Robust Regression via Heuristic Hard Thresholding

Xuchao Zhang[†], Liang Zhao[‡], Arnold P. Boedihardjo[§], Chang-Tien Lu[†]
[†]Virginia Tech, Falls Church, VA, USA
[‡]George Mason University, Fairfax, VA, USA
[§]U. S. Army Corps of Engineers, Alexandria, VA, USA
[†]{xuczhang, ctlu}@vt.edu, [‡]lzhao9@gmu.edu, [§]arnold.p.boedihardjo@usace.army.mil

Abstract

The presence of data noise and corruptions recently invokes increasing attention on Robust Least Squares Regression (*RLSR*), which addresses the fundamental problem that learns reliable regression coefficients when response variables can be arbitrarily corrupted. Until now, several important challenges still cannot be handled concurrently: 1) exact recovery guarantee of regression coefficients 2) difficulty in estimating the corruption ratio parameter; and 3) scalability to massive dataset. This paper proposes a novel Robust Least squares regression algorithm via Heuristic Hard thresholding (*RLHH*), that concurrently addresses all the above challenges. Specifically, the algorithm alternately optimizes the regression coefficients and estimates the optimal uncorrupted set via heuristic hard thresholding without corruption ratio parameter until it converges. We also prove that our algorithm benefits from strong guarantees analogous to those of state-of-the-art methods in terms of convergence rates and recovery guarantees. Extensive experiment demonstrates that the effectiveness of our new method is superior to that of existing methods in the recovery of both regression coefficients and uncorrupted sets, with very competitive efficiency.

1 Introduction

The presence of noises and corruptions in real-world data can be inevitably caused by the experimental errors, accidental outliers, or even adversarial data attacks. As the traditional least squares regression methods are vulnerable to outlier observations [Maronna *et al.*, 2006], we study Robust Least Square Regression (*RLSR*) to handle the problem of learning a reliable set of regression coefficients given the presence of several adversarial corruptions in its response vector. A commonly adopted model from existing methods assumes that the observed response is obtained from the generative model $\boldsymbol{y} = X^T\boldsymbol{\beta}^* + \boldsymbol{u}$, where $\boldsymbol{\beta}^*$ is the true regression coefficients that we wish to recover and $\boldsymbol{u} \in \mathcal{R}^n$ is the corruption vector with arbitrary values. Due to the ubiquitousness of data corruptions and popularity of regression methods, *RLSR* has become a critical component of several important real-world applications in various domains such as signal processing [Zoubir *et al.*, 2012; Studer *et al.*, 2012], economics [Rousseeuw and Leroy, 2005], bioinformatics [Lourenco *et al.*, 2011] and image processing [Naseem *et al.*, 2012; Wright *et al.*, 2009]. A large body of literature on robust regression problem has been built up over the last few decades, but most of them [Smolic and Ohm, 2000; Huber and Ronchetti, 2009; She and Owen, 2011; Jung *et al.*, 2016] lack the theoretical guarantee of regression coefficients recovery. To theoretically guarantee the recovery performance, [Chen *et al.*, 2013] proposed a trimmed inner product based algorithm, but the recovery boundary of their method is not tight in a massive dataset. Also, [Mcwilliams *et al.*, 2014] proposed a sub-sampling algorithm for large scale corrupted linear regression, the recovery result they provide is not close to a exact recovery [Bhatia *et al.*, 2015]. To pursuit the exact recovery results for *RLSR* problem, some work focused on L_1 penalty based convex formulations [Wright and Ma, 2010; Nguyen and Tran, 2013]. However, these methods imposed severe restrictions on the data distribution such as row-sampling from an incoherent orthogonal matrix [Nguyen and Tran, 2013].

Several studies requires the corruption ratio parameter which is difficult to be determined manually. For instance, instead of the exact corruption ratio, [Chen *et al.*, 2013] requires the upper bound of outliers number which is also difficult to estimate. [She and Owen, 2011] relies on a regularization parameter to control the size of uncorrupted set based on soft-thresholding. Recently, [Bhatia *et al.*, 2015] proposed a hard-thresholding algorithm for the *RLSR* problem: $\arg\min \sum_{i\in S}(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta})^2$ with the constraint: $|S| \geq (1-\gamma)n$, where $|S|$ is the size of the uncorrupted set and γ is the corruption ratio. Although the method guarantees an exact recovery of $\boldsymbol{\beta}$ under a mild assumption for covariate matrix, their results are highly dependent on the corruption ratio parameter γ inputed by users. Specifically, the parameter is required to be larger than the exact corruption ratio γ^* to ensure its convergence. Unfortunately, it is seldom practical to estimate the corruption ratio under the assumption that the response vector is arbitrarily corrupted. Furthermore, empirical results show that if the parameter γ is more than 50% off the true value, the recovery error can be more than double in size.

To address the lack of rigorous analysis on corruption ratio and theoretical guarantee on exact recovery, we proposed a new model, Robust Least squares regression algorithm via

Heuristic Hard thresholding (*RLHH*). The main contributions of our study are summarized as follows: 1) *The design of an efficient algorithm to address the RLSR problem without parameterizing its corruption.* The algorithm *RLHH* is proposed to recover the regression coefficients and uncorrupted set efficiently. Unlike with a fixed corruption ratio, our method alternately estimates the optimal corruption ratio based on residual errors using optimized regression coefficients in each iteration. 2) *An exact recovery guarantee under a mild assumption regarding input variables.* We prove that our *RLHH* algorithm converges at a geometric rate and recovers $\boldsymbol{\beta}^*$ exactly under the assumption that the least squares function satisfies both the Subset Strong Convexity (*SSC*) and Subset Strong Smoothness (*SSS*) properties. Specifically, we prove that our heuristic hard thresholding function ensures that the residual of the estimated uncorrupted set in each iteration has a tight upper error bound for the true uncorrupted set. 3) *Empirical effectiveness and efficiency.* Our proposed algorithm was evaluated with 6 competing methods in synthetic data. The results demonstrate that our approach consistently outperforms existing methods in both regression coefficients and uncorrupted set recovery, delivering a competitive running time.

