Erasmus University Rotterdam
Erasmus School of Economics

Master Thesis Business Analytics and Quantitative Marketing

Fast and Robust Bootstrap for Logistic Regression

Author: Harmen Schot
Student ID number: 413644

Supervisor: Dr. A. Alfons
Second assessor: Dr. M. Zhelonkin

Date: April 19, 2021

Abstract

This study describes the derivation of a fast and robust bootstrapping procedure for an estimator of the logistic regression model. The (weighted) Bianco & Yohai estimator is used as starting point, but this estimator does not meet the smoothness requirements necessary to derive such a procedure. This study proposes smooth alternatives to this estimator and derives the fast and robust bootstrapping procedures accordingly. A simulation study is conducted, showing that differences in performances between the smooth and original procedures are negligible. Moreover, coverages obtained from confidence intervals using the proposed bootstrapping procedure are very close to those obtained from empirical asymptotic confidence intervals in each setting, except for the smooth weighted Bianco & Yohai estimator in a high-dimensional setting.

Contents

1 Introduction

In many applications, point estimates alone do not give enough information to describe econometric relationships. A scale estimate is necessary to verify significance and in some applications quantiles or even entire distributions may be desired. For most estimators, formulas for asymptotic variances are given, but these rely on assumptions (e.g. symmetry or normality of regressors) that do not necessarily hold in every situation. Consider mediation analysis for instance, where the effect of an independent variable on a dependent variable through a mediator variable is estimated. Shrout and Bolger (2002) showed that in finite samples, indirect effects in mediation analysis are rarely normally distributed, even when the underlying regressors are. The authors propose to extend the bootstrap by Efron (1979) to designs involving multiple mediation. A useful feature of the bootstrap is that it requires less assumptions on the behavior of the regressors. The simplicity of the bootstrap makes it applicable to a wide range of problems and is therefore an interesting starting point for this study. With the bootstrap, the user repeatedly draws subsets from the full sample by random sampling with replacement, computes the desired estimator on each of these subsets, and extracts an empirical distribution from these estimates. If the number of replications is large enough, this empirical distribution should closely reflect the actual distribution, allowing the user to get results without relying on theoretic analysis. However, this procedure does not work well when outliers are present in the data, as shown by Singh (1998). Salibian-Barrera and Zamar (2002) explains that, even when it is used in combination with robust estimation methods, two major issues arise:

- Computational cost: getting an accurate empirical distribution generally requires a large number of replications, with a long computation time as a result, especially when dealing with high-dimensional data. This problem gets magnified when robust estimators are used, because they take more time to compute than non-robust estimators to begin with.
- Numerical instability: every estimator has a breakdown point (Donoho and Huber (1983)), which refers to the fraction of outliers present in the data that can make the performance of the estimator arbitrarily bad. In the context of variance estimation,

> a bad performance may refer to a variance that is arbitrarily large (explosion) or a variance that is arbitrarily close to zero (implosion). Even if the fraction of outliers in the full dataset does not exceed the breakdown point of a specific estimator, the fraction of outliers in a randomly drawn sample can exceed the breakdown point, causing the estimator to break down on that sample. As a result, variance estimates or confidence intervals based on the empirical bootstrap distribution can break down, even if the full-sample estimator does not. In other words: the bootstrap estimator of the sampling distribution of the estimator does not attain the same level of robustness as the estimator itself.

To overcome these issues, the authors proposed the Fast and Robust Bootstrap (FRB). In their paper, the procedure is illustrated in combination with MM-regression by Yohai (1987). First, the estimator and robustness weights are calculated on the full sample. Then, instead of performing all steps of MM-regression on each bootstrap sample, the estimator and the robustness weights of the full sample are used to compute a weighted least-squares fit on each bootstrap sample. To correct for the loss of variability due to the same estimators being used in each replication, a linear correction based on a Taylor expansion around the limit of the estimator is applied. Incorporating the robustness information from the full sample in each bootstrap replication sharply reduces the effect of oversampling outliers. In addition, the most time-consuming calculations are performed only once, greatly contributing to the speed of the procedure. The MM-regression is used as an illustration, but the authors state that the procedure can be applied to other robust methods as well. The most important condition is that the estimator of $\boldsymbol{\theta}$ can be represented as a solution to a smooth fixed-point equation $\hat{\boldsymbol{\theta}}_n = g_n(\hat{\boldsymbol{\theta}}_n)$, in which $g_n(\cdot)$ is a data-dependent function. Smoothness generally refers to the differentiability up to an order k of the function of interest, meaning that its first k derivatives are all continuous. In this thesis, a function is considered smooth if all of its derivatives that are used throughout this thesis are continuous.

The promising results in terms of speed and robustness have motivated other applications of the procedure in the literature. For example, Salibian-Barrera, Van Aelst, and Willems (2006) used FRB with multivariate MM-estimators of location and scatter to perform robust principal component analysis. Salibian-Barrera, Van Aelst, and Willems (2008) used

FRB with S-estimators of location and scatter by P. Rousseeuw and Yohai (1984) to perform robust principal component analysis, multivariate linear regression, and discriminant analysis. Camponovo, Scaillet, and Trojani (2012) extended the procedure to accommodate M-estimators, defined by

$$\Psi_n(\hat{\boldsymbol{\theta}}_n) = \sum_{i=1}^{n} \psi_i(\hat{\boldsymbol{\theta}}_n) = 0, \tag{1}$$

in which $\psi_i(\cdot)$ is a function depending on observation i for $i = 1, \ldots, n$. Peremans, Segaert, Van Aelst, and Verdonck (2017) used this extension to apply the procedure to the Mallows's quasi-likelihood estimator by Cantoni and Ronchetti (2001) in a Poisson regression setting. Poisson regression belongs to the class of generalized linear models (GLMs) and is often used to model count data. This paper focuses on another popular GLM: the logistic regression model. Logistic regression is the most commonly used algorithm for binomial models and it has applications in several fields. Examples are client churn prediction, fraud detection, or estimating the probability of contracting a given disease. To the best of my knowledge, the literature lacks application of the FRB procedure to this model. This study aims to fill this gap by deriving an FRB procedure for the (Weighted) Bianco & Yohai ((W)BY) estimator for logistic regression by Croux and Haesbroeck (2003), which is an extension to the BY estimator proposed by Bianco and Yohai (1996). This is an M estimator and can thus be defined by (1), but an FRB procedure can not yet be accurately derived. The BY estimator uses a piecewise residual weighting function that is not smooth and therefore disrupts the applicability of the Taylor expansion. Its weighted extension faces the same problem and adds two additional ones: firstly, the WBY estimator uses the Minimum Covariance Determinant (MCD) algorithm by P. J. Rousseeuw and Driessen (1999) for computing robust estimates of location and scatter, which is based on selecting a “best” subset of observations and therefore has non-smooth estimating equations. Secondly, it uses a binomial weighting function for filtering out high leverage outliers that is by definition not smooth. I propose to make the following adjustments to smoothen out the (W)BY estimator:

- Replace the residual weighting function by a smooth alternative that resembles the original, but uses a polynomial around the original inflection point to achieve smoothness;

- In the weighted variant, replace the binomial weighting function by a smooth alternative that uses a transformed version of Tukey's biweight family to create a smooth transition from one to zero instead of a leap;
- In the weighted variant, replace the MCD estimators by multivariate MM estimators for location and scatter by Tatsuoka and Tyler (2000), for which the applicability of the FRB procedure has already been proven by Salibian-Barrera et al. (2006).

The resulting estimators will be referred to as the smooth Bianco & Yohai (SBY) estimator and the smooth weighted Bianco & Yohai (SWBY) estimator. The outline of the rest of this thesis is as follows: Section 2 gives an overview of the original methods that are used throughout the thesis, followed by my proposed adjustments and my derivation for an FRB procedure in the logistic regression model. In addition, I describe how asymptotic variances and sampling distributions of bootstrap estimates can be used to derive confidence intervals. Section 3 describes a simulation study that is conducted to evaluate the performance of the proposed procedure. The results of the simulation study are presented in Section 4, followed by my conclusion in Section 5. Lastly, I reflect on my study and suggest further research in Section 6

2 Methodology

This section provides the theoretical framework for my analysis. As the goal of this study is to derive an FRB procedure for the logistic regression model, it is useful to firstly describe the FRB procedure and the (W)BY estimator in general, followed by the proposed adjustments. Next, I provide FRB derivations for the SBY estimator and the multivariate MM estimator individually, of which the latter has previously been derived by Salibian-Barrera et al. (2006). These two results can then be used as building blocks to derive the FRB procedure for the SWBY estimator. Lastly, I describe methods of constructing confidence intervals that are used to verify the performance of the proposed procedure.

2.1 Fast and robust bootstrapping

Consider a generalized setting with dataset $\boldsymbol{Z} = (\boldsymbol{X}, \boldsymbol{y})$, with $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ and $\boldsymbol{y} \in \mathbb{R}^n$, such that row i of $\boldsymbol{Z}$ is defined by $\boldsymbol{z}_i = (\boldsymbol{x}_i^\top, y_i)^\top$. Let there exist a relation within the ith row of $\boldsymbol{Z}$, defined by

$$f(y_i) = \boldsymbol{x}_i^\top \boldsymbol{\theta} + \varepsilon_i \qquad \text{for } i = 1, \dots, n$$

in which $f(\cdot)$ is a link function, the model parameters are denoted by $\boldsymbol{\theta}$, and ε_i is random noise. Let $\hat{\boldsymbol{\theta}}_n$ be the estimate obtained by robust estimation on the full sample and suppose that it can be represented as a solution of fixed-point equations:

$$\hat{\boldsymbol{\theta}}_n = g_n(\hat{\boldsymbol{\theta}}_n), \tag{2}$$

in which $g_n(\cdot)$ is a data-dependent function. Now let $\boldsymbol{Z}^*$ be a bootstrap sample that is obtained by randomly sampling with replacement from $\boldsymbol{Z}$. Applying the same methods on $\boldsymbol{Z}^*$ would yield bootstrap estimate

$$\hat{\boldsymbol{\theta}}_n^* = g_n^*(\hat{\boldsymbol{\theta}}_n^*). \tag{3}$$

However, this is not desirable due to the risk of oversampling of outliers and the computational burden explained in Section 1. Instead, we can compute the approximation

$$\hat{\boldsymbol{\theta}}_n^{1*} = g_n^*(\hat{\boldsymbol{\theta}}_n),$$

in which the full-sample estimate $\hat{\boldsymbol{\theta}}_n$ is used for evaluating $g_n^*(\cdot)$ on the bootstrap sample, diminishing the effects of the oversampling of outliers in a bootstrap sample. It can be seen as a one-step estimation of the bootstrap estimate with initial value $\hat{\boldsymbol{\theta}}_n$. Because the same initial value is used in each bootstrap sample, the variability in the parameters may be underestimated. To fix this, a linear correction is calculated using the following Taylor expansion:

$$\hat{\boldsymbol{\theta}}_n = g_n(\boldsymbol{\theta}) + \nabla g_n(\boldsymbol{\theta})(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}) + \boldsymbol{r}_n,$$

in which $\boldsymbol{r}_n$ is a remainder term and $\nabla g_n(\boldsymbol{\theta})$ is the gradient matrix of $g_n(\cdot)$, confirming the importance of the smoothness of the estimating equations. When $\boldsymbol{r}_n$ is negligible ($\boldsymbol{r}_n = o_p(1/\sqrt{n})$), we can write

$$\sqrt{n}(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}) \sim [\boldsymbol{I} - \nabla g_n(\boldsymbol{\theta})]^{-1}\sqrt{n}(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}).$$

Under certain conditions, $\sqrt{n}(\hat{\boldsymbol{\theta}}_n^* - \hat{\boldsymbol{\theta}}_n)$ and $\sqrt{n}(g_n^*(\boldsymbol{\theta}) - \boldsymbol{\theta})$ converge to the same limiting distributions as $\sqrt{n}(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta})$ and $\sqrt{n}(g_n(\hat{\boldsymbol{\theta}}_n) - \hat{\boldsymbol{\theta}}_n)$, respectively. If we then approximate $[\boldsymbol{I} - \nabla g_n(\boldsymbol{\theta})]^{-1}$ by $[\boldsymbol{I} - \nabla g_n(\hat{\boldsymbol{\theta}}_n)]^{-1}$ we get

$$\sqrt{n}(\hat{\boldsymbol{\theta}}_n^* - \hat{\boldsymbol{\theta}}_n) \sim [\boldsymbol{I} - \nabla g_n(\hat{\boldsymbol{\theta}}_n)]^{-1}\sqrt{n}(g_n^*(\hat{\boldsymbol{\theta}}_n) - \hat{\boldsymbol{\theta}}_n),$$

in which $\boldsymbol{I}$ denotes the identity matrix. Now, a corrected bootstrap approximation can be defined by

$$\hat{\boldsymbol{\theta}}_n^{(b)} = \hat{\boldsymbol{\theta}}_n + [\boldsymbol{I} - \nabla g_n(\hat{\boldsymbol{\theta}}_n)]^{-1}(g_n^*(\hat{\boldsymbol{\theta}}_n) - \hat{\boldsymbol{\theta}}_n). \tag{4}$$

For a more elaborate derivation and proofs, I refer to Salibian-Barrera (2000) and Salibian-Barrera and Zamar (2002). The resulting bootstrap estimates are fast to compute and more robust than ordinary bootstrap estimates as defined by (3).

