Qualifying Paper Report for Estimation of High Dimensional Mean Regression in the Absence of Symmetry and Light Tail Assumptions

Naitong Chen

November 25, 2022

1 Introduction

Linear regression is an easy-to-use and highly interpretable statistical inference method that has found itself in a wide range of applications. In the fields such as genomics and proteomics, we are often presented with high dimensional problems where the number of predictors is much greater than the number of observations. In such settings, linear regression in its simplest form may not have a unique solution, and so directly applying this method may not yield outputs that are interpretable for statistical inference. While a number of methods have been developed to address this issue, the corresponding theoretical guarantees on the inference quality for most of these methods rely on the assumption that the error distribution is symmetric and light-tailed. However, these assumptions may not be reasonable in practice, thus challenging the reliability of these methods. In this report, we discuss the regression estimator RA-lasso developed in Fan et al. (2017), which is designed to handle high dimensional data whose underlying error distribution may be neither symmetric nor light-tailed.

This report is organized as follows: for the remainder of this section, we more carefully motivate the RA-lasso estimator. In Section 2, we present the proposed method along with some intuition on its theoretical guarantees as well as some practical considerations when applying this method. We comment on their simulation studies in terms of how well they justify the claims made in the paper in Section 3 and reproduce their simulation studies in Section 4. This report is concluded with some final discussions of RA-lasso in Section 5.

In this report, we consider the linear regression model

$$y_i = x_i^T \beta^\star + \epsilon_i$$

where $\{x_i\}_{i=1}^n$ are independent and identically distributed p-dimensional covariate vectors and $\{\epsilon_i\}_{i=1}^n$ are independent and identically distributed errors (note that the paper also covers the heteroscedastic case where the error ϵ_i depends on x_i for all $i = 1, \ldots, n$, but we focus on the homoscedastic case in this report). $\beta^\star$ is a p-dimensional regression coefficient vector that is assumed to be weakly sparse. In other words, many elements of $\beta^\star$ is exatly or close to 0. We assume that $p \gg n$ and that both the distributions of x and ϵ have mean 0. The goal is to find $\hat{\beta} \in \mathbb{R}^p$ such that $x^T\hat{\beta}$ is close to $\mathbb{E}_\epsilon[y|x] = x^T\beta^\star$, the conditional expectation of the response y given x.

If we consider $\{x_i\}_{i=1}^n$ to be fixed and further assume that $\mathrm{Var}(\epsilon) = \sigma^2 < \infty$, then

$$\hat{\beta}_{OLS} \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \frac{1}{n} \sum_{i=1}^n \left(y_i - x_i^T\beta\right)^2, \tag{1}$$

the ordinary least squares (OLS) estimator is the linear unbiased estimator for $\beta^\star$ with the lowest possible variance. While it is tempting to use OLS for all regression problems, as mentioned above, in the high dimensional setting, there may not be a unique solution to Eq. (1). Thus the OLS estimator in high dimensional settings likely will not be interpretable.

To handle this problem of having potentially multiple solutions, given some $\lambda > 0$, an L1 regularization term can be added to Eq. (1) to construct a different estimator:

$$\hat{\beta}_{LASSO} \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - x_i^T \beta\right)^2 + \lambda \|\beta\|_1. \tag{2}$$

This is called the least absolute shrinkage and selection operator (LASSO). This method encourages sparsity in the estimated parameter $\hat{\beta}_{LASSO}$ with the λ parameter controlling the level of sparsity. This estimator has been shown to be consistent (i.e. $\hat{\beta}_{LASSO}$ converges in probability to $\beta^\star$ as n goes to infinity) under certain assumptions. However, this method is not robust to outliers where the response is unusually large or small compared to other observations with similar covariate vectors. We can see this from Eq. (2). In particular, the squared term means that this objective function penalizes large differences between the observed response y_i and fitted value $x_i^T \beta$ more so than it does smaller differences. As a result, a couple of outliers in the data set may cause the resulting estimated parameter to change drastically. This sensitivity to outliers is also reflected in one of the assumptions required for the LASSO to be consistent. Namely, one of these assumptions is that the distribution of ϵ is light-tailed, i.e., the tail of the error distribution does not decay slower than that of an exponential distribution. In other words, for the LASSO to be consistent, it is required that when the covariates of some observations are (close to) identical, there is not a substantial amount of variation among the corresponding responses.

