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

The evaluation of weight functions in Cellwise Robust M regression

Jaco de Hoog (572567jh)

Abstract

In robust regression outlying observations are usually downweighted to limit their influence on the estimation of the regression coefficients. However, downweighting whole observations could cause a significant information loss, because often only deviations in some of the variables determine the outlyingness of an observation. The cellwise robust M (CRM) regression estimator uses the method of Sparse Directions of Maximal Outlyingness (SPADIMO) to identify the cells contributing most to the outlyingness of an observation. These cells are treated as if they are missing and are imputed with the column means of the two nearest neighbors. In the CRM regression algorithm a weight function is used to assign outlying observations to SPADIMO. Hence, the choice of weight function together with the selection of its parameters is of great importance, because it determines how many of the outlying cells SPADIMO can possibly identify. This empirical research focuses on the use of various weight functions in CRM regression and how the use of different parameters for the weight functions influences the performance of CRM regression. The results of the simulation studies show that there is no specific weight function that can best be used in CRM regression when the amount of contamination is limited. However, one could consider using the Huber weight in CRM regression when more contamination is expected. Furthermore, it becomes clear that selecting quality parameters for the weight functions leads to more predictive power and higher estimation accuracy. Lowering the parameters leads to more robustness when the amount of casewise contamination increases, but it depends on the weight function how much the parameters should be lowered.

Supervisor:	dr. Aurore Archimbaud
Second assessor:	dr. Kathrin Gruber
Date final version:	30th June 2023

Contents

1 Introduction

The method of ordinary least squares (OLS) has favourable properties when its assumptions hold. If the underlying assumptions of OLS do not hold, the estimated coefficients are biased and inconsistent (Heij et al., 2004). Hence, robust regression estimators are more frequently preferred, because robust regression estimators provide accurate coefficient estimates despite the violation of the underlying assumptions. The presence of outliers causes the data to deviate from the underlying assumptions. Here we can distinguish between casewise outliers and cellwise outliers, where casewise outliers are considered to be whole outlying observations and cellwise outliers are the cells in the data matrix contributing to the outlyingness of an observation.

Filzmoser et al. (2020) have introduced the cellwise robust M (CRM) regression estimator which uses the method of Sparse Directions of Maximal Outlyingness (SPADIMO) (Debruyne et al., 2019) to identify cellwise outliers. The observations assigned to SPADIMO are casewise outliers for which SPADIMO determines which variables contribute most to the outlyingness. CRM uses weight functions in order to classify observations as casewise outliers. If the residuals obtained by performing MM-estimation exceed a certain parameter value, the observations corresponding to these large residuals get assigned a lower weight and are further investigated by the method of SPADIMO. By only downweighting the cellwise outliers identified by the method of SPADIMO the CRM regression estimator is considered to be cellwise robust.

The weight function used in CRM regression is of great importance in identifying the cells contributing most to the outlyingness. The used weight function together with the selected parameters for the weight function determine how many observations are classified as casewise outliers and thus how many outlying cells the method of SPADIMO can possibly identify. Besides, the weights assigned by the weight function are used by SPADIMO to calculate the robust Mahalanobis distance, such that the outlyingness of the cells can be derived. As a result, the choice of a weight function together with the selection of the parameters indirectly influences the predictive performance and estimation accuracy of the CRM regression estimator.

In this research different weight functions are considered, where the predictive performance, estimation accuracy, and ability to identify cellwise outliers is evaluated for CRM using these weight functions. The aim of this research is to answer the following central research question: *Which weight function can best be used in cellwise robust M regression and how can the most optimal parameters for the weight functions be selected?*

With the findings of this research insight into different weight functions is provided. It becomes clear how the different weight functions have their impact on the identification of outliers and how they indirectly influence the estimation accuracy of CRM. Based on the results, further research can make an informed choice which weight function to use in CRM regression. Besides, by considering various parameter values for the weight functions, it is possible to select quality parameters based on how many outliers are expected in the data.

In order to answer the central research question, various simulation studies are considered. The results of these simulation studies show that there is no specific weight function that can best be used in CRM regression when the amount of contamination is limited. However, it has become clear that the standard normal quantiles are not the most optimal parameters for the Hampel weight function, because using parameters corresponding to 95% efficiency leads to

more predictive power and higher estimation accuracy. For increasing amounts of contamination one could consider the Huber weight function, since CRM using this weight function shows less breakdown behavior. Using lower parameter values for the weight functions increases the robustness, but it differs per weight function how much the parameters should be lowered.

Besides the various simulation studies, we have evaluated the performance of CRM using different weight functions in a real data application. From this application it has become clear that the Generalized Gauss weight function may be unsuitable to be used in CRM regression. This weight function only assigns observations to SPADIMO for really large residual values, as a result of which the influence of outliers can not be limited if their magnitude is small.

This research is structured as follows. First, in Section 2 we further introduce our research on the basis of earlier work. Section 3 describes CRM regression, the method of SPADIMO, and the use of weight functions. After that, the simulation studies together with the corresponding results are presented in Section 4. Then, in Section 5 the performance of CRM using different weight functions is evaluated in a real data application. Section 6 concludes with the most important findings and suggestions for future research.

2 Literature Review

To estimate the linear relation between two or more variables the method of ordinary least squares (OLS) can be used. The least squares estimator is the best linear unbiased estimator when the normality assumptions are satisfied (Heij et al., 2004). It follows that the least squares estimator is not optimal when data deviate from the assumptions.

A possible cause of the data deviating from the assumptions of OLS is the presence of outliers. Grubbs (1969) defines an outlier as "an observation that deviates markedly from other members of the sample in which it occurs". One can distinguish between casewise and cellwise outliers. When casewise outliers are considered, one assumes that the outliers are complete observations of a multivariate predictor. However, often only a few entities of an observation deviate from the general pattern in the data. Discarding whole cases in such situations causes significant information loss, which can lead to an increase in estimation variance. The so-called cellwise outliers were first properly introduced by Alqallaf et al. (2009). Since then, more and more research has been done on estimators that are robust against cellwise outliers.

In order to control cellwise outliers, Öllerer et al. (2016) have introduced the shooting S-estimator. This estimator uses the idea of the coordinate descent algorithm, which is also known as the shooting algorithm (Fu, 1998). In the shooting algorithm, variable by variable, simple lasso regression is performed. Öllerer et al. (2016) replace the lasso estimator with the unpenalized S-estimator (Maronna et al., 2006) to obtain robustness. Another cellwise robust estimator is introduced by Filzmoser et al. (2020). Their cellwise robust M (CRM) regression estimator consists of an iteratively reweighted least squares procedure that starts with weights derived from highly robust estimates. These estimates compensate for both casewise vertical outliers and leverage points. To detect cells that contribute most to outlyingness, the CRM estimator uses the method of Sparse Directions of Maximal Outlyingness (SPADIMO) (Debruyne et al., 2019). SPADIMO is applied in each iteration, after which the reweighting scheme is then adapted to only downweight outlying cells. This leads to a highly robust regression estimate which is

obtained by applying cellwise outlier detection.

CRM regression uses an MM-estimation procedure, where a highly robust but inefficient S-estimator (Rousseeuw & Yohai, 1984) is used as a starting point for the robust M-estimator. If the obtained residuals by performing MM-estimation are large, a weight function indirectly limits the influence of the corresponding outlying observations. Huber (1964), Beaton and Tukey (1974), and Hampel et al. (1986), among others, have proposed weight functions, where the functions assign lower weights when the residual values exceed a certain parameter value.

If an observation gets assigned a lower weight, the CRM algorithm considers this observation as a casewise outlier. Each casewise outlier is allocated to the method of SPADIMO in order to detect the cells contributing most to the outlyingness of the observation. Using lower parameter values for the weight functions leads to more observations being classified as casewise outlier, as a result of which SPADIMO can possibly identify more outlying cells. Hence, the use of lower parameter values makes the CRM regression algorithm more robust. Unfortunately, this is accompanied with lower efficiency.

The efficiency measures the quality of an estimator. An efficient estimator is an estimator that minimizes the variance (Heij et al., 2004). Especially, the variance approaches the Cramér-Rao lower bound. This means that the estimator is the most precise and reliable in estimating the coefficients. The estimated coefficients are the closest to the true values assuming that the true values are known.

The robustness of an estimator is characterized by the breakdown point. This breakdown point represents the proportion of outlying observations that an estimator can handle without causing a strongly deviating estimate (Donoho & Huber, 1983). A higher breakdown point means that the estimator is more robust against outlying observations. The breakdown point depends on the properties of the objective function. In case of the Huber weight function (Huber, 1964) the corresponding objective function is unbounded, as a result of which the breakdown point is 50% (Huber, 1984). The other four weight functions considered in this research have monotonic and bounded objective functions. It is shown by Huber (1984) that it is more complicated to derive the breakdown point of a bounded objective function, because in this case the breakdown point depends on the shape of the objective function and its parameters. He et al. (1990) show that the minimal amount of contamination that drives the estimator beyond all bounds determines the breakdown point of these objective functions.

3 Methodology

In Section 3.1 it is explained how the cellwise robust M (CRM) regression algorithm works. The method of Sparse Directions of Maximal Outlyingness (SPADIMO) and the use of weight functions in CRM regression are discussed in more detail in Sections 3.2 and 3.3, respectively.

3.1 Cellwise Robust M Regression

The cellwise robust M (CRM) estimator is an estimator for the linear model that is robust against cellwise outliers and it yields a map of the detected outlying cells. Consider the linear model $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\boldsymbol{y}$ represents the dependent variable, $\boldsymbol{X}$ is the data matrix containing the

explanatory variables, and $\boldsymbol{\epsilon}$ represents the error term. To obtain an estimate of the regression coefficients vector $\boldsymbol{\beta}$ one can apply ordinary least squares (OLS). In CRM regression, however, an M-estimator is used. For the CRM estimator to be robust against both vertical outliers (i.e. outliers in the dependent variable) and leverage points (i.e. outliers in the explanatory variables) a robust starting estimator for the M-estimator must be selected. For this, a highly robust but inefficient S-estimator (Rousseeuw & Yohai, 1984) can be used.

The resulting MM-estimator can be seen as the starting point of the CRM regression algorithm. By performing MM regression, residuals are obtained. Observations are downweighted when the absolute value of the residual exceeds a certain parameter value of the weight function. If the assigned weight is lower than the weight threshold, the observation is considered to be a casewise outlier. Filzmoser et al. (2020) use a weight threshold equal to one. By lowering the weight threshold we can control the amount of observations that gets classified as outlying.

The observations with an assigned weight lower than the weight threshold are allocated to the method of Sparse Directions of Maximal Outlyingness (SPADIMO), which identifies the variables contributing most to the outlyingness of an observation. Due to SPADIMO, we can consider cellwise outliers instead of treating whole observations as casewise outliers. To limit the influence of these cellwise outliers, the corresponding cells are imputed with values based on the column means of the two nearest neighbors that are part of the 'clean' cells.

An iteratively re-weighted least squares (IRLS) (Green, 1984) procedure is applied to obtain more efficient estimates of the regression coefficients. Here, the coefficients obtained by MM-estimation are used as starting values. In each step of the IRLS process, a similar approach as described in the previous paragraph is performed. This procedure stops when the mean absolute difference of the subsequent coefficient estimates is smaller than a certain tolerance bound.

3.2 Sparse Directions of Maximal Outlyingness

Not all outliers deviate along all variables. It could be the case that an observation only deviates for some of the variables. Debruyne et al. (2019) developed the method of Sparse Directions of Maximal Outlyingness (SPADIMO) in order to detect the variables with the biggest impact on the outlyingness of an observation. Instead of controlling each outlier the same and downweighting whole observations, SPADIMO adjusts atypical values of outlying variables. This way, the valuable information contained in the non-outlying variables is preserved.

