

Robust computation of linear models by convex relaxation

Gilad Lerman · Michael B. McCoy ·
Joel A. Tropp · Teng Zhang

18 February 2012. Revised 31 May 2013.

Abstract Consider a dataset of vector-valued observations that consists of noisy inliers, which are explained well by a low-dimensional subspace, along with some number of outliers. This work describes a convex optimization problem, called REAPER, that can reliably fit a low-dimensional model to this type of data. This approach parameterizes linear subspaces using orthogonal projectors, and it uses a relaxation of the set of orthogonal projectors to reach the convex formulation. The paper provides an efficient algorithm for solving the REAPER problem, and it documents numerical experiments which confirm that REAPER can dependably find linear structure in synthetic and natural data. In addition, when the inliers lie near a low-dimensional subspace, there is a rigorous theory that describes when REAPER can approximate this subspace.

Keywords Robust linear models · Convex relaxation · Iteratively reweighted least squares

Mathematics Subject Classification (2000) 62H25 · 65K05 · 90C22

1 Introduction

Low-dimensional linear models have applications in a huge array of data analysis problems. Let us highlight some examples from computer vision, machine learning, and bioinformatics.

Communicated by Emmanuel Candès

G. Lerman
Department of Mathematics, University of Minnesota, 127 Vincent Hall, 206 Church St SE, Minneapolis, MN 55455 USA, E-mail: lerman@umn.edu

M. B. McCoy (*corresponding author*)
Cofacet, Inc. 693 E Del Mar Blvd., Pasadena, CA 91101 E-mail: mccoy@cofacet.com

J. A. Tropp
Department of Computing and Mathematical Sciences, California Institute of Technology, MC 305-16 1200 E. California Blvd. Pasadena, CA 91125, E-mail: jtropp@cms.caltech.edu

T. Zhang
University of Minnesota, Institute for Mathematics and its Applications, 207 Church Street SE, Minneapolis, MN 55455, E-mail: zhang620@umn.edu

Illumination models Images of a face—or any Lambertian object—viewed under different illumination conditions lie near a nine-dimensional subspace [28, 38, 3].

Structure from motion Feature points on a moving rigid body lie on an affine space of dimension three, assuming the affine camera model [16]. More generally, estimating structure from motion involves estimating low-rank matrices [29, Sec. 5.2].

Latent semantic indexing We can describe a large corpus of documents that concern a small number of topics using a low-dimensional linear model [23].

Population stratification Low-dimensional models of single nucleotide polymorphism (SNP) data have been used to show that the genotype of an individual is correlated with her geographical ancestry [57]. More generally, linear models are used to assess differences in allele frequencies among populations [61].

In most of these applications, the datasets are noisy, and they contain a substantial number of outliers. Principal component analysis, the standard method for finding a low-dimensional linear model, is sensitive to these non-idealities. As a consequence, good robust modeling techniques would be welcome in a range of scientific and engineering disciplines.

In recent years, researchers have started to use convex optimization to develop alternatives to principal component analysis that have more favorable robustness properties. For the most part, these formulations attempt to find a low-rank matrix that approximates the data well. They typically use the Schatten 1-norm as a convex proxy for the rank. See Section 6 for a more complete discussion. Although these ideas are compelling, it remains valuable to explore other methods because of the importance of linear modeling.

This paper describes a new technique for fitting a low-dimensional linear model to data. Our formulation is based on convex optimization, but it has a different flavor from the earlier techniques. We use a new set of ideas to develop a rigorous analysis of the performance of our method. This theory demonstrates that the approach is robust against noise in the inliers, and it can cope with a large number of adversarial outliers. We describe an efficient numerical algorithm that is guaranteed to solve the optimization problem after a modest number of spectral calculations. We also include some experiments with synthetic and natural data to verify that our technique reliably seeks out linear structure.

1.1 Notation and Preliminaries

In this paper, we work with real-valued data. We write $\|\cdot\|$ for the ℓ_2 norm on vectors and the spectral norm on matrices; $\|\cdot\|_F$ represents the Frobenius norm; $\|\cdot\|_{S_1}$ refers to the Schatten 1-norm. Angle brackets $\langle \cdot, \cdot \rangle$ denote the standard inner product on vectors and matrices, and tr refers to the trace. The curly inequality \preceq denotes the semidefinite order: For symmetric matrices A and B , we write $A \preceq B$ if and only if $B - A$ is positive semidefinite.

An *orthoprojector* is a symmetric matrix Π that satisfies $\Pi^2 = \Pi$. Each subspace L in \mathbb{R}^D is the range of a unique $D \times D$ orthoprojector Π_L . The trace of an orthoprojector equals the dimension of its range: $\text{tr}(\Pi_L) = \dim(L)$. For each point $x \in \mathbb{R}^D$, the image $\Pi_L x$ is the best ℓ_2 approximation of x in the subspace L . The orthogonal complement of a subspace L is expressed as L^\perp .

For a real number a , the notation $\lfloor a \rfloor$ refers to the greatest integer that does not exceed a , and $\lceil a \rceil$ refers to the smallest integer that is at least as large as a . These operations are usually referred to as *floor* and *ceiling*, respectively. We also define the function $[a]_+ := \max\{a, 0\}$, which returns the positive part of a real number.

Finally, we introduce the *spherization transform* for vectors:

$$\tilde{\mathbf{x}} := \begin{cases} \mathbf{x} / \|\mathbf{x}\|, & \mathbf{x} \neq \mathbf{0} \\ \mathbf{0}, & \text{otherwise.} \end{cases} \quad (1.1)$$

We extend the spherization transform to matrices by applying it separately to each column.

1.2 Linear Modeling by Principal Component Analysis

To motivate our approach to linear modeling, we summarize a classical line of research in statistics that begins with principal component analysis.

Let \mathcal{X} be a dataset¹ consisting of N points in \mathbb{R}^D . Suppose we wish to determine a d -dimensional subspace that best explains the data. For each point, we can measure the residual error in the approximation by computing the orthogonal distance from the point to the subspace. The classical method for fitting a subspace asks us to minimize the sum of the *squared* residuals:

$$\text{minimize } \sum_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x} - \mathbf{\Pi} \mathbf{x}\|^2 \quad \text{subject to } \mathbf{\Pi} \text{ is an orthoprojector and } \text{tr } \mathbf{\Pi} = d. \quad (1.2)$$

(Here and elsewhere, sums indexed by a dataset repeat each point as many times as it appears in the dataset.) The approach (1.2) is equivalent with the method of *principal component analysis* (PCA) from the statistics literature [41] and the *total least squares* (TLS) method from the linear algebra community [40].

The mathematical program (1.2) is not convex because orthoprojectors do not form a convex set, so we have no right to expect that the problem is tractable. Nevertheless, we can compute an analytic solution by means of a singular value decomposition (SVD) of the data [27, 40]. Suppose that \mathbf{X} is a $D \times N$ matrix whose columns are the data points, arranged in fixed order, and let $\mathbf{X} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^t$ be an SVD of this matrix. Form the $D \times d$ matrix \mathbf{U}_d by extracting the first d columns of \mathbf{U} ; the columns of \mathbf{U}_d are often called the *principal components* of the data. Then we can construct an optimal point $\mathbf{\Pi}_*$ for (1.2) using the formula $\mathbf{\Pi}_* = \mathbf{U}_d \mathbf{U}_d^t$.

1.3 Classical Methods for Achieving Robustness

Imagine now that the dataset \mathcal{X} contains *inliers*, points we hope to explain with a linear model, as well as *outliers*, points that come from another process, such as a different population or noise. The data are not labeled, so it may be challenging to distinguish inliers from outliers. If we apply the PCA formulation (1.2) to fit a subspace to \mathcal{X} , the rogue points can interfere with the linear model for the inliers.

To guard the subspace estimation procedure against outliers, statisticians have proposed to replace the sum of squares in (1.2) with a figure of merit that is less sensitive to outliers. One possibility is to sum the *unsquared* residuals, which reduces the contribution from

¹ A dataset is simply a finite multiset, that is, a finite set with repeated elements allowed.

large residuals that may result from aberrant data points. This idea leads to the following optimization problem.

$$\begin{aligned} \text{minimize} \quad & \sum_{x \in \mathcal{X}} \|x - \Pi x\| \quad \text{subject to} \quad \Pi \text{ is an orthoprojector} \quad \text{and} \\ & \text{tr } \Pi = d. \end{aligned} \quad (1.3)$$

In case $d = D - 1$, the problem (1.3) is sometimes called *orthogonal ℓ_1 regression* [65] or *least orthogonal absolute deviations* [58]. The extension to general d is apparently more recent [73, 25]. See the books [39, 64, 55] for an extensive discussion of other ways to combine residuals to obtain robust estimators.

Unfortunately, the mathematical program (1.3) is not convex, and, in contrast to (1.2), no *deus ex machina* emerges to make the problem tractable. Although there are many algorithms [58, 12, 2, 74, 25, 81] that attempt (1.3), none is guaranteed to return a global minimum. In fact, most of the classical proposals for robust linear modeling involve intractable optimization problems, which makes them poor options for computation in spite of their theoretical properties [55].

1.4 A Convex Program for Robust Linear Modeling

The goal of this paper is to develop, analyze, and test a rigorous method for fitting robust linear models by means of convex optimization. We propose to *relax* the hard optimization problem (1.3) by replacing the nonconvex constraint set with a larger convex set. The advantage of this approach is that we can solve the resulting convex program completely using a variety of efficient algorithms.

The idea behind our relaxation is straightforward. Each eigenvalue of an orthoprojector Π equals zero or one because $\Pi^2 = \Pi$. Although a 0–1 constraint on eigenvalues is hard to enforce, the symmetric matrices whose eigenvalues lie in the interval $[0, 1]$ form a convex set. This observation leads us to frame the following convex optimization problem. Given a dataset \mathcal{X} in \mathbb{R}^D and a target dimension $d \in \{1, 2, \dots, D - 1\}$ for the linear model, we solve

$$\begin{aligned} \text{minimize} \quad & \sum_{x \in \mathcal{X}} \|x - Px\| \quad \text{subject to} \quad \mathbf{0} \preceq P \preceq \mathbf{I} \quad \text{and} \quad \text{tr } P = d. \end{aligned} \quad (1.4)$$

We refer to (1.4) as REAPER because it attempts to harvest linear structure from data.

1.4.1 A Tighter Relaxation?

One may wonder whether it is possible to find a tighter relaxation of (1.3) than our proposed formulation (1.4). If we restrict our attention to convex programs, the answer is negative.

Fact 1.1. For each integer $d \in [0, D]$, the set $\{P \in \mathbb{R}^{D \times D} : \mathbf{0} \preceq P \preceq \mathbf{I} \text{ and } \text{tr } P = d\}$ is the convex hull of the $D \times D$ orthoprojectors with trace d .

To prove this fact, it suffices to apply a diagonalization argument and to check that the set $\{\lambda \in \mathbb{R}^D : \sum_{i=1}^D \lambda_i = d \text{ and } 0 \leq \lambda_i \leq 1\}$ is the convex hull of the set of vectors that have d ones and $D - d$ zeros. See [60] for a discussion of this result.

Fact 1.1 gives a geometric indication about why REAPER might be effective. Suppose there is a rank- d orthoprojector Π_L that provides a good linear model for the inliers. The constraint set in (1.4) is the convex hull of the rank- d orthoprojectors. In high dimensions, convex hulls tend to be very small, so there are relatively few perturbations of Π_L that remain

feasible for (1.4). At the same time, the objective function in (1.4) is a sort of ℓ_1 norm, so it has relatively few directions of descent at Π_L . We have the intuition that it is impossible to move far from Π_L into the constraint set while simultaneously reducing the objective function. This insight is ultimately the basis for our analysis.

1.4.2 Computing an Orthoprojector from the Solution of REAPER

It is easy to see that a solution P_* to the REAPER problem has rank d or greater. On the other hand, the matrix P_* does need not to be an orthoprojector, so it is not immediately clear how to obtain a d -dimensional linear model from a minimizer of REAPER. To accomplish this goal, let us consider the auxiliary problem

$$\begin{aligned} \text{minimize} \quad & \|P_* - \Pi\|_{S_1} \quad \text{subject to} \quad \Pi \text{ is an orthoprojector} \quad \text{and} \\ & \text{tr } \Pi = d. \end{aligned} \tag{1.5}$$

In other words, we find a rank- d orthoprojector Π_* that is closest to P_* in Schatten 1-norm. We use the range of Π_* as our linear model.

It is straightforward to compute a solution Π_* to the problem (1.5). We just need to construct an orthogonal projector whose range is a dominant d -dimensional invariant subspace of P_* . More precisely, we form the spectral factorization $P_* = U \Lambda U^t$ where the entries of the diagonal matrix Λ are listed in weakly decreasing order. Extract the $D \times d$ matrix U_d consisting of the first d columns of U . Then an optimal point for (1.5) is given by the formula $\Pi_* = U_d U_d^t$. (This well-known recipe for solving (1.5) can be verified using a straightforward modification of the argument leading to [4, Thm. IX.7.2].)

The range of the matrix Π_* often provides a very good fit for the inlying data points, even when there are many outliers. This paper provides theoretical and empirical support for this claim. In Section 4, we present a numerical algorithm for solving (1.4) efficiently. Section 5.1 outlines some practical issues that are important in applications.

1.5 Main Contributions

This work partakes in a larger research vision: Given a difficult nonconvex optimization problem, it is often more effective to solve a convex variant than to accept a local minimizer of the original problem.

We believe that the main point of interest is our application of convex optimization to solve a problem involving subspaces. There are two key observations here. We parameterize subspaces by orthoprojectors, and then we replace the set of rank- d orthoprojectors with its convex hull. This relaxation has a different character from previous approaches to robust linear modeling, so we have found it necessary to develop a new type of analysis to obtain theoretical results for REAPER.

We have also done some numerical work which indicates that REAPER can be more effective than its competitors for certain types of robust linear modeling. After the original version [46] of this manuscript appeared, our ideas have been applied to a difficult class of problems involving orthogonality constraints, and this approach sometimes outperforms its competitors [72]. Together, these papers suggest that relaxations like REAPER can be used to address important geometric questions in data analysis.

1.6 Roadmap

We close this introduction with an outline of the paper. In Section 2, we develop a deterministic analysis of REAPER that describes when it can recover a linear model from a noisy dataset that includes outliers. Section 3 instantiates this result for a simple random data model. In Section 4, we develop an efficient numerical method for solving the REAPER problem. Then we describe a numerical example involving an image database in Section 5. We discuss related work in Section 6. The technical details that support our work appear in the appendices.

2 Theoretical Analysis of the REAPER Problem

The goal of this section is to provide theoretical evidence that the REAPER problem (1.4) is an effective way to find a robust linear model for a dataset. To do so, we consider a very general deterministic setup where the data consists of inliers that are located near a fixed subspace and outliers that may appear anywhere in the ambient space. We then introduce summary statistics for the data that encapsulate some of its geometric properties. Using these statistics, we state our main result, Theorem 2.1, which gives a bound on how well REAPER is able to approximate the model subspace. This result indicates why REAPER may be more effective than PCA for very noisy data. At the end of the section, we summarize the main ideas in the proof of Theorem 2.1, leaving the remaining details until Appendix A.

2.1 A Deterministic Data Model

To analyze the performance of the REAPER method, we need to introduce a model for the input data. It is natural to consider the case where the dataset contains inliers that lie on or near a fixed low-dimensional subspace, while the outliers can be arrayed arbitrarily in the ambient space. We formalize this intuition in a set of assumptions that we refer to as the In & Out Model, and we direct the reader to Table 2.1 for a detailed list of the parameters.

Table 2.1 *The In & Out Model.* A deterministic model for data with linear structure that is contaminated with outliers.

D	Dimension of the ambient space
L	A proper d -dimensional subspace of \mathbb{R}^D
N_{in}	Number of inliers
N_{out}	Number of outliers
\mathcal{X}_{in}	Dataset of N_{in} inliers, located “near” the subspace L
\mathcal{X}_{out}	Dataset of N_{out} outliers, at arbitrary locations in $\mathbb{R}^D \setminus L$
\mathcal{X}	Dataset $\mathcal{X}_{\text{in}} \cup \mathcal{X}_{\text{out}}$ containing all the observations
\mathbf{X}_{out}	$D \times N_{\text{out}}$ matrix whose columns are the outliers

The key point about the In & Out Model is that all the inliers are located near a subspace L , so it is reasonable for us to investigate when an algorithm can approximate this target subspace L .

2.2 Summary Parameters for the In & Out Model

The In & Out Model is very general, so we cannot hope to approximate the target subspace L without making further assumptions on the data. In this section, we develop some geometric summary statistics that allow us to check when REAPER is effective at finding the subspace L . Heuristically, we need the inliers to provide a significant amount of evidence for the subspace, while the outliers cannot exhibit too much linear structure. Otherwise, an unsupervised algorithm would be justified in finding a subspace that describes the outliers instead of the inliers!

Let us begin with a discussion of what it means for the inliers to provide evidence for a specific subspace $M \subset \mathbb{R}^D$. Imagine that we approximate each inlier \mathbf{x} with the point $\Pi_M \mathbf{x}$ in the subspace M . These approximations must have two properties. First, we want the approximations of the inliers to corroborate all the directions in the subspace M . Second, we need to be sure that the residual error in the approximations is not too large. Our first two summary statistics are designed to address these requirements.

To quantify how well the inliers fill out a subspace $M \subset \mathbb{R}^D$, we introduce the *permeance statistic* $\mathcal{P}(M)$.

$$\mathcal{P}(M) := \inf_{\substack{\mathbf{u} \in M \\ \|\mathbf{u}\|=1}} \sum_{\mathbf{x} \in \mathcal{X}_{\text{in}}} |\langle \mathbf{u}, \Pi_M \mathbf{x} \rangle|. \quad (2.1)$$

If there is a direction in the subspace M that is orthogonal to each inlier, then the permeance statistic $\mathcal{P}(M)$ is zero. On the other hand, the permeance statistic is large when every direction \mathbf{u} in M has the property that many inliers have a component along \mathbf{u} .

Second, we introduce the *total inlier residual* $\mathcal{R}(M)$ to measure the total error that we incur by approximating the data using the subspace M .

$$\mathcal{R}(M) := \sum_{\mathbf{x} \in \mathcal{X}_{\text{in}}} \|\Pi_{M^\perp} \mathbf{x}\|. \quad (2.2)$$

Let us emphasize that the total inlier residual is less sensitive to large errors than the sum of squared residuals that drives the PCA method.

Next, let us turn to the condition that we require of the outliers. A major challenge for any robust linear modeling procedure is the possibility that both the inliers and the outliers exhibit linear structure. In this case, an algorithm may choose to fit a linear model to the outliers if they have a stronger signature.

