REVSTAT – Statistical Journal
Volume 17, Number 4, October 2019, 449–474

A NEW ROBUST PARTIAL LEAST SQUARES REGRESSION METHOD BASED ON A ROBUST AND AN EFFICIENT ADAPTIVE REWEIGHTED ESTIMATOR OF COVARIANCE

Authors: Esra Polat
– Department of Statistics, Hacettepe University, Turkey
espolat@hacettepe.edu.tr

Suleyman Gunay
– Department of Statistics, Hacettepe University, Turkey
sgunay@hacettepe.edu.tr

Received: November 2016 Revised: July 2017 Accepted: July 2017

Abstract:

- Partial Least Squares Regression (PLSR) is a linear regression technique developed as an incomplete or "partial" version of the least squares estimator of regression, applicable when high or perfect multicollinearity is present in the predictor variables. Robust methods are introduced to reduce or remove the effects of outlying data points. In the previous studies it has been showed that if the sample covariance matrix is properly robustified further robustification of the linear regression steps of the PLS1 algorithm (PLSR with univariate response variable) becomes unnecessary. Therefore, we propose a new robust PLSR method based on robustification of the covariance matrix used in classical PLS1 algorithm. We select a reweighted estimator of covariance, in which the Minimum Covariance Determinant as initial estimator is used, with weights adaptively computed from the data. We compare this new robust PLSR method with classical PLSR and four other well-known robust PLSR methods. Both simulation results and the analysis of a real data set show the effectiveness and robustness of the new proposed robust PLSR method.

Key-Words:

- *efficient estimation; Minimum Covariance Determinant (MCD); partial least squares regression; robust covariance matrix; robust estimation.*

AMS Subject Classification:

- 62F35, 62H12, 62J05.

1. INTRODUCTION

Classical PLSR is a well-established technique in multivariate data analysis. It is used to model the linear relation between a set of regressors and a set of response variables, which can then be used to predict the value of the response variables for a new sample. A typical example is multivariate calibration where the x-variables are spectra and the y-variables are the concentrations of certain constituents. Since classical PLSR is known to be severely affected by the presence of outliers in the data or deviations from normality, several PLSR methods with robust behaviour towards data contamination have been proposed (Hubert and Vanden Branden, 2003; Liebmann *et al.*, 2010). NIPALS and SIMPLS are the popular algorithms for PLSR and they are very sensitive to outliers in the dataset. For univariate or multivariate response variable several robustified versions of these algorithms have already been proposed (González *et al.*, 2009).

The two main strategies in the literature for robust PLSR are (1) the downweighting of outliers and (2) robust estimation of the covariance matrix. The early approaches for robust regression by downweighting of outliers are considered semi-robust: they had, for instance, non-robust initial weights or the weights were not resistant to leverage points (Hubert and Vanden Branden, 2003). Based on the first strategy, for example, Wakeling and Macfie (1992) worked with the PLS with multivariate response variables (which will be called PLS2) and their idea was to replace the set of regressions involved in the standard PLS2 algorithm by M estimates based on weighted regressions. Griep *et al.* (1995) compared least median of squares (LMS), Siegel's repeated median (RM) and iterative reweighted least squares (IRLS) for PLS with univariate response variable (PLS1 algorithm), but these methods are not resistant to high leverage outliers (González *et al.*, 2009). Based on the second strategy, a robust covariance estimation, the robust PLSR methods provide resistance to all types of outliers including leverage points (Hubert and Vanden Branden, 2003). For instance, Gil and Romera (1998) proposed a robust PLSR method based on statistical procedures for covariance matrix robustification for PLS1 algorithm. They selected the well-known Stahel–Donoho estimator (SDE) (Gil and Romera, 1998). Since SIMPLS is based on the empirical cross-covariance matrix between the y-variables and the x-variables and on linear Least Squares (LS) regression, the results are affected by outliers in the data set. Hence, Hubert and Vanden Branden (2003) have been suggested a robust version of this method called RSIMPLS that it is used in case of both univariate and multivariate response variables. A robust method RSIMPLS starts by applying ROBPCA on the x- and y-variables in order to replace the covariance matrices S_{xy} and S_x by robust estimates and then proceeds analogously to the SIMPLS algorithm. A robust regression method (ROBPCA regression) is performed in the second stage. ROBPCA is a robust PCA method which combines projection pursuit ideas with Minimum Covariance Determinant (MCD) covariance estimation in lower dimensions (Engelen *et al.*, 2004; Hubert and Vanden Branden, 2003).

Serneels *et al.* (2005) proposed a method called as Partial Robust M (PRM) regression that it is conceptually different from the other robust PLSR methods: instead of robust partial least squares, a partial robust regression estimator was proposed. This method uses SIMPLS algorithm and it could be used in case of univariate response. In this method, with an appropriately chosen weighting scheme, both vertical outliers and leverage points were downweighted (Serneels *et al.*, 2005). As the name suggests, it is a partial version of the robust M-regression. In an iterative scheme, weights ranging between zero and one are calculated to reduce the influence of deviating observations in the y space as well as in the space of the regressor variables. PRM is very efficient in terms of computational cost and statistical properties (Liebmann *et al.*, 2010). González *et al.* (2009) also concentrated in the case of univariate response (PLS1) and showed that if the sample covariance matrix is properly robustified the PLS1 algorithm will be robust and, therefore, further robustification of the linear regression steps of the PLS1 algorithm is unnecessary (González *et al.*, 2009).

In this paper, we concentrate in the case of univariate response (PLS1) and we present a procedure which applies the standard PLS1 algorithm to a robust covariance matrix similar to Gil and Romera (1998) and González *et al.* (2009) studies. In our study, we estimate the covariance matrix used in PLS1 algorithm robustly by using '*an adaptive reweighted estimator of covariance using Minimum Covariance Determinant (MCD) estimators in the first step as robust initial estimators of location and covariance*'.

The rest of the paper is organized as follows. Section 2 reviews briefly the PLS1 algorithm (PLS with univariate response variable). Section 3 presents the new proposed robust PLSR method 'PLS-ARWMCD'. Section 4 contains a simulation study where the performance of the new robust PLSR method is compared to classical PLSR method and other four robust PLSR methods existing in robust PLSR literature. Section 5 illustrates the performance of the new proposed robust PLSR method 'PLS-ARWMCD' in a well known set of real data in robust PLSR literature. Finally, Section 6 collects some conclusions.

2. THE CLASSICAL PLS1 ALGORITHM

It is supposed that we have a sample of size n of a $1+p$ dimensional vector $\boldsymbol{z} = (\boldsymbol{y}, \boldsymbol{X})'$ which could be decomposed as a set of p independent variables, x and a univariate response variable y. Throughout this paper, matrices are denoted by bold capital letters and vectors are denoted by bold lowercase letters. Let $\boldsymbol{S_z}$, be the sample covariance matrix of $\boldsymbol{z}$, consisting of the elements $\boldsymbol{S_z} = \begin{bmatrix} \boldsymbol{s_y^2} & \boldsymbol{s'_{y,X}} \\ \boldsymbol{s_{y,X}} & \boldsymbol{S_X} \end{bmatrix}$, where $\boldsymbol{s_{y,X}}$ is the $p \times 1$ vector of covariances between y and the x variables.

The aim of this study is to estimate the linear regression $\hat{\boldsymbol{y}} = \hat{\boldsymbol{\beta}}'\boldsymbol{x}$, and it is assumed that the response variable can be linearly explained by a set of a components $\boldsymbol{t}_1, ..., \boldsymbol{t}_k$ with $k << p$, which are linear functions of the x variables. Hence, calling $\boldsymbol{X}$ the $n \times p$ data matrix of the independent variables, and $\boldsymbol{x}_i'$ to its ith row, the following model showed by (2.1) and (2.2) holds (González *et al.*, 2009):

$$\boldsymbol{x}_i = \boldsymbol{P}\boldsymbol{t}_i + \boldsymbol{\varepsilon}_i \,, \tag{2.1}$$

$$\boldsymbol{y}_i = \boldsymbol{q}'\boldsymbol{t}_i + \boldsymbol{\eta}_i \,. \tag{2.2}$$

Here, $\boldsymbol{P}$ is the $p \times k$ matrix of the loadings of the vector $\boldsymbol{t}_i = (t_{i1}, ..., t_{ik})'$ and $\boldsymbol{q}$ is the k-dimensional vector of the y-loadings. The vectors $\boldsymbol{\varepsilon}_i$ and $\boldsymbol{\eta}_i$ have zero mean, follow normal distributions and are uncorrelated. The component matrix $\boldsymbol{T} = (\boldsymbol{t}_1, ..., \boldsymbol{t}_k)'$ is not directly observed and should be estimated. Then, it can be shown that the maximum likelihood estimation of the $\boldsymbol{T}$ matrix is given as in (2.3) (González *et al.*, 2009):

$$\boldsymbol{T} = \boldsymbol{X}\boldsymbol{W}_k \,. \tag{2.3}$$

Here, the loading matrix $\boldsymbol{W}_k = [\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_k]$ is the $p \times k$ matrix of coefficients and the vectors $\boldsymbol{w}_i$, $1 \leq i < k$ are the solution of (2.4) under the constraint in (2.5) with $\boldsymbol{w_1} \boldsymbol{\alpha} \boldsymbol{s_{y,x}}$. Consequently, we can conclude that components $(\boldsymbol{t}_1, ..., \boldsymbol{t}_k)$ are orthogonal (González *et al.*, 2009):

$$\boldsymbol{w}_i = \arg\max_{w} cov^2\left(\boldsymbol{X}\boldsymbol{w}, \boldsymbol{y}\right) \,, \tag{2.4}$$

$$\boldsymbol{w}'\boldsymbol{w} = 1 \quad \texttt{and} \quad \boldsymbol{w}_i'\boldsymbol{S_x}\boldsymbol{w}_j = 0 \;\text{ for }\; 1 \leq j < i \,. \tag{2.5}$$

