Robust Sparse Regression under Adversarial Corruption

Yudong Chen YDCHEN@UTEXAS.EDU
Constantine Caramanis CONSTANTINE@UTEXAS.EDU
Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712

Shie Mannor SHIE@EE.TECHNION.AC.IL
Department of Electrical Engineering, Technion, Haifa 32000, Israel

Abstract

We consider high dimensional sparse regression with arbitrary – possibly, severe or coordinated – errors in the covariates matrix. We are interested in understanding how many corruptions we can tolerate, while identifying the correct support. To the best of our knowledge, neither standard outlier rejection techniques, nor recently developed robust regression algorithms (that focus only on corrupted response variables), nor recent algorithms for dealing with stochastic noise or erasures, can provide guarantees on support recovery. As we show, neither can the natural brute force algorithm that takes exponential time to find the subset of data and support columns, that yields the smallest regression error.

We explore the power of a simple idea: replace the essential linear algebraic calculation – the inner product – with a robust counterpart that cannot be greatly affected by a controlled number of arbitrarily corrupted points: the trimmed inner product. We consider three popular algorithms in the uncorrupted setting: Thresholding Regression, Lasso, and the Dantzig selector, and show that the counterparts obtained using the trimmed inner product are provably robust.

1. Introduction

Linear regression in general, and sparse linear regression in particular, seek to express a response variable as the linear combination of (a small number of) covariates. They form one of the most basic procedures in statistics, engineering, and science. More recently, regression has found increasing applications in the high-dimensional regime, where the number of variables, p, is much larger than the number of measurements or observations, n. The key structural property exploited in high-dimensional regression, is that the regressor is often sparse, or near sparse, and as recent research has demonstrated, in many cases it can be efficiently recovered, despite the underdetermined nature of the problem (e.g., (Chen et al., 1999; Candes & Tao, 2005)). Another common theme in large-scale learning problems – particularly problems in the high-dimensional regime – is that we not only have big data, but we have dirty data. Recently, attention has focused on the setting where the output (or response) variable and the matrix of covariates are plagued by erasures, and/or by stochastic additive noise (Rosenbaum & Tsybakov, 2011; Loh & Wainwright, 2012; Chen & Caramanis, 2013). Yet many applications may suffer from persistent errors, that are ill-modeled by stochastic distribution. Indeed, many problems, particularly those modeling human behavior, may exhibit maliciously corrupted data.

This paper is about extending sparse high-dimensional regression to be robust to this type of noise. We call this *deterministic* or *cardinality constrained* robustness, because rather than restricting the magnitude of the noise, or any other such property of the noise, we merely assume there is a bound on how many data points, or how many coordinates of every single covariate, are corrupted. Other than this number, we make absolutely no assumptions on what the adversary can do – the adversary is virtually unlimited in computational power and knowledge about our algorithm and about the authentic points. There are two basic models we consider. In both, we assume there is an underlying generative model: $y = X\beta^* + e$, where X is the matrix of covariates and e is sub-Gaussian noise.

Proceedings of the 30^{th} International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

In the *row-corruption model*, we assume that each pair of covariates and response we see is either due to the generative model, i.e., (y_i, X_i), or is corrupted in some arbitrary way, with the only restriction that at most n_1 such pairs are corrupted. In the *distributed corruption model*, we assume that y and *each column* of X, has n_1 elements that are arbitrarily corrupted (evidently, the second model is a strictly harsher corruption model). Building efficient regression algorithms that recover at least the support of β^* accurately subject to even such deterministic data corruption, greatly expands the scope of problems where regression can be productively applied. The basic question is when is this possible: how big can n_1 be, while still allowing correct recovery of the support of β^*?

Many sparse-regression algorithms have been proposed, and their properties under non adversarial observations are well understood; we survey some of these results in Section 3. Also well-known, is that the performance of standard algorithms (e.g., Lasso, Orthogonal Matching Pursuit) breaks down even in the face of just a few corrupted points or covariate coefficients. As more work has focused on robustness in the high-dimensional regime, it has also become clear that the techniques of classical robust statistics such as outlier removal preprocessing steps cannot be applied to the high-dimensional regime (Donoho, 1982; Huber, 1981). The reason lies in the high dimensionality. In this setting, identifying outliers *a priori* is typically impossible: outliers might not exhibit any strangeness in the ambient space due to the high-dimensional noise (see (Xu et al., 2013) for a further detailed discussion), and thus can be identified only when the true low-dimensional structure is (at least approximately) known; on the other hand, the true structure cannot be computed by ignoring outliers. Other classical approaches have involved replacing the standard mean squared loss with a trimmed variant or even median squared loss (Hampel et al., 1986). First, these are non convex optimization problems, and second, it is not clear that they provide any performance guarantees, especially in high dimensions.

