1}, and then apply any RSR method. Application of this normalization procedure to the simple dataset of Figure 3a is demonstrated in Figure 3b. We remark that it is unclear how to do this normalization procedure when there is missing data or when the setting is affine instead of linear.

This procedure, as well as other scale-invariant algorithms, may miss some important information in the magnitude of inliers and outliers. The special example in Figures 3c and 3d emphasizes this issue. Here, the outliers have very small magnitudes, and so the whole dataset is well-approximated by a line. However, the small outliers actually lie around a line that is quite different than the inlier line. Normalization of the dataset then emphasizes the outlier line more than the original inlier line. Thus, Figure 3d demonstrates that, even when applying scale-invariant algorithms, the alignment of outliers still has to be restricted, although there is not any consideration of their magnitude.

Employing an exhaustive subspace search method to minimize (4) is also scale-invariant. Indeed, in a well-defined setting, any such method would find the subspace containing most of the points, independently of any scaling of the data points. Scale-invariant search algorithms can also be developed for noisy RSR by trying to minimize variants of (5). For example, in this formulation, one can use the angles between data points

and the subspace rather than the orthogonal distance, since angles are scale-invariant.

We have discussed at length the restriction of outliers since there is some flexibility in enforcing it. Using the examples and concepts explained above, we clarify this flexibility. In the case of some scale-invariant algorithms, bounding the percentage of outliers can be enough to restrict the alignment. Similarly, in the case of a non-scale-invariant algorithm, it may be sufficient to bound the magnitude and percentage of the outliers. On the other hand, when considering regimes with high percentages of outliers, outliers cannot concentrate on or around a significant d-subspace for any algorithm. Notice that, following the earlier discussion in this section, this notion must also interact with the inlier permeance. For example, the inliers in Figure 2b may require stronger assumptions on outlier alignment than the inliers in Figure 2c. We further discuss this interaction in the next section. However, in general, the restriction on alignment is often formulated with respect to the outliers alone. A case with very restricted alignment, which is needed with high percentages of outliers and is especially needed with a non-scale-invariant algorithm, is demonstrated in Figure 2f. Here, no substantial subset of outliers lies near any low-dimensional subspace, and no outliers have exceptionally large magnitude.

3) Stability: the Combination of Permeance and Alignment: We refer to both the encouragement of permeance of the inliers and restriction of alignment of the outliers as the stability constraint of the model. An example of a stability constraint is demonstrated later in §III-C2. In this example, positive permeance and alignment statistics, $\mathcal{P}$ and $\mathcal{A}$ respectively, are formed so that higher values of $\mathcal{P}$ correspond to more permeated inliers, and lower values of $\mathcal{A}$ correspond to more restricted alignment of the outliers. A stability statistic is defined by a positive linear combination of $\mathcal{P}$ and $-\mathcal{A}$, and the stability constraint is a lower bound on the stability statistic. In the noiseless case, this bound is zero. We note that satisfying this constraint near the lower bound requires some tradeoff between inlier permeance and restricted outlier alignment. Nevertheless, each of the two quantities, $\mathcal{P}$ or $\mathcal{A}$, is computed with respect to only the inliers or outliers respectively, and thus the stability constraint does not fully explore the interaction between the configurations of inliers and outliers.

Some stability constraints imply an upper bound on the percentage of outliers, or equivalently, a lower bound on the percentage of inliers. Borrowing terminology from signal processing, Zhang and Lerman [119] define the signal-to-noise ratio (SNR) of the RSR problem as the ratio of the number of inliers to the number of outliers under a given stability constraint. For a given theoretical data model, algorithms can be compared by the lowest SNR under which they can still exactly recover the underlying subspace, or nearly recover it up to a certain error. The next subsection reviews some of these theoretical data models.

4) Specific Models of RSR: In this section, we explain several models under which lowest SNRs of algorithms can be compared. The first model uses arbitrary outliers. We remark that this model only works with scale-invariant algorithms, since there is no restriction on the magnitude of the outliers, and a single outlier can make non-scale-invariant algorithms

ill-posed. Here, the restriction of the alignment of outliers is only enforced by bounding their percentage, and thus the bound on SNR is relatively high. Xu et al. [111] claim that in this model the SNR has to be larger than d, and there are indeed degenerate examples where the problem is ill-defined when the SNR is d. If, on the other hand, one encourages permeance of inliers, then lower SNR can be obtained. More careful study of this model, including guarantees for existing and new algorithms, is needed. The authors plan to address this issue in a forthcoming paper [72].

Another model is that of inliers and outliers in general position (see two similar formal definitions in §III-E and §III-G). As explained later, Hardt and Moitra [41] show that in some sense the optimal SNR in this model is $d/(D-d)$. This is much lower than the case of arbitrary outliers since the outliers exhibit no linear dependencies. If the SNR is bounded from below by this optimal value, then Hardt and Moitra [41] reduce the noiseless RSR problem to finding a linearly dependent D-subset, which is not hard.

Only scale-invariant algorithms can have guarantees for the general position model, because again there is no restriction on the magnitudes of the inliers and outliers. However, there are three main drawbacks regarding the applicability of this model. First, in some real datasets, such as ones involving face images under different illuminating conditions or hand-written digit images (see some relevant discussion in §V), subgroups of outliers may lie within low-dimensional subspaces. Therefore, the general position model may not be relevant to some real datasets. Second, this model is well-formulated for exact recovery in the noiseless case and does not seem to easily extend to the noisy setting of near recovery. While Hardt and Moitra [41] propose using the threshold $\det(\tilde{\boldsymbol{X}}^T \tilde{\boldsymbol{X}}) < \delta$ in the noisy case, where $\tilde{\boldsymbol{X}}$ is the subsampled dataset, it is not at all clear when this would work. For example, this determinant would be small if one of the points in $\tilde{\boldsymbol{X}}$ had very small entries, even if $\tilde{\boldsymbol{X}}$ did not contain more than d inliers. It is also not clear how to set the threshold δ even for simple statistical models of noise, such as white Gaussian noise. Third, it is hard to determine how well many of the scale-invariant algorithms behave on the general position model. The only algorithms with results for this model are RF [41] and TME [118].

Many times, the analysis of RSR methods lends itself to considering certain statistical models of generating data. We believe studying such statistical models is important because it gives more insight into the performance of algorithms than just the worst case scenario in theorems with arbitrary outliers. Indeed, this sort of average case analysis illuminates differences in the breakdown of algorithms in low SNR regimes. For example, the haystack model [56] has been used to compare the theoretical guarantees of the various algorithms. The haystack model is a simple model for RSR data, where inliers and outliers both follow Gaussian distributions. In this model, inliers are symmetrically distributed on the underlying subspace with distribution $N(\boldsymbol{0}, \sigma_{\text{in}}^2 \boldsymbol{P}_{L_*}/d)$, while outliers have an isotropic Gaussian distribution in the ambient space, given by $N(\boldsymbol{0}, \sigma_{\text{out}}^2 \boldsymbol{I}/D)$. However, this model is limited since it captures a very particular scenario. The generalized

haystack model [73], in which outliers have a general and possibly degenerate covariance and inliers have a general covariance restricted to the subspace, captures more diverse scenarios, but the model is still quite specialized.

Theoretical results so far have emphasized exact recovery of subspaces in the noiseless RSR setting under the models discussed above. They often discuss extension of the results to near recovery with small amount of noise. Only a few existing works have focused on the truly noisy setting [17, 20, 75].

B. Sequential Methods and Projection Pursuit

A simple strategy for RSR is to fit one-dimensional directions sequentially. This strategy has been pursued in various lines of work, such as the projection pursuit method we discussed in (6) and (7). However, there is no guarantee that a sequential method will recover a stationary point of an energy for d-subspace recovery. For example, for projection pursuit, such an energy is given by $\sum_{j=1}^d \rho(\boldsymbol{v}_j^T \boldsymbol{X})$ over the set of orthonormal systems $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d \in \mathbb{R}^D$ [74]. In the PCA problem formulation, one can show that joint estimation and sequential estimation of principal components result in the same subspace. However, for other energies, joint and sequential estimation do not result in the same subspace. Also, the nonconvexity of the problem has caused works to guarantee convergence to local optima in each individual subproblem (formulated in (6) and (7)) [51] or convergence to a weak approximation of the global optimum of the joint energy $\sum_{j=1}^d \rho(\boldsymbol{v}_j^T \boldsymbol{X})$ [74, 76].

One shortcoming of sequential methods is the potential for compounding errors due to noise. Suppose we have a noisy data matrix $\boldsymbol{X}$, and we find a top component $\boldsymbol{v}_1$. Then, one can try to run the same algorithm again on the data matrix $\boldsymbol{Q}_{\boldsymbol{v}_1} \boldsymbol{X}$. However, due to noise, if we expect an optimal recovery error of approximately ϵ when estimating $\boldsymbol{V}_*$, then $\boldsymbol{v}_1$ should be $\Omega(\epsilon)$ from the underlying subspace. After projection and running again, the next component could be, at worst, $\Omega(2\epsilon)$ from the underlying subspace, and so on. To recover a d-dimensional subspace, their errors may accumulate to $\Omega(d\epsilon)$. In methods where one tries to find the orthogonal complement of the underlying subspace, such as [103], errors may even accumulate to $\Omega((D-d)\epsilon)$ if one tries to sequentially fit hyperplanes.

Further, even in the noiseless case, the first sequential component may be far from the underlying subspace. For example, this is a feature of the least absolute deviations energy. If one has a subspace of dimension $d > 1$ with points well distributed on the subspace, then one can mathematically show that the minimizer of (9) over $G(D, 1)$ will not be contained in the underlying subspace in general inlier-outlier settings.

We believe that projection pursuit methods generally suffer from the deficiencies present in sequential estimation. Overall, projection pursuit methods have lacked theoretical guarantees and have instead used heuristic arguments to justify them. We are unaware of substantial theoretical work on robust subspace recovery in this area.

C. Least Absolute Deviations

Most theoretical guarantees for RSR exist for methods aiming to minimize the least absolute deviations. We review them according to the different methods they are associated with.

1) Guarantees for Outlier Pursuit: Xu et al. [111] provide theoretical guarantee for recovery by OP, which is the program outlined in (14). The permeance of inliers discussed in §III-A is quantified by the inverse of an incoherence parameter. This parameter appears in other works on nuclear norm minimization, such as RPCA and matrix completion [12, 13, 14, 15]. The notion of incoherence and its parameter μ are defined for the low-rank inlier matrix $\boldsymbol{L}$ as follows:

Definition 1. *A rank d matrix $\boldsymbol{L}$ with $(1-\alpha)N$ non-zero columns, for $\alpha \in (0,1)$, and with SVD $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$, is said to be μ-incoherent if*

$$\max_i \|\boldsymbol{V}^T \boldsymbol{e}_i\|^2 \leq \frac{\mu d}{(1-\alpha)N}. \tag{35}$$

Here, $\boldsymbol{e}_i$ are the unit coordinate vectors.

In the special case of generating the inliers from a spherically symmetric Gaussian distribution within the underlying d-subspace, the incoherence parameter is $\mu = O(\max(1, \log(N)/d))$ [12].

We note that the parameter $1-\alpha$ is the fraction of inliers, which are represented by non-zero columns in $\boldsymbol{L}$, so α is the fraction of outliers and the SNR is $(1-\alpha)/\alpha$. Xu et al. [111] provided the following lower bound on the SNR for exact recovery by outlier pursuit:

Theorem 1 (Xu et al. [111]). *Suppose the data matrix $\boldsymbol{X} \in \mathbb{R}^{D \times N}$ can be represented as $\boldsymbol{X} = \boldsymbol{L} + \boldsymbol{C}$, where $\boldsymbol{L}$ has rank d and incoherence parameter μ, $\boldsymbol{C}$ is column sparse and supported on at most αN columns that are not in the column space of $\boldsymbol{L}$, and $\lambda = 3/(7\sqrt{N_{\text{out}}})$. Then, if*

$$SNR \geq \frac{121\mu d}{9}, \tag{36}$$

outlier pursuit recovers the matrices $\boldsymbol{L}$ and $\boldsymbol{C}$.

Suppose on the other hand that $\boldsymbol{X} = \boldsymbol{L} + \boldsymbol{C} + \boldsymbol{N}$, where $\boldsymbol{L}$ and $\boldsymbol{C}$ are as above, with $SNR \geq 1024\mu d/9$, and $\boldsymbol{N}$, the noise matrix, satisfies $\|\boldsymbol{N}\|_F \leq \epsilon$, then the output $\tilde{\boldsymbol{L}}$ and $\tilde{\boldsymbol{C}}$ of outlier pursuit satisfy $\|\tilde{\boldsymbol{L}} - \boldsymbol{L}'\|_F \leq 20\sqrt{N}\epsilon$ and $\|\tilde{\boldsymbol{C}} - \boldsymbol{C}'\|_F \leq 18\sqrt{N}\epsilon$, where $\boldsymbol{L}' + \boldsymbol{C}' = \boldsymbol{L} + \boldsymbol{C}$, $\boldsymbol{L}'$ has the same column space as $\boldsymbol{L}$ and $\boldsymbol{C}'$ has the same column support as $\boldsymbol{C}$.

Nevertheless, we remark that this theory is quite weak for the following reasons. First, the SNR for arbitrary outliers and permeated inliers is relatively weak (see [72]). Furthermore, it is unclear how to obtain lower SNR for other scenarios with more restriction on the alignment of outliers, where exact recovery can be obtained with significantly lower SNR (see for example Table I). Finally, the bounds of near recovery for noise are relatively large.