In order to bypass this assumption of light-tailed error, which may not be reasonable in many cases in practice, a solution is to replace the squared loss with the absolute loss:

$$\hat{\beta}_{LAD} \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \left|y_i - x_i^T \beta\right|. \tag{3}$$

This least absolute deviance (LAD) estimator is associated with estimating the conditional median of the response y given some covariate x. Median, as an alternative measure of centre, is more robust to outliers than the mean. Therefore, we can expect $\hat{\beta}_{LAD}$ to be less sensitive to the outliers described above. Note that we can combine Eq. (3) with an L1 regularization term term in order to get a robust estimate of the regression parameters that is also sparse.

However, for the LAD estimator combined with L1 regularization to be a good substitute for the LASSO, we are implicitely asking the median of the error distribution to be close to the mean. In the case that the median of the error distribution is equal to the mean, we are essentially making the implicit assumption that the error distribution is symmetric. When this implicit assumption does not hold, using the LAD estimator in place of the OLS estimator will introduce some systematic bias to our estimated regression parameter. Therefore, it is desirable to develop a high dimensional mean regresison estimator that is consistent even when the underlying error distribution is neither symmetric nor light-tailed.

2 Proposed method

In the case where the error distribution is heavy-tailed and asymmettic, by using the absolute loss in place of the squared loss, we hope to obtain an regression estimator that is more robust to outliers at the cost of introducing some bias. The magnitude of this bias is dependent on the distribution of the error. However, note that the OLS estimator is unbiased for all error distributions with mean 0 and a finite variance. These observations motivate the idea of using a loss function that balances the unbiasedness of an OLS estimator and the robustness of an LAD estimator. One way of achieving this balance is through the use of the Huber loss with a blending parameter $\alpha \geq 0$:

$$l_\alpha(x) = \begin{cases} 2\alpha^{-1}|x| - \alpha^{-2} & \text{if } |x| > \alpha^{-1} \\ x^2 & \text{if } |x| \leq \alpha^{-1} \end{cases}. \tag{4}$$

In the Huber loss, α controls the blending between squared and absolute loss: $\alpha = 0$ corresponds to the squared loss and $\alpha = \infty$ corresponds to the absolute loss. Fan et al. (2017) proposes the penalized robust approximate (RA) quadratic loss using the Huber loss for high dimensional mean regression problems with an asymmetric and heavy-tailed error distribution. When L1 regularization is chosen to be the penalty term, for some $\alpha, \lambda \geq 0$, we arrive at the RA-lasso estimator

$$\hat{\beta}_{RA-lasso} = \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} l_\alpha \left(y_i - x_i^T \beta\right) + \lambda \|\beta\|_1.$$

By tuning the α parameter, we can trade off the balance between being unbiased and being robust to outliers. Note that the Huber loss Eq. (4) is convex and differentiable at $x = \alpha^{-1}$. Therefore, we can used gradient-based optimization to obtain the RA-lasso solution for some fixed α and λ.

2.1 Discussion on theoretical results

In this section, we provide a high level sketch of the theoretical guarantee on the quality of $\hat{\beta}_{RA-lasso}$ and offer connections to the motivation behind RA-lasso.

In Fan et al. (2017), the performance of $\hat{\beta}_{RA-lasso}$ is assessed through $\|\hat{\beta}_{RA-lasso} - \beta^\star\|_2$. More specifically, we can decompose this quantity as follows.

$$\|\hat{\beta}_{RA-lasso} - \beta^\star\|_2 \leq \|\beta_\alpha^\star - \beta^\star\|_2 + \|\hat{\beta}_{RA-lasso} - \beta_\alpha^\star\|_2,$$

where $\beta_\alpha^\star = \arg\min_{\beta \in \mathbb{R}^p} \mathbb{E}\left[l_\alpha(y - x^T\beta)\right]$. We call the first term in the above bound the approximation error and the second term the estimation error. Building on top of this, theorem 3 of Fan et al. (2017) states that, under the conditions of lemmas 1 and 3, for some $q \in (0, 1]$ and $k \geq 2$, with probability at least $1 - c_1 \exp(-c_2 n)$,

$$\|\hat{\beta}_{RA-lasso} - \beta^\star\|_2 \leq d_1 \alpha^{k-1} + d_2 \sqrt{R_q} \left(\frac{\log p}{n}\right)^{\frac{1}{2} - \frac{q}{4}}, \tag{5}$$

where the first term is a bound on the approximation error and the second term is a bound on the estimation error. The technical conditions and the selection of constants in full detail can be found in Section 2 of Fan et al. (2017). It is, however, worth pointing out that this above bound only applies to α and λ values satisfying the conditions in lemmas 1 and 3.

