Huber Adaptive Regression

Virgile Rennard, Nabil Madali

ABSTRACT

Nowadays, heavy tailed data errors and outliers are more and more commonly encountered in applications which appears in the response variables or the predictors. Here, we will be considering the regression problem for heavy tailed data, where outliers are common. In this framework, the usual Ordinary Least Square estimator is known as an inefficient estimator. To overcome this problem, is needed estimators that are more robust to heavy tailed data, and therefore work without the sub-gaussian tail assumption. The Huber estimator or the least absolute deviation lends themselves well for this framework. The different new hyperparameters, while valuable, makes experimentation longer and more complicated as it requires cross-validation , which is time consuming. In (13) is introduced the Adaptive Huber Regression for simultaneous robust estimation and inference, which has as its main observation that the robustification parameter has to adapts to the sample size, dimension and moments in order to have an optimal bias-robustness tradeoff. The theoretical framework deals with data with heavy-tailed distributions and bounded $(1+\delta)$-th moment for any $\delta > 0$. In it is established a sharp transition phase for robust estimation of regression parameters in high and low dimensions. When $\delta \geq 1$, the estimator admits a sub-Gaussian-type deviation bound without sub-Gaussian assumptions on the data, while only a slower rate is available in the regime $0 < \delta < 1$.

Key words. Adaptive Huber regression – Bias and robustness tradeoff – Finite-sample inference – Heavy-tailed data, – nonasymptotic optimality, – phase transition

1. Introduction

In statistics as well as optimization, many problems requires robustness, that is, that a model should be less influenced by outliers than by inliers (9) (10) . This is a common concept in both learning and parameter estimation tasks; In those cases, a robust loss such as the absolute error loss $l_1(x, y) = |g(x) - y|$ would be preferred over a non robust loss, such as the squared error $l_2(x, y) = (g(x) - y)^2$ due to its reduced sensitivity to large errors. Lots of research has been done to develop different robust penalties with specific properties, which are summarized in (3) (16) . In (8) is shown that these losses are often interchangeable, so that researchers may experiment with different losses when designing a system. This flexibility can be useful when our noise is not Gaussian, or when the loss that is minimized during learning and estimation is different from how the learned model or estimated parameters will be evaluated

In (2) is proposed a single loss function that is a superset of many common robust loss functions. With only a single continuous parameter in the loss function, the general loss is controlled to merge several different traditional losses, and can thus be adjusted to model a wider family of functions. This allows us to generalize algorithms built around a fixed robust loss with a new "robustness" hyperparameter that can be tuned to improve performance.
The simplest form of the loss function is:

$$f(x, \alpha, c) = \frac{|\,\alpha - 2\,|}{\alpha}((\frac{(x/c)^2}{|\,\alpha - 2\,|} + 1)^{\alpha/2} - 1) \tag{1}$$

Where $\alpha \in \mathbb{R}$ is a shape parameter that controls the robustness of the loss and c >0 is a scale parameter that controls the size of the loss's quadratic part near x= 0.

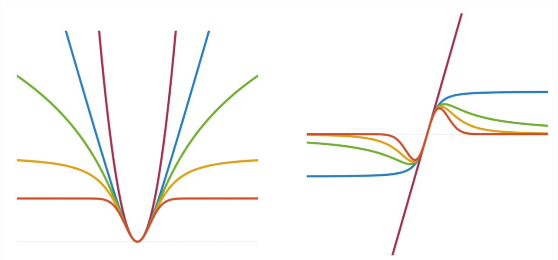


Fig. 1. Adaptive loss function (2) for different values of its shape parameter α (left) and its gradient (right) . Several values of α reproduce existing loss functions: L2 loss (α= 2), Charbonnierloss (α= 1), Cauchy loss (α= 0), Geman-McClure loss ($\alpha = -2$), and Welsch loss ($\alpha = -\infty$).

It is important to note that the loss is not defined when $\alpha = 2$, as it approaches L2 loss (squared error) in the limit:

$$\lim_{\alpha \to 2} f(x, \alpha, c) = \frac{1}{2}(x/c)^2 \tag{2}$$

When $\alpha = 1$ the loss is a smoothed form of L1 loss:

$$f(x, 1, c) = \sqrt{(x/c)^2 + 1} - 1 \tag{3}$$

This is often referred to as Charbonnier loss (4) , pseudo-Huber loss (as it resembles Huber loss (8)), or L1-L2 loss (16) .

we can take the limit off f(x,α,c) as α approaches zero:

$$\lim_{\alpha \to 0} f(x, \alpha, c) = log(\frac{1}{2}(x/c)^2 + 1) \tag{4}$$

Yielding the Cauchy loss [2]. By setting $\alpha = -2$, the loss is simply the Geman-McClure loss presented in (7) .

With this, we can explicit the final general loss function, which is simply $f(\cdot)$ with shpecial cases for its singularities when $\alpha = 0$ $\alpha = 2$ as well as its limit when $\alpha \rightarrow -\infty$.

$$\rho(x, \alpha, c) = \begin{cases} \frac{1}{2}(x/c)^2 & \alpha = 2 \\ log(\frac{1}{2}(x/c)^2 + 1) & \alpha = 0 \\ 1 - exp(-(x/c)^2) & \alpha = -\infty \\ \frac{|\alpha-2|}{\alpha}((\frac{(x/c)^2}{|\alpha-2|} + 1)^{\alpha/2} - 1) & otherwise \end{cases} \tag{5}$$

In this paper based on (13), we will begin by revisiting the robust regression that was first initiated by Peter Huber in his work Huber (1973). The asymptotic properties of the Huber estimator have been well studied in the literature.We will refer to Huber (1973), Yohai and Maronna (1979), Portnoy (1985), Mammen (1989) and He and Shao (1996, 2000) for a starting overview. The main difference is that in the previous papers, the robustification parameter is fixed according to the 95 % asymptotic efficiency rule. Thus, this procedure can not estimate the model-generating parameters consistently when the sample distribution is asymmetric.

From a non asymptotic perspective (rather than an asymptotic efficiency rule), (13) proposed to use the Huber regression with an adaptive robustification parameter, which is referred to as the adaptive Huber regression, for robust estimation and inference. The adaptive procedure achieves the non asymptotic robustness in the sense that the resulting estimator admits exponential-type concentration bounds when only low-order moments exist. Moreover, the resulting estimator is also an asymptotically unbiased estimate for the parameters of interest.

We will be giving an overview of (6), presenting the generalities needed to understand the context, and explaining the main results that are presented in the paper, as well as trying to get into perspective the importance of the results. We will also apply the Adaptive Huber Regression on our own dataset in order to compare how well it works compared to other Regression methods on heavy-tailed data. We will then try to explain how the regression algorithm works in the regularized case and try to relate the results of this paper to other works.

2. Generalities

In (6) is developed the Adaptive Huber Regression theory and in it is given different concentration bounds for the resulting estimator in the case of heavy-tailed data and in the non-asymptotic context. We will now start by presenting generalities about (6) before presenting the main results.

Definition 1 *(Huber Loss, Robustification Parameter) The Huber loss $l_\tau(.)$ is defined as :*

$$l_\tau(x) = \begin{cases} x^2/2 & |x| \leq \tau \\ \tau|x| - \tau^2/2 & |x| > \tau \end{cases}$$

where τ is the robustification parameter that balances bias and robustness

We can see that the Huber loss merges both the L_1 loss, which is robust for large values and the Quadratic loss which is smoother. This merge is controlled by the robustification parameter τ. It is quadratic for small values of x, and linear if $x > \tau$, thus down weighting the outliers and granting more robustness to large errors. We can remark that if $\tau = \infty$,we simply have the L_2 loss and an L_1 loss if $\tau = 0$. It is therefore necessary to tune the robustification parameter well in order to minimize the loss of robustification.