SPADIMO uses the weights assigned by the weight function to determine a weighted mean and weighted covariance matrix. By calculating the robust Mahalanobis distance using this mean and covariance matrix, the outlyingness of a point can be determined. It is proven by Debruyne et al. (2019) that the outlyingness of any point $\boldsymbol{x} \in \mathbb{R}^p$ is equal to the solution of the following maximization problem:

$$r(\boldsymbol{x}; \boldsymbol{X}) = \max_{\boldsymbol{a} \in \mathbb{R}^p, \|\boldsymbol{a}\|=1} \frac{|\boldsymbol{x}^\intercal \boldsymbol{a} - \hat{\boldsymbol{\mu}}_w^\intercal \boldsymbol{a}|}{\sqrt{\boldsymbol{a}^\intercal \hat{\boldsymbol{\Sigma}}_w \boldsymbol{a}}}. \tag{1}$$

Here, $\hat{\boldsymbol{\mu}}_w$ and $\hat{\boldsymbol{\Sigma}}_w$ represent the weighted mean and weighted covariance matrix, respectively. The value of $\boldsymbol{a}$ that obtains the maximum for Equation 1 is the 'direction of maximal outlyingness'. This direction of maximal outlyingness is used to determine which variables contribute the most

to the outlyingness of an observation. As a result, the method of SPADIMO is able to adjust outlying cells instead of downweighting whole observations.

The direction of maximal outlyingness can also be expressed as a normalized least squares problem. An initial estimate is needed for this least squares problem in order to derive the direction of maximal outlyingness. SPADIMO applies sparse partial least squares (SPLS) regression (Chun & Keleş, 2010) to obtain an initial estimate with the capability of producing a sparse vector of regression coefficients. This vector provided by SPLS contains a subset of zero entries due to a sparsity penalty η that is applied to the weighting vector when $\eta > 0$. The direction obtained by performing SPLS, $\boldsymbol{a}(\eta, \boldsymbol{x})$, is not the same as the direction $\boldsymbol{a}$ that maximizes Equation 1. However, when $\eta \to 0$ the direction $\boldsymbol{a}(\eta, \boldsymbol{x})$ converges to $\boldsymbol{a}$.

A grid-search over the interval $[0, 1)$ is performed to derive the optimal sparsity parameter η. For $\eta = 0$ all variables are included in the model, while for η close to 1 nearly all variables are set equal to zero. The optimal value for η is the one for which the minimal amount of variables are removed from the model such that it is not outlying anymore.

It should be noted that it is not possible to calculate the robust Mahalanobis distance if the number of variables p exceeds the number of observations n. In this case, some of the eigenvalues are equal to zero, which results in a non-existing covariance matrix. Hence, when SPADIMO is applied in CRM, robust principal component analysis (PCA) (Hubert et al., 2005) is used in situations where $p > n$.

3.3 Weight Functions

In CRM regression, M-estimation is performed where a S-estimator is used as a starting estimator. The class of M-estimators is given by

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_i \rho\left(\frac{r_i(\boldsymbol{\beta})}{\hat{\sigma}}\right), \tag{2}$$

where $r_i(\boldsymbol{\beta})$ is the i-th regression residual, $\hat{\sigma}$ is a robust scale estimator of the residuals, and ρ is the objective function. The robustness properties of the M-estimator are dependent of the derivative of the objective function ρ, $\rho' = \psi$. There are several robust estimators for the influence function ψ. The weight function corresponding to a certain influence function is obtained by dividing the influence function by the residual value r. The CRM regression algorithm uses a weight function to assign outlying observations to the method of SPADIMO. Hence, it is important to select quality parameters for the weight function in order to smoothen the process of identifying outlying cells.

Filzmoser et al. (2020) use the Hampel redescending function (Hampel et al., 1986), which reweighting representation is given by

$$w_{HA}(r) = \begin{cases} 1, & \text{if } |r| \leq Q_1 \\ \frac{Q_1}{|r|}, & \text{if } Q_1 < |r| \leq Q_2 \\ \frac{Q_3 - r}{Q_3 - Q_2}\frac{Q_1}{r}, & \text{if } Q_2 < |r| \leq Q_3 \\ 0, & \text{if } |r| > Q_3 \end{cases}. \tag{3}$$

The Hampel weight function depends on the parameters Q_1, Q_2, and Q_3. This weight function is applied to regression residuals, for which it can be assumed that they are standard normally distributed. For this reason, Filzmoser et al. (2020) use the 0.950, 0.975, and 0.999 quantiles of the standard normal distribution.

Another popular objective function used for redescending M-estimators is Tukey's bisquare function (Beaton & Tukey, 1974). The influence function of Tukey's bisquare function vanishes for values of Q outside the interval $[-Q, +Q]$. Hence, Q is the optimal tuning parameter in the weight function

$$w_T(r) = \begin{cases} \left(1 - (\frac{r}{Q})^2\right)^2, & \text{if } |r| \leq Q \\ 0, & \text{if } |r| > Q \end{cases}. \tag{4}$$

In this case the constant $Q = 4.685$ obtains 95% efficiency of the regression estimator.

The objective function introduced by Huber (1964) has least squares behavior for small residuals. For large residuals the Huber function has hyperbolically decreasing weights, as becomes clear from the weight function given by

$$w_{HU}(r) = \begin{cases} 1, & \text{if } |r| \leq Q \\ \frac{Q}{|r|}, & \text{if } |r| > Q \end{cases}. \tag{5}$$

For $Q = 1.345$ 95% efficiency of the regression estimator is obtained.

Besides, the Generalized Gauss weight function (Koller & Stahel, 2011) is considered. This function has the property of reaching 0 only asymptotically and it is possible to fix the maximal rate of descent. It is defined by

$$w_{GGW}(r) = \begin{cases} 1, & \text{if } |r| \leq Q \\ exp\Big(-\frac{1}{2}\frac{(|r|-Q)^b}{a}\Big), & \text{if } |r| > Q \end{cases}, \tag{6}$$

and the parameters $a = 1.387$, $b = 1.5$, and $Q = 1.063$ lead to a 95% efficient estimator.

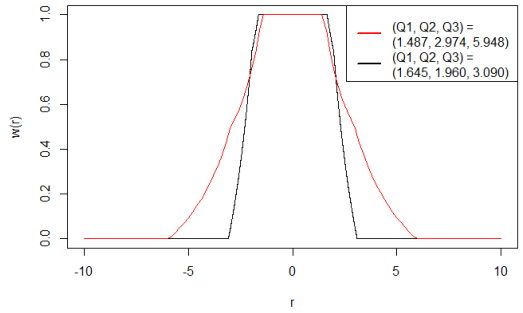
Koller and Stahel (2011) proposed the linear quadratic quadratic function. This function is slowly redescending and obtains more accurate results than other redescending estimators in settings with many variables. Its reweighting representation is given by

$$w_{LQQ}(r) = \begin{cases} 1, & \text{if } |r| \leq Q1 \\ 1 - \frac{S}{2Q_2}\frac{(|r|-Q_1)^2}{|r|}, & \text{if } Q_1 < |r| \leq Q_1 + Q_2 \\ \frac{1}{|r|}(Q_1 + Q_2 - \frac{Q_2 S}{2} + \frac{S-1}{Q_3}(\frac{1}{2}\tilde{r}^2 - Q_3\tilde{r})), & \text{if } Q_1 + Q_2 < |r| \leq Q_1 + Q_2 + Q_3 \\ 0, & \text{if } |r| > Q_1 + Q_2 + Q_3 \end{cases}, \tag{7}$$

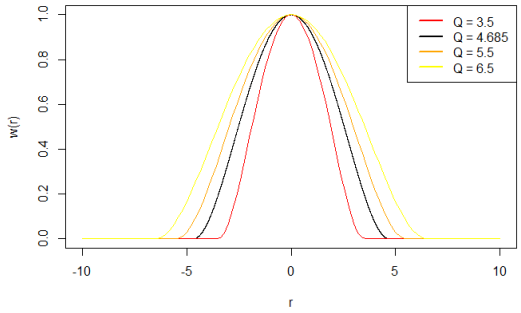
where $\tilde{r} := |r| - Q_1 - Q_2$ and $Q_3 := \frac{2Q_1 + 2Q_2 - Q_2 S}{S-1}$. The maximal rate of descent is controlled by S and it holds that $S = \frac{Q_2}{Q_1}$. It follows that the parameters $Q_1 = 0.982$, $Q_2 = 1.473$, and $S = 1.5$ provide 95% efficiency.

In Figure 1 the described weight functions for different parameter values are presented. In case of the Hampel weight function, the parameters represented by the black line are the 0.950, 0.975, and 0.999 quantiles of the standard normal distribution. The black lines in the other

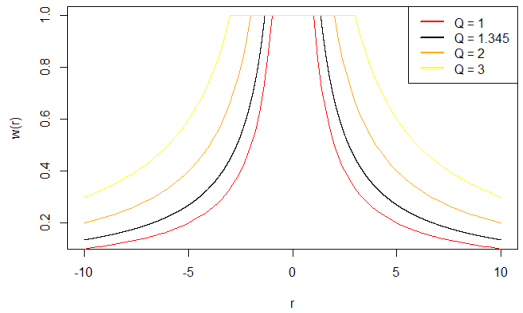
graphs represent the parameters for which a 95% efficient regression estimator is obtained. If the parameters are set lower, the behavior of the different weight functions is shown by the red lines. The orange and yellow lines are used for parameter values above the initial parameter values, such that we can distinguish between parameter values below and above the initial parameters by using different color shades. Only in case of the Hampel weight function there is only one additional combination of parameter values presented. The shape of this weight function does not really change when the parameters are gradually decreased or increased. Hence, the red line shows a combination with bigger jumps between the parameter values to show that the shape of the Hampel weight function depends on the slope between the parameter values.



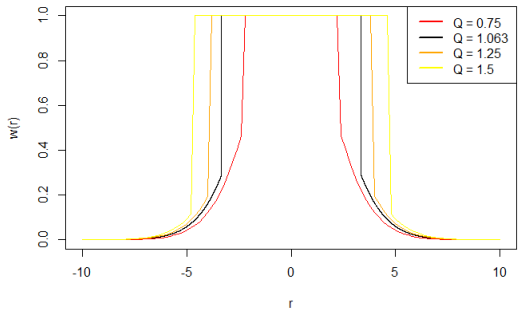
(a) Hampel weight function.



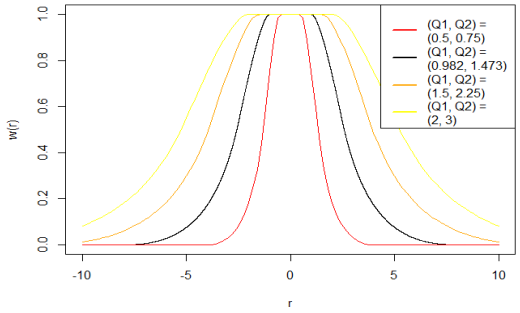
(b) Tukey's bisquare weight function.



(c) Huber weight function.



(d) Generalized Gauss weight function.



(e) Linear quadratic quadratic weight function.

Figure 1: Various weight functions for different parameter values.

From Figure 1 it becomes directly clear that for all weight functions a weight close or equal to zero is more quickly assigned when the parameter values are lower. However, the weight functions have contrasting behavior. The Hampel, Huber, Generalized Gauss, and linear quadratic quadratic weight functions assign a weight equal to one for multiple residual values, while the Tukey's bisquare weight function only assigns a weight of one when the residual value is equal to zero. We notice that the Tukey's bisquare and linear quadratic quadratic weight functions have a similar shape and that the parameter values determine the slope and thus how quickly

a zero weight is assigned. Also the Hampel and Huber weight functions have similar shapes. However, it is immediately observable that the Hampel weight function allows more fine tuning due to the parameters determining the shape of the function. The Generalized Gauss weight function only assigns lower weights for relatively large residual values. For this weight function, it is notable that the descent becomes steeper when Q increases. It should be noted that in Figure 1e the values of a and b are held constant at 1.387 and 1.5, respectively. Hence, the shape of the Generalized Gauss weight function can change when different values for a and b are used.

4 Simulation Studies

In this section we describe the simulation studies and the corresponding results. First, in Section 4.1 we introduce the default simulation setting which forms the basis for the different simulation studies. After that, an initial comparison of CRM using the different weight functions is performed in Section 4.2. Section 4.3 shows the performance evaluation of CRM where the parameters for the weight functions are selected based on a certain efficiency level or breakdown point. Furthermore, Section 4.4 presents how CRM using different weight functions handles increasing amounts of contamination. Lastly, in Section 4.5 we propose to lower the parameter values in order to increase the robustness of CRM regression.