It is easy to see that, when the error distribution is symmetric (i.e. when the mean and median of the error distribution are euqal), $\beta_\alpha^\star = \beta^\star$ for all α, which implies that the approximation error is 0. When the error distribution is asymmetric, the bound on the approximation error can be thought of as the amount of bias introduced through the use of the Huber loss with blending parameter α. Note that this bound increases with α. Indeed, the larger we set α, the closer the Huber loss is to the absolute loss, which implies a larger bias induced by the RA-lasso estimator. This aligns with our intuition on the trade off between being unbiased and robust to outliers. The bound on the estimation error, on the other hand, is independent of α, and it converges to 0 as the number of observations n goes to infinity. By controlling the rate at which α increases, it is possible to have $\|\hat{\beta}_{RA-lasso} - \beta^\star\|_2$ converging to 0 at the rate of the bound on the estimation error from Eq. (5), which is the same as the optimal rate under the light tail situation Raskutti et al. (2011).

Before presenting the procedure for applying the RA-lasso estimator, we note that this estimator can be used as a mean estimator by treating it as a univariate linear regression problem where the covariate equals 1. Similarly, by noting that each element of the covariance matrix σ_{ij} for $i, j \in \{1, \cdots, p\}$ can be written as an expectation of the product between the ith and jth element of the centred covariate vector, the RA-lasso estimator can then also be used to estiamte the covariance matrix. Concentration inequalities for both of these cases are developed in Fan et al. (2017). Through this lens, Fan et al. (2017) also extended an mean estimator of heavy-tailed distributions developed in Catoni (2012) to the linear regression setting and developed a corresponding high probability bound similar to that of Eq. (5). However, these extensions are out of the scope of this report, and are thus not discussed in any more details here.

2.2 Practical implementation

While the bound from Eq. (5) aligns with our intuition on the trade off between unbiasedness and robustness against outliers, this result only holds for specific values of α and λ. In addition, the choice of α and λ values depend on universal constants that are typically not known in practice. As a result, Fan et al. (2017) suggests selecting α and λ by a two-dimensional grid search using cross-validation for a set of values that minimize the mean-squared error or an information-based criterion. More specifically, it is suggested that the search grid is formed by values that are equally spaced in log scale. Although not explicitely stated in the paper, it is recommended to form a search grid that contains candidate α and λ values ranging across multiple orders of magnitude. Algorithm 1 outlines the procedure for applying the RA-lasso estimator, where the hyperparameters α and λ are chosen from candidate values $(\alpha_i)_{i=1}^{C_\alpha}$ and $(\lambda_i)_{i=1}^{C_\lambda}$ using K-fold cross-validation with the mean-squared

loss. Note that we write

$$\forall j \in \{1, \cdots, p\}, \quad x^j = \begin{bmatrix} x_{1j} & \cdots & x_{nj} \end{bmatrix}^T \in \mathbb{R}^n, \quad \text{and} \quad y = \begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix}^T \in \mathbb{R}^n.$$

Note that since the RA-lasso objective is convex, we can use composite gradient descent to obtain the RA-lasso estimate (using optimization packages such as `CVXR` in `R` or `CVX` in `MATLAB`). Fan et al. (2017) shows that this algorithm, with high probability, has the same convergence rate as Eq. (5).

Algorithm 1 RA-lasso procedure

Require: $(x_i, y_i)_{i=1}^n$, $(\alpha_i)_{i=1}^{C_\alpha}$, $(\lambda_i)_{i=1}^{C_\lambda}$, K

