Journal of Statistics & Applied Sciences, Issue -10
ISSN 2718-0999

Research Article

Received: date:08.08.2024
Accepted: date:16.12.2024
Published: date:31.12.2024

A Robust Initial Basic Subset Selection Method for Outlier Detection Algorithms in Linear Regression

Mehmet Hakan Satman[1*]

[1]İstanbul University, Department of Econometrics, İstanbul Türkiye; mhsatman@istanbul.edu.tr
Orcid:0000-0002-9402-1982[1]
*Correspondence: mhsatman@istanbul.edu.tr

Abstract: The main motivation of this study is to develop an efficient algorithm for diagnosing and detecting outliers in linear regression up to a reasonable level of contamination. The algorithm initially obtains a robust version of the hat matrix at the linear algebra level. The basic subset obtained in the first stage is improved through concentration steps as defined in the fast-LTS (Least Trimmed Squares) regression algorithm. The method can be plugged into another algorithm as a basic subset selection state. The algorithm is effective against outliers in both X and Y directions by a rate of 25%. The complexity of the algorithm increases linearly with the number of observations and parameters. The algorithm is quite fast as it does not require iterative calculations. The success of the algorithm against a specific contamination level is demonstrated through simulations.
Keywords: Linear regression, Outlier Detection, Robust Regression

1. Introduction

Suppose the linear regression model is

$$y = X\beta + \varepsilon$$

where y is the n-vector of the response variable, X is the $n \times p$ design matrix, ε is the n-vector of the stochastic error term with zero mean and constant variance, n is the number of observations, and p is the number of regression parameters. The ordinary least squares estimate of β, noted as $\hat{\beta}$,

$$\hat{\beta} = (X'X)^{-1}X'y$$

is the most efficient estimator among the linear and unbiased estimators, that is, $\hat{\beta}$ minimizes

$$MSE(\hat{\beta}) = Var(\hat{\beta}) + Bias^2(\hat{\beta}). \quad (1)$$

When data contains unusual observations Eq. 1 may increase drastically depending on the direction and level of the contamination. The LAD (Least Absolute Deviations) estimator is resistant up to 50% of contamination on the vertical outliers since it minimizes the conditional median of the response variable, that is, the LAD estimator $\hat{\beta}_{LAD}$ minimizes the objective function

$$min \sum | y - X\hat{\beta}_{LAD} |$$

which in turn can be written as a linear programming problem

$$min \sum_{i=1}^{n} e_i^- + e_i^+$$

Subject to:

$$X_{(i,)}\beta_{LAD} + e_i^- - e_i^+ = y_i \quad (2)$$
$$e_i^-, e_i^+ \geq 0 \quad , i = 1, 2, \ldots, n$$
$$\beta_{LAD} \in \mathbb{R}^p$$

Citation : M. H. Satman, "A Robust Initial Basic Subset Selection Method for Outlier Detection Algorithms in Linear Regression," Journal of Statistics and Applied Sciences, no. 10, pp. 76–85, Dec. 2024, doi:10.52693/jsas.1512794

where $X_{(i,)}$ is the ith row of X, $e_i^- \geq 0$ if i-th residual is located under the regression object, otherwise $e_i^- = 0$, $e_i^+ \geq 0$ if i-th residual is located above the regression object, otherwise $e_i^+ = 0$. If $e_i^- = e_i^+ = 0$ then the regression equation exactly fits the i-th observation. Eq. 2 has an exact solution to the robust method LAD, however, LAD is not resistant to X-space outliers, namely, bad leverage points [1].
LTS (Least Trimmed Squares) estimator $\hat{\beta}_{LTS}$ is resistant to both vertical and X-space outliers up to 50% contamination and it is defined as