Recently, the works in (Laska et al., 2009; Nguyen et al., 2011; Li, 2011) have proposed an approach to handle arbitrary corruption in the *response variable*. As we show, this approach faces serious difficulties when the *covariates* are also corrupted, and is bound to fail in this setting. One might modify this approach in the spirit of Total Least Squares (TLS) (Zhu et al., 2011) to account for noise in the covariates (discussed in Section 3), but it leads to non convex problems. Moreover, the approaches proposed in these papers are the natural convexification of the (exponential time) brute force algorithm that searches over all subsets of covariate/response pairs (i.e., rows of the measurement matrix and corresponding entries of the response vector) and subsets of the support (i.e., columns of the measurement matrix) and then returns the vector that minimizes the regression error over the best selection of such subsets. Perhaps surprisingly, we show that the brute force algorithm itself has remarkably weak performance. Another line of work has developed approaches to handle stochastic noise or small bounded noise in the covariates (Herman & Strohmer, 2010; Rosenbaum & Tsybakov, 2010; 2011; Loh & Wainwright, 2012; Chen & Caramanis, 2013). The corruption models studied by these authors, however, are different from ours which allows arbitrary and malicious noise; those results seem to depend crucially on the assumed structure of the noise and cannot handle the setting in this paper.

More generally, even beyond regression, in, e.g., robust PCA and robust matrix completion (Chandrasekaran et al., 2011; Candes et al., 2009; Xu et al., 2012; Chen et al., 2011; Lerman et al., 2012), recent robust recovery in high dimensions results have for the most part depended on convex optimization formulations. We show in Section 4 that for our setting, *convex-optimization based approaches that try to relax the brute-force formulation fail to recover support, with even a constant number of outliers*. Accordingly, we develop a different line of robust algorithms which utilize non-convex operations.

In summary, to the best of our knowledge, no robust sparse regression algorithm has been proposed that can provide performance guarantees, and in particular, support recovery, under *arbitrarily and maliciously corrupted* covariates and response variables.

We believe that robustness is of great interest both in practice and in theory. Modern applications often involve "big but dirty data", where outliers are ubiquitous either due to adversarial manipulation or to the fact some samples are generated from a model different from the assumed one. It is thus desirable to develop robust sparse regression procedures. From a theoretical perspective, it is somewhat surprising that the addition of a few outliers can transform a simple problem to a hard one; we discuss the difficulties in more detail in the subsequent sections.

Paper Contributions: We first present two negative results that are somewhat surprising. We show that a broad class of convex optimization-based methods fail in our setting. This is in sharp contrast with the strong performance of these methods when *only* y is corrupted. Moreover, even a natural brute force algo-

rithm has limited performance. On the positive side, we propose a general framework for high-dimensional robust regression, based on a simple idea: since global outlier rejection in high dimensions is (generally) impossible, we do it locally in low dimensions, by replacing the key vector operations used by all algorithms, the inner product, with its robust counterpart: the trimmed inner product (we define this precisely below). The idea is simple, and while the trimming operation is non-convex, it is computationally tractable. We consider three popular algorithms for sparse recovery: Thresholding Regression, Lasso, and the Dantzig selector. We show how our idea applies to each, and then analyze the resulting robust counterparts of these three algorithms. Our main theorems provide bounds on the number of corrupted points each can tolerate, while still guaranteeing support recovery and/or small ℓ_2 errors. In particular, all three algorithms are guaranteed to have small ℓ_2 errors in the setting where both response variables and covariate variables are arbitrarily corrupted; we are unaware of any other algorithm with such guarantees in this high-dimensional and arbitrary-error-in-variable regime.

2. Problem Setup

We consider the problem of sparse linear regression. The unknown parameter $\beta^* \in \mathbb{R}^p$ is assumed to be k-sparse ($k < p$), i.e., has only k nonzeros. The observations take the form of covariate-response pairs $(x_i, y_i) \in \mathbb{R}^p \times \mathbb{R}$, $i = 1, \ldots, n + n_1$. These pairs, if not corrupted, would obey the following linear model

$$y_i = \langle x_i, \beta^* \rangle + e_i;$$

here e_i is additive noise and $p \geq n$. The actual observed pairs are corrupted by one of the models below.

Definition (Row Corruption)**.** *Out of these $n + n_1$ pairs, up to n_1 of them are arbitrarily corrupted, with* both x_i *and* y_i *being potentially corrupted.*

Definition (Distributed Corruption)**.** *We allow arbitrary corruption of* any n_1 *elements of each column of the covariate matrix X and of the response y.*

In particular, the corrupted entries need not lie in the same n_1 rows in the second model. Clearly this includes the first model as a special case.

Note that in both models, we impose *no assumption whatsoever on the corrupted pairs.* They might be unbounded, non-stochastic, and even dependent on the authentic samples. They are unconstrained other than in their cardinality – the number of rows or coefficients corrupted. We illustrate both of these corruption models pictorially in Figure 1.

Goal: Given these observations $\{(x_i, y_i)\}$, the goal is to obtain a reliable estimate $\hat{\beta}$ of β^* with bounded error $\left\|\hat{\beta} - \beta^*\right\|_2$ and correct support. A fundamental question, therefore, is to understand in each given model, given p, n, and k, how many outliers (n_1) can an estimator handle?

As we show in Sections 4 and 5, models that consider corruption only in y are fundamentally easier, and in particular, algorithms successful in that regime fail in the more difficult one we consider. Note that in the distributed corruption model, an equivalent model with corruptions only in y might require all entries in y to be corrupted – an absurd setting to hope for a solution.

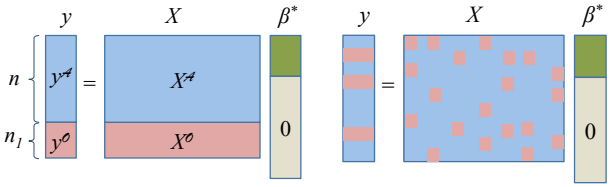

Figure 1. Illustrations of the row corruption model (left) and the distributed corruption model (right).