1. standardize the response and each of the p predictors to have mean 0 and unit variance
   $\tilde{y} = (y - \texttt{mean}(y))/\texttt{sd}(y)$;
   $\forall j = 1, \cdots, p, \quad \tilde{x}^j = (x^j - \texttt{mean}(x^j))/\texttt{sd}(x^j)$
2. randomly split the index set $\mathcal{I} = \{1, \ldots, n\}$ into K (equally sized) disjoint sets $\mathcal{I}_1, \cdots, \mathcal{I}_K \subset \mathcal{I}$ such that $\mathcal{I}_1 \cup \cdots \cup \mathcal{I}_K = \mathcal{I}$
3. perform K-fold cross-validation
   `cv_loss = matrix(0,`C_α, C_λ`)`
   for `i in 1,...,`C_α **do**
   for `j in 1,...,`C_λ **do**
   for `k in 1,...,K` **do**
   $\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{n - |\mathcal{I}_k|} \sum_{m \notin \mathcal{I}_k} l_{\alpha_i}\left(\tilde{y}_m - \tilde{x}_m^T \beta\right) + \lambda_j \|\beta\|_1$
   `cv_loss[i,j]` $+= \sum_{m \in \mathcal{I}_k} \left(\tilde{y}_m - \tilde{x}_m^T \hat{\beta}\right)^2$
   end for
   end for
   end for
   $i^\star, j^\star =$ `which(cv_loss == min(cv_loss), arr.ind = TRUE)`
4. refit the model using all data with the selected hyperparameters
   $\tilde{\beta}_{RA-lasso} = \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^n l_{\alpha_{i^\star}}\left(\tilde{y}_i - \tilde{x}_i^T \beta\right) + \lambda_{j^\star} \|\beta\|_1$
5. destandardize the coefficients back to original scale (optional)
   $\hat{\beta}_{RA-lasso,0} = \texttt{mean}(y) - \sum_{j=1}^p \frac{\texttt{sd}(y)}{\texttt{sd}(x^j)} \texttt{mean}(x^j) \tilde{\beta}_{RA-lasso,j}$ ▷ intercept term
   $\forall j = 1, \ldots, p, \quad \hat{\beta}_{RA-lasso,j} = \frac{\texttt{sd}(y)}{\texttt{sd}(x^j)} \tilde{\beta}_{RA-lasso,j}$ ▷ slope terms

3 Discussion on simulation

In this section, we take a closer look at the simulation studies conducted in Fan et al. (2017). In particular, we focus on the simulation results from Table 2 in Fan et al. (2017) on a synthetic sparse, high dimensional homoscedastic model under various centred error distributions. The code used to generate the results can be found at `https://fan.princeton.edu/publications-software.html`. By homoscedastic, we mean that the error is independent of the observations and the true regression coefficients. With $n = 100$ and $p = 400$, the model takes on the following form:

$$y_i = x_i^T \beta^\star + \epsilon_i, \quad x_i \overset{\text{i.i.d.}}{\sim} N(0, I_p) \quad \forall i \in \{1, \ldots, n\}, \quad \text{and} \quad \beta^\star = \big[3, \ldots, 3, 0, \ldots, 0\big]^T,$$

where the first 20 are all equal to 3 and the rest are all equal to 0. Five different error distributions exhibiting all possible combinations of heavy/light-tailed and (a)symmetric error distributions are tested, including a light-tailed and symmetric error distribution being the baseline. The detailed settings of these error distributions can be found in Table 1 in Fan et al. (2017). We note that this setup can help isolate the change in performance caused by the tail behaviour of the error distribution and whether the error distribution is symmetric.

In this simulation study, RA-lasso is compared against two other estimators: the LASSO and R-lasso, which is the LAD estimator with L1 regularization. Recall that while LASSO is the go-to method for sprase high-dimensional mean regression, it may struggle with heavy-tailed errors. R-lasso addresses this issue with LASSO by replacing the squared loss in the objective with absolute loss. However, R-lasso imposes a strong implicit assumption that the error distribution is symmetric. By comparing these three methods, we can easily see how well RA-lasso handles the heavy-tailedness and/or asymmetricity of the error distributions compared to both the default method and a previous approach in the setting of sparse high-dimensional mean regression. On the other hand, the inclusion of the baseline case with a light-tail and symmetric error distribution allows us to check whether we lose any efficiency when the error distribution is indeed light-tailed and symmetric.

To quantitatively assess the performance of the estimators, the L2 and L1 errors between the true regression coefficients and the estimates are computed to indicate how close each regression estimate is to the true value. At the same time, the number of noise predictors that are selected (false positive) and the number of true predictors that are not selected (false negative) are also computed to assess the quality of the selected predictors. Finally, a measure of the relative gain of RA-lasso with respect to R-lasso and LASSO