$$min \sum_{i=1}^{h} | y_i - X_{(i,)}\hat{\beta}_{LTS}|^2$$

where $| y_i - X_{(i,)}\hat{\beta}_{LTS}|^2$ is the i-th ordered squared residual, h is at least half of the number of observations n. The Fast-LTS algorithm requires selecting many basic subsets and then iterating and enlarging the basic subset until some convergence criteria is met [2]. Since the algorithm is based on comprehensive iterations of calculations, more computation time is required to obtain parameter estimates. When the data is large, metaheuristics [18] can be used to minimize the objective function, however, the required computation time is still a problem just because the optimization problem has not a closed form as in many robust regression estimators.
In this paper, a robust version of vector and matrix algebra is defined to obtain a more robust version of the well-known hat matrix. This hat matrix is then used to construct a small basic subset which is supposed to be free of outliers. This basic subset is then fed into a robust regression estimator to obtain robust regression parameter estimates. The proposed method is based on a single-pass algorithm and fast. It's shown that the proposed method is resistant up to 25% level of contamination.
In Section 1 the problem is introduced. In Section 2, the basics of the robust hat matrix construction is introduced. In Section 3, the use of the robust hat matrix based initial basic subset is introduced in a robust regression estimator, LTS. In Section 4, a suite of simulation studies is performed to assess the performance of the algorithm for varying dimensions and levels of contamination. Finally, in Section 5, the results are discussed and we conclude.

2. Preliminaries

Suppose that the p-vectors x and y are defined as

$$x = \left[x_1, x_2, \ldots, x_p\right]^T$$

and

$$y = \left[y_1, y_2, \ldots, y_p\right]^T$$

The classical dot product sums up the products of corresponding elements of vectors x and y, that is,

$$x'y = \sum_{i=1}^{p} x_i y_i$$

The dot product $x'y$ can also be written as

$$x'y = f(x_1y_1, x_2y_2, \ldots, x_py_p) \times p$$

where

$$f(x,y) = \sum_{i=1}^{p} (x_i\, y_i) \,/\, p.$$

In other terms, the sum is p times the sample mean. Now suppose that f^* is a robust location estimator. Then the robust dot product of x and y can be written as

$$x' \otimes y = f^*(x_1y_1, x_2y_2, \ldots, x_py_p) \times p. \quad (3)$$

Since $f^*(.)$ is a robust location estimator then $f^*(.) \times p$ is a robust estimate of sum. Using $x' \otimes y$ instead of $x'y$ results in a more robust version of the vector product.

2.1. Robust Matrix Multiplication

Suppose that X and y are $m \times n$ and $n \times p$ matrices, respectively. Then the i-th row and the j-th column of the robust matrix multiplication of $C = A \otimes B$ is defined as

$$c_{ij} = X_{(i,)} \otimes Y_{(,j)} \quad (4)$$

where $X_{(i,)}$ is the i-th row vector of X and $y_{(,j)}$ is the j-th column vector of y, $\otimes$ is the multiplication operator as defined in Equation 3, and f^* is a univariate robust location estimator.

2.2. Robust Hat Matrix

The original hat matrix of linear regression is based on the design matrix and it is defined as

$$H = X(X'X)^{-1}X'$$

and the diagonal elements of H are investigated for bad leverage points [3]. When the design matrix contains unusual observations, because of the summation operator, the corresponding diagonal elements of H tend to have smaller values and cause the masking effect. Replacing the matrix multiplication operator with its robust counterpart yields

$$H_R = X(X' \otimes X)^{-1}X'$$

where $\otimes$ is the robust multiplication operator. Since the part $(X' \otimes X)$ is supposed to be outlier free for some contamination level, it is expected that the unusual observations will have relatively larger values on the corresponding diagonal elements of H_R.

2.3. Trimean as a Robust Location Estimator

Trimean estimator is a robust estimator of the location parameter which is defined as

$$Trimean(x) = \frac{Q_{25}(x) + 2Q_{50}(x) + Q_{75}(x)}{4}$$

where $Q_{25}(x)$ is the first quartile of x, $Q_{50}(x)$ is the sample median of x, and $Q_{75}(x)$ is the third quartile of x, respectively [4]. The Trimean is less robust than the sample median, however, it's more efficient. The break-down point of Trimean is 25%, that is, it stays resistant to outliers when the fraction of contamination is up to 25% of data. Moreover, Trimean is a suitable estimator due to its efficiency resembling the arithmetic mean and its robustness akin to the median.

In Table 1, the use of the robust version of the Hat matrix (H_R) is represented using a set of simulated data. The robust location estimator is selected as $f^* = Trimean$. The data consist of two variables, namely x_1 and x_2, with the sample size of n = 10. The last two observations have unusual values. The 9th and 10th observations are manipulated by distorting the values corresponding to the first variable. $Diag(H)$ and $Diag(H_R)$ represent the diagonal elements of the hat matrix and robust hat matrix, respectively.

Table 1. Illustrative example of a set of simulated data.

