Erasmus University Rotterdam
Erasmus School of Economics
Bachelor Thesis Econometrics and Operations Research

Extension of the Sparse Direction of Maximal Outlyingness Based on Different Sparse Regressions

Joep Weterman (533233)

Abstract

Cellwise robust M (CRM) regression is an estimator which provides a map of cellwise outliers. Additionally, it also produces vector regression coefficients that are robust against vertical outliers and leverage points. It does so by making use of the SPADIMO algorithm, which is an outlier detection method. Two alternative versions of the cellwise robust M (CRM) regression estimator are introduced, using a different sparse regression method to estimate the vector regression in the SPADIMO algorithm. Initially, a Sparse Non-linear Iterative Partial Least Squares (SNIPLS) regression is used for the estimation of the vector regression. There are two alternative versions of the cellwise robust M regression estimator, one implements a Least Absolute Shrinkage and Selection Operator (LASSO) and the other implements an Elastic Net regression instead of the sparse non-linear iterative partial least squares regression. The new regression methods are tested on varying levels of contamination using Mean Squared Error of Prediction (MSEP), Mean Absolute Error (MAE) and Rooted Mean Squared Error of Imputation (RMEI) as the main evaluation measures. The overall performance of the original cellwise robust M regression is more desirable. However, in some specific cases, the original cellwise robust M regression is outperformed by one of the introduced cellwise robust M regressions. This suggests that the cellwise robust M regression estimator could be improved. We test the different methods on a simulated dataset and a real-world Swiss nutrient composition dataset.

Supervisor:	Aurore Archimbaud
Second assessor:	Kathrin Gruber
Date final version:	29th June 2023

Contents

1 Introduction

Linear regression is a widely used statistical technique that is crucial in various fields, such as economics, finance, social sciences, engineering, and data analysis. Thus developing methods that estimate sufficient parameters are favourable, since it improves the predictive performance of these models. The least squares estimator is one of the most commonly used methods since it adheres to several optimality criteria under the assumption of a normal distribution. However, when these criteria are not satisfied the least squares estimator is not optimal. Recently developed methods still obtain satisfactory regression parameters which lead to decent predictive performance of the models even in the presence of casewise deviations. Casewise deviations represent the differences between the predicted values and the actual values of the target variable for each observation. In the presence of cellwise deviations, robust linear regression methods often outperform least squares methods. These methods are extremely useful for financial analysis, fraud detection and network intrusion detection. One of these methods is the CRM regression (Filzmoser, Höppner, Otner, Serneels & Verdonck, 2020) which yields a map of cellwise outliers, which are individual data points within a dataset that significantly deviate at the level of individual cells from the expected distribution. Additionally, it also produces vector regression coefficients that are robust against vertical outliers and leverage points. Vertical outliers are observations in a dataset that have extreme values on the vertical axis, which has a significant effect on the estimation of the regression models. Leverage points are observations that are extreme in their predictor values and significantly affect the shape and position of the regression.

In order to account for casewise vertical outliers and leverage points, the CRM regression estimator uses an iteratively re-weighted least squares algorithm. The weights initially used are obtained from a robust MM estimator. In each iteration the Sparse Direction of Maximal Outlyingness (SPADIMO) algorithm (Debruyne, Höppner, Serneels & Verdonck, 2019) is applied, and the cells that are significant for outlyingness are then detected. Outlyingness is defined as the degree to which an individual observation deviates from the typical pattern of the rest of the data. These cells are then down-weighted by the re-weighting algorithm. The CRM regression produces regression coefficients that are highly robust and also obtain reliable cellwise outlier detection. Filzmoser et al. (2020), showed that the CRM method is more efficient than a casewise robust estimator.

This paper mainly focuses on the regression method used to estimate the vector regression in the SPADIMO procedure and its function in the CRM method. SPADIMO estimates the vector regression using a Sparse Partial Least Squares (SPLS) regression. SPLS (Chun & Keleş, 2010) regression can be estimated using the SNIPLS (Hoffmann, Filzmoser, Serneels & Varmuza, 2016) algorithm, due to the fact that it is a univariate regression problem. On the other hand, the vector regression can also be estimated using other sparse regression techniques such as LASSO (Tibshirani, 1996) and elastic net (Zou & Hastie, 2005). According to Huang et al. (2008), the main advantage of using LASSO is that its continuity and thus more stable than subset selection, in which different combinations of predictor variables are evaluated and a model is fit for each combination. Additionally, The LASSO is also computationally feasible for high-dimensional data. Another option for the initial vector regression estimator in SPADIMO is the adaptive LASSO (Huang, Ma & Zhang, 2008), which also has the oracle property even

when the number of covariates exceeds the sample size. The oracle property states that if the true underlying model is sparse and the design matrix satisfies certain conditions, then LASSO will provide accurate estimates of the response variable. Zou & Hastie (2005) found that the LASSO regression is often outperformed by the Elastic Net regression, while the models are similar in sparsity. If the number of predictors exceeds the number of observations, the Elastic Net is especially useful. A downside of the SPADIMO procedure is that it can only handle up to 50% casewise contamination as shown in Filzmoser et al. (2020) and is not tested for varying percentages of cellwise contamination. The expectation is that with the implementation of LASSO or Elastic Net, instead of SNIPLS in the SPADIMO algorithm, a higher percentage of contamination can be handled than by the original CRM regression. The research question tackled in this paper is;

Which of the three regression methods, which are SNIPLS, LASSO or Elastic Net, is the most effective at estimating the vector regression in SPADIMO used in cellwise robust M regressions under different percentages and sorts of contamination?

We developed two alternative versions of CRM, CRM-LASSO and CRM-ElasticNet. These alternative versions are the result of replacing the SNIPLS regression in SPADIMO with a LASSO or Elastic Net regression. To answer the research question we performed several simulation studies and examined a real-world dataset. We performed four different simulation studies, one where the parameters were fixed, one where there was varying magnitude of outlyingness, one where there was varying percentage of casewise contamination and one where there was varying percentage of cellwise contamination. We found that generally CRM is preferred. This is mainly due to the fact that CRM is significantly better at imputation than the other models, and the predictive performance and bias are similar for the models. In the real-world example, CRM-LASSO was preferred over the other models.

All source code and data used in this work can be found in the GitHub URL [1]. The remainder of this paper is organized as follows. In section 2, related works regarding the SPADIMO algorithm and CRM regression are presented. Next, in section 3, we will discuss the real-world dataset that is used in this paper. Section 4, explains the different regression methods and evaluation methods used in this paper. Section 5, the simulation study and different simulation settings will be clarified. Then, in section 6, the result of the simulation study and real-world dataset application will be discussed. Finally, in section 7, the conclusion and answer to our research question will be provided with suggestions for future work.

2 Related Works

This section discusses the main findings and conclusions from the papers concerned with CRM regressions. This section also elaborates more on why the findings from the past papers discussed below are relevant to the current research conducted in this paper. Additionally, we discuss how our findings are a relevant contribution to the literature.

[1] https://github.com/joepweterman/CRM-LASSO-ENET.git

The paper by Filzmoser et al. (2020) introduced a new method for robust outlier detection named cellwise robust M (CRM) regressions. The paper applied CRM to the robust estimator of an MM regression in a simulation study, in which CRM is compared to multiple different estimator functions in order to assess the performance of the model. The other estimator functions were: a conventional MM regression, MM regression combined with Deviating Data Cells (DDC), Ordinary Least Squares (OLS) regression, and OLS combined with DDC. It was found that the robust regression methods have a significantly better predictive performance than the least squares method. If assumed that a linear model has generated the data, then the authors found that CRM performance at imputation and detection of deviating cells is preferred. Finally, the authors found that CRM is also preferred for estimating individual regression coefficients. An important downside of CRM is that it will break down if applied to datasets that contain over 50% of casewise outliers. We will use CRM, (Filzmoser et al., 2020), as the baseline model in this current research.