It can be shown that vectors $\boldsymbol{w}_i$ are found as the eigenvectors linked to the largest eigenvalues of the matrix is given as in (2.6):

$$\left(\boldsymbol{I} - \boldsymbol{P}_x(i)\right) \boldsymbol{s_{y,x}} \boldsymbol{s_{y,x}}' \,. \tag{2.6}$$

$\boldsymbol{P}_x(i)$ is the projection matrix on the space spanned by $\boldsymbol{S_X}\boldsymbol{W}_i$, given by $\boldsymbol{P}_x(i) = (\boldsymbol{S_x}\boldsymbol{W}_i)\left[(\boldsymbol{S_x}\boldsymbol{W}_i)'(\boldsymbol{S_x}\boldsymbol{W}_i)\right]^{-1}(\boldsymbol{S_x}\boldsymbol{W}_i)'$. From these results it is easy to see that the vectors $\boldsymbol{w}_i$ can be computed recursively as in below:

$$\boldsymbol{w_1} \boldsymbol{\alpha} \boldsymbol{s_{y,x}} \,, \tag{2.7}$$

$$\boldsymbol{w_{i+1}} \boldsymbol{\alpha} \boldsymbol{s_{y,x}} - \boldsymbol{S_x}\boldsymbol{W}_i \left(\boldsymbol{W}_i'\boldsymbol{S_x}\boldsymbol{W}_i\right)^{-1} \boldsymbol{W}_i'\boldsymbol{s_{y,x}}, \quad 1 \leq i < k \,. \tag{2.8}$$

It could be mentioned that by using the expressions given by (2.7) and (2.8), it is not necessary to calculate the PLS components $\boldsymbol{t}_i$. In each step of the

algorithm, $\boldsymbol{w}_{i+1}$ only depends on the value of the i previous vectors $\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_i$, on $\boldsymbol{S}_x$ and on $\boldsymbol{s_{y,x}}$. Moreover, as $\boldsymbol{w}_1$ only depends on $\boldsymbol{s_{y,x}}$, the calculation of $\boldsymbol{W}$ is completely fixed by the values of $\boldsymbol{S}_x$ and $\boldsymbol{s_{y,x}}$. Finally, as the regression coefficients in (2.2) are uncorrelated, due to the uncorrelation of the t variables, it is easy to see that the regression coefficients $\hat{\boldsymbol{\beta}}_{\boldsymbol{k}}^{\boldsymbol{PLS}}$ are given by (2.9) (González *et al.*, 2009):

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{k}}^{\boldsymbol{PLS}} = \boldsymbol{W}_k \left(\boldsymbol{W}_k' \boldsymbol{S_x} \boldsymbol{W}_k\right)^{-1} \boldsymbol{W}_k' \boldsymbol{s_{y,x}} \,. \tag{2.9}$$

The application of this algorithm can be seen as a two step procedure: (1) the weights $\boldsymbol{w}_i$, defining the new orthogonal regressor $\boldsymbol{t}_i$, are computed with (2.7) and (2.8) by using the covariance matrix of the observations; (2) the y-loadings $\boldsymbol{q}_i$ are computed by regressing y on individual regressor $\boldsymbol{t}_i$. As it is shown in (2.9) these two steps depend only on the covariance matrix of the observations and it may be thought that if this matrix is properly robustified the procedure will be robust (González *et al.*, 2009).

3. THE NEW PROPOSED ROBUST PLSR METHOD

In this section, the new robust PLSR method, which we proposed based on '*an adaptive reweighted estimator of covariance using MCD estimators in the first step as robust initial estimators of location and covariance*', will be introduced. This adaptive reweighted estimator of covariance will be used in order to robustify the sample covariance matrix, $\boldsymbol{S_z}$, in the PLS1 algorithm. Hence, while defining this estimator, the equations are examined on $\boldsymbol{z}_i = (y_i, \boldsymbol{x}_i),\ i = 1, ..., n \in \Re^{p'}$, here, $p' = p + 1$. In this method, the MCD estimator is calculated by well-known 'FAST-MCD' algorithm. Hence, in this section, firstly, information about MCD estimator and operation of the FAST-MCD algorithm will be given.

Besides high outlier resistance, if robust multivariate estimators are to be of practical use in statistical inference they should offer a reasonable efficiency under the normal model and a manageable asymptotic distribution. However, Minimum Volume Ellipsoid (MVE) and MCD estimators are not in this category. Gervini (2003) stated that by taking care of both robustness and efficiency considerations, the best choice seems to be a two-stage procedure. In this procedure, firstly, a highly robust but perhaps inefficient estimator is computed, which is used for detecting outliers and computing the sample mean and covariance of the 'cleaned' data set as in Rousseeuw and Van Zomeren (1990). This procedure consists of discarding those observations whose Mahalanobis distances exceed a certain fix threshold value. In the previous studies, the MVE was commonly used as initial estimator for these procedures. However, Rousseeuw and Van Driessen (1999) have proposed an algorithm for calculating MCD estimator,

although this algorithm does not guarantee that the exact estimator is found, it is faster and more accurate than previously existing algorithms even for very large data sets ($n >> p' = p + 1$). This fact, added to its $1/\sqrt{n}$ rate of convergence, seems to point to the MCD method using the FAST-MCD algorithm as the current best choice in comparison to MVE for initial estimator of a two-step procedure (Gervini, 2003).

MCD method, proposed by Rousseeuw (1984), is searching for those h data points for which the determinant of the classical covariance matrix is minimal. Hence, the MCD estimators of location and covariance will be the mean and covariance matrix of these h data points, respectively. The calculation of MCD estimation is not simple. Let $\boldsymbol{z}_i' = (y_i, \boldsymbol{x}_i)'$, $i = 1, ..., n$ be an unified data set. The MCD estimator can only be applied to data sets where the number of observations is larger than the number of variables ($n > p' = p + 1$). The reason is that if $p' > n$ then also $p' > h$, and the covariance matrix of any h data points will always be singular, leading to a determinant of zero. Thus, each subset of h data points would lead to the smallest possible determinant, resulting in a non-unique solution (Filzmoser *et al.*, 2009; Polat, 2014).

FAST-MCD algorithm could deal with a sample size n in the tens of thousands. FAST-MCD finds the exact solution for small data sets and it is faster and more accurate than previously existing algorithms, even for very large data sets. Rousseeuw and Van Driessen (1999) suggested to use FAST-MCD algorithm in order to estimate location and covariance as considering the its statistical efficiency and fastness in computation (Rousseeuw and Van Driessen, 1999). In FAST-MCD algorithm as the raw MCD estimators of location and covariance are reweighted in order to improve the finite sample efficiency, they are called as Reweighted Minimum Covariance Determinant (RMCD) estimators (Hubert and Vanden Branden, 2003; Moller *et al.*, 2005).

3.1. Construction of the FAST-MCD algorithm

3.1.1. Basic theorem and the C-step for the FAST-MCD algorithm

A key step of the FAST-MCD algorithm is the fact that starting from any approximation to the MCD, it is possible to compute another approximation with an even lower determinant. 'C-step' procedure, which is used in FAST-MCD algorithm, given in following Theorem 3.1 (Rousseeuw and Van Driessen, 1999).

Theorem 3.1. *Since* $\boldsymbol{z}_i' = (y_i, \boldsymbol{x}_i)'$, $i = 1, ..., n$ *consider a data set* $\boldsymbol{Z}_n = \{\boldsymbol{z}_1, ..., \boldsymbol{z}_n\}$ *of* $p' = p + 1$*-variate observations. Let a set of observations that de-*

fined as $H_1 \subset \{1, ..., n\}$ *with* $|H_1| = h$. *Here,* H_1 *shows the subset of* h *observations having the lowest determinant. Hence, as the location and covariance for subset of* h *observations* $\hat{\boldsymbol{\mu}}_1 := (1/h) \sum_{i \in H_1} \boldsymbol{z}_i$ *and* $\hat{\boldsymbol{\Sigma}}_1 := (1/h) \sum_{i \in H_1} (\boldsymbol{z}_i - \hat{\boldsymbol{\mu}}_1) \cdot (\boldsymbol{z}_i - \hat{\boldsymbol{\mu}}_1)'$, *respectively, if* $det\left(\hat{\boldsymbol{\Sigma}}_1\right) \neq 0$ *then the relative distances are defined as* $d_1(i) := \sqrt{(\boldsymbol{z}_1 - \hat{\boldsymbol{\mu}}_1)' \hat{\boldsymbol{\Sigma}}_1^{-1} (\boldsymbol{z}_1 - \hat{\boldsymbol{\mu}}_1)}$, $i = 1, ..., n$. *Then, a set of observations* H_2 *is taken such that* $\{d_1(i); i \in H_2\} := \{(d_1)_{1:n}, ..., (d_1)_{h:n}\}$ *where* $(d_1)_{1:n} \leq (d_1)_{2:n} \leq \cdots \leq (d_1)_{n:n}$ *are the ordered distances, and* $\hat{\boldsymbol{\mu}}_2$ *and* $\hat{\boldsymbol{\Sigma}}_2$ *are computed based on* H_2. *Then,* $det\left(\hat{\boldsymbol{\Sigma}}_2\right) \leq det\left(\hat{\boldsymbol{\Sigma}}_1\right)$ *with equality if and only if* $\hat{\boldsymbol{\mu}}_2 = \hat{\boldsymbol{\mu}}_1$ *and* $\hat{\boldsymbol{\Sigma}}_2 = \hat{\boldsymbol{\Sigma}}_1$ *(Polat, 2014; Rousseeuw and Van Driessen, 1999).*

If $det\left(\hat{\boldsymbol{\Sigma}}_1\right) > 0$, applying the Theorem 3.1 yields $\hat{\boldsymbol{\Sigma}}_2$ with $det\left(\hat{\boldsymbol{\Sigma}}_2\right) \leq det\left(\hat{\boldsymbol{\Sigma}}_1\right)$. In FAST-MCD algorithm the construction in Theorem 3.1 is referred to as 'C-step', where 'C' can be taken to stand for 'covariance' since $\hat{\boldsymbol{\Sigma}}_2$ is the covariance matrix of H_2, or for 'concentration' since we concentrate on the h observations with smallest distances, and $\hat{\boldsymbol{\Sigma}}_2$ is more concentrated (has a lower determinant) than $\hat{\boldsymbol{\Sigma}}_1$ (Rousseeuw and Van Driessen, 1999).