3. Related Work

Under the high-dimensional setting $p \geq n$, there is a large body of literature on sparse recovery when there is no corruption. It is now well-known that recovery of β^* is possible only when the covariate matrix X satisfies certain conditions, such as the Eigenvalue Property (Bickel et al., 2009). Various ensembles of random matrices are shown to satisfy these conditions with high probability. Many estimators have been proposed, most notably Basis Pursuit (a.k.a. Lasso) (Tibshirani, 1996; Donoho et al., 2006), which solves an ℓ_1-regularized least squares problem

$$\min_{\beta} \quad \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 ,$$

as well as Orthogonal Matching Pursuit (OMP) e.g., (Tropp, 2004), which is a greedy algorithm that estimates the support of β^* sequentially. Both Lasso and OMP, as well as many other estimators, are guaranteed to recover β^* with good accuracy when X is well-conditioned, and the number of observations satisfies $n \gtrsim k \log p$. (Here we mean there exists a constant c, independent of k, n, p, such that the statement holds. We use this notation throughout the paper.)

Most existing methods are not robust to outliers; for example, standard Lasso and OMP fail even if only one entry of X or y is corrupted. One might consider

a natural modification of Lasso in the spirit of Total Least Squares, and solve

$$\min_{\beta,E} \|(X-E)\beta - y\|_2^2 + \lambda\|\beta\|_1 + \eta\|E\|_*, \qquad (1)$$

where E accounts for corruption in the covariate matrix, and $\|\cdot\|_*$ is a norm. When E is known to be row sparse (as is the case in our row-corruption model), one might choose $\|\cdot\|_*$ to be $\|\cdot\|_{1,2}$ or $\|\cdot\|_{1,\infty}$[1]; the work in (Zhu et al., 2011) considers using $\|\cdot\|_* = \|\cdot\|_F$ (similar to TLS), which is more suitable when E is dense yet bounded. The optimization problem (1) is, however, non convex due to the bilinear term $E\beta$, and we are not aware of any tractable algorithm with provable performance guarantees.

Another modification of Lasso accounts for the corruption in the response via an additional variable z (Laska et al., 2009; Li, 2011; She & Owen, 2010):

$$\min_{\beta,z} \|X\beta - y - z\|_2^2 + \lambda\|\beta\|_1 + \gamma\|z\|_1. \qquad (2)$$

We refer to this approach as Justice Pursuit (JP) after (Laska et al., 2009). Unlike the previous approach, the problem in (2) is convex. In fact, it is the natural convexification of the brute force algorithm:

$$\begin{aligned} \min_{\beta,z} \quad & \|X\beta - y - z\|_2 \\ \text{s.t.} \quad & \|\beta\|_0 \le k, \|z\|_0 \le n_1, \end{aligned} \qquad (3)$$

where $\|u\|_0$ denotes the number of nonzero entries in u. It is easy to see (and well known) that the so-called Justice Pursuit relaxation (2) is equivalent to minimizing the Huber loss function plus the ℓ_1 regularizer, with an explicit relation between γ and the parameter of the Huber loss function (Fuchs, 1999). Formulation (2) has excellent recovery guarantees when *only* the response variable is corrupted, delivering exact recovery under a constant fraction of outliers. However, we show in the next section that a broad class of convex optimization-based approaches, with (2) as a special case, fail when the covariate matrix X is also corrupted. In the subsequent section, we show that even the original brute force formulation is problematic: while it can recover from some number n_1 of corrupted rows, that number is order-wise worse than what our algorithms guarantee. Neither the brute force algorithm above, nor its relaxation, JP, are appropriate for our second model for *distributed corruption*.

For standard linear regression problems in the classical scaling $n \gg p$, various robust estimators have been proposed, including M-, R-, and S-estimators (Huber, 1981; Maronna et al., 2006), as well as those based on ℓ_1-minimization (Kekatos & Giannakis, 2011). Many of these estimators lead to non-convex optimization problems, and even for those that are convex, it is unclear how they can be used in the high-dimensional scaling with sparse β^*. Another difficulty in applying classical robust methods to our problems arises from the fact that the covariates, x_i, also lie in a high-dimensional space, and thus defeat many outlier detection techniques that might otherwise work well in low-dimensions. Again, for our second model of corruption, outlier detection seems even more hopeless.

[1] $\|E\|_{1,2}$ ($\|E\|_{1,\infty}$) is the sum of the ℓ_2 (ℓ^∞, respectively) norms of the rows of E.

4. Failure of the Convex Optimization Approach

We consider a broad class of convex optimization-based approaches of the following form:

$$\min_{\beta} \quad f(y - X\beta), \quad \text{s.t. } h(\beta) \le R.$$

Here R is a radius parameter that can be tuned. Both $f(\cdot)$ and $h(\cdot)$ are convex functions, which can be interpreted as a loss function (of the residual) and a regularizer (of β), respectively. For example, one may take $f(v) = \min_z \|v - z\|_2^2 + \gamma\|z\|_1$ and $h(\beta) = \|\beta\|_1$, which recovers the Justice Pursuit (2) by Lagrangian duality; note that this $f(v)$ is convex by (Fuchs, 1999). The function $f(\cdot)$ can also be any other robust convex loss function including the Huber loss function.