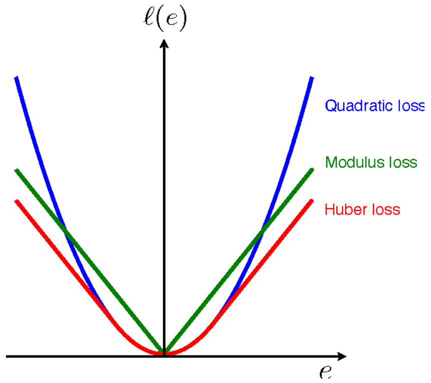


Fig. 2. The Huber loss compared to the L_1 and L_2 losses for a robustification parameter $\tau = 0$

In linear regression (1) one often assumes that the error term in the linear relationship between the dependent variable Y and some feature vector X is normally distributed with mean zero and constant variance σ^2,i.e $Y \mid X \sim X^T\beta + \epsilon$ with $\epsilon \in \mathcal{N}(0, \sigma^2)$ and β being a set of variational parameters. One is interested in finding the best estimate $\hat{\beta}$ that minimizes a quadratic cost function (corresponding to the log-likelihood of the distribution of ϵ). The estimate $\hat{y}$ for a given X is then simply $\hat{y}(X) = E[Y \mid X \sim X^T\hat{\beta}]$.Huber regression replaces the normal distribution with a more heavy tailed distribution but still assumes a constant variance.

Let us consider a simple one dimensional problem based on (1) with a skew normal error distribution of non-constant variance, i.e. y = x + $\epsilon_{skewnormal}$. We have used the SciPy skewnorm object with skewness parameter a=100 , loc = 0 and scale = 1 +| x $|^2$ and created a linearly spaced grid of N = 10^6 examples from the interval $[-500, 500]$.Further, we trained three LightGBM models. A root mean squared error (RMSE) model on g(y), a mean absolute error (MAE) model on g(y) and a GHL model, all with link function g(x) = sgn(x) log(1 +| x |) The models were trained with early stopping based on the R^2- score metric by using a 3-fold cross validation approach. The free parameter in the GHL function was also cross validated and chosen to be $\alpha = 0.67$. All sampling parameters in LightGBM were set to 1 and $num_{leaves} = 31$ while leaving all other parameters as default.

In general one needs a good starting vector in order to converge to the minimum of the GHL loss function. Here we have first trained a small LightGBM model of only 20 trees on g(y) with the classical Huber objective function (Huber parameter $\alpha = 2$). The output of this model was then used as the starting vector ($init_{score}$) of the GHL model.

Figure below shows a (binned) scatter plot of test set (0.7/0.3 split ratio) predictions of all three models with the prediction closest to the trend line coming from the GHL model.The GHL model achieves significantly better results on the test set while at the same time needing less trees than the other two models.

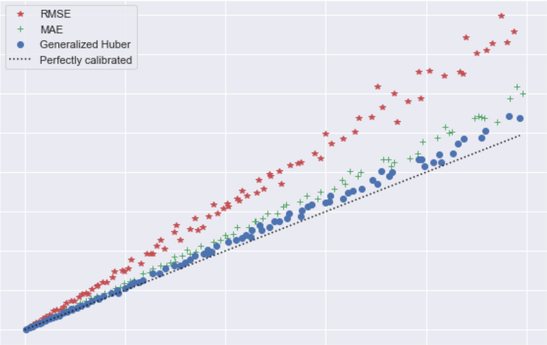


Fig. 3. Comparison of RMSE, MAE and Generalized Huber Source (1).

The problem to solve in low and high dimension is not the same, as we need to add parameters in the case of high dimensions in order to get an optimal estimator. Therefore, the resulting empirical huber estimator will not be obtained in the same way if the dimension of the data is considerably large. In low dimension, we define the loss as

$$\mathcal{L}_\tau(\beta) = \frac{1}{n}\sum_{i=1}^{n} l_\tau(y_i - < x_i, \beta >) \qquad \text{for } \beta \in \mathbb{R}^d \tag{6}$$

We then have :

Definition 2 *Huber Estimator* $\widehat{\beta_\tau} = argmin_{\beta\in\mathbb{R}^d}\mathcal{L}_\tau(\beta)$

Whereas in High Dimensions, for the same loss function, we solve the regularized adaptive Huber regression, which is the same problem with a Lasso type regularization :

Definition 3 $\widehat{\beta_{\tau,\lambda}} = argmin_{\beta\in\mathbb{R}^d}\{\mathcal{L}_\tau + \lambda||\beta||_1\}$

It is important to note that the Lasso type regularization is well suited to high dimensional data problem, as it leads to the sparsification of the parameters for a larger regularization parameter λ.

One of the main idea introduced in (6) is the way to choose the robustification parameter in the Huber loss τ, which will now, instead of being fixed for a certain threshold, adapt to different aspects of the data : the sample size, dimension and moments. The resulting parameter will lead to having an optimal bias-robustification tradeoff.

In low dimension, we need to restrict ourselves to the case where the empirical means of the $1 + \delta$-th moment is finite. In other words, $v_\delta = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[|\epsilon_i|^{1+\delta}] < \infty$. Under this condition, we have that our resulting estimator $\widehat{\beta_\tau}$, where our robustification parameter $\tau \asymp min(\mathrm{v}_\delta^{\frac{1}{1+\delta}}, v_1^{\frac{1}{2}})\mathrm{n}^{max(\frac{1}{1+\delta},\frac{1}{2})}$ achieves a tight bound to β^* of the order of $d^{1/2}\tau^{-(max(\delta,1))} \asymp d^{1/2}n^{-min(\frac{1}{1+\delta},\frac{1}{2}))}$, where the transition is when $\delta = 1$. Having the second moment ($\delta \geq 1$) bounded leads to a sub-Gaussian type deviation inequality without the sub-Gaussian data requirement.

As was explained, in the case of high dimension, we solve the regularized version of the problem, with two hyperparameters τ, λ. In this framework, $\tau \asymp v_\delta(\frac{n}{ln(d)})^{max(\frac{1}{1+\delta},\frac{1}{2})}$ and $\lambda \asymp v_\delta(\frac{ln(d)}{n})^{min(\frac{\delta}{1+\delta},1/2}$ where $v_\delta = min(\frac{1}{1+\delta},\frac{1}{2})$. Moreover, we introduce s the size of the support of β^*, the bound will depend on the size of the support and be on the order of $s^{1/2}\frac{\lambda}{v_\delta}$ with high probability. We point out that the $s^{1/2}ln(d)^{min(\delta/(1+\delta),1/2)}$ shows that the sparser the model is, the faster it will converge, once again either at a slow pace for $\delta < 1$, and getting a gaussian type bound if $\delta \geq 1$

This bounds need to be unified, as they do not depend on the same problem. (6) introduces two objects in order to do so, the effective dimensions, which is d in low dimensions and s in high dimensions, which is the number of nonzero parameters of the problem. The other object is the effective sample size which is either n in low dimension and $n/ln(d)$ in high dimensions. The generalized upper bound results in

$$||\widehat{\beta} - \beta^*||_2 \leq d_{eff}^{1/2} n_{eff}^{-min(\delta/(1+\delta),1/2}$$

3. The Adaptive Huber Regression in low dimension

In order to make the bounds work, the following restriction is required :

Condition 1 *Let* $S_n = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^t$ *the Empirical Gram Matrix be non singular, and have bounded eigenvalues.*

Then, $\forall\tau > 0, \widehat{\beta_\tau}$ is an estimator of $\beta_\tau^* = argmin_{\beta\in\mathbb{R}^d}\mathbb{E}[\mathcal{L}_\tau(\beta)] = argmin_{\beta\in\mathbb{R}^d}\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[l_\tau(y_i - < x_i, \beta)]$. This is needed in order to explicit the bias - robustness tradeoff. Here, we have β_τ^* as the Huber Regression coefficient, and the estimation bias being measured by $||\beta_\tau^* - \beta^*||_2$. As explained earlier, when $\tau \longrightarrow +\infty$ the bias tends to 0. The following equation holds :

$$||\widehat{\beta_\tau} - \beta^*||_2 \leq ||\widehat{\beta_\tau} - \beta_\tau^*||_2 + ||\beta_\tau^* - \beta^*||_2$$

In order to have satisfying result on the tradeoff, there is a need to first give an upper bound to the approximation error, $||\beta_\tau^* - \beta^*||_2$.