To measure the amount of linear structure in the outliers, we introduce the *alignment statistic* $\mathcal{A}(M)$ with respect to a target subspace M .

$$\mathcal{A}(M) := \|\mathbf{X}_{\text{out}}\| \cdot \|\widetilde{\Pi_{M^\perp} \mathbf{X}_{\text{out}}}\|, \quad (2.3)$$

where \mathbf{X}_{out} is the matrix whose columns are the outlying data points and the spherization operator $\widetilde{\cdot}$ normalizes the columns of a matrix. It is somewhat harder to understand what the alignment statistic $\mathcal{A}(M)$ reflects. First, observe that the spectral norm $\|\mathbf{X}_{\text{out}}\|$ tends to be large when the outliers are collinear, and it is small when the outliers are weakly correlated. The other term in the alignment statistic asks about the collinearity of the outliers after we have removed their components in the subspace M .

Finally, we present one more statistic that weighs the influence of the inliers against the influence of the outliers. The *stability statistic* $\mathcal{S}(M)$ of the data with respect to a subspace $M \subset \mathbb{R}^D$ is the quantity

$$\mathcal{S}(M) := \frac{\mathcal{P}(M)}{4\sqrt{\dim(M)}} - \mathcal{A}(M). \quad (2.4)$$

The stability statistic tends to be large when the inliers provide a lot of evidence for the subspace M and the outliers contain relatively little distracting linear structure. As we will see, when $\mathcal{S}(M)$ is large, the REAPER method can be very effective at approximating the subspace M , even when the inliers are noisy.

2.3 Performance of REAPER with Deterministic Data

The main theoretical result in this paper describes the behavior of the REAPER method when it is applied to data that meet the assumptions of the In & Out Model from Table 2.1.

Theorem 2.1 (Performance Analysis for REAPER). *Fix any d -dimensional subspace L of \mathbb{R}^D , and assume that \mathcal{X} is a dataset that conforms to the In & Out Model on page 6. Let \mathbf{P}_\star be a solution to the REAPER problem (1.4), and find the nearest d -dimensional orthoprojector $\mathbf{\Pi}_\star$ by solving (1.5). Then we have the error bound*

$$\|\mathbf{\Pi}_\star - \mathbf{\Pi}_L\|_{S_1} \leq \frac{4\mathcal{R}(L)}{[\mathcal{S}(L) - \mathcal{R}(L)]_+}.$$

The stability statistic $\mathcal{S}(L)$ is defined in (2.4), and the total inlier residual $\mathcal{R}(L)$ is defined in (2.2).

An overview of the proof of Theorem 2.1 appears below in Section 2.4. Before we present the argument, let us explain the content of this result.

1. Assume that all the inliers are contained within the target subspace L . Then the total residual $\mathcal{R}(L) = 0$. If the stability statistic $\mathcal{S}(L) > 0$, then Theorem 2.1 ensures that $\mathbf{\Pi}_\star = \mathbf{\Pi}_L$. In other words, we recover the subspace L without error.
2. Again, suppose that the inliers are located within the target subspace L . As we begin to move the inliers away from L , the error in approximating the subspace increases at a linear rate proportional to $\mathcal{S}(L)^{-1}$. Therefore, when the stability statistic is large, the noise in the inliers has a very small impact on the approximation error.
3. The effect of outliers appears only through the alignment statistic (2.3). When the inliers lie in the subspace L , the alignment statistic is the largest when the outliers cluster along a one-dimensional subspace in L^\perp . With adversarial outliers, our theory indicates that a very large permeance (2.1) is required to counteract linear structure in the outliers.
4. We have measured the distance between the projectors using the Schatten 1-norm, which provides a very strong bound indeed. To appreciate the value of this type of estimate, note that it follows from [4, p. 202] that for any two d -dimensional subspaces M, M' of \mathbb{R}^D ,

$$\|\mathbf{\Pi}_M - \mathbf{\Pi}_{M'}\|_{S_1} = 2 \sum_{i=1}^d \sin \theta_i(M, M') \geq \frac{4}{\pi} \sum_{i=1}^d \theta_i(M, M'),$$

where $\theta_i(M, M')$ is the i th principal angle between the subspaces, and we use the fact that $\sin(\theta) \geq 2\theta/\pi$ for $0 \leq \theta \leq \pi/2$. Therefore, our error bound allows us to control all the principal angles between the computed subspace $\text{range}(\mathbf{\Pi}_\star)$ and the target subspace L .

5. Imagine that we knew in advance which points were inliers. Then we could pose the oracle ℓ_1 orthogonal regression problem:

$$\begin{aligned} \text{minimize} \quad & \sum_{\mathbf{x} \in \mathcal{X}_{\text{in}}} \|(\mathbf{I} - \mathbf{\Pi})\mathbf{x}\| \quad \text{subject to} \quad \mathbf{\Pi} \text{ is an orthoprojector} \quad \text{and} \\ & \text{tr } \mathbf{\Pi} = d. \end{aligned}$$

Let $\mathbf{I}_{\text{oracle}}$ be a solution to this (apparently intractable) problem. Then the subspace $L_{\text{oracle}} := \text{range}(\mathbf{I}_{\text{oracle}})$ minimizes the total inlier residual $\mathcal{R}(M)$ over d -dimensional subspaces $M \subset \mathbb{R}^D$. When we apply Theorem 2.1 with $L = L_{\text{oracle}}$, we discover that REAPER identifies a linear model that is close to the oracle ℓ_1 model—provided that the oracle model is sufficiently stable. This observation is interesting even when there are no outliers.

6. How does REAPER compare with standard PCA? The formulation (1.2) shows that PCA searches for a subspace by minimizing the sum of squared residuals. On the other hand, we have just seen that REAPER is (almost) capable of finding a subspace that minimizes the sum of unsquared residuals. It is well known that the sum of unsquared residuals tends to be much less sensitive to large errors than the sum of squared residuals. As a consequence, we expect that REAPER will be more effective at ignoring data points that contribute large errors. See Figure 3.2 below for numerical evidence of this phenomenon.

In short, Theorem 2.1 indicates that REAPER has the qualitative features that one desires in a method for robust linear modeling. In Section 3, we instantiate the result for a simple random model to offer some insight about how the summary statistics scale.

2.4 Proof of Theorem 2.1

This section contains the main steps in the proof of Theorem 2.1. Most of the technical details are encapsulated in two lemmata, which we establish in Appendix A. Throughout this section and the appendix, we retain the notation and assumptions of the In & Out Model from page 6.

The argument is based on several ideas. First, if the inliers are contained within a low-dimensional subspace L , then REAPER can identify this subspace whenever the stability statistic $\mathcal{S}(L) > 0$. To show that (1.4) recovers L exactly in this case, we prove that every feasible perturbation of the projector \mathbf{I}_L increases the objective. This type of primal analysis is similar in spirit to the argument in [68, 69], but the technical details are harder because we are working with matrices. It contrasts with the style of analysis that dominates recent papers on convex methods for robust linear modeling, which are usually based on elaborate constructions of dual certificates.

Second, when the inliers are not contained in the subspace L , we can use a perturbation analysis to assess how much the noise impacts the solution to the optimization problem. The key idea here is to replace the objective function in (1.4) with a nearby objective function. This alteration allows us to take advantage of the exact recovery results that we mentioned in the last paragraph. The approach is based on some classic arguments in optimization; see [7, Sec. 4.4.1]. We do not believe these ideas have been applied in the literature on convex relaxations of data analysis problems.

To begin, we introduce some notation. The REAPER problem (1.4) can be framed as

$$\text{minimize } f(\mathbf{P}) \quad \text{subject to } \mathbf{P} \in \Phi. \quad (2.5)$$

with objective function

$$f(\mathbf{P}) := \sum_{\mathbf{x} \in \mathcal{X}} \|(\mathbf{I} - \mathbf{P})\mathbf{x}\| \quad (2.6)$$

and feasible set

$$\Phi := \{\mathbf{P} : \mathbf{0} \preceq \mathbf{P} \preceq \mathbf{I} \quad \text{and} \quad \text{tr}(\mathbf{P}) = d\}. \quad (2.7)$$

Let \mathbf{P}_\star be any solution to (1.4). Next, we find a solution $\mathbf{I}\mathbf{I}_\star$ to the problem

$$\text{minimize } \|\mathbf{P}_\star - \mathbf{I}\mathbf{I}\|_{S_1} \quad \text{subject to } \mathbf{I}\mathbf{I} \text{ is a rank-}d \text{ orthoprojector.} \quad (2.8)$$

Our aim is to compare the computed projector $\mathbf{I}\mathbf{I}_\star$ with the target projector $\mathbf{I}\mathbf{I}_L$.

The main technical insight is to use the target projector $\mathbf{I}\mathbf{I}_L$ to construct a perturbation g of the objective function f of the REAPER problem:

$$g(\mathbf{P}) := \sum_{\mathbf{x} \in \mathcal{X}_{\text{in}}} \|(\mathbf{I} - \mathbf{P})\mathbf{I}\mathbf{I}_L \mathbf{x}\| + \sum_{\mathbf{x} \in \mathcal{X}_{\text{out}}} \|(\mathbf{I} - \mathbf{P})\mathbf{x}\|. \quad (2.9)$$

To perform the analysis, we pass from the original optimization problem (2.5) to the perturbed problem

$$\text{minimize } g(\mathbf{P}) \quad \text{subject to } \mathbf{P} \in \Phi. \quad (2.10)$$

Observe that, if the inliers are contained in the target subspace L , then the perturbed problem (2.10) coincides with the original problem (2.5).

The argument requires two technical results. The first lemma shows that the total inlier residual $\mathcal{R}(L)$ controls the difference between the perturbed objective g and the original objective f . The second lemma shows, in particular, that $\mathbf{I}\mathbf{I}_L$ is the unique minimizer of (2.10) when the stability statistic $\mathcal{S}(L) > 0$. Together, these estimates allow us to conclude that the solution to the original problem (2.5) is not far from $\mathbf{I}\mathbf{I}_L$.

More precisely, we demonstrate that the perturbed objective g is close to the original objective f for matrices close to $\mathbf{I}\mathbf{I}_L$.

Lemma 2.2 (Controlling the Size of the Perturbation). *Introduce the difference $h := f - g$ between the two objectives. Then*

$$|h(\mathbf{I}\mathbf{I}_L) - h(\mathbf{I}\mathbf{I}_L + \mathbf{\Delta})| \leq \mathcal{R}(L) \cdot [2 + \|\mathbf{\Delta}\|_{S_1}].$$

for any symmetric matrix $\mathbf{\Delta}$. The total inlier residual $\mathcal{R}(L)$ is defined in (2.2).

The proof of Lemma 2.2 appears in Appendix A.1.

We also argue that the perturbed objective function g increases quickly when we move away from the point $\mathbf{I}\mathbf{I}_L$ into the feasible set.

Lemma 2.3 (Rate of Ascent of the Perturbed Objective). *Assume that $\mathbf{I}\mathbf{I}_L + \mathbf{\Delta} \in \Phi$. Then*

$$g(\mathbf{I}\mathbf{I}_L + \mathbf{\Delta}) - g(\mathbf{I}\mathbf{I}_L) \geq \mathcal{S}(L) \cdot \|\mathbf{\Delta}\|_{S_1}.$$

The stability statistic $\mathcal{S}(L)$ is defined in (2.4).

The proof of Lemma 2.3 appears in Appendix A.2.

Granted these two results, we quickly complete the proof of Theorem 2.1. Define the function $h := f - g$. Adding and subtracting terms, we find that

$$g(\mathbf{P}_\star) - g(\mathbf{I}\mathbf{I}_L) = [h(\mathbf{I}\mathbf{I}_L) - h(\mathbf{P}_\star)] + [f(\mathbf{P}_\star) - f(\mathbf{I}\mathbf{I}_L)] \leq h(\mathbf{I}\mathbf{I}_L) - h(\mathbf{P}_\star). \quad (2.11)$$

The inequality in (2.11) holds because the second bracket is nonpositive. Indeed, \mathbf{P}_\star minimizes f over the feasible set Φ , and $\mathbf{I}\mathbf{I}_L$ is also a member of the feasible set. Set $\mathbf{\Delta} = \mathbf{P}_\star - \mathbf{I}\mathbf{I}_L$, and apply Lemmas 2.2 and 2.3 to bound the right- and left-hand sides of (2.11). We reach

$$\mathcal{S}(L) \cdot \|\mathbf{P}_\star - \mathbf{I}\mathbf{I}_L\|_{S_1} \leq 2\mathcal{R}(L) + \mathcal{R}(L) \cdot \|\mathbf{P}_\star - \mathbf{I}\mathbf{I}_L\|_{S_1}.$$

Solve this inequality to reach the bound

$$\|P_\star - \Pi_L\|_{S_1} \leq \frac{2\mathcal{R}(L)}{[\mathcal{S}(L) - \mathcal{R}(L)]_+}. \quad (2.12)$$

To finish the argument, note that

$$\|\Pi_\star - \Pi_L\|_{S_1} \leq \|\Pi_\star - P_\star\|_{S_1} + \|P_\star - \Pi_L\|_{S_1} \leq 2\|P_\star - \Pi_L\|_{S_1} \leq \frac{4\mathcal{R}(L)}{[\mathcal{S}(L) - \mathcal{R}(L)]_+}.$$

The first bound follows from the triangle inequality. The second estimate holds because the distance from Π_\star to P_\star is no greater than the distance from Π_L to P_\star because Π_\star is a minimizer of (2.8). The last inequality follows from (2.12).

3 Theoretical Example: The Haystack Model

The In & Out Model is very general, so Theorem 2.1 applies to a wide variety of specific examples. To see the kind of results that are possible, let us apply Theorem 2.1 to study the behavior of REAPER for data drawn from a simple random model. We use standard tools from high-dimensional probability to compute the values of the summary statistics.

3.1 The Haystack Model

Let us consider a simple generative random model for a dataset. We call this the Haystack Model, and we refer the reader to Table 3.1 for a list of the assumptions and the parameters. The Haystack Model is not intended as a realistic description of data. Instead, the goal is to capture the idea that inliers admit a low-dimensional linear model, while the outliers are totally unstructured.

Table 3.1 *The Haystack Model.* A generative random model for data with linear structure that is contaminated with outliers. The abbreviation *i.i.d.* stands for *independent and identically distributed*.

Parameters

D	Dimension of the ambient space
L	A proper d -dimensional subspace of \mathbb{R}^D containing the inliers
N_{in}	Number of inliers
N_{out}	Number of outliers
ρ_{in}	Inlier sampling ratio $\rho_{\text{in}} := N_{\text{in}}/d$
ρ_{out}	Outlier sampling ratio $\rho_{\text{out}} := N_{\text{out}}/D$
σ_{in}^2	Variance of the inliers per subspace dimension
σ_{out}^2	Variance of the outliers per ambient dimension

Data

\mathcal{X}_{in}	Set of N_{in} inliers, drawn i.i.d. $\text{NORMAL}(\mathbf{0}, (\sigma_{\text{in}}^2/d) \Pi_L)$
\mathcal{X}_{out}	Set of N_{out} outliers, drawn i.i.d. $\text{NORMAL}(\mathbf{0}, (\sigma_{\text{out}}^2/D) \mathbf{I}_D)$
\mathcal{X}	The set $\mathcal{X}_{\text{in}} \cup \mathcal{X}_{\text{out}}$ containing all the data points

There are a few useful intuitions associated with this model. As the inlier sampling ratio ρ_{in} increases, the inliers fill out the subspace L more completely so the linear structure becomes more evident. As the outlier sampling ratio ρ_{out} increases, the outliers become more distracting and they may even start to exhibit some linear structure due to chance. Next, observe that we have scaled the points so that their energy does not depend on the dimensional parameters:

$$\mathbb{E} \|x\|^2 = \sigma_{\text{in}}^2 \quad \text{for } x \in \mathcal{X}_{\text{in}} \quad \text{and} \quad \mathbb{E} \|x\|^2 = \sigma_{\text{out}}^2 \quad \text{for } x \in \mathcal{X}_{\text{out}}.$$

As a result, when $\sigma_{\text{in}}^2 = \sigma_{\text{out}}^2$, we cannot screen outliers just by looking at their energy. The sampling ratios and the variances contain most of the information about the behavior of this model.

3.2 Analysis of the Haystack Model

Using methods from high-dimensional probability, we can analyze the stability statistic $\mathcal{S}(L)$ for a dataset drawn at random from the Haystack Model.

Theorem 3.1 (Analysis of the Haystack Model). *Fix a number $\beta > 0$, and assume that $1 \leq d \leq (D-1)/2$. Let L be an arbitrary d -dimensional subspace of \mathbb{R}^D , and draw the dataset \mathcal{X} at random according to the Haystack Model on page 11. The stability statistic satisfies the bound*

$$\mathcal{S}(L) \geq \frac{\sigma_{\text{in}}}{\sqrt{32\pi}} [\rho_{\text{in}} - \pi(4 + 2\beta)] - 6\sigma_{\text{out}} [\rho_{\text{out}} + 1 + \beta],$$

except with probability $3.5e^{-\beta d}$.

The proof of Theorem 3.1 appears in Appendix B. The restriction $d \leq (D-1)/2$ above simplifies the result; see Theorem B.1 for a comprehensive statement valid for $1 \leq d \leq D-1$.

To appreciate what this result means, it is helpful to set $\sigma_{\text{in}} = \sigma_{\text{out}} = 1$ and to suppress the values of the constants:

$$\mathcal{S}(L) \geq C_{\text{in}}\rho_{\text{in}} - C_{\text{out}}\rho_{\text{out}} - C_{\beta}. \quad (3.1)$$

We see that the stability statistic grows linearly with the inlier sampling ratio, and it decreases linearly with the outlier sampling ratio.

Since the inliers in the Haystack Model are contained in the subspace L , Theorem 2.1 shows that REAPER recovers L perfectly when the stability statistic is positive. Therefore, a sufficient condition for exact recovery is that ρ_{in} , the number of inliers *per subspace dimension*, should be at least a constant multiple of ρ_{out} , the number of outliers *per ambient dimension*. As a consequence, we can find low-dimensional linear structure in a high-dimensional space given a small number of examples, even when the number of outliers seems exorbitant.

Numerical experiment Figure 3.1 displays the results of a numerical experiment for REAPER under the Haystack Model. We fix the ambient dimension $D = 100$ and take L a subspace of dimension $d = 1$ or $d = 10$. The number N_{in} of inliers and the number N_{out} of outliers vary over an equally-spaced² grid. Note that the specific choice of the subspace L is immaterial because the model is rotationally invariant. The variance parameters are fixed ($\sigma_{\text{in}}^2 = \sigma_{\text{out}}^2 = 1$).

² In both figures, N_{out} increases in increments of twenty, while N_{in} increases in increments of two.