We assume that $f(\cdot)$ and $h(\cdot)$ obey a very mild condition, which is satisfied by any non-trivial loss function and regularizer that we know of. In the sequel we use $[z_1; z_2]$ to denote the concatenation of two column vectors z_1 and z_2.

Definition (Standard Convex Optimization (SCO) Condition)**.** *We say $f(\cdot)$ and $h(\cdot)$ satisfy the SCO Condition if $\lim_{\alpha\to\infty} f(\alpha v) = \infty$ for all $v \neq 0$, $f([v_1; v_2]) \ge f([0; v_2])$ for all v_1, v_2, and $h(\cdot)$ is invariant under permutation of coordinates.*

We also assume $R \ge h(\beta^*)$ because otherwise the formulation is not consistent even when there are no outliers. The following theorem shows that under this assumption, the convex optimization approach fails when both X and y are corrupted. We only show this for our first corruption model, since it is a special case of the second distributed model. As illustrated in Figure 1, let $\mathcal{A}$ and $\mathcal{O}$ be the (unknown) sets of indices corresponding to authentic and corrupted observations, respectively, and $X^{\mathcal{A}}$ and $X^{\mathcal{O}}$ be the authentic and corrupted rows of the covariate matrix

$X = [x_1, \ldots, x_{n+n_1}]^\top$. The vectors $y^{\mathcal{A}}$ and $y^{\mathcal{O}}$ are defined similarly. Also let Λ^* be the support of β^*. With this notation, we have the following.

Theorem 1. *Suppose f and h satisfy the SCO Condition. When $n_1 \geq 1$, the adversary can corrupt X and y in such a way that for all R with $R \geq h(\beta^*)$, any optimal solution does not have the correct support.*

The proof is given in the supplement. Our proof proceeds by using a simple corruption strategy. Certainly, there are natural approaches to deal with this specific example, e.g., removing entries of X with large values. But discarding such large-value entries is not enough, as there may exist more sophisticated corruption schemes where simple magnitude-based clipping is ineffective. We illustrate this with a concrete example in the simulation section, where Justice Pursuit along with large-value-trimming fails to recover the correct support. Indeed, this example serves merely to illustrate more generally the inadequacy of a purely convex-optimization-based approach.

More importantly, while the idea of considering an unbounded outlier is not new and has been used in classical robust statistics and more recently in (Yu et al., 2012), the above theorem highlights the sharp contrast between the success of convex optimization (e.g., JP) under corruption in only y, and its complete failure when both X and y are corrupted. Corruptions in X not only break the linear relationship between y and X, but also destroy properties of X necessary for existing sparse regression approaches. In the high dimensional setting where support recovery is concerned, there is a fundamental difference between the hardness of the two corruption models.

5. The Natural Brute Force Algorithm

The brute force algorithm (3) can be restated as: it considers all $n \times k$ submatrices of X and picks the one that gives the smallest regression error w.r.t. the corresponding subvector of y. Formally, let $X_\Lambda^{\mathcal{S}}$ denote the submatrix of X corresponding to row indices $\mathcal{S}$ and column indices Λ, and let $y^{\mathcal{S}}$ denote the subvector of y corresponding to indices $\mathcal{S}$. The algorithm solves

$$\min_{\theta \in \mathbb{R}^k, \mathcal{S}, \Lambda} \quad \left\| y^{\mathcal{S}} - X_\Lambda^{\mathcal{S}} \theta \right\|_2 \qquad (4)$$
$$\text{s.t.} \quad |\mathcal{S}| = n, |\Lambda| = k.$$

Suppose the optimal solution is $\hat{\mathcal{S}}, \hat{\Lambda}, \hat{\theta}$. Then, the algorithm outputs $\hat{\beta}$ with $\hat{\beta}_{\hat{\Lambda}} = \hat{\theta}$ and $\hat{\beta}_{\hat{\Lambda}^c} = 0$. Note that this algorithm has exponential complexity in n and k, and $\mathcal{S}^c$ can be considered as an operational definition of outliers. We show that even this algorithm has poor performance and cannot handle large n_1.

To this end, we consider the simple Gaussian design model, where the entries of $X^{\mathcal{A}}$ and e are independent zero-mean Gaussian random variables with variance $\frac{1}{n}$ and $\frac{\sigma_e}{n}$, respectively. The $\frac{1}{n}$ factor is simply for normalization and no generality is lost. We consider the setting where $\sigma_e^2 = k$ and $\beta^*_{\Lambda^*} = [1, \ldots, 1]^\top$. If $n_1 = 0$, existing methods (e.g., Lasso and standard OMP), and the brute force algorithm as well, can recover the support of β^* with high probability provided $n \gtrsim k \log p$. Here and henceforth, by *with high probability* we mean with probability at least $1 - p^{-2}$. However, when there are outliers, we have the following negative result.

Theorem 2. *Under the above setting, if $n \gtrsim k^3 \log p$ and $n_1 \gtrsim \frac{3n}{k+1}$, then with probability at least $1 - p^{-2}$ the adversary can corrupt X and y to make the brute force algorithm fail to output the correct support Λ^*.*

The proof is given in the supplement. We believe the condition $n \gtrsim k^3 \log p$ is an artifact of our proof and is not necessary. This theorem shows that the brute force algorithm can only handle $O\left(\frac{n}{k}\right)$ outliers. In the next section, we propose a simple, tractable algorithm that *outperforms* this brute force algorithm and can handle $O\left(\frac{n}{\sqrt{k}}\right)$ outliers.