The reminder of this paper is organized as follows. Section 2 gives a formal problem formulation. The proposed *RLHH* algorithm is presented in Section 3. Section 4 presents the proof for the recovery guarantees. In Section 5, the experimental results are analyzed and the paper concludes with a summary of our work in Section 6.

2 Problem Formulation

In this study, we consider the problem of *RLSR* with adversarially corrupted data. Given a covariate matrix $X = [\boldsymbol{x}_1, ..., \boldsymbol{x}_n]$, where each column $\boldsymbol{x}_i \in \mathcal{R}^{p\times 1}$ and $\boldsymbol{\beta}^*$ represents the ground truth coefficients of the regression model, we assume the corresponding response vector $\boldsymbol{y} \in \mathcal{R}^{n\times 1}$ is generated using the following model:

$$\boldsymbol{y} = \boldsymbol{y}^* + \boldsymbol{u} + \boldsymbol{\varepsilon} \tag{1}$$

where $\boldsymbol{y}^* = X^T\boldsymbol{\beta}^*$ and $\boldsymbol{u}$ is the unbounded corruption vector introduced by an adversary. $\boldsymbol{\varepsilon}$ represents the additive dense noise, where $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$. The goal of our problem is to learn a new problem, which is to recover the regression coefficients $\boldsymbol{\beta}^*$ and simultaneously determine the uncorrupted point set $\hat{S}$. The problem is formally defined as follows:

$$\begin{aligned} \hat{\boldsymbol{\beta}}, \hat{S} &= \arg\min_{\boldsymbol{\beta}, S} \|\boldsymbol{y}_S - X_S^T\boldsymbol{\beta}\|_2^2 \\ s.t.\ \ & S \subset [n],\ |S| \geq \mathcal{G}(\boldsymbol{\beta}) \end{aligned} \tag{2}$$

Given a subset $S \subset [n]$, $\boldsymbol{y}_S$ restricts the row of $\boldsymbol{y}$ to indices in S and X_S signifies that the columns of X are restricted to indices in S. Therefore, we have $\boldsymbol{y}_S \in \mathcal{R}^{|S|\times 1}$ and $X_S \in \mathcal{R}^{p\times|S|}$. We use the notation $S_* = \overline{supp(\boldsymbol{u})}$ to denote the set of uncorrupted points. Also, for any vector $\boldsymbol{v} \in \mathcal{R}^n$, the notation $\boldsymbol{v}_S$ represents the $|S|$-dimensional vector containing the components in S. The function $\mathcal{G}(\cdot)$ is to determine the size of set S according to the regression coefficients $\boldsymbol{\beta}$, which is explained in Section 3.

To prove the theoretical recovery of regression coefficients, we require that the least squares function satisfies the *Subset Strong Convexity (SSC)* and *Subset Strong Smoothness (SSS)*, which are defined as follows:

Definition 1. ***SSC and SSS Property***. *The least squares function $f(\boldsymbol{\beta}) = \|\boldsymbol{y}_S - X_S^T\boldsymbol{\beta}\|_2^2$ satisfies $2\zeta_\alpha$-Subset Strong Convexity Property and $2\kappa_\alpha$-Subset Strong Smoothness if the following holds:*

$$\zeta_\alpha I \preceq \frac{1}{2}\nabla^2 f_S(\boldsymbol{\beta}) \preceq \kappa_\alpha I \quad \text{for } \forall S \in S_\alpha \tag{3}$$

Equation (3) *is equivalent to:*

$$\zeta_\alpha \leq \min_{S\in S_\alpha} \lambda_{min}(X_S X_S^T) \leq \max_{S\in S_\alpha} \lambda_{max}(X_S X_S^T) \leq \kappa_\alpha$$

where λ_{min} and λ_{max} are denoted as the smallest and largest eigenvalues of matrix X, respectively.

The optimization problem in Equation (2) is non-convex (jointly in $\boldsymbol{\beta}$ and S) in general and existing methods cannot guarantee the exact recovery and efficient convergence rate.

3 The Proposed Methodology

In order to solve the problem in Equation (2) efficiently with the guarantee on the exact recovery of regression coefficients, we propose a novel heuristic hard thresholding based robust regression algorithm, *RLHH*. The algorithm iteratively optimizes the regression coefficients $\boldsymbol{\beta}$ and uncorrupted set S until convergence. The optimization of S is very challenging because it mounts to a non-convex discrete optimization problem. To handle it, we propose a heuristic corruption estimator to determine the size of set S and then apply the estimated uncorrupted size into heuristic hard thresholding method for the optimization of S elements.

Denoting residual vector $\boldsymbol{r} = \boldsymbol{y} - X^T\boldsymbol{\beta}$ and $r_{\delta(k)}$ be the k^{th} elements of $\boldsymbol{r}$ in ascending order of magnitude, the heuristic estimator $\mathcal{G}(\cdot)$ determines the size of optimal uncorrupted set τ_e by optimizing the following problem:

$$\begin{aligned} \tau_* &= \arg\min_{\tau} \mathcal{L}(\tau) \\ s.t.\ \ & r_{\delta(\tau)} \leq \min\left(\frac{2\tau r_{\delta(\tau_o)}}{\tau_o}, \frac{r_{\delta(n)} + r_{\delta(\tau_o)}}{2}\right) \end{aligned} \tag{4}$$

where the function $\mathcal{L}$ is defined as

$$\mathcal{L}(\tau) := \frac{(r_{\delta(\tau)} + 1)/\tau}{(r_{\delta(n)} - r_{\delta(\tau)})/(n-\tau)} \tag{5}$$

The variable τ_o in the constraint is defined as follows:

$$\tau_o = \arg\min_{1\leq\tau\leq n} \left| r_{\delta(\tau)}^2 - \frac{\|r_{S_{\tau'}}\|_2^2}{\tau'} \right| \tag{6}$$

where $\tau' = \tau - \lceil n/2 \rceil$ and $S_{\tau'}$ is the position set containing the smallest τ' elements in residual $\boldsymbol{r}$. The constraint is imposed to avoid the case when τ is close to n, where the residual becomes so arbitrary that the denominator can become very large, making $\mathcal{L}$ much smaller than the value of the estimated threshold τ_e.