In general, algorithms aiming to minimize (9) are sensitive to even a single outlier with very large magnitude (without modifications such as normalization of data points to the sphere). However, since the nuclear norm is a very crude approximation of the rank, the contribution of an outlier, or more precisely, its component orthogonal to the underlying subspace, is similar to both parts of the cost function: $\|\boldsymbol{L}\|_*$ and $\|\boldsymbol{C}\|_{1,2}$. Since the constant λ of the cost function is often very small, the outlier column is included in $\boldsymbol{C}$ and not $\boldsymbol{L}$. Outlier pursuit is thus scale-invariant for sufficiently large SNR.

2) Guarantees for GMS and REAPER: Zhang and Lerman [119] consider the development of deterministic stability conditions that ensure subspace recovery by GMS, whose estimator was defined in (16). They also discuss the types of outliers that can make subspace recovery hard and provide visualizations of these (see Figure 1 in [119]). They then show that the deterministic stability condition holds under certain sub-Gaussian inlier-outlier mixture models as well as the haystack model with overwhelming probability. By introducing a perturbation argument, they extend their results to near recovery when the inliers lie near a subspace. Their restriction on the alignment of outliers is very strong, and, in practice, they require at least $1.5D$ outliers filling out the ambient space. If this condition is not satisfied, then GMS does not have good accuracy. Zhang and Lerman [119] provide three solutions to this, although it is not clear how well these would perform in general. In our numerical experiments in §V, we test their solution of adding $1.5D$ spherically symmetric Gaussian outliers in the ambient space.

The work of [56] on the REAPER algorithm, which uses the estimator given in (17), also gives a deterministic recovery result when a dataset satisfies a stability criterion. They define the permeance statistic $\mathcal{P}(L_*)$ of a dataset on the underlying subspace L_* as a measure of the notion of permeance of the inliers projected onto the subspace L_*. Note that this definition assumes inliers possibly near the underlying subspace and that is why they project them onto the subspace. They also define the alignment statistic $\mathcal{A}(L_*)$ that quantifies the restriction of the alignment of outliers. The definition of $\mathcal{P}(L_*)$ and $\mathcal{A}(L_*)$ appear in equation (2.1) and (2.3) of [56]. The stability statistic, $\mathcal{S}$, is defined as

$$\mathcal{S}(L_*) = \frac{\mathcal{P}(L_*)}{4\sqrt{d}} - \mathcal{A}(L_*). \tag{37}$$

In the noiseless case, their theory implies that positive stability at the underlying subspace L_* guarantees exact recovery of this subspace by REAPER. Their theory also provides a probabilistic lower bound on the stability statistic under the haystack model. This implies exact recovery with overwhelming probability under the SNR indicated in Table I.

In the general case of RSR, $\mathcal{S}(L_*)$ needs to be larger than what they call the total inlier residual with respect to L_*, which is defined by

$$\mathcal{R}(L_*) := \sum_{\boldsymbol{x}_i \in \mathcal{X}_{\text{in}}} \|\boldsymbol{P}_{L_*^{\perp}} \boldsymbol{x}_i\|. \tag{38}$$

When this condition is satisfied the REAPER solution approximates well the underlying subspace L_* in the following way.

Theorem 2 (Lerman et al. [56]). *Suppose $\mathcal{X}$ is a general RSR dataset in $\mathbb{R}^D$ with an underlying d-dimensional subspace L_*, $\tilde{\boldsymbol{P}}$ is a solution to the REAPER problem (17), and $\tilde{\boldsymbol{\Pi}} =$*

$U_d U_d^T$, where $U_d \in \mathbb{R}^{D \times d}$ is the matrix whose columns are the top d eigenvectors of $\tilde{P}$. Then,

$$\|\tilde{\Pi} - P_{L_*}\|_* \leq \frac{4\,\mathcal{R}(L_*)}{\max(\mathcal{S}(L_*) - \mathcal{R}(L_*), 0)}. \quad (39)$$

Notice that the fraction in (39) is only meaningful when $\mathcal{S}(L_*) > \mathcal{R}(L_*)$.

There is also an interesting noise-robustness analysis for the GMS and REAPER algorithms that is given in [20]. Here, the authors prove that the sample complexity of these algorithms is approximately the same order as that of the sample covariance for sub-Gaussian distributions. This observation implies nontrivial robustness to noise.

3) Guarantees for Nonconvex Formulations of Least Absolute Deviations: We discuss existing theoretical guarantees or the lack thereof for the following nonconvex least absolute deviation methods according to this order: R1PCA, the pure energy minimization in (9), FMS, GGD, TORP, and DPCP. These methods were laid out in §II-D2.

For general datasets, convergence for all of the following algorithms is proven to a stationary point at best. Furthermore, we do not know in general whether or not this stationary point recovers something useful. Because of this, some works have resorted to further restrictions on the data. These restrictions are used to show when the algorithms converge to an underlying subspace and also to show the speed of convergence.

The work of Ding et al. [24] on R1PCA was originally claimed to be convex, but they actually optimize a nonconvex problem formulation. Thus, they do not have guarantees of global optimality for their minimization and no guarantees of subspace recovery.

Lerman and Zhang [55] prove exact subspace recovery w.o.p. by minimization of the least absolute deviations energy (9) under a certain probabilistic model of data. The datasets considered involve a mixture model with i.i.d. inliers distributed uniformly on $S^{D-1} \cap L_1^*$ and i.i.d. outliers distributed uniformly on S^{D-1} and the intersection of S^{D-1} with $K - 1$ subspaces $L_2^*, \ldots, L_K^*$. It is further assumed that the asymptotic fraction of points on L_1^* is greater than the asymptotic fraction of points on $L_2^*, \ldots, L_K^*$ combined. This work shows the least absolute deviations energy can handle any fixed fraction of i.i.d. outliers distributed uniformly on S^{D-1}. However, this work only focuses on analysis of the pure minimization problem and not of an algorithm for minimizing it. Furthermore, its model is restrictive, and its estimates require large sample sizes.

Lerman and Maunu [53] provide some guarantees for the FMS algorithm, although they are somewhat limited. We remind the reader that the FMS procedure tries to directly minimize (9) using iteratively reweighted least squares. They prove that the FMS algorithm converges to a stationary point in general and is able to decrease the least absolute deviations energy monotonically from its starting point. However, they do not guarantee that this stationary point is a local minimum in general settings. They further show that the FMS algorithm can nearly recover an underlying subspace in two special settings: 1) when outliers are spherically symmetric and inliers are spherically symmetric within the underlying subspace or 2)

outliers are spherically symmetric or lie on a one-dimensional less significant subspace, and inliers lie on a significant one-dimensional subspace. In the first setting, the analysis shows that FMS can nearly recover the underlying subspace for any fixed fraction of outliers (less than 1). For both settings the convergence of FMS is locally r-linear. Nevertheless, the estimates in [53] require large sample sizes.

Maunu et al. [73] formulate a deterministic stability condition that guarantees nice behavior of the energy landscape of (9) in a local neighborhood around L_* (more details are described below). They also show that under this stability condition, a geodesic gradient descent (GGD) algorithm for (9) initialized in this neighborhood exactly recovers the underlying subspace. They further show that a similar deterministic stability condition ensures that the PCA d-subspace lies in this neighborhood. Therefore, GGD initialized by PCA has an exact recovery guarantee under both stability conditions simultaneously.

The stability condition was inspired by the previous ideas of [56] and focuses again on a difference of two statistics: an inlier permeance and outlier alignment. For simplicity, we discuss here only the noiselss case. The permeance and alignment statistics can be seen in (9) and (10) of [73]. Since the condition is local, a parameter $0 < \gamma < \pi/2$ determines how large of a neighborhood is considered. This neighborhood is defined in the following way:

$$B(L_*, \gamma) = \{L \in G(D, d) : \theta_1(L, L_*) < \gamma\}. \quad (40)$$

Here, $\theta_1(L_1, L_2)$ is the largest principal angle between two subspaces L_1 and L_2. Using the bounds given in [73], it is easier to interpret the following lower bound on the stability statistic:

$$\mathcal{S}(\gamma, L_*) \geq \cos(\gamma)\lambda_d \left(\sum_{x_i \in \mathcal{X}_{\text{in}}} \frac{x_i x_i^T}{\|x_i\|} \right) - \sqrt{N_{\text{out}}}\|X_{\text{out}}\|_2.$$
$$(41)$$

Here, $\lambda_d(\cdot)$ is the dth eigenvalue of the input matrix. The first term measures how well the inliers "fill out" the underlying subspace, while the second term measures how aligned the outliers are in any direction.

The stability condition for the noiseless case is positivity of this statistic. The theory outlined earlier can be precisely formulated as follows.

Theorem 3 (Maunu et al. [73])**.** *Suppose that an inliers-outliers dataset with an underlying subspace L_* satisfies $\mathcal{S}(\gamma, L_*) > 0$, for some $0 < \gamma < \pi/2$. Then, all points in $B(L_*, \gamma) \setminus \{L_*\}$ have a directional subdifferential strictly less than $-\mathcal{S}(\gamma, L_*)$, that is, it is a direction of decreasing cost. This implies that L_* is the only local minimizer in $B(L_*, \gamma)$. Suppose further that the initial GGD iterate is $L_1 \in B(L_*, \gamma)$. Then, for sufficiently small s, GGD with step size $s/\sqrt{k}$ converges to L_* with rate $\theta_1(L_k, L_*) < O(1/\sqrt{k})$.*

Under an additional "strong gradient condition" specified in (21) of [73], for sufficiently small s and sufficiently large K, GGD with step size $t^k = s/2^{\lfloor k/K \rfloor}$ linearly converges to L_*.

Initialization in this neighborhood is guaranteed by the following lemma, which is a consequence of the Davis-Kahan $\sin\Theta$ Theorem [21].

Lemma 1. *Suppose that, for a noiseless inliers-outliers dataset,*

$$\sin(\gamma)\lambda_d(\boldsymbol{X}_{\mathrm{in}}\boldsymbol{X}_{\mathrm{in}}^T) > \|\boldsymbol{X}_{\mathrm{out}}\|_2^2. \tag{42}$$

Then, the PCA d-subspace is in $B(L_*,\gamma)$.

The stability condition is shown to hold with overwhelming probability under a variety of models of data, and it is also shown to be stable with small noise. In particular, GGD is shown to have recovery guarantees almost on par with the strongest convex methods on the haystack model discussed later in §III-H. The downside for GGD is that it requires slightly larger sample estimates: $N = O(D^2\log(D))$ versus $N = O(D)$ for convex methods like REAPER and GMS. GGD also has a guarantee of recovery for any fixed percentage of outliers under this model in the large sample limit (when one allows $N \to \infty$).

Cherapanamjeri et al. [17] give theoretical guarantees for TORP with arbitrary outliers and noise, when the fraction of outliers is known. The authors prove that the algorithm works with arbitrary corruptions up to an SNR of order $\Omega(d)$, although the constants are quite poor.

Theorem 4 (Cherapanamjeri et al. [17]). *Suppose the data matrix $\boldsymbol{X} \in \mathbb{R}^{D\times N}$ can be represented as $\boldsymbol{X} = \boldsymbol{L}+\boldsymbol{C}$, where $\boldsymbol{L}$ has rank d and incoherence parameter μ, $\boldsymbol{C}$ is supported on at most γN columns that are not in the column space of $\boldsymbol{L}$, where γ is an input parameter for TORP. Then, if*

$$SNR \equiv \frac{1-\gamma}{\gamma} \geq 128\mu^2 d - 1, \tag{43}$$

the TORP algorithm linearly converges to a point that exactly recovers the column space of $\boldsymbol{L}$.

Suppose on the other hand that $\boldsymbol{X} = \boldsymbol{L} + \boldsymbol{C} + \boldsymbol{N}$, where $\boldsymbol{L}$ and $\boldsymbol{C}$ are as above and $\boldsymbol{N}$ is added noise. Then, the TORP algorithm linearly converges to a subspace $\boldsymbol{U}$ such that $\|(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^T)\boldsymbol{L}\|_F \leq 60\sqrt{d}\|\boldsymbol{N}\|_F$. Under the more restrictive assumptions that $\boldsymbol{N}$ has entries i.i.d. $N(0,\sigma^2)$ and $SNR \geq 1024\mu^2 d - 1$, TORP linearly converges to a subspace $\boldsymbol{U}$ such that $\|(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^T)\boldsymbol{L}\|_F \leq 4\sqrt{\log(d)}\|\boldsymbol{N}\|_2$ w.o.p.

Since results are only proven for arbitrary corruptions, the bounds for certain generative models of data (such as the haystack model) are weaker than those given in [56, 73]. We note that TORP linearly converges to the solution in all of the restricted settings in Theorem 4. The authors also have an analysis to noise that is similar to that in [20]. They show that the sample complexity is similar to that of PCA on the noisy inlier distribution.

DPCP [103], which solves the program in (11), is able to prove recovery of subspace structures under some deterministic conditions by finding a sequence of nested hyperplanes. However, the conditions are quite hard to interpret, especially when one is finding nested structures. It is even hard to calculate what the conditions mean for a given statistical model of data, such as the haystack model.

D. L_1-PCA

We are currently not aware of any recovery or robustness guarantees for L_1-PCA, which was outlined in (19) and (26). Recovery guarantees for the RPCA problem in (24), which is similar to the L_1-PCA problem of (19), are reviewed in [105].

E. Robust Covariance Estimation

For quantification of the robustness of covariance estimators, the study of breakdown points has been important [64]. Essentially, the robust covariances are consistent estimators of covariance matrices for elliptical distributions with nontrivial breakdown points. This means they can tolerate some percentage of arbitrary outliers and still estimate the underlying elliptical covariance well, which, in turn, means they could be able to estimate an underlying principal subspace well. However, the study of this principal subspace for RSR is only analyzed in [118].