The source code of the simulation studies can be found on `https://github.com/j4c0d3h00g/CRMsimulations`. In these simulations our developed R package `CRMwf` is used which can be downloaded from `https://github.com/j4c0d3h00g/CRMwf`. The package `CRMwf` builds upon the `crmReg` package of Filzmoser et al. (2020).

4.1 Default Simulation Setting and Performance Evaluation

The default setting for our simulation studies is similar to the simulation setting in the research of Filzmoser et al. (2020). Hence, also in our research the data are generated from a p-dimensional multivariate normal distribution with center $\boldsymbol{\mu} = (0, ..., 0)^\mathsf{T}$ and covariance matrix $\boldsymbol{\Sigma}$. Here, $\boldsymbol{\Sigma}_{i,i} = 1$ for $i = 1, ..., p$, $\boldsymbol{\Sigma}_{j,j+1} = \boldsymbol{\Sigma}_{j+1,j} = 0.5$ for $j = 1, ..., p-1$, and $\boldsymbol{\Sigma}$ is zero otherwise. In total there are $n = 400$ cases generated with $p = 50$ corresponding variables, which results in the data matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$. From the data matrix $\boldsymbol{X}$ 20 rows are selected, where in each of these 20 rows 5 cells are contaminated. Hence, in total there are 100 cells contaminated. In this contaminated matrix $\boldsymbol{X}^c$, the cell corresponding to variable j of observation i is contaminated according to

$$x_{ij}^c = \bar{x}_j + k s_j + e = \bar{x}_j + k \sqrt{\frac{1}{n-1} \sum_{l=1}^{n} (x_{lj} - \bar{x}_j)^2} + e, \tag{8}$$

where $k = 6$ times the standard deviation s_j and the random standard normal distributed value e are added to the mean value $\bar{x}_j$. However, contrary to the research of Filzmoser et al. (2020), the data are generated beforehand instead of in each replication of the simulation study. In this way, certain functions used in the simulation can not change the seed of the generator providing the random drawings from the multivariate normal distribution.

To evaluate the relative performance of using the different weight functions in CRM regression we use several criteria. The mean squared error of prediction (MSEP) is used to evaluate

the predictive performance.

$$\text{MSEP} = \frac{1}{n_{\text{clean}}} \sum_{i \in I} (\hat{y}_i - y_i)^2, \tag{9}$$

where I is the set of clean, uncontaminated cases and n_{clean} is the number of uncontaminated observations.

The mean absolute error (MAE), given by

$$\text{MAE} = \frac{1}{p} \sum_{j=1}^{p} |\hat{\beta}_j - \beta_j|, \tag{10}$$

is used to evaluate how much the individual regression coefficients differ from the true values.

Besides, we use the precision and recall to evaluate how well CRM using the different weight functions is able to detect the cellwise outliers. The precision measures how many of the cells classified as cellwise outliers are actually cellwise outliers, while the recall measures how many of the cellwise outliers are actually classified as cellwise outliers.

4.2 Initial Comparison Weight Functions

To show that it could be valuable to use other weight functions, we perform an initial comparison where different weight functions are used in CRM regression. For this part of our research we use the default simulation setting described in Section 4.1. In total we perform hundred replications of applying CRM using different weight functions. The performance is evaluated based on the averaged MSEP, MAE, precision, and recall across the hundred replications.

In this initial comparison the parameter values stated in Section 3.3 are used. Hence, in case of the Hampel weight function we use the 0.950, 0.975, and 0.999 quantiles of the standard normal distribution, while for the other four weight functions parameters corresponding to 95% efficiency are used.

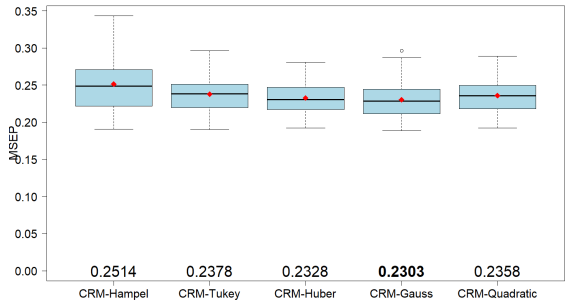
Filzmoser et al. (2020) use a weight threshold equal to one, as a result of which the CRM regression algorithm only classifies an observation as casewise outlier when the assigned weight is smaller than one. As mentioned earlier in Section 3.3, the Tukey's bisquare weight function only assigns a weight of one when the residual value is equal to zero. Hence, nearly all observations are identified as casewise outliers, because the residual value is almost never exactly zero. To overcome this problem, the weight threshold for which an observation is classified as casewise outlier is lowered to 0.7 for Tukey's bisquare weight function, such that this weight function identifies a similar amount of casewise outliers as the other weight functions.

In Figure 2 the performance evaluation results for CRM using different weight functions are presented. The results are illustrated as boxplots with the average result across hundred iterations shown at the bottom. The best average result among the different weight functions used in CRM regression is shown in bold. Strong deviations from the general pattern in performance evaluation scores are represented by the white dots, while the median and average are shown by the bold black line and the red diamond shape, respectively.

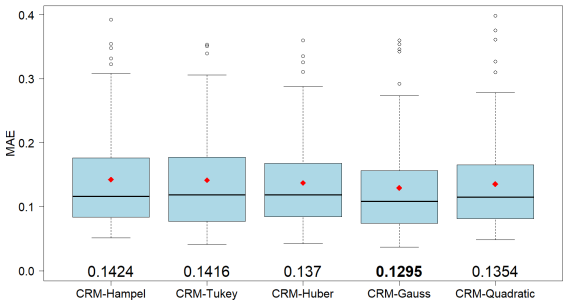
From Figure 2a it becomes clear that CRM using the Generalized Gauss weight function provides the most accurate predictions. However, the average MSEP for CRM using the Tukey's bisquare, Huber, and linear quadratic quadratic weight functions does not differ that much from

the MSEP for CRM using the Generalized Gauss weight function. It is striking that CRM using the Hampel weight function has the least predictive power. The average MAE values for CRM using the different weight functions do not differ much from each other, as shown by Figure 2b. The lowest average MAE is obtained by CRM using the Generalized Gauss weight function.

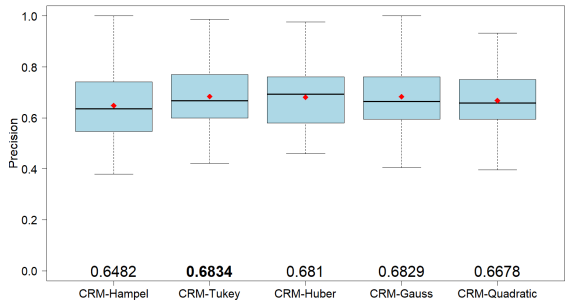
When looking at the precision and recall in Figures 2c and 2d, respectively, it is directly noticeable that the recall is relatively high and does not really differ at all across the different weight functions used in CRM regression. The recall is around 0.89 for all cases, which means that CRM is able to detect about 89% of the cellwise outliers, regardless of the used weight function. CRM using Tukey's bisquare weight function identifies the smallest amount of actual cellwise outliers, because it achieves the lowest recall score. If the weight threshold is set higher for this weight function, it could be possible that more of the actual cellwise outliers are identified. However, this could lead to a lower precision, because CRM using Tukey's bisquare weight function would classify more cells as outlying where a large part of these cells is not an actual cellwise outlier. In general, CRM classifies more outlying cells than there actually are, as shown by the low precision scores. The highest average value for the precision is obtained by CRM using the Tukey's bisquare weight function. We notice that the precision score for the Hampel weight functions is relatively low in comparison with the other weight functions.



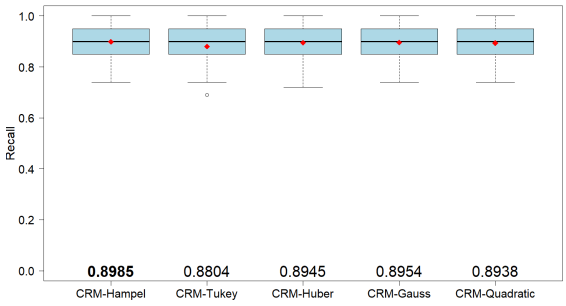
(a) Boxplot for the MSEP results of CRM.



(b) Boxplot for the MAE results of CRM.



(c) Boxplot for the precision results of CRM.



(d) Boxplot for the recall results of CRM.

Figure 2: Relative performance evaluation results for CRM using different weight functions.

Based on the results shown in Figure 2 we can not conclude that there is a certain weight function that can best be used in CRM regression. There is no weight function that clearly achieves the best results. CRM using the Hampel weight function with the 0.950, 0.975, and 0.999 quantiles of the standard normal distribution as parameters obtains the worst results. However, it could be possible that better results are obtained when other parameters are used.

It should be noted that the values of the MSEP, MAE, precision, and recall for CRM using the Hampel weight function with the standard normal quantiles as parameters differ from the

results of CRM in the research of Filzmoser et al. (2020). This difference exists due to fact that the data in our research are generated beforehand instead of during each replication.

4.3 Efficiency-Robustness Tradeoff

To investigate the effects of both the efficiency and the breakdown point of an estimator we evaluate CRM using the different weight functions for various efficiency levels and breakdown points. In this way, we show how selecting the parameters for the weight functions based on a certain efficiency level or breakdown point affects the performance of CRM regression.

The data are generated in a similar way as described in Section 4.1 and again we use the MSEP, MAE, precision, and recall to evaluate the relative performance. However, instead of hundred replications, we now perform fifty replications. In case of comparing the different efficiency levels, the parameters for the weight functions are derived based on a 85, 90, 95, 97.5, and 99% efficiency level. When evaluating the performance for different breakdown points, we consider the 10, 20, 30, 40, and 50% breakdown points. The Huber weight function is omitted when evaluating the different breakdown points, because this weight function has a fixed breakdown point of 50% and thus we are unable to derive parameters corresponding to other breakdown points.

Tables 1 and 3 show the MSEP, MAE, precision, and recall of CRM using different weight functions for various efficiency levels and breakdown points, respectively. The MSEP, MAE, precision, and recall values are the average value across fifty replications. The values of the performance measure in bold represent the best result for each weight function. Besides, when the value of the performance measure is also underlined, the value is the best result across the different weight functions. In Section A of the Appendix the parameters corresponding to a certain efficiency level or breakdown point can be found.

From Table 1 it becomes clear that the lowest values for the MSEP and MAE are obtained when parameters are used that correspond to high efficiency levels. For almost all weight functions the lowest value for the MSEP or MAE is obtained for efficiency levels higher than 95%. In case of the MAE, it differs per weight function which efficiency level achieves the lowest value. Hence, it is difficult to conclude which specific efficiency level can best be used to select the parameters for the weight functions in CRM regression. However, for all the weight functions we recommend to use parameters corresponding to at least 95% efficiency, because for efficiency levels higher than 95% the lowest values for the MSEP and MAE are obtained.

The precision and recall scores present how many of the actual cellwise outliers are identified and whether there are too many cells classified as outlying. It is noticeable that the highest recall scores are achieved for parameters corresponding to low efficiency levels. This makes sense, because the parameters are lower and therefore more observations are flagged as casewise outliers. As a result, SPADIMO considers more observations and identifies more cells as outlying. However, this is also accompanied with the fact that too many cells are identified as outliers, as becomes clear from the precision scores shown in Table 1. From the precision scores it becomes clear that CRM is the most accurate in classifying cells as outlying when parameters corresponding to a high efficiency level are used. Together with the small differences in recall scores, these results emphasize that it is recommended to use parameters corresponding to at

least 95% efficiency for the weight functions when applying CRM regression.

Table 1: MSEP, MAE, precision, and recall of CRM using different weight functions for various efficiency levels.