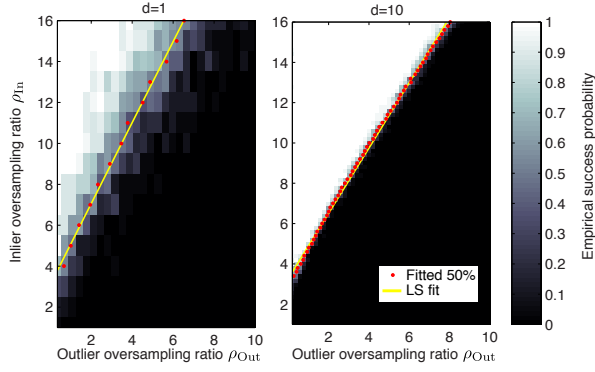


Fig. 3.1 *Exact subspace recovery with REAPER.* The heat maps show the empirical probability that REAPER identifies a target subspace under the Haystack Model with varying inlier ρ_{in} and outlier ρ_{out} oversampling ratios. We perform the experiments in ambient dimension $D = 100$ with inlier dimension $d = 1$ (left) and $d = 10$ (right). For each value of ρ_{in} , we find the 50% empirical success ρ_{out} (red dots). The yellow line indicates the least-squares fit to these points. In this parameter regime, a linear trend is clearly visible, which suggests that (3.1) captures the qualitative behavior of REAPER under the Haystack Model.

We find \mathbf{P}_\star by solving REAPER (1.4) with the algorithm described in Section 4 below, and then determine the orthoprojector $\mathbf{I}\mathbf{I}_\star$ using (1.5). We assess whether this procedure identifies the true subspace $\mathbf{I}\mathbf{I}_L$ subspace by declaring the experiment a success when the error $\|\mathbf{P}_\star - \mathbf{I}\mathbf{I}_L\|_{S_1} < 10^{-5}$. For each pair $(\rho_{\text{in}}, \rho_{\text{out}})$, we repeat the experiment 25 times and calculate an empirical success probability. For each value of ρ_{in} , we find the 50% empirical success ρ_{out} using a logistic fit. We fit a line to these points using standard least-squares. These results indicate that the linear trend suggested by the theoretical bound (3.1) reflects the empirical behavior of REAPER.

3.2.1 Noisy inliers

To understand how REAPER behaves when the inlying set \mathcal{X}_{in} does not lie precisely within the target subspace, we introduce the *Noisy Haystack Model*. This model expands the standard Haystack Model from Table 3.1 with the additional parameter σ_{noise}^2 that controls the amount of noise present in the inliers. In this extended model, the inlying data \mathcal{X}_{in} is given by

$$\mathcal{X}_{\text{in}}: \text{Set of } N_{\text{in}} \text{ inliers drawn i.i.d. } \text{NORMAL} \left(\mathbf{0}, \frac{\sigma_{\text{in}}^2}{d} \mathbf{I}\mathbf{I}_L + \frac{\sigma_{\text{noise}}^2}{D-d} \mathbf{I}\mathbf{I}_{L^\perp} \right). \quad (3.2)$$

All other parameters and data agree with the Haystack Model of Table 3.1.

The definition (3.2) of the inlying data ensures that the stability statistic $\mathcal{S}(L)$ has the same distribution under the Noisy Haystack Model as under the plain Haystack Model. In particular, the relationship (3.1) holds under the Noisy Haystack Model. On the other hand, the inlier residual statistic $\mathcal{R}(L)$ (2.2) is not equal to zero under the noisy model, but rather satisfies

$$\mathbb{E}[\mathcal{R}(L)] = \sum_{i=1}^{N_{\text{in}}} \mathbb{E}[\|\mathbf{g}_i\|] \leq N_{\text{in}} \mathbb{E}[\|\mathbf{g}_1\|^2]^{1/2} = \sigma_{\text{noise}} N_{\text{in}},$$

where $\mathbf{g}_i \sim \text{NORMAL}(\mathbf{0}, (\sigma_{\text{noise}}^2/(D-d)) \mathbf{I}\mathbf{I}_{L^\perp})$. The inequality is Jensen's, and the last expression uses the fact that the squared norm of a Gaussian random variable on the $(D-d)$ -dimensional subspace is $D-d$.

A basic concentration result indicates that the residual statistic will not exceed its mean by more than a factor of, say, two with overwhelming probability. (This claim is easily made precise using the result [6, Thm. 1.7.6].) Combining this observation with (3.1) and Theorem 2.1, we see that with high probability

$$\|\mathbf{I}_\star - \mathbf{I}_L\|_{S_1} \leq \frac{2N_{\text{in}} \times \text{SNR}}{[C_{\text{in}}\rho_{\text{in}} - C_{\text{out}}\rho_{\text{out}} - C_\beta - 2N_{\text{in}} \times \text{SNR}]_+}$$

where we define the signal-to-noise ratio $\text{SNR} := \sigma_{\text{in}}/\sigma_{\text{noise}}$. This inequality suggests that REAPER is stable under the Noisy Haystack Model in the regime where the stability statistic $\mathcal{S}(L) = O(1)$ and the signal-to-noise ratio $\text{SNR} = O(N_{\text{in}}^{-1})$. Our numerical experience suggests that this SNR restriction is conservative.

Numerical experiment Figure 3.2 compares the results of a numerical experiment under the Noisy Haystack Model using both REAPER and PCA. As in the experiment for the basic Haystack Model, we set $D = 100$ and perform the experiment for a linear subspace L of dimension $d = 10$ and $d = 1$. The variance parameters are $\sigma_{\text{in}} = \sigma_{\text{out}} = 1$, and we fix $\text{SNR} = \sigma_{\text{in}}/\sigma_{\text{noise}} = 10$. For each equally-spaced value³ of N_{in} and N_{out} , we draw the data \mathcal{X} from the Noisy Haystack Model. We determine a projector \mathbf{I}_\star by solving REAPER (1.4) and finding the closest subspace (1.5), and then we compute the error $\|\mathbf{I}_\star - \mathbf{I}_L\|_{S_1}$. We determine the same statistic for the projection given by PCA (1.5). We repeat this experiment 25 times for each value of $(\rho_{\text{in}}, \rho_{\text{out}})$.

The heat map in Figure 3.2 shows the mean error $\|\mathbf{I}_\star - \mathbf{I}_L\|_{S_1}$ over these trials for both REAPER and PCA. The blue region of the heat map begins where the error is less than 10% of the maximum possible error

$$\text{MaxError} := \max_{\dim(M)=d} \|\mathbf{I}_M - \mathbf{I}_L\|_{S_1} = 2d.$$

We see that REAPER is in the blue region over more of the parameter regime than PCA, which indicates that REAPER is more stable than PCA under the Noisy Haystack Model.

4 An Iterative Reweighted Least-Squares Algorithm for REAPER

Theorem 2.1 suggests that the REAPER problem (1.4) can be a valuable tool for robust linear modeling. On the other hand, REAPER is a semidefinite program, so it may be prohibitively expensive to solve using the standard interior-point methods. If we intend REAPER to be a viable approach for data analysis problems, it is incumbent that we produce a numerical method with more favorable scaling properties.

In this section, we describe a numerical algorithm for solving the REAPER problem (1.4). Our approach is based on the iterative reweighted least squares (IRLS) framework [5, Sec. 4.5.2]. At each step of the algorithm, we solve a weighted least-squares problem whose weights evolve as the algorithm proceeds. This subproblem admits a closed-form solution that we can obtain from a single SVD of the (weighted) data. The IRLS method exhibits linear convergence in practice, so it can achieve high accuracy without a substantial

number of iterations. In this experiment, N_{in} increases in increments of two while N_{out} increases in increments of 20.

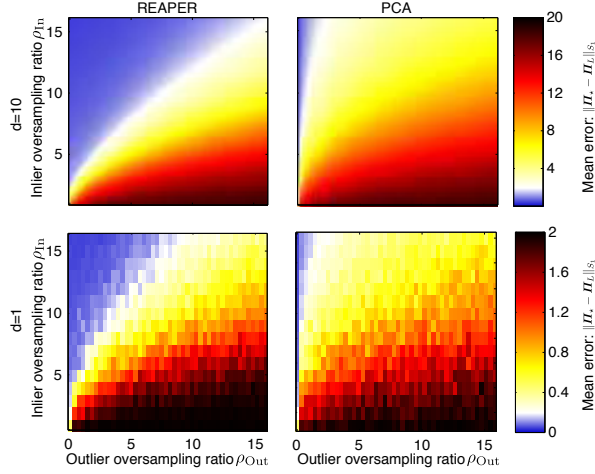


Fig. 3.2 Approximate subspace recovery with REAPER and PCA. The heat maps show the mean error $\|\Pi_* - \Pi_L\|_{S_1}$ for the projection computed by REAPER and PCA. The ambient dimension is $D = 100$, and we perform the experiment for both $d = 10$ (top) and $d = 1$ (bottom). The blue region indicates where the mean error is less than 10% of the maximum possible error.

4.1 Solving REAPER via IRLS

IRLS is based on the idea that we can solve many types of weighted least-squares problems efficiently. Therefore, instead of solving the REAPER problem (1.4) directly, we replace it with a sequence of weighted least-squares problems.

To motivate the approach, suppose we have an estimate $\beta_x \approx \|x - P_*x\|^{-1}$ for each $x \in \mathcal{X}$. Then the REAPER objective at P_* satisfies

$$\sum_{x \in \mathcal{X}} \|x - P_*x\| \approx \sum_{x \in \mathcal{X}} \beta_x \|x - P_*x\|^2,$$

and so it seems plausible that the minimizer of the following quadratic program is close to P_* .

$$\text{minimize } \sum_{x \in \mathcal{X}} \beta_x \|x - Px\|^2 \quad \text{subject to } \mathbf{0} \preceq P \preceq \mathbf{I} \quad \text{and} \quad \text{tr } P = d. \quad (4.1)$$

We can efficiently solve problem (4.1) by performing a spectral computation and a water-filling step that ensures $\mathbf{0} \preceq P \preceq \mathbf{I}$. The water-filling step differentiates the new algorithm from the earlier work [80]. The details appear in a box labeled Algorithm 4.1, and a proof of correctness appears in Appendix C.1.

The heuristic above motivates an iterative procedure for solving (1.4). Let δ be a (small) positive regularization parameter. Initialize the iteration counter $k \leftarrow 0$ and the weights $\beta_x \leftarrow 1$ for each $x \in \mathcal{X}$. We solve (4.1) with the weights β_x to obtain a matrix $P^{(k)}$, and then we update the weights according to the formula

$$\beta_x \leftarrow \frac{1}{\max\{\delta, \|x - P^{(k)}x\|\}} \quad \text{for each } x \in \mathcal{X}.$$

In other words, we emphasize the observations that are explained well by the current model. The presence of the regularization parameter δ ensures that no single point can gain undue influence. We increment k , and we repeat the process until it has converged. See the box labeled Algorithm 4.2 for the details.

The following result shows that Algorithm 4.2 is guaranteed to converge to a point whose value is close to the optimal value of the REAPER problem (1.4).

Theorem 4.1 (Convergence of IRLS). *Assume that the set \mathcal{X} of observations does not lie in the union of two strict subspaces of \mathbb{R}^D . Then the iterates of Algorithm 4.2 with $\varepsilon = 0$ converge to a point \mathbf{P}_δ that satisfies the constraints of the REAPER problem (1.4). Moreover, the objective value at \mathbf{P}_δ satisfies the bound*

$$\sum_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x} - \mathbf{P}_\delta \mathbf{x}\| - \sum_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x} - \mathbf{P}_\star \mathbf{x}\| \leq \frac{1}{2} \delta |\mathcal{X}|,$$

where \mathbf{P}_\star is an optimal point of REAPER.

The proof of Theorem 4.1 is similar to established convergence arguments [80, Thms. 11 and 12], which follow the schema in [13, 71]. See Appendix C for a summary of the proof.

4.2 Computational Costs for Algorithm 4.2

Let us take a moment to summarize the computational costs for the IRLS method, Algorithm 4.2. When reading through this discussion, keep in mind that linear modeling problems

Algorithm 4.1 Solving the weighted least-squares problem (4.1)

INPUT:

- A dataset \mathcal{X} of observations in \mathbb{R}^D
- A nonnegative weight $\beta_{\mathbf{x}}$ for each $\mathbf{x} \in \mathcal{X}$
- The dimension parameter d in (4.1), where $d \in \{1, 2, \dots, D-1\}$

OUTPUT:

- A $D \times D$ matrix \mathbf{P}_\star that solves (4.1)

PROCEDURE:

- 1 Form the $D \times D$ weighted covariance matrix

$$\mathbf{C} \leftarrow \sum_{\mathbf{x} \in \mathcal{X}} \beta_{\mathbf{x}} \mathbf{x} \mathbf{x}^t$$

- 2 Compute an eigenvalue decomposition $\mathbf{C} = \mathbf{U} \cdot \text{diag}(\lambda_1, \dots, \lambda_D) \cdot \mathbf{U}^t$ with eigenvalues in nonincreasing order: $\lambda_1 \geq \dots \geq \lambda_D \geq 0$

- 3 **if** $\lambda_{d+1} = 0$ **then**

- a **for** $i = 1, \dots, D$ **do**

$$v_i \leftarrow \begin{cases} 1, & i \leq d \\ 0, & \text{otherwise} \end{cases}$$

- 4 **else**

- a **for** $i = d+1, \dots, D$ **do**
 - i Set

$$\theta \leftarrow \frac{i-d}{\sum_{k=1}^i \lambda_k^{-1}}$$

- ii **if** $\lambda_i > \theta \geq \lambda_{i+1}$ **then break for**

- b **for** $i = 1, \dots, D$ **do**

$$v_i \leftarrow \begin{cases} 1 - \frac{\theta}{\lambda_i}, & \lambda_i > \theta \\ 0, & \lambda_i \leq \theta \end{cases}$$

- 5 **return** $\mathbf{P}_\star := \mathbf{U} \cdot \text{diag}(v_1, \dots, v_D) \cdot \mathbf{U}^t$
-

typically involve datasets where the number N of data points is somewhat larger than the ambient dimension D .

The bulk of the computation in Algorithm 4.2 occurs when we solve the subproblem in Step 2b using the weighted-least squared method from Algorithm 4.1. The bulk of the computation in Algorithm 4.1 takes place during the spectral calculation in Steps 1 and 2. In general, we need $\mathcal{O}(ND^2)$ arithmetic operations to form the weighted covariance matrix, and the spectral calculation requires $\mathcal{O}(D^3)$. The remaining steps of both algorithms have lower order.

In summary, each iteration of Algorithm 4.2 requires $\mathcal{O}(ND^2)$ arithmetic operations. The algorithm converges linearly in practice, so we need $\mathcal{O}(\log(1/\eta))$ iterations to achieve an error of η .

In the statement of Algorithm 4.1, we have presented the weighted least-squared calculation in the most direct way possible. In practice, it is usually more efficient to form a $D \times N$ matrix \mathbf{W} with columns $\sqrt{\beta_x} \mathbf{x}$ for $\mathbf{x} \in \mathcal{X}$, to compute a thin SVD $\mathbf{W} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^t$, and to set $\mathbf{A} = \mathbf{\Sigma}^2$. This approach is also more stable. In some situations, such as when \mathbf{C} can be guaranteed to be low rank at each iteration, it is possible to accelerate the spectral calculations using randomized dimension reduction as in [35].

4.3 Empirical Convergence Rate of Algorithm 4.2

Many algorithms based on IRLS exhibit linear convergence [13]. Under some additional assumptions, we can prove that Algorithm 4.2 generates a sequence $\{\mathbf{P}^{(k)}\}$ of iterates that converges linearly to an optimal point \mathbf{P}_\star of REAPER. This argument has a small quotient of

Algorithm 4.2 IRLS algorithm for solving the REAPER problem (1.4)

INPUT:

- A dataset \mathcal{X} of observations in \mathbb{R}^D
- The dimension parameter d in (1.4), where $d \in \{1, 2, \dots, D-1\}$
- A regularization parameter $\delta > 0$
- A stopping tolerance $\varepsilon > 0$

OUTPUT:

- A $D \times D$ matrix \mathbf{P}_\star that satisfies $\mathbf{0} \preccurlyeq \mathbf{P}_\star \preccurlyeq \mathbf{I}$ and $\text{tr } \mathbf{P}_\star = d$

PROCEDURE:

- 1 Initialize the variables:
 - a Set the iteration counter $k \leftarrow 0$
 - b Set the initial error $\alpha^{(0)} \leftarrow +\infty$
 - c Set the weight $\beta_x \leftarrow 1$ for each $\mathbf{x} \in \mathcal{X}$
- 2 **do**
 - a Increment $k \leftarrow k + 1$
 - b Use Algorithm 4.1 to compute an optimal point $\mathbf{P}^{(k)}$ of (4.1) with weights β_x
 - c Let $\alpha^{(k)}$ be the optimal value of (4.1) at $\mathbf{P}^{(k)}$
 - d Update the weights:

$$\beta_x \leftarrow \frac{1}{\max\{\delta, \|\mathbf{x} - \mathbf{P}^{(k)}\mathbf{x}\|\}} \quad \text{for each } \mathbf{x} \in \mathcal{X}$$

until the objective fails to decrease: $\alpha^{(k)} \geq \alpha^{(k-1)} - \varepsilon$

- 3 **Return** $\mathbf{P}_\star = \mathbf{P}^{(k)}$
-

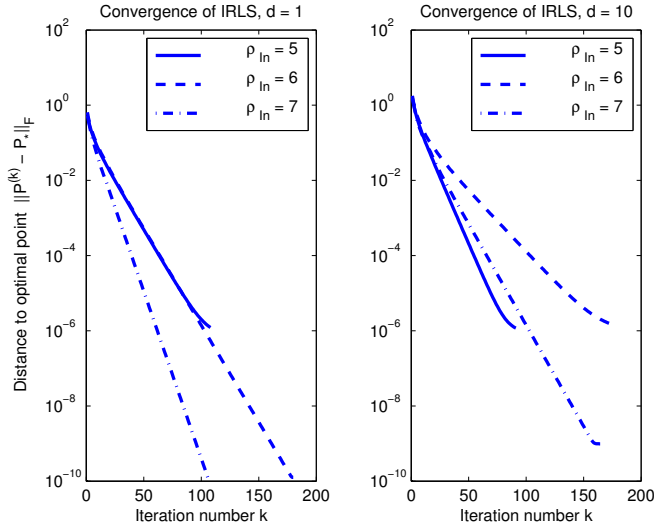


Fig. 4.1 *Convergence of IRLS to an optimal point.* The data are drawn from the Haystack Model on page 11 with ambient dimension $D = 100$ and $N_{\text{out}} = 200$ outliers. Each curve is associated with a particular choice of model dimension d and inlier sampling ratio $\rho_{\text{in}} = N_{\text{in}}/d$. We use Algorithm 4.2 to compute a sequence $\{\mathbf{P}^{(k)}\}$ of iterates, which we compare to an optimal point \mathbf{P}_* of the REAPER problem (1.4). See the text in Section 4.3 for more details of the experiment.

novelty relative to the amount of technical maneuver required, so we have chosen to omit the details. Our analysis does not provide a realistic estimate for the rate of convergence, so we have undertaken some numerical investigations to obtain more insight.