$$\frac{\|\hat{\beta}_{\text{LASSO}} - \beta^\star\| - \|\hat{\beta}_{\text{oracle}} - \beta^\star\|}{\|\hat{\beta}_{\text{RA-lasso}} - \beta^\star\| - \|\hat{\beta}_{\text{oracle}} - \beta^\star\|}, \quad \frac{\|\hat{\beta}_{\text{R-lasso}} - \beta^\star\| - \|\hat{\beta}_{\text{oracle}} - \beta^\star\|}{\|\hat{\beta}_{\text{RA-lasso}} - \beta^\star\| - \|\hat{\beta}_{\text{oracle}} - \beta^\star\|},$$

in terms of both the L1 and L2 norms, are also computed. Note that $\hat{\beta}_{\text{oracle}}$ denotes the OLS estimator by using only the first 20 predictors. These relative gain values offer an intuitive comparion between RA-lasso and another method: a relative gain greater than 1 indicates that the RA-lasso estimate is closer to the true $\hat{\beta}$ than LASSO or R-lasso, and vice versa.

To select the optimal hyperparameters of each method, 100 independently generated validation sets are used. More specifically, a grid search is performed to find the hyperparameters that minimize the mean L2-loss between the true regression coefficients and the estimates. Each of the regression estimates are then constructed using the corresponding optimal hyperparameters. The error metrics with respect to these regression estimates are then computed. This procedure is repeated 100 times, and Table 2 in Fan et al. (2017) shows the mean of each error metric for a given error distribution and regression method.

3.1 Relevance and practicality of the simulation study

Although this simulation study covers a wide range of symmetric and asymmetric error distributions with different tail behaviours, some of the procedures appear to be arbitrarily chosen and impractical. Furthermore, the simulation studies does not seem to reflect the theoretical convergence guarantees discussed in the paper. Here we provide a detailed discussion.

In this simulation study, since the underlying model and the true error distribution is known, it is feasible to tune the hyperparameters using independently generated validation sets. Furthermore, the hyperparameters are chosen to be the ones that minimize the mean L2-loss between the true regression coefficients and the estimates. However, in practice, we typically do not have access to independent validation sets, let alone the true regression parameters. As a result, it is perhaps more reasonable to tune the hyperparameters of each regression methods using cross-validation procedure instead. And instead of using the mean L2-loss between the true regression coefficients and the estimates, the mean-squared error between the true and fitted responses may be used instead. In fact, this is what is recommended at the end of Section 2 in Fan et al. (2017). Note that the mean-squared error is appropriate here even under heavy-tailed and potentially asymmetric errors, since the goal here is to perform high dimensional mean regression.

It is also worth noting that the paper claims that by running the composite gradient descent algorithm, the RA-lasso estimator converges to the true regression coefficient as the number of observations goes to infinity. However, all datasets used in the simulation studies have a fixed size. In other words, none of the experiments demonstrate the claimed convergence rates of the RA-lasso algorithm. It would have been nice to include plots showing how $\|\hat{\beta}_{RA-lasso} - \beta^\star\|_2$ decays as we increase the number of observations in the dataset while keeping everything else fixed.

Finally, we note that only the mean values of the error metrics (L1- and L2-loss, FP and FN counts, as well as relative gains) are reported in Table 2 of Fan et al. (2017). While these values allow us to compare the average performance across all regression estimators and error distributions, there is no quantification of uncertainty. Without knowing the magnitudes of the variances of these measurements, it is difficult to decide whether one method outperforms another conclusively.

		Lasso	5% / 95% pc	RA Lasso	5% / 95% pc
N(0,4)	L2-loss	4.30	(2.70, 6.91)	4.58	(2.70, 7.76)
	L1-loss	28.15	(17.32, 45.93)	29.81	(17.17, 50.34)
	FP	67.53	(58.90, 74.10)	64.01	(55.95, 71.00)
	FN	0.07	(0,0)	0.10	(0,1)
	RG-L2			0.95	(0.74, 1.08)
	RG-L1			0.93	(0.73, 1.08)
2t3	L2-loss	5.08	(2.59, 9.18)	5.10	(2.65, 9.52)
	L1-loss	33.84	(16.20, 61.74)	33.21	(16.24, 62.74)
	FP	69.05	(60.00, 77.05)	63.29	(55.95, 71.00)
	FN	0.15	(0, 1)	0.25	(0,2)
	RG-L2			1.06	(0.80, 1.62)
	RG-L1			1.05	(0.77, 1.58)
MixN	L2-loss	8.86	(6.86, 11.31)	8.94	(6.76, 11.20)
	L1-loss	54.78	(40.23, 71.51)	54.34	(40.61, 70.38)
	FP	54.34	(47, 62)	52.41	(45.95, 60.00)
	FN	1.59	(0, 5)	1.79	(0.00, 5.05)
	RG-L2			1.01	(0.92, 1.09)
	RG-L1			0.99	(0.90, 1.07)