Weight function	Efficiency	MSEP	MAE	Precision	Recall
Hampel	85.0%	0.2504	0.1400	0.6224	**0.9024**
	90.0%	0.2413	0.1408	0.6532	0.8936
	95.0%	0.2359	**0.1243**	0.6622	0.8914
	97.5%	0.2299	0.1358	0.6721	0.8902
	99.0%	**0.2280**	0.1337	**0.6901**	0.8910
Tukey's bisquare	85.0%	0.2564	0.1504	0.6340	0.8884
	90.0%	0.2457	0.1465	0.6588	0.8756
	95.0%	0.2411	0.1436	0.6549	**0.8892**
	97.5%	0.2335	**0.1269**	0.6873	0.8848
	99.0%	**0.2292**	0.1422	**0.6939**	0.8872
Huber	85.0%	0.2421	0.1354	0.6575	**0.9076**
	90.0%	0.2358	**0.1266**	0.6715	0.8932
	95.0%	0.2323	0.1438	0.6745	0.8922
	97.5%	0.2322	0.1418	0.7018	0.8928
	99.0%	**0.2320**	0.1438	**0.7401**	0.8804
Generalized Gauss	85.0%	0.2477	0.1441	0.6589	0.8818
	90.0%	0.2374	0.1376	0.6680	0.8766
	95.0%	0.2301	**0.1206**	0.7035	0.8786
	97.5%	**0.2272**	0.1354	0.7023	0.8834
	99.0%	0.2274	0.1312	**0.7156**	**0.8920**
Linear quadratic quadratic	85.0%	0.2533	0.1442	0.6298	**0.9078**
	90.0%	0.2430	0.1398	0.6357	0.8986
	95.0%	0.2364	0.1544	0.6645	0.8934
	97.5%	0.2303	0.1401	**0.6786**	0.8970
	99.0%	**0.2269**	**0.1265**	0.6750	0.8970

Note. Values presented in bold are the best results for each weight function; Values underlined are the best results across the different weight functions.

Table 2 shows the MSEP, MAE, precision, and recall of CRM using the Hampel weight function with the 0.950, 0.975, and 0.999 quantiles of the standard normal distribution as parameters. These results differ slightly from the results in Section 4.2, because now fifty replications are performed instead of hundred replications. By performing fifty replications we can directly compare the results of CRM using the Hampel weight function with the standard normal quantiles as parameters to CRM using the Hampel weight function where the parameters are derived based on a certain efficiency level. We notice that CRM obtains a lower MSEP when parameters corresponding to at least 95% efficiency are used for the Hampel weight function than when the standard normal quantiles are used as parameters. Besides, CRM achieves higher precision and recall scores when using parameters corresponding to at least 95% efficiency. From these results we can conclude that it is better to select the parameters for the Hampel weight function in CRM regression based on an efficiency level of at least 95% instead of using parameters corresponding to the 0.950, 0.975, and 0.999 quantiles of the standard normal distribution.

Table 2: MSEP, MAE, precision, and recall of CRM using the Hampel weight function with the 0.950, 0.975, and 0.999 quantiles of the standard normal distribution as parameters.

Weight function	MSEP	MAE	Precision	Recall
Hampel	0.2490	0.1309	0.6490	0.8892

In Table 3 it is shown that the lowest values for the MSEP and MAE are obtained for parameters corresponding to low breakdown points. These low breakdown points correspond to high efficiency levels. Due to this, CRM is able to estimate coefficients closer to the true values and predict more accurately. The parameters corresponding to a 10% breakdown point

are similar to the parameters corresponding to a 95% efficiency level. As a result, the MSEP and MAE values for a 10% breakdown point are in line with the results presented in Table 1. We notice that the MSEP and MAE are a bit higher for high breakdown points and therefore we advise against using parameters corresponding to high breakdown points.

The highest recall scores are obtained for parameters corresponding to a 50% breakdown point. This is understandable, because due to the high breakdown point the parameters are lower as a result of which more observations are classified as casewise outlier. Thereafter, the SPADIMO method is able to detect more outlying cells, because it considers more casewise outliers. The weight functions using the parameters corresponding to a 50% breakdown point obtain recall scores that are higher than the highest recall scores when using parameters that are selected according to a certain efficiency level. From this we can conclude that we can best select the parameters for the weight functions corresponding to high breakdown point levels if we want to detect the most outlying cells. However, this is of course never the most important goal when we apply CRM regression. Using these parameters corresponding to high breakdown points leads to really low precision scores, which means that way too many cells are identified as outlying. As a result, too many cells get imputed values even if they are not outlying.

Table 3: MSEP, MAE, precision, and recall of CRM using different weight functions for various breakdown points.

Weight function	Breakdown point	MSEP	MAE	Precision	Recall
Hampel	10.0%	**0.2356**	**0.1317**	**0.6587**	0.8934
	20.0%	0.2561	0.1518	0.6163	0.9022
	30.0%	0.2702	0.1498	0.5897	0.9030
	40.0%	0.2789	0.1640	0.5401	0.9128
	50.0%	0.2865	0.1707	0.4868	**0.9268**
Tukey's bisquare	10.0%	**0.2331**	**0.1321**	**0.6611**	0.8956
	20.0%	0.2571	0.1374	0.6293	0.8878
	30.0%	0.2804	0.1509	0.5907	0.8958
	40.0%	0.2744	0.1345	0.5611	0.9016
	50.0%	0.2823	0.1594	0.5080	**0.9110**
Generalized Gauss	10.0%	**0.2616**	0.1612	0.6062	0.8794
	20.0%	0.2831	0.1670	0.5112	0.8850
	30.0%	0.2640	**0.1611**	**0.6086**	0.8844
	40.0%	0.2828	0.1812	0.5553	0.8924
	50.0%	0.3006	0.1902	0.5054	**0.8928**
Linear quadratic quadratic	10.0%	**0.2393**	0.1361	**0.6646**	0.8942
	20.0%	0.2492	**0.1326**	0.6236	0.9074
	30.0%	0.2733	0.1591	0.5387	0.9284
	40.0%	0.2835	0.1544	0.5057	0.9430
	50.0%	0.2840	0.1738	0.4591	**0.9518**

Note. Values presented in bold are the best results for each weight function; Values underlined are the best results across the different weight functions.

Based on the results in Tables 1 and 3 we are not able to conclude that using a specific efficiency level or breakdown point is optimal in selecting parameters for a certain weight function used in CRM regression. However, we recommend to use parameters corresponding to an efficiency level of at least 95%. Unfortunately, high efficiency is accompanied with low breakdown points, as a result of which less outlying cells are identified. However, we do not have to worry too much about this, because by using parameters corresponding to high efficiency levels still around 89% of the outlying cells are identified in CRM regression. Besides, a higher percentage of the cells flagged as outlying are actually cellwise outliers when using parameters corresponding to a high efficiency level.

4.4 Controlling Contamination

In order to evaluate the breakdown behavior of CRM using different weight functions, we consider three different situations where we gradually increment the amount of contamination. In the first situation we keep the amount of contaminated cells fixed at 10% and we gradually increment the amount of casewise outliers up to 50% (in steps of 5%). The second situation is comparable to the first situation, but now we keep the amount of casewise outliers fixed at 10%, while we increase the amount of contaminated cells in steps of 5%. The last situation we increment both the casewise outliers and contaminated cells up to 50%, such that we contaminate up to 25% of the data matrix in this situation. The simulation setting is similar to the description given in Section 4.1. However, now the number of variables p is equal to 60, such that we can easily increment the amount of contaminated cells in steps of 5%. We perform fifteen replications where in each replication the different levels of contamination are evaluated based on the MAE. Table 4 provides an overview how the amount of outliers are incremented in the different situations.

Table 4: Overview of the amount of outliers in the different situations.

Situation	Casewise outliers	Contaminated cells
1	Incremented in steps of 5%	Fixed at 10%
2	Fixed at 10%	Incremented in steps of 5%
3	Incremented in steps of 5%	Incremented in steps of 5%

Figures 3, 4, and 5 show the average MAE across fifteen replications for CRM using different weight functions where the amount of contamination is increased according to situations 1, 2, and 3, respectively. These figures consist of multiple panels where in each panel a weight function used in CRM regression is highlighted. This highlighted weight function is represented by the bold green line, while the other weight functions are represented by the thinner grey lines. This way we are able to differentiate between the different weight functions used in CRM regression and the differences across the weight functions become more clear. The top left panel of Figures 3, 4, and 5 represents CRM using the Hampel weight function with the standard normal quantiles as parameters, while in the other panels parameters corresponding to 95% efficiency are used for the weight functions.

In Figure 3 the average MAE for CRM using different weight functions is presented, where the amount of contamination is increased according to situation 1. In this situation the amount of contaminated cells is kept fixed at 10%, while the amount of casewise outliers is increased in steps of 5%. From the top left panel in this figure it becomes clear that CRM using the Hampel weight function with the standard normal quantiles as parameters is highly biased for small amounts of casewise outliers. When there is less than 10% casewise contamination, the average MAE for CRM using the Hampel weight function with the standard normal quantiles as parameters is almost twice as high as the average MAE for CRM using the other weight functions and CRM using the Hampel weight function with parameters corresponding to a 95% efficiency level. From this we can conclude that it is better to use other weight functions or other parameters for the Hampel weight function in CRM regression when the amount of casewise outliers is limited. For more than 10% casewise outliers, CRM using the Hampel weight function with the standard normal quantiles as parameters seems to perform similar to CRM using the other weight functions. We notice that CRM using the Hampel weight function with

95% efficiency parameters and CRM using Tukey's bisquare weight function show breakdown behavior earlier, as shown by the top middle and top right panel in Figure 3, respectively. Their average MAE is around 0.40 when the amount of casewise contamination is between 15 and 30%, while the MAE for CRM using the other weight functions is around 0.32.

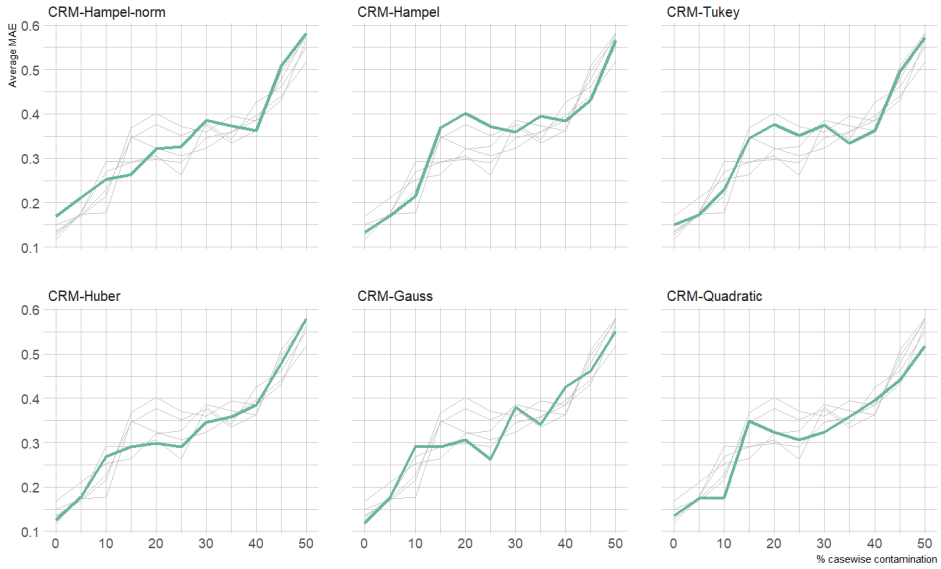


Figure 3: Average MAE for CRM using different weight functions where the amount of contamination is increased according to situation 1.

In situation 2 the amount of casewise outliers is kept fixed at 10%, while the amount of contaminated cells is increased in steps of 5%. The average MAE for CRM using different weight functions where the amount of contamination is increased according to this situation is shown in Figure 4. It is shown by the top left panel that the average MAE for CRM using the Hampel weight function with the standard normal quantiles as parameters is higher than the MAE of CRM using the other weight functions when the percentage of contaminated cells is between 0 and 5%. The average MAE of CRM quickly increases when the amount of contaminated cells is incremented. It is difficult to draw conclusions for situations where more than 20% of the cells are contaminated, because the average MAE for all weight functions seems to alternate between 0.25 and 0.40. However, we notice that the MAE for CRM using the Huber weight function is generally lower than the MAE for CRM using the other weight functions. In situation 2 the average MAE never rises above 0.45, and average MAE values as high as in situation 1 are not reached. This means that the amount of casewise outliers has a larger impact on the breakdown behavior of CRM. The increasing amount of cellwise contamination for each casewise outlier does not lead to more breakdown when the amount of casewise outliers is limited.