Figure 4.1 indicates that, empirically, Algorithm 4.2 does exhibit linear convergence. In this experiment, we have drawn the data from the Haystack Model on page 11 with ambient dimension $D = 100$ and $N_{\text{out}} = 200$ outliers. Each curve marks the performance of a single run of Algorithm 4.2 with a unique choice of the model dimension d and the inlier sampling ratio $\rho_{\text{in}} = N_{\text{in}}/d$. For this plot, we run Algorithm 4.2 with the regularization parameter $\delta = 10^{-10}$ and the error tolerance $\varepsilon = 10^{-15}$ to obtain a sequence $\{\mathbf{P}^{(k)}\}$ of iterates. We compare the computed iterates with an optimal point \mathbf{P}_* of the REAPER problem (1.4) obtained by solving REAPER with the Matlab package CVX [33, 34] at the highest-precision setting. The error is measured in Frobenius norm.

For synthetic data, the number of iterations required for Algorithm 4.2 seems to depend on the difficulty of the problem instance. Indeed, it may take as many as 200 iterations for the method to converge on challenging examples. In experiments with natural data, we usually obtain good performance after 20 iterations or so. In practice, Algorithm 4.2 is also much faster than algorithms [78, 56] for solving the low-leverage decomposition problem (6.2).

The stopping criterion in Algorithm 4.2 is motivated by the fact that the objective value must decrease in each iteration. This result is a consequence of the proof of Theorem 4.1; see (C.7). Taking ε on the order of machine precision ensures that the algorithm terminates when the iterates are dominated by numerical errors. In practice, we achieve very precise results when $\varepsilon = 10^{-15}$ or even $\varepsilon = 0$. In many applications, this degree of care is excessive, and we can obtain a reasonable solution for much larger values of ε .

5 Numerical Example: REAPER Applied to an Image Database

In this section, we present a numerical experiment that describes the performance of REAPER for a stylized problem involving natural data.

5.1 Some Practical Matters

Although the REAPER formulation is effective on its own, we can usually obtain better linear models if we preprocess the data before solving (1.4). Let us summarize the recommended procedure, which appears as Algorithm 5.1.

First, the REAPER problem assumes that the inliers are approximately centered. When they are not, it is important to identify a centering point c_* for the dataset and to work with the centered observations. We can compute a centering point c_* robustly by solving the Euclidean median problem [39, 55]:

$$\text{minimize} \quad \sum_{x \in \mathcal{X}} \|x - c\|. \quad (5.1)$$

It is also possible to incorporate centering by modifying the optimization problem (1.4). For brevity, we omit the details.

Second, the REAPER formulation can be sensitive to outliers with large magnitude. A simple but powerful method for addressing this challenge is to spherize the data points before solving the optimization problem. For future reference, we write down the resulting convex program.

$$\text{minimize} \quad \sum_{x \in \mathcal{X}} \|\tilde{x} - P\tilde{x}\| \quad \text{subject to} \quad \mathbf{0} \preceq P \preceq \mathbf{I} \quad \text{and} \quad \text{tr} P = d. \quad (5.2)$$

The tilde denotes the spherization transform (1.1). We refer to (5.2) as the S-REAPER problem. In most (but not all) of our experimental work, we have found that S-REAPER outperforms REAPER. The idea of spherizing data before fitting a subspace was proposed in the paper [51], where it is called *spherical PCA*.

Finally, we regard the parameter d in REAPER and S-REAPER as a proxy for the dimension of the linear model. While the rank of an optimal solution P_* to (1.4) or (5.2) cannot be smaller than d because of the constraints $\text{tr} P = d$ and $P \preceq \mathbf{I}$, the rank of P_* often exceeds d . We recommend solving (1.5) to find a closest projector to P_* .

Algorithm 5.1 *Prototype algorithm for robust computation of a linear model*

INPUT:

- A set \mathcal{X} of observations in \mathbb{R}^D
- The target dimension d for the linear model, where $d \in \{1, 2, \dots, D-1\}$

OUTPUT:

- A d -dimensional subspace of \mathbb{R}^D

PROCEDURE:

- 1 (Optional.) Solve (5.1) to obtain a center c_* , and center the data: $x \leftarrow x - c_*$ for each $x \in \mathcal{X}$
 - 2 (Optional.) Spherize the data: $x \leftarrow x / \|x\|$ for each nonzero $x \in \mathcal{X}$
 - 3 Solve the REAPER problem (1.4) with dataset \mathcal{X} and parameter d to obtain an optimal point P_*
 - 4 Solve (1.5) by finding a dominant d -dimensional invariant subspace of P_*
-

5.2 Experimental Setup

To solve the REAPER problem (1.4) and the S-REAPER problem (5.2), we use the IRLS method, Algorithm 4.2. We set the regularization parameter $\delta = 10^{-10}$ and the stopping tolerance $\varepsilon = 10^{-15}$. We postprocess the computed optimal point \mathbf{P}_\star of REAPER or S-REAPER to obtain a d -dimensional linear model by solving (1.5).

5.3 Comparisons

By now, there are a huge number of proposals for robust linear modeling, so we have limited our attention to methods that are computationally tractable. That is, we consider only formulations that have a polynomial-time algorithm for constructing a global solution (up to some tolerance). We do not discuss techniques that involve Monte Carlo simulation, nonlinear programming, etc. because the success of these approaches depends largely on parameter settings and providence. As a consequence, it is hard to evaluate their behavior in a consistent way.

We consider two standard approaches, PCA [41] and spherical PCA [51]. Spherical PCA rescales each observation so it lies on the Euclidean sphere, and then it applies standard PCA. Simulations performed with several different robust PCA methods in [53] lead Maronna et al. [55] to recommend spherical PCA as a reliable classical robust PCA algorithm.

We also consider a more recent proposal [77, 78, 56], which is called *low-leverage decomposition* (LLD) or *outlier pursuit*. This method decomposes the $D \times N$ matrix \mathbf{X} of observations by solving the optimization problem

$$\text{minimize } \|\mathbf{P}\|_{S_1} + \gamma \|\mathbf{C}\|_{1 \rightarrow 2}^* \quad \text{subject to } \mathbf{X} = \mathbf{P} + \mathbf{C} \quad (5.3)$$

where $\|\cdot\|_{S_1}$ is the Schatten 1-norm and $\|\cdot\|_{1 \rightarrow 2}^*$ returns the sum of Euclidean norms of the column. The idea is that the optimizer $(\mathbf{P}_\star, \mathbf{C}_\star)$ will consist of a low-rank model \mathbf{P}_\star for the data along with a column-sparse matrix \mathbf{C}_\star that identifies the outliers. We always use the parameter choice $\gamma = 0.8\sqrt{D/N}$, which seems to be effective in practice.

We do not make comparisons with the rank-sparsity decomposition [14], which has also been considered for robust linear modeling in [11]. It is not effective for the problem that we consider here.

5.4 Faces in a Crowd

This experiment is designed to test how well several robust methods are able to fit a linear model to face images that are dispersed in a collection of random images. Our setup allows us to study how well the robust model generalizes to faces we have not seen.

We pull 64 images of a single face under different illuminations from the Extended Yale Face Database B [45]. We use the first 32 faces for the sample, and we reserve the other 32 for the out-of-sample test. Next, we add all 467 images from the BACKGROUND/Google folder of the Caltech101 database [10, 30]. The Caltech101 images are converted to grayscale and downsampled to 192×168 pixels to match the native resolution of the Yale face images. We center the images by subtracting the Euclidean median (5.1). Then we apply PCA, spherical PCA, LLD, REAPER, and S-REAPER to fit a nine-dimensional subspace to the data. See [3] for justification of the choice $d = 9$. This experiment is similar to work reported in [50, Sec. VI].

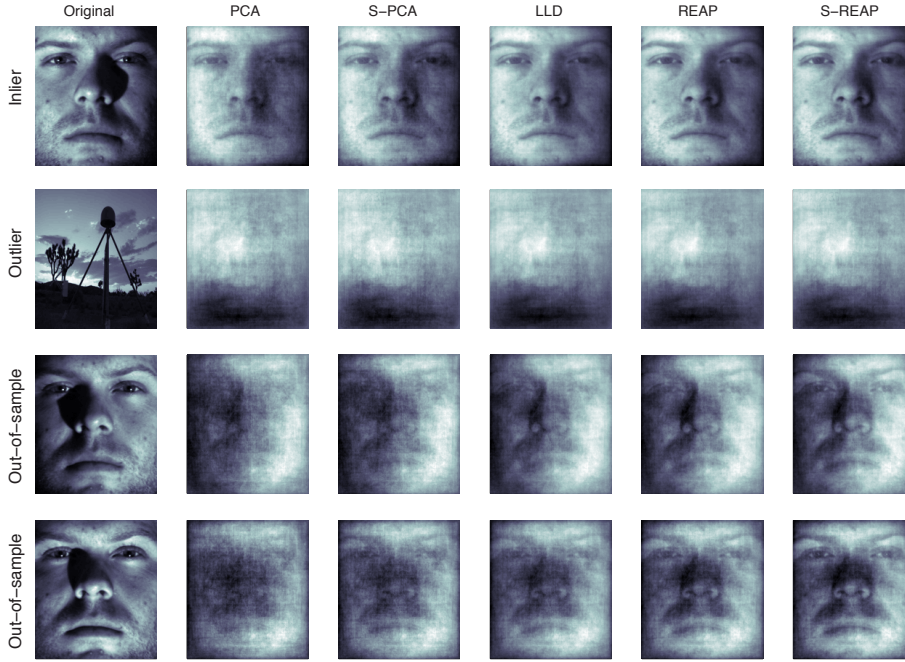


Fig. 5.1 Face images projected onto nine-dimensional linear model. The dataset consists of 32 images of a single face under different illuminations and 400 random images from the Caltech101 database. The original images (left column) are projected onto the nine-dimensional subspaces computed using five different modeling techniques. The first two rows indicate how well the models explain the in-sample faces versus the random images. The last two rows show projections of two out-of-sample faces, which were not used to compute the linear models. See Section 5.4 for details.

Figure 5.1 displays several images from the sample projected onto the computed nine-dimensional subspace (with the centering added back after projection). For every method, the projection of an in-sample face image onto the subspace is recognizable as a face. Meanwhile, the out-of-sample faces are described poorly by the PCA subspace. All of the robust subspaces capture the facial features better, with S-REAPER producing the clearest images.

Figure 5.2 shows the ordered distances of the 32 out-of-sample faces to the robust linear model as a function of the ordered distances to the model computed with PCA. A point below the 1:1 line means that the i th closest point is closer to the robust model than the i th closest point is to the PCA model. Under this metric, S-REAPER is the dominant method, which explains the qualitative behavior seen in Figure 5.1. This plot clearly demonstrates that S-REAPER computes a subspace that generalizes better than the subspaces obtained with the other robust methods.

6 Related Work

Robust linear modeling has been an active subject of research for over three decades. Although many classical approaches have strong robustness properties *in theory*, the proposals usually involve either intractable computational problems or algorithms that are designed for a different problem like robust covariance estimation. More recently, researchers have devel-

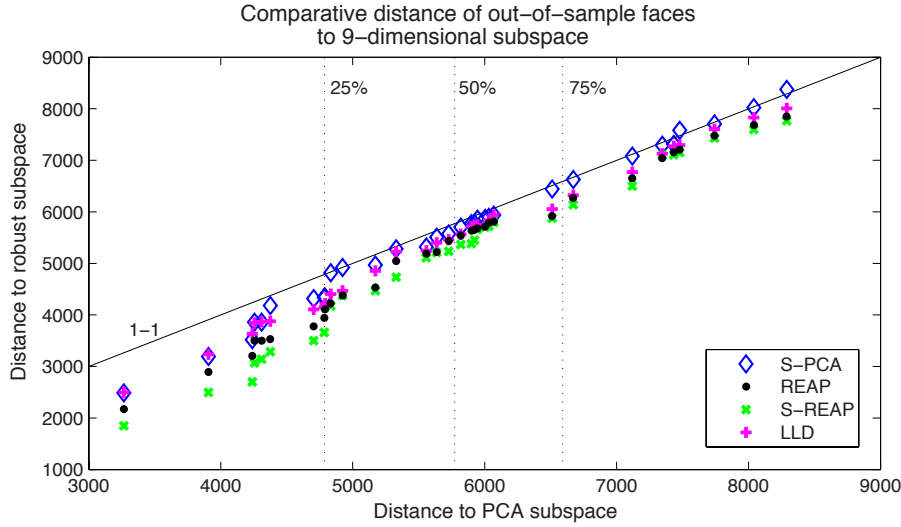


Fig. 5.2 Approximation of out-of-sample face images by several linear models. The ordered distances of the out-of-sample face images to each robust model as a function of the ordered distances to the PCA model. The model was computed from 32 in-sample images; this graph shows how the model generalizes to the 32 out-of-sample images. Lower is better. See Section 5.4 for details.

oped several techniques, based on convex optimization, that are computationally efficient and admit some theoretical guarantees. In this section, we summarize classical and contemporary work on robust linear modeling, with a focus on the numerical aspects. We recommend the books [39, 55, 64] for a comprehensive discussion of robust statistics.

6.1 Classical Strategies for Robust Linear Modeling

We begin with an overview of the major techniques that have been proposed in the statistics literature. The theoretical contributions in this area focus on breakdown points and influence functions of estimators. Researchers tend to devote less attention to the computational challenges inherent in these formulations. Moreover, the notion of the breakdown point for quantifying the robustness of estimators for vectors and matrices does not generalize to subspace estimation. We recall that, roughly speaking, the breakdown point measures the proportion of arbitrarily placed outliers an estimator can handle before giving an arbitrarily bad result. However, this idea does not directly extend to subspaces since the set of subspaces is compact. Following instead [11, 78], we quantify robustness of subspace estimation via exact recovery and stability to noise.

6.1.1 Robust Combination of Residuals

Historically, one of the earliest approaches to linear regression is to minimize the sum of (nonorthogonal) residuals. This is the principle of *least absolute deviations* (LAD). Early proponents of this idea include Galileo, Boscovich, Laplace, and Edgeworth. See [36, 37, 26] for some historical discussion. It appears that *orthogonal* regression with LAD was first considered in the late 1980s [59, 65, 58]; the extension from orthogonal regression to PCA

seems to be even more recent [73, 25]. LAD has also been considered as a method for hybrid linear modeling in [81, 48]. We are not aware of a tractable algorithm for these formulations.

There are many other robust methods for combining residuals aside from LAD. An approach that has received wide attention is to minimize the median of the squared residuals [63, 64]. Other methods appear in the books [39, 55]. These formulations are generally not computationally tractable.

6.1.2 Robust Estimation of Covariance Matrix

Another standard technique for robust PCA is to form a robust estimate of the covariance matrix of the data [54, 39, 55, 24, 22, 79, 19]. The classical approaches to robust estimates of covariance are based on maximum likelihood principles that lead to M-estimators. Most often, IRLS algorithms are used to compute these M-estimators of covariance.

There are some formal similarities between the minimization program (1.4) and the formulation of classical M-estimators. In particular, the computational complexity of these estimators scales comparably with our own IRLS algorithm. However, there are also important differences between the classical covariance estimators and the subspace estimation procedure we consider here. In particular, the common M-estimators for robust covariance estimation fail for the exact- and near-subspace recovery problems. See [80, Sec. 3.1] for elaboration on these points.

In addition to the basic M-estimators computed with IRLS, there are many other covariance estimators, including S-estimators, the minimum covariance determinant (MCD), the minimum volume ellipsoid (MVE), and the Stahel–Donoho estimator. We are not aware of any scalable algorithm with guarantees of correctness for implementing these latter estimators. See [55, Sec. 6] for a review.

6.1.3 Projection Pursuit PCA

Projection pursuit (often abbreviated PP-PCA) is a procedure that constructs principal components one at a time by finding a direction that maximizes a robust measure of scale, removing the component of the data in this direction, and repeating the process. The initial proposal appears in [39, 1st edn.], and it has been explored by many other authors [49, 1, 18, 43, 75]. We are aware of only one formulation that provably (approximately) maximizes a robust measure of scale at each iteration [56], but there are no overall guarantees for PP-PCA algorithms.

6.1.4 Screening for Outliers

Another common approach is to remove possible outliers and then estimate the underlying subspace by PCA [15, 66, 67]. The classical methods offer very limited guarantees. There are some recent algorithms that are provably correct [9, 76] under some model assumptions and with particular correctness criteria that are tailored to the individual algorithm.

6.1.5 RANSAC

The randomized sample consensus (RANSAC) method is a randomized iterative procedure for fitting models to noisy data consisting of inliers and outliers [31]. Under some assumptions, RANSAC will eventually identify a linear model for the inliers, but there are no guarantees on the number of iterations required.

6.1.6 Spherical PCA

A useful method for fitting a robust linear model is to center the data robustly, project it onto a sphere, and then apply standard PCA. This approach is due to [51]. Maronna et al. [55] recommend it as a preferred method for robust PCA. The technique is computationally practical, but it has limited theoretical guarantees.

6.2 Approaches Based on Convex Optimization

Recently, researchers have started to develop effective techniques for robust linear modeling that are based on convex optimization. These formulations invite a variety of tractable algorithms, and they have theoretical guarantees under appropriate model assumptions.

6.2.1 Demixing Methods

One class of techniques for robust linear modeling is based on splitting a data matrix into a low-rank model plus a corruption. The first approach of this form is due to Chandrasekaran et al. [14]. Given an observed matrix \mathbf{X} , they solve the semidefinite problem

$$\text{minimize } \|\mathbf{P}\|_{S_1} + \gamma \|\mathbf{C}\|_{\ell_1} \quad \text{subject to } \mathbf{X} = \mathbf{P} + \mathbf{C}. \quad (6.1)$$

Minimizing the Schatten 1-norm $\|\cdot\|_{S_1}$ promotes low rank, while minimizing the vector ℓ_1 norm promotes sparsity. The regularization parameter γ negotiates a tradeoff between the two goals. Candès et al. [11] study the performance of (6.1) for robust linear modeling in the setting where individual entries of the matrix \mathbf{X} are subject to error.

A related proposal is due to Xu et al. [77, 78] and independently to McCoy & Tropp [56]. These authors recommend solving the decomposition problem

$$\text{minimize } \|\mathbf{P}\|_{S_1} + \gamma \|\mathbf{C}\|_{1 \rightarrow 2}^* \quad \text{subject to } \mathbf{X} = \mathbf{P} + \mathbf{C}. \quad (6.2)$$

The norm $\|\cdot\|_{1 \rightarrow 2}^*$ returns the sum of Euclidean norms of the columns of its argument. This formulation is appropriate for inlier-outlier data models, where entire columns of the data matrix may be corrupted, in contrast to the formulation (6.1) that is used for corruptions of individual matrix elements.