Table 1: Simulation results for lasso and RA lasso under homoscedastic model

4 Reproduced and revised simulation studies

In this section, we begin by reproducing a subset of the simulation studies in Table 2 of Fan et al. (2017). We also address some of the practical concerns mentioned in Section 3.1 and incorporate them to a revised simulation study with an expanded set of error distributions and regression estimators. All of the code used to conduct the simulations can be found at `https://github.com/NaitongChen/QP-2`.

Table 1 shows the mean error metrics for three of the error distributions from Table 2 of Fan et al. (2017) using LASSO and RA-lasso. Here we follow the setup in the original code of the paper as much as we can. Specifically, to perform hyperparameter tuning, a $2d$ grid of 15 equally spaced α values ranging from 0.0001 to 1 and 30 equally spaced λ values ranging from 0.0001 to 1.5 are used. Since the sequence of λ values used to tune LASSO is not provided by the authors, a sequence of 200 values ranging from 0.00001 to 10 is used. This sequence is chosen because the number of candidate hyperparameter values is on the same order of magnitude of the total sets of hyperparameters tested for RA-lasso. At the same time, the spacing between the candidate λ values for LASSO and RA-lasso are both kept to be around 0.05. Note that to solve the (convex) optimization problem in order to get the RA-lasso estimates, the `CVXR` package in `R` is used. This is an equivalent package of `CVX` in `MATLAB`, which is used in the code to generate the results in the paper. The `glmnet` package is used to obtain the LASSO estimates. In addition to the mean values, the corresponding 5^{th} and 95^{th} percentiles as a measure of uncertainty are also reported. We note that, similar to Table 2 in Fan et al. (2017), LASSO and RA-lasso perform similarly across all three scenarios.

		Lasso	5% / 95% pc	RA Lasso	5% / 95% pc	PENSE	5% / 95% pc
N(0,4)	L2-loss	5.8	(4.42, 7.63)	5.26	(3.31, 8.89)	10.12	(4.05, 12.97)
	L1-loss	57.17	(30.56, 82.48)	33.22	(22.19, 63.80)	64.18	(30.12, 80.14)
	FP	259.1	(28.00, 379.05)	60.85	(29, 81)	69.1	(64.00, 77.05)
	FN	0.05	(0.00, 0.05)	0.1	(0, 1)	5	(0.0, 11.1)
	RG-L2			2.14	(0.89, 4.01)	1.06	(0.45, 2.61)
	RG-L1			1.3	(0.62, 2.01)	0.64	(0.31, 1.40)
t2	L2-loss	5.98	(4.65, 7.37)	5	(2.47, 8.48)	8.8	(2.72, 12.57)
	L1-loss	57.07	(30.23, 82.01)	31.76	(17.40, 59.04)	58.02	(19.53, 80.68)
	FP	255.95	(43.15, 379.05)	61.85	(42.05, 80.00)	73.85	(65.7, 79.0)
	FN	0.25	(0.00, 0.25)	0.25	(0.00, 1.15)	3.35	(0.0, 9.1)
	RG-L2			2.36	(0.83, 4.42)	1.34	(0.42, 4.02)
	RG-L1			1.53	(0.71, 3.02)	0.9	(0.36, 2.44)
MixN	L2-loss	9.44	(7.12, 12.24)	10.58	(7.54, 13.35)	12.66	(11.54, 13.54)
	L1-loss	67.39	(46.14, 93.34)	61.08	(45.98, 85.75)	77.54	(60.58, 91.52)
	FP	140.25	(3, 379)	42.35	(6.75, 80.10)	54.25	(1.00, 81.15)
	FN	3.55	(0, 13)	5.3	(0.0, 13.3)	10.2	(4, 20)
	RG-L2			1.12	(0.84, 1.74)	0.86	(0.48, 1.16)
	RG-L1			0.87	(0.62, 1.07)	0.69	(0.46, 0.98)
Lev1	L2-loss	4.19	(1.65, 6.75)	4.53	(1.63, 12.48)	7.73	(2.10, 13.47)
	L1-loss	36.35	(11.09, 75.37)	27.76	(10.9, 61.6)	49.23	(13.15, 77.11)
	FP	174.4	(61.20, 378.05)	60.8	(20.65, 79.05)	65.45	(17.15, 74.20)
	FN	0.7	(0.0, 0.7)	1.4	(0.00, 12.15)	3.3	(0.00, 17.15)
	RG-L2			1.56	(0.61, 3.52)	0.84	(0.19, 2.03)
	RG-L1			1.14	(0.43, 1.96)	0.59	(0.20, 1.21)
Lev2	L2-loss	5.3	(2.99, 6.73)	2.92	(1.44, 7.07)	7.9	(2.03, 13.52)
	L1-loss	58.46	(21.25, 78.36)	19.24	(9.47, 38.83)	52.35	(14.40, 88.31)
	FP	329.75	(77.60, 379.05)	67.95	(51.00, 79.05)	74.65	(67.7, 80.0)
	FN	0	(0, 0)	0	(0, 0)	3	(0.00, 11.05)
	RG-L2			4.61	(0.92, 10.29)	1.63	(0.50, 4.58)
	RG-L1			2.88	(0.86, 6.56)	0.99	(0.34, 2.81)