Figure 5 shows the average MAE for CRM using different weight functions where the amount of contamination is increased according to situation 3. In this situation the amount of both the casewise outliers and the contaminated cells is increased in steps of 5%. The behavior of CRM in this situation is comparable to the behavior of CRM in situation 1 presented in Figure 3, because also in this situation CRM shows breakdown behavior when more than 10% casewise contamination is present. However, in this situation the effect on the average MAE is more extreme due to the fact that the outlying cells are also increased in steps of 5% instead of kept fixed at 10%. We notice that CRM using the Huber weight function achieves much lower average

MAE values when the percentage of both casewise and cellwise contamination is between 15 and 40%, as becomes clear from the bottom left panel of Figure 5.

Figure 4: Average MAE for CRM using different weight functions where the amount of contamination is increased according to situation 2.

Figure 5: Average MAE for CRM using different weight functions where the amount of contamination is increased according to situation 3.

This simulation study shows that using the Huber weight function makes CRM regression more robust against increasing amounts of contamination, as the bottom left panels of Figures 3, 4, and 5 show that less breakdown behavior occurs when the Huber weight is used in CRM regression. In particular the results of situation 3 show that CRM using the Huber weight function obtains much lower MAE values. However, it should be noted that this situation where the casewise outliers and contaminated cells cohere is unrealistic in practice, because most of the times there are only a few values of specific variables that affect the outlyingness of an observation. Nevertheless, one could consider using the Huber weight function in CRM regression when it is expected that the data contain many outliers.

4.5 Increasing Robustness

Using lower parameter values for the weight function makes the CRM regression algorithm more robust against cellwise contamination, because SPADIMO investigates more casewise outliers. Hence, in situations with many casewise outliers present, one would prefer to use lower parameter values. However, the question remains how much the parameters should be lowered. In this simulation study we propose to lower the parameters corresponding to 95% efficiency with a constant $\alpha \in \{0.6, 0.7, 0.8, 0.9, 1.0\}$. This means that we adjust the parameters according to the following expression:

$$Q_{\text{adjusted}} = \alpha Q, \tag{11}$$

where Q is the parameter value corresponding to 95% efficiency and Q_{adjusted} is the adjusted parameter used by the weight function in CRM regression. In case of the Hampel redescending function, all three parameters are lowered with the factor α. For the Generalized Gauss weight function, the parameter b is kept fixed at 1.5, while a and Q are lowered with the factor α. The parameter S which controls the maximal rate of descent in case of the linear quadratic quadratic weight function is kept fixed at 1.5, while Q_1 and Q_2 are lowered with the factor α.

The data are generated in a similar way as described in Section 4.1. We consider $n = 400$ observations with $p = 60$ corresponding variables. The amount of casewise outliers is increased in steps of 5% until 50% casewise contamination reached, while the amount of outlying cells of each casewise outlier is kept fixed at 10%. This means that we increase the amount of contamination according to situation 1 described in Section 4.4. In total ten replications are performed where in each replication the different values of α are evaluated based on the MAE for different amounts of casewise outliers.

In practice it is unknown how many outliers are present. However, one could approximate how many outliers there are present when looking at the patterns in the data. Hence, based on the results of this simulation study where different values of α are considered, one can get an idea how much the parameters of the weight functions should be adjusted if a certain amount of outliers is expected in the data.

Figure 6 shows the average MAE across ten replications plotted against the percentage of casewise outliers for CRM using different weight functions where the parameters are adjusted with a factor α. The exact MAE values of CRM using a specific weight function where the parameters are lowered with a factor α for different percentages of casewise outliers can be found in Section B of the Appendix.

From this figure it becomes clear that for 0 to 10% casewise outliers the factor $\alpha = 1.0$ obtains either the lowest MAE or an MAE value really close to the lowest MAE value, regardless of the used weight function. In most of the panels of Figure 6, the yellow line representing $\alpha = 1.0$ is below the other lines when the percentage of casewise contamination is less than 10%. Hence, we can conclude that it is unnecessary to adjust the parameters corresponding to 95% efficiency when the amount of casewise outliers is limited.

It depends on the used weight function whether the parameters should be adjusted when there is a larger amount of casewise outliers. Besides, if the parameters should be adjusted, it differs per weight function how much the parameters should be lowered.

When the percentage of casewise outliers is between 10 and 20%, it seems to be better

for the Hampel and Tukey's bisquare weight function to adjust the parameters with a factor $\alpha = 0.9$. The green line representing $\alpha = 0.9$ is below the other lines in the top left and top middle panel in Figure 6. Hence, adjusting the parameters leads to a lower MAE and thus the estimated regression coefficients are closer to the true values when the percentage of casewise outliers is between 10 and 20%. When the percentage of casewise outliers exceeds 25%, CRM shows breakdown behavior when the Hampel or Tukey's bisquare weight funtion is used.

In case of the Huber weight function, the value of α for which the lowest MAE is obtained varies by the percentage level of casewise outliers. When the percentage of casewise outliers is between 15 and 40%, lowering the parameters with a factor α equal to 0.8 or 0.9 leads to more robustness, as shown by the light blue and green line in the top right panel of Figure 6. For more than 40% casewise contamination, lowering the parameter with a factor $\alpha = 0.6$ leads to a relatively low MAE compared to the other α values, as shown by the purple line. Where CRM using the Hampel and Tukey's bisquare weight functions shows clear breakdown behavior, CRM using the Huber weight function seems to be able to estimate accurate regression coefficients, even when the percentage of casewise outliers is really high.

The parameters for the Generalized Gauss weight function should be lowered when more than 15% of the observations is considered to be a casewise outlier. It can even be argued that the parameters should be lowered with a factor α lower than 0.8, because CRM using the Generalized Gauss weight function obtains the lowest MAE when a value of α between 0.6 and 0.8 is used. The purple, dark blue, and light blue lines representing $\alpha = 0.6$, 0.7, and 0.8, respectively, are below the other lines corresponding to $\alpha = 0.9$ and 1.0 when there is more than 15% casewise contamination present.

CRM using the linear quadratic quadratic weight function where the parameters are lowered with a factor α does not necessarily obtain lower MAE values, as shown by the bottom middle panel of Figure 6. Hence, for this weight function it is recommended to use the unadjusted parameters corresponding to 95% efficiency.

Figure 6: Average MAE plotted against the percentage of casewise outliers for CRM using different weight functions where the parameters are adjusted with a factor α.

5 Real Data Application

A real data set is used to see how CRM regression works in practice. We evaluate how using different weight functions in CRM regression has an impact on the estimated regression coefficients, the identification of cellwise outliers, and the predictive power. In this application, parameters corresponding to 95% efficiency are used for the weight functions. The data set is taken from the Swiss nutrition database 2015 and consists of food products where each product has multiple components of nutrients. Table 5 presents the considered variables where cholosterol is the dependent variable of the regression. These variables are logarithmically transformed to reduce the skewness. In total, 193 products that do not contain missing values are considered.

Table 5: Description of the variables in the nutrients data set.

Variable	Description
cholesterol	Cholesterol in milligram per 100g edible portion
energy_kcal	Energy in kcal per 100g edible portion
protein	Protein in gram per 100g edible portion
water	Water in gram per 100g edible portion
carbohydrates	Carbohydrates in gram per 100g edible portion
sugars	Sugars in gram per 100g edible portion

The estimated regression coefficients obtained by applying CRM regression using different weight functions on the nutrients data are shown in Table 6. From this table it becomes clear that CRM provides similar coefficient estimates when the Hampel, Tukey's bisquare, Huber, and linear quadratic quadratic weight functions are used. We notice that the estimated coefficients by CRM using the Generalized Gauss weight function differ substantially from the coefficients estimated by CRM using the other four weight functions.

Table 6: Estimated regression coefficients by CRM using different weight functions.

	Weight function				
Variable	Hampel	Tukey's bisquare	Huber	Generalized Gauss	Linear quadratic quadratic
(Intercept)	-31.75947	-30.99637	-29.02460	-14.56553	-33.16771
log.energy_kcal	3.69692	3.46683	3.66009	1.75543	3.91689
log.protein	0.57835	0.95788	0.46003	0.68042	0.53923
log.water	3.33875	3.37734	2.78560	1.21120	3.45607
log.carbohydrates	-0.03160	-0.32805	-0.27086	-0.57338	-0.04122
log.sugars	-0.13003	0.26979	0.05412	0.16906	-0.15621

Table 7 shows the amount of indicated casewise outliers having at least one contaminated outlying cell by CRM using different weight functions. We directly notice that CRM using the Generalized Gauss weight function does not identify any outlying cells at all, as the number of indicated casewise outliers is equal to zero. This could explain the deviating coefficient estimates in Table 6. It is also noticeable that CRM using the linear quadratic quadratic weigth function does indicate 103 out of 193 food products as casewise outlier having at least one contaminated outlying cell. However, it should be noted that the imputed values for the outlying cells do not deviate that much from the original values.

Table 7: Number of indicated casewise outliers by CRM using different weight functions.

	Weight function				
	Hampel	Tukey's bisquare	Huber	Generalized Gauss	Linear quadratic quadratic
Number of casewise outliers	36	14	29	0	103

Figure 7 shows the 10% trimmed root mean squared error of prediction (RMSEP) that is obtained after performing 10-fold cross-validation on the nutrients data. By trimming we exclude the 10% largest and 10% smallest values for the RMSEP, as a result of which the RMSEP is robust against outliers. From this figure it becomes clear that CRM using the Generalized Gauss weight function has the least predictive power. The Generalized Gauss weight function only assigns lower weights for relatively large residual values. As a result, the Generalized Gauss weight function does not assign observations to SPADIMO, because the obtained residuals are too small. The influence of the outlying cells can not be limited, which causes distorted coefficient estimates and a lower predictive power. Hence, we advise against using the Generalized Gauss weight function in CRM regression if it is expected that the outliers have a small magnitude, because the Generalized Gauss weight function does only assign observations to SPADIMO for really large residual values.

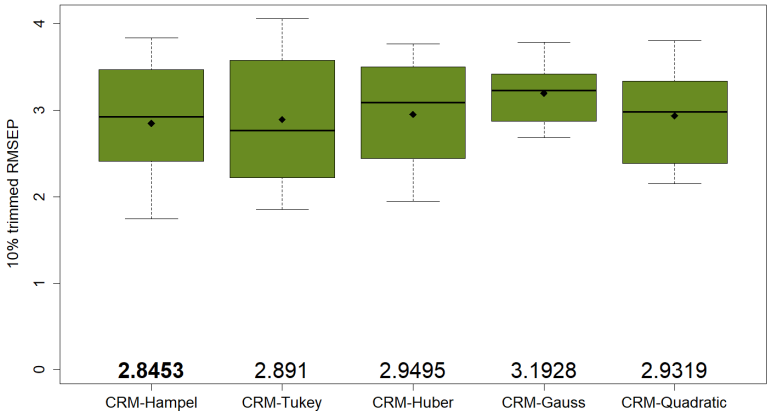


Figure 7: Boxplot for the 10% trimmed RMSEP values from 10-fold cross-validation for CRM using different weight functions.

6 Conclusion

In this research we evaluate the use of different weight functions in CRM regression and how the parameters for the weight functions affect the performance of CRM regression. This research is performed based on the following central research question: *Which weight function can best be used in cellwise robust M regression and how can the most optimal parameters for the weight functions be selected?*

In order to answer this central research question we have considered various simulation studies. First, we have done an initial comparison of the use of different weight functions in CRM regression. Besides, the use of different parameters based on a certain efficiency level or breakdown point is considered. Also the breakdown behavior of CRM using different weight functions is considered where the amount of contamination is increased in various ways. Lastly, we have evaluated how the robustness of CRM could be increased by lowering the parameters for the weight functions.

Based on the results of these simulation studies we conclude that there is no specific weight function that can best be used in CRM regression when the amount of casewise contamination is limited. The results show that CRM achieves similar predictive performance and estimation

accuracy regardless of the used weight function. However, it has become clear that using the Hampel weight function with the standard normal quantiles as parameters, as in the research of Filzmoser et al. (2020), is not optimal in CRM regression.

By evaluating the use of parameters corresponding to a certain efficiency level or breakdown point we conclude that there is no specific efficiency level or breakdown point that provides the most optimal parameters for the weight functions. The results show that CRM achieves higher predictive accuracy and estimated coefficients closer to the true values when parameters for the weight functions are used that correspond to high efficiency levels. Hence, we recommend to use parameters for the weight functions corresponding to at least 95% efficiency when applying CRM regression. Besides, we can conclude that the results for CRM using the Hampel weight function where the parameters are selected based on an efficiency level of at least 95%, are better than when the standard normal quantiles are used as parameters for this weight function.