Obs.	Cons.	x_1	x_2	$Diag(H)$	$Diag(H_R)$
1	1.0	0.493	0.520	0.319	0.649
2	1.0	0.332	0.534	0.307	0.648
3	1.0	0.895	0.797	0.121	0.236
4	1.0	0.910	0.545	0.258	1.023
5	1.0	0.779	0.967	0.345	0.871

6	1.0	0.201	0.765	0.164	0.867
7	1.0	0.622	0.760	0.122	0.130
8	1.0	0.311	0.960	0.374	1.769
9	1.0	2.900	0.756	0.283	14.42
10	1.0	4.300	0.764	0.702	38.24

In Table 1, it is shown that the two largest diagonal elements incorrectly belong to the observations 8 and 10, whereas, observations 9 and 10 have relatively large values on the diagonal elements of the Trimean based robust hat matrix. As it is expected, larger and smaller values are omitted in the summation part of the vector multiplication stage.

3. The Main Algorithm

Constructing a basic subset and then iterating and enlarging this subset until some convergence criteria is met is not a new concept in detecting outliers in linear regression [5, 6, 7]. [5] constructs a subset of p+1 observations and then enlarges these subsets using the DFFITS [8] statistics. Similarly [6] and [7] enlarge the basic subset using multivariate statistics calculated on the design matrix in order to detect bad-leverage points. [9] and [10] construct and iterate an initial estimate and then update the weights of an iteratively weighted least squares estimate. [11] introduces a two-stage method for detecting outliers in linear regression. In the first stage of the method, a subset of outlier-free observations is created using a robust covariance matrix estimation inspired by the Comediance [12] statistic. Several iterations are performed for constructing initial subsets as the essential steps of the LMS [13], LTS [2], and LTA [14] estimators. However, the initially selected subsets may contain outliers, and in the later stages of the outlier detection algorithm (or the robust estimator), this could lead to clean observations being reported as outliers and outliers being reported as clean observations. The developed algorithm has been designed to find a better initial subset in a shorter time and to reduce the computational overhead in subsequent iterations of the detection algorithms

The steps of the devised initial basic subset construction algorithm is given as follows:

Main Algorithm:

Step 0. Construct the design matrix X with dimensions n and p.

Step 1. Compute the robust hat matrix using the formula $X(X' \otimes X)^{-1}X'$ where the operator $\otimes$ is defined as in Eq. 3 and Eq. 4, respectively.

Step 2. Let $|h_{ii}|$ is the ordered diagonal elements of the robust hat matrix. Record the indices of first $p+1$ elements of $|h_{ii}|$.

Step 3. Perform C-Steps of the LTS algorithm for the initial basic subset obtained in the previous step. Report $\hat{\beta}_{LTS}$ and the other necessary output of the final regression.

The main algorithm is basically based on constructing an initial basic subset of length $p+1$ which is supposed to be free of outliers. The final step is based on the concentration steps (C-Steps) of the LTS algorithm but only a single initial basic subset is fed for obtaining a better larger subset with size h. The algorithm differs from the LTS algorithm as it constructs a single basic subset instead of selecting many random basic subsets.

Iterating C-Steps is the inner part of the outlier detection algorithm. C-Steps, as defined in [2], start with estimating the linear regression coefficients using a $p+1$ subset of observations. The smallest h ordered absolute residuals obtained in a previous step are then used to estimate the regression parameters. h is a number that covers at least half of the observations. Subsequent iterations are performed until the difference of the last two objective function values are less than a small value. The Fast-LTS algorithm iterates these steps for many randomly drawn $p+1$ subsets. The devised algorithm is fast as it uses a single initial basic subset which is obtained using the robust hat matrix.

Table 2 represents regression coefficients estimated using the Hawkins, Bradu, Kass's (HBK) Artificial Data [15] by several estimators. HBK data consists of three independent variables and has 75 observations. The model is supposed to be $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \varepsilon$. LMS, LTS, BCH [7], and the devised algorithm R_{HAT} have similar results as $\widehat{\beta_0} < 0$, $\widehat{\beta_1} > 0$, $\widehat{\beta_2} > 0$, and $\widehat{\beta_3} < 0$. OLS and LAD end up with a different set of signs of estimations. This set of results is a clean indicator of the dataset