The paper by Debruyne et al. (2019) introduced a method that estimates the univariate direction of maximal outlyingness. The authors found that it is possible to reformulate the estimation of the direction of maximal outlyingness into the normed solution of a least squares regression problem. The authors suggest tackling that problem with SPLS (Chun & Keleş, 2010) regression, preferably by using the SNIPLS (Hoffmann et al., 2016) algorithm. According to the simulation study, SPADIMO has an average detection rate between 92.794% and 100% and can find practically all contaminated variables. However, the average detection rate drops to about 80% when the number of observations is smaller than the number of variables. SPADIMO obtained a swamping rate, which is the rate where the estimated regression coefficients are based towards zero when the true coefficients are non-zero, between 0.286% and 5.019%. However, for swamped cases, SPADIMO typically loses performance after the first variable has been screened. By implementing a LASSO or Elastic Net regression instead of the SNIPLS regression, we expect to obtain a model which minimizes the swamping rate.

The paper by Hoffman et al. (2016) introduced a sparse non-linear iterative partial least squares algorithm, which is a sparse regression method that searches for a sparse set of variables that are essential for forecasting the outcome in cases where the ratio of the number of predictors to the number of observations is significantly greater. The authors of this paper found, that instead of using numerical optimization to calculate SPLS, the SNIPLS algorithm is able to use the exact partial least squares solutions. Another advantage found by the authors is, that for the outlyingness problem, the algorithm needs only a single model component, such as a single latent variable, in order to detect the variables which contribute to outlyingness.

The paper by Tibshirani (1996) first introduced the LASSO regression method, which is used for estimation in linear models. Since the LASSO regression estimates some coefficients that are exactly zero while also estimating a subset of the coefficients as non-zero, which makes the LASSO regression by definition a sparse regression method. The authors found that the LASSO regression method has the advantages of subset selection and the Ridge regression. Having the stability of the Ridge regression and having the interpretability of subset selection. The authors also found that the LASSO regression performed best when there was a small to moderate number of moderate-sized effects, outperforming the ridge regression and subset selection. In

the scenario with a large number of small effects the ridge regression performs the best, however, the LASSO regression does perform well.

The paper by Zhou & Hastie (2005) introduced the Elastic Net regression method, which uses regularization and variable selection. While also being a sparse regression method, the authors found that the Elastic Net regression method regularly outperforms the LASSO regression method. When the number of predictors is much larger than the number of observations the Elastic Net regression method is especially preferred. The authors found that the Elastic Net regression method obtains a sparse model with good predictive performance.

There currently does not exist any research regarding the optimal regression method to use for estimating the vector regression in SPADIMO. Knowing which regression method is most effective at estimating the vector regression would lead to a better predictive performance of SPADIMO. Additionally, CRM discards not as much information in the data as casewise robust estimators, which means that an improved CRM would be useful in fields of research where data is scarce.

3 Methodology

In this section, SPADIMO and cellwise robust M regression are explained. Other sparse regression estimate methods used, such as SPLS, SNIPLS, LASSO and Elastic Net are also explained. Also, we will explain different evaluation techniques used to assess the performance of the different regression methods discussed in this research. The codes of these methods can be found on GitHub[2].

3.1 SPADIMO

SPADIMO, as described in Debruyne et al. (2019), is an outlier detection algorithm that takes as input a data matrix $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_n)^T$ with dimension n x p, vector of case weights $\boldsymbol{w}$ are initially obtained from MM regression estimator, index $i \in \{1, ..., n\}$ of the observation on which to apply SPADIMO, and grid of values $\mathcal{L} = [\ell_1, \ell_2]$ within $[0, 1]$.

The weights w_i are based on the squared robust Mahalanobis distance for every point $\mathbf{x} \in \mathbb{R}^p$ are defined as follows:

$$m(\mathbf{x} : \hat{\boldsymbol{\mu}}_r, \hat{\boldsymbol{\Sigma}}_r)^2 = (\mathbf{x} - \hat{\boldsymbol{\mu}}_r)^T \, \hat{\boldsymbol{\Sigma}}_r^{-1} \, (\mathbf{x} - \hat{\boldsymbol{\mu}}_r) \tag{1}$$

where $\hat{\boldsymbol{\mu}}_r$ and $\hat{\boldsymbol{\Sigma}}_r$ denote robust estimates of location and scatter for $\mathbf{X}$. The distance between point $\boldsymbol{x}$ and the robust location is measured by the robust Mahalanobis distances. Under the assumption of multivariate normality, the squared Mahalanobis distances are asymptotically χ^2_p distributed. Weights can then be obtained for each observation as follows:

$$w_i = \begin{cases} 1 & m(\mathbf{x} : \hat{\boldsymbol{\mu}}_r, \hat{\boldsymbol{\Sigma}}_r)^2 \leq \chi^2_{p,0.975} \\ 0 & otherwise \end{cases} \tag{2}$$

The output of the SPADIMO algorithm is a sparse direction of maximal outlyingness $\mathbf{a}(\eta, \mathbf{x}_i)$

[2]https://github.com/joepweterman/CRM-LASSO-ENET/tree/main

for each $\eta \in \mathcal{L}$ and the corresponding subset of variable(s) contributing to outlyingness. A path of sparse direction of maximal outlyingness $\mathbf{a}(\eta, \mathbf{x}_i)$ is defined by

$$\begin{aligned} \mathbf{a}(\eta, \mathbf{x}_i) &= \frac{\theta(\eta)}{||\theta(\eta)||}, \text{ with} \\ \theta(\eta) &= \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{argmin} \left(||\mathbf{y}_w^i - \mathbf{X}_w \boldsymbol{\beta}||^2 + \eta \sum_{j=1}^{p} |\boldsymbol{\beta}_j| \right) \end{aligned} \tag{3}$$

where η is the optimal sparsity parameter for which the minimal number of parameters is accepted, $\boldsymbol{\beta}$ the vector of the regression coefficients, y_w^i the ith basis vector in $\mathbb{R}^n$ and $\mathbf{X}_w = (\sqrt{w_1}(\mathbf{x}_1 - \hat{\boldsymbol{\mu}}_w)^T, ..., (\sqrt{w_n}(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_w)^T)^T$.

The first step of the SPADIMO algorithm is to standardize $\boldsymbol{X}$ to $\boldsymbol{Z}$. This is done by first subtracting a robust estimate for location and then dividing by a robust scale estimate. The second step is to check the weight w_i of the observation is equal to zero. If so then replace that weight with a significantly small weight (e.g. 0.00001). The third step is to construct $\mathbf{Z}_w = (\sqrt{w_1}\mathbf{z}_1^T, ..., \sqrt{w_n}\mathbf{z}_n^T)^T$ and $\mathbf{y}_w^i$. The fourth step is to set $\mathbf{Z}^{(\eta)} = \mathbf{Z}_w$ to start the algorithm. The fifth and last step is to decrease ℓ_2 to ℓ_1, then for each $\eta \in \mathcal{L}$ do:

Algorithm 1 SPADIMO algorithm

1: Estimate $\boldsymbol{\theta}(\eta)$, which is the SPLS vector of regression coefficients regressing $\mathbf{y}_w^i$ on $\mathbf{Z}^{(\eta)}$ at $h = 1$.
2: Calculate $\mathbf{a}(\eta, \mathbf{x}_i) = \frac{\theta(\eta)}{||\theta(\eta)||}$.
3: Determine $v = \{j : \boldsymbol{\theta}_j(\eta) \neq 0\}$, which is the subset of variables that contributes the most to outlyingness.
4: Update $\mathbf{Z}^{(\eta)} = \frac{\mathbf{Z}^{(\eta)}}{\{Z_j | j \in v\}}$, with Z_j denoting the jth column of $\mathbf{Z}$.
5: then compute $\mathbf{r}(\mathbf{z}_i^{(\eta)}; \mathbf{Z}^{(\eta)})$, where $\mathbf{z}_i^{(\eta)}$ denotes the ith row of $\mathbf{Z}^{(\eta)}$.

The stopping criteria of the algorithm is $\mathbf{r}(\mathbf{z}_i^{(\eta)}; \mathbf{Z}^{(\eta)})^2 < \chi^2_{\alpha,q}$, where α is the required level χ^2 significance and q is the number of remaining columns of $\mathbf{Z}^{(\eta)}$.

3.2 CRM

Cellwise robust M regression estimator was first introduced by Filzmoser et al. (2020). It is the first estimator which provides a map of cellwise outliers, which is consistent with the linear model. It also simultaneously provides a vector, consisting of regression coefficients, that is robust against leverage points and vertical outliers.