When the amount of contamination increases, one could consider the Huber weight function in CRM regression, because CRM using the Huber weight functions shows less breakdown behavior. Besides, for small amounts of contamination, we can conclude that is better to use other weight functions or other parameters for the Hampel weight function than using the Hampel weight function with the standard normal quantiles as parameters.

Lowering the parameters for the weight functions used in CRM regression leads to more robustness when the amount of casewise contamination increases. However, it differs per weight function how much the parameters should be lowered.

Besides the various simulation studies, we have evaluated the performance of CRM using different weight functions in a real data application. In this real data application the Generalized Gauss weight function does not assign any observations to SPADIMO due to the fact that the Generalized Gauss weight function only assigns lower weights for relatively large residual values. As a result, the influence of outlying cells can not be limited. Hence, we advise against using the Generalized Gauss weight function in CRM regression if it is expected that the outliers have a small magnitude.

For future research, it could be interesting to evaluate the performance of CRM using different weight functions for varying magnitudes of contamination. Besides, one could evaluate the performance of CRM compared to other cellwise robust regression estimators, e.g. the shooting S-estimator (Öllerer et al., 2016). As an extension to this shooting S-estimator, one could investigate replacing the S-estimator by the MM-estimator.

References

Alqallaf, F., Van Aelst, S., Yohai, V. J. & Zamar, R. H. (2009). Propagation of outliers in multivariate data. *The Annals of Statistics*, *37*(1), 311–331.

Beaton, A. E. & Tukey, J. W. (1974). The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. *Technometrics*, *16*(2), 147–185.

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*(4), 707–723.

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

Filzmoser, P., Höppner, S., Ortner, I., Serneels, S. & Verdonck, T. (2020). Cellwise robust m regression. *Computational Statistics & Data Analysis*, *147*, 106944.

Fu, W. J. (1998). Penalized regressions: the bridge versus the lasso. *Journal of computational and graphical statistics*, *7*(3), 397–416.

Green, P. J. (1984). Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, *46*(2), 149–170.

Grubbs, F. E. (1969). Procedures for detecting outlying observations in samples. *Technometrics*, *11*(1), 1–21.

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. & Stahel, W. A. (1986). *Robust statistics: the approach based on influence functions*. New York: Wiley.

He, X., Jurečková, J., Koenker, R. & Portnoy, S. (1990). Tail behavior of regression estimators and their breakdown points. *Econometrica*, *58*(5), 1195–1214.

Heij, C., De Boer, P., Franses, P. H., Kloek, T. & Van Dijk, H. K. (2004). *Econometric methods with applications in business and economics*. Oxford, UK: Oxford University Press.

Huber, P. J. (1964). Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, *35*(1), 73–101.

Huber, P. J. (1984). Finite sample breakdown of m-and p-estimators. *The Annals of Statistics*, *12*(1), 119–126.

Hubert, M., Rousseeuw, P. J. & Vanden Branden, K. (2005). Robpca: a new approach to robust principal component analysis. *Technometrics*, *47*(1), 64–79.

Koller, M. & Stahel, W. A. (2011). Sharpening wald-type inference in robust regression for small samples. *Computational Statistics & Data Analysis*, *55*(8), 2504–2515.

Maronna, R. A., Martin, R. D. & Yohai, V. J. (2006). *Robust statistics*. New York: Wiley.

Öllerer, V., Alfons, A. & Croux, C. (2016). The shooting s-estimator for robust regression. *Computational Statistics*, *31*(3), 829–844.

Rousseeuw, P. J. & Van den Bossche, W. (2018). Detecting deviating data cells. *Technometrics*, *60*(2), 135–145.

Rousseeuw, P. J. & Yohai, V. J. (1984). Robust regression by means of s-estimators. *Robust and Nonlinear Time Series Analysis. Lecture Notes in Statistics*, *26*, 256–272.

A Parameter Values Efficiency-Robustness Tradeoff

In Tables 8 and 9 the parameters are shown that correspond to the different efficiency levels and breakdown points, respectively. This way it is possible to verify which specific parameters are used for a certain efficiency level or breakdown point in Section 4.3 of our main research.

Table 8: Parameters corresponding to different efficiency levels for the weight functions used in CRM regression.

Weight function	Efficiency	Parameter values
Hampel	85.0%	$(Q_1, Q_2, Q_3) = (0.954, 1.908, 3.817)$
	90.0%	$(Q_1, Q_2, Q_3) = (1.105, 2.210, 4.421)$
	95.0%	$(Q_1, Q_2, Q_3) = (1.382, 2.764, 5.528)$
	97.5%	$(Q_1, Q_2, Q_3) = (1.664, 3.327, 6.654)$
	99.0%	$(Q_1, Q_2, Q_3) = (2.011, 4.023, 8.046)$
Tukey's bisquare	85.0%	$Q = 3.444$
	90.0%	$Q = 3.883$
	95.0%	$Q = 4.685$
	97.5%	$Q = 5.597$
	99.0%	$Q = 7.041$
Huber	85.0%	$Q = 0.732$
	90.0%	$Q = 0.982$
	95.0%	$Q = 1.345$
	97.5%	$Q = 1.655$
	99.0%	$Q = 2.018$
Generalized Gauss	85.0%	$(Q, a, b) = (0.759, 0.837, 1.5)$
	90.0%	$(Q, a, b) = (0.871, 1.028, 1.5)$
	95.0%	$(Q, a, b) = (1.063, 1.386, 1.5)$
	97.5%	$(Q, a, b) = (1.256, 1.782, 1.5)$
	99.0%	$(Q, a, b) = (1.511, 2.349, 1.5)$
Linear quadratic quadratic	85.0%	$(Q_1, Q_2, S) = (0.705, 1.058, 1.5)$
	90.0%	$(Q_1, Q_2, S) = (0.809, 1.214, 1.5)$
	95.0%	$(Q_1, Q_2, S) = (0.982, 1.473, 1.5)$
	97.5%	$(Q_1, Q_2, S) = (1.152, 1.729, 1.5)$
	99.0%	$(Q_1, Q_2, S) = (1.375, 2.063, 1.5)$

Note. For each weight function, the bold efficiency level obtains the lowest MSEP, while the underlined efficiency level obtains the lowest MAE.

Table 9: Parameters corresponding to different breakdown points for the weight functions used in CRM regression.

Weight function	Breakdown point	Parameter values
Hampel	**10.0%**	$(Q_1, Q_2, Q_3) = (1.368, 2.736, 5.472)$
	20.0%	$(Q_1, Q_2, Q_3) = (0.894, 1.789, 3.577)$
	30.0%	$(Q_1, Q_2, Q_3) = (0.663, 1.327, 2.653)$
	40.0%	$(Q_1, Q_2, Q_3) = (0.511, 1.023, 2.046)$
	50.0%	$(Q_1, Q_2, Q_3) = (0.396, 0.793, 1.585)$
Tukey's bisquare	**10.0%**	$Q = 5.182$
	20.0%	$Q = 3.420$
	30.0%	$Q = 2.56$
	40.0%	$Q = 1.988$
	50.0%	$Q = 1.548$
Generalized Gauss	**10.0%**	$(Q, a, b) = (0.455, 0.304, 0.5)$
	20.0%	$(Q, a, b) = (0.260, 0.219, 0.5)$
	30.0%	$(Q, a, b) = (0.503, 0.452, 1.5)$
	40.0%	$(Q, a, b) = (0.385, 0.302, 1.5)$
	50.0%	$(Q, a, b) = (0.296, 0.204, 1.5)$
Linear quadratic quadratic	**10.0%**	$(Q_1, Q_2, S) = (0.957, 1.435, 1.5)$
	20.0%	$(Q_1, Q_2, S) = (0.622, 0.932, 1.5)$
	30.0%	$(Q_1, Q_2, S) = (0.456, 0.685, 1.5)$
	40.0%	$(Q_1, Q_2, S) = (0.348, 0.523, 1.5)$
	50.0%	$(Q_1, Q_2, S) = (0.268, 0.402, 1.5)$

Note. For each weight function, the bold breakdown point obtains the lowest MSEP, while the underlined breakdown point obtains the lowest MAE.

B Exact MAE Values Increasing Robustness

Table 10 shows the average MAE across ten replications for CRM using different weight functions where the parameters are adjusted with a factor α. The average MAE is evaluated for different percentages of casewise outliers. In each step the amount of casewise outliers is increased with an additional 5% until 50% casewise contamination is attained. This table shows the exact MAE values that correspond to the results presented in Figure 6 of Section 4.5.

Table 10: Average MAE for CRM using different weight functions where the parameters are adjusted with a factor α.

		α				
Weight function	% casewise outliers	0.6	0.7	0.8	0.9	1.0
Hampel	0	0.1768	0.2028	0.1741	**0.1323**	0.1505
	5	0.1874	0.1755	0.1730	0.1788	**0.1597**
	10	0.3212	0.3237	0.2685	**0.2010**	0.2448
	15	0.2691	**0.2655**	0.3184	0.2861	0.3241
	20	0.3505	0.4201	0.3039	**0.2591**	0.2947
	25	0.4534	0.3627	**0.3457**	0.4394	0.3731
	30	0.4501	0.4258	0.4126	0.4187	**0.3554**
	35	0.4701	**0.3882**	0.3957	0.4715	0.4240
	40	0.3949	0.3914	0.4497	**0.3752**	0.4436
	45	0.5783	0.4801	0.4692	**0.4471**	0.4714
	50	0.5850	0.6073	0.5572	0.6040	**0.5164**
Tukey's bisquare	0	0.2055	0.2150	0.1966	0.1924	**0.1261**
	5	0.1740	**0.1400**	0.1556	0.2225	0.1709
	10	0.3426	0.2992	**0.2712**	0.3697	0.3321
	15	0.3956	0.3830	0.3440	**0.2825**	0.3486
	20	0.3667	0.4254	0.3337	**0.2973**	0.3610
	25	0.3946	0.3751	**0.3100**	0.4111	0.3669
	30	0.3754	0.4545	0.4151	0.3341	**0.3088**
	35	0.3887	0.3785	**0.3543**	0.4492	0.3973
	40	0.4339	**0.4186**	0.4884	0.4840	0.5206
	45	0.5090	0.5445	0.5233	**0.4545**	0.4682
	50	0.5903	0.5997	**0.5346**	0.5819	0.5659
Huber	0	0.1490	**0.1255**	0.1442	0.1329	0.1325
	5	**0.1565**	0.1574	0.1831	0.1700	0.1569
	10	**0.2514**	0.2915	0.2915	0.3626	0.3098
	15	0.3436	0.3453	**0.2796**	0.3587	0.3503
	20	**0.2430**	0.4038	0.3270	0.2662	0.3267
	25	0.3634	0.3635	0.3522	**0.3207**	0.3578
	30	**0.3106**	0.4049	0.3667	0.3557	0.4325
	35	0.3657	0.3518	0.3445	**0.3146**	0.3999
	40	0.3745	0.4686	**0.3257**	0.3548	0.4759
	45	**0.4068**	0.4526	0.4609	0.5131	0.4677
	50	**0.5343**	0.5489	0.5600	0.5476	0.5788
Generalized Gauss	0	0.1686	0.1482	0.1576	0.1385	**0.1269**
	5	0.1911	**0.1615**	0.1810	0.1648	0.1686
	10	0.3387	0.3295	0.3275	0.3255	**0.2753**
	15	0.3056	0.3027	**0.2590**	0.3642	0.3224
	20	**0.2631**	0.3859	0.3605	0.4012	0.4035
	25	0.4073	0.3964	**0.3765**	0.4264	0.4687

	30	0.3582	**0.3501**	0.3536	0.3719	0.3848
	35	0.4593	**0.3336**	0.3653	0.4250	0.4325
	40	0.4439	0.4153	**0.3551**	0.5010	0.4681
	45	0.5270	0.4846	**0.4590**	0.4940	0.4932
	50	0.6126	**0.5445**	0.5889	0.5737	0.5909
Linear quadratic quadratic	0	0.2319	0.2024	0.1999	**0.1488**	0.1587
	5	0.2972	0.1843	0.1814	0.1898	**0.1425**
	10	0.3068	0.2799	**0.2182**	0.3778	0.2228
	15	0.4060	0.3098	0.3278	**0.2764**	0.2842
	20	0.3287	**0.2769**	0.2979	0.3367	0.2835
	25	0.3947	0.3485	**0.3084**	0.3113	0.3160
	30	0.3557	0.3509	0.3697	**0.3410**	0.3433
	35	0.4223	0.3687	0.4094	**0.3328**	0.3948
	40	0.4790	0.4269	0.4481	0.4322	**0.3934**
	45	0.4587	**0.4394**	0.5110	0.4522	0.4641
	50	0.5713	0.5533	**0.5432**	0.6085	0.5630

Note. For each weight function, the lowest MAE value for a certain level of casewise contamination is presented in bold.