Both (6.1) and (6.2) possess some theoretical guarantees under appropriate model assumptions, but we restrict our discussion to (6.2) because it is tuned to the In & Out Model that we consider here. In the noiseless case, Xu et al. [78] show that (6.2) will exactly recover the underlying subspace under the In & Out Model so long as the inlier-to-outlier ratio exceeds a constant times the inlier dimension d .⁴ For the Haystack Model with $\sigma_{\text{in}} = \sigma_{\text{out}}$ and $d \ll D$, the lower bound (3.1) is positive when the fraction of inliers exceeds a constant times d/D . Hence, Theorem 2.1 endows REAPER with an exact recovery guarantee that is stronger than the results of [78] for (6.2) in the $d \ll D$ regime; a similar statement holds for the stability of REAPER. Moreover, the work of Coudron & Lerman [17], which appeared after the submission of this work, provides sample-complexity guarantees for REAPER that mirrors that of standard PCA when the data \mathcal{X} is drawn i.i.d. from a subgaussian distribution.

The most common algorithmic framework for demixing methods of the form (6.1) and (6.2) uses the alternating direction method of multipliers (ADMM) [8]. These algorithms

⁴ More precisely, the inlier-to-outlier ratio must exceed $(121\mu/9)d$, where $\mu \geq 1$ depends on the data.

can converge slowly, so it may take excessive computation to obtain a high-accuracy solution. Indeed, our limited numerical experiments indicate that REAPER tends to be significantly faster than the ADMM implementation of [77, 78] and [56].

Nevertheless, the demixing strategy readily adapts to different situations such as missing observations or entrywise corruptions [78, 11], while it is not immediately clear how to adapt the REAPER framework to these scenarios.

6.2.2 Precedents for the REAPER Problem

The REAPER problem (1.4) is a semidefinite relaxation of the ℓ_1 orthogonal distance problem (1.3). Our work extends an earlier relaxation of (1.3) proposed by Zhang & Lerman [80]:

$$\text{minimize } \sum_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x} - \mathbf{P}\mathbf{x}\| \quad \text{subject to} \quad \text{tr}(\mathbf{I} - \mathbf{P}) = 1, \quad (6.3)$$

where the minimum occurs over symmetric \mathbf{P} . Although not obvious, the formulation above is equivalent to REAPER with the specific choice $d = D - 1$. Indeed, any optimal point \mathbf{P}_\star of (6.3) satisfies $\mathbf{I} - \mathbf{P}_\star \succcurlyeq \mathbf{0}$ [80, Lem. 14], and this fact, together with the trace constraint $\text{tr}(\mathbf{I} - \mathbf{P}_\star) = 1$, implies that $\mathbf{I} - \mathbf{P}_\star \preccurlyeq \mathbf{I}$. Thus, the REAPER constraints $\mathbf{0} \preccurlyeq \mathbf{P} \preccurlyeq \mathbf{I}$ are implicit in (6.3), and so the observation $\text{tr}(\mathbf{I}) = D$ yields the claimed equivalence.

The present work extends the earlier formulation by freeing the parameter d to search for subspaces of a specific dimension, which provides a tighter relaxation for finding d -dimensional orthoprojectors. In [80], however, the authors show that the optimal point \mathbf{P}_\star of (6.3) is more analogous to a robust inverse covariance matrix than to an approximate orthoprojector. This allows the determination of the dimension d using the eigenvalues of \mathbf{P}_\star , while in the present work, we treat d as a known parameter.

Our analysis of REAPER builds on the ideas first presented in [47, 80], but it incorporates a number of refinements that simplify and improve the theoretical guarantees. In particular, the present results do not require an oracle condition like [80, Eqs. (8,9)], and our stability statistic $\mathcal{S}(L)$ supersedes the earlier exact recovery and stability requirements [80, Eqs. (6,7) & (10,11)]. The exact recovery guarantees under the Haystack Model are somewhat stronger for REAPER than the analogous guarantees for (6.3) [80, Sec. 2.6.1]. The IRLS algorithm for REAPER and the convergence analysis that we present in Section 4 also extend ideas from the earlier work.

From a broad perspective, the idea of relaxing a difficult nonconvex program like (1.3) to obtain a convex problem is well established in the literature on combinatorial optimization. Research on linear programming relaxations is summarized in [70]. Some significant works on semidefinite relaxation include [52, 32].

A Theorem 2.1: Supporting lemmas

This appendix contains the proofs of the two technical results that animate Theorem 2.1. Throughout, we maintain the notation and assumptions from the In & Out Model on page 6 and from the statement and proof of Theorem 2.1.

A.1 Controlling the Size of the Perturbation

In this section, we establish Lemma 2.2. On comparing the objective function f from (2.6) with the perturbed objective function g from (2.9), we see that the only changes involve the terms containing inliers. As a

consequence, the difference function $h = f - g$ depends only on the inliers:

$$h(\mathbf{P}) = \sum_{\mathbf{x} \in \mathcal{X}_{\text{in}}} [\|(\mathbf{I} - \mathbf{P})\mathbf{x}\| - \|(\mathbf{I} - \mathbf{P})\mathbf{\Pi}_L\mathbf{x}\|],$$

where \mathbf{P} is an arbitrary matrix. Apply the lower triangle inequality to see that

$$\begin{aligned} |h(\mathbf{P})| &\leq \sum_{\mathbf{x} \in \mathcal{X}_{\text{in}}} \|(\mathbf{I} - \mathbf{P})(\mathbf{I} - \mathbf{\Pi}_L)\mathbf{x}\| \\ &\leq \sum_{\mathbf{x} \in \mathcal{X}_{\text{in}}} [\|(\mathbf{I} - \mathbf{\Pi}_L)\mathbf{x}\| + \|(\mathbf{\Pi}_L - \mathbf{P})(\mathbf{I} - \mathbf{\Pi}_L)\mathbf{x}\|] \\ &\leq [1 + \|\mathbf{\Pi}_L - \mathbf{P}\|] \cdot \sum_{\mathbf{x} \in \mathcal{X}_{\text{in}}} \|(\mathbf{I} - \mathbf{\Pi}_L)\mathbf{x}\| \\ &\leq [1 + \|\mathbf{\Pi}_L - \mathbf{P}\|_{S_1}] \cdot \mathcal{R}(L). \end{aligned} \quad (\text{A.1})$$

To justify the second inequality, write $\mathbf{I} - \mathbf{P} = (\mathbf{I} - \mathbf{\Pi}_L) + (\mathbf{\Pi}_L - \mathbf{P})$, and then apply the upper triangle inequality. Use the fact that the projector $\mathbf{I} - \mathbf{\Pi}_L$ is idempotent to simplify the resulting expression. The third inequality depends on the usual operator-norm bound. We reach (A.1) by identifying the sum as the total inlier residual $\mathcal{R}(L)$, defined in (2.2), and invoking the fact that the Schatten 1-norm dominates the operator norm.

Applying (A.1) twice, it becomes apparent that

$$|h(\mathbf{\Pi}_L) - h(\mathbf{\Pi}_L + \mathbf{\Delta})| \leq |h(\mathbf{\Pi}_L)| + |h(\mathbf{\Pi}_L + \mathbf{\Delta})| \leq [2 + \|\mathbf{\Delta}\|_{S_1}] \mathcal{R}(L).$$

There are no restrictions on the matrix $\mathbf{\Delta}$, so the demonstration of Lemma 2.2 is complete.

A.2 The Rate of Ascent of the Perturbed Objective

In this section, we establish Lemma 2.3. This result contains the essential insights behind Theorem 2.1, and the proof involves some amount of exertion.

Assume that $\mathbf{\Pi}_L + \mathbf{\Delta} \in \Phi$. Since the perturbed objective g defined in (2.9) is a continuous convex function, we have the lower bound

$$g(\mathbf{\Pi}_L + \mathbf{\Delta}) - g(\mathbf{\Pi}_L) \geq g'(\mathbf{\Pi}_L; \mathbf{\Delta}). \quad (\text{A.2})$$

The right-hand side of (A.2) refers to the one-sided directional derivative of g at the point $\mathbf{\Pi}_L$ in the direction $\mathbf{\Delta}$. That is,

$$g'(\mathbf{\Pi}_L; \mathbf{\Delta}) = \lim_{t \downarrow 0} \frac{1}{t} [g(\mathbf{\Pi}_L + t\mathbf{\Delta}) - g(\mathbf{\Pi}_L)].$$

We can be confident that this limit exists and takes a finite value [62, Thm. 23.1 et seq.].

Let us find a practicable expression for the directional derivative. Recalling the definition (2.9) of the perturbed objective, we see that the difference quotient takes the form

$$\begin{aligned} &\frac{1}{t} [g(\mathbf{\Pi}_L + t\mathbf{\Delta}) - g(\mathbf{\Pi}_L)] \\ &= \sum_{\mathbf{x} \in \mathcal{X}_{\text{in}}} \frac{1}{t} [\|(\mathbf{\Pi}_{L^\perp} - t\mathbf{\Delta})\mathbf{\Pi}_L\mathbf{x}\| - \|\mathbf{\Pi}_{L^\perp}\mathbf{\Pi}_L\mathbf{x}\|] + \sum_{\mathbf{x} \in \mathcal{X}_{\text{out}}} \frac{1}{t} [\|(\mathbf{\Pi}_{L^\perp} - t\mathbf{\Delta})\mathbf{x}\| - \|\mathbf{\Pi}_{L^\perp}\mathbf{x}\|]. \end{aligned} \quad (\text{A.3})$$

We analyze the two sums separately. First, consider the terms that involve inliers. For $\mathbf{x} \in \mathcal{X}_{\text{in}}$,

$$\frac{1}{t} [\|(\mathbf{\Pi}_{L^\perp} - t\mathbf{\Delta})\mathbf{\Pi}_L\mathbf{x}\| - \|\mathbf{\Pi}_{L^\perp}\mathbf{\Pi}_L\mathbf{x}\|] = \|\mathbf{\Delta}\mathbf{\Pi}_L\mathbf{x}\| \quad \text{for all } t > 0.$$

We have used the fact that $\mathbf{\Pi}_{L^\perp}\mathbf{\Pi}_L = \mathbf{0}$ twice. Next, consider the terms involving outliers. By the assumptions of the In & Out Model on page 6, each outlier $\mathbf{x} \in \mathcal{X}_{\text{out}}$ has a nontrivial component in the subspace L^\perp . We may calculate that

$$\begin{aligned} \|(\mathbf{\Pi}_{L^\perp} - t\mathbf{\Delta})\mathbf{x}\| &= [\|\mathbf{\Pi}_{L^\perp}\mathbf{x}\|^2 - 2\langle t\mathbf{\Delta}\mathbf{x}, \mathbf{\Pi}_{L^\perp}\mathbf{x} \rangle + \|t\mathbf{\Delta}\mathbf{x}\|^2]^{1/2} \\ &= \|\mathbf{\Pi}_{L^\perp}\mathbf{x}\| \left[1 - \frac{2t}{\|\mathbf{\Pi}_{L^\perp}\mathbf{x}\|} \langle \mathbf{\Delta}\mathbf{x}, \widetilde{\mathbf{\Pi}_{L^\perp}\mathbf{x}} \rangle + \mathcal{O}(t^2) \right]^{1/2} \\ &= \|\mathbf{\Pi}_{L^\perp}\mathbf{x}\| - t \langle \mathbf{\Delta}\mathbf{x}, \widetilde{\mathbf{\Pi}_{L^\perp}\mathbf{x}} \rangle + \mathcal{O}(t^2) \quad \text{as } t \downarrow 0, \end{aligned}$$

where the spherization transform $\mathbf{x} \mapsto \tilde{\mathbf{x}}$ is defined in (1.1). Therefore,

$$\frac{1}{t} [\|(\Pi_{L^\perp} - t\Delta)\mathbf{x}\| - \|\Pi_{L^\perp}\mathbf{x}\|] \rightarrow -\langle \Delta\mathbf{x}, \widetilde{\Pi_{L^\perp}\mathbf{x}} \rangle \quad \text{as } t \downarrow 0.$$

Introducing these facts into the difference quotient (A.3) and taking the limit as $t \downarrow 0$, we determine that

$$g'(\Pi_L; \Delta) = \sum_{\mathbf{x} \in \mathcal{X}_{\text{in}}} \|\Delta \Pi_L \mathbf{x}\| - \sum_{\mathbf{x} \in \mathcal{X}_{\text{out}}} \langle \Delta\mathbf{x}, \widetilde{\Pi_{L^\perp}\mathbf{x}} \rangle. \quad (\text{A.4})$$

It remains to produce a lower bound on this directional derivative.

The proof of the lower bound has two components. We require a bound on the sum over outliers, and we require a bound on the sum over inliers. These results appear in the following sublemmas, which we prove in Sections A.2.1 and A.2.2.

Sublemma A.1 (Outliers). *Under the prevailing assumptions,*

$$\sum_{\mathbf{x} \in \mathcal{X}_{\text{out}}} \langle \Delta\mathbf{x}, \widetilde{\Pi_{L^\perp}\mathbf{x}} \rangle \leq \|\Pi_{L^\perp} \widetilde{\mathbf{X}_{\text{out}}}\| \cdot \|\mathbf{X}_{\text{out}}\| \cdot \|\Delta\|_{S_1} = \mathcal{A}(L) \cdot \|\Delta\|_{S_1},$$

where the alignment statistic $\mathcal{A}(L)$ is defined in (2.3).

Sublemma A.2 (Inliers). *Under the prevailing assumptions,*

$$\sum_{\mathbf{x} \in \mathcal{X}_{\text{in}}} \|\Delta \Pi_L \mathbf{x}\| \geq \left[\frac{1}{4\sqrt{d}} \cdot \inf_{\substack{\mathbf{u} \in L \\ \|\mathbf{u}\|=1}} \sum_{\mathbf{x} \in \mathcal{X}_{\text{in}}} |\langle \mathbf{u}, \mathbf{x} \rangle| \right] \cdot \|\Delta\|_{S_1} = \frac{\mathcal{P}(L)}{4\sqrt{d}} \cdot \|\Delta\|_{S_1},$$

where the permeance statistic $\mathcal{P}(L)$ is defined in (2.1).

To complete the proof of Lemma 2.3, we substitute the bounds from Sublemmas A.1 and A.2 into the expression (A.4) for the directional derivative. This step yields

$$g'(\Pi_L; \Delta) \geq \left[\frac{\mathcal{P}(L)}{4\sqrt{d}} - \mathcal{A}(L) \right] \cdot \|\Delta\|_{S_1}.$$

Since $d = \dim(L)$, we identify the bracket as the stability statistic $\mathcal{S}(L)$ defined in (2.4). Combine this bound with (A.2) to finish the argument.

A.2.1 Outliers

The result in Sublemma A.1 is straightforward. First, observe that

$$\sum_{\mathbf{x} \in \mathcal{X}_{\text{out}}} \langle \Delta\mathbf{x}, \widetilde{\Pi_{L^\perp}\mathbf{x}} \rangle = \sum_{\mathbf{x} \in \mathcal{X}_{\text{out}}} \langle \Delta, \widetilde{\Pi_{L^\perp}\mathbf{x}} \cdot \mathbf{x}^t \rangle = \langle \Delta, \Pi_{L^\perp} \widetilde{\mathbf{X}_{\text{out}}} \cdot \mathbf{X}_{\text{out}}^t \rangle.$$

The first relation follows from the fact that $\langle \mathbf{A}\mathbf{b}, \mathbf{c} \rangle = \langle \mathbf{A}, \mathbf{c}\mathbf{b}^t \rangle$. We obtain the second identity by drawing the sum into the inner product and recognizing the product of the two matrices. Therefore,

$$\left| \sum_{\mathbf{x} \in \mathcal{X}_{\text{out}}} \langle \Delta\mathbf{x}, \widetilde{\Pi_{L^\perp}\mathbf{x}} \rangle \right| \leq \|\Pi_{L^\perp} \widetilde{\mathbf{X}_{\text{out}}} \cdot \mathbf{X}_{\text{out}}^t\| \cdot \|\Delta\|_{S_1}.$$

The last bound is an immediate consequence of the duality between the spectral norm and the Schatten 1-norm. To complete the argument, we apply the operator norm bound $\|\mathbf{A}\mathbf{B}^t\| \leq \|\mathbf{A}\| \|\mathbf{B}\|$.

A.2.2 Inliers

The proof of Sublemma A.2 involves several considerations. We first explain the overall structure of the argument and then verify that each part is correct. The first ingredient is a lower bound for the minimum of the sum over inliers:

$$\sum_{\mathbf{x} \in \mathcal{J}_{\text{in}}} \|\Delta \Pi_L \mathbf{x}\| \geq \left[\inf_{\substack{\mathbf{u} \in L \\ \|\mathbf{u}\|=1}} \sum_{\mathbf{x} \in \mathcal{J}_{\text{in}}} |\langle \mathbf{u}, \mathbf{x} \rangle| \right] \cdot \|\Delta \Pi_L\|_{\text{F}}. \quad (\text{A.5})$$

See Section A.2.3. The second step depends on a simple comparison between the Frobenius norm and the Schatten 1-norm of a matrix:

$$\|\Delta \Pi_L\|_{\text{F}} \geq \frac{1}{\sqrt{d}} \|\Delta \Pi_L\|_{S_1}. \quad (\text{A.6})$$

The inequality (A.6) follows immediately when we express the two norms in terms of singular values and apply the fact that $\Delta \Pi_L$ has rank d ; it requires no further comment. Third, we argue that

$$\|\Delta \Pi_L\|_{S_1} \geq \frac{1}{4} \|\Delta\|_{S_1} \quad \text{when } \Pi_L + \Delta \in \Phi. \quad (\text{A.7})$$

This demonstration appears in Section A.2.4. Combining the bounds (A.5), (A.6), and (A.7), we obtain the result stated in Sublemma A.2

A.2.3 Minimization with a Frobenius-norm constraint

To establish (A.5), we assume that $\|\Delta \Pi_L\|_{\text{F}} = 1$. The general case follows from homogeneity.

Let us introduce a (thin) singular value decomposition $\Delta \Pi_L = U \Sigma V^t$. Observe that each column $\mathbf{v}_1, \dots, \mathbf{v}_d$ of the matrix V is contained in L . In addition, the singular values σ_j satisfy $\sum_{j=1}^d \sigma_j^2 = 1$ because of the normalization of the matrix.