Proposition 1 *If condition 1 holds, and* v_δ *the average of the* $\delta + 1$ *moments is finite, then the vector* β_τ^* *of Huber regression coefficients satisfies :*

$$||\beta_\tau^* - \beta^*||_2 \leq 2c_l^{-1/2}v_\delta\tau^{-\delta}$$

if $\tau \geq (4v_{2)_{\frac{1}{1+\delta}}}$ *for* $0 < \delta < 1$ *or* $\tau \geq (2v_1)^{1/2}M$ *else, for* $M = max_i||S_n^{-1/2}x_i||_2$

Which gives a tight bound for the approximation bias of the order of $\tau^{-\delta}$, showing that if τ indeeds tends to ∞, then the approximation bias disappears at a speed of δ, and that as the order of the highest bounded moment increases, so does the tightness of the bound.

Theorem 1 *If Condition 1 holds, and* $v_\delta < +\infty$ *for some* $\delta > 0$, *with* $L = max_i||x_i||_\infty$ *and* $n \geq C(L, c_l)d^2t$ *for* $C(L, c_l) > 0$, *then,* $\forall t > 0,\ \tau_0 \geq v_\delta$, *the estimator* $\widehat{\beta_\tau}$ *with* $\tau = \tau_0(\frac{n}{t})^{max(\frac{1}{1+\delta},\frac{1}{2})}$ *satisfies :*

$$||\widehat{\beta_\tau} - \beta^*||_2 \leq 4c_l^{-1}L\tau_0 d^{1/2}(\frac{t}{n})^{min(\delta/1+\delta,1/2)}$$

and that with probability at least $1 - (2d + 1)e^{-t}$

Showing that with only a $(1+\delta)$-th bounded moment, the Huber estimator has an upper bound of the order of $d^{1/2}n^{-min(\delta/1+\delta,1/2}$. Clearly, the lower the dimensionality of the data the tighter the bound, and once again, if $\delta > 1$, the convergence is of the order of data with sub-gaussian tail. To show this bounds optimality is a matching lower bound needed. Let us denote $\mathcal{P}_{\delta}^{v_\delta}$ the distribution on $\mathbb{R}$ with a bounded absolute $(1+\delta)$-th moment, and $X = (x_1, \ldots, x_n)^t$, $\mu_n = u : u \in \{-1; 1\}$, the lower bound is given by

Theorem 2 ϵ_i *i,i,d from a distribution in* $\mathcal{P}_{\delta}^{v_\delta}$*, with* $\delta > 0$*. Suppose* $\exists u$ *in* μ_n *such that* $\|\frac{1}{n}X^t u\|_{min} \geq \alpha$ *for some* $\alpha > 0$*, then,* $\forall t \in [0, n/2]$, $\widehat{\beta} = \widehat{\beta}(y_1, \ldots, y_n, t)$ *then*

$$\underset{sup}{\mathbb{P} \in \mathcal{P}_{\delta}^{v_\delta}} \mathbb{P}[\|\widehat{\beta} - \beta^*\|_2 \geq \alpha c_u^{-1} v_\delta d^{1/2} (\frac{t}{n})^{min(\delta/(1+\delta),1/2)}] \geq \frac{e^{-2t}}{2}$$

4. The Adaptive Huber Regression in high dimensions

This section will treat the Adaptive Huber Estimator in high dimensions; where d is allowed to grow with n exponentially ($d >> n$). We will be denoting by $H_\tau(\beta) = \nabla^2 \mathcal{L}_\tau(\beta)$ the Hessian, $S = supp(\beta^*) \subset \{1, \ldots, d\}$ the number of non zero elements of β, with $|S| = s$. The following property will be required for the bounds to hold.

H_τ satisfies the localized restricted eigenvalues condition, that is, $\kappa_l \leq \kappa_-(k, \gamma, \pi) \leq \kappa_+(k, \gamma, \pi) \leq \kappa_u$ for some constants $\kappa_u, \kappa_l > 0$

to be clear, let's remind what are the localized restricted eigenvalues :

Definition 4 *The localized restricted eigenvalue of* H_τ *is defined as*

$$\kappa_+(m, \gamma, r) = sup\{< u, H_\tau(\beta)u >: (u, \beta) \in C(m, \gamma, r)\},$$
$$\kappa_-(m, \gamma, r) = inf\{< u, H_\tau(\beta)u >: (u, \beta) \in C(m, \gamma, r)\},$$

where $C(m, \gamma, r) := \{(u, \beta) \in \mathbb{S}^{d-1} \times \mathbb{R}^d : \forall J \in \{1, \ldots, d\}$ *satisfying* $S \in J, |J| \leq m$, $\|u_{J^c}\|_1 \leq \gamma \|u_J\|_1, \|\beta - \beta^*\|_1 \leq r$ *is a local* l_1 *cone*

as well as the restricted eigenvalues :

Definition 5 *The restricted maximum and minimum eigenvalues of* S_n *are defined respectively as :*

$$\rho_+(m, \gamma) = sup\{< u, S_n u >: u \in C(m, \gamma\},$$
$$\rho_-(m, \gamma) = inf\{< u, S_n u >: u \in C(m, \gamma)\},$$

where $C(m, \gamma) = \{u \in \mathbb{S}^{d-1} : \forall J \subset \{1, \ldots, d\}$ *satisfying* $S \subset J, |J| \leq m, \|u_{J^c}\|_1 \leq \gamma \|u_J\|_1\}$

Another restriction is that :

S_n satisfies the restricted eigenvalue condition : $\kappa_l \leq \rho_- \leq \rho_+ \leq \kappa_u$

The conditions are all set to present the theoretical upper bound for the AHE in high dimensions

Theorem 3 *We assume condition 3 holds with* $(k, \gamma) = (2s, 3), v_\delta < \infty$ *for some* $0 < \delta \leq 1$*. For any* $t > 0$ *and* $\tau_0 \geq v_\delta$*, let* $\tau = \tau_0(n/t)^{max(1/1+\delta,1/2)}$, $\lambda \geq 4L_{\tau 0}(t/n)^{min(\delta/(1+\delta),1/2)}$ *and* $r > 12\kappa_l^{-1}s\lambda$ *Then with probability at least* $1 - (2s+1)e^{-t}$*, the* l_1 *regularized Huber estimator* $\widehat{\beta}_{\tau,\lambda}$ *satisfies :*

$$\|\widehat{\beta}_{\tau,} - \beta^*\|_2 \leq 3\kappa_l^{-1} s^{1/2} \lambda \tag{7}$$

as long as $n \geq C(L, \kappa_l)s^2 t$ *for some* $C(L, \kappa_l)$ *depending only on* $(L, kappa_l)$*. In particular, with* $t = (1+c)log(d)$ *for* $c > 0$ *we have*

$$\|\widehat{\beta}_{\tau,\lambda} - \beta^*\|_2 \leq (1/\kappa_l)L\tau_0 s^{1/2} (\frac{(1+c)log(d)}{n})^{min(\delta/(1+\delta),1/2)} \tag{8}$$

that with probability at least $1 - d^{-c}$

This upper bound guarantees that the regularized AHE converges at the rate of $s^{1/2}(\frac{log(d)}{n}))^{min(\delta/1+\delta,1/2)}$, which is the same rate of convergence as the lasso, except that we have dropped the sub-gaussian tail requirement.

A s was the case for the low dimensional case, a matching lower bound is required.

Theorem 4 *(lower bound) Regression errors are iid from a distribution* $\mathcal{P}_{\delta}^{v_\delta}$ *. Suppose cond 3 with* $k = 2s$ *and* $\gamma = 0$*. There exists a set* $\mathcal{A}$ *qt* $|\mathcal{A}| = s$ *and* $u \in \mathcal{U}_n$ *st* $\|X_A^T u/n\|_{min} \geq \alpha$ *for some* $\alpha > 0$*. Then for any* $A > 0$ *and* $s - sparse$ *estimator* $\widehat{\beta} = \widehat{\beta}(y_1, \ldots, y_n, A)$ *possibly depending on A, we have :*

$$sup_{P \in \mathcal{P}_{\delta}^{v_\delta}} P[\|\widehat{\beta} - \beta^*\|_2 \geq v_\delta \frac{\alpha s^{1/2}}{\kappa_u} (\frac{Alog(d)}{2n})^{min(\delta/(1+\delta),1/2)}] \geq (1/2)d^{-A}, \tag{9}$$

as long as $n \geq 2(Alog(d) + log(2))$

5. The case of Heavy Tailed data

So far, the Adaptive Huber Regression was treated in the general case, however, one aspect of this type of regression is its robustness to heavy tailed data.