These sorts of breakdown points hold for the estimation of covariances since the space of these matrices is non-compact and there is a notion of a covariance matrix with arbitrarily large magnitude. On the other hand, a similar definition of a breakdown point does not hold for subspace recovery since the Grassmannian is compact. The notion of lowest SNR allowing exact subspace recovery or sufficiently near recovery is clearly weaker.

Zhang [118] demonstrated that TME can also be used for subspace recovery. The stability condition in [118] requires a lower bound on the SNR as well as general positions of both inliers and outliers. We say that the inliers are in general position with respect to L_* if, any d of them are linearly independent. Similarly, we say that the outliers are in general position with respect to $L_*^\perp$ if, after projecting them onto L_* any $D - d$ of them are linearly independent. Using this definition, the theorem is formulated as follows:

Theorem 5 (Zhang [118]). *Assume that $\mathcal{X} \subset \mathbb{R}^D$ is a noiseless inliers-outliers dataset in $\mathbb{R}^D$ with an underlying d-subspace L_*. If the inliers are in general position with respect to L_*, outliers are in general position with respect to $L_*^\perp$, and $SNR > d/(D - d)$, then TME exactly recovers L_*.*

The theorem that extends subspace recovery by TME to noisy datasets is quite weak and hard to interpret, and so we do not state it here (see Theorem 3.1 in [118]). We remark that a clear advantage of TME is that it is scale-invariant. Indeed, it is obvious from (29) that scaling any data point by an arbitrary non-zero constant will not affect the estimator. Spherical PCA is also scale-invariant, as can be seen from (30), but it is not able to exactly recover subspaces like TME.

F. Other Energy Minimization and Filtering Outliers

The outlier removal energy of Xu and Yuille [113] does not have any associated guarantees for subspace recovery.

The bounds for the performance of HR-PCA and DHR-PCA are hard to interpret for arbitrary datasets [31, 112]. The authors chose to focus on a quantity called expressed variance (EV). Suppose one would like to measure the quality of an

orthogonal basis matrix $\boldsymbol{U} \in O(D,d)$ against the optimal subspace, represented by $\boldsymbol{U}_* \in O(D,d)$. Then, the expressed variance is given by

$$EV(\boldsymbol{U}) = \frac{\|\boldsymbol{U}^T \boldsymbol{X}_{\text{in}}\|_F^2}{\|\boldsymbol{U}_*^T \boldsymbol{X}_{\text{in}}\|_F^2}. \tag{44}$$

The expressed variance takes values between 0 and 1, and it measures the proportion of underlying variance captured by the basis. Xu et al. [112] prove lower bounds on the expressed variance, although these bounds are quite weak. For example, in the case of spherically symmetric Gaussian inliers on a subspace and spherically symmetric Gaussian outliers, their lower bound on EV is 0.09 [53] (while an EV of 1 amounts to exact recovery).

Arias-Castro et al. [4] guaranteed their method for removing outliers in the setting of robust manifold clustering. In the case of robust recovery of a single manifold, their theorem implies exact identification of outliers when the inliers are uniformly sampled from a τ-neighborhood in $[0,1]^D$ of a certain C^2 submanifold of $[0,1]^D$, the outliers are uniformly sampled from the complement of that τ-neighborhood in $[0,1]^D$ and the SNR is of order $\Omega(\max(\log(N) \cdot N^{-2(D-d)/(2D-d)}, \tau^{D-d}))$. Here, the τ neighborhood contains all points that have distance less than or equal to τ with the submanifold.

Soltanolkotabi and Candès [93] guaranteed their method for removing outliers, which is similar in spirit to [4, 16], in the setting of noiseless robust subspace clustering. In the case of a single subspace recovery, their theorem implies exact removal of outliers when the outliers are uniform in S^{D-1}, the N_{in} inliers are uniform on the intersection of S^{D-1} with a uniformly random d-subspace and the SNR is of order $\Omega(\frac{d}{D} \cdot ((\frac{N_{\text{in}}-1}{d})^{\frac{cD}{d}-1} - 1)^{-1})$, as long as $N < e^{c\sqrt{D}}/D$. In this method and the below work by Rahmani and Atia [90], the authors need to assume that the inlier subspace is uniformly random, which is a stronger assumption than other methods make.

You et al. [116] proved exact recovery of outliers in the noiseless setting of robust subspace recovery under certain conditions. They did not verify that these conditions hold under a generative model. It is interesting to note that one of the conditions, namely equation (7) in [116], is reminiscent of the stability condition of [56] for exact recovery in the noiseless case, that is $\mathcal{S}(L) > 0$, where $\mathcal{S}(L)$ is defined in (37).

Rahmani and Atia [90] prove recovery by CP with over-whelming probability in the same setting as Soltanolkotabi and Candès [93] but with a single random subspace, where they achieve SNR on the order of $\Omega(\frac{dD}{\sqrt{D}(D-d^2)} \frac{1}{\sqrt{N_{\text{out}}}})$. The authors also prove a recovery result for the special case of inliers uniform on the intersection of S^{D-1} with a uniformly random d-subspace and a small percentage of outliers distributed close to a random line. They also prove that CP can recover a set of inliers with small amounts of additive Gaussian noise in these models. However, the theory is lacking in some important regards. First, only very special models are considered, and it is hard to see how things perform in general. A further issue is that, in the noisy case, the span of a core set of recovered inliers may not represent the underlying subspace very well.

Thus, while they may be able to find a subset of the inliers, they do not give bounds on subspace approximation error for their subspace identification algorithm.

G. Exhaustive Subspace Search

Hardt and Moitra [41] show that RF and DRF, which were discussed in §II-I, can recover a subspace in the noiseless case for very low SNRs if the dataset is in general position with respect to the underlying subspace L_*. That is, any D data points are linearly independent if and only if at most d of them are inliers from L_*. This means that L_* is the only low-dimensional structure in the data. Note that this definition is similar but different than the one in §III-E, where inliers are in general position with respect to L_* and outliers are in general position with respect to $L_*^{\perp}$. Their theorems for RF and DRF are formally stated as follows (with an improvement on the expected number of iterations by Arias-Castro and Wang [2]):

Theorem 6 (Arias-Castro and Wang [2], Hardt and Moitra [41])**.** *Assume that $\mathcal{X} \subset \mathbb{R}^D$ is a noiseless inliers-outliers dataset in $\mathbb{R}^D$ with an underlying d-subspace L_*. If $\mathcal{X}$ is in general position with respect to L_* and $SNR > d/(D-d)$, then RandomizedFind outputs L_* with expected number of iterations that is $O(1)$, and DeRandomizedFind outputs L_* in polynomial time.*

On the other hand, if $SNR < d/(D-d)$, the problem becomes small set expansion hard. The small set expansion problem is conjectured to be NP-hard.

There is no existing theory for RandomizedFind and De-RandomizedFind in noisy settings.

As long as the noiseless problem is well-defined, RANSAC will succeed in finding the underlying subspace. However, in low SNR regimes, the computational time becomes an issue, as we will discuss in the next section. Further, when the parameters are set correctly, one can show that near recovery is possible with RANSAC under further assumptions on the alignment of outliers [72].

Arias-Castro et al. [3] proved that their mutli-scale, multi-orientation scan statistics may recover inliers sampled uniformly from a d-dimensional graph in $[0,1]^D$ of an m-differentiable function, when the outliers are uniform in $[0,1]^D$ and the SNR is $\Omega(N^{-m(D-d)/(d+m(D-d))})$. They also mention results for other kinds of surfaces.

H. Recovery with the Haystack Model

In Table I, we compare the various theoretical guarantees under a Gaussian model of data. This is one of the simplest models to compare the theoretical SNR of algorithms outside of the worst-case outliers (a table for the latter case will be provided in [72]). Here, inliers are distributed i.i.d. $N(\boldsymbol{0}, \sigma_{\text{in}}^2 \boldsymbol{P}_{L_*}/d)$ and outliers are distributed i.i.d. $N(\boldsymbol{0}, \sigma_{\text{out}}^2 \boldsymbol{I}/D)$. Under this model, we can compare the various recovery guarantees given in the works outlined throughout the whole section. The results for the haystack model are summed up well in Table 1 of [73], which is extended to more methods in Table I. We use the earlier abbreviations from the text.

GGD	$N_{\text{in}}/N_{\text{out}} \geq \max\left(4\sqrt{2}\frac{\sigma_{\text{out}}}{\sigma_{\text{in}}}\frac{d}{\sqrt{D(D-d)}}, 2\frac{\sigma_{\text{out}}^2}{\sigma_{\text{in}}^2}\frac{d}{D}\right) (N = O(D^2))$ $N_{\text{in}}/N_{\text{out}} \gtrsim 0 \ (N \to \infty)$ *Deterministic condition, results for a variety of data models.*
FMS	$N_{\text{in}}/N_{\text{out}} \text{``} \gtrsim \text{''} 0 \ (N \to \infty)$ *Approximate recovery for large samples from spherized haystack or from two one-dimensional subspaces on the sphere.*
REAPER	$N_{\text{in}}/N_{\text{out}} \geq 16\frac{\sigma_{\text{out}}}{\sigma_{\text{in}}}\frac{d}{D} \ (N = O(D), 1 \leq d \leq (D-1)/2)$ *Deterministic condition, results for haystack where $d < (D-1)/2$.*
GMS	$N_{\text{in}}/N_{\text{out}} \geq 4\frac{\sigma_{\text{out}}}{\sigma_{\text{in}}}\frac{d}{\sqrt{(D-d)D}} \ (N = O(D))$ *Deterministic condition, results for haystack that extends to elliptical outliers.*
OP	$N_{\text{in}}/N_{\text{out}} \geq \frac{121d}{9}O\left(\max(1, \log(N)/d)\right) \ (N = O(D))$ *Deterministic condition (formulated for arbitrary outliers) with last term in the above formula replaced by an inlier incoherence parameter μ.*
HR-PCA	$N_{\text{in}}/N_{\text{out}} \to \infty \ (N \to \infty)$ *Weak lower bound on the expressed variance, requires fraction of outliers as input.*
TME/(D)RF	$N_{\text{in}}/N_{\text{out}} > \frac{d}{D-d} \ (N = O(D))$ *Result for "general-position" data, but does not extend to noise.*
TORP	$N_{\text{in}}/N_{\text{out}} \geq 128d \max(1, \log(N)/d)^2 \ (N = O(D))$ *Deterministic condition (formulated for arbitrary outliers) with last term replaced by an inlier incoherence parameter μ, requires fraction of outliers as input.*
CP	$N_{\text{in}}/N_{\text{out}} \gtrsim d/(D-d^2) \ (N = O(D), d < \sqrt{D})$ $N_{\text{in}}/N_{\text{out}} \gtrsim 0 \ (N \to \infty, d < \sqrt{D})$ *Exact recovery for the spherized haystack model with a random inlier subspace and $d < \sqrt{D}$, recovery guarantees for a special model of outliers around a line.*

TABLE I: Comparison of lower bounds on the SNR and a summary of guarantees. The properties of each algorithm are described in two rows. The first row provides the largest lower bounds on the SNR in the haystack model for different orders of N. The second row briefly comments on other guarantees under possibly different models.

Here, we also display the sample size necessary for the probability of recovery in each result to become close to 1. Notice that for sample sizes $N = O(D)$, the optimal SNR for all $1 \leq d < D$ is on the order of $d/\sqrt{D(D-d)}$. Notice that GGD achieves this optimal bound, but requires $N = O(D^2)$, and so it has guarantees that are almost on par with state-of-the-art convex ones. If we let $N \to \infty$, we see that two methods, GGD and CP, can tolerate any fixed fraction of outliers. The FMS can also tolerate any fixed fraction of outliers but can only nearly recover the underlying subspace up to a regularization dependent precision. Although the result for FMS dealt with the spherized haystack model, the result can also be extended to the non-spherized haystack model with minimal effort.

We remark that CP is included here even though its model assumes a uniformly random underlying subspace, which makes the analysis easier. Nonetheless, when $N \to \infty$ this assumption makes no difference. We also note that the CP theory require that $d < \sqrt{D}$, which is a major restriction compared to other methods. The guarantees for REAPER have the weaker requirement of $d < (D-1)/2$. Other methods can tolerate any $d < D$.

IV. COMPUTATIONAL COMPLEXITY AND MEMORY REQUIREMENTS

An important tradeoff in robust subspace recovery explores the accuracy of an algorithm versus its computational complexity or memory requirement. Because of this, it is necessary to clearly state the complexity and memory requirement of the various algorithms to see how they all scale. The complexity requirements for the various RSR algorithms are given in Ta-

ble II. For ConstApprox, nnz refers to the number of non-zero entries in the input matrix $\boldsymbol{X}$, and the number ϵ is the desired approximation accuracy. For ACOS, the numbers ρ_1 and ρ_2 are the row and column sampling fractions, respectively. We also examine the memory requirement for RSR algorithms in Table III. The parameters for ACOS are the same as those in Table II.

We first discuss at length the results presented in Table II. Many algorithms are iterative and for simplicity we assume that the number of iterations is a constant, which we denote by T, but this is in general problematic. Indeed, for nonconvex algorithms, we expect cases of very slow convergence since the problem is NP-hard. The following algorithms are iterative: GMS, REAPER, R1PCA, TORP, MDR, OP, FMS, GGD, TME, Dümbgen's M-estimator, RANSAC, and RF. Among these, under certain conditions, only GMS [119] and GGD [73] have guarantees for r-linear convergence and TORP [17] has a guarantee for linear convergence. Also, FMS [53] has a weak guarantee of local r-linear convergence in a very special case. The conditions for GGD can be weakened at the expense of a sublinear convergence rate, and OP and MDR have sublinear convergence in general. For convergence rate, we present the number of iterations required to achieve ϵ-accuracy for the given iterative algorithms. For the online algorithms, we use T to denote the number of passes over the dataset, which is often very high.