Applying the optimal uncorrupted set size generated by $\mathcal{G}(\cdot)$, the heuristic hard thresholding is defined as follows:

Definition 2. ***Heuristic Hard Thresholding***. *Denoting $\delta_{\boldsymbol{r}}^{-1}(i)$ as the position of the i^{th} element in residual vector $\boldsymbol{r}$'s ascending order of magnitude. The heuristic hard thresholding of $\boldsymbol{r}$ is defined as*

$$\mathcal{H}_{\mathcal{G}}(\boldsymbol{\beta}) = \{i \in [n] : \delta_{\boldsymbol{r}}^{-1}(i) \leq \mathcal{G}(\boldsymbol{\beta})\} \tag{7}$$

Algorithm 1: RLHH ALGORITHM

Input: Corrupted training data $\{\boldsymbol{x}_i, y_i\}$, i = 1...n, tolerance ϵ
Output: solution $\hat{\boldsymbol{\beta}}$

1 S_0 = [n], $t \leftarrow 0$
2 **repeat**
3 $\quad \boldsymbol{\beta}^{t+1} \leftarrow (X_{St}X_{St}^T)^{-1}X_{St}\boldsymbol{y}_{St}$
4 $\quad$ **for** *i = 1..n* **do**
5 $\quad\quad r_i^{t+1} \leftarrow |y_{S_t i} - \boldsymbol{x}_{S_t i}^T\boldsymbol{\beta}|$
6 $\quad S_{t+1} \leftarrow \mathcal{H}_{\tau_*}(\boldsymbol{r}^{t+1})$, where τ_* is solved by Equation (4).
7 $\quad t \leftarrow t+1$
8 **until** $\|\boldsymbol{r}_{S_{t+1}}^{t+1} - \boldsymbol{r}_{S_t}^{t}\|_2 < \epsilon n$
9 **return** $\boldsymbol{\beta}^{t+1}$

The optimization of S is formulated as solving the Equation (7). The set returned by $\mathcal{H}_{\mathcal{G}}(\boldsymbol{\beta})$ will be used as the estimated uncorrupted set.

We will first present the reasoning behind our choice of function in this section and then show that our heuristic function can indeed ensure a rigorous recovery of regression coefficients $\boldsymbol{\beta}$ in Section 4. Basically, the function follows a natural intuition that data points with unbounded corruption always have a higher residual $r_i = y_i - X_i\boldsymbol{\beta}$ in magnitude compared to uncorrupted data. Moreover, as this corruption is arbitrary and unbounded, when the residual vector $\boldsymbol{r}$ is sorted in ascending order, the slope of overall corrupted data is always much larger than the slope of the uncorrupted data. As Figure 1 shows, point p^* has the minimum $\mathcal{L}$ value in the feasible domain. Therefore, we can estimate the corresponding threshold τ_* of point p^* as the optimal threshold. To avoid a zero value for the numerator of $\mathcal{L}(\tau)$, we add 1 to all the values in the residual vector.

In Algorithm 1, an efficient robust regression algorithm, *RLHH*, based on heuristic hard thresholding is proposed to solve the *RLSR* problem. It follows an intuitive strategy of updating $\boldsymbol{\beta}$ to provide a better fit for the current estimated set S in Line 3, and updating the residual vector $\boldsymbol{r}$ in Line

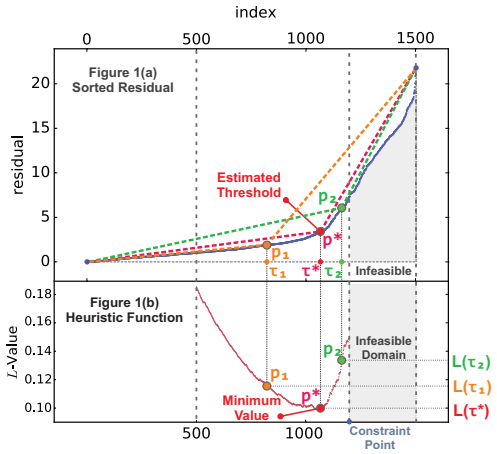


Figure 1: Blue line in figure (a) indicates the values of residual vector $\boldsymbol{r}$ in ascending order, and red point in figure (b) shows the corresponding value of heuristic function. τ_* is the estimated threshold with the minimum $\mathcal{L}$; τ_1 and τ_2 are the candidate threshold values in τ_*'s left and right hand side, respectively.

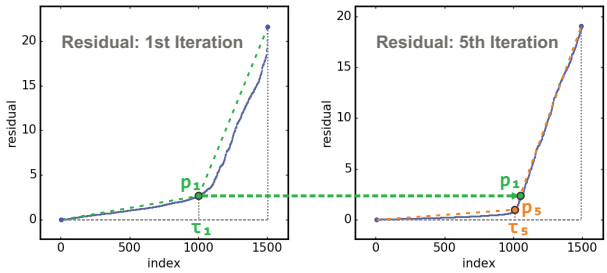


Figure 2: Residual $\boldsymbol{r}$ in ascending order for the 1st (left) and 5th (right) iterations.

4-5. It then estimates an active set S of uncorrupted points via heuristic hard thresholding in Line 6 based on the residual vector $\boldsymbol{r} = \boldsymbol{y} - X\boldsymbol{\beta}$ in the current iteration. The active set is initialized using the entire data samples in Line 1. The algorithm continues until the change in the residual vector falls within a small range. Figure 2 shows that the residual of the uncorrupted set in the 1st and 5th iteration, respectively. It intuitively explains the convergence progress of our algorithm: the optimization steps of $\boldsymbol{\beta}$ based on S_t makes the $\boldsymbol{r}_{S_t}$ smaller than its previous iteration, and it leads to smaller $\mathcal{L}$ values for items in S_t. Then these items in S_t have much higher possibility to kept in S_{t+1} than items in $[n] \setminus S_t$. This progress continues until the active set is fixed.

4 Theoretical Recovery Analysis

For convenience, the convergence analyses for the case without dense noise will be presented, i.e. $\boldsymbol{y} = X^T\boldsymbol{\beta} + \boldsymbol{u}$. The convergence proof relies on the optimality of two steps carried out by the algorithm, the $\boldsymbol{\beta}$ optimization step that selects the best coefficients based on the uncorrupted set, and the heuristic hard threshold step that automatically discovers the best active set based on the current regression coefficients.