Repeating C-steps yields an iteration process. If $det\left(\hat{\boldsymbol{\Sigma}}_2\right) = 0$ or $det\left(\hat{\boldsymbol{\Sigma}}_2\right) = det\left(\hat{\boldsymbol{\Sigma}}_1\right)$ we stop; otherwise we run another C-step yielding $det\left(\hat{\boldsymbol{\Sigma}}_3\right)$, and so on. The sequence $det\left(\hat{\boldsymbol{\Sigma}}_1\right) \geq det\left(\hat{\boldsymbol{\Sigma}}_2\right) \geq det\left(\hat{\boldsymbol{\Sigma}}_3\right) \geq ...$ is nonnegative and hence must converge. In fact, since there are only finitely many h subsets there must be an index m such that $det\left(\hat{\boldsymbol{\Sigma}}_m\right) = 0$ or $det\left(\hat{\boldsymbol{\Sigma}}_m\right) = det\left(\hat{\boldsymbol{\Sigma}}_{m-1}\right)$, hence convergence is reached. In practice, m is often below 10. Afterwards, running the C-step on $\left(\hat{\boldsymbol{\mu}}_m, \hat{\boldsymbol{\Sigma}}_m\right)$ no longer reduces the determinant. This is not sufficient for $det\left(\hat{\boldsymbol{\Sigma}}_m\right)$ to be the global minimum of the MCD objective function, but it is a necessary condition (Rousseeuw and Van Driessen, 1999). Thus, Theorem 3.1 provides a partial idea for an algorithm: '*Take many initial choices of* H_1 *and apply C-steps to each until convergence, and keep the solution with lowest determinant*'. However, several things must be decided to make this idea operational: how to generate sets H_1 to begin with, how many H_1 are needed, how to avoid duplication of work since several H_1 may yield the same solution, can't we do with fewer C-steps, what about large sample sizes, and so on. These matters will be discussed in the next sections.

3.1.2. Creating initial subsets H_1

In order to apply the algorithmic idea given in the previous section, it must be decided how to construct the initial subsets H_1. For this purpose, first of all, a

random $(p'+1)$-subset J must be drawn according to method given in Rousseeuw and Van Driessen (1999) study and then $\hat{\boldsymbol{\mu}}_0 := ave(J)$ and $\hat{\boldsymbol{\Sigma}}_0 := cov(J)$ must be computed. If $det\left(\hat{\boldsymbol{\Sigma}}_0\right) = 0$ then extend J by adding another random observation, and continue adding observations until $det\left(\hat{\boldsymbol{\Sigma}}_0\right) > 0$. Then compute the distances $d_0^2(i) := (\boldsymbol{z}_i - \hat{\boldsymbol{\mu}}_0)' \hat{\boldsymbol{\Sigma}}_0^{-1} (\boldsymbol{z}_i - \hat{\boldsymbol{\mu}}_0)$ for $i = 1, ..., n$. Sort them into $d_0(\pi(1)) \leq ... \leq d_0(\pi(n))$ and put $H_1 := \{\pi(1), ..., \pi(h)\}$. Rousseeuw and Van Driessen (1999) mentioned that it would be useless to draw fewer than $p'+1$ points, since then $\hat{\boldsymbol{\Sigma}}_0$ is always singular (Polat, 2014; Rousseeuw and Van Driessen, 1999).

3.1.3. Selective iteration

Each C-step calculates a covariance matrix, its determinant, and all relative distances. Therefore, reducing the number of C-steps would improve the speed. Rousseeuw and Van Driessen (1999) mentioned that often the distinction between good (robust) solutions and bad solutions already becomes visible after two or three C-steps. Moreover, they proposed to take only two C-steps from each initial subsample, select the 10 different subsets with the lowest determinants, and only for these 10 to continue taking C-steps until convergence (Rousseeuw and Van Driessen, 1999).

3.1.4. Nested extensions

For a small sample size n, the above algorithm, which was mentioned in Section 3.1.1, does not take much time. But when n grows, the computation time increases, mainly due to the n distances that needed to be calculated each time. To avoid doing all the computations in the entire data set, Rousseeuw and Van Driessen (1999) considered a special structure. When $n > 1500$, the algorithm generates a nested system of subsets which looks like in Figure 1, where the arrows mean 'is a subset of'.

In Figure 1 the five subsets of size 300 do not overlap, and together they form the merged set of size 1500, which in turn is a proper subset of the data set of size n. Since the method showed in Figure 1 work with two stages, 'nested' name is used. To construct the Figure 1 the algorithm draws 1500 observations, one by one, without replacement. The first 300 observations, that it encounters, are put in the first subset, and so on. Because of this mechanism each subset of size 300 is roughly representative for the data set, and the merged set with 1500 cases even more representative. When $n < 600$ the algorithm operates as in the

previous Section 3.1.1. However, when $n \geq 1500$ Figure 1 is used (Rousseeuw and Van Driessen, 1999).

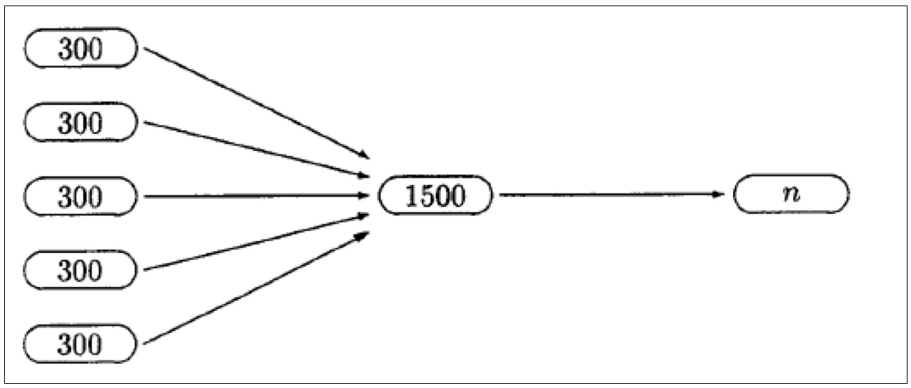

Figure 1: Nested system of subsets generated by the FAST-MCD algorithm.

3.2. The implementation of the FAST-MCD algorithm

Combining all the components of the preceding sections yields the FAST-MCD algorithm. The steps of the algorithm for $p' = p + 1$ dimensional unified vector $\boldsymbol{z}_i' = (y_i, \boldsymbol{x}_i)'$, $i = 1, ..., n$ are given as in below (Polat, 2014; Rousseeuw and Van Driessen, 1999).

Step 1: The MCD estimates can resist $(n - h)$ outliers, hence the number h (or equivalently the proportion $\alpha = h/n$) determines the robustness of the estimator. The default h value is $[(n + p' + 1)/2]$ in FAST-MCD algorithm and the highest resistance towards contamination is achieved by taking this value. However, the user may choose any integer h with $[(n + p' + 1)/2] \leq h < n$. When a large proportion of contamination is presumed in data set, h should thus be chosen $h = [0.5n]$ with $\alpha = 0.5$. Otherwise if it is exact that the data contains less than 25% of contamination, which is usually the case, a good compromise between breakdown value and statistical efficiency is obtained by putting $h = [0.75n]$ (Polat, 2014; Rousseeuw and Van Driessen, 1999).

Step 2: From here on $h < n$ and $p' \geq 2$. If n is small (say, $n < 600$) then,

- repeat (say) 500 times:
  - construct an initial h-subset H_1 using method in Section 3.1.2, i.e. starting from a random $(p' + 1)$-subset,
  - carry out two C-steps described in Section 3.1.1;

- for the 10 results with lowest $det\left(\hat{\boldsymbol{\Sigma}}_3\right)$:
  - carry out C-steps until convergence;
- report the solution $\left(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}\right)$ with the lowest $det\left(\hat{\boldsymbol{\Sigma}}\right)$.