Camponovo et al. (2012) provided an extension to accommodate M estimators defined by relations of the form

$$\Psi_n(\hat{\boldsymbol{\theta}}_n) = \sum_{i=1}^{n} \psi_i(\hat{\boldsymbol{\theta}}_n) = 0.$$

Following similar steps as the generalized setting, Camponovo et al. (2012) provided bootstrap approximations for these M estimators, given by

$$\hat{\boldsymbol{\theta}}_n^{(b)} = \hat{\boldsymbol{\theta}}_n + [-\nabla \Psi_n(\hat{\boldsymbol{\theta}}_n)]^{-1} \Psi_n^*(\hat{\boldsymbol{\theta}}_n). \tag{5}$$

This study aims to use MM estimators for location and scatter alongside the M-estimator that is the WBY estimator. The MM estimator is a special type of M-estimator, hence it is possible to combine their FRB procedures. Using the fact that MM estimator $\hat{\boldsymbol{\theta}}_n$ satisfies (2), we can write

$$\Psi_n(\hat{\boldsymbol{\theta}}_n) = g_n(\hat{\boldsymbol{\theta}}_n) - \hat{\boldsymbol{\theta}}_n = \mathbf{0}, \tag{6}$$

such that the gradient becomes

$$\nabla \Psi_n(\hat{\boldsymbol{\theta}}_n) = \nabla g_n(\hat{\boldsymbol{\theta}}_n) - \boldsymbol{I}. \tag{7}$$

Next, plugging in (6) and (7) in (5) results in (4), confirming that writing the MM-estimator as an M-estimator results in the same expression.

2.2 Bianco & Yohai estimator

The logistic regression model belongs to the class of generalized linear models and is most frequently used to model a binomial dependent variable. In the following, let the relation between dependent variable $y_i \in \{0, 1\}$ and explanatory variables $\boldsymbol{x}_i = (1, \tilde{\boldsymbol{x}}_i)^\top$ be given by

$$P(Y_i = 1|\boldsymbol{x}_i) = F(\boldsymbol{x}_i^\top \boldsymbol{\beta}) = \frac{1}{1 + \exp(-\boldsymbol{x}_i^\top \boldsymbol{\beta})},$$

for $i = 1, \ldots, n$ and with coefficient vector $\boldsymbol{\beta} \in \mathbb{R}^{p+1}$. The BY estimator as proposed by Croux and Haesbroeck (2003) is then defined as follows. First, let the residual weighting function be defined by

$$\rho_{\text{SBY}}(r) = \begin{cases} r \exp(-\sqrt{c}) & \text{if } |r| \leq c, \\ -2 \exp(-\sqrt{r})(1 + \sqrt{r}) + \exp(-\sqrt{c})(2(1 + \sqrt{c}) + c) & \text{otherwise,} \end{cases}$$

in which c is a tuning constant that can be chosen to yield a certain level of asymptotic efficiency. A useful feature of this function is that its derivative, given by

$$\rho'_{\mathrm{BY}}(r) = \begin{cases} \exp\left(-\sqrt{c}\right) & \text{if } |r| \leq c, \\ \exp\left(-\sqrt{r}\right) & \text{otherwise,} \end{cases}$$

does not drop entirely to zero for large deviations. The derivative occurs in the estimating equations and is used for updating the coefficients in the optimization process, hence a zero derivative would imply a severe downweighting of misclassified observations. This would be undesirable, as misclassified observations are the observations that tell the estimator in which direction to move the decision hyperplane. Next, define the deviance component of observation i by

$$d(\boldsymbol{x}_i^\top \boldsymbol{\beta}; y_i) = -y_i \log F(\boldsymbol{x}_i^\top \boldsymbol{\beta}) - (1 - y_i) \log\left\{1 - F(\boldsymbol{x}_i^\top \boldsymbol{\beta})\right\}.$$

Then, a bias correction term to ensure Fisher consistency is computed as

$$C(\boldsymbol{x}_i^\top \boldsymbol{\beta}) = G(F(\boldsymbol{x}_i^\top \boldsymbol{\beta})) + G(1 - F(\boldsymbol{x}_i^\top \boldsymbol{\beta})) - G(1),$$

in which

$$G(t) = \int_0^t \rho_{\mathrm{BY}}(-\log u) du.$$

These parts are enough to define the non-weighted BY estimator, but to achieve an overall bounded influence function, a leverage based weighting step can be added. First, define the squared robust Mahalanobis distance for observation i as

$$RD_i^2 = (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_n)^\top \hat{\boldsymbol{\Sigma}}_n^{-1} (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_n),$$

in which $\hat{\boldsymbol{\mu}}_n$ and $\hat{\boldsymbol{\Sigma}}_n$ denote robust estimators for location and scatter. The default method for these estimates is the MCD estimator. The weights of observations $i = 1, \ldots, n$ are then defined as

$$w_i = \begin{cases} 1, & \text{if } RD_i^2 \leq \chi^2_{p,0.975} \\ 0, & \text{otherwise,} \end{cases}$$

in which $\chi^2_{p,0.975}$ denotes the 97.5% critical value of the chi-squared distribution with p degrees of freedom. Combining the definitions given above, the WBY estimator is defined as

$$\hat{\boldsymbol{\beta}}_n = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} w_i \psi_{\mathrm{BY}}(\boldsymbol{x}_i^\top \boldsymbol{\beta}; y_i) = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} w_i \big(\rho_{\mathrm{BY}}(d(\boldsymbol{x}_i^\top \boldsymbol{\beta}; y_i)) + C(\boldsymbol{x}_i^\top \boldsymbol{\beta}) \big),$$

which reduces to the BY estimator when the unit weighting function $w_i = 1$ is used. The associated asymptotic variance can be calculated using the results from Maronna and Yohai (1981) and by taking the calculated weights into account, so that we get

$$\hat{\boldsymbol{V}} = \frac{1}{\sum_{i=1}^n w_i} \hat{\boldsymbol{D}}^{-1} \hat{\boldsymbol{C}} \hat{\boldsymbol{D}}^{-1}, \tag{8}$$

where the matrices $\hat{\boldsymbol{D}}$ and $\hat{\boldsymbol{C}}$ are defined as

$$\hat{\boldsymbol{D}} = \frac{1}{\sum_{i=1}^n w_i} \sum_{i=1}^{n} \psi''_{\mathrm{BY}}(\boldsymbol{x}_i^\top \boldsymbol{\beta}; y_i) \boldsymbol{x}_i \boldsymbol{x}_i^\top$$

and

$$\hat{\boldsymbol{C}} = \frac{1}{\sum_{i=1}^n w_i} \sum_{i=1}^{n} \psi'_{\mathrm{BY}}(\boldsymbol{x}_i^\top \boldsymbol{\beta}; y_i)^2 \boldsymbol{x}_i \boldsymbol{x}_i^\top. \tag{9}$$

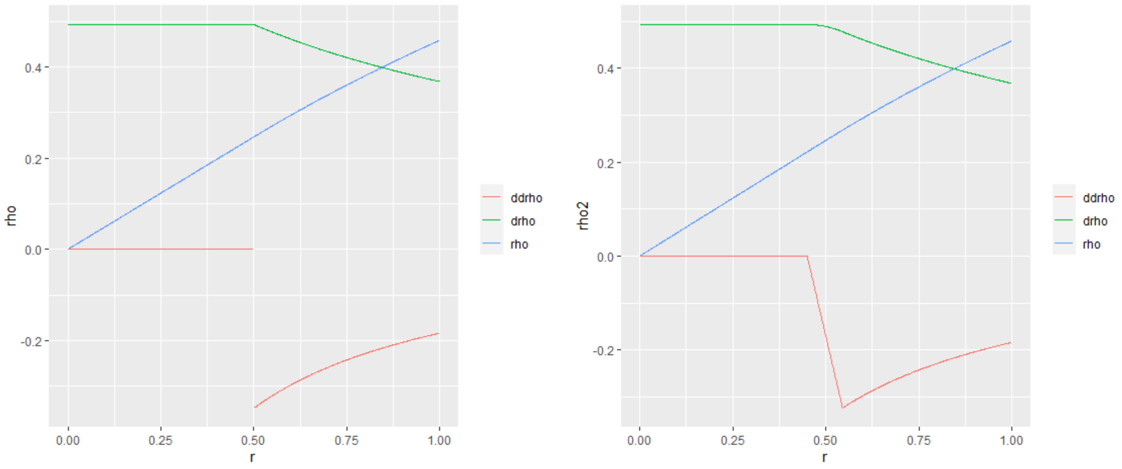
When the unit weighting function is used, the terms $\sum_{i=1}^n w_i$ in (8-9) reduce to $\sum_{i=1}^n 1 = n$. The asymptotic standard errors are then given by

$$\hat{\sigma}(\beta_{n,j}) = \sqrt{\boldsymbol{V}_{j+1,j+1}}, \qquad j = 0, \ldots, p, \tag{10}$$

where $\hat{\boldsymbol{V}}_{j+1,j+1}$ denotes the $(j+1)$th diagonal element of asymptotic variance matrix $\hat{\boldsymbol{V}}$.

2.3 Smooth Bianco & Yohai estimator

The Taylor expansion of the FRB algorithm requires the estimating equations to be smooth, meaning that their gradient should be continuous. The reason for this is that the gradient in



(a) Original ρ

(b) Smooth ρ

Figure 1: Comparison between original and smooth ρ-functions and their first and second derivatives.

a given point should give an accurate representation of the change in function value around that point. Since the derivative of the ρ-function is used in the estimating equations of the BY estimator, we need its second derivative to be continuous. Its second derivative is given by

$$\rho''_{\text{BY}}(r) = \begin{cases} 0 & \text{if } |r| \leq c, \\ -\frac{1}{2\sqrt{t}} \exp\left(-\sqrt{t}\right) & \text{otherwise,} \end{cases}$$

which is not continuous around the constant c, because

$$\lim_{r \uparrow c} \rho''_{\text{BY}}(r) = 0 \quad \text{and} \quad \lim_{r \downarrow c} \rho''_{\text{BY}}(r) = -\frac{1}{2\sqrt{c}} \exp\left(-\sqrt{c}\right).$$

The slope of the tangent line in point c could therefore be anything between these two values. Technically, the function approaches continuity as c approaches infinity, but that choice would imply a linear residual weighting function which is not robust. The original ρ-function along with its first and second derivatives are shown in Figure 1a, where the default constant $c = 0.5$ is used. I propose to use a third-degree polynomial on a narrow region around c so that the second derivative reduces to a line that connects the two parts of the function that were previously not connected. The newly proposed residual weighting function is then of the following form:

$$\rho_{\text{SBY}}(r) = \begin{cases} \alpha_1 r & \text{if } |r| \leq c_1, \\ \alpha_2 r^3 + \alpha_3 r^2 + \alpha_4 r + \alpha_5 & \text{if } c_1 < |r| \leq c_2, \\ -2\exp(-\sqrt{r})(1+\sqrt{r}) + \exp(-\sqrt{c})(2(1+\sqrt{c})+c) & \text{otherwise,} \end{cases} \tag{11}$$

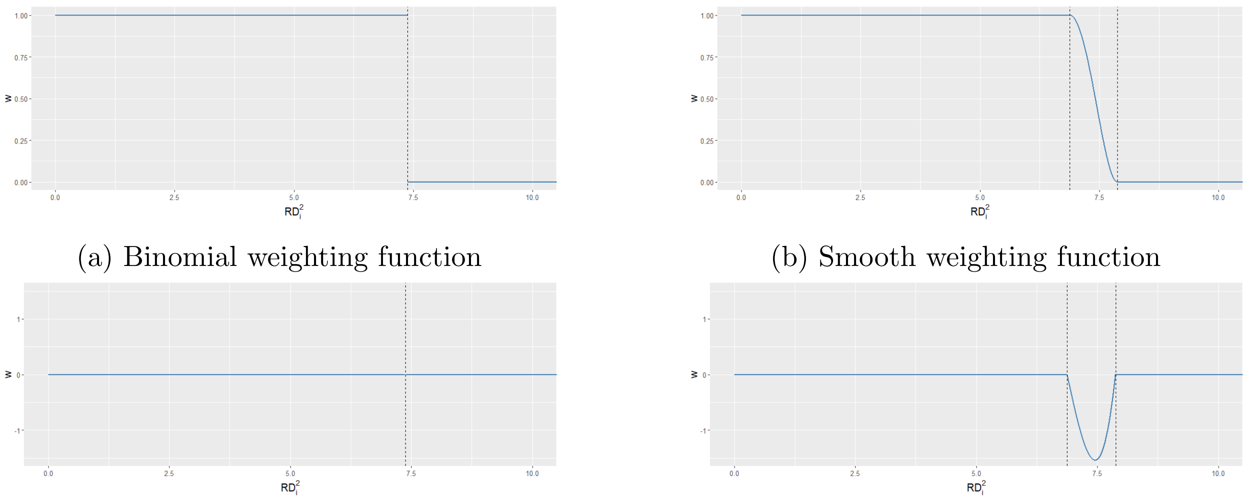
in which $c_1, c_2, \alpha_1, \alpha_2, \alpha_3, \alpha_4$, and α_5 are constants that help determine the location and shape of the polynomial. The tuning parameter c is the same as used in the original ρ-function. The derivative is given by

$$\rho'_{\text{SBY}}(r) = \begin{cases} r & \text{if } |r| \leq c_1, \\ 3\alpha_2 r^2 + 2\alpha_3 r + \alpha_4 & \text{if } c_1 < |r| \leq c_2, \\ \exp(-\sqrt{r}) & \text{otherwise,} \end{cases}$$

and the second derivative by

$$\rho''_{\text{SBY}}(r) = \begin{cases} 0 & \text{if } |r| \leq c_1, \\ 6\alpha_2 r + 2\alpha_3 & \text{if } c_1 < |r| \leq c_2, \\ -\frac{1}{2\sqrt{r}}\exp(-\sqrt{r}) & \text{otherwise.} \end{cases}$$

The idea is to choose $c_1 \in (0, c)$ and then compute the other parameters by solving a system of equations ensuring that all function parts align and meet with the original functions at c_1 and c_2. The derivation for these computations is given in Appendix A. The newly proposed ρ-function along with its first and second derivatives are shown in Figure 1b, where $c = 0.5$ and $c_1 = 0.45$ are used. There exists a tradeoff between smoothness of the new function and resemblance to the original. Choosing c_1 further from c implies a more smooth transition but a larger difference from the original function and vice versa. Investigating the optimal choice of c_1 will be left for further research. The SBY estimator is then defined as the BY estimator with the smooth residual weighting function in (11).