We can express the quantity of interest as

$$\sum_{\mathbf{x} \in \mathcal{J}_{\text{in}}} \|\Delta \Pi_L \mathbf{x}\| = \sum_{\mathbf{x} \in \mathcal{J}_{\text{in}}} \|\Sigma V^t \mathbf{x}\| = \sum_{\mathbf{x} \in \mathcal{J}_{\text{in}}} \left[\sum_{j=1}^d \sigma_j^2 |\langle \mathbf{v}_j, \mathbf{x} \rangle|^2 \right]^{1/2}$$

The first identity follows from unitary invariance of the Euclidean norm. In the second relation, we have just written out the Euclidean norm in detail. To facilitate the next step, abbreviate $p_j = \sigma_j^2$ for each index j . Calculate that

$$\begin{aligned} \sum_{\mathbf{x} \in \mathcal{J}_{\text{in}}} \|\Delta \Pi_L \mathbf{x}\| &= \sum_{\mathbf{x} \in \mathcal{J}_{\text{in}}} \left[\sum_{j=1}^d p_j |\langle \mathbf{v}_j, \mathbf{x} \rangle|^2 \right]^{1/2} \\ &\geq \min_j \sum_{\mathbf{x} \in \mathcal{J}_{\text{in}}} |\langle \mathbf{v}_j, \mathbf{x} \rangle| \\ &\geq \inf_{\substack{\mathbf{u} \in L \\ \|\mathbf{u}\|=1}} \sum_{\mathbf{x} \in \mathcal{J}_{\text{in}}} |\langle \mathbf{u}, \mathbf{x} \rangle|. \end{aligned}$$

Indeed, the right-hand side of the first line is a concave function of the variables p_j . By construction, these variables lie in the convex set determined by the constraints $p_j \geq 0$ and $\sum_j p_j = 1$. The minimizer of a concave function over a convex set occurs at an extreme point, which delivers the first inequality. To reach the last inequality, recall that each \mathbf{v}_j is a unit vector in L . This expression implies (A.5).

A.2.4 Feasible directions

Finally, we need to verify the relation (A.7), which states that $\|\Delta \Pi_L\|_{S_1}$ is comparable with $\|\Delta\|_{S_1}$ provided that $\Pi_L + \Delta \in \Phi$. To that end, we decompose the matrix Δ into blocks:

$$\Delta = \underbrace{\Pi_L \Delta \Pi_L}_{=: \Delta_1} + \underbrace{\Pi_L^\perp \Delta \Pi_L}_{=: \Delta_2} + \underbrace{\Pi_L \Delta \Pi_L^\perp}_{=: \Delta_3} + \underbrace{\Pi_L^\perp \Delta \Pi_L^\perp}_{=: \Delta_4}. \quad (\text{A.8})$$

We claim that

$$\|\Delta_1\|_{S_1} = \|\Delta_3\|_{S_1} \quad \text{whenever } \Pi_L + \Delta \in \Phi. \quad (\text{A.9})$$

Granted this identity, we can establish the equivalence of norms promised in (A.7). Indeed,

$$\begin{aligned} \|\Delta \Pi_L\|_{S_1} &\geq \max \{ \|\Delta_1\|_{S_1}, \|\Delta_2\|_{S_1} \} \\ &\geq \frac{1}{4} [2\|\Delta_1\|_{S_1} + 2\|\Delta_2\|_{S_1}] \\ &= \frac{1}{4} [\|\Delta_1\|_{S_1} + \|\Delta_3\|_{S_1} + \|\Delta_2\|_{S_1} + \|\Delta_2^\dagger\|_{S_1}] \\ &\geq \frac{1}{4} \|\Delta\|_{S_1}. \end{aligned}$$

The first inequality holds because projection reduces the Schatten 1-norm of a matrix. The second inequality is numerical. The equality in the third line depends on the claim (A.9). The last bound follows from the subadditivity of the norm and (A.8).

To check (A.9), we recall the definition of the feasible set:

$$\Phi = \{P : \mathbf{0} \preceq P \preceq \mathbf{I} \quad \text{and} \quad \text{tr} P = d\}.$$

The condition $\Pi_L + \Delta \in \Phi$ implies the semidefinite relation $\Pi_L + \Delta \preceq \mathbf{I}$. Conjugating by the orthoprojector Π_L , we see that

$$\Pi_L + \Delta_1 = \Pi_L(\Pi_L + \Delta)\Pi_L \preceq \Pi_L.$$

As a consequence, $\Delta_1 \preceq \mathbf{0}$. Similarly, the relation $\mathbf{0} \preceq \Pi_L + \Delta$ yields

$$\mathbf{0} \preceq \Pi_{L^\perp}(\Pi_L + \Delta)\Pi_{L^\perp} = \Delta_3.$$

Therefore, $\Delta_3 \succeq \mathbf{0}$. Since $\text{tr}(\Pi_L + \Delta) = d$ and $\text{tr} \Pi_L = d$, it is clear that $0 = \text{tr} \Delta = \text{tr}(\Delta_1 + \Delta_3)$. We conclude that

$$\|\Delta_1\|_{S_1} = \text{tr}(-\Delta_1) = \text{tr}(\Delta_3) = \|\Delta_3\|_{S_1}.$$

The first and third equality hold because the Schatten 1-norm of a positive-semidefinite matrix coincides with its trace.

B Analysis of the Haystack Model

In this appendix, we establish exact recovery conditions for the Haystack Model. To accomplish this goal, we study the probabilistic behavior of the permeance statistic and the alignment statistic. Our main result for the Haystack Model, Theorem B.1, follows when we introduce the probability bounds into the deterministic recovery result, Theorem 2.1. The simplified result for the Haystack Model, Theorem 3.1, is a consequence of the following more detailed theory.

Theorem B.1. *Fix a parameter $c > 0$. Let L be an arbitrary d -dimensional subspace of \mathbb{R}^D , and draw the dataset \mathcal{X} at random according to the Haystack Model on page 11. Let $1 \leq d \leq D-1$. The stability statistic satisfies*

$$\mathcal{S}(L) \geq \frac{\sigma_{\text{in}}}{\sqrt{8\pi}} \left[\rho_{\text{in}} - \frac{(2+c)}{\sqrt{2/\pi}} \sqrt{\rho_{\text{in}}} \right] - \sigma_{\text{out}} \sqrt{\frac{D}{D-d-0.5}} \left[\sqrt{\rho_{\text{out}}} + 1 + c \sqrt{\frac{d}{D}} \right]^2 \quad (\text{B.1})$$

except with probability $3.5e^{-c^2 d/2}$.

We verify this expression below in Sections B.1 and B.2. Now, we demonstrate that Theorem B.1 contains the simplified result for the Haystack model, Theorem 3.1.

Proof of Theorem 3.1 from Theorem B.1. To begin, we collect some numerical inequalities. For $\alpha > 0$, the function $f(x) = x - \alpha\sqrt{x}$ is convex when $x \geq 0$, so that

$$x - \alpha\sqrt{x} = f(x) \geq f(\alpha^2) + f'(\alpha^2)(x - \alpha^2) = \frac{1}{2}(x - \alpha^2).$$

For $1 \leq d \leq (D-1)/2$, we have the numerical bounds

$$\frac{D}{D-d-0.5} \leq 2 \quad \text{and} \quad \frac{d}{D} \leq 0.5.$$

Finally, recall that $(a+b)^2 \leq 2(a^2+b^2)$ and $(a+b+e)^2 \leq 3(a^2+b^2+e^2)$ as a consequence of Hölder's inequality.

To prove Theorem 3.1, we apply our numerical inequalities to weaken the bound (B.1) from Theorem B.1 to

$$\mathcal{S}(L) \geq \frac{\sigma_{\text{in}}}{\sqrt{32\pi}} [\rho_{\text{in}} - \pi(4+c^2)] - 6\sigma_{\text{out}} [\rho_{\text{out}} + 1 + 0.5c^2]$$

Set $c = \sqrt{2\beta}$ to reach the conclusion. \square

B.1 Tools for Computing the Summary Statistics

The proof of Theorem B.1 requires probability bounds on the permeance statistic \mathcal{P} and the alignment statistic \mathcal{A} . These bounds follow from tail inequalities for Gaussian and spherically distributed random vectors that we develop in the next two subsections.

B.1.1 Tools for the Permeance Statistic

In this section, we develop the probability inequality that we need to estimate the permeance statistic $\mathcal{P}(L)$ for data drawn from the Haystack model.

Lemma B.2. *Suppose $\mathbf{g}_1, \dots, \mathbf{g}_n$ are i.i.d. $\text{NORMAL}(\mathbf{0}, \mathbf{I})$ vectors in \mathbb{R}^d . For each $t \geq 0$,*

$$\inf_{\|\mathbf{u}\|=1} \sum_{i=1}^n |\langle \mathbf{u}, \mathbf{g}_i \rangle| > \sqrt{\frac{2}{\pi}} \cdot n - 2\sqrt{nd} - t\sqrt{n}, \quad (\text{B.2})$$

except with probability $e^{-t^2/2}$.

Proof. Add and subtract the mean from each summand on the left-hand side of (B.2) to obtain

$$\inf_{\|\mathbf{u}\|=1} \sum_{i=1}^n |\langle \mathbf{u}, \mathbf{g}_i \rangle| \geq \inf_{\|\mathbf{u}\|=1} \sum_{i=1}^n [|\langle \mathbf{u}, \mathbf{g}_i \rangle| - \mathbb{E}|\langle \mathbf{u}, \mathbf{g}_i \rangle|] + \inf_{\|\mathbf{u}\|=1} \sum_{i=1}^n \mathbb{E}|\langle \mathbf{u}, \mathbf{g}_i \rangle| \quad (\text{B.3})$$

The second sum on the right-hand side has a closed form expression because each term is the expectation of a half-Gaussian random variable: $\mathbb{E}|\langle \mathbf{u}, \mathbf{g}_i \rangle| = \sqrt{2/\pi}$ for every unit vector \mathbf{u} . Therefore,

$$\inf_{\|\mathbf{u}\|=1} \sum_{i=1}^n \mathbb{E}|\langle \mathbf{u}, \mathbf{g}_i \rangle| = \sqrt{\frac{2}{\pi}} \cdot n. \quad (\text{B.4})$$

To control the first sum on the right-hand side of (B.3), we use a standard argument. To bound the mean, we symmetrize the sum and invoke a comparison theorem. To control the probability of a large deviation, we apply a measure concentration argument.

To proceed with the calculation of the mean, we use the Rademacher symmetrization lemma [44, Lem. 6.3] to obtain

$$\mathbb{E} \sup_{\|\mathbf{u}\|=1} \sum_{i=1}^n [(\mathbb{E}|\langle \mathbf{u}, \mathbf{g}_i \rangle|) - |\langle \mathbf{u}, \mathbf{g}_i \rangle|] \leq 2 \mathbb{E} \sup_{\|\mathbf{u}\|=1} \sum_{i=1}^n \varepsilon_i |\langle \mathbf{u}, \mathbf{g}_i \rangle|.$$

The random variables $\varepsilon_1, \dots, \varepsilon_n$ are i.i.d. Rademacher random variables that are independent from the Gaussian sequence. Next, invoke the Rademacher comparison theorem [44, Eqn. (4.20)] with the function $\varphi(\cdot) = |\cdot|$ to obtain the further bound

$$\mathbb{E} \sup_{\|\mathbf{u}\|=1} \sum_{i=1}^n [(\mathbb{E}|\langle \mathbf{u}, \mathbf{g}_i \rangle|) - |\langle \mathbf{u}, \mathbf{g}_i \rangle|] \leq 2 \mathbb{E} \sup_{\|\mathbf{u}\|=1} \sum_{i=1}^n \varepsilon_i \langle \mathbf{u}, \mathbf{g}_i \rangle = 2 \mathbb{E} \left\| \sum_{i=1}^n \varepsilon_i \mathbf{g}_i \right\|.$$

The identity follows when we draw the sum into the inner product and maximize over all unit vectors. From here, the rest of the argument is very easy. Use Jensen's inequality to bound the expectation by the root-mean-square, which has a closed form:

$$\mathbb{E} \sup_{\|\mathbf{u}\|=1} \sum_{i=1}^n [(\mathbb{E} |\langle \mathbf{u}, \mathbf{g}_i \rangle|) - |\langle \mathbf{u}, \mathbf{g}_i \rangle|] \leq 2 \left[\mathbb{E} \left\| \sum_{i=1}^n \varepsilon_i \mathbf{g}_i \right\|^2 \right]^{1/2} = 2\sqrt{nd}. \quad (\text{B.5})$$

Note that the mean fluctuation (B.5) is dominated by the centering term (B.4) when $n \gg d$.

To control the probability that the fluctuation term is large, we use a standard concentration inequality [6, Theorem 1.7.6] for a Lipschitz function of independent Gaussian variables. Define a real-valued function on $d \times n$ matrices: $f(\mathbf{Z}) = \sup_{\|\mathbf{u}\|=1} \sum_{i=1}^n (\sqrt{2/\pi} - |\langle \mathbf{u}, \mathbf{z}_i \rangle|)$, where \mathbf{z}_i denotes the i th column of \mathbf{Z} . Compute that

$$|f(\mathbf{Z}) - f(\mathbf{Z}')| \leq \sup_{\|\mathbf{u}\|=1} \sum_{i=1}^n |\langle \mathbf{u}, \mathbf{z}_i - \mathbf{z}'_i \rangle| \leq \sum_{i=1}^n \|\mathbf{z}_i - \mathbf{z}'_i\| \leq \sqrt{n} \|\mathbf{Z} - \mathbf{Z}'\|_F.$$

Therefore, f has Lipschitz constant \sqrt{n} with respect to the Frobenius norm. In view of the estimate (B.5) for the mean, the Gaussian concentration bound implies that

$$\mathbb{P} \left\{ \sup_{\|\mathbf{u}\|=1} \sum_{i=1}^n [(\mathbb{E} |\langle \mathbf{u}, \mathbf{g}_i \rangle|) - |\langle \mathbf{u}, \mathbf{g}_i \rangle|] \geq 2\sqrt{nd} + t\sqrt{n} \right\} \leq e^{-t^2/2}. \quad (\text{B.6})$$

Introduce the bound (B.6) and the identity (B.4) into (B.3) to complete the proof. \square

B.1.2 Tools for the Alignment Statistic

In this section, we develop the probability inequalities that we need to estimate the alignment statistic $\mathcal{A}(L)$ for data drawn from the Haystack model. First, we need a tail bound for the maximum singular value of a Gaussian matrix. The following inequality is a well-known consequence of Slepian's lemma. See [20, Thm. 2.13] and the errata [21] for details.

Proposition B.3. *Let \mathbf{G} be an $m \times n$ matrix whose entries are i.i.d. standard normal random variables. For each $t \geq 0$,*

$$\mathbb{P} \{ \|\mathbf{G}\| \geq \sqrt{m} + \sqrt{n} + t \} < 1 - \Phi(t) < e^{-t^2/2},$$

where $\Phi(t)$ is the Gaussian cumulative density function

$$\Phi(t) := \frac{1}{\sqrt{2\pi}} \int_{-\infty}^t e^{-\tau^2/2} d\tau.$$

We also need a related result for random matrices with independent columns that are uniformly distributed on the sphere. The argument bootstraps from Proposition B.3.

Lemma B.4. *Let \mathbf{S} be an $m \times n$ matrix whose columns are i.i.d. random vectors distributed uniformly on the sphere \mathbb{S}^{m-1} in \mathbb{R}^m . For each $t \geq 0$,*

$$\mathbb{P} \left\{ \|\mathbf{S}\| \geq \frac{\sqrt{n} + \sqrt{m} + t}{\sqrt{m-0.5}} \right\} \leq 1.5 e^{-t^2/2}. \quad (\text{B.7})$$

Proof. Fix $\theta > 0$. The Laplace transform method shows that

$$P := \mathbb{P} \left\{ \|\mathbf{S}\| \geq \frac{\sqrt{n} + \sqrt{m} + t}{\sqrt{m-0.5}} \right\} \leq e^{-\theta(\sqrt{n} + \sqrt{m} + t)} \cdot \mathbb{E} e^{\theta\sqrt{m-0.5}\|\mathbf{S}\|}.$$

We compare $\|\mathbf{S}\|$ with the norm of a Gaussian matrix by introducing a diagonal matrix of χ -distributed variables. The rest of the argument is purely technical.

Let $\mathbf{r} = (r_1, \dots, r_n)$ be a vector of i.i.d. χ -distributed random variables with m degrees of freedom. Recall that $r_i \tilde{\mathbf{g}}_i \sim \mathbf{g}_i$, where $\tilde{\mathbf{g}}_i$ is uniform on the sphere and \mathbf{g}_i is standard normal. The mean of a χ -distributed variable satisfies an inequality due to Kershaw [42]:

$$\mathbb{E} r \geq \sqrt{m-0.5} \quad \text{when} \quad r \sim \chi_m.$$

Using Kershaw's bound and Jensen's inequality, we obtain

$$\mathbb{E} e^{\theta \sqrt{m-0.5} \|S\|} \leq \mathbb{E} e^{\theta \|\mathbb{E}_r \text{diag}(r) S\|} \leq \mathbb{E} e^{\theta \|G\|},$$

where G is an $m \times n$ matrix with i.i.d. standard normal entries.

Define a random variable $Z := \|G\| - \sqrt{n} - \sqrt{m}$, and let $Z_+ := \max\{Z, 0\}$ denote its positive part. Then

$$e^{\theta t} \cdot P \leq \mathbb{E} e^{\theta Z} \leq \mathbb{E} e^{\theta Z_+} = 1 + \int_0^\infty e^{\theta \tau} \cdot \mathbb{P}\{Z_+ > \tau\} d\tau.$$

Apply the cdf bound in Proposition B.3, and identify the complementary error function erfc .

$$e^{\theta t} \cdot P \leq 1 + \frac{\theta}{2} \int_0^\infty e^{\theta \tau} \cdot \text{erfc}\left(\frac{\tau}{\sqrt{2}}\right) d\tau,$$

A computer algebra system will report that this frightening integral has a closed form:

$$\theta \int_0^\infty e^{\theta \tau} \cdot \text{erfc}\left(\frac{\tau}{\sqrt{2}}\right) d\tau = e^{\theta^2/2} (\text{erf}(\theta) + 1) - 1 \leq 2e^{\theta^2/2} - 1.$$

We have used the simple bound $\text{erf}(\theta) \leq 1$ for $\theta \geq 0$. In summary,

$$P \leq e^{-\theta t} \cdot \left[\frac{1}{2} + e^{\theta^2/2} \right]$$

Select $\theta = t$ to obtain the advertised bound (B.7). \square

B.2 Proof of Theorem B.1

Suppose that the dataset \mathcal{X} is drawn from the Haystack model on page 11. Let \mathbf{X}_{out} be a $D \times N_{\text{out}}$ matrix whose columns are the outliers $\mathbf{x} \in \mathcal{X}_{\text{out}}$, arranged in fixed order. Recall that the inlier sampling ratio $\rho_{\text{in}} := N_{\text{in}}/d$ and the outlier sampling ratio $\rho_{\text{out}} := N_{\text{out}}/D$.