Step 3: If n is larger (say, $n \geq 600$) then,

- construct up to five disjoint random subsets of size n_{sub} according to Section 3.1.4 (say, subsets of size $n_{sub} = 300$);
- inside each subset, repeat $500/5 = 100$ times:
  - construct an initial subset H_1 of size $h_{sub} = [n_{sub}\,(h/n)]$,
  - carry out two C-steps, using n_{sub} and h_{sub},
  - keep the 10 best results $\left(\hat{\boldsymbol{\mu}}_{sub}, \hat{\boldsymbol{\Sigma}}_{sub}\right)$;
- pool the subsets, yielding the merged set (say, of size $n_{merged} = 1500$);
- in the merged set, repeat for each of the 50 solutions $\left(\hat{\boldsymbol{\mu}}_{sub}, \hat{\boldsymbol{\Sigma}}_{sub}\right)$:
  - carry out two C-steps, using n_{merged} and $h_{merged} = [n_{merged}\,(h/n)]$,
  - keep the 10 best results $\left(\hat{\boldsymbol{\mu}}_{merged}, \hat{\boldsymbol{\Sigma}}_{merged}\right)$;
- in the full data set, repeat for the m_{full} best results:
  - take several C-steps, using n and h,
  - keep the best final result $\left(\hat{\boldsymbol{\mu}}_{full}, \hat{\boldsymbol{\Sigma}}_{full}\right)$.

Here, m_{full} and the number of C-steps (preferably, until convergence) depend on how large the data set is (Polat, 2014; Rousseeuw and Van Driessen, 1999).

This algorithm called as FAST-MCD. It is affine equivariant: when the data are translated or subjected to a linear transformation, the resulting $\left(\hat{\boldsymbol{\mu}}_{full}, \hat{\boldsymbol{\Sigma}}_{full}\right)$ will transform accordingly. For convenience, the computer program contains two more steps (Rousseeuw and Van Driessen, 1999):

Step 4: In order to obtain consistency when the data come from a multivariate normal distribution, $\hat{\boldsymbol{\mu}}_{\text{MCD}} = \hat{\boldsymbol{\mu}}_{full}$ and $\hat{\boldsymbol{\Sigma}}_{\text{MCD}} = \dfrac{\text{med}_i\; d^2_{\left(\hat{\boldsymbol{\mu}}_{full}, \hat{\boldsymbol{\Sigma}}_{full}\right)}(i)}{\chi^2_{p',0.5}} \hat{\boldsymbol{\Sigma}}_{full}$ are putted.

Step 5: In order to obtain 'one-step reweighted' estimates, each observation is reweighted as in (3.1). Hence, by using these weights, the RMCD estimators are obtained as in (3.2):

$$w_i = \begin{cases} 1, \text{ if } \left(\boldsymbol{z}_i - \hat{\boldsymbol{\mu}}_{\text{MCD}}\right)' \hat{\boldsymbol{\Sigma}}^{-1}_{\text{MCD}} \left(\boldsymbol{z}_i - \hat{\boldsymbol{\mu}}_{\text{MCD}}\right) \leq \chi^2_{p',0.975}\,, \\ 0, \text{ otherwise}\,. \end{cases} \tag{3.1}$$

$$
\begin{aligned}
\hat{\boldsymbol{\mu}}_{\mathrm{RMCD}} &= \frac{\sum_{i=1}^{n} w_i \boldsymbol{z}_i}{\sum_{i=1}^{n} w_i}\,, \\
\hat{\boldsymbol{\Sigma}}_{\mathrm{RMCD}} &= \frac{\sum_{i=1}^{n} w_i \left(\boldsymbol{z}_i - \hat{\boldsymbol{\mu}}_{\mathrm{RMCD}}\right)\left(\boldsymbol{z}_i - \hat{\boldsymbol{\mu}}_{\mathrm{RMCD}}\right)'}{\sum_{i=1}^{n} w_i}\,.
\end{aligned}
\tag{3.2}
$$

The FAST-MCD algorithm code named as '*mcdcov*' could be found in MATLAB LIBRA Toolbox which is written by Verboven and Hubert (2005). The implementation of mcdcov function could be given briefly as in below (Polat, 2014; Verboven and Hubert, 2005):

- The data set contains n observations and $p' = p + 1$ variables. When $n < 600$, the algorithm analyzes the data set as a whole. When the data set is analyzed as a whole, a subsample of $p' + 1$ observations is taken, of which of them the mean and covariance matrix are calculated. The h observations with smallest relative distances are used to calculate the next mean and covariance matrix, and this cycle is repeated two C-step times. FAST-MCD algorithm is a resampling algorithm. 500 subsets of size $p' + 1$ out of n are drawn randomly. Afterwards, the 10 best solutions (means and corresponding covariance matrices) are used as starting values for the final iteration. The number of the subsets is chosen as '500' to ensure a high probability of sampling at least one clean subset. These iterations stop when two subsequent determinants become equal. At most three C-step iteration are done. The solution with smallest determinant (location and covariance) is retained.

- However, when $n \geq 600$ (whether $n < 1500$ or not), the algorithm does part of the calculations on (at most) 5 non-overlapping subsets of (roughly) 1500 observations. In this case, the algorithm functions in three stages.
  - Stage 1: For each H_1 subsample in each subset, two C-steps iterations are carried out in that subset. In this stage, 5 subsets and 500 subsamples are chosen. For each subset, the 10 best solutions (location and covariance) are stored.
  - Stage 2: Then the subsets are pooled, yielding a merged set with at most 1500 observations. If n is large, the merged set is a proper subset of the entire data set. In this merged set, each of these (at most 50) best solutions $\left(\hat{\boldsymbol{\mu}}_{sub}, \hat{\boldsymbol{\Sigma}}_{sub}\right)$ of Stage 1 are used as starting values for C-step iterations. In this stage, starting from each $\left(\hat{\boldsymbol{\mu}}_{sub}, \hat{\boldsymbol{\Sigma}}_{sub}\right)$, it is continued taking C-steps by using all 1500 observations in the merged set. Also here, the 10 best solutions $\left(\hat{\boldsymbol{\mu}}_{merged}, \hat{\boldsymbol{\Sigma}}_{merged}\right)$ are stored.
  - Stage 3: This stage depends on n, the total number of observations in the data set. Finally, each of these 10 solutions is extended to the

full data set in the same way and the best $\left(\hat{\boldsymbol{\mu}}_{full}, \hat{\boldsymbol{\Sigma}}_{full}\right)$ solution is obtained. Since the final computations are carried out in the entire data set, they take more time when n increases. Rousseeuw and Van Driessen (1999) mentioned that the number of initial solutions $\left(\hat{\boldsymbol{\mu}}_{merged}, \hat{\boldsymbol{\Sigma}}_{merged}\right)$ and/or the number of C-steps in the full data set could be limited in order to speed up the algorithm as n becomes large (Rousseeuw and Van Driessen, 1999; Verboven and Hubert, 2005). Therefore, the default values of '*mcdcov*' function are: If $n \leq 5000$, all 10 preliminary solutions are iterated. If $n > 5000$, only the best preliminary solution is iterated. The number of iterations decreases to 1 according to $n \times p$. If $n \times p \leq 100000$, the number of C-steps take on the full data set in the Stage 3 iterate three times, whereas for $n \times p > 1000000$ only one iteration step is taken.

In the next section, information about 'a robust and efficient adaptive reweighted covariance estimator', which was proposed in Gervini (2003), will be given. This robust covariance estimator is constructed by using MCD estimators in the first step as robust initial estimators of location and covariance.

3.3. A robust and efficient adaptive reweighted estimator of covariance

In the context of linear regression, many estimators have been proposed that aim to reconcile high efficiency and robustness. Overall, if one wants to take care of both robustness and efficiency considerations, the best choice seems to be a two-stage procedure. Gervini (2003) proposed essentially an improvement over Rousseeuw and Van Zomeren (1990). It consists of a reweighted one-step estimator that uses adaptive threshold values. This adaptive reweighting scheme is able to maintain the outlier resistance of the initial estimator in breakdown and bias and, at the same time, attain 100% efficiency at the normal distribution. This kind of adaptive reweighting was first proposed in Gervini (2002) for the linear regression model. In Gervini (2003), this idea is extended and an adaptive method is proposed for multivariate location and covariance estimation.

Given a sample $\boldsymbol{z}_1, ..., \boldsymbol{z}_n$ in $\Re^{p'}$ with $p' = p+1$ and initial robust estimators of location and covariance $\left(\hat{\boldsymbol{\mu}}_{0n}, \hat{\boldsymbol{\Sigma}}_{0n}\right)$ consider the Mahalanobis distances given in (3.3) (Gervini, 2003; Polat, 2014):

$$d_i := d\left(\boldsymbol{z}_i, \hat{\boldsymbol{\mu}}_{0n}, \hat{\boldsymbol{\Sigma}}_{0n}\right) = \left\{\left(\boldsymbol{z}_i - \hat{\boldsymbol{\mu}}_{0n}\right)' \hat{\boldsymbol{\Sigma}}_{0n}^{-1} \left(\boldsymbol{z}_i - \hat{\boldsymbol{\mu}}_{0n}\right)\right\}^{1/2} . \tag{3.3}$$

An outlier will typically have a larger Mahalanobis distance than a 'good' observation. If one assumes a normal distribution, d_i^2 is approximately $\chi_{p'}^2$ distributed and it is reasonable to suspect of those observations with, for instance,

$d_i^2 \geq \chi^2_{p',0.975}$. What Rousseeuw and Van Zomeren (1990) propose is to skip those outlying observations and compute the sample mean and covariance matrix of the rest of the data, obtaining in this way new estimators $\left(\hat{\boldsymbol{\mu}}_{1n}, \hat{\boldsymbol{\Sigma}}_{1n}\right)$ (Gervini, 2003; Polat, 2014).