(a) Binomial weighting function

(b) Smooth weighting function

(c) Derivative of binomial weighting function

(d) Derivative of smooth weighting function

Figure 2: Comparison of two weighting functions and their derivatives with turning point $\chi^2_{2,0.975}$.

2.4 Smooth weighted Bianco & Yohai estimator

The WBY estimator extends the BY estimator with a preceding weighting step. The idea of this weighting step is to filter out outliers in the covariate space beforehand, such that their leverage cannot cause disruptions in the estimation process and an overall bounded influence function can be achieved. Although this approach adds robustness in certain circumstances and has an intuitive interpretation, it faces the same problem as the residual weighting function: its estimating equations lack smoothness. The binomial weighting function and its derivative are shown in Figures 2a and 2c and show that the derivative is undefined for $RD_i^2 = \chi^2_{p,0.975}$ and zero for all other positive values. The problem that arises becomes clear when looking at observations that are close to the critical value. As an example, consider a situation where $(\hat{\boldsymbol{\beta}}_n^\top, \hat{\boldsymbol{\mu}}_n^\top, \text{vec}(\hat{\boldsymbol{\Sigma}}_n)^\top)^\top$ are known and there exists a point $\boldsymbol{x}_i \in \mathbb{R}^p$ with corresponding squared Mahalanobis distance $RD^2(\boldsymbol{x}_i; \hat{\boldsymbol{\mu}}_n, \hat{\boldsymbol{\Sigma}}_n) = \chi^2_{p,0.975} - \epsilon$ and $\epsilon \downarrow 0$, so that $w_i = 1$. Now assume that new estimates $(\hat{\boldsymbol{\beta}}_n^{*\top}, \hat{\boldsymbol{\mu}}_n^{*\top}, \text{vec}(\hat{\boldsymbol{\Sigma}}_n^*)^\top)^\top$ would be calculated on the bootstrap sample and that $RD^2(\boldsymbol{x}_i; \hat{\boldsymbol{\mu}}_n^*, \hat{\boldsymbol{\Sigma}}_n^*) > \chi^2_{p,0.975}$ so that a new weighting would lead to $w_i = 0$. This change is completely ignored when the binomial weighting function is used, because the Taylor expansion uses the derivative which is zero (or even undefined in extreme cases). I propose to use a smooth alternative to the binomial weighting function, based on Tukey's biweight family. The smooth weighting function is defined by

$$w_i = \begin{cases} 1, & \text{if } RD_i^2 \leq c - \frac{h}{2}, \\ \frac{1}{h^4}\left(h^2 - \left(RD_i^2 - \left(c - \frac{h}{2}\right)\right)^2\right)^2, & \text{if } c - \frac{h}{2} < RD_i^2 \leq c + \frac{h}{2}, \\ 0, & \text{otherwise,} \end{cases} \tag{12}$$

in which $c = \chi^2_{p,0.975}$ and h is a tuning parameter representing the width of the horizon on which the function is smoothed out. I use $h = 1$ by default, but other positive values of h are possible as well. Again, there exists a tradeoff between smoothness and resemblance to the original, which can be investigated in later research. Figures 2b and 2d show the behavior of the smooth weighting function, from which it is visible that the derivative more accurately represents the change in function value around the critical value than its binomial counterpart. The derivative of the smooth weighting function is given by

$$\frac{\partial w_i}{\partial RD_i^2} = \begin{cases} 0, & \text{if } RD_i^2 \leq c - \frac{h}{2}, \\ -\frac{4}{h^4}\left(RD_i^2 - \left(c - \frac{h}{2}\right)\right)\left(h^2 - \left(RD_i^2 - \left(c - \frac{h}{2}\right)\right)^2\right)^2, & \text{if } c - \frac{h}{2} < RD_i^2 \leq c + \frac{h}{2}, \\ 0, & \text{otherwise.} \end{cases}$$

The inputs for the weighting function are Mahalanobis distances between the observations and robust estimates of location and scatter. The MCD estimator for location and scatter is the default method for obtaining these estimates. However, this estimator is based on selecting an optimal subset of observations and cannot be written as a solution of smooth fixed-point equations. Applying an FRB procedure is therefore not feasible and another method is required. I propose to use the multivariate MM estimators for location and scatter by Tatsuoka and Tyler (2000) instead of the MCD estimator. Salibian-Barrera et al. (2006) already applied the FRB procedure in combination with a slightly adapted version of this MM estimator. The idea of the method is to estimate the scale using a highly robust S estimator and then use this scale estimate to compute a highly efficient M estimator for location and scatter. Both steps use a function from Tukey's biweight family as residual weighting function, defined by

$$\rho(r) = \begin{cases} \frac{r^2}{2} - \frac{r^4}{2c^2} + \frac{r^6}{6c^4} & \text{if } |u| \leq c, \\ 0 & \text{otherwise,} \end{cases}$$

in which c is a tuning constant. Let the determinant of an arbitrary matrix $\boldsymbol{A}$ be denoted by $|\boldsymbol{A}|$. The S estimators for location and scatter $(\tilde{\boldsymbol{\mu}}_n, \tilde{\boldsymbol{\Sigma}}_n)$ are then defined by $(\boldsymbol{m}_0, \boldsymbol{S}_0)$ that minimize $|\boldsymbol{S}_0|$ subject to

$$\frac{1}{n}\sum_{i=1}^{n} \rho_0\Big([(\boldsymbol{x}_i - \boldsymbol{m}_0)^\top \boldsymbol{S}_0^{-1}(\boldsymbol{x}_i - \boldsymbol{m}_0)]^{\frac{1}{2}}\Big) = b,$$

in which b and the tuning constant of $\rho_0(\cdot)$ are typically chosen to yield a 50% breakdown point. Next, the MM estimators for location and shape $(\hat{\boldsymbol{\mu}}_n, \hat{\boldsymbol{\Gamma}}_n)$ are defined by $(\boldsymbol{m}, \boldsymbol{G})$ that minimize

$$\frac{1}{n}\sum_{i=1}^{n} \rho_1\Big([(\boldsymbol{x}_i - \boldsymbol{m})^\top \boldsymbol{G}^{-1}(\boldsymbol{x}_i - \boldsymbol{m})]^{\frac{1}{2}} / |\boldsymbol{S}_0|^{\frac{1}{2p}}\Big) \tag{13}$$

subject to $|\boldsymbol{G}| = 1$, in which the tuning constant of $\rho_1(\cdot)$ is typically chosen to yield 95% efficiency. The MM estimator of scatter is then given by

$$\hat{\boldsymbol{\Sigma}}_n = |\tilde{\boldsymbol{\Sigma}}_n|^{\frac{1}{p}} \hat{\boldsymbol{\Gamma}}_n. \tag{14}$$

The SWBY estimator is then defined as the WBY estimator with the smooth residual weighting function in (11) and the smooth weighting function in (12), with the robust Mahalanobis distances taken with respect to the multivariate MM estimators for location and scatter in (13-14).

2.5 Fast and robust bootstrapping derivations

Because of the extensive derivations, the FRB procedure for the SWBY estimator is split into parts. First, I derive the procedure for the non-weighted SBY estimator, followed by the procedure for the multivariate MM estimator. Then, these two procedures are combined into a procedure for the SWBY estimator. In each derivation, the goal is to derive expressions for $\Psi_n(\hat{\boldsymbol{\theta}}_n)$ and $\nabla\Psi_n(\hat{\boldsymbol{\theta}}_n)$ in $\hat{\boldsymbol{\theta}}_n^{(b)} = \hat{\boldsymbol{\theta}}_n + [-\nabla\Psi_n(\hat{\boldsymbol{\theta}}_n)]^{-1}\Psi_n^*(\hat{\boldsymbol{\theta}}_n)$.

Smooth Bianco & Yohai estimator

Consider the BY estimator $\hat{\boldsymbol{\beta}}_n$ with corresponding objective function

$$\sum_{i=1}^n \Big(\rho_{\text{SBY}}(d(\boldsymbol{x}_i^\top\boldsymbol{\beta}; y_i)) + C(\boldsymbol{x}_i^\top\boldsymbol{\beta})\Big)$$

that has to be minimized. The estimating equation follows from differentiating the function above. In the following, define $F_i = F(\boldsymbol{x}_i^\top\boldsymbol{\beta}) = 1/(1+\exp(-\boldsymbol{x}_i^\top\boldsymbol{\beta}))$ for ease of notation. Next, calculate

$$\begin{aligned}
\frac{\partial F_i}{\partial\boldsymbol{\beta}} &= \frac{\exp(-\boldsymbol{x}_i^\top\boldsymbol{\beta})\boldsymbol{x}_i}{(1+\exp(-\boldsymbol{x}_i^\top\boldsymbol{\beta}))^2} && = F_i(1-F_i)\boldsymbol{x}_i,\\
\frac{\partial \log F_i}{\partial\boldsymbol{\beta}} &= \frac{F_i(1-F_i)\boldsymbol{x}_i}{F_i} && = (1-F_i)\boldsymbol{x}_i,\\
\frac{\partial \log\{1-F_i\}}{\partial\boldsymbol{\beta}} &= \frac{-F_i(1-F_i)\boldsymbol{x}_i}{1-F_i} && = -F_i\boldsymbol{x}_i,\\
\frac{\partial d(\boldsymbol{x}_i^\top\boldsymbol{\beta}; y_i)}{\partial\boldsymbol{\beta}} &= -y_i(1-F_i)\boldsymbol{x}_i + (1-y_i)F_i\boldsymbol{x}_i && = (F_i - y_i)\boldsymbol{x}_i,\\
\frac{\partial G(F_i)}{\partial\boldsymbol{\beta}} &= \frac{\partial}{\partial\boldsymbol{\beta}}\int_0^{F_i}\rho'_{\text{SBY}}(-\log u)du && \overset{(L)}{=} F_i(1-F_i)\rho'_{\text{SBY}}(-\log F_i)\boldsymbol{x}_i,\\
\frac{\partial G(1-F_i)}{\partial\boldsymbol{\beta}} &= \frac{\partial}{\partial\boldsymbol{\beta}}\int_0^{1-F_i}\rho'_{\text{SBY}}(-\log u)du && \overset{(L)}{=} -F_i(1-F_i)\rho'_{\text{SBY}}(-\log\{1-F_i\})\boldsymbol{x}_i,\\
\frac{\partial G(1)}{\partial\boldsymbol{\beta}} &= 0,
\end{aligned}$$

where the superscripts (L) follow from the use of Leibniz' integral rule, defined by

$$\frac{\partial}{\partial x}\int_0^{f(x)} g(u)du = g(f(x))\frac{\partial}{\partial x}f(x).$$

Collecting terms and using the chain rule for derivatives, the resulting estimating equation becomes

$$\Psi_{\text{SBY}}(\boldsymbol{\beta}) = \sum_{i=1}^n \Psi_i(\boldsymbol{\beta}) \tag{15}$$

$$= \sum_{i=1}^{n} \Big((F_i - y_i)\rho'_{\text{SBY}}(d(\boldsymbol{x}_i^\top \boldsymbol{\beta}; y_i)) + F_i(1 - F_i)\big(\rho'_{\text{SBY}}(-\log F_i) - \rho'_{\text{SBY}}(-\log\{1 - F_i\})\big)\Big)\boldsymbol{x}_i.$$

The following intermediate results are useful for the computation of the gradient matrix:

$$\begin{aligned}
\frac{\partial (F_i - y_i)\rho'_{\text{SBY}}(d(\boldsymbol{x}_i^\top \boldsymbol{\beta}))}{\partial \boldsymbol{\beta}} &= F_i(1 - F_i)\rho'_{\text{SBY}}(d(\boldsymbol{x}_i^\top \boldsymbol{\beta}))\boldsymbol{x}_i + (F_i - y_i)^2 \rho''_{\text{SBY}}(d(\boldsymbol{x}_i^\top \boldsymbol{\beta}))\boldsymbol{x}_i, \\
\frac{\partial F_i(1 - F_i)}{\partial \boldsymbol{\beta}} &= F_i(1 - F_i) - 2F_i^2(1 - F_i) = 2F_i^3 - 3F_i^2 + F_i, \\
\frac{\partial \rho'_{\text{SBY}}(-\log F_i)}{\partial \boldsymbol{\beta}} &= -(1 - F_i)\rho''_{\text{SBY}}(-\log F_i)\boldsymbol{x}_i, \\
\frac{\partial \rho'_{\text{SBY}}(-\log\{1 - F_i\})}{\partial \boldsymbol{\beta}} &= F_i \rho''_{\text{SBY}}(-\log\{1 - F_i\}).
\end{aligned}$$

Combining the results above and again using the chain rule for derivatives, the gradient matrix can be computed as

$$\begin{aligned}
\nabla\Psi(\boldsymbol{\beta}) = \sum_{i=1}^{n} \Big(& F_i(1 - F_i)\rho'(d(\boldsymbol{x}_i^\top \boldsymbol{\beta})) + (F_i - y_i)^2 \rho''_{\text{SBY}}(d(\boldsymbol{x}_i^\top \boldsymbol{\beta})) \\
& + (2F_i^3 - 3F_i^2 + F_i)\big(\rho'_{\text{SBY}}(-\log F_i) - \rho'_{\text{SBY}}(-\log\{1 - F_i\})\big) \\
& + F_i(1 - F_i)\big(- (1 - F_i)\rho''_{\text{SBY}}(-\log F_i) - F_i \rho''_{\text{SBY}}(-\log\{1 - F_i\})\big)\Big)\boldsymbol{x}_i\boldsymbol{x}_i^\top. \quad (16)
\end{aligned}$$