Lemma 1. *For a given residual vector $\boldsymbol{r} \in \mathcal{R}^n$, let $\delta(k)$ be the k-th position of the ascending order in vector $\boldsymbol{r}$, i.e. $r_{\delta(1)} \le r_{\delta(2)} \le ... \le r_{\delta(n)}$. For any $1 \le \tau_1 < \tau_2 \le n$, let $S_1 = \{\delta(i)|1 \le i \le \tau_1\}$ and $S_2 = \{\delta(i)|1 \le i \le \tau_2\}$. We then have $\|\boldsymbol{r}_{S_1}\|_2^2 \le \frac{\tau_1}{\tau_2}\|\boldsymbol{r}_{S_2}\|_2^2 \le \|\boldsymbol{r}_{S_2}\|_2^2$.*

Proof. Let $S_3 = \{\delta(i) : \tau_1 + 1 \le i \le \tau_2\}$. Clearly , we have $\|\boldsymbol{r}_{S_2}\|_2^2 = \|\boldsymbol{r}_{S_1}\|_2^2 + \|\boldsymbol{r}_{S_3}\|_2^2$. Moreover, since each element in S_3 is larger than any of the element in S_1, we have $\|\boldsymbol{r}_{S_1}\|_2^2 \le \|\boldsymbol{r}_{S_2}\|_2^2 + \frac{|S_3|}{|S_1|}\|\boldsymbol{r}_{S_1}\|_2^2 \le \frac{|S_1|}{|S_1|+|S_3|}\|\boldsymbol{r}_{S_2}\|_2^2 = \frac{\tau_1}{\tau_2}\|\boldsymbol{r}_{S_2}\|_2^2 \le \|\boldsymbol{r}_{S_2}\|_2^2$. □

Lemma 2. *Let S_t be the estimated uncorrupted set at the t^{th} iteration. If $\tau_t \ge \tau_* = \gamma n$, then $|S_* \cap S_t| \ge \tau_t - \frac{n}{2}$.*

Proof. When S_t contains all the elements of $[n] \setminus S_*$, $|S_* \cap S_t|$ gets the smallest value $\tau_t - |[n] \setminus S_*|$. So we have

$$|S_* \cap S_t| \ge \tau_t - |[n] \setminus S_*| = \tau_t - (1-\gamma)n \quad (8)$$

Because $\gamma > \frac{1}{2}$, we have

$$|S_* \cap S_t| \ge \tau_t - n + \frac{n}{n} = \tau_t - \frac{n}{2} \quad (9)$$

□

Lemma 3. *Let τ_t be the estimated uncorrupted threshold at the t-th iteration. If $\tau_t > \tau_* = \gamma n$, then $\|\boldsymbol{r}^t_{S_t}\|_2^2 \leq \Big[1 + \frac{128(1-\gamma)}{2\gamma-1}\Big]\|\boldsymbol{r}^t_{S_*}\|_2^2$.*

Proof. To simplify the notation, we will omit all the subscripts t that signify the t-th iteration in the explanation below and assumes the residual vector $\boldsymbol{r}$ is sorted in ascending order of magnitude. According to the optimization step in equation (4), we have the following properties:

According to the optimization step in equation (4), we have the following properties:

$$\begin{aligned} r_\tau \leq & 2 \cdot \frac{\tau r_{\tau_o}}{\tau_o} \overset{(a)}{\leq} 8 \cdot r_{\tau_o} \\ r_\tau^2 \overset{(b)}{\leq} & \frac{64}{\tau'}\|\boldsymbol{r}_{S_* \cap S_t}\|_2^2 \\ |S_t \setminus S_*| r_\tau^2 \overset{(c)}{\leq} & (1-\gamma)\cdot n \cdot \frac{64}{\tau'}\|\boldsymbol{r}_{S_* \cap S_t}\|_2^2 \end{aligned} \tag{10}$$

The inequality *(a)* follows $\tau_o \geq \tau/4$ and inequality *(b)* follows the definition of τ_o in equation (6) and the fact that $|S_* \cap S_t| \geq \tau'$ in Lemma 2. The inequality *(c)* follows $|S_t \setminus S_*| \leq (1-\gamma)\cdot n$ and $\|\boldsymbol{r}_{S_t \setminus S_*}\|_2^2 \leq |S_t \setminus S_*| r_\tau^2$. Then we have

$$\begin{aligned} \|\boldsymbol{r}_{S_t \setminus S_*}\|_2^2 \leq & \Big[(1-\gamma)\cdot n \cdot \frac{64}{\tau'} + 1\Big]\|\boldsymbol{r}_{S_* \setminus S_t}\|_2^2 \\ & + \Big[(1-\gamma)\cdot n \cdot \frac{64}{\tau'}\Big]\|\boldsymbol{r}_{S_* \cap S_t}\|_2^2 \\ \|\boldsymbol{r}_{S_t \setminus S_*}\|_2^2 + \|\boldsymbol{r}_{S_* \cap S_t}\|_2^2 \overset{(d)}{\leq} & \Big[(1-\gamma)\cdot n \cdot \frac{64}{\tau'} + 1\Big]\|\boldsymbol{r}_{S_*}\|_2^2 \\ \|\boldsymbol{r}_{S_t}\|_2^2 \overset{(e)}{\leq} & \Big[1 + \frac{128(1-\gamma)}{2\gamma - 1}\Big]\|\boldsymbol{r}_{S_*}\|_2^2 \end{aligned} \tag{11}$$

The inequality *(d)* follows $\|\boldsymbol{r}_{S_*}\|_2^2 = \|\boldsymbol{r}_{S_* \setminus S_t}\|_2^2 + \|\boldsymbol{r}_{S_* \cap S_t}\|_2^2$. The inequality *(e)* follows $\tau' = \tau_t - \frac{n}{2}$. □