In the sake of clear explanation, we split up CRM into two distinct algorithms. The first algorithm (Algorithm 2) will be denoted as the iteratively re-weighted least squares algorithm and the second as the imputation algorithm. We first define how the complete algorithm functions. First, we apply a MM regression on the original observations in order to obtain the initial estimator $\hat{\boldsymbol{\beta}}$. Then the iteratively re-weighted least squares algorithm is run, with the initial estimator $\hat{\boldsymbol{\beta}}$ obtained from the MM regression. Then the iteratively re-weighted least squares algorithm is run again, with the newly obtained imputed data $\boldsymbol{X}_I$ and $\boldsymbol{Y}_I$ from the last step, from which a new estimator $\hat{\boldsymbol{\beta}}$ is obtained and used. Then run the iteratively re-weighted least

squares algorithm again with the weighted data and the least squared estimator obtained from the previous step. The stopping criteria is satisfied when the mean absolute difference (MAD) of the last two regression estimates is less than the predefined tolerance bound, until then it repeats the last step. A tolerance bound is a threshold or limit that is set to determine whether an observation is considered an outlier or not. The iteratively re-weighted least squares algorithm is defined as follows:

Algorithm 2 iteratively re-weighted least squares algorithm

1: The residuals are calculated based on the estimator $\hat{\boldsymbol{\beta}}$
$r_i = y_i - \boldsymbol{x}_i^T \hat{\boldsymbol{\beta}} \qquad \text{for } i \in \{1, ..., n\}.$
2: Observations are detected as outliers if they satisfy
$\frac{|r_i|}{cmed_j|r_j|} > z_{0.95}$
where $c = 1.4826$ such that the mean absolute deviation is consistent.
3: Then for each outlying case the following steps will be taken:
* In order to obtain the outlying variables SPADIMO is applied.
* The values in the outlying variables will be imputed as is done in the imputation algorithm (Algorithm 3) if not all variables have contributed to outlyingness.
* The new imputed matrix is denoted by $\hat{\boldsymbol{X}}$.
4: Update the residuals using the new imputed matrix
$\hat{r}_i = y_i - \hat{\boldsymbol{x}}_i \hat{\boldsymbol{\beta}} \qquad \text{for } i \in \{1, ..., n\}.$
5: Then use the Hampel weight function to calculate the case weights
$w_i = w_H(\frac{|\hat{r}_i|}{cmed_j|\hat{r}_j|})$
with again $c = 1.4826$.
6: The diagonal matrix with the case weights as the diagonal elements is defined as $\boldsymbol{\Omega} = Diag(\sqrt{w_1}, ..., \sqrt{w_n})$.
Then update the imputed data as follows
$\boldsymbol{X}_I = \boldsymbol{\Omega}\hat{\boldsymbol{X}} \qquad \text{and} \qquad \boldsymbol{Y}_I = \boldsymbol{\Omega}\boldsymbol{y}.$

The imputation algorithm is defined as follows:

Algorithm 3 imputation algorithm

1: The index of an outlier $\boldsymbol{x}_i$ is defined as i.
2: The set of cellwise outliers detected in $\boldsymbol{x}_i$ is defined as C.
3: Then detect $\boldsymbol{x}_{k_1}$ and $\boldsymbol{x}_{k_2}$, which are the two nearest neighbors of outlier $\boldsymbol{x}_i$ in the subspace $\{1, ..., p\} \setminus C$. The neighbors also need to have $w_j = 1$ for observation $\boldsymbol{x}_j$.
4: Then the outlying cells need to be imputed $\hat{x}_{iq} = (x_{k_1 q} + x_{k_2 q})/2$ with $q \in C$.

3.3 Regression Methods

In the following section, all the relevant regression methods used in this research are discussed. The regression methods are Sparse Partial Least Squares (SPLS), Sparse Non-linear Iterative Partial Least Squares (SNIPLS), LASSO and Elastic Net.

3.3.1 SPLS

SPLS is used in situations where there are a large number of predictors and potential collinearity among them. The main goal of SPLS regression is to identify a subset of predictors that are most relevant to the response variable while simultaneously minimizing the number of predictors used in the model.

SPLS, as defined in Chun & Keleş (2010), operates under the assumption that the response matrix and predictor matrix have a basic latent decomposition. The response matrix is defined as $Y = TQ^T + F$ where $Y \in \mathbb{R}^{n \times q}$ and predictor matrix as $X = TP^T + E$ where $X \in \mathbb{R}^{n \times p}$. P and Q are matrices of coefficients (loadings), E and F are matrices of random error terms, and K are the linear combination scores which are generated by matrix T. Then we solve subsequent optimization problems in order to find the columns of $W = (w_1, ..., w_k)$. Sparsity can then be obtained from the component matrix $T = XW$. The following formula is used to determine the kth direction vector w_k for univariate Y:

$$w_k = \underset{w}{argmax}\{corr^2(Y, Xw)var(Xw)\} \quad \text{subject to} \quad w^T w = 1, \quad w^T \Sigma_{XX} w_j = 0 \tag{4}$$

for $j = 1, ..., k-1$, where Σ_{XX} is the covariance of X.

3.3.2 SNIPLS

SNIPLS is an algorithm that can be used to solve the SPLS regressions. SNIPLS (Hoffmann et al., 2016) seeks to find a sparse set of variables that are most important for predicting the response in scenarios where the number of predictors is significantly higher than the number of observations.

We use the same notation as used by Hoffman et al. (2016). Define X as a column-wise centered matrix and y as its centered response. We define the matrix $E_1 = X$, and then for $h = 1, ..., H$ we calculate the weighting vector as follows:

$$v_h = (|z_h| - \eta max_i|z_{ih}|) \odot I(|z_h| - \eta max_i|z_{ih}| > 0) \odot sgn(z_h) \tag{5}$$

where $z_h = \frac{E_h^T y}{||E_h^T y||}$ and the weighting factors are penalized by η, which is a fraction of its largest element. $\odot$ is the Hadamard (or element-wise) matrix product. The next deflated matrix E_h is obtained by $E_{h+1} = E_h - \frac{t_h t_h^T E_h}{||t_h||^2}$ where $t_h = E_h v_h$. The weighting vectors of these deflated matrices construct the columns of V. Then $W = V(V^T X^T XV)^{-1}$ define the direction vectors for the transformation of X and $T = XW$ the scores.

3.3.3 LASSO

LASSO (Tibshirani, 1996) is a sparse regression method used for regularization and variable selection. It is especially useful in high-dimensional datasets, specifically when the number of predictors is lower than the number of observations. The LASSO regression adds a penalty term to the ordinary least squares objective function. The penalty term is defined as the sum of absolute values of the regression coefficients multiplied by λ, which is a tuning parameter.

Define $\mathbf{x}^i = (x_{i1}, ..., x_{ip})^T$ as the predictor variables and y_i as the response variable, where

$i = 1, ..., N$. Denote $\hat{\beta}$ as the LASSO estimate and define $\hat{\beta} = (\hat{\beta}_1, ..., \hat{\beta}_p)^T$. Then the LASSO estimates are calculated as follows:

$$\hat{\beta} = argmin\{\sum_{i=1}^{N}(y_i - \sum_j \beta_j x_{ij})^2\} \quad \text{subject to} \quad \sum_j |\beta_j| \leq \lambda. \tag{6}$$

Here, we assume that x_{ij} is standardized and satisfies the following constraints; $\Sigma_i x_{ij}/N = 0$ and $\Sigma_i x_{ij}^2/N = 1$. We also assume that the tuning parameter is $\lambda \leq 0$, which regulates the amount of shrinkage in the regression.

3.3.4 Elastic Net

Elastic Net (Zou & Hastie, 2005) is a combination of the LASSO regression and Ridge regression method (McDonald, 2009). It is a combination of L1 regularization (LASSO) and L2 regularization (Ridge), which results in a balance between feature selection and coefficient shrinkage. Just like LASSO, it is useful in high-dimensional datasets. However, it overcomes the limitations of LASSO in situations with high multicollinearity among the predictors.