The worst complexities are for the maximization L_1-PCA algorithms [67]. The exact maximization L_1-PCA algorithms run in $O(N^{\text{rank}(\boldsymbol{X})})$ for $N \geq D$ [67] and $O(2^N)$ for $D < N$. It is important to note that an algorithm running in $O(N^D)$ or $O(2^N)$ is not efficient at all for big datasets.

Method	Complexity	Convergence Rate
Maximization L_1-PCA [67]	$O(N^{\mathrm{rank}(\boldsymbol{X})})$ or $O(2^N)$	No iteration
Dümbgen's M-estimator [27]	$T \cdot O(N^2 D^2)$	No result[1]
Spatial Kendall's tau [106]	$O(N^2 D^2)$	No iteration
SRO [116]	$O(N^2 D + N^3)$	No iteration
CP [90]	$O(N^2 D)$	No iteration
GMS [119]	$T \cdot O(N D^2 + D^3)$	r-linear convergence under the 2-subspaces criterion
TME [118]	$\dots$	No result[1]
RF [41]	$T \cdot O(D^3)$	$O(1)$ w.h.p. when SNR$\geq d/(D-d)$ in noiseless RSR [2]
REAPER [56]	$T \cdot O(N D^2)$	No result[2]
OP [74, 111]	$T \cdot O(N D^2)$	$O(\epsilon^{-1/2})$
MDR [74]	$\dots$	$O(\epsilon^{-1/2})$
(D)HR-PCA [31, 112]	$\dots$	$O(1)$
RMD [38]	$T \cdot O(D^2)$	No result
MKF [121]	$T \cdot O(Dd)$	No result
R1PCA [24]	$T \cdot O(NDd)$	No result
FMS [53]	$\dots$	No general result, local r-linear convergence for special model
GGD [73]	$\dots$	$O(\epsilon^{-2})$ under stability condition, r-linear under further condition
Projection pursuit [1, 18, 44, 51, 57]	$\dots$	No result
TORP [17]	$\dots$	Linear convergence in the settings of Theorem 4
RANSAC [33]	$\dots$	$O(1)$ w.h.p. when SNR$\gtrsim d$ in noiseless general position RSR [2]
SPCA [63]	$O(NDd)$	No iteration
ConstApprox [19]	$O(\mathrm{nnz}(\boldsymbol{X}) + \mathrm{poly}(d/\epsilon))$	No iteration
ACOS [58]	$T \cdot O(N_{\mathrm{in}} Dd + \rho_1 \rho_2 ND \max(\rho_1 D, \rho_2 N))$	$O(\epsilon^{-1/2})$

TABLE II: Complexity of the various RSR algorithms with constant iteration count T.

MKF [121]	$O(dD)$
RMD [38] RF [41]	$O(D^2)$
ConstApprox [19]	$O(\mathrm{nnz}(\boldsymbol{X}) + Dd)$
ACOS [58]	$O(\rho_1 \rho_2 ND + \rho_1 Dd)$
R1PCA [24] FMS [53] GGD [73] SPCA [63] RANSAC [33] Projection pursuit [1, 18, 44, 51, 57, 74] TORP [17] OP [111] L_1-PCA [67]	$O(ND)$
CP [90] SRO [116]	$O(N^2 + ND)$
REAPER [56] GMS [119] MDR [74] TME [118] Spatial Kendall's tau [106] Dümbgen's M-estimator [27]	$O(ND + D^2)$

TABLE III: Memory requirement of the RSR algorithms.

Other very slow algorithms are Dümbgen's M-estimator [27] and spatial Kendall's tau [106] that run in $T \cdot O(N^2 D^2)$ and $O(N^2 D^2)$ time, respectively. Calculation of the spatial Kendall's tau matrix is more efficient because there is no iteration.

[1]All results on TME also apply to Dümbgen's M-estimator. For TME, Kent and Tyler [49] proved convergence without rate guarantees in a setting that may fit near recovery in noisy RSR, and Zhang [118] proved convergence without rate guarantees to a singular matrix in a setting for exact recovery in noiseless RSR. We note that r-linear convergence was proved for the similar Maronna M-estimator in a setting that may fit near recovery in noisy RSR [5]. Lemma 1 of [39] proves global linear convergence of a regularized version of TME, but the required lower bound on the regularization parameter seems impractical for RSR.

[2]Lerman et al. [56] proved convergence with no rate guarantee for the REAPER procedure.

For CP, we show the complexity of calculating the full Gram matrix on all the points in $\mathcal{X}$ in the full dimension D. The authors advocate using a random projection and column subsampling to decrease complexity, but these ideas can be extended to many of the other methods listed here, as was done in [58]. Using these strategies also tends to decrease the accuracy of the given algorithm.

The next slowest algorithms run in $T \cdot O(ND^2)$ or $T \cdot O(D^3)$ time. For example, GMS, REAPER, and TME must calculate the full covariance, which takes $O(ND^2)$ time [56, 118, 119]. TME and GMS require matrix inversions, which require $O(D^3)$ time [118, 119]. RF requires a determinant calculation/solving a system of linear equations, which takes $O(D^3)$ time [41]. Finally, solving OP or MDR using a method such as proximal gradient descent takes $T \cdot O(ND^2)$ time.

Other algorithms operate in complexity $T \cdot O(NDd)$. This is also the complexity of using the power method to compute the PCA subspace (through the top d singular vectors of the data matrix). These methods include FMS [53], GGD [73], and TORP [17]. Although the RANSAC variant of Arias-Castro and Wang [2] runs in $T \cdot O(Dd)$, we believe that the algorithm may not be as stable as classical RANSAC [33]. Therefore, we display the complexity of classical RANSAC in Table II. Arias-Castro and Wang [2] bound the number of iterations required for their variant of RANSAC to exactly recover the underlying subspace when SNR$\geq O(d)$, the data is noiseless, and the data is in general position with respect to the underlying subspace. They also show that the number of iterations becomes exponential in d for lower SNR under the same assumptions. Despite the fact that the arguments of Arias-Castro and Wang [2] were proven for their variant of RANSAC, one can use these arguments for the classical RANSAC paradigm as well.

ConstApprox [19] is able to account for sparse input ma-

trices, and thus operates in time dependent on the number of non-zero entries in $\boldsymbol{X}$, which is denoted by $\mathtt{nnz}(\boldsymbol{X})$. In the case of a dense matrix $\boldsymbol{X}$, this complexity is still $O(ND)$, which is approximately $O(NDd)$ when d is small. Although this method may be fast, it has no guarantee of recovering a subspace.

Beyond the $O(NDd)$ limit for exact algorithms, some have tried to pursue even faster algorithms for approximating the underlying subspace. For example, the work of [58] uses row and column subsampling of the matrix to reduce the N and D factors and speed up computational time. The outliers can then be identified resulting in a speed-up of the algorithm. However, in this case, it makes the theoretical guarantees of any algorithm used in the subsampled case somewhat weaker. Further, after filtering the outliers, one must still calculate the inlier subspace, which takes at worst $O(N_{\mathrm{in}}Dd)$. Thus, depending on the number of inliers, it may not improve much over $O(NDd)$. Indeed, since ACOS is an approximation of OP, and OP is only guaranteed for large percentages of inliers, this can still take quite long.

We also include two online algorithms in our comparisons. The Median K-Flats (MKF) algorithm [121] operates in $T \cdot O(Dd)$ time, while a slower robust mirror descent (RMD) algorithm is given in [38], which operates in $T \cdot O(D^2)$ time. However, these algorithms must pass over the data at least once, and so there is a hidden factor of N in the iteration complexity for each of these algorithms. Further, since the sample complexity for these methods is not known, the number of iterations (or passes over the data) required for these methods can be quite large, and, in practice, can require even more time than the other $T \cdot O(ND^2)$ methods.

Next, we discuss the memory requirements presented in Table III. Here, the factor of $O(N^2)$ seen for CP and SRO is typical of all strategies that follow [16], due to the need to store the $N \times N$ weight matrix. The $O(D^2)$ factor is typical of methods that need to store a covariance type estimator [27, 38, 56, 106, 118, 119], methods that use the lifting convex relaxation technique [74], or methods that require a set of D points [41]. The $O(ND)$ factor is typical of methods that need to store the whole data matrix in memory or calculate the SVD of a dense matrix. Online algorithms may have improved memory because they can stream the data and only need to store an estimator at each iteration, which is the case for MKF [121]. ConstApprox [19] improves over other algorithms by accounting for sparse inputs. Finally, ACOS [58] subsamples the input matrix and reduces the amount of memory needed when running OP.

V. Numerical Experiments and Applications

Numerical experimentation is very important for proper evaluation of RSR algorithms. In this section, we outline what has currently been done to evaluate RSR algorithms on both synthetic and real datasets and what remains to be done.

A fundamental issue of the RSR problem is how to measure accuracy. The use of energy-based metrics, which may use the energies described in §II-B–§II-H, is problematic since they are inherently tied to the methods that optimize them.

For example, if we wanted to evaluate subspaces by their least absolute deviations energy, we would expect the least absolute deviations algorithms to give lower energy than another method not designed to optimize that energy. For synthetic experiments, where one knows the underlying subspace, an easy choice of metric is the subspace's distance from ground truth [53, 56, 119]. For real data, the metric depends on the application. For example, when using RSR for robust dimension reduction for enhanced clustering [53], the actual metric by which RSR algorithms are compared is clustering accuracy. This demonstrates that accuracy should be determined by application, not by some general energy.

In the following subsections, we will discuss the application of RSR algorithms on data examples. First, §V-A will discuss what experiments have been done with real data and how to evaluate them. Then, §V-B will discuss experiments on datasets that reflect both synthetic settings of theoretical interest and stylized applications as a way to compare RSR algorithms.

A. Experiments with RSR on Real Datasets

Experimentation with RSR methods on real data is somewhat lacking due to the fact that it is a general purpose tool rather than a solver for any specific application. We can compare this to the more classical subspace modelling tool of PCA. PCA is a natural and ubiquitous data processing method, due to the fact that it can reduce the dimension of a dataset and also provide an orthogonal set of descriptive directions within the data. As such, PCA is not suited to completely solve any one problem, even though it can give insight through its descriptive factors and can act as a valuable dimension-reduction submethod. However, as discussed earlier, PCA is not robust to outliers within a dataset.

RSR should mimic the applicability of PCA and be a general purpose tool for dimension reduction, while at the same time not being as sensitive to corrupted data. In this way, the hope would be that an RSR algorithm would perform as well or better than PCA on most, if not all, datasets that require some form of dimension reduction. We remark that if one wants descriptive robust orthogonal factors with reduced dimension, then one may use PCA on the projection of the dataset, or its estimated inliers, onto the subspace obtained by an RSR algorithm.

Because RSR is such a general purpose tool, it is hard to point to any one stand out application. And, as we will discuss in the coming subsections, most of what has been done in the literature is quite artifical. In what has been done, we find the dimensionality reduction and denoising aspects of RSR algorithms to be the most intriguing. An important thrust for future research should be testing RSR algorithms in more real data scenarios. In particular, it would be useful to compile a database of example datasets to further test RSR algorithms as a dimension-reduction preprocessor.

In the following two subsections, we will outline experiments that have been done with RSR algorithms on real datasets. In §V-A1, we discuss the use of RSR algorithms for dimension reduction and data preprocessing. Then, in

§V-A2, we discuss the application of RSR algorithms to image datasets.

1) Dimensionality Reduction for Data Preprocessing: One intriguing property of PCA is its ability to reduce dimensionality of data while simultaneously reducing noise [9, 47]. Analogously, it has also been found that robust subspaces can have great descriptive power in the presence of noise and outliers. For example, the potential application of dimension reduction by RSR algorithms in astrophysics data was first explored in [11], and then later considered again in [53]. Lerman and Maunu [53] also demonstrate the descriptive power of RSR on clustering activity time series. More examples of robust dimension reduction for classification and regression can be seen in [45].

McCoy and Tropp [74] test their low-leverage decomposition (which is the same as OP) on Fisher's iris data. They show that a low-dimensional, robust subspace can describe the observations from one of the flower varieties quite well. In this experiment, they use a dataset with many observations from one flower type and "corrupt" the sample with observations of other flower types. The results visually show that RSR can capture more variation of the inlier flower type than PCA, although the authors do not give a quantitative measure of this.

Other work has considered using the RSR representation for visualization of genomics data [79, 86]. Here, outliers are filtered and PCA is done on the resulting datasets. In particular, Novembre et al. [79] show that this combination of filtering and PCA yields insightful visualizations that compare genes and geography.

In all of these experiments, RSR is a useful off-the-shelf tool for dimension reduction, data preprocessing, and visualization. Some quantification of the success of RSR methods appears in [53] and [45], although this is only done for a few datasets. More extensive experimentation with a large database is needed to study the effectiveness of RSR as such a tool.

2) Image Data: A popular task in machine learning is recognition of handwritten digits. Inspired by this, Xu et al. [111] considered a stylized experiment to show the capability of an RSR method to find a descriptive subspace to recognize differences between 1's and 7's. However, this experiment is only visual and does not have any quantitative measures of performance.

Many researchers have also tried to apply RSR algorithms to video surveillance data [119]. However, we argue that this is not a proper application of RSR algorithms, and it seems that RPCA, which addresses sparse-corruptions, models this application better. And indeed, RPCA works have shown impressive results on video surveillance data [42, 43].

Other works have studied the use of RSR algorithms on datasets of face images [53, 56, 118]. Such experiments are usually synthetic in some sense, and so we leave their discussion to §V-B. The datasets are generated based on the observation that images of the same face under changing illumination approximately lie on a linear subspace [8].