To extend the idea of Adaptive Huber Regression to heavy tails, the studied framework will be this one : A high dimensional regime where $d >> n, \beta^* \in \mathbb{R}^d$ is sparse, and $s = \|\beta^*\|_0 << n$. In the case of Huber Regression, the "linear" part of the loss, i.e when $x > \tau$, penalizes the residuals in order to robustify the quadratic loss as the outliers will be down weighted or all around removed. In this way, the AHE would not necessarily be robust against heavy tailed covariates. To deal with this **article** devised some modification to the regression to let us robustify the covariates and regression errors.

As usual, we observe iid data $\{(y_i, x_i)\}_{i=1}^n$ associated to the linear model $y =< x, \beta^* > +\epsilon$. To robustify x_i, is introduces the truncated covariates $x_i^u = (\psi_u(x_{i1}), \ldots, \psi_u(x_{id}))^T$, with $\psi_u(x) = min\{max(-u, x), u\}$ and $u > 0$ is a tuning parameter. The new modified adaptive huber estimator is then the minimizer of another problem :

$$\widehat{\beta}_{\tau,u,\lambda} \in argmin_{\beta \in \mathbb{R}^d}\{\mathcal{L}_\tau^u(\beta) + \lambda\|\beta\|_1\}$$

with the loss modified in order to accomodate to the new tuning parameter $\mathcal{L}_\tau^u(\beta) = \frac{1}{n}\sum_{i=1}^n l_\tau(y_i - < x_i^u, \beta >)$ and $\lambda > 0$ is the regularization parameter. One restriction is however needed to give a concentration bound in the case of the heavy-tailed design. With $H^u_\tau(\beta) = \nabla^2\mathcal{L}^u_\tau(\beta)$ the hessian matrix of the Huber loss modified for the heavy tailed data.

- $\mathbb{E}[\epsilon] = 0$
- $\sigma^2 = \mathbb{E}[\epsilon^2] > 0$
- $v_3 = \mathbb{E}[\epsilon^4] < \infty$
- $x = (x_1, \ldots, x_d)^T \in \mathbb{R}^d$ is independent to ϵ and satisfies $M_4 = max_{1\leq j\leq d}\mathbb{E}[x_j^4] < \infty$

Under these conditions, can the following concentration bound hold.

Theorem 5 *Let $H^u_\tau(\cdot)$ satisfy condition 2 where $k = 2s, \gamma = 3$ and $r > 12\kappa_l^{-1}\lambda s$. The modified Adaptive Huber Estimator $\widehat{\beta}_{\tau,u,\lambda}$ satisfies, on the event that $\mathcal{E}(\tau, u, \lambda) = \{\|(\nabla\mathcal{L}^u_\tau(\beta^*))_S\|_\infty \leq \lambda/2\}$ that*

$$\|\widehat{\beta}_{\tau,u,\lambda} - \beta^*\|_2 \leq 3\kappa_l^{-1}s^{1/2}\lambda$$

where for any $t > 0$, the triplet (τ, u, λ) satisfies

$$\lambda \geq 2M_4\|\beta^2\|_2 s^{1/2}u^{-2} + 8\{v_2M_2^{1/2} + M_4\|\beta^*\|_2^3 s^{3/2}\}\tau^{-2}$$
$$+2(2\sigma^2M_2 + 2M_4\|\beta^*\|_2^2 s)^{1/2}(\frac{t}{n})^{1/2} + u\tau\frac{t}{n}$$

where $v_2 = \mathbb{E}[|\epsilon|^3]$ and $M_2 = max_1\mathbb{E}(x_j^2)$ than $\mathbb{P}(\mathcal{E}(\tau, u, \lambda))1 - 2se^{-t}$

where the near optimal convergence for our estimator is of a rate of $s(\frac{log(d)}{n})^{1/2}$ when the hyperparameters scale as :

$$\tau \asymp s^{1/2}(\frac{n}{log(d)})^{1/4},\ u \asymp (\frac{n}{log(d)})^{1/4},\ \lambda \asymp \sqrt{\frac{slog(d)}{n}}$$

6. Implementation

In this section will be presenting how to implement the Regularized Adaptive Huber Regression. Methods such as the interior point are not scalable, defeating the purpose of the Adaptive Huber Approach. Oneway to solve this problem is with the local adaptive majorize-minimization principle, the LAMM, presented in (6).

We say that a function $g(\beta \mid \beta^{(k)})$ majorizes $f(\beta)$ at the point $\beta^{(k)}$ if

$$g(\beta \mid \beta^{(k)}) \geq f(\beta) \qquad \text{and } g(\beta^{(k)} \mid \beta^{(k)}) = f(\beta^{(k)}) \tag{10}$$

To minimize a general function $f(\beta)$ a majorize-minimization (MM) algorithm ini-tializes at $\beta^{(0)}$, and then iteratively computes $\beta^{(k+1)} = argming(\beta \mid \beta^{(k)})$ for $k = 0, 1, ...$ The objective value of such an algorithm decreases in each step, since

$$f(\beta^{(k+1)}) \leq g(\beta^{(k+1)} \mid \beta^{(k)}) \leq g(\beta^{(k)} \mid \beta^{(k)}) = f(\beta^{(k)}) \tag{11}$$

As pointed out by Fan et al. (2018), the majorization requirement only needs to holdlocally at $\beta^{(k+1)}$ when starting from $\beta^{(k)}$ We therefore locally majorize $L_\tau(\beta)$ at $\beta^{(k)}$ by an isotropic quadratic function.

$$g(\beta \mid \beta^{(k)}) = L_\tau(\beta^{(k)}) + < \nabla L_\tau(\beta^{(k)}), \beta - \beta^{(k)} > + \frac{\phi_k}{2} \parallel \beta - \beta^{(k)} \parallel_2^2 \tag{12}$$

where ϕ_k is a quadratic parameter such that $g(\beta^{(k+1)} \mid \beta^{(k)}) \geq L_\tau(\beta^{(k+1)})$.

The isotropic form also allows a simple analytic solution to the subsequent majorized optimization problem:

$$\beta^{(k+1)} = T_{\lambda,\phi_k}(\beta^{(k)}) = S(\beta^{(k)} - \phi_k^{-1}\nabla L_\tau(\beta^{(k)}), \phi_k^{-1}\lambda) \tag{13}$$

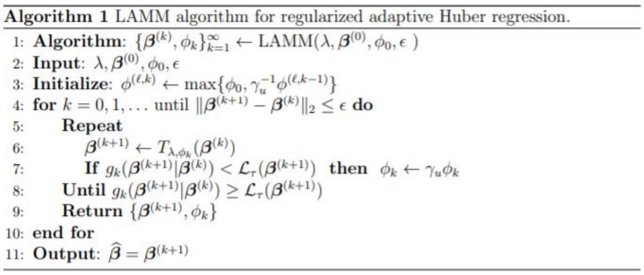
It can be shown that is minimized at

$$min_{\beta\in\mathbb{R}^d}\left\{L_\tau(\beta^{(k)}) + < \nabla L_\tau(\beta^{(k)}), \beta - \beta^{(k)} > + \frac{\phi_k}{2} \parallel \beta - \beta^{(k)} \parallel_2^2\right\} \tag{14}$$

where S(x,λ) is the soft-thresholding operator defined by

$$S(x, \lambda) = sign(x_i)max(\mid x_i \mid -\lambda, 0) \tag{15}$$

The simplicity of this updating rule is due to the fact that is an unconstrained optimization problem.