Denote the given data by $(\mathbf{y}, \mathbf{X})$, the penalty parameter by (λ_1, λ_2) and the augmented data by $(\mathbf{y}^*, \mathbf{X}^*)$. We use the same notation as in (Zou & Hastie, 2005). A LASSO-type problem is solved by the naive Elastic Net as followed:

$$\hat{\beta}^* = \underset{\beta^*}{argmin}|\mathbf{y}^* - \mathbf{X}^*\beta^*| + \frac{\lambda_1}{\sqrt{(1+\lambda_2)}}|\beta^*|_1 \tag{7}$$

Then, using the fact that $\hat{\beta}(elastic\,net) = \sqrt{1+\lambda_2}\hat{\beta}^*$ and that $\hat{\beta}(naive\,elastic\,net) = (1/\sqrt{1+\lambda_2})\hat{\beta}^*$ then it holds that $\hat{\beta}(elastic\,net) = (1+\lambda_2)\hat{\beta}(naive\,elastic\,net)$.

Résumé Regression Methods

In the two alternative CRM methods introduced in this paper, the SNIPLS algorithm in SPADIMO is replaced by a LASSO regression and an Elastic Net regression. We used the R-package "crmReg" [3] as the basis for the code. However, some changes to specific functions were required. The files that needed new versions were the crm.R, spadimo.R, predict.crm.R. In the SPADIMO.R file the main thing that needed to be changed was the spadimo.exs function, which was the initial estimation of the vector regression in SPADIMO, which is originally a SNIPLS regression. In our code we replaced the SNIPLS regression with a LASSO and Elastic Net regression, expecting that it would improve the initial estimation of the vector regression and thus the performance of the CRM model. In order to evaluate the two alternative CRM models different evaluation methods were used, which will be discussed in the next section.

3.4 Evaluation Methods

Three different evaluation methods are used to assess the relative performance of the different methods. The evaluation methods are the Mean Squared Error of Prediction (MSEP), the Mean Absolute Error (MAE) and the Root Mean Squared Error of Imputation (RMSEI).

[3] https://cran.r-project.org/web/packages/crmReg/index.html

3.4.1 MSEP

MSEP evaluates the performance of a predictive model, it does so by quantifying the average squared difference between the predicted values and the true values of the dependent variable. Which is useful for assessing the predictive performance of a regression. The MSEP of a model is defined by Filzmoser et al. (2020) as followed:

$$MSEP = \frac{1}{n_{clean}} \sum_{i \in I} (\hat{y}_i - y_i)^2 \tag{8}$$

where n_{clean} is the number of uncontaminated cases and I contains the indices of the uncontaminated cases.

3.4.2 MAE

The MAE is used to evaluate bias for the individual regression coefficients. MAE can also be used to assess the robustness to outliers of the regression method. The MAE is defined by Filzmoser et al. (2020) as followed:

$$MAE = \frac{1}{p} \sum_{j=1}^{p} |\hat{\beta}_j - \beta_j| \tag{9}$$

where p is the number of variables, $\hat{\beta}$ is the model predicted β.

3.4.3 RMSEI

The RMSEI is used to measure the average difference between the true values and the imputed values of the dataset. CRM generates an imputed matrix of the contaminated matrix $\boldsymbol{X}^c$, which we denote by $\boldsymbol{X}^{imp}$. The RMSEI reports the performance of each imputed matrix $\boldsymbol{X}^{imp}$ and is defined by Filzmoser et al. (2020) as follows:

$$RMSEI(\boldsymbol{X}^{imp}, \boldsymbol{X}) = \sqrt{\frac{1}{np} \sum_{i=1}^{n} \sum_{j=1}^{p} (x_{ij}^{imp} - x_{ij})^2} \tag{10}$$

where n is the number of cases generated and p is the number of variables.

4 Simulation Study

In this simulation study, the performance of CRM (Filzmoser et al., 2020), applied to the robust coefficient estimator of an MM regression (Maronna, Martin, Yohai & Salibián-Barrera, 2019) is compared to two alternative versions of CRM. The version where the SNIPLS regression is replaced with a LASSO regression will be referred to as *CRM-LASSO*, and the version where SNIPLS is replaced by Elastic Net will be referred to as *CRM-ElasticNet*. The simulation study will establish which version of CRM is most efficient under different levels and types of contamination. The simulation study performed is similar to the simulation study performed by Filzmoser et al. (2020). The upcoming sections will discuss the simulation setting, the method of adding contamination and the regression method setting.

4.1 Simulation Setting

In this simulation study, we use a p-dimensional multivariate normal distribution to generate the data. The distribution has $\boldsymbol{\mu} = (0, ..., 0)^T$ and covariance matrix $\boldsymbol{\Sigma}$, that is structured such that $\boldsymbol{\Sigma}_{i,i} = 1$ for $i = 1, ..., p$, $\boldsymbol{\Sigma}_{j+1,j} = \boldsymbol{\Sigma}_{j,j+1} = 0.5$ for $j = 1, ..., p-1$ and $\boldsymbol{\Sigma}$ is zero everywhere else. The number of observations generated is equal to 400, so $n = 400$, and the number of variables generated is equal to 50, so $p = 50$. Then, we obtain the data matrix $\boldsymbol{X} \in \mathbb{R}^{nxp}$.

The response variable for the uncontaminated data is generated as follows:

$$\boldsymbol{y} = \mathbf{1}_n \beta_0 + \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{11}$$

where the intercept is set to 10 and $\boldsymbol{\beta}$ is defined as a vector with length p. The random values of $\boldsymbol{\beta}$ are obtained from a standard normal distribution that has been normalized to length 10. We define the error term $\boldsymbol{\epsilon}$ as a length n vector of random values from a normal distribution with a mean of 0 and a standard deviation of 0.5. The clean data is represented by $(\boldsymbol{y}, \boldsymbol{X})$ and the regression coefficients are represented by $(\beta_0, \boldsymbol{\beta})$.

4.2 Adding Contamination

In order to establish which version of CRM is most efficient under different levels of contamination, we need to define the method of adding contamination. The contaminated matrix will be denoted by $\boldsymbol{X}^c$. The casewise outliers are generated by randomly selecting a fraction $r = 5\%$ of the observations in $\boldsymbol{X}$. So, in this study $r \times n = 20$ rows of $\boldsymbol{X}$ will be casewise contaminated. The fraction of contamination is mainly fixed, however, later the different magnitudes of contamination will be varied. Denote $I^c \subset \{1, ..., n\}$ as the random subset of 20 selected case indices. The cellwise outliers are generated by randomly selecting $\check{r} = 10\%$ for each case $i \in I^c$ of the predictor variables. In this study, $\check{r} \times p = 5$ cells will be contaminated, so there will be 5 cellwise outliers. Let $J_i^c \subset \{1, ..., p\}$ denote the subset of 5 selected variable indices for each $i \in I^c$. The effect of varying percentages of $\check{r}$ will also be investigated, in order to investigate the capabilities of the models at handling different levels of cellwise outliers.

The contaminated matrix $\boldsymbol{X}^c$ is defined as follows:

$$x_{ij}^c = \bar{x}_j + k s_j + e = \bar{x}_j + k \sqrt{\frac{1}{n-1} \sum_{l=1}^{n} (x_{lj} - \bar{x}_j)^2} + e \tag{12}$$

for all $i \in I^c$ and $j \subset J_i^c$. Where $\bar{x}_j$ is the mean value of variable j, k is the level of casewise contamination, s_j is the standard deviation of variable j, and e a random variable of the standard normal distribution. The contaminated data is represented by $(\boldsymbol{y}, \boldsymbol{X^c})$. We will repeat the simulation study for different levels of k later in the paper.

4.3 Regression Methods Setting

The CRM, CRM-LASSO and CRM-ElasticNet are fitted to the contaminated data $(\boldsymbol{y}, \boldsymbol{X^c})$. For all the CRM regression estimations the same parameter settings as (Filzmoser et al., 2020) are used. The relative tolerance of converging the regression coefficients is set to 0.01 and the

outlyingness factor used for SPADIMO is set to 1.5. The maximal number of simulations is also set to 100. In our code we made use of the package 'glmnet' [4], for the LASSO regression we use the standard settings used in the package. For the Elastic Net regression, we also use the standard settings, except for the tuning parameter α which is set equal to 0.5.

5 Results

In this section, the results of the simulation study will be discussed according to the previously defined evaluation methods. We performed four different simulation studies and one real-world example, the results of each study will be discussed in the corresponding subsection.