B. Experiments with RSR on Synthetic and Stylized Datasets

With the lack of real experimentation pointed out in the last section, we will resort to looking at synthetic experiments and stylized applications in this section. We first discuss previous experiments on synthetic data in §V-B1. We will then include a baseline simulation with the haystack model in §V-B2. Next, we supplement this with a stylized application of face subspace recovery in §V-B3.

1) Review of Experiments on Synthetic Data: In most of the works we have reviewed, experiments were run on synthetic data to show the usefulness of the developed methods. It is hard to find a complete comparison of the various subspace recovery methods, and, to our knowledge, Lerman and Maunu [53] provide the most comprehensive comparison of RSR algorithms on Gaussian generative models.

The extensive experiments in [53] compare all of the various algorithms on synthetic data drawn from the haystack model of [56] in various regimes. For most cases, the authors found FMS and TME to be the most robust to high percentages of outliers in these models. However, TME had a higher runtime, which matches the larger computational complexity $O(T \cdot \max(ND^2, D^3))$ versus $O(T \cdot NDd)$ for FMS. Algorithms that were not sufficiently accurate were MKF [120], REAPER [56], $R1$-PCA [24], GMS [119], RMD [38], RPCA [14, 61, 62], HR-PCA and DHR-PCA [31, 112], LLD and MDR [74], and OP [111].

Another interesting experiment can be seen in [119], where the authors test robustness of various algorithms with respect to asymmetric outliers. Here, outliers are distributed i.i.d. from the uniform distribution on $[0, 1]^D$, and inliers follow a Gaussian distribution on a random subspace in $\mathbb{R}^D$. In this model, the outliers are highly asymmetric with respect to the underlying subspace.

Other works have used stylized applications to test RSR algorithms on datasets with some real characteristics. For example, the inliers in a common real data example are images of a single person's face with constant pose and varying illuminations. In this case, the face images are known to approximately lie on a 9-dimensional linear subspace [8]. The "Faces in a Crowd" experiment is one stylized example of identifying a face subspace in a dataset with outliers [53, 56, 118]. Here, the outliers are taken to be other natural images, and the goal is to recover the underlying face subspace. Such a dataset is obviously stylized, since it arises nowhere in practice

2) Haystack Model Simulation: While there is a great need for new statistical models of data, we believe that comparison of performance under the haystack model has value. The main deficiencies of the haystack model (and to a certain extent, the generalized haystack model), are: 1) when not normalized to the sphere some simple statistics may distinguish inliers from outliers, 2) recovery under the haystack model can be easy for some algorithms. Although this may raise concerns, we believe that some algorithms successful on the haystack model will succeed on many other models and settings. To this end, we also include additional tests in §V-B3.

Our summary experiment on the haystack model is given in Figure 4. We run an analogous experiment to that in Lerman and Maunu [53], which includes as many algorithms as

possible. Here, we fix the parameters $N = 400$, $D = 200$, and $d = 10$, and we generate inliers i.i.d. $N(\mathbf{0}, \boldsymbol{P}_{L_*}/d)$ and outliers i.i.d. $N(\mathbf{0}, \boldsymbol{I}/D)$. We perturb all points by additional noise distributed i.i.d. $N(\mathbf{0}, 10^{-4}\boldsymbol{I})$. We generate 20 datasets at each fixed outlier percentages $5\%, 10\%, \ldots, 95\%$, resulting in 400 errors and times for each algorithm. These are summarized in box plots, whose x-values are the log-errors and y-values are the log-mean times for each algorithm. The edges of the boxes represent the 25th and 75th percentiles of the log-errors, and the red line represents the median log-error. The extreme ends of the whiskers represent a 99.3% coverage interval under the assumption that the log-errors are Gaussian. The red points are errors that lie outside of this interval. The further down and left an algorithm is, the better it performs.

Acronyms or names for the algorithms are as follows: TME (Tyler's M-estimator [118]), (S)FMS ((Spherized) Fast Median Subspace [53]), (S)GGD ((Spherized) Geodesic Gradient Descent [73]), REAPER [56], GMS (Geometric Median Subspace [119]), GMSO (Geometric Median Subspace with $1.5D$ added spherically symmetric Gaussian outliers [119]), OP (Outlier Pursuit [74, 111]), MDR (Maximum Mean Absolute Deviation Rounding [74]), DHRPCA (Deterministic High-Dimensional Robust PCA [31]), R1PCA (Rotational Invariant L_1-norm PCA [24]), (S)PCA ((Spherized) Principal Component Analysis), MKF (Median K-Flats [120]), SRO (Self-Representation Outlier Detection [116]), RPCA (Robust PCA, for which principal component pursuit was used) [60], RMD (Robust Online Mirror-Descent PCA [38]), ACOS (Adaptive Compressive Sampling [58]), TORP (Thresholding Based Outlier-Robust PCA [17]), CP (Coherence Pursuit [90]), and RANSAC [2, 33]. All algorithms are run with default parameters using code produced by the authors when available. For RF, we choose the determinant threshold δ to be 10^{-3}. OP uses $\lambda = 0.8\sqrt{D/N}$, as was used in [119], and we found this choice to perform much better than the recommended $3/(7\sqrt{N_{\text{out}}})$. ACOS uses this same λ when it calls OP and also uses a subsampling rate of 1/5. For TORP and DHRPCA, we set the percentage of outliers to be $\alpha = 0.5$ because there is no easy procedure to estimate this parameter in general. MKF passes over the data ten times and RMD passes over the data twice. For RANSAC, we use the RSR variant described in §II-I. We run 500 iterations and return the subspace with the best consensus number out of these iterations. We set the consensus threshold to be 10^{-3}. For CP, we implemented Algorithm 2 in [90] with recommended parameters. This procedure, which is advocated by the authors of [90] for dealing with noise, was not implemented in their code, and direct implementation of their original code for noiseless RSR was not satisfying. We set the threshold parameter to be the standard deviation of the noise, which is unknown to the user, and thus running CP in this way is somewhat unrealistic. We also set the projection dimension to be $2 \cdot d$, and the algorithm is run 5 times. PCA is used to find the underlying subspace on the set of all inliers identified in the 5 runs put together.

In these tests, we do not compare with DPCP [103], since the code provided online is really just an iterative application of a slower version of the FMS algorithm, and generally DPCP is meant for the setting of large d.

As we can see from this plot, the most accurate algorithms are FMS, SFMS, TME, SGGD, and SRO. Out of these algorithms, FMS is the fastest. TORP also performs well on this data when the correct percentage of outliers is used, but we cannot assume that this is known in practice. Even so, TORP is not as accurate and fast as FMS. DHRPCA does not perform well even if the true percentage of outliers is used. CP, despite having higher complexity than many other algorithms, is faster due to the fact that it is non-iterative (although it will not scale as well to large datasets).

3) The Blurryface Model and Simulations: We propose the *blurryface model* for statistically generating data in a stylized application of recognizing the most significant face in a dataset with many face images. We simulate such data and test all implemented RSR algorithms. The motivation here is to generate data with statistics that mimic real data. As was mentioned earlier, images of a single person's face under varying illumination and constant pose approximately lie on a subspace of dimension 9 [8]. This experiment tries to recover this 9-dimensional subspace in a dataset with outliers.

We take images from the Extended Yale B face database [52] and center each subject's subset. Since there are only 64 images per person in the database, we develop the following procedure for generating low-dimensional inlier faces. First, we take all centered images of the first subject's face and calculate the sample covariance, $\hat{\boldsymbol{\Sigma}}$, along with its eigenvalue decomposition. We keep only the top 9 eigenvectors and eigenvalues and store them in $\boldsymbol{U}_*$ and $\boldsymbol{S}_*$, respectively, so that $\hat{\boldsymbol{\Sigma}} \approx \boldsymbol{U}_* \boldsymbol{S}_* \boldsymbol{U}_*^T$. Synthetic inlier faces are generated i.i.d. $N(\mathbf{0}, c_1 \boldsymbol{U}_* \boldsymbol{U}_*^T/d)$ or i.i.d. $N(\mathbf{0}, c_2 \boldsymbol{U}_* \boldsymbol{S}_* \boldsymbol{U}_*^T)$. The first model is the *spherically symmetric* inlier model and the second model is the *elliptical* inlier model. The constant c_1 is the average squared norm of all centered faces in the database and $c_2 = c_1/(\text{Tr}(\boldsymbol{S}_*))$. These constants are designed to give inliers comparable magnitude to the original centered faces. In both experiments, outliers are sampled without replacement from the other faces in the database. In both experiments, we also perturb all points by small Gaussian noise sampled i.i.d. $N(\mathbf{0}, 10^{-4}\boldsymbol{I}/D)$ (which gives rise to "blurry" faces).

We note that the distribution of the inlier faces is quite different than the natural face images, which lie in a cone. This experiment is just meant to approximate this distribution to some reasonable degree and allow for easy generation of samples. Outliers all lie in a cone and are asymmetric, which makes subspace recovery more challenging than in the haystack model.

Figure 5 illustrates comparisons between the implemented RSR algorithms using data generated from the two instances of the blurryface model. We generate 20 datasets at each fixed outlier percentage $(5\%, 10\%, \ldots, 95\%)$ resulting in 400 errors and times for each algorithm and inlier model (spherically symmetric and elliptical). These are summarized in box plots, whose x-values are the log-errors and y-values are the log-mean times for each algorithm. An explanation of the boxes, whiskers, and points were given in §V-B2. The results for the spherically symmetric inlier model are displayed on the left in Figure 5, and the results for the elliptical inlier model are given on the right. Algorithm settings are as before. We modify
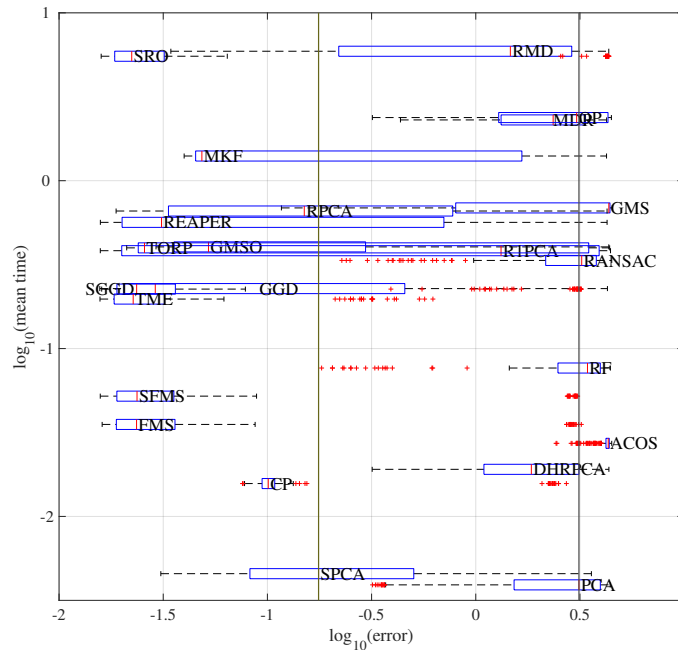
Fig. 4: Accuracy-time comparison for various RSR algorithms under the haystack model. Here, we generate inliers i.i.d. $N(\mathbf{0}, \boldsymbol{P}_{L_*}/d)$ and outliers $N(\mathbf{0}, \boldsymbol{I}/D)$, with $N = 400$, $D = 200$, and $d = 10$. We also generate added noise distributed i.i.d. $N(\mathbf{0}, 10^{-4}\boldsymbol{I}/D)$. Twenty datasets are generated at each percentage of outliers 5%, 10%, ..., 95%, and the RSR algorithms are run to calculate a robust subspace. Error is calculated as distance to ground truth (square root of sum of squared principal angles), and the runtime is recorded. The y-value of the boxplot represents the log-mean runtime and the x-value represents the log-error.

the thresholds for RF and RANSAC to now be $10^{-3}c$, where c is the mean norm of data points in the given dataset. This threshold is designed to be almost on the order of the smallest deviation of inlier faces from the underlying subspace.

For the spherically symmetric face model, TORP and TME appear to give the best accuracy. The next most accurate algorithms are FMS/SFMS, GGD/SGGD, GMSO, and REAPER. Out of all of these accurate algorithms, the fastest is FMS/SFMS. For the elliptical face model, the best accuracy is given by TME, followed by FMS/SFMS, SGGD, and REAPER. Out of these algorithms, FMS and SFMS are the fastest, while TME is much slower. Again, CP is fast due to its non-iterative nature and the small size of the dataset.

We note that, while GMSO has some success with the spherically symmetric inlier models, it struggles with the elliptical inliers. TORP performs very well on the spherically symmetric blurryface data with the correct percentage of outliers, but again we cannot assume this is known in practice. Even with the correct percentage, TORP fails on the elliptical blurryface model. Again, DHRPCA does not perform well on either example even if the true percentage of outliers is used.

Following the discussion on minimization L_1-PCA in §II-E, one may suggest testing RSR on models with heavy tailed elementwise noise. For example, one may try Laplacian noise as mentioned in [7]. However, this is not the RSR problem formulated in this survey. We briefly tested such settings and noticed that most RSR algorithms are comparable on it, while RPCA [60] performs somewhat better. This result is not surprising as the RPCA problem was discussed as a loose

relaxation of the version of L1-PCA in (19).

VI. INFLUENCE OF RSR ON OTHER METHODS

The study of RSR can influence the development of theory and algorithms for other challenging problems. In this section we discuss the important relationship of RSR with other mathematical problems.