Since the MCD method calculated by FAST-MCD algorithm is improved as a good alternative to MVE method, Gervini (2003) stated that MCD estimators could be used as the initial robust estimators of location and covariance in the 'adaptive reweighted' method. Hence, in this study, in 'adaptive reweighted' method using the MCD estimators $\left(\hat{\boldsymbol{\mu}}_{\text{MCD}}, \hat{\boldsymbol{\Sigma}}_{\text{MCD}}\right)$ as initial robust estimators of location and covariance $\left(\hat{\boldsymbol{\mu}}_{0n}, \hat{\boldsymbol{\Sigma}}_{0n}\right)$, the obtained robust location and covariance estimators $\left(\hat{\boldsymbol{\mu}}_{1n}, \hat{\boldsymbol{\Sigma}}_{1n}\right)$ are called as 'Adaptive Reweighted Minimum Covariance Determinant/ARWMCD' estimators $\left(\hat{\boldsymbol{\mu}}_{\text{ARWMCD}}, \hat{\boldsymbol{\Sigma}}_{\text{ARWMCD}}\right)$(Gervini, 2003; Polat, 2014).

This reweighting step given in Gervini (2003) is known to improve the efficiency of the initial estimator while retaining (most of) its robustness. However, the threshold value $\chi^2_{p',0.975}$ is an arbitrary number. For large data sets a considerable number of observations have to be discarded from the analysis even if they follow the normal model. One way to avoid this problem is to increase the threshold value to another arbitrary fix number, however, in this case the bias of the reweighted estimator will be affected. Hence, a better alternative is to use 'an adaptive threshold value' that increases with n if the data is 'clean' but remains bounded if there are outliers in the sample. Gervini (2003), proposed a method of constructing such adaptive threshold values. Let (3.4) be the empirical distribution of the squared Mahalanobis distances (Gervini, 2003; Polat, 2014):

$$G_n(u) = \frac{1}{n}\sum_{i=1}^{n} I\left(d^2\left(\boldsymbol{z}_i, \hat{\boldsymbol{\mu}}_{\text{MCD}}, \hat{\boldsymbol{\Sigma}}_{\text{MCD}}\right) \leq u\right) . \tag{3.4}$$

Let $G_{p'}(u)$ be the $\chi^2_{p'}$ distribution function. For a normally distributed sample it is expected to G_n to converge to $G_{p'}$. Therefore, a way to detect outliers is to compare the tails of G_n with the tails of $G_{p'}$. If $\eta = \chi^2_{p',1-\alpha}$ for a certain small α, say $\alpha = 0.025$, (3.5) is defined (Gervini, 2003; Polat, 2014)

$$\alpha_n = \sup_{u \geq \eta} \left\{G_{p'}(u) - G_n(u)\right\}^+ , \tag{3.5}$$

where $\{\cdot\}^+$ indicates the positive part. This α_n can be regarded as a measure of outliers in the sample. Since a negative difference would not indicate presence of outliers, it is only taken into account positive differences in (3.5). If $d^2_{(i)}$ denotes the ith order statistic of the squared Mahalanobis distances and $i_0 = max\left\{i : d^2_{(i)} < \eta\right\}$, then (3.5) comes down to as in (3.6) (Gervini, 2003; Po-

lat, 2014):

$$\alpha_n = \max_{i>i_0} \left\{ G_{p'}(d^2_{(i)}) - \frac{i-1}{n} \right\}^+ . \tag{3.6}$$

Those observations corresponding to the largest $\lfloor \alpha_n n \rfloor$ distances are considered as outliers and eliminated in the reweighting step. Here $\lfloor a \rfloor$, is the largest integer that is less than or equal to a. The cut-off value is then defined as in (3.7) where as usual $G_n^{-1}(u) = \min\{s : G_n(s) \geq u\}$. Note that $c_n = d^2_{(i_n)}$ with $i_n = n - \lfloor \alpha_n n \rfloor$ and that $i_n > i_0$ as a consequence of the definition of α_n. Therefore, $c_n > \eta$ (Gervini, 2003; Polat, 2014):

$$c_n = G_n^{-1}(1 - \alpha_n) . \tag{3.7}$$

To define the reweighted estimator, weights of the form in (3.8) are used (Gervini, 2003; Polat, 2014):

$$w_{in} = w\left(\frac{d^2\left(\boldsymbol{z}_i, \hat{\boldsymbol{\mu}}_{\text{MCD}}, \hat{\boldsymbol{\Sigma}}_{\text{MCD}}\right)}{c_n} \right) . \tag{3.8}$$

Here, the weight function that satisfies $(\boldsymbol{W})$ $w : [0, \infty) \to [0, 1]$ is non-increasing, $w(0) = 1$, $w(u) > 0$ for $u \in [0, 1)$ and $w(u) = 0$ for $u \in [1, \infty)$. The simplest choice among those functions satisfying $(\boldsymbol{W})$ is the hard-rejection function $w(u) = I(u < 1)$ which is the one most commonly used in the practice.

Once weights in (3.8) are computed, the one-step reweighted estimators $\left(\hat{\boldsymbol{\mu}}_{\text{ARWMCD}}, \hat{\boldsymbol{\Sigma}}_{\text{ARWMCD}}\right)$ are defined as in (3.9) and (3.10) (Gervini, 2003; Polat, 2014):

$$\hat{\boldsymbol{\mu}}_{\text{ARWMCD}} = \frac{\sum_{i=1}^n w_{in} \boldsymbol{z}_i}{\sum_{i=1}^n w_{in}} , \tag{3.9}$$

$$\hat{\boldsymbol{\Sigma}}_{\text{ARWMCD}} = \frac{\sum_{i=1}^n w_{in} \left(\boldsymbol{z}_i - \hat{\boldsymbol{\mu}}_{\text{ARWMCD}}\right)\left(\boldsymbol{z}_i - \hat{\boldsymbol{\mu}}_{\text{ARWMCD}}\right)'}{\sum_{i=1}^n w_{in}} . \tag{3.10}$$

It is clear that under appropriate conditions, the threshold values in (3.7) will tend to infinity under the multivariate normal model and then (3.9) and (3.10) will be asymptotically equivalent to the common sample mean and covariance, and thus attain full asymptotic efficiency (Gervini, 2003).

Finally, in this study, first of all, by using robust covariance estimator $\hat{\boldsymbol{\Sigma}}_{\text{ARWMCD}}$ that it is given in (3.10), the robust covariance estimator $\hat{\boldsymbol{S}}_z$ of $\boldsymbol{S}_z = \begin{bmatrix} \boldsymbol{s}_{\boldsymbol{y}}^2 & \boldsymbol{s}'_{\boldsymbol{y},\boldsymbol{X}} \\ \boldsymbol{s}_{\boldsymbol{y},\boldsymbol{X}} & \boldsymbol{S}_{\boldsymbol{X}} \end{bmatrix}$ is obtained. Then, by using robust covariance estimator $\hat{\boldsymbol{S}}_z$ in

the alternative definition of PLS1 algorithm given between (2.7)–(2.9), a new robust PLSR method called 'PLS-ARWMCD' is proposed. The steps of the PLS-ARWMCD algorithm could be given as in (3.11) (Polat, 2014):

$$
\begin{aligned}
& \boldsymbol{w_1}\boldsymbol{\alpha}\hat{\boldsymbol{s}}_{y,x}\,, \\
& \boldsymbol{w}_{i+1}\boldsymbol{\alpha}\hat{\boldsymbol{s}}_{y,x} - \hat{\boldsymbol{S}}_x\boldsymbol{W}_i\left(\boldsymbol{W}_i'\hat{\boldsymbol{S}}_x\boldsymbol{W}_i\right)^{-1}\boldsymbol{W}_i'\hat{\boldsymbol{s}}_{y,x}, \quad 1 \le i < k\,, \\
& \hat{\boldsymbol{\beta}}_k^{\mathbf{PLS-ARWMCD}} = \boldsymbol{W}_k\left(\boldsymbol{W}_k'\hat{\boldsymbol{S}}_x\boldsymbol{W}_k\right)^{-1}\boldsymbol{W}_k'\hat{\boldsymbol{s}}_{y,x}\,.
\end{aligned}
\tag{3.11}
$$

Here, the robust covariance estimations $\hat{\boldsymbol{s}}_{y,x}$ and $\hat{\boldsymbol{S}}_x$ are obtained by decomposing the robust covariance estimation of unified data set $\boldsymbol{z}_i' = (y_i, \boldsymbol{x}_i)'$, $i = 1, ..., n$, which is calculated by ARWMCD estimator, as in $\hat{\boldsymbol{S}}_z = \begin{bmatrix} \hat{\boldsymbol{s}}_{\boldsymbol{y}}^2 & \hat{\boldsymbol{s}}_{\boldsymbol{y},\boldsymbol{X}}' \\ \hat{\boldsymbol{s}}_{\boldsymbol{y},\boldsymbol{X}} & \hat{\boldsymbol{S}}_{\boldsymbol{X}} \end{bmatrix}$ (Polat, 2014).

4. SIMULATION STUDY

In this section, the new proposed robust PLS-ARWMCD method is compared with other four robust PLSR methods RSIMPLS (Hubert and Vanden Branden, 2003), PRM (Serneels *et al.*, 2005), PLS-SD (Gil and Romera, 1998), PLS-KurSD (González *et al.*, 2009) and the classical PLSR method in order to validate the good properties of the new PLSR robustification. The new proposed robust PLS-ARWMCD method and the other five methods (including the classical method) are compared in terms of efficiency, goodness-of-fit (GOF) and predictive ability by performing a simulation study on uncontaminated and contaminated data sets.