Multivariate MM estimators for location and scatter

The following derivations are based on the descriptions given by Salibian-Barrera et al. (2006). Let the S estimators and MM estimators for location and scatter be represented by $\boldsymbol{m}_0$, $\boldsymbol{S}_0$, $\boldsymbol{m}$, and $\boldsymbol{S}$. In addition, define the following variables for notational convenience:

$$\begin{aligned}
d_i &= [(\boldsymbol{x}_i - \boldsymbol{m})^\top \boldsymbol{S}^{-1}(\boldsymbol{x}_i - \boldsymbol{m})]^{\frac{1}{2}}, & \tilde{d}_i &= [(\boldsymbol{x}_i - \boldsymbol{m}_0)^\top \boldsymbol{S}_0^{-1}(\boldsymbol{x}_i - \boldsymbol{m}_0)]^{\frac{1}{2}}, \\
a &= \sum_{i=1}^{n} \frac{\rho'_1(d_i)}{d_i}, & a_0 &= \sum_{i=1}^{n} \frac{\rho'_0(\tilde{d}_i)}{\tilde{d}_i}, \\
\boldsymbol{b} &= \sum_{i=1}^{n} \frac{\rho'_1(d_i)}{d_i}\boldsymbol{x}_i, & \boldsymbol{b}_0 &= \sum_{i=1}^{n} \frac{\rho'_0(\tilde{d}_i)}{\tilde{d}_i}\boldsymbol{x}_i, \\
\boldsymbol{V} &= \sum_{i=1}^{n} \frac{\rho'_1(d_i)}{d_i}(\boldsymbol{x}_i - \boldsymbol{m})(\boldsymbol{x}_i - \boldsymbol{m})^\top, & \boldsymbol{V}_0 &= \frac{p}{nb}\sum_{i=1}^{n} \frac{\rho'_0(\tilde{d}_i)}{\tilde{d}_i}(\boldsymbol{x}_i - \boldsymbol{m}_0)(\boldsymbol{x}_i - \boldsymbol{m}_0)^\top,
\end{aligned}$$

$$\begin{aligned}
\operatorname{vec}(\boldsymbol{V}) &= \sum_{i=1}^{n} \frac{\rho_1'(d_i)}{d_i}(\boldsymbol{x}_i - \boldsymbol{m}) \otimes (\boldsymbol{x}_i - \boldsymbol{m}), & \operatorname{vec}(\boldsymbol{V}_0) &= \frac{p}{nb} \sum_{i=1}^{n} \frac{\rho_1'(\tilde{d}_i)}{\tilde{d}_i}(\boldsymbol{x}_i - \boldsymbol{m}_0) \otimes (\boldsymbol{x}_i - \boldsymbol{m}_0), \\
\phi_i &= \frac{\rho_1'(d_i) - d_i\rho_1''(d_i)}{d_i^3}, & \tilde{\phi}_i &= \frac{\rho_0'(\tilde{d}_i) - \rho_0''(\tilde{d}_i)}{\tilde{d}_i^3}, \\
& & \tilde{w} &= \frac{1}{nb}\sum_{i=1}^{n}(\rho_0(\tilde{d}_i) - \rho_0'(\tilde{d}_i)d_i).
\end{aligned}$$

Using the comparison to M estimators from Section 2.1, the estimating equations for the multivariate MM algorithm can then be written as follows:

$$\Psi_n(\boldsymbol{m}) = \boldsymbol{b}/a - \boldsymbol{m}, \tag{17}$$

$$\Psi_n(\boldsymbol{S}) = |\boldsymbol{S}_0|^{\frac{1}{p}}|\boldsymbol{V}|^{-\frac{1}{p}}\boldsymbol{V} - \boldsymbol{S}, \tag{18}$$

$$\Psi_n(\boldsymbol{m}_0) = \boldsymbol{b}_0/a_0 - \boldsymbol{m}_0, \tag{19}$$

$$\Psi_n(\boldsymbol{S}_0) = \boldsymbol{V}_0 + \tilde{w}_n\boldsymbol{S}_0 - \boldsymbol{S}_0. \tag{20}$$

The partial derivatives of (17-20) can be used to construct the following gradient matrix.

Next, we are interested in the partial derivatives of the equations above, for which the following intermediary results are useful:

$$\begin{aligned}
\frac{\partial \rho_1'(d_i)/d_i}{\partial d_i} &= \frac{d_i\rho_1''(d_i) - \rho_1'(d_i)}{d_i^2} = -d_i\phi_i, \\
\frac{\partial d_i}{\partial \boldsymbol{m}} &= -\frac{1}{d_i}\boldsymbol{S}^{-1}(\boldsymbol{x}_i - \boldsymbol{m}), \\
\frac{\partial \rho_1'(d_i)/d_i}{\partial \boldsymbol{m}} &= \phi_i\boldsymbol{S}^{-1}(\boldsymbol{x}_i - \boldsymbol{m}), \\
\frac{\partial d_i}{\partial \operatorname{vec}(\boldsymbol{S})} &= -\frac{1}{2d_i}\boldsymbol{S}^{-1}(\boldsymbol{x}_i - \boldsymbol{m}) \otimes \boldsymbol{S}^{-1}(\boldsymbol{x}_i - \boldsymbol{m}), \qquad (21) \\
\frac{\partial \rho_1'(d_i)/d_i}{\partial \operatorname{vec}(\boldsymbol{S})} &= \frac{1}{2}\phi_i\boldsymbol{S}^{-1}(\boldsymbol{x}_i - \boldsymbol{m}) \otimes \boldsymbol{S}^{-1}(\boldsymbol{x}_i - \boldsymbol{m}), \\
\frac{\partial \operatorname{vec}(\boldsymbol{V})}{\partial \boldsymbol{m}} &= \sum_{i=1}^{n}\Bigg(\phi_i[(\boldsymbol{x}_i - \boldsymbol{m}) \otimes (\boldsymbol{x}_i - \boldsymbol{m})](\boldsymbol{x}_i - \boldsymbol{m})^\top\boldsymbol{S}^{-1} - \\
&\qquad \frac{\rho_1(d_i)}{d_i}[\boldsymbol{I}_p \otimes (\boldsymbol{x}_i - \boldsymbol{m}) + (\boldsymbol{x}_i - \boldsymbol{m}) \otimes \boldsymbol{I}_p]\Bigg), \qquad (22) \\
\frac{\partial \operatorname{vec}(\boldsymbol{V})}{\partial \operatorname{vec}(\boldsymbol{S})} &= \frac{1}{2}\sum_{i=1}^{n}\phi_i[(\boldsymbol{x}_i - \boldsymbol{m}) \otimes (\boldsymbol{x}_i - \boldsymbol{m})][\boldsymbol{S}^{-1}(\boldsymbol{x}_i - \boldsymbol{m}) \otimes \boldsymbol{S}^{-1}(\boldsymbol{x}_i - \boldsymbol{m})]^\top,
\end{aligned}$$

and similarly

$$\begin{aligned}
\frac{\partial \rho_0'(\tilde{d}_i)/\tilde{d}_i}{\partial \tilde{d}_i} =& \frac{\tilde{d}_i\rho_0''(\tilde{d}_i) - \rho_0'(\tilde{d}_i)}{\tilde{d}_i^2} = -\tilde{d}_i\tilde{\phi}_i, \\
\frac{\partial \tilde{d}_i}{\partial \boldsymbol{m}_0} =& -\frac{1}{\tilde{d}_i}\boldsymbol{S}_0^{-1}(\boldsymbol{x}_i - \boldsymbol{m}_0), \\
\frac{\partial \rho_0'(\tilde{d}_i)/\tilde{d}_i}{\partial \boldsymbol{m}_0} =& \tilde{\phi}_i\boldsymbol{S}_0^{-1}(\boldsymbol{x}_i - \boldsymbol{m}_0), \\
\frac{\partial \tilde{d}_i}{\partial \operatorname{vec}(\boldsymbol{S}_0)} =& -\frac{1}{2\tilde{d}_i}\boldsymbol{S}_0^{-1}(\boldsymbol{x}_i - \boldsymbol{m}_0) \otimes \boldsymbol{S}_0^{-1}(\boldsymbol{x}_i - \boldsymbol{m}_0), \qquad (23) \\
\frac{\partial \rho_0'(\tilde{d}_i)/\tilde{d}_i}{\partial \operatorname{vec}(\boldsymbol{S}_0)} =& \frac{1}{2}\tilde{\phi}_i\boldsymbol{S}_0^{-1}(\boldsymbol{x}_i - \boldsymbol{m}_0) \otimes \boldsymbol{S}_0^{-1}(\boldsymbol{x}_i - \boldsymbol{m}_0), \\
\frac{\partial \operatorname{vec}(\boldsymbol{V}_0)}{\partial \boldsymbol{m}_0} =& \frac{p}{nb}\sum_{i=1}^{n}\Bigg(\tilde{\phi}_i[(\boldsymbol{x}_i - \boldsymbol{m}_0) \otimes (\boldsymbol{x}_i - \boldsymbol{m}_0)](\boldsymbol{x}_i - \boldsymbol{m}_0)^\top\boldsymbol{S}_0^{-1} - \\
& \frac{\rho_0(\tilde{d}_i)}{\tilde{d}_i}[\boldsymbol{I}_p \otimes (\boldsymbol{x}_i - \boldsymbol{m}_0) + (\boldsymbol{x}_i - \boldsymbol{m}_0) \otimes \boldsymbol{I}_p]\Bigg), \qquad (24) \\
\frac{\partial \operatorname{vec}(\boldsymbol{V}_0)}{\partial \operatorname{vec}(\boldsymbol{S}_0)} =& \frac{p}{2nb}\sum_{i=1}^{n}\tilde{\phi}_i[(\boldsymbol{x}_i - \boldsymbol{m}_0) \otimes (\boldsymbol{x}_i - \boldsymbol{m}_0)][\boldsymbol{S}_0^{-1}(\boldsymbol{x}_i - \boldsymbol{m}_0) \otimes \boldsymbol{S}_0^{-1}(\boldsymbol{x}_i - \boldsymbol{m}_0)]^\top.
\end{aligned}$$

Most of these calculations follow from simple differentiation rules and require no further explanation. The less straight-forward calculations in (21), (22), (23), and (24) are explained in Appendix B. Lastly, we write

$$\begin{aligned}
\frac{\partial \tilde{w}}{\partial m_0} =& \frac{1}{nb}\sum_{i=1}^{n}\Bigg(-\frac{\rho_0'(\tilde{d}_i)}{\tilde{d}_i} + \frac{\rho_0'(\tilde{d}_i)}{\tilde{d}_i} + \rho_0''(\tilde{d}_i)\Bigg)(\boldsymbol{x}_i - \boldsymbol{m}_0)^\top\boldsymbol{S}_0^{-1} \\
=& \frac{1}{nb}\sum_{i=1}^{n}\rho_0''(\tilde{d}_i)(\boldsymbol{x}_i - \boldsymbol{m}_0)^\top\boldsymbol{S}_0^{-1}, \\
\frac{\partial \tilde{w}}{\partial \operatorname{vec}(\boldsymbol{S}_0)} =& \frac{1}{2nb}\sum_{i=1}\rho_0''(\tilde{d}_i)[\boldsymbol{S}_0^{-1}(\boldsymbol{x}_i - \boldsymbol{m}_0) \otimes \boldsymbol{S}_0^{-1}(\boldsymbol{x}_i - \boldsymbol{m}_0)]^\top.
\end{aligned}$$

Some partial derivatives require differentiation of the determinant of a matrix that can be calculated using Jacobi's formula:

$$|\boldsymbol{A}|' = \operatorname{tr}(\operatorname{adj}(\boldsymbol{A})\boldsymbol{A}'),$$

in which $\operatorname{adj}(\boldsymbol{A})$ denotes the adjugate matrix of $\boldsymbol{A}$. However, since we are working with differentiation over vectors, it can be more elegantly written as

$$|\boldsymbol{A}|' = \operatorname{tr}(\operatorname{adj}(\boldsymbol{A})\boldsymbol{A}') = \operatorname{tr}(|\boldsymbol{A}|\boldsymbol{A}^{-1}\boldsymbol{A}') = |\boldsymbol{A}| \operatorname{vec}(\boldsymbol{A}^{-1})^\top \operatorname{vec} \boldsymbol{A}'.$$