One problem mentioned earlier is robust subspace clustering. A mathematical formulation of this problem assumes inliers sampled from (or around) a union of subspaces and outliers sampled from a different, though somewhat restricted, distribution. The goal is to identify the different underlying subspaces. This problem was addressed in [4, 16, 54, 55, 94, 116, 121] by initially applying methods for filtering outliers that are the same as those in RSR. Indeed, the idea of identifying outliers by affinities that express presence in an underlying subspace (see §II-H) works equally well for multiple underlying subspaces.

RSR methods may improve robust subspace clustering in addition to filtering outliers. A naive approach for solving robust subspace clustering is to sequentially fit a robust subspace or a single robust direction or sequentially remove a robust direction. Some previous works have already applied such sequential RSR strategies to solve this problem. For example, one of the methods in [114] sequentially fits a single subspace by using RANSAC. Furthermore, the method of [89] can be explained as sequentially searching for a least significant orthogonal direction $\boldsymbol{b}$ that aims to minimize (11). Equivalently, it can be described by sequentially searching
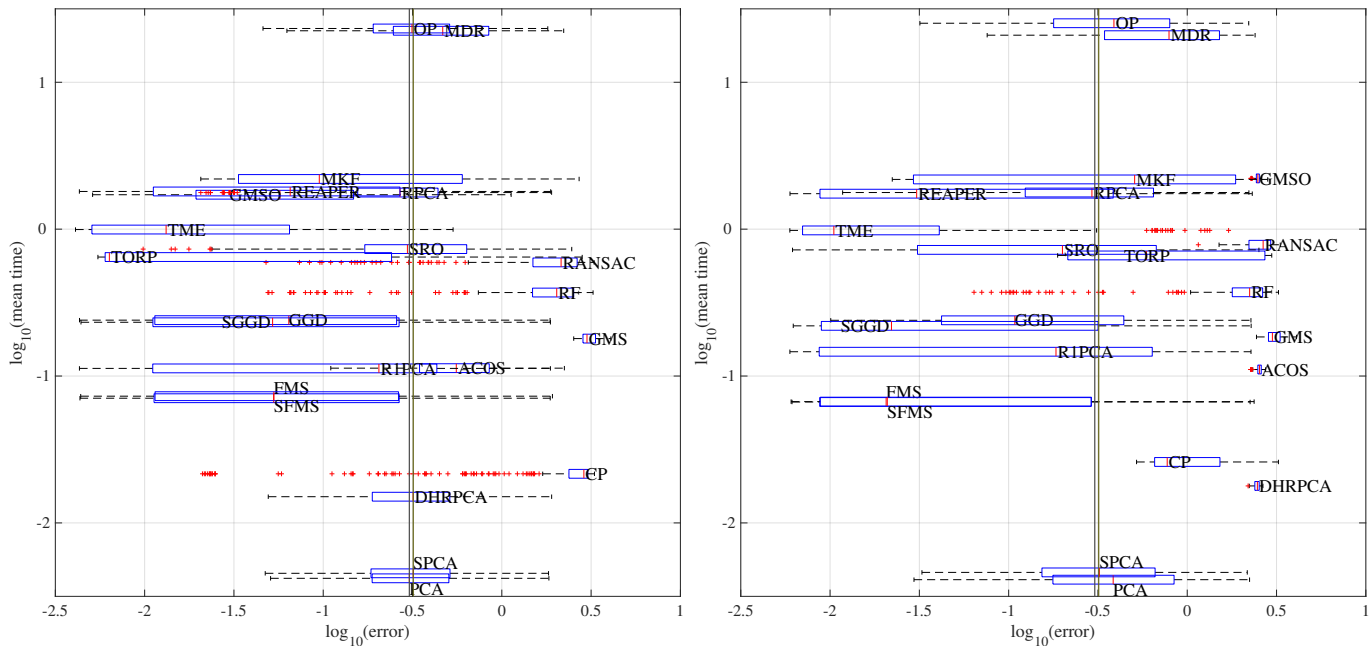
Fig. 5: Accuracy-time comparison for various RSR algorithms on the blurryface data. Here, we generate inliers on the 9-dimensional face subspace estimated from the data, and outliers are random images of other faces. On the left, we generate inliers i.i.d. $N(\mathbf{0}, c_1 \boldsymbol{U}_* \boldsymbol{U}_*^T /9)$, and on the right we generate inliers i.i.d. $N(\mathbf{0}, c_2 \boldsymbol{U}_* \boldsymbol{S}_* \boldsymbol{U}_*^T)$. We fix $N = 500$ and $D = 400$, and we also generate added noise distributed i.i.d. $N(\mathbf{0}, 10^{-4} \boldsymbol{I}/D)$. Twenty datasets are generated at each percentage of outliers 5%, 10%, ..., 95%, and the RSR algorithms are run to calculate a robust subspace. Error is calculated as distance to ground truth (square root of sum of squared principal angles). The y-value of the boxplot represents the log-mean runtime and the x-value represents the log-error.

for a least absolute deviation hyperplane, minimizing (9) (see clarification right after (11)). Tsakiris and Vidal [103] sequentially use the same RSR formulation in (11) to solve the problem of hyperplane clustering, that is, when the dimensions of all underlying subspaces are $D - 1$. However, there are many geometric obstacles to any sequential RSR approach for general robust subspace clustering, unless one assumes a very restrictive setting.

Another possibility is to use RSR within a K-subspace algorithm, which generalizes K-means to subspaces (see, e.g., [54, 121]). However, theoretical guarantees are not developed yet for such an algorithmic approach (Lerman and Zhang [54] provide guarantees for the oracle minimization of such an approach, but not for an algorithm minimizing it). Furthermore, this method requires knowledge of the intrinsic dimensions of the subspaces, unlike [29]. Application of the K-subspace strategy with RPCA instead of RSR was suggested by [108] to address some problems in image denoising (with nonstandard noise) and blind inpainting. In these problems, the subspaces are used as approximate models and there is some flexibility in choosing the dimensions of the subspaces. Another possible application of RSR to robust subspace clustering is the use of robust energies in the framework proposed in [122]. Furthermore, the local best-fit flats in this strategy can be the output of an RSR algorithm.

RSR might be extended to the more general problem of robust manifold clustering, where the inliers are sampled from (or around) a union of manifolds and the goal is to recover the underlying manifolds [4]. Indeed, this might be possible by restricting RSR methods to local neighborhoods. Similarly, such a strategy can apply to the problem of robust recovery of a single manifold.

Developments within RSR can be beneficial for other kinds of modeling problems. For example, convex algorithms for RSR that rely on IRLS procedures [56, 119] inspired the development of methods for two problems in computer vision: robust recovery of camera locations from corrupted pairwise directions [84] and robust synchronization [107], that is, robustly estimating unknown rotations (in particular, camera orientations) from a set of corrupted pairwise rotations [107]. Furthermore, the proof of the main theorem in [107] (Theorem 4.1) borrows and adapts ideas from [56, 119]. Similarly, ideas of filtering outliers, which are weakly reminiscent of [16], were used in [92] to enhance solutions of the camera location problem. However, the latter problem is more challenging. In this setting, the outliers are associated with pairwise directions between points and not the points themselves. Moreover, any 3 uncorrupted pairwise directions lie on a two-dimensional subspace, but the subspaces defined in this way have no direct relationships. Thus, the set of uncorrupted pairwise directions does not have a simple geometric model, such as a subspace. Other methods of filtering outlier pairwise directions have to be developed to take advantage of the more complicated geometry here.

It is likely that the recent theoretical work of [73] can be generalized to other NP-hard recovery problems that can

be formulated via optimization over continuous, nonconvex sets. One immediate candidate is robust synchronization over the special orthogonal group, and we have already mentioned the influence of RSR methodology there [107]. In these optimization problems, one may possibly extend the deterministic conditions in [73] by considering appropriate notions of permeance and alignment in the new setting. One may further guarantee recovery by a gradient descent algorithm under these extended conditions.

Another problem that requires RSR is that of finding the sparsest vector within a subspace. Indeed, Qu et al. [88] formulated this problem by using (11) with Xb instead of $X^T b$ (note the dimension of b changes as well). They pointed out its connection with sparse dictionary learning and sparse PCA.

Robust fundamental or essential matrix estimation in computer vision can also be cast as an RSR problem. One way to calculate such matrices is to use PCA on a set of data points, although in these settings there are frequently many outliers. Further, inlier and outlier points tend to exhibit very asymmetric distributions. Specially tailored variants of RSR methods may be able to outperform existing methods for robust fundamental matrix estimation [91, 101, 102].

VII. FUTURE WORK

One option for future work is to better understand large-sample and high-dimensional limits for RSR. Some online algorithms have been proposed for robust subspace estimation [38, 121], but their theoretical guarantees are not satisfying and their performance is disappointing. It is also not known how many samples are needed for these methods to converge. Huroyan and Lerman [45] have considered distributed models for RSR. Here, the authors assume that a dataset is distributed across many nodes and communication is limited by the network structure. They show under certain conditions that it is still possible to optimize some previously proposed problem formulations [53, 56, 119] in this setting.

Affine subspace estimation is not well studied, but a potential important extension of current work. One can consider estimation on the affine Grassmannian [59], although the estimation considered in [59] may not be tight enough. A simple idea can be seen in the IRLS procedures of GMS, REAPER, and FMS [53, 56, 119], which have a trivial extension to affine subspaces. However, we have not seen a practical advantage to including this extension for real data, and it remains an open question to see if considering affine subspaces can add real value over centering data by the geometric median. Another possibility comes from considering Dümbgen's M-estimator [27], spatial Kendall's tau [106], or any symmetrized version of a robust covariance estimator [78]. For example, both Dümbgen's M-estimator and spatial Kendall's tau have been considered for independent component analysis by Oja et al. [81]. As mentioned earlier, it is not immediately obvious how to estimate the offset for the affine subspace with these estimators, though.

Another potential realm that is not well understood is estimation of the subspace dimension for the RSR problem. Some

work has gone into dimension estimation for PCA [25, 50], but there are no analogous works for RSR. And, indeed, the fastest RSR algorithms require knowledge of the subspace dimension d a priori. One potential issue of direct application of these methods is that, unlike PCA, RSR methods do not give nested subspaces. This makes it harder to compare subspaces across dimensions and makes heuristic strategies, such as the elbow method, hard to motivate and costly to compute. One must also determine a good metric to compare across dimensions, for which there is no easy or obvious choice. Thus, the development of methods for this problem would be an interesting direction for future work.

Robustness to noise in the PCA problem is also a relatively unstudied problem. One possible path is to pursue ideas similar to those in [20]. The work of [17] also has a nontrivial result to noise, where they achieve similar rates as PCA to sub-Gaussian noise, even in the presence of outliers. However, [17] requires knowledge of the fraction of outliers, which makes the setting of the robustness to outliers easier. But, perhaps the future of noise analysis lies in looking at heavy-tailed distributions and limits in the various models of inliers and outliers. One intriguing idea for heavy-tailed noise is given by [75], where the author uses the idea of median of means to construct a robust covariance estimator. This estimator can then be used to find a PCA subspace that exhibits asymptotic sub-Gaussian estimation bounds, even in the presence of heavier tailed data. The determination of optimal rates and consideration of other noise regimes remain open problems.

One question is where the recovery theory of RSR should go next. Indeed, theoretical guarantees of recovery under special models are not the primary goal of RSR. Instead, we wish to have methods that are useful in practice. Recent work on robustness has considered how well an algorithm can perform in the presence of adversarial corruption [23, 99]. Although adversarial outliers have been considered in the context of some RSR algorithms, such as [111], current results are weak, and it seems that better algorithms and guarantees can be developed for these cases [72].

So far, analysis of the inliers and outliers in RSR has been separated. This has led to the separate notions of permeance and restricted alignment we discussed in §III-A, which are each formulated with respect to the inliers or outliers alone. These independent formulations are then combined to form a stability constraint on the model. The work of Hardt and Moitra [41] does not separate between two conditions of inliers and outliers, but it has a very simplistic and well-defined setting. It would be interesting to find out if there are more refined stability conditions that involve both inliers and outliers together. In contrast to the work of Hardt and Moitra [41], these conditions would need to be more general and allow both inliers and outliers to lie on lower dimensional subspaces to capture a wider range of examples.

In terms of RSR, one can raise the question of whether or not the study of high-dimensions is really needed. Using ideas from the Johnson-Lindenstrauss lemma [46], one may think that a few random projections will maintain most of the important statistics of the data, including the low-dimensional subspace structure. However, short simulations have shown

that while this can work for low percentages of outliers, it becomes harder in low SNR regimes. This should not be surprising, because the low SNR regimes experience the hardness threshold of $d/(D-d)$ [41]. Indeed, the SNR threshold increases as D decreases. Quantifying how well random projections work for RSR is an interesting avenue for future work.

Finally, as was mentioned in §V-A, there is a need for more experimentation on RSR methods. One thing we advocate is the development of a database of examples to test RSR methods on. Datasets within this database must also have properly defined measures of success that are tied to the specific application. A possible start to this database could involve wider experimentation on robust dimensionality reduction for a variety of tasks. Another option is to develop more stylized applications to test RSR methods on datasets that mirror real data in some way.

A supplemental webpage with code and data will be provided at https://twmaunu.github.io/rsr_overview/.

VIII. Acknowledgements

References

[1] L. P. Ammann. Robust singular value decompositions: A new approach to projection pursuit. *Journal of the American Statistical Association*, 88(422):pp. 505–514, 1993. ISSN 01621459.

[2] E. Arias-Castro and J. Wang. RANSAC Algorithms for Subspace Recovery and Subspace Clustering. *ArXiv e-prints*, November 2017.

[3] E. Arias-Castro, D. L. Donoho, X. Huo, and C. A. Tovey. Connect the dots: how many random points can a regular curve pass through? *Adv. in Appl. Probab.*, 37 (3):571–603, 2005.