5.1 Simulation Results Fixed Parameters

In this simulation study, all the parameters are fixed, in order to test how to models perform in a fixed environment and to see if we obtain the same results as Filzmoser et al. (2020). We assume k, denoting the magnitude of outlyingness, to be equal to 6, the percentage of casewise contamination to be equal to 5% and the percentage of cellwise contamination to be equal to 10%. Each result is the reported average of 100 simulation instances. In Figs. 1-2, boxplots are used to display the results. At the bottom of the figures, which displays the average outcomes for each approach, the best result is printed in bold.

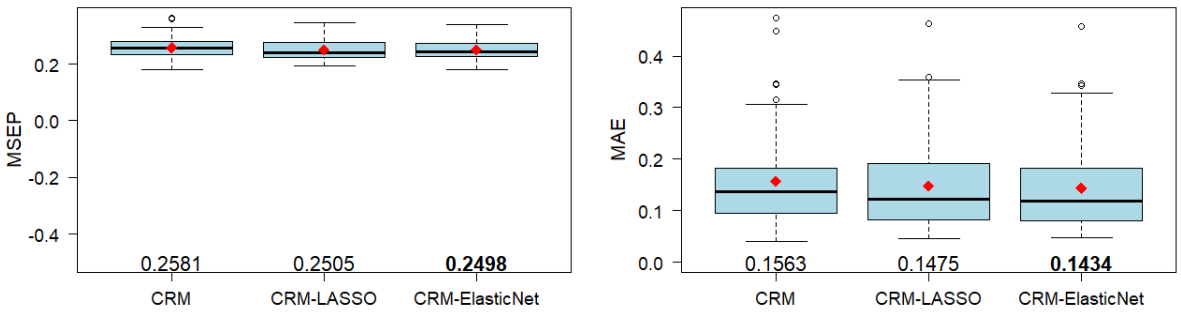

Figure 1: Boxplot of the MSEP (left) and MAE (right) for CRM, CRM-LASSO and CRM-ElasticNet

Fig. 1 shows that all three methods have a similar MSEP, meaning that all three methods have a similar predictive performance. Please note that both CRM-LASSO and CRM-ElasticNet slightly outperform the regular CRM according to MSEP. From Fig. 1, we can derive the bias of the regression coefficients in the presence of cellwise outliers of the different methods. A lower MAE means that the regression method is less biased towards the regression coefficients. Fig. 1 shows that CRM-ElasticNet is the least biased, however, the difference in the values is minimal.

[4]https://cran.r-project.org/web/packages/glmnet/index.html

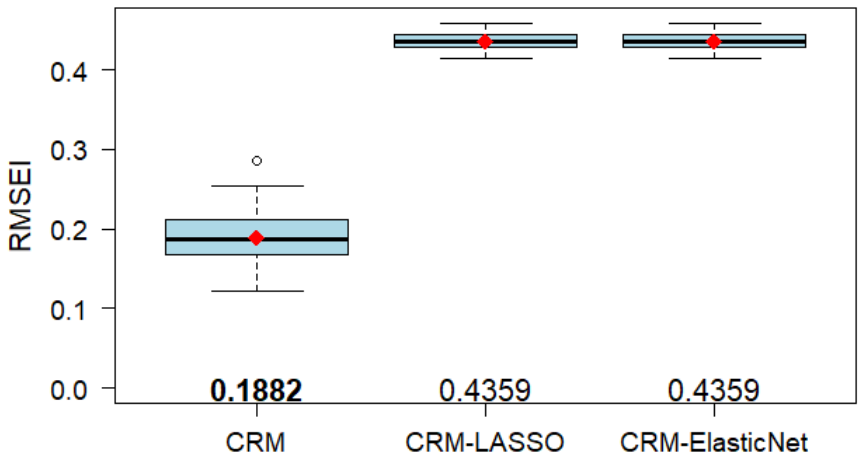

Figure 2: Boxplot of the RMSEI for CRM, CRM-LASSO and CRM-ElasticNet.

From Fig. 2, we can derive the performance of the different CRM regressions at imputing the true values for cellwise outliers. Fig. 2 shows that CRM significantly outperforms CRM-LASSO and CRM-ElasticNet. Thus, it can be concluded that CRM is significantly better at imputing the true values for the cellwise outliers than the other models. Please note that the CRM-LASSO and CRM-ElasticNet have the exact same RMSEI, this is probably caused by the fact that the two models are so similar in nature. Both models have similar predictive performance and bias, however, CRM performs significantly better at imputation. Hence, we conclude that the CRM, in the fixed parameter setting, is preferred over the CRM-LASSO and CRM-ElasticNet.

What should be noted is that we do not obtain the same values for the CRM regression as Filzmoser et al. (2020). Although the difference in the obtained values is likely insignificant, it should be noted that the difference is likely due to a different seed and or a different processor. All models are computationally efficient, however, CRM-LASSO and CRM-ElasticNet do take significantly less time to execute. The execution times for the CRM algorithm were on average 9 seconds, for the CRM-LASSO algorithm it took on average of 5 seconds and for the CRM-ElasticNet it took on average of 5.1 seconds. The execution times were measured on an Intel core i7 10th generation with 1.30 GHz and 10.2 GB RAM.

5.2 Simulation Results varying Magnitude of Outlyingness

In the previously presented figures, the magnitude of outlyingness was fixed. Now, we will vary the parameter k, which controls the magnitude of outlyingness. The goal of this simulation study is to find the optimal sparse regression for SPADIMO under varying magnitudes of outlyingness. We report the simulation result for $k \in \{0, 1, 2, ..., 8\}$ for the MSEP, MAE and RMSEI. Figs. 3-4 illustrate the average results across 10 simulations.

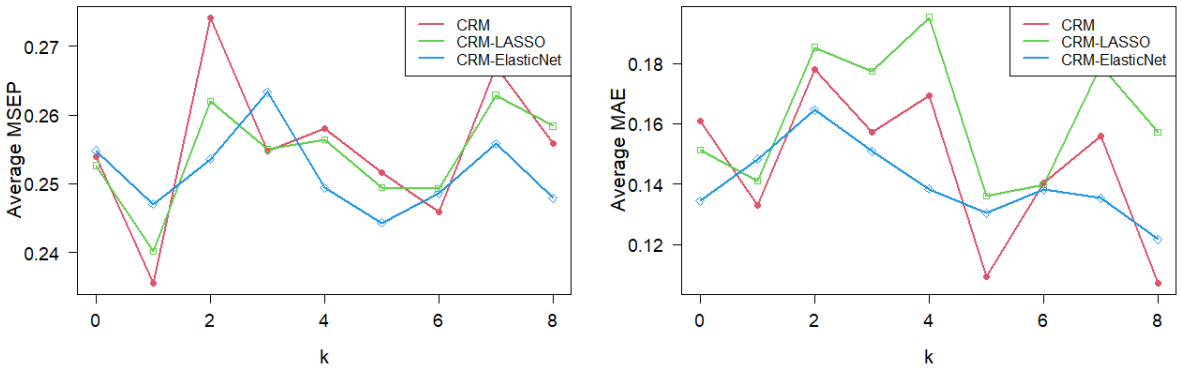

Figure 3: Average MSEP (left) and average MAE (right) for CRM, CRM-LASSO and CRM-ElasticNet for varying levels of magnitude of outlyingness, denoted by parameter k.

Fig. 3 shows that the MSEP indicates that for less severe contamination, where k is less than 5, CRM-LASSO performs the most consistent. However, when the contamination gets more severe it can be seen that the three CRM versions perform similarly, with CRM-ElasticNet slightly outperforming the other models. Fig. 3 shows that in the case of $k = 1, 4, 6, 7$ CRM-ElasticNet has the least biased parameter estimation. We can thus conclude from the MAE that CRM-ElasticNet outperforms the other methods regarding parameter estimation.