Using these results, the partial derivatives follow from applying simple chain rules:

$$\begin{aligned}
\frac{\partial \Psi_n(\boldsymbol{m})}{\partial \boldsymbol{m}} =& -\frac{\boldsymbol{b}}{a^2}\sum_{i=1}^n \phi_i(\boldsymbol{x}_i - \boldsymbol{m})^\top \boldsymbol{S}^{-1} + \frac{1}{a}\sum_{i=1}^n \phi_i \boldsymbol{x}_i(\boldsymbol{x}_i - \boldsymbol{m})^\top \boldsymbol{S}^{-1} - \boldsymbol{I}_p \\
=& \frac{1}{a^2}\sum_{i=1}^n \phi_i(a\boldsymbol{x}_i - \boldsymbol{b})(\boldsymbol{x}_i - \boldsymbol{m})^\top \boldsymbol{S}^{-1} - \boldsymbol{I}_p, \\
\frac{\partial \Psi_n(\boldsymbol{m})}{\partial \operatorname{vec}(\boldsymbol{S})} =& \frac{1}{2a^2}\sum_{i=1}^n \phi_i(a\boldsymbol{x}_i - \boldsymbol{b})[(\boldsymbol{x}_i - \boldsymbol{m})^\top \boldsymbol{S}^{-1} \otimes (\boldsymbol{x}_i - \boldsymbol{m})^\top \boldsymbol{S}^{-1}], \\
\frac{\partial \Psi_n(\operatorname{vec}(\boldsymbol{S}))}{\partial \boldsymbol{m}} =& |\boldsymbol{S}_0|^{\frac{1}{p}}\left(-\frac{1}{p}|\boldsymbol{V}|^{-\frac{1}{p}} \operatorname{vec}(\boldsymbol{V}) \operatorname{vec}(\boldsymbol{V}^{-1})^\top \frac{\partial \operatorname{vec}(\boldsymbol{V})}{\partial \boldsymbol{m}} + |\boldsymbol{V}|^{-\frac{1}{p}} \frac{\partial \operatorname{vec} \boldsymbol{V}}{\partial \boldsymbol{m}}\right) \\
=& |\boldsymbol{S}_0|^{\frac{1}{p}}|\boldsymbol{V}|^{-\frac{1}{p}}\left(\boldsymbol{I}_{p^2} - \frac{1}{p}\operatorname{vec}(\boldsymbol{V}) \operatorname{vec}(\boldsymbol{V}^{-1})^\top\right)\frac{\partial \operatorname{vec}(\boldsymbol{V})}{\partial \boldsymbol{m}}, \\
\frac{\partial \Psi_n(\operatorname{vec}(\boldsymbol{S}))}{\partial \operatorname{vec}(\boldsymbol{S})} =& |\boldsymbol{S}_0|^{\frac{1}{p}}|\boldsymbol{V}|^{-\frac{1}{p}}\left(\boldsymbol{I}_{p^2} - \frac{1}{p}\operatorname{vec}(\boldsymbol{V}) \operatorname{vec}(\boldsymbol{V}^{-1})^\top\right)\frac{\partial \operatorname{vec}(\boldsymbol{V})}{\partial \operatorname{vec}(\boldsymbol{S})} - \boldsymbol{I}_{p^2}, \\
\frac{\partial \Psi_n(\operatorname{vec}(\boldsymbol{S}))}{\partial \operatorname{vec}(\boldsymbol{S}_0)} =& \frac{1}{p}|\boldsymbol{S}_0|^{\frac{1}{p}}|\boldsymbol{V}|^{-\frac{1}{p}} \operatorname{vec}(\boldsymbol{V}) \operatorname{vec}(\boldsymbol{S}_0^{-1})^\top, \\
\frac{\partial \Psi_n(\operatorname{vec}(\boldsymbol{S}_0))}{\partial \operatorname{vec}(\boldsymbol{S}_0)} =& \frac{p}{2nb}\sum_{i=1}^n \tilde{\phi}_i[(\boldsymbol{x}_i - \boldsymbol{m}_0) \otimes (\boldsymbol{x}_i - \boldsymbol{m}_0)][\boldsymbol{S}_0^{-1}(\boldsymbol{x}_i - \boldsymbol{m}_0) \otimes \boldsymbol{S}_0^{-1}(\boldsymbol{x}_i - \boldsymbol{m}_0)]^\top + \\
& (\tilde{w} - 1)\boldsymbol{I}_{p^2} + \frac{1}{2nb}\operatorname{vec}(\boldsymbol{S}_0)\sum_{i=1}^n [\boldsymbol{S}_0^{-1}(\boldsymbol{x}_i - \boldsymbol{m}_0) \otimes \boldsymbol{S}_0^{-1}(\boldsymbol{x}_i - \boldsymbol{m}_0)]^\top, \\
\frac{\partial \Psi_n(\operatorname{vec}(\boldsymbol{S}_0))}{\partial \boldsymbol{m}_0} =& \frac{p}{nb}\sum_{i=1}^n \Bigg(\tilde{\phi}_i[(\boldsymbol{x}_i - \boldsymbol{m}_0) \otimes (\boldsymbol{x}_i - \boldsymbol{m}_0)](\boldsymbol{x}_i - \boldsymbol{m}_0)^\top \boldsymbol{S}_0^{-1} - \\
& \frac{\rho_0'(\tilde{d}_i)}{\tilde{d}_i}[\boldsymbol{I}_p \otimes (\boldsymbol{x}_i - \boldsymbol{m}_0) + (\boldsymbol{x}_i - \boldsymbol{m}_0) \otimes \boldsymbol{I}_p]\Bigg) + \\
& \frac{1}{nb}\operatorname{vec}(\boldsymbol{S}_0)\sum_{i=1}^n \rho_0''(\tilde{d}_i)(\boldsymbol{x}_i - \boldsymbol{m}_0)^\top \boldsymbol{S}_0^{-1}, \\
\frac{\partial \Psi_n(\boldsymbol{m}_0)}{\partial \operatorname{vec}(\boldsymbol{S}_0)} =& \frac{1}{2a_0^2}\sum_{i=1}^n \tilde{\phi}_i(a_0\boldsymbol{x}_i - \boldsymbol{b}_0)[\boldsymbol{S}_0^{-1}(\boldsymbol{x}_i - \boldsymbol{m}_0) \otimes \boldsymbol{S}_0^{-1}(\boldsymbol{x}_i - \boldsymbol{m}_0)]^\top, \\
\frac{\partial \Psi_n(\boldsymbol{m}_0)}{\partial \boldsymbol{m}_0} =& \frac{1}{a_0^2}\sum_{i=1}^n (\boldsymbol{x}_i - \boldsymbol{m}_0)^\top \boldsymbol{S}_0^{-1} - \boldsymbol{I}_p.
\end{aligned}$$

Now that all building blocks are set, we can construct the gradient matrix for the estimating equations of parameter vector $\boldsymbol{\theta}_{\mathrm{MM}} = (\boldsymbol{m}^\top, \operatorname{vec}(\boldsymbol{S})^\top, \boldsymbol{m}_0^\top, \operatorname{vec}(\boldsymbol{S}_0)^\top)^\top$ as

$$\nabla\Psi_n(\boldsymbol{\theta}_{\mathrm{MM}}) = \begin{pmatrix} \frac{\partial\Psi_n(\boldsymbol{m})}{\partial\boldsymbol{m}} & \frac{\partial\Psi_n(\boldsymbol{m})}{\partial\operatorname{vec}(\boldsymbol{S})} & \frac{\partial\Psi_n(\boldsymbol{m})}{\partial\boldsymbol{m}_0} & \frac{\partial\Psi_n(\boldsymbol{m})}{\partial\operatorname{vec}(\boldsymbol{S}_0)} \\ \frac{\partial\Psi_n(\operatorname{vec}(\boldsymbol{S}))}{\partial\boldsymbol{m}} & \frac{\partial\Psi_n(\operatorname{vec}(\boldsymbol{S}))}{\partial\operatorname{vec}(\boldsymbol{S})} & \frac{\partial\Psi_n(\operatorname{vec}(\boldsymbol{S}))}{\partial\boldsymbol{m}_0} & \frac{\partial\Psi_n(\operatorname{vec}(\boldsymbol{S}))}{\partial\operatorname{vec}(\boldsymbol{S}_0)} \\ \frac{\partial\Psi_n(\boldsymbol{m}_0)}{\partial\boldsymbol{m}} & \frac{\partial\Psi_n(\boldsymbol{m}_0)}{\partial\operatorname{vec}(\boldsymbol{S})} & \frac{\partial\Psi_n(\boldsymbol{m}_0)}{\partial\boldsymbol{m}_0} & \frac{\partial\Psi_n(\boldsymbol{m}_0)}{\partial\operatorname{vec}(\boldsymbol{S}_0)} \\ \frac{\partial\Psi_n(\operatorname{vec}(\boldsymbol{S}_0))}{\partial\boldsymbol{m}} & \frac{\partial\Psi_n(\operatorname{vec}(\boldsymbol{S}_0))}{\partial\operatorname{vec}(\boldsymbol{S})} & \frac{\partial\Psi_n(\operatorname{vec}(\boldsymbol{S}_0))}{\partial\boldsymbol{m}_0} & \frac{\partial\Psi_n(\operatorname{vec}(\boldsymbol{S}_0))}{\partial\operatorname{vec}(\boldsymbol{S}_0)} \end{pmatrix}.$$

Smoothly Weighted Bianco & Yohai estimator

The SWBY estimator follows from extending the BY estimator with a smooth weighting step. The smooth weighting step uses the multivariate MM estimator for location and scatter, so that the complete parameter vector becomes $\boldsymbol{\theta}_{\mathrm{SWBY}} = (\boldsymbol{\beta}^\top, \boldsymbol{m}^\top, \operatorname{vec}(\boldsymbol{S})^\top, \boldsymbol{m}_0^\top, \operatorname{vec}(\boldsymbol{S}_0)^\top)^\top$. The estimating equation of $\boldsymbol{\beta}$ follows simply from adding the weighting step to (15) so that we get

$$\Psi_{\mathrm{SWBY}}(\boldsymbol{\beta}) = \sum_{i=1}^{n} w_i \Psi_i(\boldsymbol{\beta}). \tag{25}$$

The complete estimating equation then follows from stacking the equation above on top of (17-20) and the corresponding gradient matrix becomes

$$\nabla\Psi_n(\boldsymbol{\theta}_{\mathrm{SWBY}}) = \begin{pmatrix} \frac{\partial\Psi_n(\boldsymbol{\beta})}{\partial\boldsymbol{\beta}} & \frac{\partial\Psi_n(\boldsymbol{\beta})}{\partial\boldsymbol{m}} & \frac{\partial\Psi_n(\boldsymbol{\beta})}{\partial\operatorname{vec}(\boldsymbol{S})} & \frac{\partial\Psi_n(\boldsymbol{\beta})}{\partial\boldsymbol{m}_0} & \frac{\partial\Psi_n(\boldsymbol{\beta})}{\partial\operatorname{vec}(\boldsymbol{S}_0)} \\ \frac{\partial\Psi_n(\boldsymbol{m})}{\partial\boldsymbol{\beta}} & \frac{\partial\Psi_n(\boldsymbol{m})}{\partial\boldsymbol{m}} & \frac{\partial\Psi_n(\boldsymbol{m})}{\partial\operatorname{vec}(\boldsymbol{S})} & \frac{\partial\Psi_n(\boldsymbol{m})}{\partial\boldsymbol{m}_0} & \frac{\partial\Psi_n(\boldsymbol{m})}{\partial\operatorname{vec}(\boldsymbol{S}_0)} \\ \frac{\partial\Psi_n(\operatorname{vec}(\boldsymbol{S}))}{\partial\boldsymbol{\beta}} & \frac{\partial\Psi_n(\operatorname{vec}(\boldsymbol{S}))}{\partial\boldsymbol{m}} & \frac{\partial\Psi_n(\operatorname{vec}(\boldsymbol{S}))}{\partial\operatorname{vec}(\boldsymbol{S})} & \frac{\partial\Psi_n(\operatorname{vec}(\boldsymbol{S}))}{\partial\boldsymbol{m}_0} & \frac{\partial\Psi_n(\operatorname{vec}(\boldsymbol{S}))}{\partial\operatorname{vec}(\boldsymbol{S}_0)} \\ \frac{\partial\Psi_n(\boldsymbol{m}_0)}{\partial\boldsymbol{\beta}} & \frac{\partial\Psi_n(\boldsymbol{m}_0)}{\partial\boldsymbol{m}} & \frac{\partial\Psi_n(\boldsymbol{m}_0)}{\partial\operatorname{vec}(\boldsymbol{S})} & \frac{\partial\Psi_n(\boldsymbol{m}_0)}{\partial\boldsymbol{m}_0} & \frac{\partial\Psi_n(\boldsymbol{m}_0)}{\partial\operatorname{vec}(\boldsymbol{S}_0)} \\ \frac{\partial\Psi_n(\operatorname{vec}(\boldsymbol{S}_0))}{\partial\boldsymbol{\beta}} & \frac{\partial\Psi_n(\operatorname{vec}(\boldsymbol{S}_0))}{\partial\boldsymbol{m}} & \frac{\partial\Psi_n(\operatorname{vec}(\boldsymbol{S}_0))}{\partial\operatorname{vec}(\boldsymbol{S})} & \frac{\partial\Psi_n(\operatorname{vec}(\boldsymbol{S}_0))}{\partial\boldsymbol{m}_0} & \frac{\partial\Psi_n(\operatorname{vec}(\boldsymbol{S}_0))}{\partial\operatorname{vec}(\boldsymbol{S}_0)} \end{pmatrix}.$$

Analogously to (25), the upper-left element of this matrix is computed as a weighted representation of (16). Next, the estimating equations of the MM estimator do not depend on the regression coefficients and hence

$$\frac{\partial\Psi_n(\boldsymbol{m})}{\partial\boldsymbol{\beta}} = \frac{\partial\Psi_n(\boldsymbol{m}_0)}{\partial\boldsymbol{\beta}} = \boldsymbol{O}_{p\times(p+1)} \quad \text{and} \quad \frac{\partial\Psi_n(\operatorname{vec}(\boldsymbol{S}))}{\partial\boldsymbol{\beta}} = \frac{\partial\Psi_n(\operatorname{vec}(\boldsymbol{S}_0))}{\partial\boldsymbol{\beta}} = \boldsymbol{O}_{p^2\times(p+1)}.$$

The same holds for

$$\frac{\partial \Psi_n(\boldsymbol{\beta})}{\partial \boldsymbol{m}_0} = \boldsymbol{O}_{(p+1)\times p} \qquad \text{and} \qquad \frac{\partial \Psi_n(\boldsymbol{\beta})}{\partial \operatorname{vec}(\boldsymbol{S}_0)} = \boldsymbol{O}_{(p+1)\times p^2},$$

because the weighting step uses the efficient MM estimates and not the preliminary S estimates. The cross-terms that remain are $\frac{\partial \Psi_n(\boldsymbol{\beta})}{\partial \boldsymbol{m}}$ and $\frac{\partial \Psi_n(\boldsymbol{\beta}}{\partial \operatorname{vec}(\boldsymbol{S})}$ and they can be calculated as

$$\frac{\partial \Psi_n(\boldsymbol{\beta})}{\partial \boldsymbol{m}} = \sum_{i=1}^{n} \Psi_n(\boldsymbol{\beta}) \frac{\partial w_i}{\partial RD_i^2} \frac{\partial RD_i^2}{\partial \boldsymbol{m}} \qquad \text{and} \qquad \frac{\partial \Psi_n(\boldsymbol{\beta})}{\partial \operatorname{vec}(\boldsymbol{S})} = \sum_{i=1}^{n} \Psi_n(\boldsymbol{\beta}) \frac{\partial w_i}{\partial RD_i^2} \frac{\partial RD_i^2}{\partial \operatorname{vec}(\boldsymbol{S})},$$

of which only the partial derivatives of the squared Mahalanobis distances are yet unknown. Their derivations resemble those of the non-squared alternative given in Appendix B and they are calculated as

$$\frac{\partial RD_i^2}{\partial \boldsymbol{m}} = -2(\boldsymbol{x}_i - \boldsymbol{m})^\top \boldsymbol{\Sigma}^{-1} \qquad \text{and} \qquad \frac{\partial RD_i^2}{\partial \operatorname{vec}(\boldsymbol{S})} = -[\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{m}) \otimes \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{m})]^\top.$$