Theorem 4. *Let $X = [\boldsymbol{x}_1, ..., \boldsymbol{x}_n] \in \mathcal{R}^p$ be the given data matrix and $\boldsymbol{y} = X^T\boldsymbol{\beta}^* + \boldsymbol{u}$ be the corrupted output with $\|\boldsymbol{u}\|_0 = \gamma n$. Let Σ_0 be an invertible matrix such that $\tilde{X} = \Sigma_0^{-1/2}X$, $f(\boldsymbol{\beta}) = \|\boldsymbol{y}_S - \tilde{X}_S\boldsymbol{\beta}\|_2^2$ satisfies the SSC and SSS properties at level α, γ with $2\zeta_{\alpha,\gamma}$ and $2\kappa_{\alpha,\gamma}$ If the data satisfies $\frac{\kappa_\gamma}{\zeta_{1-\alpha}} < \frac{1}{\sqrt{\lambda}}(\sqrt{2}-1)$, then after $t = \mathcal{O}\left(\log_{\frac{1}{\eta}} \frac{\mu\|\boldsymbol{u}\|_2}{\epsilon}\right)$ iterations, Algorithm 1 yields an ϵ-accurate solution $\boldsymbol{\beta}^t$.*

Proof. Let $G_t = (X_{S_t}X_{S_t}^T)^{-1}X_{S_t}$, the t-th iteration of Algorithm 1 satisfies

$$\boldsymbol{\beta}^{t+1} = G_t\boldsymbol{y}_{S_t} = G_t(X_{S_t}^T\boldsymbol{\beta}^* + \boldsymbol{u}_{S_t}) = \boldsymbol{\beta}^* + G_t\boldsymbol{u}_{S_t}$$

Thus, the residual in t+1-th iteration for any set $S \subset [n]$, yields

$$\boldsymbol{r}_S^{t+1} = \boldsymbol{y}_S - X_S^T\boldsymbol{\beta}^{t+1} = \boldsymbol{u}_S - X_S^TG_t\boldsymbol{u}_{S_t}$$

For each iteration, we have two conditions when choosing different values of τ^{t+1}. For *condition 1*, $\tau^{t+1} \leq \tau_*$, we have $\|\boldsymbol{r}^{t+1}_{S_{t+1}}\|_2^2 \leq \|\boldsymbol{r}^{t+1}_{S_*}\|_2^2$ (see Lemma 1).

$$\begin{aligned} \|\boldsymbol{u}_{S_{t+1}}\|_2^2 &= \|\boldsymbol{u}_{S_{t+1}} - X_{S_{t+1}}^TG_t\boldsymbol{u}_{S_t}\|_2^2 - \|X_{S_{t+1}}^TG_t\boldsymbol{u}_{S_t}\|_2^2 \\ &\quad + 2\boldsymbol{u}_{S_{t+1}}^TX_{S_{t+1}}^TG_t\boldsymbol{u}_{S_t} \\ &\overset{(a)}{\leq} \|X_{S_*}^TG_t\boldsymbol{u}_{S_t}\|_2^2 - \|X_{S_{t+1}}^TG_t\boldsymbol{u}_{S_t}\|_2^2 + 2\boldsymbol{u}_{S_{t+1}}^TX_{S_{t+1}}^TG_t\boldsymbol{u}_{S_t} \\ &\overset{(b)}{\leq} \frac{\kappa_{\alpha_1}^2}{\zeta_{1-\gamma}^2}\|\boldsymbol{u}_{S_t}\|_2^2 + 2\frac{\kappa_{\alpha_1}}{\zeta_{1-\gamma}}\|\boldsymbol{u}_{S_t}\|_2\|\boldsymbol{u}_{S_{t+1}}\|_2 \end{aligned} \tag{12}$$

where $\alpha_1 = \max_t\{1 - \frac{\tau_t}{n}\}$. The inequality *(a)* follows $\|\boldsymbol{r}^{t+1}_{S_{t+1}}\|_2^2 \leq \|\boldsymbol{r}^{t+1}_{S_*}\|_2^2$, and inequality *(b)* follows from the setting $\tilde{X} = \Sigma_0^{-1/2}X$, *SSC*/*SSS* properties, $|S_t| \leq (1-\gamma)\cdot n$ and $|S_* \setminus S_{t+1}| \leq \alpha_1 \cdot n$. Solving the quadratic equation for the corruption vector gives us

$$\|\boldsymbol{u}_{S_{t+1}}\|_2 \leq (1+\sqrt{2})\frac{\kappa_{\alpha_1}}{\zeta_{1-\gamma}}\|\boldsymbol{u}_{S_t}\|_2 \tag{13}$$

For *condition 2*, $\tau^{t+1} > \tau_*$. According to Lemma 3, $\|\boldsymbol{r}^t_{S_t}\|_2^2 \leq \lambda\|\boldsymbol{r}^t_{S_*}\|_2^2$ where $\lambda = 1 + \frac{128(1-\gamma)}{2\gamma-1}$, we have

$$\begin{aligned} \|\boldsymbol{u}_{S_{t+1}}\|_2^2 &= \|\boldsymbol{u}_{S_{t+1}} - X_{S_{t+1}}^TG_t\boldsymbol{u}_{S_t}\|_2^2 - \|X_{S_{t+1}}^TG_t\boldsymbol{u}_{S_t}\|_2^2 \\ &\quad + 2\boldsymbol{u}_{S_t}^TX_{S_{t+1}}^TG_t\boldsymbol{u}_{S_t} \\ &\overset{(c)}{\leq} \lambda\|X_{S_*}^TG_t\boldsymbol{u}_{S_t}\|_2^2 - \|X_{S_{t+1}}^TG_t\boldsymbol{u}_{S_t}\|_2^2 + 2\boldsymbol{u}_{S_t}^TX_{S_{t+1}}^TG_t\boldsymbol{u}_{S_t} \\ &\overset{(d)}{\leq} \lambda\frac{\kappa_\gamma^2}{\zeta_{1-\alpha_2}^2}\|\boldsymbol{u}_{S_t}\|_2^2 + 2\frac{\kappa_\gamma}{\zeta_{1-\alpha_2}}\|\boldsymbol{u}_{S_t}\|_2\|\boldsymbol{u}_{S_{t+1}}\|_2 \\ &\overset{(e)}{\leq} \lambda\frac{\kappa_\gamma^2}{\zeta_{1-\alpha_2}^2}\|\boldsymbol{u}_{S_t}\|_2^2 + 2\sqrt{\lambda}\frac{\kappa_\gamma}{\zeta_{1-\alpha_2}}\|\boldsymbol{u}_{S_t}\|_2\|\boldsymbol{u}_{S_{t+1}}\|_2 \end{aligned} \tag{14}$$

where $\alpha_2 = \max_t\{1 - \frac{\tau_t}{n}\}$. The inequality *(c)* follows Lemma 3, inequality *(d)* follows from the definition of *SSC*/*SSS* properties, $|S_t| \leq (1-\alpha_2)\cdot n$ and $|S_* \setminus S_{t+1}| \leq \gamma\cdot n$. Inequality *(e)* notices the fact that $\sqrt{\lambda} \geq 1$. Solving the quadratic equation in Equation (14) gives us

$$\|\boldsymbol{u}_{S_{t+1}}\|_2 \leq (1+\sqrt{2})\sqrt{\lambda}\frac{\kappa_\gamma}{\zeta_{1-\alpha_2}}\|\boldsymbol{u}\|_2 \tag{15}$$