According to the initial models given in (2.1) and (2.2), and following a simulation design similar as the one described in González *et al.* (2009), we have generated the data sets as in (4.1):

$$
\begin{aligned}
& \boldsymbol{T} \sim N_2\left(\mathbf{0}_2, \boldsymbol{\Sigma}_t\right)\,, \\
& \boldsymbol{X} = \boldsymbol{T}\boldsymbol{I}_{2,p} + N_p\left(\mathbf{0}_p, 0.1\boldsymbol{I}_p\right)\,, \\
& \boldsymbol{y} = \boldsymbol{T}\boldsymbol{A}_{2,1} + N(0,1)\,.
\end{aligned}
\tag{4.1}
$$

Here, $(\boldsymbol{I}_{k,p})_{i,j} = 1$, for $i = j$ and $(\boldsymbol{I}_{k,p})_{i,j} = 0$, otherwise; $\boldsymbol{I}_p$ is $p \times p$ dimensional identity matrix; $\mathbf{0}_2 = (0,0)'$ is a two-dimensional vector of zeros and $\boldsymbol{A}_{2,1} = (1,1)'$ is a two-dimensional vector of ones and $\boldsymbol{T}$ is the $n \times 2$ dimensional component matrix. Furthermore, we select $n = 200$, $p = 5$, $k = 2$ and we set $\boldsymbol{\Sigma}_t = \begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix}$.

In this simulation study, the performance of the new proposed robust PLS-ARWMCD method is compared with other four robust PLSR methods existing in the literature and the classical method in the presence of five types of contamination.

1. Bad leverage points, which occurs when an observation is far away from the regression hyperplane while its projection onto the regression hyperplane falls outside the large majority of the projected observations (good observations):

$$\text{Bad Leverage Points}: \begin{array}{l} \boldsymbol{T}_\epsilon \sim N_2\left(\mathbf{10}_2, \boldsymbol{\Sigma}_t\right), \\ \boldsymbol{X}_\epsilon = \boldsymbol{T}_\epsilon \boldsymbol{I}_{2,p} + N_p\left(\mathbf{0}_p, 0.1\boldsymbol{I}_p\right). \end{array}$$

2. Vertical outliers, which are observations with large distance from the hyperplane but with projections within the large majority of the projected observations:

$$\text{Vertical outliers}: \ \boldsymbol{y}_\epsilon = \boldsymbol{T}\boldsymbol{A}_{2,1} + N(10, 0.1).$$

3. Good leverage points, which are observations located in the vicinity of the hyperplane but far away from the cluster of the large majority of the observations:

$$\text{Good Leverage Points}: \begin{array}{l} \boldsymbol{T}_\epsilon \sim N_2\left(\mathbf{10}_2, \boldsymbol{\Sigma}_t\right), \\ \boldsymbol{X}_\epsilon = \boldsymbol{T}_\epsilon \boldsymbol{I}_{2,p} + N_p\left(\left(\mathbf{0}_2, \mathbf{10}_{p-2}\right), 0.1\boldsymbol{I}_p\right). \end{array}$$

4. Concentrated Outliers, which are clusters of bad leverage points:

$$\text{Concentrated Outliers}: \begin{array}{l} \boldsymbol{T}_\epsilon \sim N_2\left(\mathbf{10}_2, \boldsymbol{\Sigma}_t\right), \\ \boldsymbol{X}_\epsilon = \boldsymbol{T}_\epsilon \boldsymbol{I}_{2,p} + N_p\left(\mathbf{10}_p, 0.001\boldsymbol{I}_p\right). \end{array}$$

5. Orthogonal outliers, which were first used by Hubert and Vanden Branden (2003). They have the property that they lie far from the t-space, but they become regular observations after projection in the t-space. Hence they will not badly influence the computation of the regression parameters, but they might influence the loadings:

$$\text{Orthogonal outliers}: \ \boldsymbol{X}_\epsilon = \boldsymbol{T}\boldsymbol{I}_{2,p} + N_p\left(\left(\mathbf{0}_2, \mathbf{10}_{p-2}\right), 0.1\boldsymbol{I}_p\right).$$

For each situation, $m = 1000$ data sets were generated. The efficiency of the considered methods is evaluated by means of the MSE of the estimated regression parameters $\hat{\boldsymbol{\beta}}$ that is defined as in (4.2). Moreover, it is clear that the true parameter vector is determined as $\boldsymbol{\beta}_{p,1} = \boldsymbol{I}'_{p,2}\boldsymbol{A}_{2,1}$. Here, $\hat{\boldsymbol{\beta}}_k^{(l)}$ denotes the estimated parameter based on k components in the lth simulation. The MSE indicates to what extent the slope and intercept are correctly estimated. Therefore, the aim is to obtain a MSE value close to zero (Engelen *et al.*, 2004):

$$\text{MSE}_k\left(\hat{\boldsymbol{\beta}}\right) = \frac{1}{m}\sum_{l=1}^{m}\left\|\hat{\boldsymbol{\beta}}_k^{(l)} - \boldsymbol{\beta}\right\|^2. \tag{4.2}$$

Furthermore, we are interested on how well the methods fit the regular data points. Because of the simulation settings, we know exactly their indices as we store in the set G_r. Then, the GOF criterion is defined as in (4.3). Here $r_{i,k}$ is the residual of the ith observation when k components are computed. The objective is to obtain a GOF value close to 1 (Engelen *et al.*, 2004):

$$\text{GOF}_k = 1 - \frac{\underset{i \in G_r}{var}\,(r_i, k)}{\underset{i \in G_r}{var}\,(y_i)}\,. \tag{4.3}$$

The predictive ability of the methods could be measured by means of the Root Mean Squared Error (RMSE). First a test set G_t of uncontaminated data points with size $n_t = 100$ is generated and then (4.4) is computed. Here, $\hat{y}_{i,k}$ is the predicted y-value of observation i from the test set when the regression parameter estimates are based on the training set (X, Y) of size n and k components are retained in the model (Engelen *et al.*, 2004):

$$\text{RMSE}_k = \sqrt{\frac{1}{n_t}\sum_{i=1}^{n_t}(y_i - \hat{y}_{i,k})^2}\,. \tag{4.4}$$

After $m = 1000$ replications, the mean angle (denoted by mean(angle)) between the estimated slope $\hat{\boldsymbol{\beta}}_{[\boldsymbol{y}_\epsilon, \boldsymbol{X}_\epsilon], \boldsymbol{k}}$ and the true slope β are also evaluated and included in the simulation results (González *et al.*, 2009; Hubert and Vanden Branden, 2003).

The results obtained according to simulation settings given in above for the data sets uncontaminated and contaminated by replacing first 10% and 20% of the observations by different types of outliers: bad leverage points, vertical outliers, good leverage points, concentrated outliers and orthogonal outliers. The simulation results for the $n = 200$, $p = 5$, $k = 2$ when the proportion of outliers is 10% given in Table 1. The simulation results for the same simulation setting when the proportion of outliers is 20% given in Table 2.

Table 1 shows that in case of no contamination is added the new proposed robust PLS-ARWMCD method and the four robust PLSR methods existing in the literature (RSIMPLS, PRM, PLS-SD, PLS-KurSD) have nearly close performance to classical PLSR method in terms of efficiency, fitting to data and predictive ability. However, when the data set is contaminated by different types of outliers, the four robust PLSR methods existing in literature and the new proposed robust PLSR method outperform the classical PLSR method especially in terms of efficiency and predictive ability. Especially when the data contain bad leverage points or concentrated outliers, the performance of classical PLSR method in terms of efficiency, fitting to data and predictive ability is much lower than the new proposed robust PLS-ARWMCD method. The mean angle values

between the estimated slope and the true slope for the classical PLSR method are also higher than the new proposed robust PLS-ARWMCD method for these two types of outliers.

Table 1: The sample size is $n = 200$, $p = 5$ and $k = 2$, the proportion of outliers is 10%.

	PLSR	RSIMPLS	PRM	PLS- SD	PLS-KurSD	**PLS-ARWMCD**
No Contamination						
MSE	0.0092	0.0111	0.0101	0.0104	0.01	0.0105
GOF	0.8312	0.8308	0.8308	0.8308	0.8309	0.8308
RMSE	1.0961	1.0974	1.0969	1.0973	1.0968	1.0974
Mean(angle)	0.0446	0.0519	0.0462	0.0492	0.0477	0.0491
Bad Leverage Points						
MSE	1.7184	0.0115	0.0688	0.0969	0.0109	0.0104
GOF	0.2585	0.8306	0.8177	0.8098	0.8307	0.8309
RMSE	2.2892	1.0996	1.1413	1.1654	1.0996	1.0989
Mean(angle)	1.1403	0.0515	0.0796	0.0943	0.0496	0.0478
Vertical Outliers						
MSE	0.0489	0.0107	0.0121	0.0118	0.0113	0.0106
GOF	0.817	0.8295	0.8294	0.8296	0.8299	0.83
RMSE	1.1384	1.0989	1.0998	1.0998	1.0987	1.0981
Mean(angle)	0.113	0.0467	0.0516	0.0526	0.0507	0.0485
Good Leverage Points						
MSE	1.0282	0.0118	1.0346	0.0162	0.0109	0.0103
GOF	0.6988	0.8307	0.7721	0.8305	0.8307	0.8309
RMSE	1.4658	1.0996	1.2789	1.1002	1.0996	1.0988
Mean(angle)	0.768	0.053	0.7027	0.0583	0.0496	0.0476
Concentrated Outliers						
MSE	1.9646	0.0118	1.6318	0.03	0.0109	0.0104
GOF	0.5093	0.8307	0.7503	0.8281	0.8307	0.8309
RMSE	1.8671	1.0996	1.3228	1.1078	1.0996	1.0989
Mean(angle)	1.1031	0.0529	0.6964	0.0707	0.0496	0.0478
Orthogonal Outliers						
MSE	0.1815	0.0137	0.1341	0.0107	0.0109	0.0103
GOF	0.7847	0.8295	0.7988	0.8298	0.8298	0.83
RMSE	1.2316	1.1002	1.1917	1.0996	1.0997	1.099
Mean(angle)	0.2821	0.0575	0.2323	0.0494	0.0503	0.0488

Table 1 shows that there are no big differences between the classical method and the robust PLSR methods (including the new proposed robust PLS-ARWMCD method) in terms of fitting to data for the contaminated data sets with the exception of good leverage points, bad leverage points and concentrated outliers. It could be mentioned that for all the types of outliers the new proposed robust PLS-ARWMCD method comes to the forefront with robust RSIMPLS and PLS-KurSD methods existing in the literature especially in terms of efficiency. Overall, for all the types of outliers the new proposed robust PLS-ARWMCD method with more or less differences gives better results than robust PRM method in terms

of efficiency, fitting to data and predictive ability. Furthermore, for all types of outliers but especially when the data contain bad leverage points or concentrated outliers, the new proposed robust PLS-ARWMCD method outperforms robust PLS-SD method in terms of efficiency, fitting to data and predictive ability. The mean angle values between the estimated slope and the true slope for the PLS-AWMCD method is also lower than the classical method (as expected) and all the other four robust PLSR methods for all types of outliers with the exception of vertical outliers. Because when the data contain vertical outliers RSIMPLS gives somewhat lower mean(angle) value than the PLS-ARWMCD method.