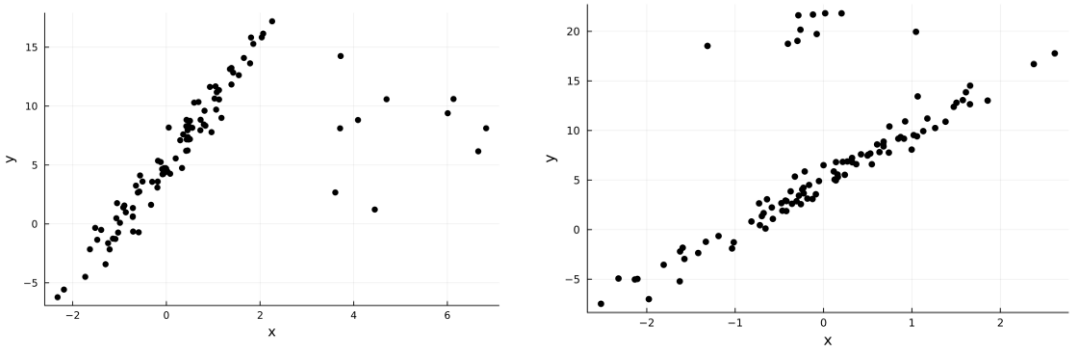
at hand has possible X-space outliers. The last column of Table 2 represents the relative time consumed by the estimators. The fastest estimator is OLS and its relative time is set to 1x. It can be said that the exact LAD estimation is 12.21 times slower than the OLS. The devised estimator is the fastest one among others as it is 1.52 times slower than the OLS estimator. Summarizing the results, it can be said that the devised algorithm is fast enough and it is capable of handling contaminated data with similar results obtained with the high breakdown estimators.

Table 2. Estimated regression coefficients using the Hawkins, Bradu, Kass's Artificial Data by several estimators.

Estimator	$\widehat{\beta_0}$	$\widehat{\beta_1}$	$\widehat{\beta_2}$	$\widehat{\beta_3}$	Time
OLS	-0.387550	0.239185	-0.334548	0.383341	1
LAD	-0.881474	0.091312	0.154760	0.214647	12.21
LMS	-0.677950	0.266808	0.112131	-0.144488	91.47
LTS	-0.637726	0.253676	0.108266	-0.139344	62.13
BCH	-0.535321	0.227364	0.049063	-0.097577	10.16
R_{HAT}	-0.578735	0.261031	0.045958	-0.103124	1.52

4. The Simulation Study

A set of simulation studies is performed for measuring the performance of the devised algorithm. In each single simulation, a random regression data is created using a specific data generating process. The regression model has always an intercept term. x_i variables are drawn from a standard Normal distribution for $i = 1,2,\ldots,p$. The error term also follows a Normal distribution with zero mean and unit variance. Regression parameters are set to 5 for all β_i. Generated data is contaminated either in X or y directions. Contamination with a given level is always guaranteed by adding random noise to largest values on the corresponding set of variables. Simulations are repeated for null contamination level, that is, results contain the case of absence of outliers. LinRegOutliers [19] package of the Julia programming language [20] is used for the calculations. Figure 1 represents random regression data with contamination on either X-space (a) or vertical (b) direction. Note that the represented data has only a single independent variable, however, simulations are always performed for p > 2. In Figure 1 (a) 10 observations are located in a manner that is considered inappropriate as they don't follow the same structure with the remaining data points by considering the X-space. In Figure 1 (b), 10 observations are vertically incompatible with the regression structure and they have values larger than at least the maximum of the clean data points in vertical direction.



(**a**) (**b**)

Figure 1. Simulation data with n = 100, p = 2, and contamination level of 10% in either X or y directions, respectively. (a) represents the case of the contamination on X, (b) represents the case of the contamination on the response variable, y.

Figures 2 - 7 represent the simulation results. The horizontal axis shows the level of contamination, that is, when the contamination is 0%, the data is free of outliers, otherwise, data is contaminated by the ratio of the given level. In the vertical axis, average of the parameter estimates is given, that is, an unbiased

and robust estimator should have values around a constant line of y = 5. Moreover, if the fluctuation around the constant is low, then it can be concluded that the estimator has a smaller variance. The ideal graphics follow the $\hat{\beta} = 5$ line with minimum variance.

