

Fast Model Selection for Robust Calibration Methods

S. Engelen, M. Hubert

Revised version ACA04-1476Rev

Abstract: One of the main issues in Principal Component Regression (PCR) and Partial Least Squares Regression (PLSR) is the selection of the number of principal components. To this end, the curve with the root mean squared error of cross-validated prediction (RMSECV) is often described in the literature as a very helpful graphical tool. In this paper, we focus on model selection for robust calibration methods. We first propose a robust RMSECV value and then use it to define a new criterion for the selecting of the optimal number of components. This robust component selection (RCS) statistic combines the goodness-of-fit and the predictive power of the model. As the algorithms to compute these robust PCR and PLSR estimators are more complex and slower than the classical approaches, cross-validation becomes very time consuming. Hence, we propose fast algorithms to compute the robust RMSECV values. We evaluate the developed procedures at several data sets.

1 Introduction

Many calibration methods such as Principal Component Regression (PCR) and Partial Least Squares Regression (PLSR) require the determination of the optimal number of components, which we will denote by k_{opt} . If k_{opt} is chosen too small, too much information contained in the data is lost and consequently the regression model will not fit the data very well. Assigning a too large value to k_{opt} leads to overfitting, which implies that the model will fit the calibration data very closely, but it will not be very accurate in predicting the response value of new samples.

Many parametric techniques already exist to determine this optimal number of regressors, among which the Akaike's Information Criterion, the R_p^2 -criterion, Mallows's C_p criterion (see e.g. [1] and [2]). In all those methods some function concerning the fit of the model to the calibration data has to be optimized and k_{opt} is then taken as the value for which the optimum is reached. Also non-parametric methods are available, see e.g. [3], [4] and [5].

Another class of component selection methods evaluates the predictive ability of the model such as the adjusted Wold's R criterion [6],[7], the Osten's F-criterion [7] and criteria based on the Root Mean Squared Error for Prediction (see e.g. [2] and [8]). The optimal number of components to include in the final model is that k for which the predictive criterion is small enough.

For this second group of criteria, typically a validation set is necessary. However such a validation set is not always available and mainly poses a problem for small data sets that occur frequently in the domain of chemometrics and bio-informatics. Therefore, one often relies on cross-validation (CV), where the model is fitted on a large part of the data and evaluated on the remaining observations. In particular we will here focus on one-fold cross-validation where the calibration set consists of all but one data point. This is a rather time-consuming approach, but several studies have shown its appropriateness for model selection (see eg. [9]) and fast algorithms have been proposed for its computation [10].

In this paper we consider the problem of model selection for robust PCR (RPCR) [11] and robust PLSR (RSIMPLS) [12] methods. As the algorithms to compute these estimators are more complex and hence slower than the classical approaches, it becomes very important to take care of the computational aspects of one-fold CV. Therefore we have constructed fast algorithms to compute the root mean squared error of cross-validated prediction (RMSECV) for several robust calibration techniques. Moreover, we propose a new criterion to select k_{opt} that combines the goodness-of-fit and the predictive power of the model.

In Section 2 we start by defining this robust component selection (RCS) method. In Section 3 we explain how fast cross-validation algorithms can be constructed for the robust PCR method introduced in [11]. For this, we distinguish between univariate and multivariate response variables, for which the LTS regression [13] respectively the MCD-regression [14] is used. Cross-validation for the robust RSIMPLS method [12] is outlined in Section 4. Software availability and conclusions are described in Section 5.