Table 2 shows that for all the types of outliers with the exception of vertical outliers when the proportion of outliers increases, it is seen that the performance of robust PRM method decreases especially in terms of efficiency and predictive ability, moreover, the mean angle values between the estimated slope and the true slope for this robust method is also higher than the other four robust PLSR methods (including the new proposed robust PLS-ARWMCD method). Especially when the proportion of concentrated outliers or orthogonal outliers is 20% in the data set, PRM method performs worse even than classical PLSR method in terms of MSE, GOF, RMSE and mean(angle) criterions. Furthermore, when there is 20% proportion of good leverage points PRM performs worse than classical PLSR method in terms of efficiency.

It is clear that when there is 20% proportion of bad leverage points or vertical outliers in the data set, the new proposed robust PLS-ARWMCD method, robust PLS-KurSD and RSIMPLS methods existing in the literature are the three forefront methods in terms of efficiency and predictive ability. Moreover, the mean angle values between the estimated slope and the true slope of these three robust methods are also lower than the robust PRM and PLS-SD methods for these two types of outliers. The concentrated outliers are the hardest type of outliers to cope with. It is seen that when there is 20% proportion of bad leverage points or concentrated outliers in the data set, the new proposed robust PLS-ARWMCD method performs better than both robust PLS-SD and PRM methods existing in the literature in terms of efficiency, fitting to data and predictive ability. Furthermore, PLS-ARWMCD method's mean angle values are also lower than these two robust methods for these two types of outliers. It could be mentioned that when the proportion of outliers in the data set gets a high-level as 20%, the new proposed robust PLS-ARWMCD method still gives better results than classical PLSR method for all the types of outliers in terms of efficiency, fitting to data and predictive ability.

Overall, both of from Table 1 and Table 2, it could be concluded that the new proposed robust PLS-ARWMCD method outperforms especially its two robust competitors (PRM and PLS-SD) existing in the literature with more or less differences in terms of efficiency, fitting to data and predictive ability for five different types of outliers.

Table 2: The sample size is $n = 200$, $p = 5$ and $k = 2$, the proportion of outliers is 20%.

	PLSR	RSIMPLS	PRM	PLS- SD	PLS-KurSD	**PLS-ARWMCD**
No Contamination						
MSE	0.0092	0.0111	0.0101	0.0104	0.01	0.0105
GOF	0.8312	0.8308	0.8308	0.8308	0.8309	0.8308
RMSE	1.0961	1.0974	1.0969	1.0973	1.0968	1.0974
Mean(angle)	0.0446	0.0519	0.0462	0.0493	0.0477	0.0491
Bad Leverage Points						
MSE	1.8946	0.0122	1.7726	0.4134	0.0121	0.0109
GOF	0.1858	0.8309	0.2395	0.7143	0.831	0.8313
RMSE	2.4002	1.1012	2.3205	1.4282	1.1011	1.0998
Mean(angle)	1.3018	0.0537	1.1833	0.2467	0.054	0.05
Vertical Outliers						
MSE	0.0791	0.0115	0.0174	0.0176	0.0126	0.0112
GOF	0.8057	0.8278	0.8265	0.8267	0.8282	0.8286
RMSE	1.1681	1.1002	1.106	1.1063	1.1003	1.0989
Mean(angle)	0.1437	0.0471	0.0632	0.0656	0.054	0.0503
Good Leverage Points						
MSE	0.9975	0.0128	1.0568	0.044	0.0121	0.0109
GOF	0.6741	0.831	0.6817	0.8282	0.831	0.8313
RMSE	1.5213	1.1011	1.5049	1.1102	1.1011	1.0998
Mean(angle)	0.7739	0.057	0.7813	0.1165	0.0539	0.05
Concentrated Outliers						
MSE	1.8527	0.0128	1.926	0.1628	0.0121	0.0109
GOF	0.4929	0.831	0.485	0.8107	0.831	0.8313
RMSE	1.8946	1.1012	1.9104	1.1648	1.1011	1.0998
Mean(angle)	1.1091	0.0569	1.1119	0.2307	0.0539	0.05
Orthogonal Outliers						
MSE	0.1987	0.0176	0.2332	0.0108	0.0115	0.0104
GOF	0.7806	0.831	0.7718	0.8319	0.8319	0.8322
RMSE	1.2488	1.1026	1.2739	1.1007	1.101	1.0999
Mean(angle)	0.2982	0.066	0.3247	0.0504	0.0519	0.0491

5. APPLICATION TO FISH DATA

In this section, the new proposed robust PLSR method and four robust PLSR methods, existing in the literature, will be compared on a real data including outliers in terms of goodness-of-fit and predictive ability by using (4.3) and (4.4). For this purpose, the fish data which was given in Naes (1985) will be used. The fish data comprise 45 observations and the last 7 are outliers (in the words of Naes, 'abnormal samples'). In this example, fat concentration (percentage, %) of 45 fish samples (rainbow trout) and independent variables of the absorbance at 9 Near Infrared Reflectance (NIR) wavelengths measured after sample homogenisation. The aim of the analysis made on this data set is to model the relationships between the fat concentration (one response variable) and these nine spectrums (independent variables). In this study, the data set is divided into two parts.

The first 20 observations are the test set and the other remained 25 samples are the training set (Gil and Romera, 1998; Hardy *et al.*, 1996; Naes, 1985).

Firstly, similar to the our simulation studies, while computing the GOF values 7 outliers are removed from training set that occurs of 25 samples. However, while computing the RMSE values the models are constituted using the training set including the 7 outliers. Then, by using the regression coefficients obtained from these models, the predictions are made from clean test set that occurs of 20 samples. Hence, the predictive ability of the new robust PLSR method 'PLS-ARWMCD' is examined especially against the classical PLSR method and the other four robust methods.

The GOF or RMSE values could be considered while selecting the number components that will be retained in the model. The optimal number of components could be selected as the k for which the GOF values are no more change. However, as it is mentioned before in Engelen *et al.* (2004), it is more convenient to consider the RMSE values while selecting the optimal number of components. The significant point while selecting the optimal number of components retaining in the model is that adding one more component whether cause an important decrease or not in RMSE value. Hence, both the aim of data reduction is not deviated and an unnecessary component is not added to model. In Figure 2, the figure of RMSE values against the number of components in the model is drawn.

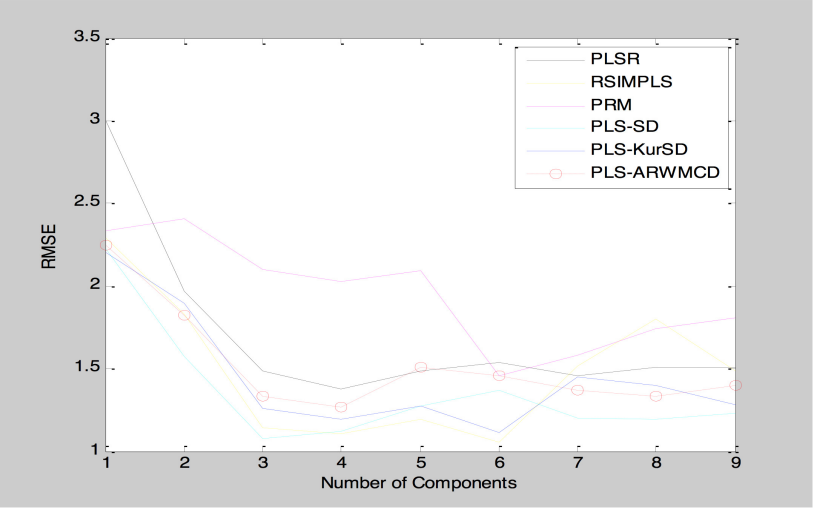

Figure 2: The RMSE values against the number of components in the model for fish data with the training set of 25 samples and the test set of 20 samples.

When Figure 2 is examined, it is seen that it is right to select the number of components retaining in the model as three for this data set. Because from the figure it is seen that adding the third component to the model causes a

significant decrease in the RMSE values of all the methods. It could be seen also much clearly from Table 3 that the optimal number of components should be selected as $k_{opt} = 3$, as adding the third component to the model cause an important decrease in the RMSE values for all the robust methods. Furthermore, it is clear that the fitting to data also improves for all the methods after adding the third component to the model. Table 3 shows that the new proposed robust PLS-ARWMCD method fitting to the data better and it has a higher predictive ability than both classical PLSR method and robust PRM method for $k_{opt} = 3$.