Algorithm 1 LAMM algorithm for regularized adaptive Huber regression.

```
1: Algorithm: $\{\beta^{(k)}, \phi_k\}_{k=1}^\infty \leftarrow \text{LAMM}(\lambda, \beta^{(0)}, \phi_0, \epsilon)$
2: Input: $\lambda, \beta^{(0)}, \phi_0, \epsilon$
3: Initialize: $\phi^{(\ell,k)} \leftarrow \max\{\phi_0, \gamma_u^{-1}\phi^{(\ell,k-1)}\}$
4: for $k = 0, 1, \ldots$ until $\|\beta^{(k+1)} - \beta^{(k)}\|_2 \leq \epsilon$ do
5:     Repeat
6:         $\beta^{(k+1)} \leftarrow T_{\lambda,\phi_k}(\beta^{(k)})$
7:         If $g_k(\beta^{(k+1)}|\beta^{(k)}) < \mathcal{L}_\tau(\beta^{(k+1)})$ then $\phi_k \leftarrow \gamma_u\phi_k$
8:     Until $g_k(\beta^{(k+1)}|\beta^{(k)}) \geq \mathcal{L}_\tau(\beta^{(k+1)})$
9:     Return $\{\beta^{(k+1)}, \phi_k\}$
10: end for
11: Output: $\widehat{\beta} = \beta^{(k+1)}$
```

7. Gradient descent for Adaptive Robust Loss Function

In (2) is proposed a useful approach for gradient descent. A first-order iterative optimization algorithm for finding the minimum of the Adaptive Robust Loss Function seen above.To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient (or approximate gradient) of the function at the current point. If, instead, one takes steps proportional to the positive of the gradient, one approaches a local maximum of that function; the procedure is then known as gradient ascent. Gradient descent was originally proposed by Cauchy in 1847. To enable gradient-based optimization we can derive the derivative of $\rho(x, \alpha, c)$ with respect to x:

$$\frac{\partial\rho}{\partial x}(x, a, c) = \begin{cases} \frac{x}{c^2} & \alpha = 2 \\ \frac{2x}{x^2+2c^2} & \alpha = 0 \\ \frac{x}{c^2}exp(\frac{-1}{2}(\frac{x}{c})^2) & \alpha = -\infty \\ \frac{x}{c^2}(\frac{(x/c)^2}{|\alpha-2|} + 1)^{(\alpha/2-1)} & otherwise \end{cases} \tag{16}$$

The shape of the derivative gives some intuition as to how α affects behavior when our loss is being minimized by gradient descent or some related method. For all values of α the derivative is approximately linear when $\mid x \mid < c$ so the effect of a small residual is always linearly proportional to that residual's magnitude. If $\alpha = 2$. the derivative's magni-tude stays linearly proportional to the residual's magnitude — a larger residual has a correspondingly larger effect.If $\alpha = 1$ the derivative's magnitude saturates to a constant $\frac{1}{c}$ as $\mid x \mid$ grows larger than c, so as a residual increases its effect never decreases but never exceeds a fixed amount. If $\alpha < 1$ the derivative's magnitude begins to decrease as $\mid x \mid$ grows larger than c so as the residual of an outlier increases, that outlier has less effect during gradient descent. The effect of an outlier diminishes as α becomes more negative, and as α approaches $-\infty$ an outlier whose residual magnitude is larger than 3c is almost completely ignored.

We can also reason about α in terms of averages. Because the empirical mean of a set of values minimizes total squared error between the mean and the set, and the empirical median similarly minimizes absolute error, minimizing our loss with $\alpha = 2$ is equivalent to estimating a mean, and with $\alpha = 1$ is similar to estimating a median. Minimizing our loss with $\alpha = -\infty$ is equivalent to local mode-finding.Values of α between these extents can be thought of as smoothly interpolating between these three kinds of averages during estimation.

The loss function has several useful properties that we will take advantage of. The loss is smooth with respect to x, α and $c > 0$ and is therefore well-suited to gradient-based optimization over its input and its param-eters.
We can initialize α such that the loss is convex and then gradually reduce α during optimization, thereby enabling robust esti-mation that avoids local minima.

8. Numerical Studies

8.1. Finite Sample Performance

For numerical studies and real data analysis, in the case where the actual order of moments is unspecified, we presume the variance is finite and we generate data from the linear mode :

$$y_i =< x_i, \beta^* > +\epsilon_i \qquad i = 1, .., n \qquad (17)$$

here ϵ_i are i.i.d. regression errors and β^*=$(5, -2, 0, 0, 3, \underbrace{0, ..., 0}_{\text{d-5}})^T \in R^d$.

Independent of ϵ_i, we generate x_i from standard multivariate normal distribution N(0,I_d).As the original paper, we set (n,d) = (100,5), and generate regression errors from three dif-ferent distributions :

- the normal distribution $\mathcal{N}(0, 4)$
- the t-distribution with degrees of freedom 1.5
- the log-normal distribution $log\mathcal{N}(0, 4)$

Both t and log-normal distributions are heavy-tailed, and produce outliers with high chance.

Table 1. Results for adaptive Huber regression (AHR) and ordinary least squares(OLS) when n= 100 and d= 5. The mean and standard deviation (std) of ℓ_2-error based on 100 simulations are reported.

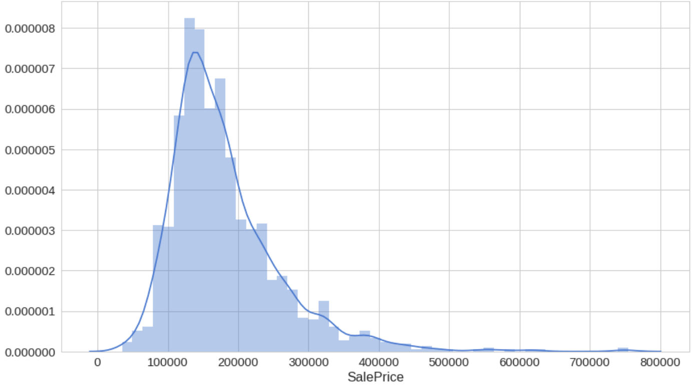
Noise	AHR		OLS	
	mean	std	mean	std
Normal	3.779	0.550	3.817	0.538
Student's t	46.358	213.389	46.414	213.403
Log-normal	3.685	0.559	3.728	0.575

The results on ℓ_2-error for adaptive Huber regression and the least squares es-timator, averaged over 100 simulations, are summarized in Table 1. In the case of normally distributed noise, the adaptive Huber estimator performs as well as the least squares. With heavy-tailed regression errors following Student's t or log-normal distribution, the adaptive Huber regression significantly outperforms the least squares.

8.2. A Real Data Example: House Price Prediction

Founded in 2010, Kaggle is a Data Science platform where users can share, collaborate, and compete. One key feature of Kaggle is "Competitions", which offers users the ability to practice on real-world data and to test their skills with, and against, an international community.

We'll work through the House Prices: Advanced Regression Techniques competition.The challenge is to predict the final sale price of the homes. This information is stored in the SalePrice column.



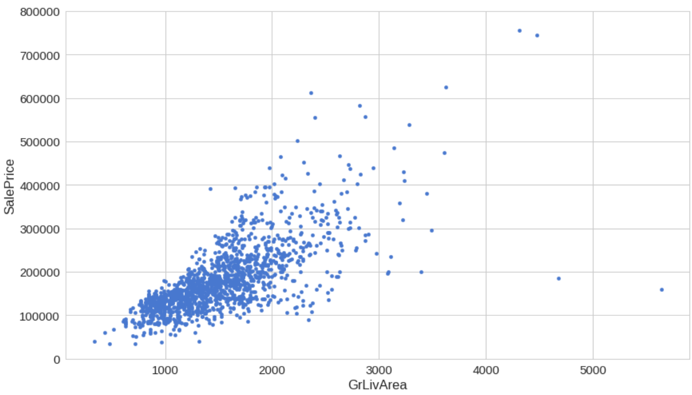
The average sale price of a house in the dataset is close to $180,000, with most of the values falling within the $130,000 to $215,000 range.It is apparent that SalePrice doesn't follow normal distribution, so before performing regression it has to be transformed. While log transformation does pretty good job, the best fit is obtained through unbounded Johnson distribution.