C README Files

In Section C.1 the README file of our developed R package `CRMwf` is provided. In this README file it is shortly described how this package extends the `crmReg` package of Filzmoser et al. (2020). It is mentioned which weight functions are included in the package and how the package can be installed. The source code of this package can be found on `https://github.com/j4c0d3h00g/CRMwf`.

The simulation studies that are used to evaluate the performance of CRM regression using different weight functions can be found on `https://github.com/j4c0d3h00g/CRMsimulations`. Section C.2 states the R files corresponding to the simulation studies of our research.

C.1 CRMwf

This repository contains the R package `CRMwf`.
The R package `CRMwf` contains the implementation of the Cellwise Robust M-regression (CRM) algorithm. Here, CRM is extended with additional input arguments. Namely, using this package it is possible to use other weight functions in CRM regression where also the parameters for the weight function can be adjusted. Besides the implementation of the CRM algorithm (`crm_functional.R`), this package contains the following weight functions:

- The Hampel weight function (`HampelWeightFunction.R`)
- The Tukey's bisquare weight function (`TukeyWeightFunction.R`)
- The Huber weight function (`HuberWeightFunction.R`)
- The Andrews-sine weight function (`AndrewsWeightFunction.R`)
- The Generalized Gauss weight function (`GaussWeightFunction.R`)
- The linear quadratic quadratic weight function (`QuadraticWeightFunction.R`)

Installation

The package can be installed using the `devtools` package by calling `devtools::install_github("j4c0d3h00g/CRMwf")`.

Acknowledgements

The implementation of the R package `CRMwf` is based on the R package `crmReg` of Filzmoser, P., Höppner, S., Ortner, I., Serneels, S., and Verdonck, T (`https://github.com/SebastiaanHoppner/CRM`)

C.2 CRMsimulations

This repository contains simulation studies where the performance of cellwise robust M (CRM) regression using different weight functions is evaluated. In these simulation studies the package `CRMwf` is used, which can be found on `https://github.com/j4c0d3h00g/CRMwf`. The repository `CRMsimulations` contains the following R files:

- `CRM_plot_weightfunctions.R`: creates plots of the different weight functions for various parameters.
- `CRM_simulation_comparison.R`: compares the use of different weight functions in CRM based on the MSEP, MAE, precision, and recall.
- `CRM_simulation_efficiency.R`: evaluates the use of different weight functions in CRM based on the MSEP, MAE, precision, and recall. Here the parameters are selected based on different efficiency levels.
- `CRM_simulation_breakdown.R`: evaluates the use of different weight functions in CRM based on the MSEP, MAE, precision, and recall. Here the parameters are selected based on different breakdown points.
- `CRM_simulation_contamination_situation1.R`: evaluates the breakdown behavior of CRM using different weight functions. Here, only the amount of casewise contamination is increased.
- `CRM_simulation_contamination_situation2.R`: evaluates the breakdown behavior of CRM using different weight functions. Here, only the amount of cellwise contamination is increased.
- `CRM_simulation_contamination_situation3.R`: evaluates the breakdown behavior of CRM using different weight functions. Here, both the amount of casewise and cellwise contamination is increased.
- `CRM_simulation_alpha.R`: evaluates whether the robustness of the CRM regression estimator could increase when the parameters for the weight functions are lowered.
- `CRM_simulation_realdata.R`: evaluates the predictive performance of CRM using different weight functions in a real data application. To illustrate how CRM regression works in practice, a cellwise heatmap that shows which values are imputed for the outlying cells is included for CRM using the Hampel weight function.

D Replication Cellwise Robust M Regression

D.1 Performance Evaluation

The main focus of this research is the evaluation of different weight functions in CRM regression. Hence, the comparison of CRM using the Hampel weight function to other regression methods is omitted from the main research. However, it is still interesting to look at these results as it shows that the CRM regression estimator is at least on par with the casewise robust MM-estimator when evaluating the predictive power. Besides, we can evaluate whether it is possible to replicate the results of Filzmoser et al. (2020).

The performance of CRM is compared to regular MM regression, MM regression combined with Detecting Deviating Data Cells (DDC), ordinary least squares (OLS) regression, and OLS regression in combination with DDC. The DDC method proposed by Rousseeuw and Van den Bossche (2018) flags cells deviating from the general pattern in a column corresponding with a variable. The unflagged cells are used to predict values for each data cell. A cell is considered to be a cellwise outlier if the value of a cell strongly differs from this predicted value. To limit the influence of these cellwise outliers, the DDC method forms an imputed data matrix where the cellwise outliers are replaced with the predicted values. By using this imputed data matrix instead of the original data matrix, the regression is robust against cellwise outliers.

Both CRM and DDC generate an imputed data matrix. To evaluate whether the imputed values for the cellwise outliers are similar to the cells in the uncontaminated data matrix, we consider the root mean squared error of imputation (RMSEI) given by

$$\text{RMSEI}(\boldsymbol{X}^{imp}, \boldsymbol{X}) = \sqrt{\frac{1}{np}\sum_{i=1}^{n}\sum_{j=1}^{p}(x_{ij}^{imp} - x_{ij})^2}. \tag{12}$$

Here, x_{ij}^{imp} and x_{ij} are the cells corresponding to row i and column j in the imputed and uncontaminated data matrix, respectively. The number of observations is denoted by n, while p represents the number of variables. We do not consider this performance measure in the main research, because we expect the differences in RMSEI for CRM using different weight functions to be insignificant.

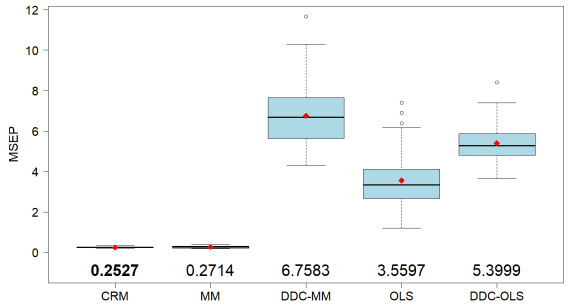
D.2 Simulation Results Comparison Regression Methods

Figure 8 shows the performance evaluation results of CRM using the Hampel weight function and several other regression methods. The data for this simulation study are generated according to the simulation setting described by Filzmoser et al. (2020). Due to the fact that in our main research we generated the data beforehand instead of during each replication, the performance evaluation values for CRM using the Hampel weight function in the initial comparison of the weight functions in Section 4.2 differ slightly from the values of CRM presented in Figure 8.

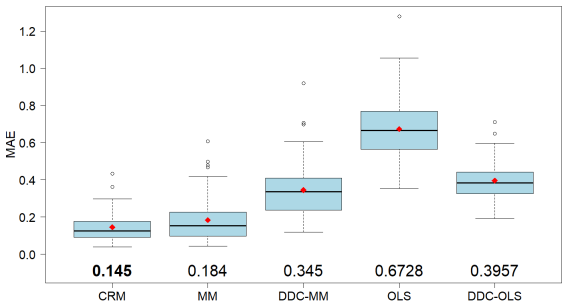
From the results shown in Figure 8a it becomes clear that CRM has a predictive power similar to the benchmark MM regression. Besides, these results show that applying DDC as preprocessing step does not increase predictive performance, because both MM and OLS regression provide more accurate predictions when DDC is not used.

Figure 8b shows that OLS provides biased regression coefficients when outliers are present. Both CRM and regular MM regression have a higher statistical efficiency than the methods using DDC as a preprocessing step. CRM regression provides regression coefficients deviating the least from the true value, because it achieves the lowest average value for the MAE.

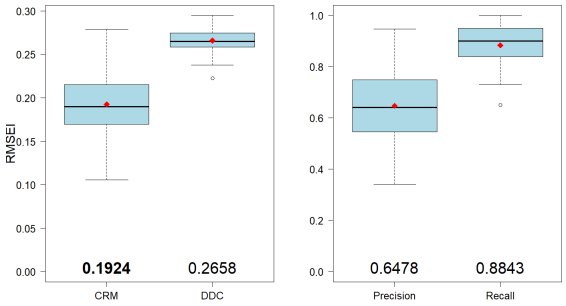
Besides, the imputed values by CRM are closer to the values of the uncontaminated data matrix than the predicted values by DDC. This becomes clear from Figure 8c as the RMSEI is lower for CRM than for DDC. This figure also shows the precision and recall of the identification of cellwise outliers by CRM. Most of the cellwise outliers are actually identified by CRM. However, there are also cases where a cell is identified as cellwise outlier while this is actually not the case, because the precision is relatively low with respect to the recall.



(a) Boxplot for the MSEP results of the different regression methods.



(b) Boxplot for the MAE results of the different regression methods.



(c) Boxplot for the RMSEI, precision and recall of the different regression methods.

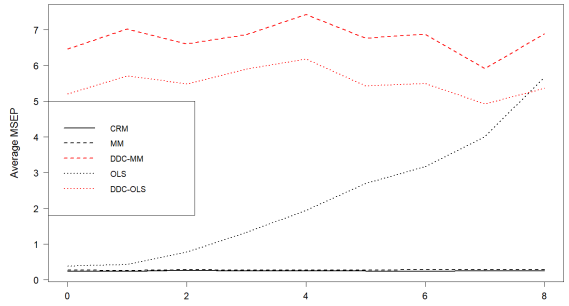
Figure 8: Performance evaluation results for CRM compared to several other regression methods.

D.3 Simulation Results Varying Magnitude of Contamination

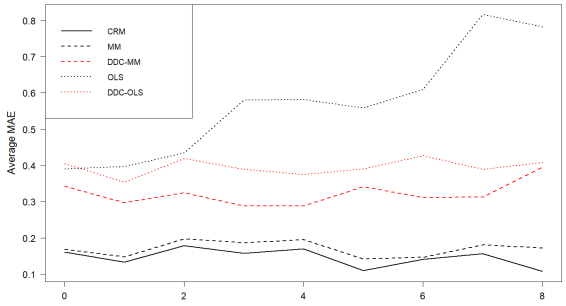
Contamination is added to the data matrix $\boldsymbol{X}$ according to Equation 8 described in Section 4.1 of the main research. In order to compare CRM with the other regression methods for different magnitudes of contamination, different values for k are considered. In Figure 9 several performance evaluation results are presented for $k \in \{0, 1, 2, ..., 8\}$. Note that for $k = 0$ the cells are just equal to the column means. In this figure, the average results across ten replications for the performance measures are presented. CRM is represented by the solid black line. We distinguish MM and OLS regression from CRM by using dashed lines and dotted lines, respectively. Besides, the lines are red when DDC is applied in the preprocessing phase.

CRM, MM regression, and the regression methods combined with DDC do not perform worse when the magnitude of contamination increases, as becomes clear from Figures 9a and 9b. In

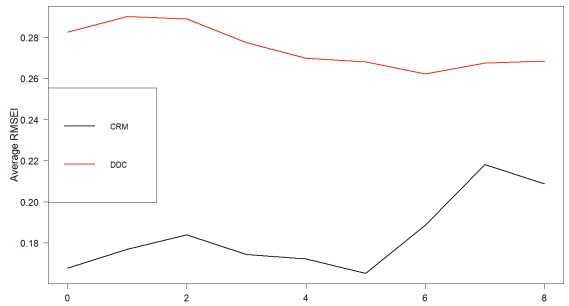
contrast, regular OLS regression destabilizes when the magnitude of contamination increases. The MSPE and MAE increase for higher magnitudes of contamination in case of OLS, which means that OLS provides less accurate predictions and estimated coefficients differ more from the true values. It is surprising that the RMSEI of CRM suddenly increases for $k \geq 5$. However, the RMSEI of CRM is still lower than the RMSEI of DDC, as can be seen in Figure 9c. Figure 9d shows that CRM is only able to identify outlying cells for higher magnitudes of contamination, because the precision and recall are higher for larger values of k.