Combine these two conditions and let t_1 be the iterations for the case of condition 1. We get

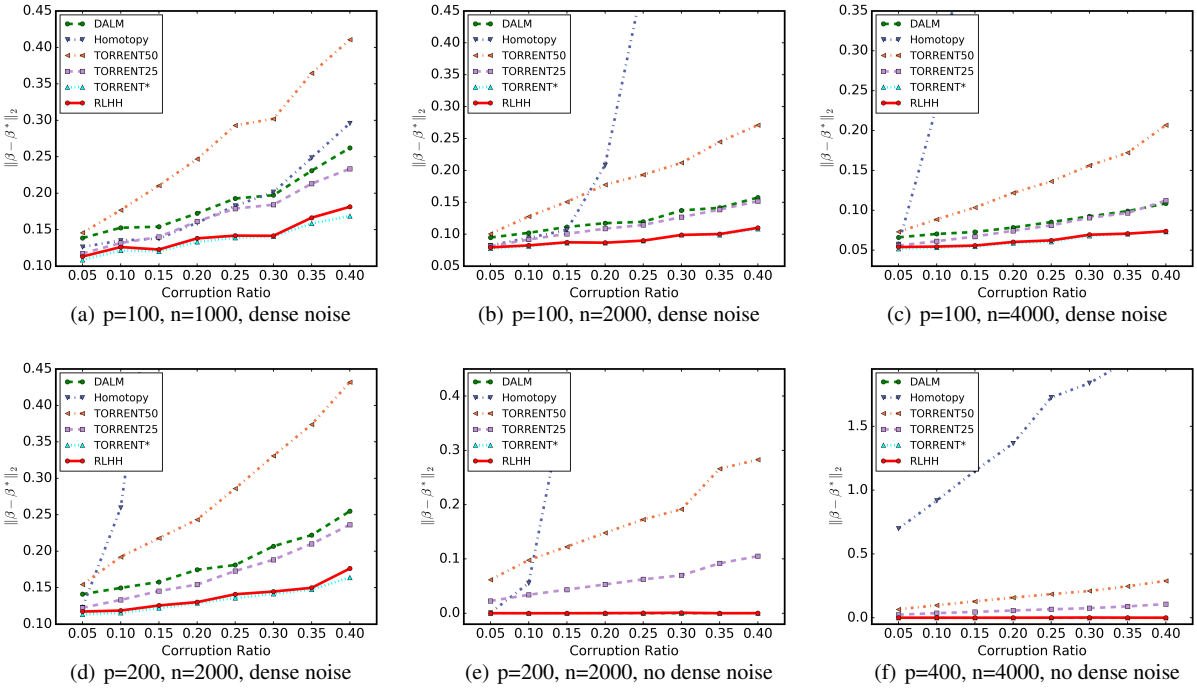
$$\begin{aligned} \|\boldsymbol{\beta}^{t+1} - \boldsymbol{\beta}^*\|_2 &= \|G_t\boldsymbol{u}_{S_t}\|_2 \leq \mu\|\boldsymbol{u}_{S_t}\|_2 \\ &\leq \mu \cdot \eta_1^{t_1} \cdot \eta^{t+1-t_1}\|\boldsymbol{u}\|_2 \leq \mu \cdot \eta^{t+1}\|\boldsymbol{u}\|_2 \end{aligned}$$

where $\mu = \max\left\{\frac{\sqrt{\kappa_{\alpha_1}}}{\zeta_{1-\gamma}}, \frac{\sqrt{\kappa_\gamma}}{\zeta_{1-\alpha_2}}\right\}$, $\eta_1 = \frac{(1+\sqrt{2})\kappa_{\alpha_1}}{\zeta_{1-\gamma}}$, $\eta = \frac{(1+\sqrt{2})\sqrt{\lambda}\kappa_\gamma}{\zeta_{1-\alpha_2}}$. When $\frac{\kappa_\gamma}{\zeta_{1-\alpha_2}} < \frac{\sqrt{2}-1}{\sqrt{\lambda}}$, we have $\eta < 1$ and after $t = \mathcal{O}\left(\log_{\frac{1}{\eta}} \frac{\mu\|\boldsymbol{u}\|_2}{\epsilon}\right)$, $\|\boldsymbol{\beta}^{t+1} - \boldsymbol{\beta}^*\|_2 \leq \epsilon$. □

5 Experimental Results

In this section, we report the extensive experimental evaluation carried out to verify the robustness and efficiency of the proposed method. All the experiments were conducted on a 64-bit machine with Intel(R) core(TM) quad-core processor (i7CPU@3.6GHz) and 32.0GB memory. Details of both the source code and sample data used in the experiment can be downloaded here[1].

[1]https://github.com/xuczhang/RLHH



(a) p=100, n=1000, dense noise (b) p=100, n=2000, dense noise (c) p=100, n=4000, dense noise

(d) p=200, n=2000, dense noise (e) p=200, n=2000, no dense noise (f) p=400, n=4000, no dense noise

Figure 3: Performance on regression coefficients recovery.