Table 2: Additonal simulation results for lasso, RA lasso, and PENSE

We now revise the simulation study to address some of the issues discussed in Section 3.1. Firstly, in addition to LASSO and RA-lasso, we also add another robust regression estimator PENSE, as a competitor to RA-lasso, to the comparison (Freue et al., 2019). Furthermore, instead of conducting hyperparameter tuning using independent validation sets and the L2-loss between the true regression parameters and the estimates, we use a 5-fold cross validation with the mean-squared error between the true and fitted responses.

In terms of the sequence of hyperparameters, instead of setting the range of α and λ values somewhat arbitrarily (0.0001 to 1 and 0.0001 to 1.5, respectively), as is done in the authors' original code, we follow the guideline in Section 2 of Section 3.1 and use a sequence of hyperparameters that are equidistant in log scale. Specifically, 15 λ values ranging from e^{-15} to e^2 are used for LASSO and PENSE, and the same sequence of α and λ values ranging from e^{-10} to e^2 are used for RA-lasso. These values are chosen so that the minimum value in each sequence is as small as possible without causing numerical issues. This is a typical approach given any problem in practice: by setting the sequence to be eqidistant in log scale, we can explore hyperparameters in a wide range of orders of magnitude.

We also expand the simulation study by including two scenarios with various degrees of high leverage observations (unusual covariates) to check the robusteness of RA-lasso. We follow the procedure in Maronna (2011) and contaminate 10% of the observations in each dataset. Specifically, we replace the predictors x_i with

$$\tilde{x}_i = \eta_i + \frac{k_{\text{lev}}}{\sqrt{a^T a}} a, \tag{6}$$

where $\eta_i \sim N(0, 0.1^2 I_p)$ and $a = \tilde{a} - \frac{1}{p}\tilde{a}^T \mathbf{1}_p$ with the entries of $\tilde{a}$ following Unif($\{-1, 1\}$). The error distributions in these cases are set to $N(0, 1)$. We also change the error distribution in the second scenario from $2t_3$ to t_2 in order to make the heavy-tailedness more pronounced. Finally, instead of centring the error distributions by their empirical average, we centre them using their analytical mean.

Table 2 shows the mean error metrics and their 5^{th} and 95^{th} percentiles for each scenario of the revised simulation study. Note that `lev1` and `lev2` in the table correspond to $k_{\text{lev}} = 1$ and $k_{\text{lev}} = 2$ in Eq. (6), respectively. Also note that due to computational constraints, these values are based on 20 instead of 100 trials.

5 Conclusion

References

Olivier Catoni. Challenging the empirical mean and empirical variance: a deviation study. In *Annales de l'IHP Probabilités et statistiques*, volume 48, pages 1148–1185, 2012.

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(1):247–265, 2017.

Gabriela V Cohen Freue, David Kepplinger, Matías Salibián-Barrera, and Ezequiel Smucler. Robust elastic net estimators for variable selection and identification of proteomic biomarkers. *The Annals of Applied Statistics*, 13(4):2065–2090, 2019.

Ricardo A Maronna. Robust ridge regression for high-dimensional data. *Technometrics*, 53 (1):44–53, 2011.

Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Minimax rates of estimation for high-dimensional linear regression over ℓ_q-balls. *IEEE transactions on information theory*, 57 (10):6976–6994, 2011.