Figures 2 and Figure 3 represent the simulation results for n = 50 and p = 3 when the direction of contamination is X and y, respectively. It is shown that the parameter estimates are very close to 5 which is the true regression parameters given in the data generating process. When the level of contamination increases, parameter estimates tend to have larger variances and a negative bias emerges in case of presence of X outliers. Bias of the estimates remain near to zero in the presence of y-outliers. In both Figure 2 and Figure 3, it is shown that the parameter estimates drastically disrupt when the level of contamination is larger than 25%.

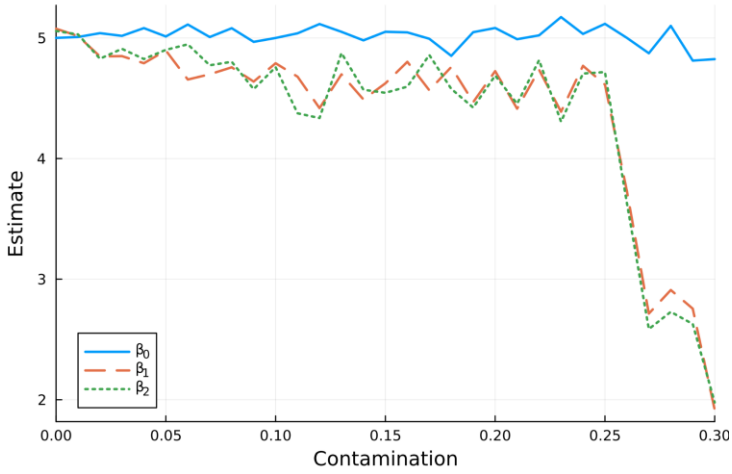


Figure 2. Parameter estimations for n = 50, p = 3, and contamination in X-direction

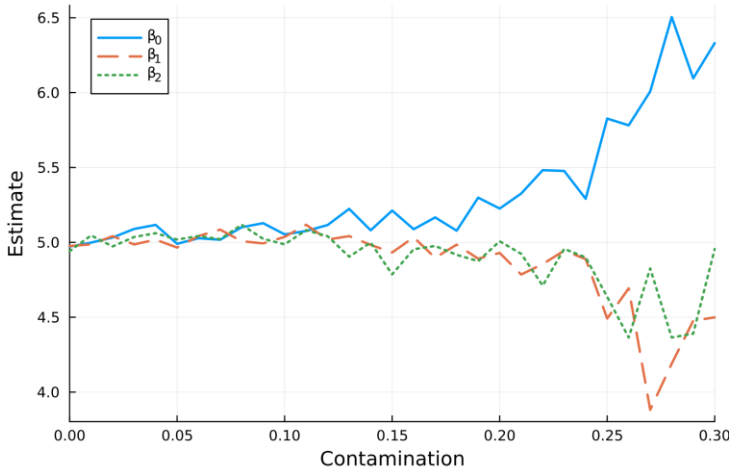


Figure 3. Parameter estimations for n = 50, p = 3, and contamination in y-direction

Figures 4 and Figure 5 represent the simulation results for n = 100 and p = 5. The results are similar when they are compared to the previous ones for n = 50 and p = 3, however, in presence of X-outliers, the bias and variance seem to be more stable when the level of contamination is $c \leq 0.25$. The intercept parameter estimate remains unbiased with a small increase in the variance. When the data is contaminated in the vertical direction, the results obtained by the configurations n = 50 and n = 100 are similar.

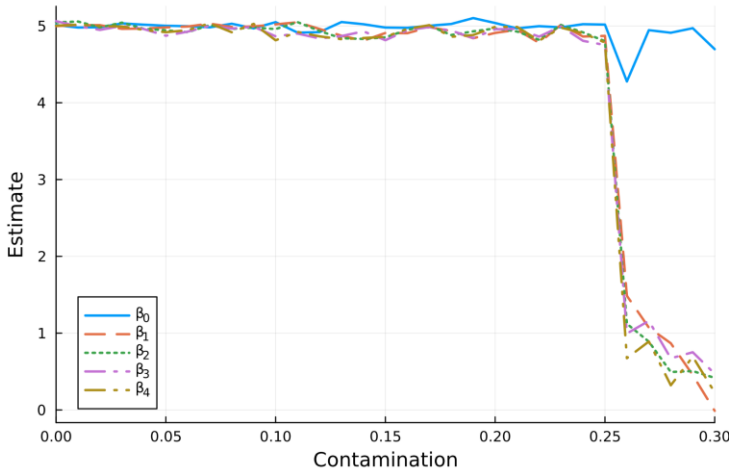


Figure 4. Parameter estimations for n = 100, p = 5, and contamination in X-direction