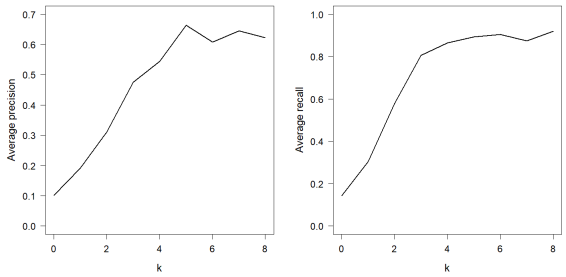
(a) MSEP results of the regression methods for different values of k.



(b) MAE results of the regression methods for different values of k.



(c) RMSEI results of the regression methods for different values of k.



(d) Precision and recall results of the regression methods for different values of k.

Figure 9: Several relative performance evaluation results of CRM and other regression methods for different values of k, where k controls the magnitude of contamination.

D.4 Simulation Results Breakdown

When comparing CRM with the different regression methods, as in Section D.2, the contamination is fixed at 5%. Now, the amount of contamination is gradually increased in steps of 5% (until 50%). Only the amount of casewise outliers is increased. For each casewise outlier, the amount of outlying cells is still fixed at 10%. Figure 10 shows the breakdown behavior of the different regression methods. For each considered percentage level of contamination the average MAE across 10 replications is presented. It becomes clear that OLS is the most efficient if no contamination is present. However, even if there is only a small amount of contamination, OLS is highly biased. We notice that the regression methods combined with DDC do not really show breakdown behavior, because the average MAE is relatively stable for the different percentage levels of contamination. This shows the robustness of DDC. However, this comes with large bias when only a small amount of contamination is present. CRM and MM regression show similar breakdown behavior, but CRM seems to be a little less biased than MM regression.

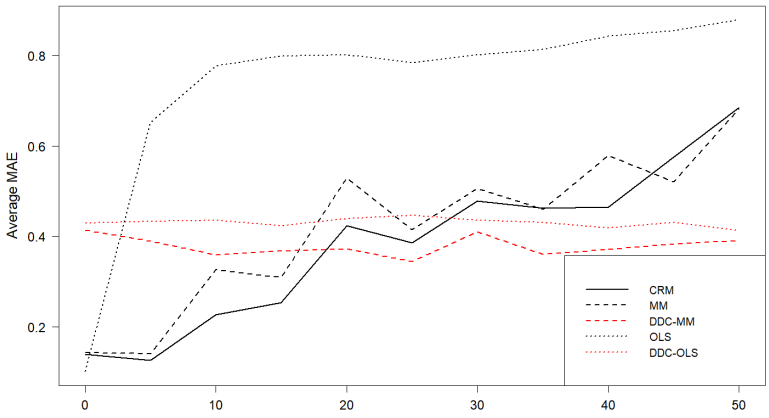


Figure 10: Average MAE of different regression methods for different levels of contamination.

D.5 Real Data Example

A real data set is used to see how CRM regression works in practice and how it compares to other regression methods. The data set is taken from the Swiss nutrition database 2015 and consists of food products where each food product has multiple components of nutrients. This is the same data set as we used in Section 5 of our main research. Hence, we refer to Table 5 of Section 5 for a description of the variables that are included in this real data example. Also in this case the variables are logarithmically transformed and we consider 193 food products that do not contain missing values.

Figure 11 shows the outliers that are detected by the CRM regression algorithm when applied to the nutrients data. The outlying cells are colored blue when they are downwards deviating, while they are colored red when they are upwards deviating.

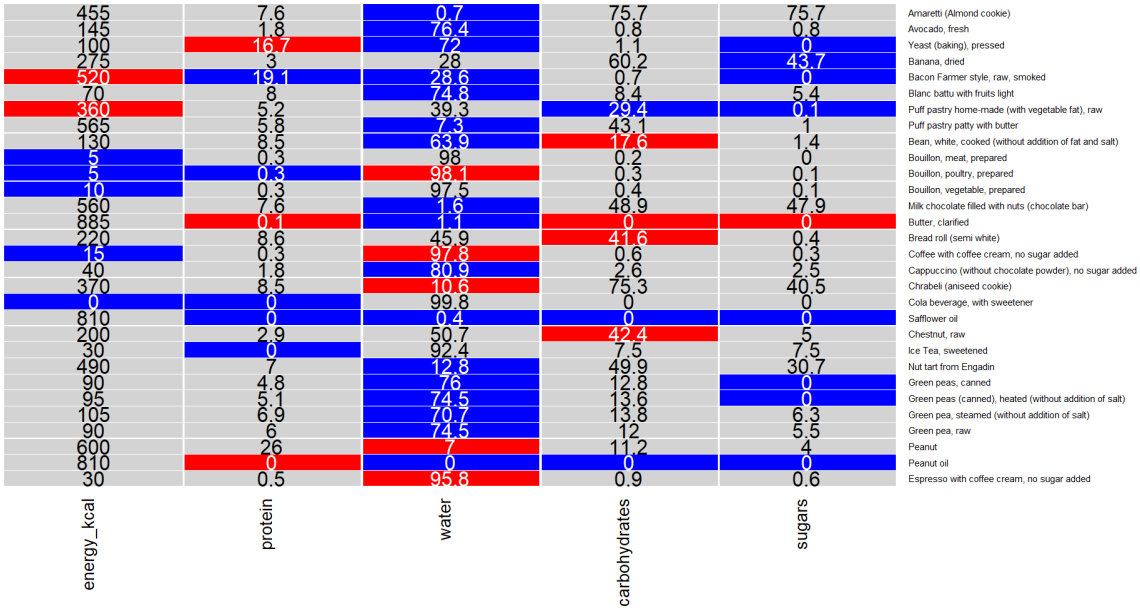


Figure 11: Heatmap of the outlying cells detected by CRM when applied to the nutrients data.

In Figure 12 it is shown what values are imputed by CRM for the outlying cells. Cells with a blue color are replaced by higher values and the red cells are replaced by lower values. A darker color represents a bigger difference between the original value and the imputed value.

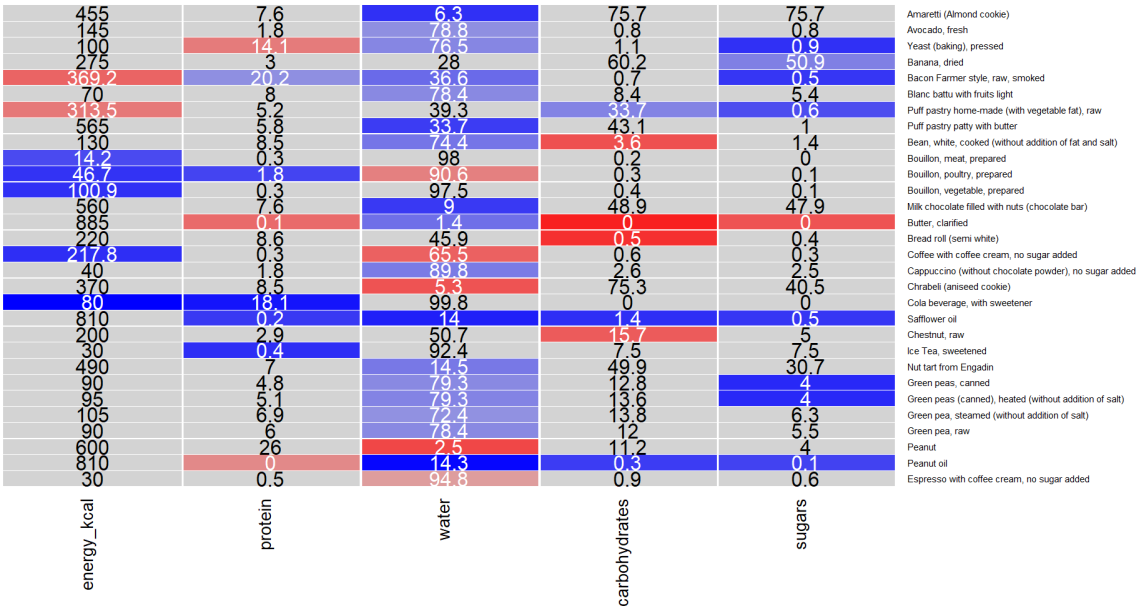


Figure 12: Heatmap of the outlying cells detected by CRM when applied to the nutrients data with the blue and red cells containing the imputed values.

Table 11 shows the estimated regression coefficients obtained by performing CRM regression on the nutrients data.

Table 11: Estimated regression coefficients obtained by performing CRM on the nutrients data.

Variable	Estimated coefficient
(Intercept)	-33.73173
log.energy_kcal	3.62970
log.protein	0.98341
log.water	3.78561
log.carbohydrates	0.05336
log.sugars	-0.10999

The performance of CRM regression in this real data example is compared to regular MM regression, MM regression combined with DDC, OLS regression, and OLS regression in combination with DDC. These regression methods are evaluated based on the 10% trimmed root mean squared error of prediction (RMSEP) that is obtained after performing 10-fold cross-validation. Figure 13 shows the comparison of the regression methods based on the 10% trimmed RMSEP.

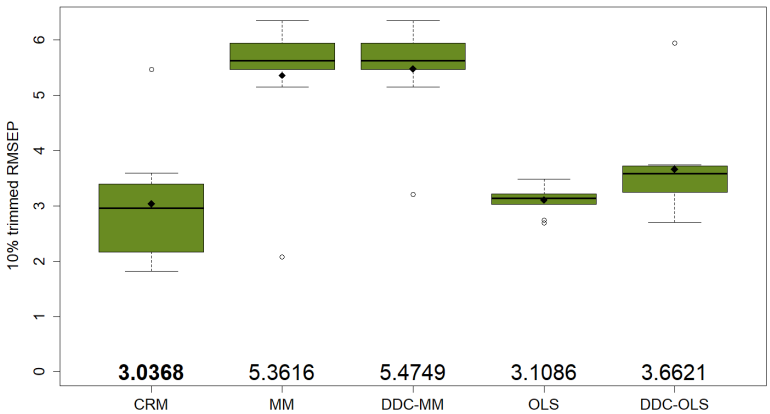


Figure 13: Boxplot for the 10% trimmed RMSEP values from 10-fold cross-validation for the different regression methods.

From Figure 13 it becomes clear that CRM regression has more predictive power than both MM regression and MM regression combined with DDC, but we also notice that CRM does not perform necessarily better than OLS when evaluating based on the trimmed RMSEP. It seems that applying the DDC method before performing OLS does not have the desired effect, because the trimmed RMSEP is slightly higher for OLS combined with DDC than for OLS.

D.6 Discussion

Filzmoser et al. (2020) developed the R package `crmReg` which included the CRM regression algorithm and the method of SPADIMO. On their Github page (`https://github.com/SebastiaanHoppner/CRM`) the source code of this package together with the simulation studies can be found. Due to this, it is possible to replicate the results obtained in the research by Filzmoser et al. (2020). In their research they refer to the website of the Swiss Nutrition Association as the source for the nutrients data set used in the real data example. However, it is not possible anymore to find the Swiss nutrition database 2015 on this website. Fortunately this did not cause any problems, because the data set is also included in the R package `robCompositions`.

In Section 4.2 of our main research it can be seen that the values of the performance measures for CRM using the Hampel weight function differ slightly from the results of CRM in the research of Filzmoser et al. (2020). This is due to the fact that in our research the data are generated beforehand instead of during a replication. We noticed that the data generated during the replications deviate from the data generated beforehand. Besides, when we perform CRM during a replication more (or less) often, the data differ mutually and also deviate from the data generated beforehand. This means that the data are dependent of the number of calls to the CRM regression algorithm during a replication. It is likely that the seed used by the generator providing the random drawings from the multivariate normal distribution is changed when the CRM regression algorithm is applied. Hence, in our main research we have chosen to generate the data beforehand, such that this occurrence is excluded.