2.6 Confidence Intervals

Two procedures for estimating confidence intervals are considered. In the first, the estimator and its standard deviation are calculated directly using the asymptotic variance. Asymptotic confidence intervals are then constructed based on a normal distribution around the estimated coefficients. That is, a $100q\%$- confidence interval for regression coefficient β_j is computed as

$$\mathrm{ACI}_j = [\hat{\beta}_{n,j} - z_{\frac{1-q}{2}} \hat{\sigma}(\beta_{n,j}), \hat{\beta}_{n,j} + z_{1-\frac{1-q}{2}} \hat{\sigma}(\beta_{n,j})], \tag{26}$$

in which z_α represents the $100\alpha\%$ critical value of the standard normal distribution and $\hat{\beta}_{n,j}$ and $\hat{\sigma}(\beta_{n,j})$ represent the estimate and standard deviation of the jth regression coefficient calculated using (10). The asymptotic standard deviations of the SBY and SWBY estimators are calculated with the same logic as those of the BY and WBY estimators, which is described in (8-10), but with the weighting functions and the first and second derivatives of the objective functions replaced by the corresponding smooth alternatives. In the second procedure, the Fast and Robust Bootstrap is applied to obtain B bootstrap estimates. These estimates can be seen as an empirical distribution on itself and from that we can derive confidence intervals. Lower and upper bounds of the confidence intervals are chosen to be equal to the $100\frac{1-q}{2}$th and the $100(1-\frac{1-q}{2})$th percentile, respectively.

3 Simulation

This section describes a Monte Carlo simulation study that is designed for assessing the performance of the proposed bootstrapping procedures in combination with the SBY and the SWBY estimator. First, the general data generating process (DGP) is described, where I have chosen for a close resemblance to the DGP that is used in the original work by Croux and Haesbroeck (2003). Next, the methods for obtaining confidence intervals are introduced, followed by a description of the conducted experiment.

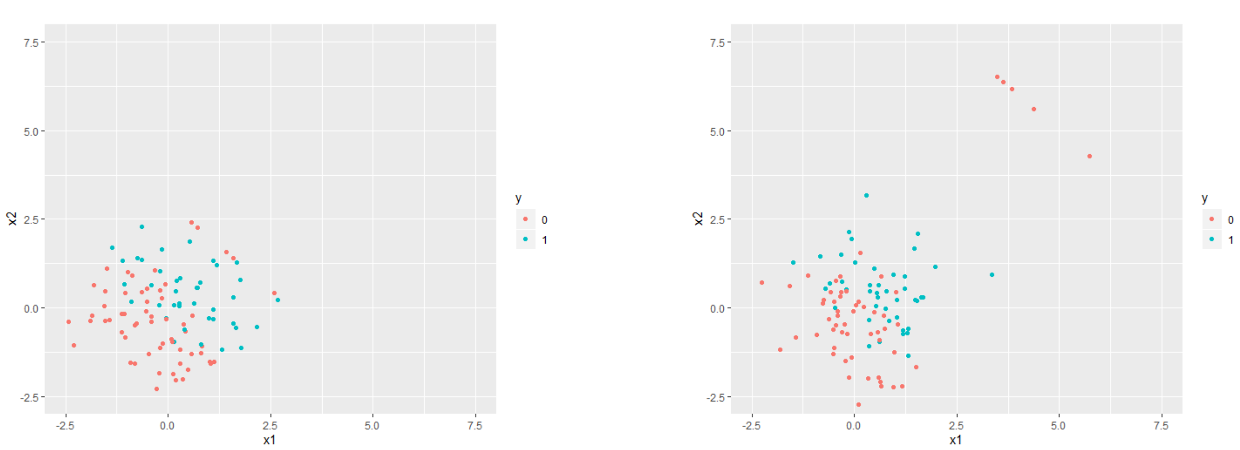
3.1 Data Generating Process

Consider a sample of n observations. The explanatory variables for observation i are $\boldsymbol{x}_i = (1, \tilde{\boldsymbol{x}}_i^\top)^\top$ of which the variable part is distributed as $\tilde{\boldsymbol{x}}_i \sim N(\mathbf{0}, \boldsymbol{I}_p)$. The regression coefficients are denoted as $\boldsymbol{\beta} \in \mathbb{R}^{p+1}$. Next, dependent variable y_i is drawn from a Bernouilli distribution with probability

$$\Pr[Y_i = 1|\boldsymbol{x}_i] = \frac{1}{1 + \exp(-\boldsymbol{x}_i^\top \boldsymbol{\beta})},$$

such that the decision hyperplane is given by $\boldsymbol{x}^\top \boldsymbol{\beta} = 0$. Then, in the case of contamination, a fraction ϵ of the sample is replaced by outliers. The outliers are generated as bad leverage points in the covariate space on a contamination hyperplane parallel to the decision hyperplane. The distance between the decision hyperplane and the contamination hyperplane is denoted by $\delta\sqrt{p}$. Illustrations of the the datasets for $n = 100$, $\boldsymbol{\beta} = (0, 2, 2)^\top$ and $\epsilon = 0.05$ are shown in Figure 3. The left panel shows a dataset with distance parameter $\delta = 1.5$ and the right panel shows a dataset with distance parameter $\delta = 5$. At first glance it can be assumed that the SWBY estimator will outperform the SBY estimator on the dataset with $\delta = 5$, because the bad leverage points can easily be filtered out. However, this advantage vanishes on the dataset with $\delta = 1.5$, because the outliers here are less outlying in the covariate space and hence less likely to be filtered out by the weighting step.

Datasets are generated with different settings to get a broad view of the method's performance. First, we set $n = 100$ and $\boldsymbol{\beta} = (0, 2, 2)^\top$ to assess the method's performance on datasets with the same characteristics as the datasets used by Croux and Haesbroeck (2003). Next, we apply the procedures to a smaller sample size of $n = 20$ with again $\boldsymbol{\beta} = (0, 2, 2)^\top$.



(a) $\delta = 1.5$

(b) $\delta = 5$

Figure 3: Example datasets of $n = 100$ observations, with true parameter $\beta = (0, 2, 2)^\top$ and 5% contamination.

Finally, we evaluate the performance on higher dimensional data by taking $n = 100$ and $\boldsymbol{\beta} = (1, \ldots, 1)^\top/3\sqrt{11} \in \mathbb{R}^{11}$. For each size and dimensionality, three types of contamination are introduced: no contamination, 5% contamination with $\delta = 1.5$, and 5% contamination with $\delta = 5$. Datasets with these types of contamination will be referred to as clean datasets, type I datasets, and type II datasets, respectively.

3.2 Experiment

For this simulation study, $R = 1000$ datasets are generated for each configuration as described in Section 3.1. On each dataset, the estimates and corresponding asymptotic variances are calculated, which can then be used to compute asymptotic confidence intervals. To get an insight whether the proposed smoothness adjustments to the original estimators have an impact on their performance, both the original and the smooth estimators are used. Next, bootstrap confidence intervals are constructed for the smooth estimators, where the number of bootstrap replications that is used is $B = 2000$. The horizon parameter h of the smooth weighting function is set to 1. For each confidence interval, an indicator function checking whether or not the interval contains the true coefficient value is evaluated, as well as the interval length. Coverage of the regression coefficient β_j is then calculated as

$$\widehat{\text{Coverage}}_j = \frac{1}{R}\sum_{r=1}^{R} \mathbb{I}[\beta_j \in \text{CI}_{j,r}],$$

in which β_j denotes the true coefficient for the jth regressor and $\text{CI}_{j,r}$ is the computed confidence interval for the coefficient for the jth regressor in replication r. The target coverage is 90% in each setting. The mean absolute error (MAE) of the computed coverage compared to the target coverage is the key performance indicator here. This is calculated using the formula

$$\text{MAE} = \frac{1}{p+1}\sum_{j=0}^{p} |\text{Coverage}_j - \text{Coverage}^*|,$$

in which Coverage* denotes the target coverage. The mean interval length is reported as well. Situations may occur in which the algorithm does not converge or does not yield finite standard deviations for the regression coefficients. This mainly happens when the number of observations is small or the number of regressors is relatively large, because the risk of having no overlap is larger under these circumstances. When this occurs, the dataset is dropped and a new dataset is generated.

4 Results

4.1 Smooth Bianco & Yohai estimator

4.1.1 Sample size of 100 and 2 regressors

The results of the experiment on the datasets with $n = 100$ observations and $p = 2$ regressors are shown in Table 1. The results of the intervals obtained with the asymptotic variance are denoted by ACI, and the intervals obtained with the bootstrap estimates are denoted by BCI. For the original BY estimator, no legitimate FRB procedure can be derived, hence the BCIs are only computed for the SBY estimator. In each setting, the ACIs for both estimators yield similar results in terms of coverage and mean interval length, implying that the proposed adjustments in combination with the chosen configurations do not have a substantial effect on the quality of the confidence intervals. The BCIs are slightly further off target in each setting, with the difference being larger on the contaminated datasets than on the clean datasets. On the clean data, the coverages of the BCIs have an MAE of 2.9 percentage points, which is only 0.3 percentage points higher than the coverages of the ACIs. On the type I datasets, the MAE of the BCIs is 30.6 percentage points, which is 1.4 percentage points higher than the coverages of the ACIs. On the type II datasets, the MAE of the BCIs is 7.4 percentage points, which is 1.6 percentage points higher than the coverages of the ACIs.

4.1.2 Sample size of 20 and 2 regressors

The results of the experiment on the datasets with $n = 20$ observations and $p = 2$ regressors are shown in Table 2. In each setting, the differences between the coverages and between the mean interval lengths of the ACIs of both estimators are negligible. Again, the BCIs are slightly further off target, with the difference being larger on the contaminated datasets. On the clean data, the MAE of the BCIs is 2.7 percentage points, which is 0.5 percentage points higher than obtained with the ACIs. On the type I data, the MAE of the coverages of the BCI is 9.4 percentage points, which is 1.7 percentage points higher than obtained with the ACIs. On the type II data, the MAE of the coverages obtained using the BCIs is 14.7

		BY	SBY	
Contam.	Coeff.	ACI	ACI	BCI
Clean	β_0	0.917	0.917	0.921
	β_1	0.940	0.940	0.945
	β_2	0.920	0.920	0.920
MAE		**0.026**	**0.026**	**0.029**
I	β_0	0.853	0.855	0.859
	β_1	0.486	0.484	0.461
	β_2	0.488	0.486	0.461
MAE		**0.291**	**0.292**	**0.306**
II	β_0	0.909	0.910	0.919
	β_1	0.825	0.826	0.803
	β_2	0.809	0.809	0.795
MAE		**0.058**	**0.058**	**0.074**

		BY	SBY	
Contam.	Coeff.	ACI	ACI	BCI
Clean	β_0	1.08	1.08	1.08
	β_1	1.86	1.86	1.86
	β_2	1.86	1.86	1.86
Mean		**1.60**	**1.60**	**1.60**
I	β_0	0.80	0.80	0.80
	β_1	1.66	1.66	1.66
	β_2	1.65	1.65	1.65
Mean		**1.37**	**1.37**	**1.37**
II	β_0	0.91	0.91	0.91
	β_1	2.04	2.04	2.04
	β_2	2.03	2.03	2.03
Mean		**1.66**	**1.66**	**1.66**

Table 1: The left panel shows an overview of the coverage obtained by the different confidence intervals for different types of contamination, with a target coverage of 0.9. The mean interval lengths are reported in the right panel. The number of observations in each dataset is 100.

		BY	SBY	
Contam.	Coeff.	ACI	ACI	BCI
Clean	β_0	0.960	0.960	0.969
	β_1	0.906	0.906	0.906
	β_2	0.899	0.899	0.906
MAE		**0.022**	**0.022**	**0.027**
I	β_0	0.952	0.952	0.964
	β_1	0.822	0.825	0.804
	β_2	0.799	0.797	0.778
MAE		**0.077**	**0.077**	**0.094**
II	β_0	0.951	0.953	0.966
	β_1	0.716	0.716	0.706
	β_2	0.727	0.725	0.719
MAE		**0.136**	**0.137**	**0.147**

		BY	SBY	
Contam.	Coeff.	ACI	ACI	BCI
Clean	β_0	4.46	4.47	4.43
	β_1	9.08	9.15	9.01
	β_2	8.80	8.83	8.70
Mean		**7.45**	**7.52**	**7.38**
I	β_0	3.35	3.33	3.32
	β_1	7.04	7.03	6.96
	β_2	6.51	6.48	6.41
Mean		**5.63**	**5.61**	**5.56**
II	β_0	2.64	2.64	2.63
	β_1	4.63	4.62	4.58
	β_2	4.83	4.81	4.77
Mean		**3.37**	**3.36**	**3.33**

Table 2: The left panel shows an overview of the coverage obtained by the different confidence intervals for different types of contamination, with a target coverage of 0.9. The mean interval lengths are reported in the right panel. The number of observations in each dataset is 20.

percentage points, which is 1.0 percentage points and 1.1 percentage points higher than the MAEs of the coverages obtained using the ACIs of the SBY estimator and the BY estimator, respectively.