In order to improve the linearity of the data, we will perform a log-transformation on the target variable. Of course it means that the resulting predictions will be logarithms of the true ones. The inverse transformation will therefore be done.

In order to understand the data, we can look at each variable and try to understand their meaning and relevance to this problem.

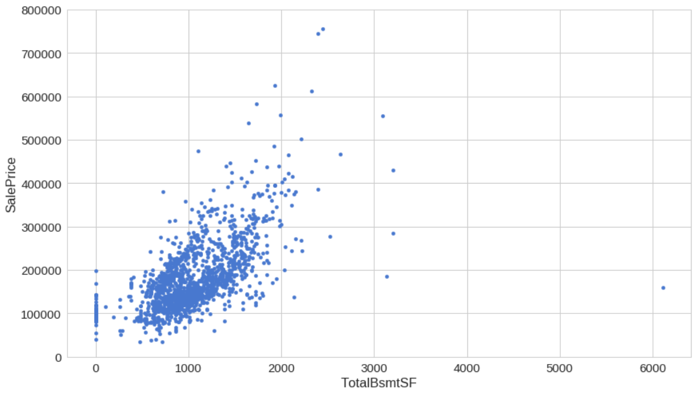
We can filter the spreadsheet and look carefully to the variables with 'High' 'Expectation'. Then, we can rush into some scatter plots between those variables and 'SalePrice', filling in the 'Conclusion' column which is just the correction of our expectations.

If you take a look at the greater living area (square feet) against the sale price You might've expected that larger living area should mean a higher price.



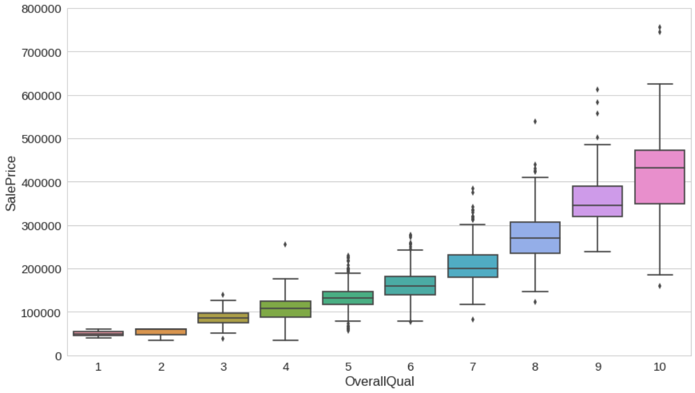
The above chart shows that it's generally correct. But we still have those 2–3 "cheap" houses offering huge living area.

One column you might not think about exploring is the TotalBsmtSF — Total square feet of the basement area, but let's do it anyway:



The basement area seems like it might have a lot of predictive power for our model.We can feel tempted to eliminate some observations (e.g. TotalBsmtSF > 3000) but We suppose it's not worth it. We can live with that, so we'll not do anything.

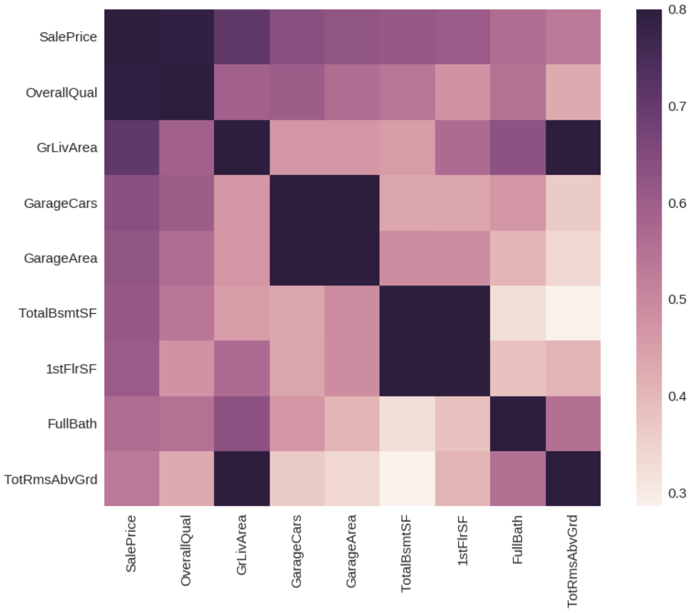
Finally. Let's look at OverallQual — overall material and finish quality. Of course, this one seems like a much more subjective feature, so it might provide a bit different perspective on the sale price.



We can see both OverallQual and GrLivArea and TotalBsmtSF follow a linear model, but have some outliers we may want to look into. For instance, there are multiple houses with an overall quality of 10, but have suspisciously low prices. We can see similar behavior in GrLivArea and TotalBsmtSF. GarageCars and GarageArea both follow more of a quadratic fit. It seems that having a 4 car garage does not result in a higher house price and same with an extremely large area.

Heatmaps are great to detect this kind of multicollinearity situations and in problems related to feature selection like this project, it comes as an excellent exploratory tool.

Let's have a more general view on the top 8 correlated features with the sale price:



Another aspect we observed here is the 'SalePrice' correlations.As it is observed that 'GrLivArea', 'TotalBsmtSF', and 'OverallQual' are highly correlated to SalePrice, however we cannot exclude the fact that rest of the features have some level of correlation to the SalePrice.

To measure the predictive performance, we consider a robust prediction loss: the mean absolute error (MAE) defined as :

$$\frac{1}{n_{test}} = \sum_{i}^{n_{test}} | y_i^{test} - < x_i^{test}, \hat{\beta} > |$$

where y_i^{test} and x_i^{test},i=1,...,n_{test} denote the observations of the response and predictor variables in the test data, respectively.

Table 2 reports the MAE for the considered methods.

Method	MAE
LM	0.1144187
Lasso	0.1005206
AHuber	0.1070608

Lasso clearly shows the smallest MAE, followed by AHuber and Linear Model. The Lasso produces a fairly large model despite the small sample.Now it has been recognized that Lasso tends to select many noise variables along with the significant ones, especially when data exhibits heavy tails.

Due to the singularity of the input matrix we can not use the standard Huber Regression,in our experiment, we have seen that if we have a singular input matrix, the adaptive Huber regression will have the same performance as the standard regression algorithms, but if we condition the fact that the input must be nonsingular, the Adabtive huber regression will surpass the standard regression algorithm.

9. In depth study of the LAMM solving algorithm in general and in our case

To implement the Regularized Adaptive Huber Regression is the LAMM algorithm needed. This section will develop the inner working of the LAMM algorithm, and how to use it in the case of the Huber Regression by basing itself on (6)

LAMM stands for local adaptive majorize-minimization, it both controls algorithmic complexity as well as the statistical error, while fitting to high dimensional models. It works in two stages, the first solving a convex program to get a rough initial estimator, refined on the second stage by solving a sequence of convex programs with smaller precision tolerances. It results in optimal statistical performances and controlled complexity for a large amount of non convex optimization problems. The usual way to get an estimator is by solving the penalized M-Estimator problem :

$$\widehat{\beta} = argmin_{\beta\in\mathbb{R}^d}\{\mathcal{L}(\beta) + \mathcal{R}_\lambda(\beta)\}$$

Where $\mathcal{L}(\cdot)$ is a smooth loss function, and $\mathcal{R}_\lambda(\cdot)$ is a sparsity-inducing penalty with a regularization parameter λ. In the Huber regression case, the loss function is indeed smooth. The classical L_1 penalty used in the regression problem also is a sparsity inducing penalty, as was explained earlier. One strength of the LAMM algorithm is that it helps solving this kind of problem even with a non convex penalty.