2 The RCS criterion

We consider the calibration problem, in which we have p independent variables or regressors (X_1, \dots, X_p) and q dependent or response variables (Y_1, \dots, Y_q) . The number of observations is indicated by n . The linear regression model states that :

$$\mathbf{y}_i = \mathbf{b}_0 + \mathbf{B}'_{q,p} \mathbf{x}_i + \mathbf{e}_i \quad \forall i = 1, \dots, n \quad (1)$$

where the error terms \mathbf{e}_i satisfy $E(\mathbf{e}_i) = \mathbf{0}$ and $\text{Cov}(\mathbf{e}_i) = \mathbf{\Sigma}_e$. The q -dimensional intercept is denoted as $\mathbf{b}_0 = (b_{01}, \dots, b_{0q})'$ and the $(p \times q)$ slope matrix as $\mathbf{B}_{p,q}$ or just \mathbf{B} to simplify the notations. We consider a vector as a one-dimensional column matrix, and use bold symbols for vectors and matrices.

PCR and PLSR start by constructing k -dimensional scores $\mathbf{t}_{i,k}$ for each sample i which should well summarize the x -variables, and then regress the y -variables onto these scores. Consequently, the estimates $\hat{\mathbf{b}}_{0,k}$, $\hat{\mathbf{B}}_k$ and $\hat{\mathbf{\Sigma}}_{e,k}$ depend on the choice of k , and even so the residuals $\mathbf{r}_{i,k} = \mathbf{y}_i - (\hat{\mathbf{b}}_{0,k} + \hat{\mathbf{B}}'_k \mathbf{x}_i)$. If the i th sample is first removed from the calibration set to determine the regression estimates, we denote $\hat{\mathbf{y}}_{-i,k}$ as the resulting fitted value for the i th case and $\mathbf{r}_{-i,k} = \mathbf{y}_i - \hat{\mathbf{y}}_{-i,k}$ as the i th cross-validated residual. In univariate regression, we denote the variance estimate of the residuals by $\hat{\sigma}_k^2$.

A robust RMSECV statistic is already proposed in [11] and [12] by adding weights w_i to the classical RMSECV value. For a model with k components and one response variable ($q = 1$), the R-RMSECV value becomes:

$$\begin{aligned} \text{R-RMSECV}_k &= \sqrt{\frac{1}{w} \sum_{i=1}^n w_i (y_i - \hat{y}_{-i,k})^2} \\ &= \sqrt{\frac{1}{w} \sum_{i=1}^n w_i r_{-i,k}^2}. \end{aligned} \quad (2)$$

with $w = \sum_{i=1}^n w_i$. When we deal with more than one dependent variable ($q > 1$), we take the root of the average of the squared R-RMSECV $_k$ values for every y -variable, which leads to:

$$\begin{aligned} \text{R-RMSECV}_k &= \sqrt{\frac{1}{qw} \sum_{j=1}^q \sum_{i=1}^n w_i (y_{ij} - \hat{y}_{-i,j,k})^2} \\ &= \sqrt{\frac{1}{qw} \sum_{i=1}^n w_i \|\mathbf{r}_{-i,k}\|^2}. \end{aligned} \quad (3)$$

Here, $y_{ij,k}$ and $\hat{y}_{-ij,k}$ are the j th components of \mathbf{y}_i and $\hat{\mathbf{y}}_{-i,k}$.

The weights should be zero (or small) for outliers, as we do not want them to influence the RMSECV value. A possible way to define the weights is as follows. First, we compute for each observation its residual distance (ResD)

$$\text{ResD}_{i,k} = \sqrt{\mathbf{r}'_{i,k} \hat{\Sigma}_{\mathbf{e},k}^{-1} \mathbf{r}_{i,k}} \quad (4)$$

which just coincides with its absolute standardized residual $|r_{i,k}/\hat{\sigma}_k|$ if $q = 1$. If a sample has a residual distance larger than $\sqrt{\chi_{q,0.975}^2}$, it receives a weight $w_{i,k} = 0$ and else $w_{i,k} = 1$. This threshold is conducted from the assumptions that if the residuals are multivariate normally distributed, the residual distances asymptotically follow a χ_q^2 -distribution.