4.1.3 Sample size of 100 and 10 regressors

The results for the datasets with $n = 100$ observations and $p = 10$ regressors are shown in Table 3. For all three types of contamination, the differences between the performances in terms of coverage of the three methods are negligible. The largest difference is visible on the clean datasets, where the BCIs yield a coverage that is on average 0.9 percentage points off target, which is 0.3 percentage points more than the coverages obtained with the ACIs. The comparability of the results of the three methods is confirmed by the mean interval lengths, which are on average equal up to two decimals on all three types of datasets. These results show that the BCIs constructed using the proposed bootstrapping method are not inferior to the ACIs on these particular settings.

		BY	SBY	
Contam.	Coeff.	ACI	ACI	BCI
Clean	β_0	0.907	0.907	0.912
	β_1	0.916	0.918	0.923
	β_2	0.891	0.889	0.889
	β_3	0.913	0.910	0.919
	β_4	0.907	0.907	0.914
	β_5	0.898	0.900	0.903
	β_6	0.895	0.896	0.900
	β_7	0.900	0.902	0.907
	β_8	0.903	0.901	0.904
	β_9	0.893	0.893	0.904
	β_{10}	0.901	0.900	0.906
MAE		**0.006**	**0.006**	**0.009**
I	β_0	0.880	0.881	0.885
	β_1	0.861	0.863	0.863
	β_2	0.864	0.862	0.865
	β_3	0.858	0.859	0.866
	β_4	0.851	0.852	0.852
	β_5	0.864	0.868	0.862
	β_6	0.854	0.850	0.848
	β_7	0.829	0.828	0.827
	β_8	0.851	0.848	0.850
	β_9	0.851	0.850	0.852
	β_{10}	0.821	0.819	0.829
MAE		**0.047**	**0.047**	**0.046**
II	β_0	0.918	0.915	0.914
	β_1	0.843	0.844	0.847
	β_2	0.851	0.851	0.846
	β_3	0.840	0.841	0.840
	β_4	0.825	0.828	0.826
	β_5	0.827	0.823	0.823
	β_6	0.834	0.835	0.833
	β_7	0.862	0.863	0.860
	β_8	0.863	0.861	0.857
	β_9	0.819	0.821	0.819
	β_{10}	0.804	0.806	0.823
MAE		**0.059**	**0.058**	**0.058**

		BY	SBY	
Contam.	Coeff.	ACI	ACI	BCI
Clean	β_0	0.80	0.80	0.80
	β_1	0.81	0.81	0.81
	β_2	0.82	0.82	0.82
	β_3	0.82	0.82	0.82
	β_4	0.81	0.81	0.81
	β_5	0.82	0.82	0.82
	β_6	0.82	0.82	0.82
	β_7	0.82	0.82	0.82
	β_8	0.82	0.82	0.82
	β_9	0.83	0.83	0.83
	β_{10}	0.82	0.82	0.82
Mean		**0.82**	**0.82**	**0.82**
I	β_0	0.79	0.79	0.79
	β_1	0.78	0.78	0.78
	β_2	0.78	0.78	0.78
	β_3	0.77	0.77	0.78
	β_4	0.78	0.78	0.78
	β_5	0.78	0.78	0.78
	β_6	0.78	0.78	0.78
	β_7	0.78	0.78	0.78
	β_8	0.78	0.78	0.78
	β_9	0.78	0.78	0.78
	β_{10}	0.68	0.68	0.68
Mean		**0.77**	**0.77**	**0.77**
II	β_0	0.80	0.80	0.81
	β_1	0.77	0.77	0.77
	β_2	0.77	0.77	0.77
	β_3	0.77	0.77	0.77
	β_4	0.77	0.77	0.77
	β_5	0.77	0.77	0.77
	β_6	0.77	0.77	0.77
	β_7	0.77	0.77	0.77
	β_8	0.77	0.77	0.77
	β_9	0.77	0.77	0.77
	β_{10}	0.67	0.67	0.67
Mean		**0.76**	**0.76**	**0.76**

Table 3: The left panel shows an overview of the coverage obtained by the different confidence intervals for different types of contamination, with a target coverage of 0.9. The mean interval lengths are reported in the right panel. The number of observations in each dataset is 100.

4.2 Smooth weighted Bianco & Yohai estimator

4.2.1 Sample size of 100 and 2 regressors

The results of the experiment on the datasets with $n = 100$ observations and $p = 2$ regressors are shown in Table 4. On the clean datasets and the type I datasets, the coverages obtained with the BCIs are again slightly further off target than the coverages obtained with the ACIs. However, the differences on the type I datasets are smaller than what we observed for the non-weighted estimators in Section 4.1. Moreover, on the type II datasets, the differences between the coverages and interval lengths are negligible.

4.2.2 Sample size of 20 and 2 regressors

The results of the experiment on the datasets with $n = 20$ observations and $p = 2$ regressors are shown in Table 5. In each setting, the coverages obtained with the BCIs are slightly further off target than the coverages obtained with the ACIs. In addition, the mean lengths of the confidence intervals are longer. The differences between the methods are smaller on the contaminated datasets. Where the differences between the MAEs of the BCI coverages versus the ACI coverages are 1.1-1.2 percentage points on the clean data, they are only 0.4-0.7 percentage points on the type I data and only 0.1-0.6 percentage points on the type II data.

		WBY	SWBY	
Contam.	Coeff.	ACI	ACI	BCI
Clean	β_0	0.917	0.917	0.928
	β_1	0.939	0.938	0.943
	β_2	0.920	0.922	0.919
MAE		**0.025**	**0.026**	**0.030**
I	β_0	0.877	0.876	0.880
	β_1	0.614	0.610	0.599
	β_2	0.606	0.603	0.595
MAE		**0.201**	**0.204**	**0.209**
II	β_0	0.916	0.915	0.917
	β_1	0.937	0.936	0.934
	β_2	0.942	0.941	0.942
MAE		**0.032**	**0.031**	**0.031**

		WBY	SWBY	
Contam.	Coeff.	ACI	ACI	BCI
Clean	β_0	1.08	1.08	1.09
	β_1	1.90	1.90	1.91
	β_2	1.89	1.89	1.90
Mean		**1.62**	**1.62**	**1.63**
I	β_0	0.84	0.84	0.84
	β_1	1.69	1.69	1.76
	β_2	1.69	1.69	1.76
Mean		**1.41**	**1.41**	**1.45**
II	β_0	1.11	1.11	1.11
	β_1	1.88	1.88	1.89
	β_2	1.88	1.89	1.89
Mean		**1.62**	**1.63**	**1.63**

Table 4: The left panel shows an overview of the coverage obtained by the different confidence intervals for different types of contamination, with a target coverage of 0.9. The mean interval lengths are reported in the right panel. The number of observations in each dataset is 100.

		WBY	SWBY	
Contam.	Coeff.	ACI	ACI	BCI
Clean	β_0	0.959	0.956	0.968
	β_1	0.908	0.910	0.918
	β_2	0.903	0.900	0.917
MAE		**0.023**	**0.022**	**0.034**
I	β_0	0.950	0.948	0.959
	β_1	0.856	0.852	0.852
	β_2	0.866	0.858	0.858
MAE		**0.043**	**0.046**	**0.050**
II	β_0	0.946	0.951	0.962
	β_1	0.901	0.902	0.899
	β_2	0.908	0.916	0.910
MAE		**0.018**	**0.023**	**0.024**

		WBY	SWBY	
Contam.	Coeff.	ACI	ACI	BCI
Clean	β_0	4.47	4.45	5.00
	β_1	8.89	9.02	10.67
	β_2	8.69	8.73	9.56
Mean		**7.35**	**7.40**	**8.41**
I	β_0	3.56	3.43	4.02
	β_1	7.55	7.16	9.99
	β_2	7.51	7.11	9.26
Mean		**6.21**	**5.90**	**7.76**
II	β_0	3.64	3.56	3.82
	β_1	5.45	5.27	5.79
	β_2	5.65	5.49	6.75
Mean		**4.91**	**4.77**	**5.45**

Table 5: The left panel shows an overview of the coverage obtained by the different confidence intervals for different types of contamination, with a target coverage of 0.9. The mean interval lengths are reported in the right panel. The number of observations in each dataset is 20.

4.2.3 Sample size of 100 and 10 regressors

The results of the experiment on the datasets with $n = 100$ observations and $p = 10$ regressors are shown in Table 6. The differences between the three methods in terms of coverages and interval lengths are larger in this experiment compared to the previous ones. On the clean data, the MAE of the BCI coverages is 3.6 percentage points, which is 2.6-2.7 percentage points larger than the MAEs of the ACI coverages. The results on the contaminated datasets show a similar pattern: on the type I and type II datasets, the MAE of the BCI coverages are 3.9 percentage points and 2.7 percentage points, being respectively 3.1-3.2 percentage points and 2.1-2.2 percentage points larger than those of the respective ACI coverages. These larger differences are reflected by the differenecs in interval lengths. In all three settings, the mean interval lengths of the BCIs are 2.5 to 4 times as large as the mean interval lengths of the ACIs.

		WBY	SWBY	
Contam.	Coeff.	ACI	ACI	BCI
Clean	β_0	0.924	0.912	0.939
	β_1	0.928	0.920	0.944
	β_2	0.886	0.886	0.924
	β_3	0.911	0.915	0.939
	β_4	0.904	0.913	0.932
	β_5	0.896	0.900	0.932
	β_6	0.885	0.887	0.925
	β_7	0.903	0.899	0.937
	β_8	0.900	0.902	0.939
	β_9	0.907	0.908	0.945
	β_{10}	0.897	0.899	0.943
MAE		**0.010**	**0.009**	**0.036**
I	β_0	0.901	0.898	0.937
	β_1	0.901	0.906	0.941
	β_2	0.913	0.916	0.949
	β_3	0.896	0.898	0.938
	β_4	0.902	0.895	0.934
	β_5	0.907	0.910	0.945
	β_6	0.908	0.903	0.940
	β_7	0.887	0.880	0.935
	β_8	0.882	0.887	0.935
	β_9	0.897	0.896	0.932
	β_{10}	0.913	0.900	0.941
MAE		**0.008**	**0.007**	**0.039**
II	β_0	0.913	0.913	0.933
	β_1	0.905	0.900	0.921
	β_2	0.893	0.890	0.926
	β_3	0.900	0.900	0.927
	β_4	0.896	0.895	0.921
	β_5	0.910	0.903	0.934
	β_6	0.911	0.909	0.931
	β_7	0.909	0.908	0.927
	β_8	0.901	0.902	0.926
	β_9	0.903	0.901	0.925
	β_{10}	0.895	0.896	0.924
MAE		**0.006**	**0.005**	**0.027**

		WBY	SWBY	
Contam.	Coeff.	ACI	ACI	BCI
Clean	β_0	0.84	0.82	2.04
	β_1	0.88	0.85	2.83
	β_2	0.88	0.85	2.84
	β_3	0.89	0.86	2.20
	β_4	0.88	0.85	2.04
	β_5	0.90	0.87	2.96
	β_6	0.88	0.85	2.45
	β_7	0.88	0.85	2.37
	β_8	0.89	0.86	2.24
	β_9	0.90	0.87	2.37
	β_{10}	0.89	0.86	2.92
Mean		**0.88**	**0.85**	**2.48**
I	β_0	0.87	0.83	2.62
	β_1	0.90	0.86	2.86
	β_2	0.92	0.87	4.04
	β_3	0.91	0.86	2.72
	β_4	0.91	0.86	3.49
	β_5	0.91	0.87	4.77
	β_6	0.91	0.87	3.14
	β_7	0.91	0.87	3.17
	β_8	0.91	0.87	3.92
	β_9	0.91	0.86	2.77
	β_{10}	0.91	0.86	3.71
Mean		**0.91**	**0.86**	**3.38**
II	β_0	0.87	0.85	1.34
	β_1	0.91	0.88	2.08
	β_2	0.91	0.88	1.81
	β_3	0.92	0.88	1.51
	β_4	0.91	0.88	1.92
	β_5	0.91	0.88	2.40
	β_6	0.91	0.87	2.38
	β_7	0.91	0.88	2.00
	β_8	0.92	0.88	2.68
	β_9	0.91	0.88	1.66
	β_{10}	0.91	0.88	2.26
Mean		**0.91**	**0.88**	**2.00**

Table 6: The left panel shows an overview of the coverage obtained by the different confidence intervals for different types of contamination, with a target coverage of 0.9. The mean interval lengths are reported in the right panel. The number of observations in each dataset is 100.

5 Conclusion

The FRB procedure provides a framework for obtaining robust bootstrapping estimates in a fast way. Because the procedure uses a Taylor expansion, smoothness of the estimating equation is an important requirement for the method's feasibility. The estimating equations of the Bianco & Yohai estimator lack smoothness due to the piecewise residual weighting function that is used. In addition, the weighted Bianco & Yohai estimator uses a binomial weighting function and the MCD estimator, which both also have non-smooth characteristics. Replacing the weighting functions with smooth approximations and the MCD estimator with the multivariate MM estimator for location and scatter results in the smooth (weighted) Bianco & Yohai estimator and offers perspective for a study about the feasibility of an FRB procedure for these estimators. The performance of the bootstrapping procedure is measured in terms of coverage and compared to the coverages of the asymptotic confidence intervals of both the smooth and non-smooth (weighted) Bianco & Yohai estimators on different types of datasets with different types of contamination. These comparisons show that the differences between the performances of the smooth and non-smooth versions of both the weighted and non-weighted Bianco & Yohai estimator are negligible. Furthermore, performances of the bootstrap confidence intervals obtained using the smooth Bianco & Yohai estimator are not far behind on each dataset. For the smooth weighted Bianco & Yohai estimator, a similar result holds, with the exception that the performances using the bootstrap confidence intervals on the high-dimensional datasets are notably inferior than using the asymptotic confidence intervals.

6 Limitations and future research

The simulation study provides some insights in the performance of the proposed methods. However, a more extensive simulation study is necessary to get a more complete view. For example, the chosen sample sizes are small, alternating between 20 and 100 observations and between 2 and 10 regressors, excluding constants, whereas real-life datasets are often a lot larger. It would be interesting to evaluate the performance on larger datasets and with more regressors. Moreover, the results show a deterioration in performance of the smooth weighted BY estimator in the setting with 10 regressors, but I did not yet discover the reason for this deterioration.