The estimator is obtained by solving a sequence of convex programs of the form

$$min_{\beta\in\mathbb{R}^d}\{\mathcal{L}(\beta) + \mathcal{R}(\lambda^{l-1} \odot \beta)\},\ forl = 1, \ldots, T$$

where $\lambda^{l-1} = (\lambda w(|\widetilde{\beta_1^{l-1}}), \ldots, \lambda w(|\widetilde{\beta_d^{l-1}}))$ where the $\beta^{(l)}$ are approximate solution of the l^{th} optimization problem, and w is a weighting function. Specifically form a tightening function class such that :

1. $\forall t_1 \geq t_2 \geq 0, w(t_1) \leq w(t_2)$
2. if $t \geq 0$, $w(t) \in [0, 1]$
3. if $t \leq 0$, $w(t) = 0$

The Majorize minimization algorithm comes to play on the first part of the LAMM, the basic algorithm is described clearly in (11). The MM algorithm basically minimize a function $f(\beta)$ at a given point $\beta^{(k)}$, by majoring it by a function $g(\beta|\beta^{(k)})$ under constraints, and computing its argmin. The idea of the first step of LAMM is to majorize the optimization problem at the l step by an isotropic quadratic function and coupling it with Taylor's in order to use the largest eigenvalue of the Hessian of the loss (This is where the restricted eigenvalue condition comes into play). These steps leads to having the solution to a step of the optimization problem to be :

$$\beta^{l,1} = T_{\lambda^{(l-1)},\phi}(\beta^{(l-1)} \equiv S(\widetilde{\beta^{(l-1)}} - \phi^{-1}\nabla\mathcal{L}(\widetilde{\beta^{(l-1)}}), \phi^{-1}\lambda^{l-1})$$

where $S(x, \lambda) = (sign(x_j).max\{|x_j| - \lambda_j, 0\}$ and ϕ is a constant bigger than the largest eigenvalues of the Hessian of the loss. Getting this ϕ is done by starting from a small isotropic parameter ϕ_0 and inflating it by a factor $\alpha > 1$. For each of the l steps, we run the small problem k times in order to get a $\phi = \alpha_u^{k-1}\phi_0$. The LAMM algorithm will solve at $\widetilde{\beta^{l-1}}$ will start with a $\phi_0 = 10^{-6}$ and increase it by the factor α inside the l^{th} step of optimization, to compute $\beta^{(l,1)} = T_{\lambda^{(l-1)},\phi^{(l,k)}}(\beta^{(l,0)}$ and having $\phi^{(l,k)} = \alpha_u^{k-1}\phi_0,\ \beta^{(l,0)} = \widetilde{\beta^{(l-1)}}$, and repeats to have $\beta^{(l,k)}$.

The steps of the optimization are separated into two stages, the contraction, when $l = 1$ and the tightening, when $l > 1$. With the contraction step, the initial value is arbitrary. This stage aims to find a good initial estimator for $\widetilde{\beta}^{(1)}$ for the subsequent optimization subproblems in the tightening stage. However, this first estimator suffers from a suboptimal rate of convergence, inferior to the one obtained by nonconvex regularization. The tightening stage refines this coarse contraction estimator into the optimal region of convergence. This is done by the steps where $l \geq 2$. As the initial estimator is already good and sparse, at each tightening step, the LAMM algorithm has geometric rate of convergence due to the sparse strong convexity.

10. Beyond the Huber Adaptive Regression : Removing the Tuning problem

In the Huber Adaptive Regression, the parameter τ has to be given in advance, however its calibration schemes are based on cross-validation or on Lepski's method (12), which is expensive, especially when we have high dimensional data.

In the original paper, (10), the $\tau = 1.345\sigma$ is used to have a 95% efficiency for normally distributed data. As was developed earlier this is indeed a poor choice for heavy-tailed data and will always perform sub-optimally.

If the variance of errors is finite, which is a desqrt strong restriction, (14) gives a way to get a good τ by simply solving a system of equations instead of using computationally costly methods in order to get a good bias-robustification tradeoff.

To do so, in the same framework as usual, let's introduce the truncated sample mean $m_\tau = \frac{1}{n}\sum_{i=1}^n \psi_\tau(X_i)$ for some $\tau > O$, with $\psi_\tau(x) = sign(x)min(|x|, \tau), x \in \mathbb{R}$, as well as $v_2 = \sqrt{\mathbb{E}[\mathbb{X}^{\nvdash}]}$ which is considerated as finite. With an ideal τ, the truncated data $\psi_\tau(X_1), \ldots, \psi_\tau(X_n)$ is a good estimator of the mean, and weakens the outliers as the truncation is well done. With this method of truncation, even with $\tau = v_2\sqrt{(n/z)}$ the deviation of the estimator scales with v_2 instead of the standard deviation, when the optimal deviation enjoyed by the sample mean with sub-gaussian data is of the order of $\sigma\sqrt{(z/n)}$, as is developed in (5). The Huber loss is continuously differentiable with $l'_\tau(x) = \psi_\tau(x)$, and we have therefore the adaptive huber estimator is

$$0 = \sum_{i=1}^n \psi_\tau(X_i - \theta) = \sum_{i=1}^n min(|X_i - \theta|, \tau)sign(X_i - \theta)$$

to solve in θ. This returns to the main result from (5) :

Theorem 6 *Let $z \geq 0$ and $v \geq \sigma$. Provided $n \geq 8z$, $\widehat{\beta}_\tau$, with $\tau = v\sqrt{n/z}$ satisfies the bound $|\widehat{\beta}_\tau - \beta| \leq 4v\sqrt{z/n}$ with probability at least $1 - 2e^{-z}$*

Which explains that $\tau \sim \sigma\sqrt{n/z}$. In order to find an option to do tuneless Adaptive Huber Regression, we can say that :

$$\frac{\mathbb{E}[\psi_\tau^2(X - \beta)]}{\tau^2} = \frac{\mathbb{E}[min((X - \beta)^2, \tau^2)]}{\tau^2} = \frac{z}{n},\ \tau > 0 \tag{18}$$

which has an unique solution $\tau_{z,\beta}$ which would satisfy $\sqrt{\mathbb{E}[min((X - \beta)^2, q_{z/n})]} \leq \tau_{z,\beta} \leq \sigma\sqrt{n/z}$ where q_α represents $q_\alpha = inf(t : \mathbb{P}(|X - \beta| > t) \leq \alpha)$. If n is large, $\tau_{z,\beta}\ \sigma\sqrt{n/z}$ as $n \rightarrow +\infty$. From this stems the fact that a good estimate of β can be obtained by simply solving

$$\begin{cases} f_1(\beta,\tau) = \sum_{i=1}^n \psi_\tau(X_i-\theta) = 0, \\ f_2(\beta,\tau) = \frac{1}{n}\sum_{i=1}^n min((X_i-\theta)^2,\tau^2)/\tau^2 - \frac{z}{n} = 0 \end{cases} \theta \in \mathbb{R}, \tau > 0 \tag{19}$$

This is solved by computing a sequence of $(\theta^{(k)},\tau^{(k)})_{k\geq1}$ satisfying $f_2(\theta^{(k-1)},\tau^{(k)} = 0$ and $f_1(\theta^{(k)},\tau^{(k)} = 0$ and θ^l is the robust estimator of β

10.1. The case of tuning free high dimensional Huber Regression

The ideas presented previously were for mean estimation, but were necessary to understand how to do a robust linear regression in a data-adaptative framework. The regression model is as usual

$$Y_i = X_i^T\beta^* + \epsilon_i,\ i = 1,\dots,n$$

10.1.1. Low dimension

As was explained earlier, the solution to Huber's M-estimator is defined as

$$\widehat{\beta_T} \in_\beta \sum_i^n l_\tau(Y_i - Z_i^T\beta)$$

The convexity of the Huber loss shows that the solution is determined by $\sum_i^n \psi_\tau(Y_i - Z_i^T\widehat{\beta_T})Z_i = 0$.