6. Robust Algorithms for Sparse Regression

As described above, standard tools such as convexity alone, or outlier rejection, do not fare well. Our approach does not try to accurately identify outliers; instead, we replace standard computations with robust counterparts less susceptible to manipulation. In particular, we replace the inner product with a more robust version: the trimmed inner product. While simple, this is the corner stone to our results, and we describe it in Algorithm 1.

Algorithm 1 Trimmed Inner Product $\langle a, b \rangle_{n_1}$

Input: $a \in \mathbb{R}^N$, $b \in \mathbb{R}^N$, n_1
Compute $q_i = a_i b_i$, $i = 1, \ldots, N$.
Sort $\{|q_i|\}$ and select the smallest $(N - n_1)$ ones.
Let Ω be the set of selected indices.
Output: $h = \sum_{i \in \Omega} q_i$.

The next sections show how this simple idea can take non-robust algorithms, and yield tractable algorithms with provable robustness properties.

6.1. Robust Thresholding Regression

The first algorithm we consider is Thresholding Regression (a.k.a. Sure Screening, and Marginal Regression). Standard TR estimates the support of β^* by

selecting the columns of X which have large (in absolute value) inner products with the response vector y. If the sparsity level k of β^* is known, then one may select the top k columns. To successfully recover the support of β^*, standard TR relies on the fact that for well-conditioned X, the inner product $h(j) = \langle y, X_j \rangle$ is close to β_j. When outliers are present, TR fails because the $h(j)$'s may be distorted significantly by maliciously corrupted x_i's and y_i's. To protect against outliers, we compute $h(j)$ using the more robust trimmed inner product. This leads to our Robust Thresholding Regression algorithm (RoTR) (Algorithms 2).

Algorithm 2 Robust Thresholding Regression

Input: X, y, k, n_1.
For $j = 1, \ldots, p$, compute $h(j) = \langle y, X_j \rangle_{n_1}$.
Sort $\{|h(j)|\}$ and select the k largest ones.
Let $\hat{\Lambda}$ be the set of selected indices.
Set $\hat{\beta}_j = h(j)$ for $j \in \hat{\Lambda}$ and 0 otherwise.
Output: $\hat{\beta}$

We note again that (a) this algorithm is no more computationally taxing than ordinary TR; (b) we are not performing outlier detection (i.e., identifying corrupted rows in X and y)– rather, mitigating the strength of the adversary to skew each step.

Our algorithm requires two parameters, n_1 and k. We discuss how to choose them after we present the performance guarantees below.

6.1.1. Performance Guarantees for RoTR

We are interested in finding conditions for (p, k, n, n_1) under which RoTR is guaranteed to recover β^* with correct support and small error. We consider the following sub-Gaussian design model. Recall that a random variable Z is sub-Gaussian with parameter σ if $\mathbb{E}[\exp(tZ)] \leq \exp(t^2\sigma^2/2)$ for all real t.

Definition. *We say that a random matrix $X \in \mathbb{R}^{n \times p}$ is sub-Gaussian with parameter $(\frac{1}{n}\Sigma, \frac{1}{n}\sigma^2)$ if:*

1) each row $x_i^\top \in \mathbb{R}^p$ is sampled independently from a zero-mean distribution with covariance $\frac{1}{n}\Sigma$, and

2) for any unit vector $u \in \mathbb{R}^p$, the random variable $u^\top x_i$ is sub-Gaussian with parameter $\frac{1}{\sqrt{n}}\sigma$.

Definition (Sub-Gaussian design)**.** *We assume the true design matrix X, before corruption, is sub-Gaussian with parameter $(\frac{1}{n}\Sigma_x, \frac{1}{n}\sigma_x^2)$, and the additive noise e is sub-Gaussian with parameter $\frac{1}{n}\sigma_e^2$.*

Note that the sub-Gaussian model covers the case of Gaussian, Bernoulli, and any other distributions with bounded support. For RoTR, we consider the special case with independent columns, i.e., $\Sigma_x = I$.

The following theorem (proof in supplement) characterizes the performance of RoTR, and shows that it can recover the correct support even when the number of outliers scales with n. In particular, this shows RoTR can tolerate an $O(1/\sqrt{k})$ fraction of distributed or row-wise outliers.

Theorem 3. *Under sub-Gaussian design with $\Sigma_x = I$ and the row or distributed corruption model, the following hold with probability at least $1 - p^{-2}$.*

(1) The output of RoTR satisfies the ℓ_2 bound:

$$\left\|\hat{\beta} - \beta^*\right\|_2 \lesssim \|\beta^*\|_2 \sqrt{1 + \frac{\sigma_e^2}{\|\beta^*\|_2^2}} \left(\sqrt{\frac{k \log p}{n}} + \frac{n_1 \sqrt{k} \log p}{n} \right).$$

(2) If the nonzero entries of β^* satisfy $|\beta_j^*|^2 \geq \left(\|\beta^*\|_2^2/n\right) \log p \left(1 + \sigma_e^2 / \|\beta^*\|_2^2\right)$, then RoTR correctly identifies the support of β^* provided

$$\begin{aligned} n &\gtrsim k \log p \cdot \left(1 + \sigma_e^2 / \|\beta^*\|_2^2\right), \text{ and} \\ \frac{n_1}{n} &\lesssim 1 / \left(\sqrt{k \left(1 + \sigma_e^2 / \|\beta^*\|_2^2\right)} \log p \right). \end{aligned}$$

In particular, our algorithm is order wise stronger than the brute force algorithm, in terms of the number of outliers it can tolerate while still correctly identifying the support (compared with Theorem 2). A few remarks are in order.