Another area of improvement is the construction of bootstrap confidence intervals. The method for obtaining bootstrap confidence intervals is centered in terms of percentiles. However, it can be assumed that when outliers are asymmetrically present in the data, tails of the empirical distribution also show more disturbance on one side than on the other. Therefore, more accurate confidence intervals can be obtained by using a method that corrects for this, especially since the outliers introduced in the simulation study are generated on one side of the decision hyperplane. An example of such a method is the bias corrected accelerated bootstrap by Efron (1987). In addition, bootstrap confidence intervals could be constructed by replacing the asymptotic standard deviation in (26) by the standard deviation calculated on the bootstrap sample. By using the same method of interval construction, differences in coverages achieved with asymptotic theory and fast and robust bootstrapping could be fully attributed to the quality of the fast and robust bootstrap and not partially to the difference in interval construction, as is the case now.

The proposed adjustments to the original (W)BY estimator are chosen in such a way that the requirements for deriving a fast and robust bootstrapping procedure are met, but this does not necessarily mean that these are the only or the optimal choices. A more elaborate study can be designed to find optimal tuning parameters for the smooth weighting functions. For instance, the smooth approximation to the residual weighting function uses tuning parameter c_1, which is by default set to $0.9c$. The results show no large difference between the non-smooth methods and the smooth methods using this setting. However, an additional study could be designed to optimize the choice of c_1. In addition, the smooth weighting function for the SWBY estimator as defined in (12) may not even be necessary. The idea of this step is to mimic the indicator function in a smooth way. That is, the robust Mahalanobis distances are mapped between 0 and 1 by a function that is differentiable on its domain. However, the MM estimators of locations and scatter that are used for calculating the Mahalanobis distances already assign weights to the observations between 0 and 1. This weighting would show less resemblance to the indicator function than (12), but it could be used to simplify the procedure and to extend the comparison of methods.

Finally, the weighting steps of both the original WBY estimator and the SWBY estimator are based on location and scatter estimates that are designed for continuous regressors. To expand the applicability of the (S)WBY estimator, it is interesting to see if an estimator of location and scatter can be used that can correctly handle dummy regressors.

Appendix

A Determining the parameters for the smooth ρ-function

The smooth residual weighting function proposed in Section 2.3 is defined as

$$\rho_{\text{SBY}}(r) = \begin{cases} \alpha_1 r & \text{if } |r| \leq c_1, \\ \alpha_2 r^3 + \alpha_3 r^2 + \alpha_4 r + \alpha_5 & \text{if } c_1 < |r| \leq c_2, \\ -2\exp(-\sqrt{r})(1+\sqrt{r}) + \exp(-\sqrt{c})(2(1+\sqrt{c})+c) & \text{otherwise,} \end{cases}$$

with first derivative

$$\rho'_{\text{SBY}}(r) = \begin{cases} r & \text{if } |r| \leq c_1, \\ 3\alpha_2 r^2 + 2\alpha_3 r + \alpha_4 & \text{if } c_1 < |r| \leq c_2, \\ \exp(-\sqrt{r}) & \text{otherwise,} \end{cases}$$

and second derivative

$$\rho''_{\text{SBY}}(r) = \begin{cases} 0 & \text{if } |r| \leq c_1, \\ 6\alpha_2 r + 2\alpha_3 & \text{if } c_1 < |r| \leq c_2, \\ -\frac{1}{2\sqrt{r}}\exp(-\sqrt{r}) & \text{otherwise.} \end{cases}$$

The goal is to find expressions for $\alpha_1, \alpha_2, \alpha_3, \alpha_4$, and α_5 as functions of c, c_1, and c_2. First, we know that

$$\lim_{r\downarrow c_1} \rho''_{\text{SBY}}(r) = \lim_{r\uparrow c_1} \rho''_{\text{SBY}}(r) \qquad \text{so that} \qquad 6\alpha_2 c_1 + 2\alpha_3 = 0$$

and similarly

$$\lim_{r\uparrow c_2} \rho''_{\text{SBY}}(r) = \lim_{r\downarrow c_2} \rho''_{\text{SBY}}(r) \qquad \text{so that} \qquad 6\alpha_2 c_2 + 2\alpha_3 = -\frac{1}{2\sqrt{c_2}}\exp(-\sqrt{c_2}).$$

Combining the two previous results yields

$$6\alpha_2(c_2 - c_1) = -\frac{1}{2\sqrt{c_2}}\exp\left(-\sqrt{c_2}\right) \qquad \rightarrow \qquad \alpha_2 = \frac{1}{12(c_2 - c_1)\sqrt{c_2}}\exp\left(-\sqrt{c_2}\right).$$

An expression for α_3 is then computed as

$$\alpha_3 = -\frac{6\alpha_2 c_1}{2} = -3\alpha_2 c_1.$$

Tedious substitutions are left out of this derivation, because we let RStudio do the computations. Next, we know that

$$\lim_{r\uparrow c_2}\rho'_{\mathrm{SBY}}(r) = \lim_{r\downarrow c_2}\rho'_{\mathrm{SBY}}(r) \qquad \text{so that} \qquad \alpha_4 = \exp\left(-\sqrt{c_2}\right) - 3\alpha_2 c_2^2 - 2\alpha_3 c_2,$$

and consequently

$$\lim_{r\downarrow c_1}\rho_{\mathrm{SBY}}(r) = \lim_{r\uparrow c_1}\rho_{\mathrm{SBY}}(r) \qquad \text{so that} \qquad \alpha_5 = \alpha_1 c_1 - \alpha_2 c_1^3 - \alpha_3 c_1^2 - \alpha_4 c_1.$$

For each expression above, define residuals $R_j(c_1, c_2)$ for $\alpha_j, j = 1, \ldots, 4$ as the differences between the left hand side and the right hand side and define objective function

$$O(c_1, c_2) = \sum_{j=1}^{4} R_j^2.$$

Now we can find c_2 from c_1 by minimizing the objective function while keeping c_1 constant. This can be done in RStudio using the `optimize` function.

B Derivation of partial derivatives

B.1 Differentiation of Mahalanobis distance

In (21) and (23) the Mahalanobis distance is differentiated with respect to location and (vectorized) scatter. The Mahalanobis distance of observation $\boldsymbol{x}$ given location $\boldsymbol{\mu}$ and scatter

$\boldsymbol{\Sigma}$ is given by

$$d = [(\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})]^{\frac{1}{2}}.$$

Its differentiation with respect to $\boldsymbol{\mu}$ is quite straightforward:

$$\frac{\partial d}{\partial \boldsymbol{\mu}} = \frac{1}{2}(-2)(\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} [(\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})]^{-\frac{1}{2}} = -\frac{1}{d}(\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}.$$

The differentiation with respect to $\operatorname{vec}(\boldsymbol{\Sigma})$ requires differentiation of an inverse and some vectorization. The derivative of the inverse can be found using an identity matrix decomposition:

$$\begin{aligned} \boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1} =& \boldsymbol{I}, \\ \boldsymbol{\Sigma}(\boldsymbol{\Sigma}^{-1})' + \boldsymbol{\Sigma}'\boldsymbol{\Sigma}^{-1} =& \boldsymbol{O}, \\ \boldsymbol{\Sigma}(\boldsymbol{\Sigma}^{-1})' =& -\boldsymbol{\Sigma}'\boldsymbol{\Sigma}^{-1}, \\ (\boldsymbol{\Sigma}^{-1})' =& -\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}'\boldsymbol{\Sigma}^{-1}. \end{aligned}$$

Next, we differentiate d with respect to a single element of $\boldsymbol{\Sigma}$:

$$\begin{aligned} \frac{\partial d}{\partial \boldsymbol{\Sigma}_{jk}} =& \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^\top \frac{\partial \boldsymbol{\Sigma}^{-1}}{\partial \boldsymbol{\Sigma}_{jk}} (\boldsymbol{x} - \boldsymbol{\mu}) [(\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})]^{-\frac{1}{2}} \\ =& -\frac{1}{2d}(\boldsymbol{x} - \boldsymbol{m})^\top \boldsymbol{\Sigma}^{-1} \frac{\partial \boldsymbol{\Sigma}}{\partial \boldsymbol{\Sigma}_{jk}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}). \end{aligned} \tag{27}$$

The middle part of this equation is a matrix with all entries equal to zero except for the entry in the jth row and the kth column, which is equal to 1. In other words, (27) is the product of the jth and kth elements of the vector $\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})$, multiplied by $-\frac{1}{2d}$. If we do this for all entries of $\boldsymbol{\Sigma}$ and put the results in a vector, we get

$$\frac{\partial d}{\partial \operatorname{vec}(\boldsymbol{\Sigma})} = -\frac{1}{2d}(\boldsymbol{x} - \boldsymbol{\mu})\boldsymbol{\Sigma}^{-1} \otimes (\boldsymbol{x} - \boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}.$$

B.2 Differentiation of Kronecker product

In (22) and (24) a differentiation of the Kronecker product of two identical vectors $\boldsymbol{v} = (\boldsymbol{x}-\boldsymbol{\mu})$ with respect to $\boldsymbol{\mu}$ is required. To see what happens, partially write out the Kronecker product:

$$\boldsymbol{v} \otimes \boldsymbol{v} = \begin{pmatrix} v_1\boldsymbol{v} \\ \vdots \\ v_p\boldsymbol{v} \end{pmatrix}.$$

Element-wise application of the product rule yields

$$\frac{\partial \boldsymbol{v} \otimes \boldsymbol{v}}{\partial \boldsymbol{v}} = \begin{pmatrix} \boldsymbol{v}\frac{\partial v_1}{\partial \boldsymbol{v}} \\ \vdots \\ \boldsymbol{v}\frac{\partial v_p}{\partial \boldsymbol{v}} \end{pmatrix} + \begin{pmatrix} v_1\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{v}} \\ \vdots \\ v_p\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{v}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{v} & \dots & \boldsymbol{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \dots & \boldsymbol{v} \end{pmatrix} + \begin{pmatrix} v_1\boldsymbol{I}_p \\ \vdots \\ v_p\boldsymbol{I}_p \end{pmatrix} = \boldsymbol{I}_p \otimes \boldsymbol{v} + \boldsymbol{v} \otimes \boldsymbol{I}_p.$$

Plugging $\boldsymbol{v} = (\boldsymbol{x}-\boldsymbol{\mu})$ back in and using $\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{\mu}} = -\boldsymbol{I}_p$, we get

$$\frac{\partial (\boldsymbol{x}-\boldsymbol{\mu}) \otimes (\boldsymbol{x}-\boldsymbol{\mu})}{\partial \boldsymbol{\mu}} = -[(\boldsymbol{x}-\boldsymbol{\mu}) \otimes \boldsymbol{I}_p + \boldsymbol{I}_p \otimes (\boldsymbol{x}-\boldsymbol{\mu})].$$

References

Bianco, A. M., & Yohai, V. J. (1996). Robust estimation in the logistic regression model. In *Robust statistics, data analysis, and computer intensive methods* (pp. 17–34). Springer.

Camponovo, L., Scaillet, O., & Trojani, F. (2012). Robust subsampling. *Journal of Econometrics*, *167*(1), 197–210.

Cantoni, E., & Ronchetti, E. (2001). Robust inference for generalized linear models. *Journal of the American Statistical Association*, *96*(455), 1022–1030.

Croux, C., & Haesbroeck, G. (2003). Implementing the bianco and yohai estimator for logistic regression. *Computational Statistics & Data Analysis*, *44*(1-2), 273–295.

Donoho, D. L., & Huber, P. J. (1983). The notion of breakdown point. *A festschrift for Erich L. Lehmann*, *157184*.

Efron, B. (1979). Computers and the theory of statistics: thinking the unthinkable. *SIAM Review*, *21*(4), 460–480.

Efron, B. (1987). Better bootstrap confidence intervals. *Journal of the American statistical Association*, *82*(397), 171–185.

Maronna, R. A., & Yohai, V. J. (1981). Asymptotic behavior of general m-estimates for regression and scale with random carriers. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, *58*(1), 7–20.

Peremans, K., Segaert, P., Van Aelst, S., & Verdonck, T. (2017). Robust bootstrap procedures for the chain-ladder method. *Scandinavian Actuarial Journal*, *2017*(10), 870–897.

Rousseeuw, P., & Yohai, V. (1984). Robust regression by means of S-estimators. In *Robust and nonlinear time series analysis* (pp. 256–272). Springer.

Rousseeuw, P. J., & Driessen, K. V. (1999). A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, *41*(3), 212–223.

Salibian-Barrera, M. (2000). *Contributions to the theory of robust inference* (Unpublished doctoral dissertation). University of British Columbia.

Salibian-Barrera, M., Van Aelst, S., & Willems, G. (2006). Principal components analysis based on multivariate mm-estimators with fast and robust bootstrap. *Journal of the*

American Statistical Association, *101*(475), 1198–1211.

Salibian-Barrera, M., Van Aelst, S., & Willems, G. (2008). Fast and robust bootstrap. *Statistical Methods and Applications*, *17*(1), 41–71.

Salibian-Barrera, M., & Zamar, R. H. (2002). Bootrapping robust estimates of regression. *The Annals of Statistics*, *30*(2), 556–582.

Shrout, P. E., & Bolger, N. (2002). Mediation in experimental and nonexperimental studies: new procedures and recommendations. *Psychological methods*, *7*(4), 422.

Singh, K. (1998). Breakdown theory for bootstrap quantiles. *The Annals of Statistics*, *26*(5), 1719–1732.

Tatsuoka, K. S., & Tyler, D. E. (2000). On the uniqueness of s-functionals and m-functionals under nonelliptical distributions. *Annals of Statistics*, 1219–1243.

Yohai, V. J. (1987). High breakdown-point and high efficiency robust estimates for regression. *The Annals of Statistics*, 642–656.