Theorem 7 *Let ϵ, X be independant and the function which associates α to $\mathbb{E}[l_\tau(\epsilon-\alpha)]$ admit an unique minimizer which satisfies*

$$\mathbb{P}(|\epsilon-\alpha_\tau| \leq \tau) > 0$$

and assume that $\mathbb{E}[ZZ^T]$ be positive definite, then

$$\beta^*_{0,,au} = \beta^*_0 + \alpha_\tau\ and\, \beta^*_\tau = \beta^*$$

Moreover, α_τ with $\tau > \sigma$ satisfies the bound :

$$|\alpha_\tau| \leq \frac{\sigma^2 - \psi^{\not\prime}_\tau(\epsilon)}{1-\tau^{-1}\sigma^2}$$

This shows that if the distribution of the errors is asymmetric, then α_τ is non zero for any $\tau > 0$, and the smaller τ is, the larger the bias becomes.

A two step method is introduced. In the first step, is the problem solved with a simple $\tau = c\sigma$ with the constant $c = 1.345$ to get the classical 95% efficiency for the normal model. σ can be estimated simultaneously with β^* by solving a system of equations, this is explained in (10). The problem is simply solved iteratively by starting with an initial estimate for $\beta^{(0)}$ and simply use $\widehat{\sigma}^{(k)}$ to get a new $\beta^{(k+1)}$. This involves two steps;

Step one : Scale estimation With the current estimate $\beta^{(k)}$ is computed the vector of residuals $r^{(k)}$ and the robustification parameter $\tau^{(k)} = 1.345\widehat{\sigma^{(k)}}$ where $\widehat{\sigma^{(k)}}$ is the median absolute deviation

Step two : Weighted least squares The second step is to simply compute the $n\times n$ matrix $W^{(k)} = diag((1+w_1^{(k)})^{-1},\dots,(1+w_n^{(k)-1}))$ where $w_i^{(k)} = |r_i^{(k)}|/\tau^{(k)} - 1$ if $|r_i^{(k)}| > \tau^{(k)}$ and 0 else. The update rule is simple, it is

$$\beta^{(k+1)} = argmin_\beta \sum_{i=1}^n \frac{(Y_i - Z_i^T\beta)^2}{1+w_i^{(k)}} = (Z^TW^{(k)}Z)^{-1}Z^TY$$

In the second step, $\beta_0^* = \mathbb{E}[\delta_i]$ with $\delta_i = Y_i - X_i^T\beta^* = \beta_0^* + \epsilon_i$ are the residuals. So in order to estimate β_0^*, with the fitted residuals $\widehat{\delta_i} = Y_i - X_i^T\widehat{\beta^I}$ the vector of coefficient gotten from the final solution of the first step, is just needed to solve the following system of equations :

$$\begin{cases} f_1(\beta_0,\tau) = (\tau^2n)^{-1}\sum_{i=1}^n min((\widehat{\delta_i}-\beta_0)^2,\tau^2) - n^{-1}ln(n) = 0, \\ f_2(\beta_0,\tau) = \sum_{i=1}^n \psi_\tau(\widehat{\delta_i}-\beta_0) = 0; \end{cases} \tag{20}$$

Solving this gives the best estimator of β^*. For assymetric regression errors with heavy tails, the huber loss introduces bias to the intercept estimation but not the slope coeffictionts. The second step used the adaptive Huber method with a divergent to re estimate the intercept. The resulting β^* has a high degree of both unbiasedness and tail robustness.

10.1.2. High dimension

As we have previously seen, the tuning free approach needs to be modified quite a bit in high dimensions. As explained before : $d >> n$ and $\beta^* = (\beta_1^*,\dots,\beta_d^*)$ sparse such that $||\beta^*||_0 = s << n$. As before, the regularized problem is considered :

$$\widehat{\beta}_H(\tau,\lambda) \in_{\beta\in\mathbb{R}} (\mathcal{L}_\tau(\beta) + \lambda||\beta_{-0}||_1)$$

Getting to have optimal τ,λ by a grid search with cross validation is too long, so the following procedure is proposed in order to estimate β^* and tunes τ at the same time. The initial $\widehat{\beta^{(0)}}$ is initialised by Lasso, and iteratively, using the previous estimate $\beta^{(k-1)}$ is computed $\tau^{(k)}$ the solution of :

$$\frac{1}{n-\widehat{s^{(k-1)}}}\sum_i^n \frac{min((Y_i - Z_i^T\widehat{\beta^{(k-1)}})^2,\tau^2)}{\tau^2}$$

with $\widehat{s^{(k-1)}} = ||\widehat{\beta^{(k-1)}}||_0$. and with $\tau = \tau^{(k)}$ is computed $\widehat{\beta^{(k)}}$ by solving the regularized problem. Here, only λ is chosen by cross validation. As the convex optimization problem posed by the regularized version of Huber Adaptive regression admits a minimizer that satisfies the KKT conditions, it can be found by solving the problem :

$$\begin{cases} -n^{-1}\sum_i \psi_\tau(Y_i - Z_i^T\widehat{\beta}) = 0, \\ -n^{-1}\sum_i \psi_\tau(Y_i - Z_i^T\widehat{\beta})X_{ij} + \lambda\widehat{\eta_j} = 0,\ j = 1,\dots,d \\ \widehat{\beta}_j - S(\widehat{\beta}_j + \widehat{\eta}_j) = 0,\ j = 1,\dots,d \end{cases} \tag{21}$$

where $\widehat{\eta}_j \in \partial|\widehat{\beta}_j|$ and $S(z) = sign(z)(|z|-1)_+$. The solving procedure is an adaptated semismooth Newton Coordinate Descent proposed in (15).

11. Conclusion

The Huber regression is good balance between simply removing the outliers, and ignoring them. You can tune the amount of influence you would like to have in the overall estimation, by that giving room for those observations without allowing them "full pull" privileges.

We have revisit the Huber loss and robustification parameter, followed by the proposal of adaptive Huber regression in both low and high dimensions. We sharply characterize the non asymptotic performance of the proposed estimators . We describe the algorithm and implementation .and we devoted to simulation studies and a real data application.we finally extend the methodology to allow possibly heavy-tailed covariates/predictor.

References

[1] Damian Draxler generalized huber regression. `https://towardsdatascience.com/generalized-huber-regression-505afaff24c`. Accessed: 2019-08-20.

[2] Jonathan T. Barron. A more general robust loss function. *CoRR*, abs/1701.03077, 2017.

[3] Michael J. Black and Anand Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *Int. J. Comput. Vision*, 19(1):57–91, July 1996.

[4] P Charbonnier, L Blanc-Feraud, G Aubert, and M Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Image Processing, 1994. Proceedings. ICIP-94., IEEE Int. Conf.*, volume 2, pages 168 –172 vol.2, 1994.

[5] Jianqing Fan, Quefeng Li, and Yuyan Wang. Estimation of high dimensional mean regression in the absence of symmetry and light tail assumptions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79, 04 2016.

[6] Jianqing Fan, Han Liu, Qiang Sun, and Tong Zhang. I-lamm for sparse learning: Simultaneous control of algorithmic complexity and statistical error. *The Annals of Statistics*, 46, 07 2015.

[7] Stuart Geman and Donald E. McClure. Bayesian image analysis methods: An application to single photon emission tomography. 1985.

[8] F.R. Hampel. *Robust Statistics: The Approach Based on Influence Functions*. Probability and Statistics Series. Wiley, 1986.

[9] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall/CRC, 2015.

[10] Peter J. Huber. *Robust Statistics*, pages 1248–1251. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[11] Kenneth Lange. The mm algorithm. *Numerical Analysis for Statisticians*, pages 189–221, 2010.

[12] Oleg Lepski and Vladimir Spokoiny. Optimal pointwise adaptive methods in nonparametric estimation. *Annals of Statistics*, 25, 12 1997.

[13] Qiang Sun, Wen-Xin Zhou, and Jianqing Fan. Adaptive huber regression. *Journal of the American Statistical Association*, 0(0):1–24, 2019.

[14] Lili Wang, Chao Zheng, Wen Zhou, and Wen-Xin Zhou. A new principle for tuning-free huber regression. 2019.

[15] Congrui Yi and Jian Huang. Semismooth newton coordinate descent algorithm for elastic-net penalized huber loss regression and quantile regression. *Journal of Computational and Graphical Statistics*, 26(3):547–557, 2017.

[16] Zhengyou Zhang. Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting. Research Report RR-2676, INRIA, October 1995.