Let us begin with a lower bound for the permeance statistic $\mathcal{P}(L)$. The N_{in} inliers are drawn from a centered Gaussian distribution on the d -dimensional space L with covariance $(\sigma_{\text{in}}^2/d) \mathbf{I}_L$. Rotational invariance and Lemma B.2, with $t = c\sqrt{d}$, together imply that the permeance statistic (2.1) satisfies

$$\mathcal{P}(L) > \frac{\sigma_{\text{in}}}{\sqrt{d}} \left[\sqrt{\frac{2}{\pi}} \cdot N_{\text{in}} - (2+c) \sqrt{N_{\text{in}} d} \right] = \sigma_{\text{in}} \sqrt{d} \left[\sqrt{\frac{2}{\pi}} \rho_{\text{in}} - (2+c) \sqrt{\rho_{\text{in}}} \right],$$

except with probability $e^{-c^2 d/2}$.

Next, we obtain an upper bound for the alignment statistic $\mathcal{A}(L)$. The N_{out} outliers are independent, centered Gaussian vectors in \mathbb{R}^D with covariance $(\sigma_{\text{out}}^2/D) \mathbf{I}$. Proposition B.3, with $t = c\sqrt{d}$ shows that

$$\|\mathbf{X}_{\text{out}}\| \leq \frac{\sigma_{\text{out}}}{\sqrt{D}} \left[\sqrt{N_{\text{out}}} + \sqrt{D} + c\sqrt{d} \right] = \sigma_{\text{out}} \left[\sqrt{\rho_{\text{out}}} + 1 + c\sqrt{\frac{d}{D}} \right],$$

except with probability $e^{-c^2 d/2}$. Rotational invariance implies that the columns of $\Pi_{L^\perp} \widetilde{\mathbf{X}}_{\text{out}}$ are independent vectors that are uniformly distributed on the unit sphere of a $(D-d)$ -dimensional space. Lemma B.4 yields

$$\|\Pi_{L^\perp} \widetilde{\mathbf{X}}_{\text{out}}\| \leq \frac{\sqrt{N_{\text{out}}} + \sqrt{D-d} + c\sqrt{d}}{\sqrt{D-d-0.5}} < \sqrt{\frac{D}{D-d-0.5}} \left[\sqrt{\rho_{\text{out}}} + 1 + c\sqrt{\frac{d}{D}} \right],$$

except with probability $1.5e^{-c^2 d/2}$. It follows that

$$\mathcal{A}(L) \leq \sigma_{\text{out}} \sqrt{\frac{D}{D-d-0.5}} \left[\sqrt{\rho_{\text{out}}} + 1 + c\sqrt{\frac{d}{D}} \right]^2$$

except with probability $2.5e^{-c^2d/2}$.

Combining these bounds, we discover that the stability statistic satisfies

$$\begin{aligned} \mathcal{S}(L) &= \frac{\mathcal{P}(L)}{4\sqrt{d}} - \mathcal{L}(L) \\ &\geq \frac{\sigma_{\text{in}}}{4} \left[\sqrt{\frac{2}{\pi}} \rho_{\text{in}} - (2+c)\sqrt{\rho_{\text{in}}} \right] - \sigma_{\text{out}} \sqrt{\frac{D}{D-d-0.5}} \left[\sqrt{\rho_{\text{out}}} + 1 + c\sqrt{\frac{d}{D}} \right]^2 \end{aligned}$$

except with probability $3.5e^{-c^2d/2}$. This completes the argument.

C Analysis of the IRLS Algorithm

This appendix contains the details of our analysis of the IRLS method, Algorithm 4.2. First, we verify that Algorithm 4.1 reliably solves the weighted least-squares subproblem (4.1). Then, we argue that IRLS converges to a point near the true optimum of the REAPER problem (1.4).

C.1 Solving the Weighted Least-Squares Problem

In this section, we verify that Algorithm 4.1 correctly solves the weighted least-squares problem (4.1). The following lemma provides a more mathematical statement of the algorithm, along with the proof of correctness. Note that this statement is slightly more general than the recipe presented in Algorithm 4.1 because it is valid for over the entire range $0 < d < D$.

Lemma C.1 (Solving the Weighted Least-Squares Problem). *Assume that $0 < d < D$, and suppose that \mathcal{X} is a set of observations in \mathbb{R}^D . For each $\mathbf{x} \in \mathcal{X}$, let $\beta_{\mathbf{x}}$ be a nonnegative weight. Form the weighted sample covariance matrix C , and compute its eigenvalue decomposition:*

$$C := \sum_{\mathbf{x} \in \mathcal{X}} \beta_{\mathbf{x}} \mathbf{x} \mathbf{x}^t = U \Lambda U^t \quad \text{where } \lambda_1 \geq \dots \geq \lambda_D \geq 0.$$

When $\text{rank}(C) \leq d$, construct a vector $\boldsymbol{\nu} \in \mathbb{R}^D$ via the formula

$$\boldsymbol{\nu} := (\underbrace{1, \dots, 1}_{[d] \text{ times}}, d - [d], 0, \dots, 0)^t. \quad (\text{C.1})$$

When $\text{rank}(C) > d$, define the positive quantity θ implicitly by solving the equation

$$\sum_{i=1}^D \frac{[\lambda_i - \theta]_+}{\lambda_i} = d. \quad (\text{C.2})$$

Construct a vector $\boldsymbol{\nu} \in \mathbb{R}^D$ whose components are

$$\nu_i := \frac{[\lambda_i - \theta]_+}{\lambda_i} \quad \text{for } i = 1, \dots, D. \quad (\text{C.3})$$

In either case, an optimal solution to (4.1) is given by

$$\mathbf{P}_\star := U \cdot \text{diag}(\boldsymbol{\nu}) \cdot U^t. \quad (\text{C.4})$$

In this statement, we enforce the convention $0/0 := 0$, and diag forms a diagonal matrix from a vector.

Proof of Lemma C.1. First, observe that the construction (C.4) yields a matrix \mathbf{P}_\star that satisfies the constraints of (4.1) in both cases.

When $\text{rank}(C) \leq d$, we can verify that our construction of the vector $\boldsymbol{\nu}$ yields a optimizer of (4.1) by showing that the objective value is zero, which is minimal. Evaluate the objective function (4.1) at the point \mathbf{P}_\star to see that

$$\sum_{\mathbf{x} \in \mathcal{X}} \beta_{\mathbf{x}} \|(\mathbf{I} - \mathbf{P}_\star) \mathbf{x}\|^2 = \text{tr}[(\mathbf{I} - \mathbf{P}_\star) C (\mathbf{I} - \mathbf{P}_\star)] = \sum_{i=1}^D (1 - \nu_i)^2 \lambda_i \quad (\text{C.5})$$

by definition of C and the fact that C and P_* are simultaneously diagonalizable. The nonzero eigenvalues of C appear among $\lambda_1, \dots, \lambda_{\lfloor d \rfloor}$. At the same time, $1 - v_i = 0$ for each $i = 1, \dots, \lfloor d \rfloor$. Therefore, the value of (C.5) equals zero at P_* .

Next, assume that $\text{rank}(C) > d$. The objective function in (4.1) is convex, so we can verify that P_* solves the optimization problem if the directional derivative of the objective at P_* is nonnegative in every feasible direction. A matrix Δ is a feasible perturbation if and only if

$$0 \preceq P_* + \Delta \preceq I \quad \text{and} \quad \text{tr } \Delta = 0.$$

Let Δ be an arbitrary matrix that satisfies these constraints. By expanding the objective of (4.1) about P_* , easily compute the derivative in the direction Δ . In particular, the condition

$$-\langle \Delta, (I - P_*)C \rangle \geq 0 \tag{C.6}$$

ensures that the derivative increases in the direction Δ . We now set about verifying (C.6) for our choice of P_* and all feasible Δ .

Note first that the quantity θ can be defined. Indeed, the left-hand side of (C.2) equals $\text{rank}(C)$ when $\theta = 0$, and it equals zero when $\theta \geq \lambda_1$. By continuity, there exists a value of θ that solves the equation. Let i_* be the largest index where $\lambda_{i_*} > \theta$, so that $v_i = 0$ for each $i > i_*$. Next, define M to be the subspace spanned by the eigenvectors u_{i_*+1}, \dots, u_D . Since v_i is the eigenvalue of P_* with eigenvector u_i , we must have $\Pi_M P_* \Pi_M = 0$. It follows that $\Pi_M \Delta \Pi_M \succcurlyeq 0$ because $\Pi_M (P_* + \Delta) \Pi_M \succcurlyeq 0$.

To complete the argument, observe that

$$(1 - v_i)\lambda_i = \lambda_i - (\lambda_i - \theta)_+ = \min\{\lambda_i, \theta\}.$$

Therefore, $(I - P_*)C = U \cdot \text{diag}(\min\{\lambda_i, \theta\}) \cdot U^t$. Using the fact that $\text{tr } \Delta = 0$, we obtain

$$\begin{aligned} \langle \Delta, (I - P_*)C \rangle &= \langle \Delta, U \cdot \text{diag}(\min\{\lambda_i, \theta\}) \cdot U^t \rangle \\ &= \langle \Delta, \underbrace{U \cdot \text{diag}(0, \dots, 0, \lambda_{i_*+1} - \theta, \dots, \lambda_D - \theta)}_{=: Z} \cdot U^t \rangle \end{aligned}$$

Since $\lambda_i \leq \theta$ for each $i > i_*$, each eigenvalue of Z is nonpositive. Furthermore, $\Pi_M Z \Pi_M = Z$. We see that

$$\langle \Delta, (I - P_*)C \rangle = \langle \Delta, \Pi_M Z \Pi_M \rangle = \langle \Pi_M \Delta \Pi_M, Z \rangle \leq 0,$$

because the compression of Δ on M is positive semidefinite and Z is negative semidefinite. In other words, (C.6) is satisfied for every feasible perturbation Δ about P_* . \square

C.2 Convergence of IRLS

In this section, we argue that the IRLS method of Algorithm 4.2 converges to a point whose value is nearly optimal for the REAPER problem (1.4). The proof consists of two phases. First, we explain how to modify the argument from [13] to show that the iterates $P^{(k)}$ converge to a matrix P_δ , which is characterized as the solution to a regularized counterpart of REAPER. The fact that the limit point P_δ achieves a near-optimal value for REAPER follows from the characterization.

Proof sketch for Theorem 4.1. We find it more convenient to work with the variables $Q := I - P$ and $Q^{(k)} := I - P^{(k)}$. First, let us define a regularized objective. For a parameter $\delta > 0$, consider the Huber-like function

$$H_\delta(x, y) = \begin{cases} \frac{1}{2} \left(\frac{x^2}{\delta} + \delta \right), & 0 \leq y \leq \delta \\ \frac{1}{2} \left(\frac{x^2}{y} + y \right), & y \geq \delta. \end{cases}$$

We introduce the convex function

$$\begin{aligned} F(Q) &:= \sum_{x \in \mathcal{X}} H_\delta(\|Qx\|, \|Qx\|) \\ &= \sum_{\{x: \|Qx\| \geq \delta\}} \|Qx\| + \frac{1}{2} \sum_{\{x: \|Qx\| < \delta\}} \left(\frac{\|Qx\|^2}{\delta} + \delta \right). \end{aligned}$$

The second identity above highlights the interpretation of F as a regularized objective function for (1.4) under the assignment $\mathbf{Q} = \mathbf{I} - \mathbf{P}$. Note that F is continuously differentiable at each matrix \mathbf{Q} , and the gradient

$$\nabla F(\mathbf{Q}) = \sum_{\mathbf{x} \in \mathcal{X}} \frac{\mathbf{Q} \mathbf{x} \mathbf{x}^\top}{\max\{\|\mathbf{Q} \mathbf{x}\|, \delta\}}.$$

The technical assumption that the observations do not lie in the union of two strict subspaces of \mathbb{R}^D implies that F is *strictly* convex; compare with the proof [80, Thm. 2]. We define \mathbf{Q}_δ to be the solution of a constrained optimization problem:

$$\mathbf{Q}_\delta := \arg \min_{\substack{\mathbf{0} \preceq \mathbf{Q} \preceq \mathbf{I} \\ \text{tr } \mathbf{Q} = D-d}} F(\mathbf{Q}).$$

The strict convexity of F implies that \mathbf{Q}_δ is well defined.

The key idea in the proof is to show that the iterates $\mathbf{Q}^{(k)}$ of Algorithm 4.2 converge to the optimizer \mathbf{Q}_δ of the regularized objective function F . We demonstrate that Algorithm 4.2 is a generalized Weiszfeld method in the sense of [13, Sec. 4]. After defining some additional auxiliary functions and facts about these functions, we explain how the argument of [13, Lem. 5.1] can be adapted to prove that the iterates of $\mathbf{Q}^{(k)} = \mathbf{I} - \mathbf{P}^{(k)} \rightarrow \mathbf{Q}_\delta$. The only innovation required is an inequality from convex analysis that lets us handle the constraints $\mathbf{0} \preceq \mathbf{Q} \preceq \mathbf{I}$ and $\text{tr } \mathbf{Q} = D - d$.

Now for the definitions. We introduce the potential function

$$G(\mathbf{Q}, \mathbf{Q}^{(k)}) := \sum_{\mathbf{x} \in \mathcal{X}} H_\delta(\|\mathbf{Q} \mathbf{x}\|, \|\mathbf{Q}^{(k)} \mathbf{x}\|).$$

Then $G(\cdot, \mathbf{Q}^{(k)})$ is a smooth quadratic function. By collecting terms, we may relate G and F through the expansion

$$G(\mathbf{Q}, \mathbf{Q}^{(k)}) = F(\mathbf{Q}^{(k)}) + \langle \mathbf{Q} - \mathbf{Q}^{(k)}, \nabla F(\mathbf{Q}^{(k)}) \rangle + \frac{1}{2} \langle \mathbf{Q} - \mathbf{Q}^{(k)}, C(\mathbf{Q}^{(k)})(\mathbf{Q} - \mathbf{Q}^{(k)}) \rangle,$$

where C is the continuous function

$$C(\mathbf{Q}^{(k)}) := \sum_{\mathbf{x} \in \mathcal{X}} \frac{\mathbf{x} \mathbf{x}^\top}{\max\{\|\mathbf{Q} \mathbf{x}\|, \delta\}}.$$

Next, we verify some facts related to Hypothesis 4.2 and 4.3 of [13, Sec. 4]. Note that $F(\mathbf{Q}) = G(\mathbf{Q}, \mathbf{Q})$. Furthermore, $F(\mathbf{Q}) \leq G(\mathbf{Q}, \mathbf{Q}^{(k)})$ because $H_\delta(x, x) \leq H_\delta(x, y)$, which is a direct consequence of the AM–GM inequality.

We now relate the iterates of Algorithm 4.2 to the definitions above. Given that $\mathbf{Q}^{(k)} = \mathbf{I} - \mathbf{P}^{(k)}$, Step 2b of Algorithm 4.2 is equivalent to the iteration

$$\mathbf{Q}^{(k+1)} = \arg \min_{\substack{\mathbf{0} \preceq \mathbf{Q} \preceq \mathbf{I} \\ \text{tr } \mathbf{Q} = D-d}} G(\mathbf{Q}, \mathbf{Q}^{(k)}).$$

From this characterization, we have the monotonicity property

$$F(\mathbf{Q}^{(k+1)}) \leq G(\mathbf{Q}^{(k+1)}, \mathbf{Q}^{(k)}) \leq G(\mathbf{Q}^{(k)}, \mathbf{Q}^{(k)}) = F(\mathbf{Q}^{(k)}). \quad (\text{C.7})$$

This fact motivates the stopping criterion for Algorithm 4.2 because it implies the objective values are decreasing: $\alpha^{(k+1)} = F(\mathbf{Q}^{(k+1)}) \leq F(\mathbf{Q}^{(k)}) = \alpha^{(k)}$.

We also require some information regarding the bilinear form induced by C . Introduce the quantity $m := \max\{\delta, \max_{\mathbf{x} \in \mathcal{X}} \{\|\mathbf{x}\|\}\}$. Then, by symmetry of the matrix \mathbf{Q} , and the fact that the inner product between positive semidefinite matrices is nonnegative we have

$$\langle \mathbf{Q}, C(\mathbf{Q}^{(k)}) \mathbf{Q} \rangle \geq \frac{1}{m} \text{tr} \left(\mathbf{Q}^2 \sum_{\mathbf{x} \in \mathcal{X}} \mathbf{x} \mathbf{x}^\top \right) \geq \|\mathbf{Q}\|_F^2 \underbrace{\left(\frac{\lambda_{\min}(\sum_{\mathbf{x} \in \mathcal{X}} \mathbf{x} \mathbf{x}^\top)}{m} \right)}_{=: \mu}.$$

The technical assumption that the observations do not lie in two strict subspaces of \mathbb{R}^D implies in particular that the observations span \mathbb{R}^D . We deduce that $\mu > 0$.

Now we discuss the challenge imposed by the constraint set. When the minimizer $Q^{(k+1)}$ lies on the boundary of the constraint set, the equality [13, Eqn. (4.3)] may not hold. However, if we denote the gradient of G with respect to its first argument by G_Q , the inequality

$$\begin{aligned} 0 &\leq \langle Q - Q^{(k+1)}, G_Q(Q^{(k+1)}, Q^{(k)}) \rangle \\ &= \langle Q - Q^{(k+1)}, \nabla F(Q^{(k)}) + C(Q^{(k)})(Q^{(k+1)} - Q^{(k)}) \rangle \end{aligned} \quad (C.8)$$

holds for every Q in the feasible set. This is simply the first-order necessary and sufficient condition for the constrained minimum of a smooth convex function over a convex set.

With the facts above, a proof that the iterates $Q^{(k)}$ converge to Q_δ follows the argument of [13, Lem. 5.1] nearly line-by-line. However, due to inequality (C.8), the final conclusion is that, at the limit point \bar{Q} , the inequality $\langle \bar{Q} - \bar{Q}, \nabla F(\bar{Q}) \rangle \geq 0$ holds for all feasible Q . This inequality characterizes the global minimum of a convex function over a convex set, so the limit point must indeed be a global minimizer. That is, $\bar{Q} = Q_\delta$. In particular, this argument shows that the iterates $P^{(k)}$ converge to $P_\delta := I - Q_\delta$ as $k \rightarrow \infty$.