5.1 Datasets and Metrics

To demonstrate the performance of our proposed method, we carried out comprehensive experiments in synthetic datasets. Specifically, the simulation samples were randomly generated according to the model in Equation (1) for *RLSR* problem, sampling the regression coefficients $\boldsymbol{\beta}^* \in \mathcal{R}^p$ as a random unit norm vector. The covariance data X was drawn independently and identically distributed from $\boldsymbol{x}_i \sim \mathcal{N}(0, I_p)$ and the uncorrupted response variables were generated as $y_i^* = \boldsymbol{x}_i^T \boldsymbol{\beta}^*$. The set of corrupted points S was selected as a uniformly random (n-τ_*)-sized subset of $[n]$, where τ_* is the size of the uncorrupted set. The corrupted response vector was generated as $\boldsymbol{y} = \boldsymbol{y}^* + \boldsymbol{u} + \boldsymbol{\varepsilon}$, where the corruption vector $\boldsymbol{u}$ was sampled from the uniform distribution $[-5\|\boldsymbol{y}^*\|_\infty, 5\|\boldsymbol{y}^*\|_\infty]$ and the additive dense noise was $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$.

Following the setting in [Bhatia *et al.*, 2015], we measured the performance of the regression coefficients recovery using the standard L_2 error $e = \|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2$, where $\hat{\boldsymbol{\beta}}$ represents the recovered coefficients for each method and $\boldsymbol{\beta}^*$ is the true regression coefficients. To validate the performance for corrupted set discovery, the F1-score is measured by comparing the discovered corrupted sets with the actual ones. To compare the scalability of each method, the CPU running time for each of the competing methods was also measured.

5.2 Comparison Methods

The following methods are included in the performance comparison presented here: *Ordinary least squares* (*OLS*). The *OLS* method ignores the corruption of data and trains the model based on the whole dataset. We also compared our method to the regularized L_1 algorithm for robust regression [Wright and Ma, 2010] [Nguyen and Tran, 2013]. For extensive L_1 minimization solvers, [Yang *et al.*, 2010] showed that the *Homotopy* and *DALM* solvers outperform other proposed methods both in terms of recovery properties and running time. Both of the L_1 solver methods are parameter free. Another recently proposed hard thresholding method, *Torrent (Abbr. Torr)*, developed for robust regression [Bhatia *et al.*, 2015] was also compared to our method. As the method requires a parameter for corruption ratio, which is difficult to estimate in practice, we chose 4 versions with different parameter settings: *TORR**, *TORR25*, *TORR50*, and *TORR80*. *TORR** uses the true corruption ratio as its parameter, and the others apply parameters that are uniformly distributed across the range of $\pm 25\%$, $\pm 50\%$, and $\pm 80\%$ off the true value, respectively. All the results will be averaged over 10 runs.

5.3 Recovery of regression coefficients

We selected 6 competing methods with which to evaluate the recovery performance of $\boldsymbol{\beta}$: *OLS*, *DALM*, *Homotopy*, *TORR**, *TORR25*, *TORR50*. As the recovery error for the *OLS* method is almost 10 times larger than those of the other methods, its result is not shown in Figure 3 in order to present the other results properly. Figures 3(a), 3(b), and 3(c) show the recovery performance for different data sizes when the feature number is fixed. Looking at the results, we can conclude that: 1) the *RLHH* method outperforms all the competing methods except for *TORR**, whose parameter is hardly given in practice. 2) The results of the *TORR* methods are significantly affected by their corruption ratio parameters; *TORR50* performs almost twice as badly as *TORR** and yields worse results than one of the L_1-Solver methods, *DALM*. However, *RLHH* performs consistently throughout, with no impact of the parame-

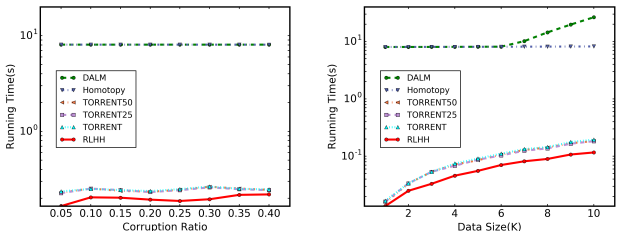
Table 1: F1 scores for the performance on uncorrupted set recovery.

	p=100, n=1000				p=100, n=2000				p=100, n=4000			
	10%	20%	30%	40%	10%	20%	30%	40%	10%	20%	30%	40%
TORR80	0.949	0.881	0.779	0.612	0.950	0.883	0.783	0.622	0.951	0.883	0.785	0.626
TORR50	0.967	0.925	0.865	0.781	0.968	0.926	0.868	0.785	0.968	0.926	0.871	0.787
TORR25	0.981	0.958	0.927	0.887	0.982	0.960	0.929	0.891	0.982	0.960	0.931	0.892
RLHH	**0.989**	**0.979**	**0.973**	**0.956**	**0.991**	**0.987**	**0.977**	**0.964**	**0.992**	**0.987**	**0.978**	**0.971**
TORR*	**0.993**	**0.987**	**0.979**	**0.971**	**0.995**	**0.990**	**0.980**	**0.972**	**0.995**	**0.989**	**0.982**	**0.975**
	p=200, n=2000				**p=200, n=4000**				**p=100, n=4000 (no dense noise)**			
	10%	20%	30%	40%	10%	20%	30%	40%	10%	20%	30%	40%
TORR80	0.950	0.882	0.783	0.613	0.950	0.883	0.784	0.622	0.953	0.889	0.793	0.636
TORR50	0.968	0.926	0.869	0.780	0.968	0.926	0.870	0.785	0.971	0.933	0.880	0.800
TORR25	0.982	0.959	0.930	0.888	0.982	0.959	0.931	0.891	0.986	0.968	0.943	0.909
RLHH	**0.990**	**0.983**	**0.974**	**0.954**	**0.990**	**0.985**	**0.978**	**0.965**	**0.994**	**0.995**	**0.994**	**0.994**
TORR*	**0.994**	**0.988**	**0.982**	**0.972**	**0.995**	**0.989**	**0.983**	**0.973**	**1.000**	**1.000**	**1.000**	**1.000**

ter. 3) The L_1-Solver methods generally exhibit worse performance than the hard thresholding based algorithms. Specifically, compared to *DALM*, *Homotopy* is more sensitive to the number of corrupted instances in the data. Figure 3(d) shows its similar performance when the feature number increases. Figures 3(e) and 3(f) show that *RLHH* performs equally as well as *TORR** without dense noise, with both achieving an exact recovery of regression coefficients β.