The weight $w_{i,k}$ then still depends on the number of components k in the model. In order to make a fair comparison among the models with $k = 1$ up to $k = k_{max}$ components, we can finally construct a global weight by taking the minimum over all possible values for k :

$$w_i = w_{i,\min} = \min_k(w_{i,k}). \quad (5)$$

Doing so, an observation receives zero weight if it is outlying in at least one of the models under consideration. Other alternatives can be considered for defining the weights, like e.g. taking the median over all k instead of the minimum. A smoother version can for example be defined by setting:

$$w_{i,k,smooth} = \begin{cases} \frac{1}{\text{ResD}_{i,k}^2} & \text{if } \text{ResD}_{i,k} > \sqrt{\chi_{q,0.975}^2} \\ 1 & \text{if } \text{ResD}_{i,k} \leq \sqrt{\chi_{q,0.975}^2} \end{cases} \quad (6)$$

with global weight $w_{i,smooth} = \text{mean}_k(w_{i,k,smooth})$.

We have experimented with several weight definitions, but as our conclusions did not vary much, we will here only present the results for the minimum weights (5). Note that the computation of the $w_{i,k}$ requires to perform the RPCR or the RSIMPLS method for each k under study, which is rather time-consuming and often unnecessary as we only want the regression results at the optimal k . A faster procedure to compute the weights $w_{i,k}$ will therefore be proposed in Section 3.3.

The R-RMSECV $_k$ statistic measures the predictive ability of the model with k components. If we are mainly interested in evaluating how well this model fits the calibration data, we

can similarly define a robust root mean squared error (RMSE) statistic by adding weights to the residuals (of the calibration set). This yields for $q = 1$

$$\text{R-RMSE}_k = \sqrt{\frac{1}{\sum_{i=1}^n w_i} \sum_{i=1}^n w_i r_{i,k}^2}$$

and for $q > 1$

$$\text{R-RMSE}_k = \sqrt{\frac{1}{q \sum_{i=1}^n w_i} \sum_{i=1}^n \sum_{j=1}^q w_i r_{ij,k}^2}$$

Finally, we propose the robust component selection (RCS) statistic by combining the R-RMSECV and the R-RMSE:

$$\text{RCS}_k = \sqrt{\gamma \text{R-RMSECV}_k^2 + (1 - \gamma) \text{R-RMSE}_k^2}$$

with γ a tuning parameter between 0 and 1. Including this extra parameter has the advantage that the user can decide whether the model mainly needs to be strong in prediction (i.e. $\gamma > 0.5$) or whether the goodness-of-fit is a primary interest (i.e. $\gamma < 0.5$). Putting $\gamma = 0.5$ corresponds with a selection criterion where the fit of the actual data and the prediction of new samples are equally important.

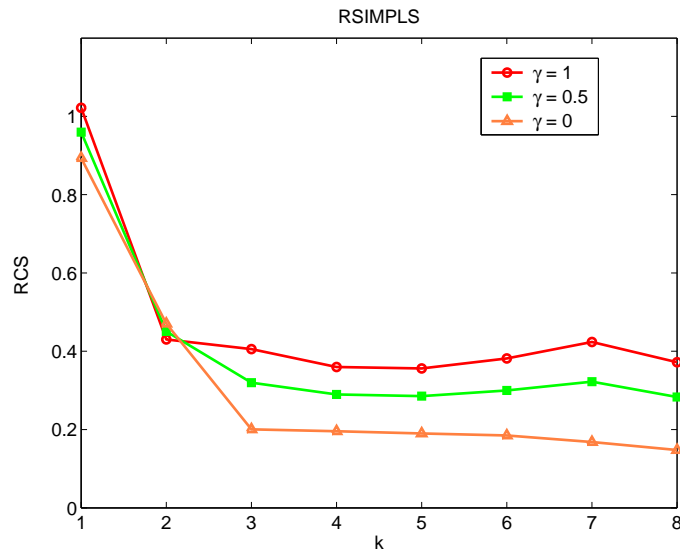


Figure 1: The RCS plot for the beer data for RSIMPLS.