Table 3: The GOF and RMSE values for fish data in case of the first 20 observations are the test set and the other 25 samples are the training set.

Number of Components		**PLSR**	**RSIMPLS**	**PRM**	**PLS-SD**	**PLS-KurSD**	**PLS-ARWMCD**
k=1	GOF	0.2912	0.4335	0.3777	0.4417	0.4444	0.4397
	RMSE	3.0001	2.2937	2.3307	2.2274	2.2029	2.2487
k=2	GOF	0.6927	0.7421	0.2713	0.7853	0.6948	0.7605
	RMSE	1.9715	1.8293	2.4072	1.573	1.8935	1.8234
k=3	GOF	0.882	0.9687	0.6166	0.9665	0.9594	0.9579
	RMSE	1.4861	1.1401	2.0993	1.0797	1.259	1.3322
k=4	GOF	0.8987	0.971	0.6277	0.9737	0.9447	0.9662
	RMSE	1.3742	1.1089	2.0237	1.1198	1.1924	1.2646
k=5	GOF	0.9113	0.979	0.6782	0.9716	0.9777	0.9713
	RMSE	1.4874	1.1918	2.0921	1.2705	1.2708	1.5054
k=6	GOF	0.9231	0.9825	0.7705	0.9796	0.9854	0.9816
	RMSE	1.5348	1.0543	1.4578	1.3727	1.1129	1.4545
k=7	GOF	0.9299	0.9829	0.7806	0.9714	0.9865	0.9862
	RMSE	1.4553	1.519	1.5835	1.2033	1.4528	1.3679
k=8	GOF	0.9463	0.9768	0.8063	0.9769	0.9868	0.9861
	RMSE	1.5056	1.7989	1.7409	1.1925	1.4019	1.33
k=9	GOF	0.9463	0.9851	0.8087	0.9812	0.9798	0.987
	RMSE	1.5052	1.4874	1.8095	1.2338	1.2843	1.399

6. CONCLUSIONS

In this study, we propose a new robust PLSR method for the linear regression model with one response variable, PLS-ARWMCD, in order to obtain robust predictions in case of outliers present in the data set.

In the simulation study, the new proposed robust PLSR method is compared with classical PLSR method and four robust PLSR methods existing in the literature in terms of efficiency, fitting to data and predictive ability on a clean data set and on contaminated data sets with bad leverage points, vertical outliers, good leverage points, concentrated outliers or orthogonal outliers.

The optimal number of components is selected as $k = 2$ at the beginning of the simulation study. 10% and 20% proportions of this data set are replaced by outliers, respectively. Thus, the increment in the proportion of outliers how affects on performances of the new proposed robust PLSR method and four robust PLSR methods (existing in the literature) is examined. When the 10% proportion of the data set is contaminated by different types of outliers, both the new proposed robust PLS-ARWMCD method and the four robust PLSR methods existing in the literature outperform classical PLSR method in terms of efficiency and predictive ability (exception of PRM method that performs not better than classical PLSR method in terms of efficiency in case of good leverage points existence). The PLS-ARWMCD method comes to the forefront as a good alternative method against robust PRM and PLS-SD methods in terms of efficiency, fitting to data and predictive ability for all the types of outliers. Moreover, PLS-ARWMCD method shows a close performance with robust RSIMPLS and PLS-KurSD methods in terms of efficiency, fitting to data, predictive ability and mean angle measures. When the proportion of outliers in the data set is reached to a high level as 20%, robust PRM method shows a lower performance than other robust methods in terms of efficiency, fitting to data and predictive ability for all the types of outliers except that vertical outliers. Furthermore, if there is 20% proportion of concentrated outliers or orthogonal outliers in the data set, robust PRM method looses its performance completely against classical PLSR method. When there is high proportion of bad leverage points or concentrated outliers in the data set, robust PLS-SD method is less efficient and it has a lower predictive ability than the other robust RSIMPLS, PLS-KurSD methods and new proposed robust PLS-ARWMCD method.

The results obtained from real data analysis show that the optimal number of components is selected as $k_{opt} = 3$, as adding the third component to the model causes a considerably decrease in the RMSE values of robust methods. It is clear from the results of the model containing $k = 3$ components that GOF values of the new proposed robust PLS-ARWMCD method are higher than both classical PLSR method and robust PRM method. Moreover, when $k_{opt} = 3$ is selected, the RMSE value for PLS-ARWMCD is lower than both classical PLSR method and robust PRM method. Generally, whatever the optimal number of the components in the model for the fish data set, the new proposed robust PLS-ARWMCD method gives better models than both classical PLSR method and robust PRM method in terms of fitting to data and predictive ability.

Consequently, it is seen that the new proposed robust PLS-ARWMCD method gives more efficient results than especially classical PLSR method in data sets contaminated by a reasonable amount of outliers. The simulation study shows that when the data contain 10% or 20% proportion of bad leverage points, the new robust PLS-ARWMCD method outperforms both of the robust PRM and PLS-SD methods in terms of efficiency and predictive ability.

When the data contain 10% proportion of vertical outliers, the new robust PLS-ARWMCD method shows a close performance to the other four robust PLSR methods existing in literature. However, when there is 20% proportion of vertical outliers in the data set; the new robust PLS-ARWMCD method, robust RSIMPLS and PLS-KurSD methods are the forefront methods in terms of efficiency and predictive ability. When the data contain 10% or 20% proportion of good leverage points; the new robust PLS-ARWMCD method has a better performance than robust PRM method both in terms of efficiency and predictive ability, however, it is only more efficient than robust PLS-SD method. When there is 10% proportion of concentrated outliers; the new robust PLS-ARWMCD method is both more efficient and it has a higher predictive ability than robust PRM method, however, it is only more efficient than robust PLS-SD method. When there is 20% proportion of concentrated outliers in the data set, the new robust PLS-ARWMCD method is both more efficient and it has a higher predictive ability than both robust PRM and PLS-SD methods. When the data contain 10% or 20% proportion of orthogonal outliers; the new robust PLS-ARWMCD method has a better performance than robust PRM method in terms of efficiency, fitting to data and predictive ability. Overall, it could be concluded that the new proposed robust PLS-ARWMCD could cope with different types and proportions of outliers efficiently and it give robust predictions.

REFERENCES

[1] Engelen S.; Hubert, M.; Vanden Branden, K. and Verboven, S. (2004). *Robust PCR and Robust PLSR: A comparative study.* In "Theory and Applications of Recent Robust Methods" (M. Hubert, G. Pison, A. Struyf and S.V. Aelst, Eds.), Birkhäuser, Basel, 105–117.

[2] Filzmoser, P.; Serneels, S.; Maronna, R. and Van Espen, P.J. (2009). *Robust multivariate methods in chemometrics.* In "Comprehensive Chemometrics" (B. Walczak, R.T. Ferre and S. Brown, Eds.), 681–722.

[3] Gervini, D. (2003). A robust and efficient adaptive reweighted estimator of multivariate location and scatter, *Journal of Multivariate Analysis*, **84**, 116–144.

[4] Gil, J.A. and Romera, R. (1998). On robust partial least squares (PLS) methods, *Journal of Chemometrics*, **12**, 365–378.

[5] González, J.; Peña, D. and Romera, R. (2009). A robust partial least squares regression method with applications, *Journal of Chemometrics*, **23**, 78–90.

[6] Hardy, A.J.; Maclaurin, P.; Haswell, S.J.; De Jong, S. and Vandeginste, B.G.M. (1996). Double-case diagnostic for outliers identification, *Chemometrics and Intelligent Laboratory Systems*, **34**, 117–129.

[7] Hubert M. and Vanden Branden, K. (2003). Robust methods for Partial Least Squares Regression, *Journal of Chemometrics*, **17**, 537–549.

[8] Liebmann, B.; Filzmoser, P. and Varmuza, K. (2010). Robust and Classical PLS Regression Compared, *Journal of Chemometrics*, **24**(3–4), 111–120.

[9] Moller, S.F.; Von Frese, J. and Bro, R. (2005). Robust Methods for Multivariate Data Analysis, *Journal of Chemometrics*, **19**(10), 549–563.

[10] Naes, T. (1985). Multivariate calibration when the error covariance matrix is structured, *Technometrics*, **27**(3), 301–311.

[11] Polat, E. (2014). *New Approaches in Robust Partial Least Squares Regression Analysis*, Ph.D. Turkish diss., Hacettepe University Department of Statistics, Ankara, Turkey.

[12] Rousseeuw, P.J. (1984). Least median of squares regression, *J. Amer. Statist. Assoc.*, **79**, 871–880.

[13] Rousseeuw, P.J. and Van Zomeren, B.C. (1990). Unmasking multivariate outliers and leverage points, *J. Amer. Statist. Assoc.*, **85**, 633–639.

[14] Rousseeuw, P.J. and Van Driessen, K. (1999). A fast algorithm for the minimum covariance determinant estimator, *Technometrics*, **41**, 212–224.

[15] Serneels, S.; Croux, C.; Filzmoser, P. and Van Espen, P.J. (2005). Partial Robust M-regression, *Chemometrics and Intelligent Laboratory Systems*, **79**, 55–64.

[16] Verboven, S. and Hubert, M. (2005). LIBRA: a MATLAB library for robust analysis, *Chemometrics and Intelligent Laboratory Systems*, **75**, 127–136.

[17] Wakeling, I.N. and Macfie, H.J.H. (1992). A Robust PLS Procedure, *Journal of Chemometrics*, **6**, 189–198.