5.4 Recovery of Uncorrupted Sets

As most competing methods do not explicitly estimate uncorrupted sets, we compared our proposed method with the *TORR* algorithm using a number of different parameter settings ranging from the true corrupted ratio up to a deviation of 30%. As the results shown in Table 1, we found that: 1) The F1 score of *RLHH* is 1.1% less than that of *TORR** on average, although it is important to note that the latter uses the true corruption ratio, which cannot be estimated exactly in practice. This indicates that our method provides very close to an optimal estimation result for an uncorrupted set. 2) The *RLHH* method significantly outperforms the other methods, especially when the true corruption ratio is high. 3) The results of the *TORR* methods are highly dependent on the corruption ratio parameter: the results for a 25% corruption estimation error are much better than those for a 50% error. However, *RLHH* is a parameter free method that is capable of obtaining a good result consistently. 4) When increasing the feature number and corruption ratio, the F1 scores slightly increase for all the methods. 5) In a no dense noise setting, *RLHH* performs a near optimal recovery result, while *TORR** exactly recovers the result only because it is using the true corruption ratio.



(a) p=400, n=4000, no dense noise (b) p=100, cr=0.1, dense noise

Figure 4: Running time for different corruption ratios and data sizes

5.5 Efficiency

To evaluate the efficiency of our proposed method, we compared the performances of all the competing methods for two different data settings: different corruption ratios and data sizes. In general, as Figure 4 shows, the hard thresholding based methods significantly outperformed the L_1-Solver based methods. Also, the running time for *RLHH* increases slowly as either the corruption ratio or the data size increases, just as in the *TORR* methods. In addition, even though *RLHH* performs the additional step of estimating the uncorrupted set in each optimization iteration, the efficiency of *RLHH* still outperforms *TORR*, which indicates that 1) the heuristic hard thresholding step in *RLHH* always performs efficiently, even under in different circumstances; and 2) The *RLHH* algorithm converges more quickly than *TORR*.

6 Conclusion

In this paper, a novel robust regression algorithm, *RLHH*, is proposed to recover the regression coefficients and the uncorrupted set in the presence of adversarial corruption in the response vector. To achieve this, we designed a heuristic hard thresholding method with which to estimate the optimal uncorrupted set that is alternately updated with the optimized regression coefficients. We demonstrate that our algorithm can recover true regression coefficients exactly, with a geometric convergence rate. Extensive experiments on massive simulation data demonstrated that the proposed algorithm outperforms other comparable methods in both effectiveness and efficiency.

References

[Bhatia *et al.*, 2015] Kush Bhatia, Prateek Jain, and Purushottam Kar. Robust regression via hard thresholding. In *Advances in Neural Information Processing Systems*, pages 721–729, 2015.

[Chen *et al.*, 2013] Yudong Chen, Constantine Caramanis, and Shie Mannor. Robust sparse regression under adversarial corruption. In Sanjoy Dasgupta and David Mcallester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, pages 774–782. JMLR Workshop and Conference Proceedings, May 2013.

[Huber and Ronchetti, 2009] Peter J. Huber and Elvezio M. Ronchetti. *The Basic Types of Estimates*, pages 45–70. John Wiley & Sons, Inc., 2009.

[Jung *et al.*, 2016] Yoonsuh Jung, Seung Pil Lee, and Jianhua Hu. Robust regression for highly corrupted response by shifting outliers. *Statistical Modelling*, 16(1):1–23, 2016.

[Lourenco *et al.*, 2011] VM Lourenco, Ana M Pires, and M Kirst. Robust linear regression methods in association studies. *Bioinformatics*, 27(6):815–821, 2011.

[Maronna *et al.*, 2006] RARD Maronna, R Douglas Martin, and Victor Yohai. *Robust statistics*. John Wiley & Sons, Chichester. ISBN, 2006.

[Mcwilliams *et al.*, 2014] Brian Mcwilliams, Gabriel Krummenacher, Mario Lucic, and Joachim M. Buhmann. Fast and robust least squares estimation in corrupted linear models. In Z. Ghahramani, M. Welling, C. Cortes, N.d. Lawrence, and K.q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 415–423. Curran Associates, Inc., 2014.

[Naseem *et al.*, 2012] Imran Naseem, Roberto Togneri, and Mohammed Bennamoun. Robust regression for face recognition. *Pattern Recognition*, 45(1):104–118, 2012.

[Nguyen and Tran, 2013] Nam H Nguyen and Trac D Tran. Exact recoverability from dense corrupted observations via l1-minimization. *IEEE transactions on information theory*, 59(4):2017–2035, 2013.

[Rousseeuw and Leroy, 2005] Peter J Rousseeuw and Annick M Leroy. *Robust regression and outlier detection*, volume 589. John wiley & sons, 2005.

[She and Owen, 2011] Yiyuan She and Art B. Owen. Outlier detection using nonconvex penalized regression. *Journal of the American Statistical Association*, 106(494):626–639, 2011.

[Smolic and Ohm, 2000] Aljoscha Smolic and Jens-Rainer Ohm. Robust global motion estimation using a simplified m-estimator approach. In *Image Processing, 2000. Proceedings. 2000 International Conference on*, volume 1, pages 868–871. IEEE, 2000.

[Studer *et al.*, 2012] Christoph Studer, Patrick Kuppinger, Graeme Pope, and Helmut Bolcskei. Recovery of sparsely corrupted signals. *IEEE Transactions on Information Theory*, 58(5):3115–3130, 2012.

[Wright and Ma, 2010] John Wright and Yi Ma. Dense error correction via l1-minimization. *IEEE Trans. Inf. Theor.*, 56(7):3540–3560, July 2010.

[Wright *et al.*, 2009] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):210–227, 2009.

[Yang *et al.*, 2010] Allen Yang, Arvind Ganesh, Shankar Sastry, and Yi Ma. Fast l1-minimization algorithms and an application in robust face recognition: A review. Technical Report UCB/EECS-2010-13, EECS Department, University of California, Berkeley, Feb 2010.

[Zoubir *et al.*, 2012] Abdelhak M Zoubir, Visa Koivunen, Yacine Chakhchoukh, and Michael Muma. Robust estimation in signal processing: A tutorial-style treatment of fundamental concepts. *IEEE Signal Processing Magazine*, 29(4):61–80, 2012.