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

[5] O. Arslan. Convergence behavior of an iterative reweighting algorithm to compute multivariate M-estimates for location and scatter. *Journal of Statistical Planning and Inference*, 118(1-2):115 – 128, 2004. ISSN 0378-3758. doi: 10.1016/S0378-3758(02)00402-0.

[6] C. Auderset, C. Mazza, and E. A. Ruh. Angular Gaussian and Cauchy estimation. *Journal of multivariate analysis*, 93(1):180–197, 2005.

[7] A. Baccini, Ph. Besse, and A. de Falguerolles. A L_1-norm PCA and a heuristic approach. In E. Diday, Y. Lechevalier, and O. Opitz, editors, *Ordinal and symbolic data analysis*, pages 359–368, New York, 1996. Springer.

[8] R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE transactions on pattern analysis and machine intelligence*, 25(2):218–233, 2003.

[9] A. Blum, J. Hopcroft, and R. Kannan. Foundations of data science. Available online at https://www.cs.cornell.edu/jeh/book.pdf, 2018.

[10] J. P. Brooks, J. H. Dulá, and E. L. Boone. A pure L1-norm principal component analysis. *Computational statistics & data analysis*, 61:83–98, 2013.

[11] T. Budavri, V. Wild, A. S. Szalay, L. Dobos, and C.-W. Yip. Reliable eigenspectra for new generation surveys. *Monthly Notices of the Royal Astronomical Society*, 394 (3):1496–1502, 2009.

[12] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009.

[13] E. J. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

[14] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[15] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

[16] G. Chen and G. Lerman. Spectral curvature clustering (SCC). *Int. J. Comput. Vision*, 81(3):317–330, 2009.

[17] Y. Cherapanamjeri, P. Jain, and P. Netrapalli. Thresholding based outlier robust PCA. In *COLT*, pages 593–628, 2017.

[18] V. Choulakian. L_1-norm projection pursuit principal component analysis. *Computational Statistics & Data Analysis*, 50(6):1441–1451, March 2006.

[19] K. L. Clarkson and D. P. Woodruff. Input sparsity and hardness for robust subspace approximation. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 310–329. IEEE, 2015.

[20] G. Coudron and G. Lerman. On the sample complexity of robust PCA. In *Advances in Neural Information Processing Systems*, pages 3221–3229, 2012.

[21] C. Davis and W. M. Kahan. The rotation of eigenvectors by a perturbation. iii. *SIAM J. on Numerical Analysis*, 7:1–46, 1970.

[22] Fernando De La Torre and M. J Black. A framework for robust subspace learning. *International Journal of Computer Vision*, 54(1-3):117–142, 2003.

[23] I. Diakonikolas, G. Kamath, D. M. Kane, J. Li, A. Moitra, and Alistair Stewart. Being robust (in high dimensions) can be practical. In *International Conference on Machine Learning*, pages 999–1008, 2017.

[24] C. Ding, D. Zhou, X. He, and H. Zha. R1-PCA: rotational invariant L_1-norm principal component anal-

ysis for robust subspace factorization. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 281–288, New York, NY, USA, 2006. ACM.

[25] E. Dobriban. Factor selection by permutation. *arXiv preprint arXiv:1710.00479*, 2017.

[26] D. L. Donoho and M. Gasko. Breakdown properties of location estimates based on halfspace depth and projected outlyingness. *Ann. Statist.*, 20(4):1803–1827, 1992. ISSN 0090-5364. URL https://doi-org.ezp3.lib.umn.edu/10.1214/aos/1176348890.

[27] L. Dümbgen. On Tyler's M-functional of scatter in high dimension. *Annals of the Institute of Statistical Mathematics*, 50(3):471–491, 1998.

[28] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20(2):303–353 (electronic), 1999. ISSN 0895-4798.

[29] E. Elhamifar and R. Vidal. Robust classification using structured sparse representation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1873–1879. IEEE, 2011.

[30] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99): 1–1, 2013. ISSN 0162-8828. doi: 10.1109/TPAMI.2013.57.

[31] J. Feng, H. Xu, and S. Yan. Robust PCA in high-dimension: A deterministic approach. *International conference on machine learning (ICML)*, 2012.

[32] J. Feng, H. Xu, and S. Yan. Online robust PCA via stochastic optimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[33] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[34] J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on computers*, 100(9):881–890, 1974.

[35] N. Gillis and S. A. Vavasis. On the complexity of robust PCA and ℓ_1-norm low-rank matrix approximation. *arXiv preprint arXiv:1509.09236*, 2015.

[36] J. Glaz and N. Balakrishnan. *Scan statistics and applications*. Springer Science & Business Media, 1999.

[37] J. Glaz, J. Naus, and S. Wallenstein. *Scan Statistics*. Springer Series in Statistics. Springer-Verlag, New York, 2001. ISBN 0-387-98819-X.

[38] J. Goes, T. Zhang, R. Arora, and G. Lerman. Robust stochastic principal component analysis. In *Artificial Intelligence and Statistics*, pages 266–274, 2014.

[39] J. Goes, G. Lerman, and B. Nadler. Robust sparse covariance estimation by thresholding Tyler's M-estimator. *arXiv preprint arXiv:1706.08020*, 2017.

[40] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, 3rd edition, 1996. ISBN 0-8018-5413-X; 0-8018-5414-8.

[41] M. Hardt and A. Moitra. Algorithms and hardness for robust subspace recovery. In *Conference on Learning Theory (COLT)*, pages 354–375, 2013.

[42] J. He, L. Balzano, and J. C. S. Lui. Online robust subspace tracking from partial information. *CoRR*, abs/1109.3827, 2011.

[43] J. He, L. Balzano, and A. Szlam. Incremental gradient on the grassmannian for online foreground and background separation in subsampled video. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1568–1575. IEEE, 2012.

[44] P. J. Huber and E. M. Ronchetti. *Robust Statistics*. Wiley Series in Probability and Statistics. Wiley, Hoboken, NJ, 2nd edition, 2009. ISBN 978-0-470-12990-6. doi: 10.1002/9780470434697.

[45] V. Huroyan and G. Lerman. Distributed robust subspace recovery. *arXiv preprint arXiv:1705.09382*, 2017.

[46] W. B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

[47] R. Kannan and S. Vempala. Spectral algorithms. *Foundations and Trends in Theoretical Computer Science*, 4 (3-4):157–288, 2009. doi: 10.1561/0400000025. URL https://doi.org/10.1561/0400000025.

[48] Q. Ke and T. Kanade. Robust L_1 norm factorization in the presence of outliers and missing data by alternative convex programming. In *Conference on Computer Vision and Pattern Recognition*, volume 1, pages 739–746. IEEE, 2005.

[49] J. T. Kent and D. E. Tyler. Maximum likelihood estimation for the wrapped Cauchy distribution. *Journal of Applied Statistics*, 15(2):247–254, 1988.

[50] S. Kritchman and B. Nadler. Determining the number of components in a factor model from limited noisy data. *Chemometrics and Intelligent Laboratory Systems*, 94 (1):19–32, 2008.

[51] N. Kwak. Principal component analysis based on L_1-norm maximization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(9):1672–1680, 2008. doi: 10.1109/TPAMI.2008.114.

[52] K.-C. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(5):684–698, 2005.

[53] G. Lerman and T. Maunu. Fast, robust and non-convex subspace recovery. *Accepted for publication, Information and Inference*, 2017.

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

[55] G. Lerman and T. Zhang. l_p-recovery of the most significant subspace among multiple subspaces with outliers. *Constructive Approximation*, 40(3):329–385, 2014.

[56] 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.

[57] G. Li and Z. Chen. Projection-pursuit approach to robust dispersion matrices and principal components: Primary theory and Monte Carlo. *Journal of the American Statistical Association*, 80(391):759–766, 1985. ISSN 01621459. doi: 10.2307/2288497.

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

[59] L. Lim, K. S. Wong, and K. Ye. Statistical estimation and the affine Grassmannian. *arXiv preprint arXiv:1607.01833*, 2016.

[60] Z. Lin, A. Ganesh, J., L. Wu, M. Chen, and Y. Ma. Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. In *In Intl. Workshop on Comp. Adv. in Multi-Sensor Adapt. Processing, Aruba, Dutch Antilles*, 2009.

[61] Z. Lin, M. Chen, and Y. Ma. The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices. *ArXiv e-prints*, September 2010.

[62] Z. Lin, R. Liu, and Z. Su. Linearized alternating direction method with adaptive penalty for low-rank representation. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 612–620. Curran Associates, Inc., 2011.

[63] N. Locantore, J. S. Marron, D. G. Simpson, N. Tripoli, J. T. Zhang, and K. L. Cohen. Robust principal component analysis for functional data. *Test*, 8(1):1–73, 1999.

[64] H. P. Lopuhaä and P. J. Rousseeuw. Breakdown points of affine equivariant estimators of multivariate location and covariance matrices. *Ann. Statist.*, 19(1):229–248, 1991.

[65] H. P. Lopuhaa and P. J. Rousseeuw. Breakdown points of affine equivariant estimators of multivariate location and covariance matrices. *The Annals of Statistics*, pages 229–248, 1991.

[66] J. I. Marden. Some robust estimates of principal components. *Statistics & probability letters*, 43(4):349–359, 1999.

[67] P. P. Markopoulos, G. N. Karystinos, and D. A. Pados. Optimal algorithms for L_1-subspace signal processing. *IEEE Transactions on Signal Processing*, 62(19):5046–5058, 2014.

[68] P. P. Markopoulos, S. Kundu, S. Chamadia, N. Tsagkarakis, and D. A. Pados. Outlier-resistant data processing with L1-norm principal component analysis. In *Advances in Principal Component Analysis*, pages 121–135. Springer, 2018.

[69] R. A. Maronna. Robust estimation of multivariate location and scatter. *Wiley StatsRef: Statistics Reference Online*, 1976.

[70] R. A. Maronna. Principal components and orthogonal regression based on robust scales. *Technometrics*, 47 (3):264–273, 2005.

[71] R. A. Maronna, R. D. Martin, and V. J. Yohai. *Robust statistics: Theory and methods*. Wiley Series in Probability and Statistics. John Wiley & Sons Ltd., Chichester, 2006. ISBN 978-0-470-01092-1; 0-470-01092-4.

[72] T. Maunu and G. Lerman. Robust subspace recovery with adverserial outliers. In preparation.

[73] T. Maunu, T. Zhang, and G. Lerman. A well-tempered landscape for non-convex robust subspace recovery. *arXiv preprint arXiv:1706.03896*, 2017.

[74] M. McCoy and J. A Tropp. Two proposals for robust PCA using semidefinite programming. *Electronic Journal of Statistics*, 5:1123–1160, 2011.

[75] S. Minsker. Geometric median and robust estimation in banach spaces. *Bernoulli*, 21(4):2308–2335, 2015.

[76] A. Naor, O. Regev, and T. Vidick. Efficient rounding for the noncommutative Grothendieck inequality. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 71–80. ACM, 2013.

[77] P. Netrapalli, U. N. Niranjan, S. Sanghavi, A. Anandkumar, and P. Jain. Non-convex robust PCA. In *Advances in Neural Information Processing Systems*, pages 1107–1115, 2014.

[78] K. Nordhausen and D. E. Tyler. A cautionary note on robust covariance plug-in methods. *Biometrika*, 102(3): 573–588, 2015.

[79] J. Novembre, T. Johnson, K. Bryc, Z. Kutalik, A. R. Boyko, A. Auton, A. Indap, K. S. King, S. Bergmann, M. R. Nelson, M. Stephens, and C. D. Bustamante. Genes mirror geography within europe. *Nature*, 456(7218):98–101, November 2008. doi: 10.1038/nature07331.

[80] H. Nyquist. Least orthogonal absolute deviations. *Computational Statistics & Data Analysis*, 6(4):361 – 367, 1988.

[81] H. Oja, S. Sirkiä, and J. Eriksson. Scatter matrices and independent component analysis. *Austrian Journal of Statistics*, 35(2&3):175–189, 2006.

[82] E. Ollila and D. E. Tyler. Regularized M-estimators of scatter matrix. *IEEE Trans. Signal Process.*, 62(22): 6059–6070, 2014. ISSN 1053-587X. doi: 10.1109/TSP. 2014.2360826. URL http://dx.doi.org/10.1109/TSP. 2014.2360826.

[83] M. R. Osborne and G. A. Watson. An analysis of the total approximation problem in separable norms, and an algorithm for the total l_1 problem. *SIAM Journal on Scientific and Statistical Computing*, 6(2):410–424, 1985.

[84] Onur Özyeşil and Amit Singer. Robust camera location estimation by convex programming. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2674–2683, 2015.

[85] D. Pimentel-Alarcón and R. Nowak. Random consensus robust pca. *Electronic Journal of Statistics*, 11(2):5232–5253, 2017.

[86] A. L. Price, N. J. Patterson, R. M. Plenge, M. E. Weinblatt, N. A. Shadick, and D. Reich. Principal components analysis corrects for stratification in genome-wide association studies. *Nature Genetics*, 38(8):904–909, 2006. URL http://www.ncbi.nlm.nih.gov/pubmed/16862161.

[87] C. Qiu and N. Vaswani. Real-time robust principal components' pursuit. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pages 591–598. IEEE, 2010.

[88] Q. Qu, J. Sun, and J. Wright. Finding a sparse vector in a subspace: Linear sparsity using alternating directions. *IEEE Trans. Information Theory*, 62(10):5855–5880, 2016. URL https://doi.org/10.1109/TIT.2016.2601599.

[89] M. Rahmani and G. Atia. Innovation pursuit: A new approach to the subspace clustering problem. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2874–2882, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/rahmani17b.html.

[90] M. Rahmani and G. K. Atia. Coherence pursuit: Fast, simple, and robust principal component analysis. *IEEE Transactions on Signal Processing*, 65(23):6260–6275, 2017.