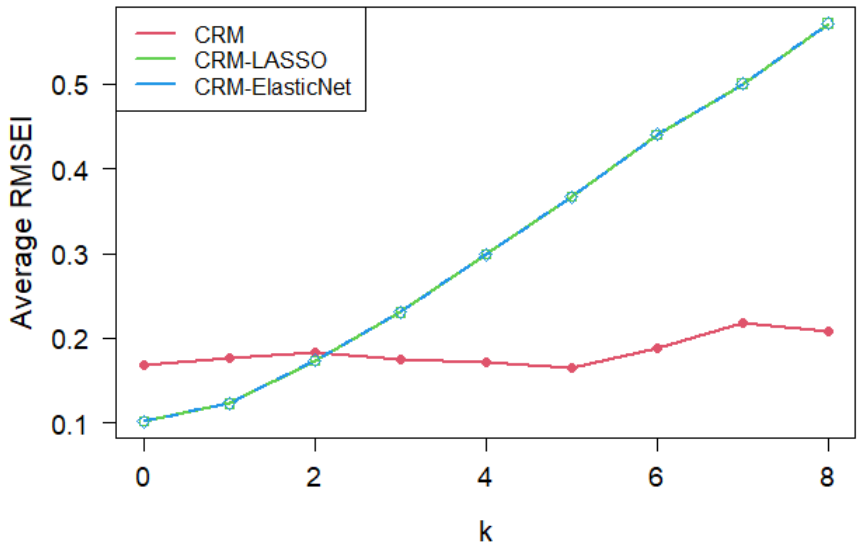

Figure 4: Average RMSEI for CRM, CRM-LASSO and CRM-ElasticNet for varying levels of magnitude of outlyingness, denoted by the parameter k.

Fig. 4 indicates that for CRM-LASSO and CRM-ElasticNet an increase in k will lead to an increase in the RMSEI, this results in a depreciating performance of the regressions at imputing the true values for cellwise outliers, this might be caused by the overfit effect. CRM

is roughly the same for the different values of k. The average RMSEI with a varying magnitude of contamination confirms again that CRM is clearly better at imputing the true values for the cellwise outliers. Also, the capabilities of CRM-LASSO and CRM-ElasticNet are identical at predicting or imputing missing values in a dataset. This could again be due to the fact that the regressions are closely related.

We can conclude that CRM is the preferred model under varying magnitudes of outlyingness. This is mainly, again, because of the difference in performance of imputation. CRM significantly outperforms the other models at imputation, and the difference in predictive performance and bias is not significant enough to prefer another model.

5.3 Simulation Results varying Percentage of Casewise Contamination

In this section, we vary the percentage of casewise contamination. The goal of this simulation study is to reveal the breakdown behaviour of the models and to investigate if one of the models is better equipped to handle a higher percentage of casewise contamination. Figs. 5-6 show the evaluation methods under different percentages of casewise contamination. Starting from no contamination, up to 50%.

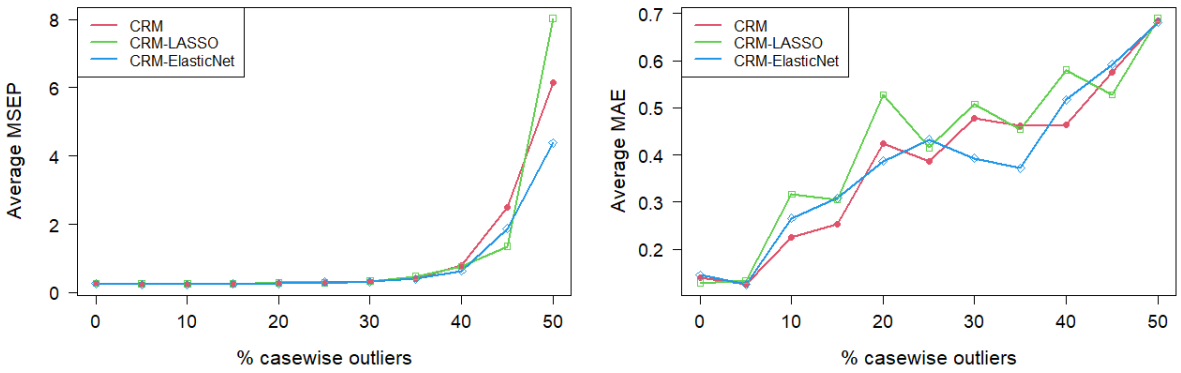

Figure 5: Average MSEP (left) and average MAE (right) for CRM, CRM-LASSO and CRM-ElasticNet for different fractions of casewise contamination.

From Fig. 5 we see that from the MSEP we can conclude that all models start to break down around the 40% contamination mark, with CRM-ElasticNet breaking down the slowest. Until the 40% casewise contamination, the models have an identical average MSEP, after the 40% casewise contamination CRM-LASSO average MSEP increases the fastest. Fig. 5 indicates that when the casewise contamination is higher the CRM-LASSO and CRM-ElasticNet are more biased than the CRM. CRM-ElasticNet is the least biased between 25% and 40% casewise contamination. Overall, CRM has the lowest average MAE and is thus the least biased.

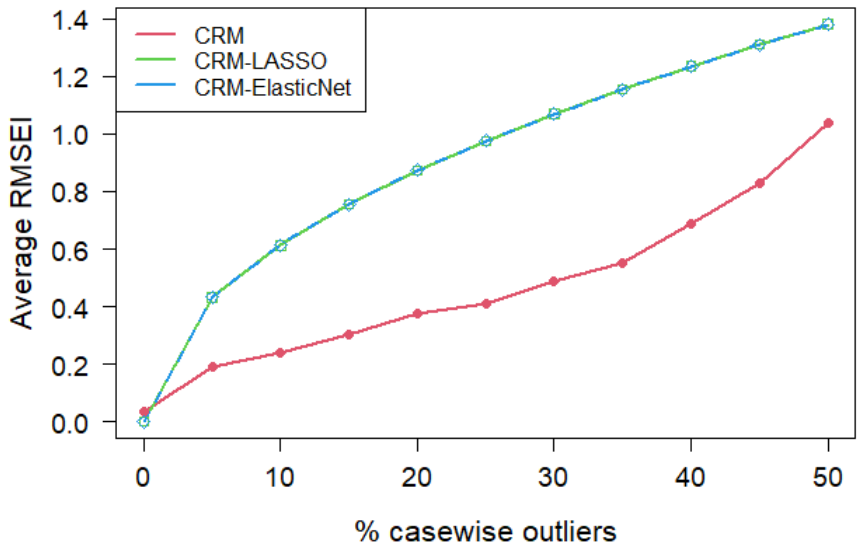

Figure 6: Average RMSEI for CRM, CRM-LASSO and CRM-ElasticNet for different fractions of contaminations.

Additionally, Fig. 6 shows that CRM has the best accuracy of imputation under varying percentages of casewise outliers. We conclude that CRM is still the preferred model under varying percentages of casewise contamination. Since the three models perform similarly in MSEP en MAE, however, CRM has a lower RMSEI for all percentages of casewise contamination. Suggesting that the predictive performance and regression biases of the models are similar, but CRM being clearly better at imputation.

5.4 Simulation Results varying Percentage of Cellwise Contamination

In this section, we vary the percentage of cellwise contamination. The goal of this simulation study is to reveal if one of the models is better equipped to handle a higher percentage of cellwise contamination. Figs. 7-8 show the evaluation methods under different percentages of cellwise contamination. Starting from 5% cellwise contamination, up to 50%. This illustrates how the different models are able to handle extreme or atypical observations. Figs. 7-8 illustrate the average results across 10 simulations.

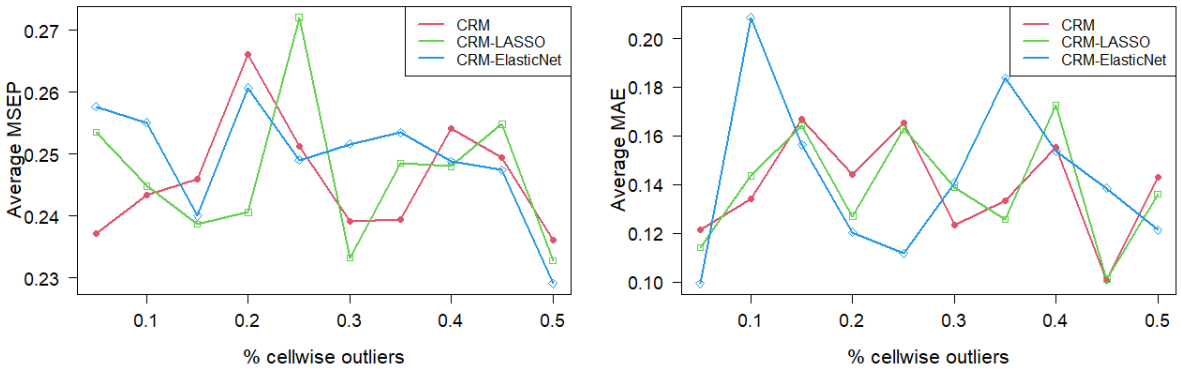

Figure 7: Average MSEP (left) and average MAE (right) for CRM, CRM-LASSO and CRM-ElasticNet for different percentages of cellwise outliers.