We illustrate the effect of γ on the *beer*-data [15] which consist of 40 NIR spectra measured over 926 wavelengths. The response variable reflects a quality measure of beer.

In Figure 1 the RCS-curves for $\gamma = 0$, $\gamma = 0.5$ and $\gamma = 1$ are plotted, with the parameter estimates obtained with RSIMPLS. When only the predictive power of the model is important ($\gamma = 1$), the upper curve (with circles) suggests to select either two components as this curve already stabilizes at $k = 2$, or to take $k = 5$ where the minimum is reached. On the other hand, when only the fit to the calibration data is taken into account ($\gamma = 0$), the lower curve (with triangles) clearly shows that the fit improves a lot by selecting at least $k = 3$ components. The curve in the middle (with boxes) interpolates between the two other curves and leads to the decision that $k_{opt} = 3$. Plotting the RCS curves for several values of γ thus yields more insight into the model selection procedure, and allows to make a decision based on a combination of goodness-of-fit and predictive ability.

To compute the RCS values, we need the R-RMSECV values which are highly time-consuming for robust methods. In the following sections, we will show how cross-validation can still be performed in a reasonable time span. We start with the RPCR method and then apply the same techniques to do fast CV for the RSIMPLS method.

3 The RPCR method

3.1 Naive cross-validation

Principal component regression [16] is a two-step procedure where in the first step $k < p$ dimensional scores $\mathbf{t}_{i,k}$ are constructed by performing PCA on the x -data:

$$\mathbf{t}_{i,k} = \mathbf{P}'_{k,p}(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_x) \quad (7)$$

with $\mathbf{P}_{p,k}$ the $(p \times k)$ loading matrix and $\hat{\boldsymbol{\mu}}_x$ the mean of the x -variables. In a second step the y -variables are regressed onto the k scores in the linear model:

$$\mathbf{y}_i = \mathbf{a}_{0,k} + \mathbf{A}'_{q,k}\mathbf{t}_{i,k} + \mathbf{f}_i \quad (8)$$

where $\mathbf{a}_{0,k}$ is the q -dimensional intercept, $\mathbf{A}_{k,q}$ the $(k \times q)$ regression matrix and \mathbf{f}_i the error terms with $\text{Cov}(\mathbf{f}_i) = \boldsymbol{\Sigma}_{\mathbf{f}}$. In classical PCR, estimates of the regression parameters in (8) are obtained through multivariate least squares regression (often denoted by MLR). Finally, the regression coefficients of the original model (1) can be computed from $\hat{\mathbf{a}}_{0,k}$, $\hat{\mathbf{A}}_{k,q}$

by means of the transformation formulas

$$\hat{\mathbf{B}}_k = \mathbf{P}_{p,k} \hat{\mathbf{A}}_{k,q} \quad \text{and} \quad \hat{\mathbf{b}}_{0,k} = \hat{\mathbf{a}}_{0,k} - \hat{\mathbf{B}}_k' \hat{\boldsymbol{\mu}}_x. \quad (9)$$

A robust version of this two-step method is described in [11], where each stage of the algorithm is replaced by a robust statistical method. This is done in the following way:

1. First, a robust PCA method for high-dimensional data, ROBPCA [17], is applied to the x -variables, yielding k robust scores $\mathbf{t}_{i,k} = (t_{i,1}, \dots, t_{i,k})$. ROBPCA combines projection pursuit ideas with the estimate of the robust MCD covariance matrix. For all details about its computation, we refer to [17]. It is important to note that the principal components found with the ROBPCA algorithm depend on the selected number of principal components. If for example we apply ROBPCA with $k = 1$, we get one component $t_{i,1}$ for each observation i . When we apply ROBPCA with $k = 2$, we find two components $(\tilde{t}_{i,1}, \tilde{t}_{i,2})$ and $\tilde{t}_{i,1}$ will in general not be equal to $t_{i,1}$.
2. To obtain robust regression estimates in model (8), the LTS-regression method [13] is used when the response variable is univariate ($q = 1$) and MCD-regression [14] when $q > 1$. These regression methods are fully described in the appendix. Note that, as the components $\mathbf{t}_{i,k}$ depend on k (see previous remark), also the explanatory variables in (8) change with k .