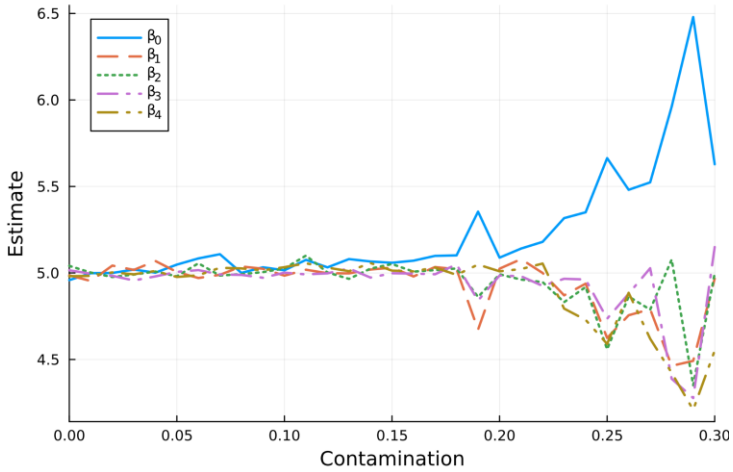


Figure 5. Parameter estimations for n = 100, p = 5, and contamination in y-direction

Figure 6 and Figure 7 represent the simulation results for n = 500 and p = 10. It is shown that the parameter estimates still remain stable for the contamination level is $c \leq 0.25$. As expected, when the contamination level is larger than $c = 0.25$, the estimates drastically disrupt. Differently, in the presence of y-outliers as it is shown in Figure 7, all of the parameter estimates take a value between a narrower range, e.g. $4.96 \leq \widehat{\beta}_i \leq 5.06$, and the range still remains stable when the contamination level increases.

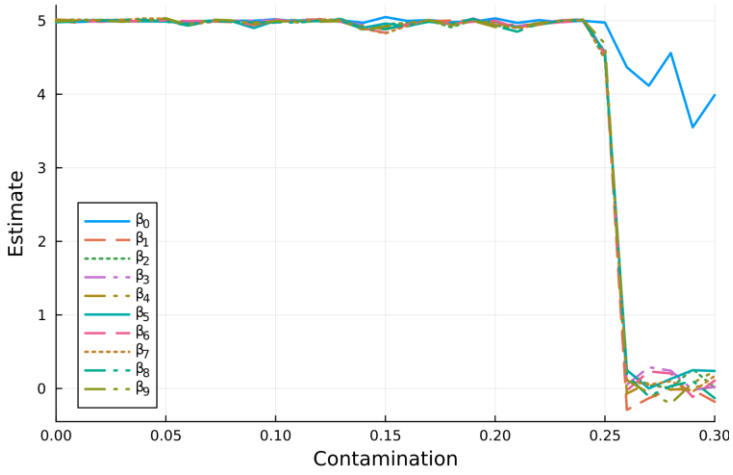


Figure 6. Parameter estimations for n = 500, p = 10, and contamination in X-direction

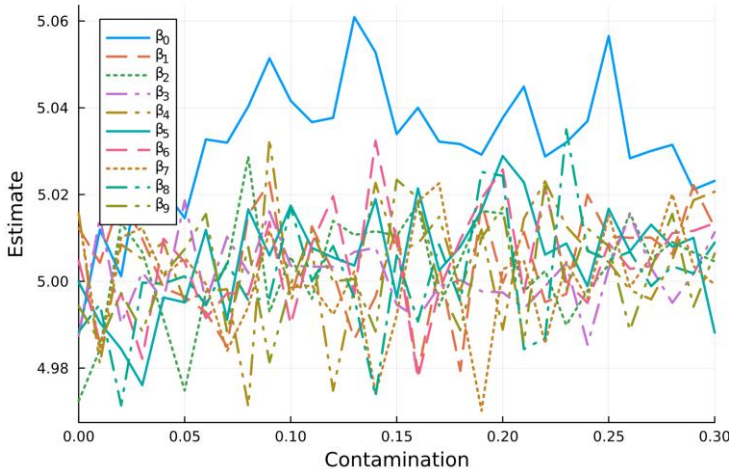


Figure 7. Parameter estimations for n = 500, p = 10, and contamination in y-direction

To investigate the performance of the algorithm, we generate simulated datasets with n = 1000 and p = 50. This configuration can be considered large for many research applications. Figure 8 and Figure 9 represent the simulation results for n = 1000 and p = 10. Simulation results have shown that even as the number of observations and parameters increase, distortions in the parameters begin at the point where the contamination level reaches 25%. Similar to previous simulation designs, in the presence of y-outliers, as shown in Figure 9, all parameter estimates fall within a narrower range, e.g., 4.96 to 5.06. This range remains stable even as the contamination level increases.