1. We emphasize that **knowledge of the exact number of outliers is not needed** – n_1 can be any **upper bound** of the number of outliers. The theorem holds even if there are less than n_1 outliers. Of course, this would result in sub-optimal bounds in the estimation due to over-conservativeness. In practice, cross-validation could be useful here.

2. We note that essentially all robust statistical procedures we are aware of have the same character noted above. This is true even for the simplest algorithms for robustly estimating the mean. If an upper bound is known on the fraction of corrupted points, one computes the analogous trimmed mean. Otherwise, one can simply compute the median, and the result will have controlled error (but will be suboptimal) as long as the number of corrupted points is less than 50% – something which, as in our case, and every case, is always impossible to know simply from the data.

3. In a similar spirit, the requirement to know k can also be relaxed. For example, if we use some $k' > k$ instead of k, then the theorem continues to hold in the sense that RoTR identifies a superset (with size k') of the support of β^*, and the ℓ_2 error bound holds with

k replaced by k'. In addition, standard procedures of estimating the sparsity level (e.g., cross-validation) may potentially be applied in our setting.

6.2. Robust Dantzig Selector and Lasso

We now consider the robustified versions of the Dantzig selector and Lasso, given in Algorithm 3 and 4, respectively. In both, the key difference is the use of the trimmed inner product.

Algorithm 3 Robust Dantzig Selector

Input: X, y, μ, τ, n_1.
Compute for all $i, j = 1, \ldots, p$,

$$\hat{\Gamma}_{ij} = \langle X_i, X_j \rangle_{n_1}, \quad \hat{\gamma}_j = \langle X_j, y \rangle_{n_1}. \tag{5}$$

Use linear programming to solve and output:

$$\hat{\beta} = \arg\min_{\beta} \|\beta\|_1 \quad \text{s.t.} \quad \left\|\hat{\Gamma}\beta - \hat{\gamma}\right\|_\infty \le \mu \|\beta\|_1 + \tau.$$

Algorithm 4 Robust Lasso

Input: X, y, R, n_1.
Compute $\hat{\Gamma}$ and $\hat{\gamma}$ using (5).
Use Projected Gradient Descent to solve and output:

$$\hat{\beta} = \arg\min_{\beta} \frac{1}{2}\beta^\top \hat{\Gamma}\beta - \hat{\gamma}^\top \beta \quad \text{s.t.} \quad \|\beta\|_1 \le R.$$

Note the $\|\beta\|_1$ term on the R.H.S. of the constraint. It accounts for the effect of the corruption in X being multiplied by β; a similar formulation appears in (Rosenbaum & Tsybakov, 2010; 2011) under a different context. The optimization in Robust Lasso is non-convex because $\hat{\Gamma}$ might have negative eigenvalues. Nevertheless, we can still use the following projected gradient descent method (Loh & Wainwright, 2012):

$$\beta^{t+1} = \mathcal{P}_R\left(\beta^t - (1/\eta)(\hat{\Gamma}\beta^t - \hat{\gamma})\right);$$

here $\mathcal{P}_R$ is the ℓ_2-projection onto the ℓ_1-ball of radius R, and η is the step size. The theoretical guarantees below hold for the output of projected gradient descent as well (see the supplementary material for details).

6.2.1. Performance Guarantees

We have the following guarantees for Robust Dantzig selector and Lasso. Here we allow for a general Σ_x.

Theorem 4. *Under the sub-Gaussian Design Model, suppose the following is satisfied:*

$$n \gtrsim \frac{\sigma_x^4}{\lambda_{\min}^2(\Sigma_x)} k \log p, \qquad \frac{n_1}{n} \lesssim \frac{\lambda_{\min}(\Sigma_x)}{\sigma_x^2 k \log p}.$$

If we choose the following parameters:

1. *for robust Dantzig selector:* $\mu = 16\frac{n_1 \log p}{n}\sigma_x^2$ *and* $\tau = 16\sqrt{\frac{\sigma_e^2 \log p}{n}} + 16\frac{n_1 \log p}{n}\sigma_x^2\sqrt{\sigma_e^2 + \sigma_x^2 \|\beta^*\|_2^2}$;
2. *for robust Lasso:* $R = \|\beta^*\|_1$;

then with probability at least $1 - p^{-2}$*, the outputs of Robust Dantzig selector and robust Lasso both satisfy the following* ℓ_2 *and* ℓ_1 *error bounds*

$$\frac{1}{2\sqrt{k}}\left\|\hat{\beta} - \beta^*\right\|_1 \le \left\|\hat{\beta} - \beta^*\right\|_2 \lesssim \frac{1}{\lambda_{\min}(\Sigma_x)}\left(\sigma_e\sqrt{\frac{k \log p}{n}} + \frac{k n_1 \log p}{n}\sigma_x^2 \|\beta\|_2 + \frac{n_1 \log p \sqrt{k}}{n}\sigma_x\sqrt{\sigma_e^2 + \sigma_x^2 \|\beta^*\|_2^2}\right).$$

Some remarks are in order:

(1) Both algorithms require some tuning parameters (μ, τ, and R), for which the theorem gives the "optimal" values, in the sense that we get the best possible bounds. But this requires knowing the statistics of the noises (σ_e and n_1), the true design matrix (σ_x), and the true solution ($\|\beta^*\|_2$ or $\|\beta^*\|_1$). If we set these parameters larger than their optimal values, then we can still get errors bounds, but they will be sub-optimal; we omit the details here due to space constraint. Note that the same is true for standard Dantzig selector and Lasso, and their modified versions in (Rosenbaum & Tsybakov, 2011; Loh & Wainwright, 2012).