[91] S. Sengupta, T. Amir, M. Galun, T. Goldstein, D. W. Jacobs, A. Singer, and R. Basri. A new rank constraint on multi-view fundamental matrices, and its application to camera location recovery. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, Hawaii, USA, June 22-25, 2017*, pages 4798–4806, 2017.

[92] Y. Shi and G. Lerman. Estimation of camera locations in highly corrupted scenarios: All about that base, no shape trouble. 2018. To appear in Proceedings of CVPR 18.

[93] M. Soltanolkotabi and E. J. Candès. A geometric analysis of subspace clustering with outliers. *Ann. Stat.*, 40(4):2195–2238, 2012. doi: 10.1214/12-AOS1034.

[94] M. Soltanolkotabi and E. J. Candès. A geometric analysis of subspace clustering with outliers. *Ann. Stat.*, 40(4):2195–2238, 2012. doi: 10.1214/12-AOS1034.

[95] Z. Song, D. P. Woodruff, and P. Zhong. Low rank approximation with entrywise l 1-norm error. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 688–701. ACM, 2017.

[96] H. Späth and G. A. Watson. On orthogonal linear approximation. *Numer. Math.*, 51:531–543, October 1987.

[97] N. Srebro and T. Jaakkola. Weighted low-rank approximations. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 720–727, 2003.

[98] W. A. Stahel. *Breakdown of covariance estimators*. Fachgruppe für Statistik, Eidgenössische Techn. Hochsch., 1981.

[99] J. Steinhardt, M. Charikar, and G. Valiant. Resilience: A criterion for learning in the presence of arbitrary outliers. In *ITCS*, 2018.

[100] Y. Sun, P. Babu, and D. P. Palomar. Regularized Tyler's scatter estimator: existence, uniqueness, and algorithms. *IEEE Trans. Signal Process.*, 62(19):5143–5156, 2014. ISSN 1053-587X. doi: 10.1109/TSP.2014.2348944.

[101] P. H. S. Torr and D. W. Murray. Outlier detection and motion segmentation. In *Sensor Fusion VI*, volume 2059, pages 432–444. International Society for Optics and Photonics, 1993.

[102] P. H. S. Torr and D. W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *International journal of computer vision*, 24(3):271–300, 1997.

[103] M. C. Tsakiris and R. Vidal. Dual principal component pursuit. In *ICCV Workshops 2015*, pages 850–858, 2015.

[104] D. E. Tyler. A distribution-free M-estimator of multivariate scatter. *Ann. Statist.*, 15(1):234–251, 1987. ISSN 0090-5364. doi: 10.1214/aos/1176350263.

[105] N. Vaswani and P. Narayanamurthy. Static and dynamic robust PCA via low-rank+sparse matrix decomposition: A review. *arXiv preprint arXiv:1803.00651*, 2018.

[106] S. Visuri, V. Koivunen, and H. Oja. Sign and rank covariance matrices. *Journal of Statistical Planning and Inference*, 91(2):557–575, 2000.

[107] L. Wang and A. Singer. Exact and stable recovery of rotations for robust synchronization. *Information and Inference*, 2013.

[108] Y. Wang, A. Szlam, and G. Lerman. Robust locally linear analysis with applications to image denoising and blind inpainting. *SIAM J. Imaging Sciences*, 6(1):526–562, 2013.

[109] G. A. Watson. *Some Problems in Orthogonal Distance and Non-Orthogonal Distance Regression*. Defense Technical Information Center, 2001. URL http://books.google.com/books?id=WKKWGwAACAAJ.

[110] A. Wiesel. Geodesic convexity and covariance estimation. *IEEE Trans. Signal Processing*, 60(12):6182–6189, 2012. doi: 10.1109/TSP.2012.2218241.

[111] H. Xu, C. Caramanis, and S. Sanghavi. Robust PCA via outlier pursuit. *IEEE Trans. Information Theory*, 58(5):3047–3064, 2012. doi: 10.1109/TIT.2011.2173156.

[112] H. Xu, C. Caramanis, and S. Mannor. Outlier-robust PCA: the high-dimensional case. *IEEE transactions on information theory*, 59(1):546–572, 2013.

[113] L. Xu and A. L. Yuille. Robust principal component analysis by self-organizing rules based on statistical physics approach. *Neural Networks, IEEE Transactions on*, 6(1):131–143, 1995. ISSN 1045-9227. doi: 10.1109/72.363442.

[114] A. Y. Yang, S. R. Rao, and Y. Ma. Robust statistical estimation and segmentation of multiple subspaces. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on*, page 99, june 2006. doi: 10.1109/CVPRW.2006.178.

[115] X. Yi, D. Park, Y. Chen, and C. Caramanis. Fast algorithms for robust pca via gradient descent. In *Advances in neural information processing systems*, pages 4152–4160, 2016.

[116] C. You, D. P. Robinson, and R. Vidal. Provable self-representation based outlier detection in a union of subspaces. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu,*

HI, USA, July 21-26, 2017, pages 4323–4332, 2017. doi: 10.1109/CVPR.2017.460.

[117] L. Yu, M. Zhang, and C. Ding. An efficient algorithm for $L1$-norm principal component analysis. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 1377–1380. IEEE, 2012.

[118] T. Zhang. Robust subspace recovery by Tyler's M-estimator. *Information and Inference*, 5(1):1–21, 2016.

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

[120] T. Zhang, A. Szlam, and G. Lerman. Median K-flats for hybrid linear modeling with many outliers. In *International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 234–241, Kyoto, Japan, 2009.

[121] T. Zhang, A. Szlam, and G. Lerman. Median K-flats for hybrid linear modeling with many outliers. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 234–241. IEEE, 2009.

[122] T. Zhang, A. Szlam, Y. Wang, and G. Lerman. Hybrid linear modeling via local best-fit flats. *International Journal of Computer Vision*, 100:217–240, 2012. ISSN 0920-5691. doi: 10.1007/s11263-012-0535-6.

[123] T. Zhang, A. Wiesel, and M. S. Greco. Multivariate generalized Gaussian distribution: Convexity and graphical models. *IEEE Trans. Signal Processing*, 61(16):4141–4148, 2013. doi: 10.1109/TSP.2013.2267740.

[124] Z. Zhou, X. Li, J. Wright, E. Candès, and Y. Ma. Stable principal component pursuit. In *International Symposium on Information Theory Proceedings (ISIT)*, pages 1518–1522. IEEE, 2010.

APPENDIX

A. Intuition for the Robust Covariance Matrices

To clarify the robust energies in (28) and (29), we express them as scaled versions of negative log-likelihood functions with respect to heavy-tailed elliptical distributions. We thus understand the Maronna and Tyler M-estimators as maximum likelihood estimators that are robust to heavy tails. We assume a centered elliptical distribution with density function f that is everywhere positive. That is, f has the form

$$f(\boldsymbol{x}; \boldsymbol{\Sigma}) = \frac{g(\boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x})}{\sqrt{\det(\boldsymbol{\Sigma})}}, \quad \text{where} \quad g : (0, \infty) \to (0, \infty). \tag{45}$$

If $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ are i.i.d. sampled from f, then the likelihood function has the form

$$L(\boldsymbol{\Sigma}|\mathcal{X}) = \frac{\prod_{i=1}^{N} g(\boldsymbol{x}_i^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x}_i)}{\det(\boldsymbol{\Sigma})^{\frac{N}{2}}}. \tag{46}$$

Setting $\rho(t) = -2\log(g(t))$, the negative log-likelihood function can be expressed as follows

$$-\frac{\log(L(\boldsymbol{\Sigma}|\mathcal{X}))}{N} = \frac{1}{2N} \sum_{i=1}^{N} \rho(\boldsymbol{x}_i^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x}_i) + \frac{1}{2} \log \det(\boldsymbol{\Sigma}). \tag{47}$$

This is the energy in (28) and its minimization is equivalent to maximization of the likelihood function. Using basic calculus, we calculate the derivative of this function as

$$-\frac{\partial}{\partial \boldsymbol{\Sigma}} \frac{\log(L(\boldsymbol{\Sigma}|\mathcal{X}))}{N} = \tag{48}$$

$$\frac{1}{2N} \sum_{i=1}^{N} \rho'(\boldsymbol{x}_i^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x}_i) \boldsymbol{\Sigma}^{-1} \boldsymbol{x}_i \boldsymbol{x}_i^T \boldsymbol{\Sigma}^{-1} + \frac{1}{2} \boldsymbol{\Sigma}^{-1}.$$

Setting (48) equal to zero, the minimizer of (47), or equivalently (28), can be obtained by solving the following equation for $\boldsymbol{\Sigma}$, where $w(t) = \rho'(t)$:

$$\frac{1}{N} \sum_{i=1}^{N} w(\boldsymbol{x}_i^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x}_i) \boldsymbol{x}_i \boldsymbol{x}_i^T = \boldsymbol{\Sigma}. \tag{49}$$

We first note that when f is a multivariate Gaussian distribution then $g(t) = \exp(-t/2)/c(D) = \exp(-t/2)/(2\pi)^{D/2}$. This implies that $w(t) = 1$ and the corresponding minimizer of (28), whose formula is expressed in (49), is the sample covariance matrix. On the other hand, when g has heavier tails, e.g. $g(t) = \exp(-t^p/2p)/c(D, p)$ for $0 < p < 1$, (49) results in more robust estimators to heavy tails. Indeed, in this case $w(t) = t^{p-1}$ and the solution of equation (49) can be interpreted as a more robust version of the covariance matrix. In the left hand side of (49), each term $\boldsymbol{x}_i \boldsymbol{x}_i^T$ is weighted by $(\boldsymbol{x}_i^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x}_i)^{p-1}$. We further note that since we want to emphasize the top d eigenvectors of $\boldsymbol{\Sigma}$, we may identify $\boldsymbol{x}_i$ as an "outlier" whenever $\boldsymbol{x}_i^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x}_i$ is relatively large, or equivalently, when $(\boldsymbol{x}_i^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x}_i)^{p-1}$ is relatively small. Therefore, the left hand side of (49) is a weighted covariance matrix, which tends to de-emphasize outliers.

Another heavy-tailed density function can be obtained by considering the D-variate Student's t-distribution with ν degrees of freedom. In this case, $g(t) = c/(t + \nu)^{(D+\nu)/2}$ for some constant c and thus $w(t) = (D + \nu)/(t + \nu)$. The tails of this distribution are heaviest when ν approaches zero. Formally, in this case, $\rho(t) = D\log(t) - 2\log(c)$, $w(t) = D/t$ and the energy in (29) corresponds to the expression in (47) divided by D with the non-constant part of ρ, that is, $\rho(t) = D\log(t)$. This energy in (29) is different than the one in (28) since its solution is not unique over the set of positive definite matrices and an additional requirement, such as $\text{tr}(\boldsymbol{\Sigma}) = 1$, is needed. On the other hand, the Maronna M-estimator assumes some conditions on ρ that guarantee a unique minimizer of (28) over the set of positive definite matrices.

A typical example of the Maronna M-estimator is the one mentioned above, where $w(t) = \rho'(t) = t^{p-1}$ for $0 < p < 1$. Notice that the non-constant part of ρ is $\rho(t) = t^p/p$. For this and other ρ's satisfying the required conditions, the Maronna M-estimator can be computed by the following iterative procedure arbitrarily initialized with any positive definite matrix $\boldsymbol{\Sigma}_0$:

$$\boldsymbol{\Sigma}_{k+1} = \frac{1}{N} \sum_{i=1}^{N} w(\boldsymbol{x}_i^T \boldsymbol{\Sigma}_k^{-1} \boldsymbol{x}_i) \boldsymbol{x}_i \boldsymbol{x}_i^T. \tag{50}$$

Numerical properties of this solution and, in particular, its convergence to the fixed point in (49) are discussed in [5]

and [69]. TME can similarly be computed by substituting $w(t) = D/t$ in (50) and dividing the resulting Σ_{k+1} by $\text{tr}(\Sigma_{k+1})$, so that it satisfies the constraint $\text{tr}(\Sigma_{k+1}) = 1$. Numerical properties of this solution for TME are discussed in [49] and [118].

As explained in [69], this framework can be formally extended to the more general setting where both μ and Σ are unknown, and one wishes to estimate them jointly. We remark that the estimate of μ in this procedure would be a robust point estimator. Alternatively, one can follow the symmetrization procedure explained in §II-F and independently estimate Σ. The mean, μ, can then be estimated separately by some robust point estimator. As alluded to in §II-F, the advantage of the latter procedure over the former one is that errors in estimating μ do not propogate errors in estimating Σ.

Gilad Lerman received the B.S. degree in Physics and Mathematics (summa cum laude) and M.S. degree in applied mathematics from Tel Aviv University in 1993 and 1995 respectively, and the Ph.D. degree in Mathematics from Yale University in 2000. After graduation, he was a Courant Instructor at the Courant Institute. Since 2004, he has been a faculty member in the School of Mathematics at the University of Minnesota, where he is currently a professor of Mathematics and director of the Minnesota Center for Industrial Mathematics and the IMA data science lab. His research interests include high-dimensional data analysis and modeling, non-convex optimization, robust statistics and machine learning. He is a recipient of an NSF CAREER award in 2010 and the Feinberg Foundation Visiting Faculty Fellowship at the Weizmann Institute in 2013.

Tyler Maunu received the B.S., M.S., and Ph.D. degrees in Mathematics from University of Minnesota, Minneapolis, MN, USA, in 2013, 2016, and 2018 respectively. He also received the M.S. degree in Statistics at the University of Minnesota, Minneapolis, MN, USA in 2018. His research interests include machine learning, robust statistics, and non-convex optimization.