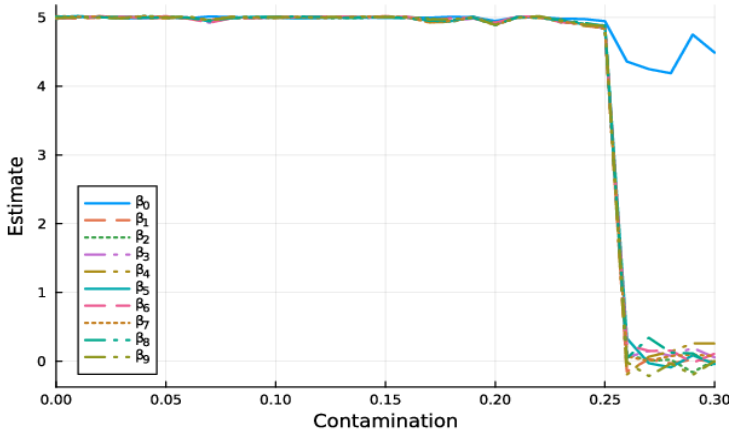


Figure 8. Parameter estimations for n = 1000, p = 10, and contamination in X-direction

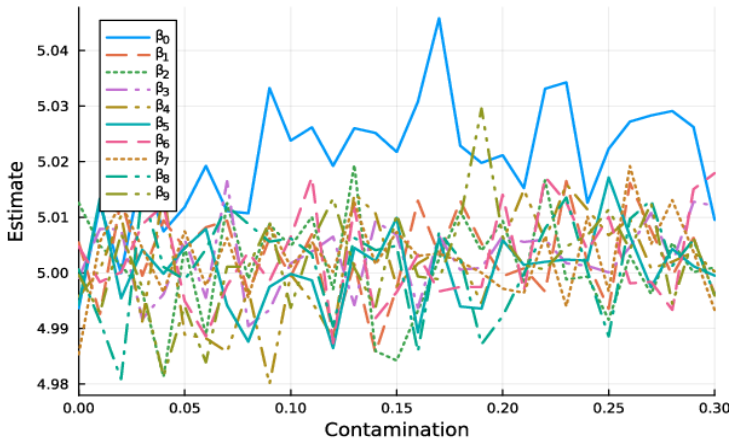


Figure 9. Parameter estimations for n = 1000, p = 10, and contamination in y-direction

5. Discussions and Conclusions

The robust statistics literature has a vast set of outlier detection and robust regression estimators including LMS, LTS, LTA, LAD, M-Estimators [16], S-Estimators [17], etc. Some of the estimators serve as high-breakdown estimators for datasets with a contamination level up to 50%. However, these methods and estimators are computationally heavy and they require more time to obtain a result.

The robust hat matrix based initial basic subset selection method is a single-pass and linear algebraic method that provides a clean input which is free of bad leverage points (X-space outliers) when the data is contaminated up to 25%. That property makes the devised method superior in datasets with a low contamination level. Note that the initial basic subset construction should be seen as an inner step of a detection algorithm and theoretical properties of the robust hat matrix are not investigated in this paper. Hence, it is not a standalone estimator but a subset selection tool only.

A suite of simulation study is performed to measure the performance of the devised method. Simulation data is generated for different levels of number of observations, parameters and several levels of contamination in both directions. The simulation results show that the method is applicable in all of the cases. The theoretical and known drawback is the maximum level of contamination. The simulation results also show that the bias and variance, hence MSE, of estimators reduce (towards zero) when the number of observations and parameters increases.

The robust hat matrix estimation is based on the Trimean estimator. The maximum breakdown is determined by this estimator. Using the sample median instead of the Trimean causes ending up with a singular matrix which prevents obtaining a matrix inverse. Selection of another robust location measure can be investigated in future works.