Fig. 7 shows the MSEP which indicates how the accuracy and predictive performance of the different models vary under different percentages of cellwise outliers. CRM-LASSO generally outperforms CRM, except for around 35% of cellwise contamination then the CRM performs the best. Fig. 7 shows the MAE which gives us the bias in the regression coefficients of the different models under varying percentages of cellwise outliers. CRM-ElasticNet is generally the least biased, however, CRM is preferred for around the 10% and 30% of cellwise contamination. What should be noted is that the CRM-ElasticNet varies the most in the values of the average MAE.

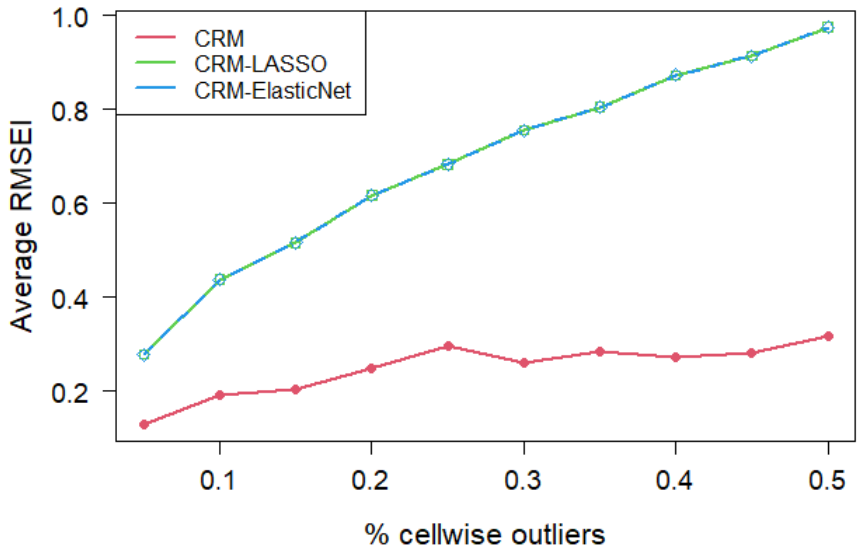

Figure 8: Average RMSEI for CRM, CRM-LASSO and CRM-ElasticNet for percentages of cellwise outliers.

Fig. 8 shows the accuracy of the imputation of the different models under varying percentages

of cellwise outliers. CRM clearly outperforms the other models, which would suggest that CRM is the best performing model regarding imputation. We can conclude that the CRM is the best equipped model to handle different percentages of cellwise contamination. In MSEP en MAE, the three models perform similarly, however, CRM has a lower RMSEI for all percentages of casewise contamination. Suggesting that the predictive performance and regression biases of the models are similar, but CRM is clearly better at imputation. Meaning that the CRM model is preferred.

5.5 Results Real-World Example

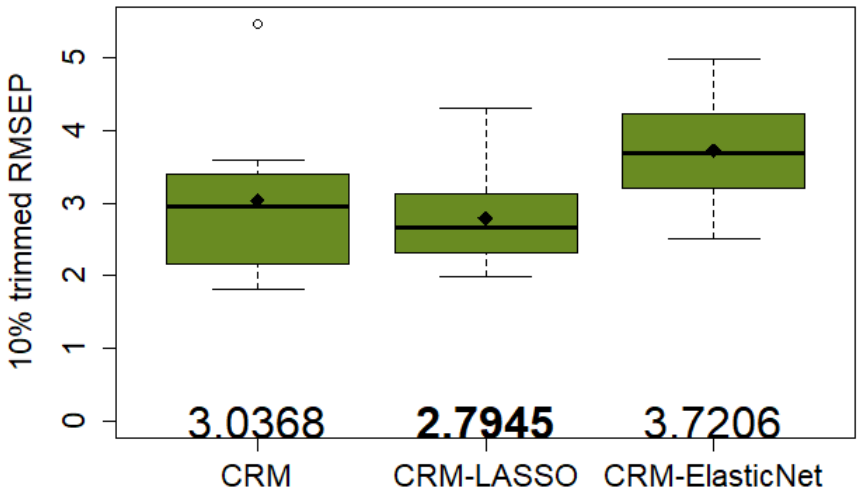

Figure 9: Boxplot of 10% trimmed RMSEP values for the CRM, CRM-LASSO and CRM-ElasticNet.

Last, a 10-fold cross-validation is performed on the real-world data, using CRM, CRM-LASSO and CRM-ElasticNet. The data used in the real-world example is obtained from the Swiss nutrition database (*Nährwerttabele, Infanger E. Schweizer*, 2015). The original data set contains 965 food products, however, since some of the food products contain missing values we only consider the first 193 food products which are complete. We consider the following variables; *energy_kcal*, *protein*, *water*, *carbohydrates*, *sugars*, and *cholesterol* is the response variable. A 10-fold cross-validation allows for a more robust evaluation of model performance. Additionally, it helps understand the bias-variance tradeoff and provides a better estimate of generalization performance. Fig. 9 shows that CRM-LASSO outperforms CRM and CRM-ElasticNet. This could be because it is a robust measure that removes the influence of extreme values. Basically removing one of the weaker points of CRM-LASSO, which could be the reason why CRM-LASSO outperforms CRM.

6 Conclusion

The purpose of this research is to assess which regression method is most effective at estimating the vector regression in SPADIMO used in cellwise robust M regressions under different percentages and types of contamination. CRM-LASSO replaces the SNIPLS regression with a LASSO regression in SPADIMO, which was then used in CRM. Similarly, CRM-ElasticNet replaces the SNIPLS regression with an Elastic Net regression in SPADIMO, which was then also used in CRM. Three main evaluation methods are used, namely MSEP, MAE and RMSEI. The simulation study is performed in three different settings: one with fixed contamination, one with varying magnitude of contamination, and one with varying levels of casewise outliers.

Taking all results previously presented into consideration it can be concluded that the three models have a very similar predictive performance in all the different simulation settings and that the models also have a similar regression bias. However, CRM significantly outperforms the other models at imputation. All things equal, this means that CRM is preferred over CRM-LASSO and CRM-ElasticNet in the different simulation settings tested. What should be noted is that CRM-LASSO is preferred in the real-world example, however, this does not outweigh the performance of CRM in the simulation studies. Thus we can conclude that CRM is the most effective regression method, out of the methods we tested, and that SNIPLS, which is used in CRM, is the best regression at estimating the vector regression in SPADIMO used in cellwise robust M regressions under different percentages and sorts of contamination. Meaning that CRM generally is preferred over CRM-LASSO and CRM-ElasticNet. Mainly because the RMSEI of CRM is in all cases much more desirable, and the results which were desirable for the other models are not significant enough to outperform the overall performance of CRM.

Some limitations of our research should be noted. Our code cannot handle every combination of parameters and observations, the number of observations needs to be at least two times as big as our number of parameters. This means that we could not test the situation where the number of parameters is bigger than the number of observations, in which we would expect the CRM-ElasticNet to outperform the other models. We would have also liked to perform more extensive research on the real-world data, which would have allowed a more in-depth analysis of which cellwise values the models would have identified. However, due to a time shortage, we were not able to do so.

For future research, one could look to implement other sparse regression methods in the SPADIMO algorithm, such as the adaptive LASSO regression (Zou, 2006) or the sparse shooting S regression (Bottmer, Croux & Wilms, 2022). Another idea for future research is to investigate the effect of different weight functions used in SPADIMO. Currently, a Mahalanobis weight function is used, however, the Hampel-Huber or Huber weight functions (Huber, 1981) could also be implemented. This could be interesting because there is no current research done in this field.

References

Bottmer, L., Croux, C. & Wilms, I. (2022). Sparse regression for large data sets with outliers. *European Journal of Operational Research*, *297*(2), 782–794.

Chun, H. & Keleş, S. (2010). Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *72*(1), 3–25.

Debruyne, M., Höppner, S., Serneels, S. & Verdonck, T. (2019). Outlyingness: Which variables contribute most? *Statistics and Computing*, *29*, 707–723.