(2) If $\Sigma_x = I$, then, in order for the ℓ_2 error to be bounded, the requirement for n_1 is worse than RoTR by a factor of $\sqrt{k}$ (and is the same as requirement for support recovery for the brute force algorithm). This is because Robust Dantzig and Lasso do not use the knowledge of Σ_x being diagonal when constructing $\hat{\Sigma}$.

(3) If we use $\hat{\Gamma} = I$ in the above case, or more generally $\hat{\Gamma} = \mathbb{E}[XX^\top]$, it is easy to prove that Robust Lasso essentially becomes (an ℓ_1 relaxation of) RoTR.

7. Experiments

We report some results for RoTR, robust Dantzig selector, and Lasso on synthetic data. The performance

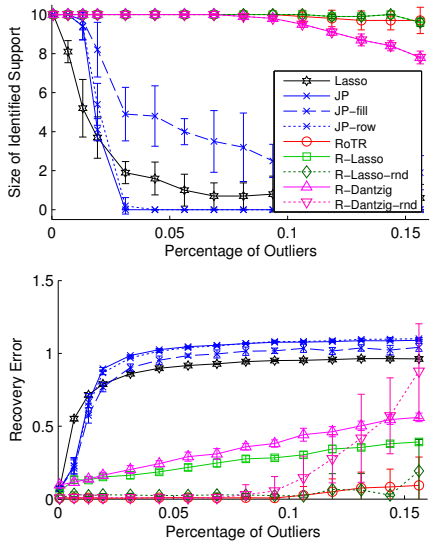

Figure 2. Support recovery and relative ℓ_2 recovery errors for different methods with independent columns of X. The error bars correspond to one standard deviation. RoMP, R-Lasso and R-Dantzig significantly outperform all other methods. Note that the ℓ_2 error of the other methods flatten out because they returns a near-zero solution.

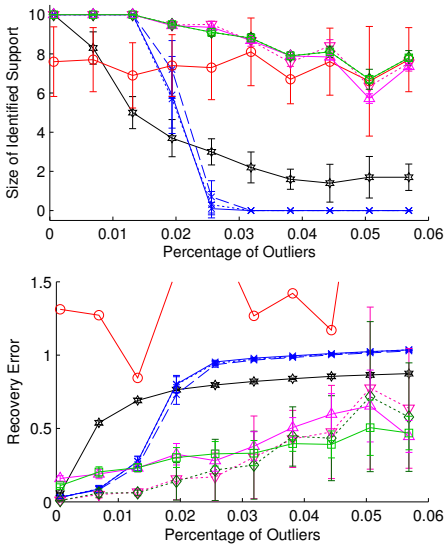

Figure 3. Support recovery and relative ℓ_2 recovery errors for different methods with correlated columns of X. The legends are the same as Figure 2. The error bars of RoMP are too large and thus not shown. R-Lasso and R-Dantzig outperform RoMP and the other methods.

is measured by ℓ_2-error ($\|\hat{\beta}-\beta^*\|_2/\|\beta^*\|_2$) and support recovery (the number of non-zero locations of β^* that are correctly identified). For the robust Dantzig selector and Lasso, we estimate the support using the locations of the largest k entries of $\hat{\beta}$. We also consider an refinement of robust Dantzig selector/Lasso (dubbed R-Dantzig-rnd/Lasso-rnd) where we re-calculate the entries in the estimated support by least squares using $\hat{\Gamma}$ and $\hat{\gamma}$, and set all the other entries to zero.

For comparison, we apply standard Lasso and JP (Laska et al., 2009; Li, 2011) to the same data. We search for the values of the tradeoff parameters λ, γ that yield the smallest ℓ_2-errors. Furthermore, we test JP with two different pre-processing procedures, both of which aim to detect and correct the corrupted entries in X directly. The first one, dubbed JP-fill, finds the set E of the largest $\frac{n_1}{n}$ portion of the entries of X, and then scales them to have unit magnitude. The second one, dubbed JP-row, discards the n_1 rows of X that contain the most entries in E.

We first consider the case where X has independent columns. The authentic rows $(X^{\mathcal{A}}, y^{\mathcal{A}})$ are generated under the sub-Gaussian Design model using Gaussian distribution with $\Sigma_x = I$, $p = 4000, n = 1600, k = 10$ and $\sigma_e = 2$, with the non-zero elements of β^* being random ± 1. The corrupted rows $(X^{\mathcal{O}}, y^{\mathcal{O}})$ are generated by following procedure: Let $\theta^* = \arg\min_{\theta \in \mathbb{R}^{p-k}: \|\theta\|_1 \le \|\beta^*\|_1} \|y^{\mathcal{A}} - X^{\mathcal{A}}_{(\Lambda^*)^c}\theta\|_2$. Set $X^{\mathcal{O}}_{\Lambda^*} = \frac{3}{\sqrt{n}}A$, where A is a random ± 1 matrix of dimension $n_1 \times k$, and $y^{\mathcal{O}} = X^{\mathcal{O}}_{\Lambda^*}(-\beta^*)$. For $i = 1, \ldots, n_1$, further set $X^{\mathcal{O}}_{i,(\Lambda^*)^c} = \left(y^{\mathcal{O}}_i / (B_i^\top \theta^*)\right) \cdot B_i^\top$, where B_i is a $(n-k)$-vector with i.i.d. standard Gaussian entries.