The only remaining claim is that $P_\delta = I - Q_\delta$ nearly minimizes (1.4). We abbreviate the objective of (1.4) under the identification $Q = I - P$ by

$$F_0(Q) := \sum_{x \in \mathcal{X}} \|Qx\|.$$

Define $Q_\star := \arg \min F_0(Q)$ with respect to the feasible set $0 \preceq Q \preceq I$ and $\text{tr}(Q) = D - d$. From the easy inequalities $x \leq H_\delta(x, x) \leq x + \frac{1}{2}\delta$ for $x \geq 0$, we see that

$$0 \leq F(Q) - F_0(Q) \leq \frac{1}{2}\delta |\mathcal{X}|.$$

Evaluate the latter inequality at Q_δ , and subtract the result from the inequality evaluated at Q_\star to reach

$$(F(Q_\star) - F(Q_\delta)) + (F_0(Q_\delta) - F_0(Q_\star)) \leq \frac{1}{2}\delta |\mathcal{X}|.$$

Since Q_δ and Q_\star are optimal for their respective problems, both terms in parenthesis above are positive, and we deduce that $F_0(Q_\delta) - F_0(Q_\star) \leq \frac{1}{2}\delta |\mathcal{X}|$. Since F_0 is the objective function for (1.4) under the map $P = I - Q$, the proof is complete. \square

Acknowledgements Lerman and Zhang were supported in part by the IMA and by NSF grants DMS-09-15064 and DMS-09-56072. McCoy and Tropp were supported by ONR awards N00014-08-1-0883 and N00014-11-1-002, AFOSR award FA9550-09-1-0643, DARPA award N66001-08-1-2065, and a Sloan Research Fellowship. The authors thank Eran Halperin, Yi Ma, Ben Recht, Amit Singer, and John Wright for helpful conversations. The anonymous referees provided many thoughtful and incisive remarks that helped us improve the manuscript immensely.

References

1. Ammann, L.P.: Robust singular value decompositions: A new approach to projection pursuit. *J. Amer. Statist. Assoc.* **88**(422), 505–514 (1993). URL <http://www.jstor.org/stable/2290330>
2. Bargiela, A., Hartley, J.K.: Orthogonal linear regression algorithm based on augmented matrix formulation. *Comput. Oper. Res.* **20**, 829–836 (1993). DOI 10.1016/0305-0548(93)90104-Q. URL <http://dl.acm.org/citation.cfm?id=165819.165826>
3. Basri, R., Jacobs, D.: Lambertian reflectance and linear subspaces. *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(2), 218–233 (2003)
4. Bhatia, R.: Matrix Analysis. No. 169 in Graduate Texts in Mathematics. Springer, New York (1997)
5. Björck, Å.: Numerical Methods for Least Squares Problems. Society for Industrial and Applied Mathematics, Philadelphia, PA (1996)
6. Bogachev, V.I.: Gaussian Measures, *Mathematical Surveys and Monographs*, vol. 62. American Mathematical Society, Providence, RI (1998)
7. Bonnans, J.F., Shapiro, A.: Perturbation Analysis of Optimization Problems. Springer Series in Operations Research. Springer (2000)

8. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning* **3**(1), 1–122 (2010). DOI 10.1561/22000000016. URL <http://www.nowpublishers.com/product.aspx?product=MAL&doi=22000000016>
9. Brubaker, S.C.: Robust PCA and clustering in noisy mixtures. In: Proc. 20th Ann. ACM-SIAM Symp. Discrete Algorithms, SODA '09, pp. 1078–1087. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2009). URL <http://portal.acm.org/citation.cfm?id=1496770.1496887>
10. Caltech 101. Online (2006). http://www.vision.caltech.edu/Image_Datasets/Caltech101/
11. Candès, E.J., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? *J. Assoc. Comput. Mach.* **58**(3) (2011)
12. Cavalier, T.M., Melloy, B.J.: An iterative linear programming solution to the Euclidean regression model. *Comput. Oper. Res.* **18**, 655–661 (1991)
13. Chan, T.F., Mulet, P.: On the convergence of the lagged diffusivity fixed point method in total variation image restoration. *SIAM J. Numer. Anal.* **36**, 354–367 (1999). DOI <http://dx.doi.org/10.1137/S0036142997327075>. URL <http://dx.doi.org/10.1137/S0036142997327075>
14. Chandrasekaran, V., Sanghavi, S., Parrilo, P.A., Willsky, A.S.: Rank-sparsity incoherence for matrix decomposition. *SIAM J. Optim.* **21**(2), 572–596 (2011). DOI 10.1137/090761793
15. Cook, R.D., Weisberg, S.: Residuals and influence in regression. Chapman and Hall, New York (1982)
16. Costeira, J., Kanade, T.: A multibody factorization method for independently moving objects. *Int. J. Comput. Vision* **29**(3), 159–179 (1998)
17. Coudron, M., Lerman, G.: On the sample complexity of robust pca. In: NIPS, pp. 3230–3238 (2012)
18. Croux, C., Filzmoser, P., Oliveira, M.: Algorithms for projection pursuit robust principal component analysis. *Chemometrics Intell. Lab. Sys.* **87**(2), 218–225 (2007)
19. Croux, C., Haesbroeck, G.: Principal component analysis based on robust estimators of the covariance or correlation matrix: Influence functions and efficiencies. *Biometrika* **87**, 603–618 (2000)
20. Davidson, K.R., Szarek, S.J.: Local operator theory, random matrices and Banach spaces. In: Handbook of the geometry of Banach spaces, Vol. I, pp. 317–366. North-Holland, Amsterdam (2001). DOI 10.1016/S1874-5849(01)80010-3. URL [http://dx.doi.org/10.1016/S1874-5849\(01\)80010-3](http://dx.doi.org/10.1016/S1874-5849(01)80010-3)
21. Davidson, K.R., Szarek, S.J.: Addenda and corrigenda to: “Local operator theory, random matrices and Banach spaces” [in *Handbook of the geometry of Banach spaces, Vol. I*, 317–366, North-Holland, Amsterdam, 2001; MR1863696 (2004f:47002a)]. In: Handbook of the geometry of Banach spaces, Vol. 2, pp. 1819–1820. North-Holland, Amsterdam (2003)
22. Davies, P.L.: Asymptotic behaviour of S-estimates of multivariate location parameters and dispersion matrices. *Ann. Statist.* **15**(3), 1269–1292 (1987). URL <http://www.jstor.org/stable/2241828>
23. Deerwester, S., Dumais, S., Landauer, T., Furna, G., Beck, L.: Improving Information Retrieval with Latent Semantic Indexing. In: C.L. Borgman, E.Y.H. Pai (eds.) *Information & Technology Planning for the Second 50 Years Proceedings of the 51st Annual Meeting of the American Society for Information Science*, vol. 25. Learned Information, Inc., Atlanta, Georgia (1988)
24. Devlin, S.J., Gnandesikan, R., Kettenring, J.R.: Robust estimation of dispersion matrices and principal components. *J. Amer. Statist. Assoc.* **76**(374), 354–362 (1981). URL <http://www.jstor.org/stable/2287836>
25. Ding, C., Zhou, D., He, X., Zha, H.: R1-PCA: Rotational invariant L_1 -norm principal component analysis for robust subspace factorization. In: ICML '06: Proc. 23rd Int. Conf. Machine Learning, pp. 281–288. Association for Computing Machinery, Pittsburgh, PA (2006). DOI <http://doi.acm.org/10.1145/1143844.1143880>
26. Dodge, Y.: An introduction to l_1 -norm based statistical data analysis. *Comput. Statist. Data Anal.* **5**(4), 239–253 (1987). DOI 10.1016/0167-9473(87)90048-X. URL <http://www.sciencedirect.com/science/article/pii/016794738790048X>
27. Eckart, C., Young, G.: A principal axis transformation for non-hermitian matrices. *Bull. Amer. Math. Soc.* **45**(2), 118–121 (1939)
28. Epstein, R., Hallinan, P., Yuille, A.L.: 5 ± 2 eigenimages suffice: An empirical investigation of low-dimensional lighting models. In: Physics-Based Modeling in Computer Vision, 1995., Proceedings of the Workshop on, p. 108 (1995). DOI 10.1109/PBMCV.1995.514675
29. Eriksson, A., van den Hengel, A.: Efficient computation of robust low-rank matrix approximations in the presence of missing data using the l_1 norm. In: Proc. 2010 IEEE Conf. Computer Vision and Pattern Recognition, pp. 771–778 (2010). DOI 10.1109/CVPR.2010.5540139
30. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from a few training examples: an incremental bayesian approach tested on 101 object categories. In: CVPR 2004, Workshop on Generative-Model Based Vision. IEEE (2004)
31. Fischler, M., Bolles, R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. Assoc. Comput. Mach.* **24**(6), 381–395 (1981)

32. Goemans, M.X., Williamson, D.P.: Improved approximation for maximum cut and satisfiability problems using semidefinite programming. *J. Assoc. Comput. Mach.* **42**, 1115–1145 (1995)
33. Grant, M., Boyd, S.: Graph implementations for nonsmooth convex programs. In: V. Blondel, S. Boyd, H. Kimura (eds.) *Recent Advances in Learning and Control, Lecture Notes in Control and Information Sciences*, pp. 95–110. Springer, London (2008). http://stanford.edu/~boyd/graph_dcp.html
34. Grant, M., Boyd, S.: CVX: Matlab software for disciplined convex programming, version 1.21. <http://cvxr.com/cvx> (2010)
35. Halko, N., Martinsson, P.G., Tropp, J.A.: Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions. *SIAM Rev.* **53**(2), 217–288 (2011)
36. Harter, H.L.: The method of least squares and some alternatives: Part I. *Int. Statist. Rev.* **42**(2), 147–174 (1974)
37. Harter, H.L.: The method of least squares and some alternatives: Part II. *Int. Statist. Rev.* **42**(3), 235–264+282 (1974)
38. Ho, J., Yang, M., Lim, J., Lee, K., Kriegman, D.: Clustering appearances of objects under varying illumination conditions. In: *Proc. 2003 IEEE Int. Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 11–18 (2003)
39. Huber, P.J., Ronchetti, E.M.: *Robust Statistics*, 2nd edn. Wiley Series in Probability and Statistics. Wiley, Hoboken, NJ (2009). DOI 10.1002/9780470434697. URL <http://dx.doi.org/10.1002/9780470434697>
40. van Huffel, S., Vandewalle, J.: *Total Least Squares: Computational Aspects and Analysis*. Society for Industrial and Applied Mathematics, Philadelphia, PA (1987)
41. Jolliffe, I.T.: *Principal Component Analysis*, 2nd edn. Springer, Berlin (2002)
42. Kershaw, D.: Some extensions of W. Gautschi's inequalities for the gamma function. *Math. Comput.* **41**(164), pp. 607–611 (1983). URL <http://www.jstor.org/stable/2007697>
43. Kwak, N.: Principal component analysis based on L_1 -norm maximization. *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(9), 1672–1680 (2008). DOI 10.1109/TPAMI.2008.114. URL <http://dx.doi.org/10.1109/TPAMI.2008.114>
44. Ledoux, M., Talagrand, M.: *Probability in Banach spaces, Ergebnisse der Mathematik und ihrer Grenzgebiete (3) [Results in Mathematics and Related Areas (3)]*, vol. 23. Springer, Berlin (1991). Isoperimetry and processes
45. Lee, K.C., Ho, J., Kriegman, D.: Acquiring linear subspaces for face recognition under variable lighting. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(5), 684–698 (2005)
46. Lerman, G., McCoy, M.B., Tropp, J.A., Zhang, T.: Robust computation of linear models, or how to find a needle in a haystack (2012). Available at [arxiv:1202.4044v1](https://arxiv.org/abs/1202.4044)
47. Lerman, G., Zhang, T.: ℓ_p -Recovery of the most significant subspace among multiple subspaces with outliers (2010). Available at [arxiv:1012.4116](https://arxiv.org/abs/1012.4116). To appear in *Constructive Approximation*
48. Lerman, G., Zhang, T.: Robust recovery of multiple subspaces by geometric ℓ_p minimization. *Ann. Statist.* **39**(5), 2686–2715 (2011)
49. Li, G., Chen, Z.: Projection-pursuit approach to robust dispersion matrices and principal components: Primary theory and Monte Carlo. *J. Amer. Statist. Assoc.* **80**(391), 759–766 (1985). DOI 10.2307/2288497. URL <http://dx.doi.org/10.2307/2288497>
50. Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y., Ma, Y.: Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(1), 171–184 (2013). DOI 10.1109/TPAMI.2012.88
51. Locantore, N., Marron, J.S., Simpson, D.G., Tripoli, N., Zhang, J.T., Cohen, K.L.: Robust principal component analysis for functional data. *Test* **8**(1), 1–73 (1999). DOI 10.1007/BF02595862. URL <http://dx.doi.org/10.1007/BF02595862>. With discussion and a rejoinder by the authors
52. Lovasz, L., Schrijver, A.: Cones of matrices and set-functions and 0–1 optimization. *SIAM J. Optim.* **1**(2), 166–190 (1991). DOI 10.1137/0801013. URL <http://link.aip.org/link/?SJE/1/166/1>
53. Maronna, R.: Principal components and orthogonal regression based on robust scales. *Technometrics* **47**(3), 264–273 (2005). DOI 10.1198/004017005000000166. URL <http://dx.doi.org/10.1198/004017005000000166>
54. Maronna, R.A.: Robust M-estimators of multivariate location and scatter. *Ann. Statist.* **4**(1), 51–67 (1976). URL <http://www.jstor.org/stable/2957994>
55. Maronna, R.A., Martin, D.R., Yohai, V.J.: *Robust Statistics*. Wiley Series in Probability and Statistics. Wiley, Chichester (2006). DOI 10.1002/0470010940. URL <http://dx.doi.org/10.1002/0470010940>. Theory and methods
56. McCoy, M., Tropp, J.A.: Two proposals for robust PCA using semidefinite programming. *Electron. J. Statist.* **5**, 1123–1160 (2011)

57. Novembre, J., Johnson, T., Bryc, K., Kutalik, Z., Boyko, A.R., Auton, A., Indap, A., King, K.S., Bergmann, S., Nelson, M., Stephens, M., Bustamante, C.D.: Genes mirror geography within Europe. *Nature* **456**(7218), 98–101 (2008). DOI 10.1038/nature07331. URL http://www.ncbi.nlm.nih.gov/pubmed/18758442?itool=EntrezSystem2.PEntrez.Pubmed.Pubmed_ResultsPanel.Pubmed_RVDocSum&ordinalpos=8
58. Nyquist, H.: Least orthogonal absolute deviations. *Comput. Statist. Data Anal.* **6**(4), 361–367 (1988). DOI 10.1016/0167-9473(88)90076-X. URL <http://www.sciencedirect.com/science/article/pii/016794738890076X>
59. Osborne, M.R., Watson, G.A.: An analysis of the total approximation problem in separable norms, and an algorithm for the total l_1 problem. *SIAM J. Sci. Statist. Comput.* **6**(2), 410–424 (1985). DOI 10.1137/0906029. URL <http://link.aip.org/link/?SCE/6/410/1>
60. Overton, M.L., Womersley, R.S.: On the sum of the largest eigenvalues of a symmetric matrix. *SIAM J. Matrix Anal. Appl.* **13**(1), 41–45 (1992)
61. Price, A.L., Patterson, N.J., Plenge, R.M., Weinblatt, M.E., Shadick, N.A., Reich, D.: 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>
62. Rockafellar, R.T.: *Convex analysis*. Princeton Mathematical Series, No. 28. Princeton University Press, Princeton, N.J. (1970)
63. Rousseeuw, P.J.: Least median of squares regression. *J. Amer. Statist. Assoc.* **79**(388), 871–880 (1984)
64. Rousseeuw, P.J., Leroy, A.M.: *Robust Regression and Outlier Detection*. Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics. Wiley, New York (1987)
65. Späth, H., Watson, G.A.: On orthogonal linear approximation. *Numer. Math.* **51**, 531–543 (1987). DOI 10.1007/BF01400354. URL <http://dl.acm.org/citation.cfm?id=34311.34315>
66. Torre, F.D.L., Black, M.J.: Robust principal component analysis for computer vision. In: *Proc. 8th IEEE Conf. Computer Vision*, vol. 1, pp. 362–369 vol.1 (2001). DOI 10.1109/ICCV.2001.937541
67. Torre, F.D.L., Black, M.J.: A framework for robust subspace learning. *Int. J. Comput. Vision* **54**, 117–142 (2003). URL <http://dx.doi.org/10.1023/A:1023709501986>. 10.1023/A:1023709501986
68. Tropp, J.A.: Just relax: convex programming methods for identifying sparse signals in noise. *IEEE Trans. Inform. Theory* **52**(3), 1030–1051 (2006). DOI 10.1109/TIT.2005.864420. URL <http://dx.doi.org/10.1109/TIT.2005.864420>
69. Tropp, J.A.: Corrigendum in “just relax: Convex programming methods for identifying sparse signals in noise”. *IEEE Trans. Inform. Theory* **55**(2) (2009)
70. Vazirani, V.V.: *Approximation Algorithms*. Springer, Berlin (2003)
71. Voss, H., Eckhardt, U.: Linear convergence of generalized Weiszfeld’s method. *Computing* **25**, 243–251 (1980). URL <http://dx.doi.org/10.1007/BF02242002>. 10.1007/BF02242002
72. Wang, L., Singer, A.: Exact and stable recovery of rotations for robust synchronization. *Information and Inference* (2013). DOI 10.1093/imaiai/iat005
73. Watson, G.A.: Some problems in orthogonal distance and non-orthogonal distance regression. In: *Proc. 2001 Symp. Algorithms for Approximation IV*. Defense Technical Information Center (2001). URL <http://books.google.com/books?id=WKKWGwAACAAJ>
74. Watson, G.A.: On the Gauss–Newton method for l_1 orthogonal distance regression. *IMA J. Numer. Anal.* **22**(3), 345–357 (2002). DOI 10.1093/imanum/22.3.345. URL <http://imajna.oxfordjournals.org/content/22/3/345.abstract>
75. Witten, D., Tibshirani, R., Hastie, T.: A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostat.* **10**(3), 515–534 (2009)
76. Xu, H., Caramanis, C., Mannor, S.: Principal Component Analysis with Contaminated Data: The High Dimensional Case. In: *Proc. 2010 Conf. Learning Theory*. OmniPress, Haifa (2010)
77. Xu, H., Caramanis, C., Sanghavi, S.: Robust PCA via outlier pursuit. In: J. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R. Zemel, A. Culotta (eds.) *Neural Information Processing Systems 23*, pp. 2496–2504. MIT Press, Vancouver (2010)
78. Xu, H., Caramanis, C., Sanghavi, S.: Robust PCA via outlier pursuit. *IEEE Trans. Inform. Theory* **58**(5), 3047–3064 (2012)
79. Xu, L., Yuille, A.L.: Robust principal component analysis by self-organizing rules based on statistical physics approach. *IEEE Trans. Neural Networks* **6**(1), 131–143 (1995). DOI 10.1109/72.363442
80. Zhang, T., Lerman, G.: A novel m-estimator for robust PCA. *J. Mach. Learn. Res.* **15**, 749–808 (2014)
81. Zhang, T., Szeliski, A., Lerman, G.: Median K -flats for hybrid linear modeling with many outliers. In: *Proc. 12th IEEE Int. Conf. Computer Vision*, pp. 234–241. Kyoto (2009). DOI 10.1109/ICCVW.2009.5457695