Filzmoser, P., Höppner, S., Otner, I., Serneels, S. & Verdonck, T. (2020). Cellwise robust m regression. *Computional Statistics Data Analysis*, *147*.

Hoffmann, I., Filzmoser, P., Serneels, S. & Varmuza, K. (2016). Sparse and robust pls for binary classification. *Journal of Chemometr*, *30*(4), 153–162.

Huang, J., Ma, S. & Zhang, C.-H. (2008). Adaptive lasso for sparse high-dimensional regression models. *Statistica Sinica*, *18*(4), 1603–1618.

Huber, P. J. (1981). *Robust statistics*. Wiley.

Maronna, R. A., Martin, R. D., Yohai, V. J. & Salibián-Barrera, M. (2019). *Robust statistics: theory and methods (with r)*. John Wiley & Sons.

McDonald, G. C. (2009). Ridge regression. *Wiley Interdisciplinary Reviews: Computational Statistics*, *1*(1), 93–100.

Nährwerttabele, infanger e. schweizer. (2015). Retrieved from `http://www.sge-ssn.ch/shop/produkt/schweizer-naehrwerttabelle/`

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *58*(1), 267–288.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, *101*, 1418–1429.

Zou, H. & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, *67*(2), 301–320.

A Code Overview

All R scripts below are the newly introduced scripts which are not already in the 'crmReg' R package. The first five scripts are dedicated to the replication of the results from this paper. The next six scripts are dedicated to the contribution of this research.

CRM_simulations_part1.R, includes the simulation study with the fixed parameters. It then calculates the average Mean Squared Error of Prediction (MSEP), Mean Absolute Error (MAE), and Rooted Mean Squared Error of Imputation (RMSEI) over a hundred simulations for the CRM, CRM-LASSO and CRM-ElasticNet.

CRM_simulations_part2.R, includes the simulation study where the level of contamination is varied. So the MSEP, MAE and RMSEI are calculated under a varying parameter k, and are then reported for the CRM, CRM-LASSO and CRM-ElasticNet.

CRM_simulations_part3.R, includes the simulation study in which the percentage of casewise contamination is varied. So, the MSEP, MAE and RMSEI are again calculated for different percentages of casewise contamination and are then reported for the CRM, CRM-LASSO and CRM-ElasticNet.

CRM_simulations_part4.R, includes the simulation study in which the percentage of cellwise contamination is varied. So, again, the MSEP, MAE and RMSEI are calculated for different percentages of cellwise contamination and are then reported for the CRM, CRM-LASSO and CRM-ElasticNet.

Nutrients.R, includes an R script in which a 10-fold cross-validation is performed, with as data the Swiss nutrients data.

crm_lasso.R, is an altered version of crm.R file, where instead of the normal spadimo.R is used the spadimo_lasso.R is used.

crm_enet.R, is an altered version of crm.R file, where instead of the normal spadimo.R is used the spadimo_enet.R is used.

spadimo_lasso.R, is an altered version of spadimo.R where the function spadimo.exs is replaced by a new function, which uses a LASSO regression instead of a SNIPLS.

spadimo_enet.R, is an altered version of spadimo.R where the function spadimo.exs is replaced by a new function, which uses a Elastic Net regression instead of a SNIPLS.

predict.crm_lasso.R, is an altered version of predict.crm.R and includes the predict function for the fitted CRM-LASSO model.

predict.crm_enet.R, is an altered version of predict.crm.R and includes the predict function for the fitted CRM-ElasticNet model.

B Appendix

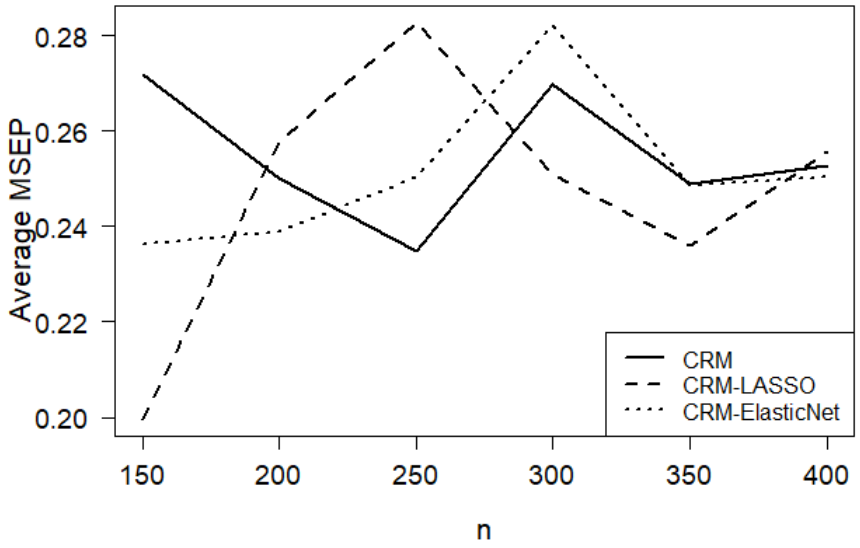

Figure 10: Average MSEP for each of the regression methods under different parameter observations ratios.

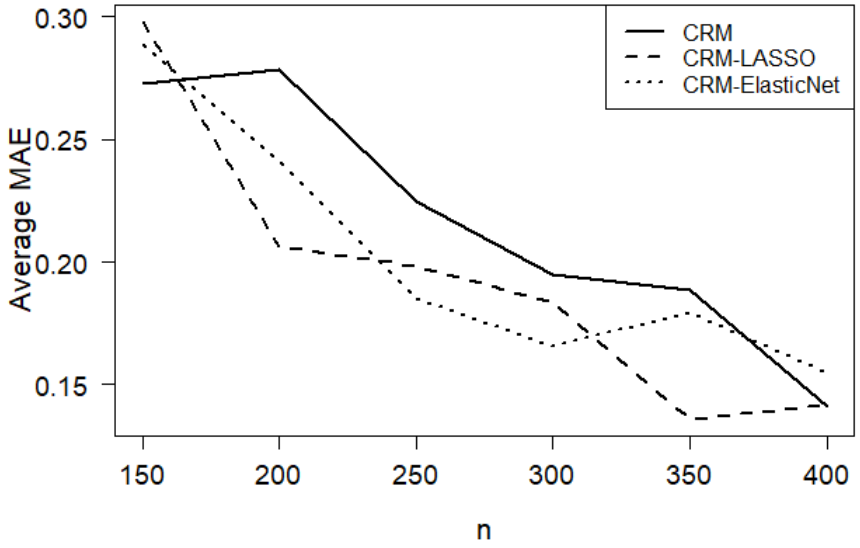

Figure 11: Average MAE for each of the regression methods under different parameter observations ratios.

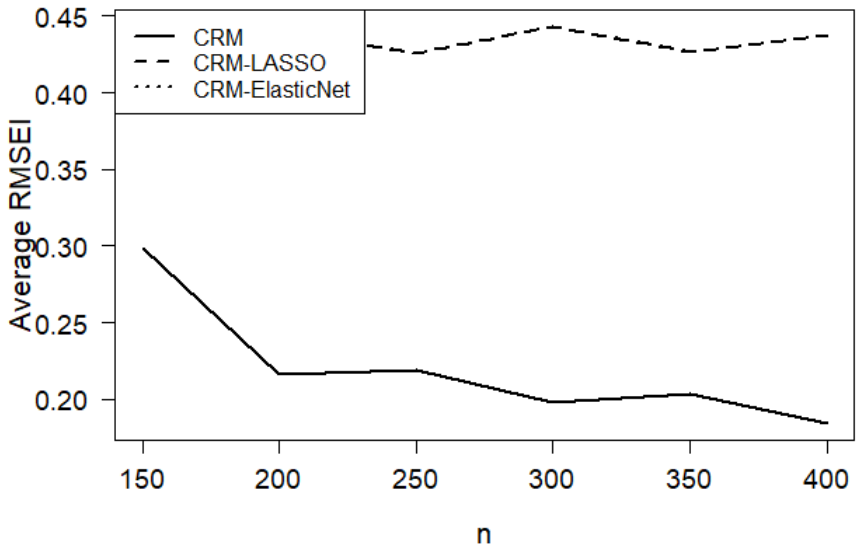

Figure 12: Average RMSEI for each of the regression methods under different parameter observations ratios.