The results are shown in Figure 2. One observes that RoTR, robust Dantzig selector and Lasso all perform better than Lasso and JP for both metrics, especially when the number of outliers is large. The ℓ_2-errors of Lasso and JP flatten out because they return near-zero solutions. Pre-processing procedures do not significantly improve performance of JP, which highlights the difficulty of outlier detection in high dimensions.

We next consider X with correlated columns. The data is generated using $\sigma_e = 1$, $(\Sigma_x)_{ii} = 1$ for all i, and $(\Sigma_x)_{ij} = 0.4$ for all $i \neq j$; other parameters are the same as before. The results are shown in Figure 3. The output of RoTR becomes unreliable (as expected), but robust Dantzig selector and robust Lasso remain stable and compare favorably with the other methods.

Acknowledgments

C. Caramanis and Y. Chen were supported by NSF Grant EECS-1056028 and DTRA grant HDTRA 1-08-0029. S. Mannor was partially supported by the Israel Science Foundation under grant No 920/12.

References

Bickel, P.J., Ritov, Y., and Tsybakov, A.B. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009.

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

Candes, E.J., Li, X., Ma, Y., and Wright, J. Robust principal component analysis? *Arxiv preprint arXiv:0912.3599*, 2009.

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

Chen, S.S., Donoho, D.L., and Saunders, M.A. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1999.

Chen, Y. and Caramanis, C. Noisy and missing data regression: Distribution-oblivious support recovery. In *ICML*, 2013.

Chen, Y., Xu, H., Caramanis, C., and Sanghavi, Sujay. Robust matrix completion with corrupted columns. In *ICML*, 2011.

Donoho, D. L. Breakdown properties of multivariate location estimators, qualifying paper, Harvard University, 1982.

Donoho, D.L., Elad, M., and Temlyakov, V.N. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory*, 52(1):6–18, 2006.

Fuchs, J.J. An inverse problem approach to robust regression. In *Proceedings of ICASSP*, volume 4, pp. 1809–1812. IEEE, 1999.

Hampel, F.R., Ronchetti, E.M., Rousseeuw, P.J., and Stahel, W.A. *Robust statistics: the approach based on influence functions*, volume 114. Wiley, 1986.

Herman, M.A. and Strohmer, T. General deviants: An analysis of perturbations in compressed sensing. *IEEE Journal of Selected Topics in Signal Processing*, 4(2):342–349, 2010.

Huber, P. *Robust Statistics*. Wiley, New York, 1981.

Kekatos, V. and Giannakis, G.B. From sparse signals to sparse residuals for robust sensing. *IEEE Transactions on Signal Processing*, 59(7), 2011.

Laska, J.N., Davenport, M.A., and Baraniuk, R.G. Exact signal recovery from sparsely corrupted measurements through the pursuit of justice. In *Asilomar Conference on Signals, Systems & Computers*, 2009.

Lerman, G., McCoy, M., Tropp, J.A., and Zhang, T. Robust computation of linear models, or how to find a needle in a haystack. *arXiv:1202.4044*, 2012.

Li, Xiaodong. Compressed sensing and matrix completion with constant proportion of corruptions. *Arxiv preprint arXiv:1104.1041*, 2011.

Loh, P.L. and Wainwright, M.J. High-dimensional regression with noisy and missing data: Provable guarantees with non-convexity. *Annals of Statistics*, 40(3):1637–1664, 2012.

Maronna, R.A., Martin, R.D., and Yohai, V.J. *Robust statistics*. Wiley, 2006.

Nguyen, N.H., Tran, T., et al. Exact recoverability from dense corrupted observations via l_1 minimization. *Arxiv preprint arXiv:1102.1227*, 2011.

Rosenbaum, M. and Tsybakov, A.B. Sparse recovery under matrix uncertainty. *The Annals of Statistics*, 38(5):2620–2651, 2010.

Rosenbaum, M. and Tsybakov, A.B. Improved matrix uncertainty selector. *arXiv:1112.4413*, 2011.

She, Y. and Owen, A. B. Outlier Detection Using Nonconvex Penalized Regression. *arXiv:1006.2592*, 2010.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

Tropp, J.A. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004.

Xu, H., Caramanis, C., and Sanghavi, S. Robust PCA via outlier pursuit. *IEEE Transactions on Information Theory*, 58(5):3047–3064, 2012.

Xu, H., Caramanis, C., and Mannor, S. Outlier-Robust PCA: The High Dimensional Case. *IEEE Transactions on Information Theory*, 59(1), 2013.

Yu, Y., Aslan, O., and Schuurmans, D. A polynomial-time form of robust regression. In *NIPS*, 2012.

Zhu, H., Leus, G., and Giannakis, G.B. Sparsity-cognizant total least-squares for perturbed compressive sampling. *IEEE Transactions on Signal Processing*, 59(5):2002–2016, 2011.