This paper is primarily motivated by the development of a clean starting set of points for later use in the Fast-LTS algorithm. However, this stage can be easily integrated into any other robust regression method and/or outlier detection methods. Future studies could investigate how the initial subset selection stage can be applied to other outlier detection methods. This would provide insights into whether the proposed approach can enhance the performance and accuracy of alternative algorithms. Additionally, its integration into robust regression methods could reveal potential improvements in handling datasets with challenging characteristics.

Funding: This research did not receive any external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Acknowledgments: The author thanks the referees for their valuable suggestions which made this paper much improved.

References

[1] X. Gao and Y. Feng, "Penalized weighted least absolute deviation regression," Statistics and its interface, vol. 11, no. 1, pp. 79–89, 2018.

[2] P. J. Rousseeuw and K. Van Driessen, "Computing LTS regression for large data sets," Data mining and knowledge discovery, vol. 12, pp. 29–45, 2006.

[3] D. C. Hoaglin and R. E. Welsch, "The hat matrix in regression and anova," The American Statistician, vol. 32, no. 1, pp. 17–22, 1978.

[4] J. W. Tukey et al., Exploratory data analysis. Reading, MA, 1977, vol. 2.

[5] A. S. Hadi and J. S. Simonoff, "Procedures for the identification of multiple outliers in linear models," Journal of the American statistical association, vol. 88, no. 424, pp. 1264–1272, 1993

[6] N. Billor, A. S. Hadi, and P. F. Velleman, "Bacon: blocked adaptive computationally efficient outlier nominators," Computational statistics & data analysis, vol. 34, no. 3, pp. 279–298, 2000.

[7] N. Billor, S. Chatterjee, and A. S. Hadi, "A re-weighted least squares method for robust regression estimation," American journal of mathematical and management sciences, vol. 26, no. 3-4, pp. 229–252, 2006.

[8] D. A. Belsley, E. Kuh, and R. E. Welsch, Regression diagnostics: Identifying influential data and sources of collinearity. John Wiley & Sons, 2005.

[9] S. Barratt, G. Angeris, and S. Boyd, "Minimizing a sum of clipped convex functions," Optimization Letters, vol. 14, pp. 2443–2459, 2020.

[10] S. Chatterjee and M. Mächler, "Robust regression: A weighted least squares approach", Communications in Statistics-Theory and Methods, vol. 26, no. 6, pp. 1381–1394, 1997.

[11] M. H. Satman, "A new algorithm for detecting outliers in linear regression," International Journal of statistics and Probability, vol. 2, no. 3, p. 101, 2013.

[12] L. Huo, T.-H. Kim, and Y. Kim, "Robust estimation of covariance and its application to portfolio optimization," Finance Research Letters, vol. 9, no. 3, pp. 121–134, 2012.

[13] P. J. Rousseeuw, “Least median of squares regression,” Journal of the American statistical association, vol. 79, no. 388, pp. 871–880, 1984.

[14] D. M. Hawkins and D. Olive, “Applications and algorithms for least trimmed sum of absolute deviations regression,” Computational Statistics & Data Analysis, vol. 32, no. 2, pp. 119–134, 1999.

[15] D. M. Hawkins, D. Bradu, and G. V. Kass, “Location of several outliers in multiple-regression data using elemental sets,” Technometrics, vol. 26, no. 3, pp. 197–208, 1984.

[16] D. De Menezes, D. M. Prata, A. R. Secchi, and J. C. Pinto, “A review on robust m-estimators for regression analysis,” Computers & Chemical Engineering, vol. 147, p. 107254, 2021.

[17] P. Rousseeuw and V. Yohai, “Robust regression by means of s-estimators,” in Robust and Non-linear Time Series Analysis: Proceedings of a Workshop Organized by the Sonderforschungs-bereich 123 “Stochastische Mathematische Modelle”, Heidelberg 1983. Springer, pp. 256–272, 1984.

[18] M.H. Satman. "A genetic algorithm based modification on the LTS algorithm for large data sets." Communications in Statistics-Simulation and Computation 41.5, pp. 644-652, 2012.

[19] M.H. Satman, S. Adiga, G. Angeris, and E. Akadal. "LinRegOutliers: A Julia package for detecting outliers in linear regression." Journal of Open Source Software 6, no. 57: 2892, 2021.

[20] J. Bezanson, S. Karpinski, V.B. Shah, V. B., and A. Edelman. Julia: A fast dynamic language for technical computing. arXiv preprint arXiv:1209.5145, 2012