When cross-validation would be naively applied, we need to run the RPCR method $n \times k_{max}$ times. Table 1 gives an overview of this procedure. It starts by removing the i th case, and then applies for each $k = 1, \dots, k_{max}$ ROBPCA on all remaining $n - 1$ observations. For each value of k and for each observation $m \neq i$, this yields the k components $\mathbf{t}_{m,k}^{-i}$. They are then used in the regression model (8) to obtain regression estimates $\hat{\mathbf{a}}_{0,k}^{-i}$ and $\hat{\mathbf{A}}_{k,q}^{-i}$. After transforming these to $hat{\mathbf{b}}_{0,k}^{-i}$ and $\hat{\mathbf{B}}_k^{-i}$ using formulas (9), we can finally compute the cross-validated fitted value $\hat{\mathbf{y}}_{-i,k}$ and the cross-validated residual $\mathbf{r}_{-i,k}$.

[Table 1 here]

This method is not really feasible, because all the robust procedures that are considered here are based on random subsampling and consequently take much more time to compute than their classical counterparts. All the robust methods have a non-convex objective function,

and hence, their computation can not be found using standard optimization techniques. In the appendix we have described the estimators and their algorithms in detail. For clarity of the main text, we here only summarize some common features of the algorithms:

1. The objective function of the methods is defined in terms of an optimal subset of size h . For example, the LTS-regression method seeks the h -subset for which the sum of the squared residuals is minimal. The MCD estimator of location and scatter looks for the h -subset with minimal covariance determinant. The number h should be larger than half of the data size $n/2$, and $n - h$ should be larger than the actual number of outliers.
2. To avoid scanning all possible h -subsets, many h -subsets are considered, and their objective function evaluated. To find initial h -subsets, many (at most 500) subsets of very small size are randomly drawn and enlarged using C-steps (see step 3).
3. Using a C-step, an h -subset can be easily modified into another h -subset which has a lower objective function, and hence is closer to the optimal solution. The definition of a C-step depends on the method (see the appendix for details).
4. The subset found by the algorithm with the lowest objective function is called the optimal h -subset H_{opt} and the estimates based on the optimal h -subset are called the raw estimates. A reweighting step can finally be applied to increase the finite-sample efficiency. In regression, this reweighting step assigns zero weight to outliers found with the raw estimates, and then applies the efficient least squares method on the non-outliers.

It may be clear from this general outline that the main computational efforts for robust methods lie in the resampling steps. We will try to avoid these as much as possible in our approximated cross-validation algorithms. A second important time improvement is obtained by performing the computations mainly for k_{max} instead of considering all $k = 1, \dots, k_{max}$ components separately.

3.2 Fast cross-validation

To obtain a fast CV algorithm, we follow the scheme shown in Table 2.

[Table 2 here]

The cross-validated residual for the i th case is obtained through the following steps:

1. First we fix $k = k_{max}$.
2. For every sample i , we remove it from the data set.
3. We apply the fast CV algorithm for ROBPCA (fully described in [18]) on the x -variables. This yields k_{max} -dimensional scores for all remaining $n - 1$ observations, denoted by $\mathbf{t}_{m,k_{max}}^{-i}$ for all $m \neq i$.
4. Next, we perform (raw) LTS/MCD-regression on the $(\mathbf{t}_{m,k_{max}}^{-i}, \mathbf{y}_i)$. However, instead of applying the full resampling algorithm, we only apply C-steps on a limited number of h -subsets. Doing so, we get $\hat{\mathbf{a}}_{0,k_{max}}^{-i}$ and $\hat{\mathbf{A}}_{k_{max},q}^{-i}$ for LTS-regression, and $\hat{\boldsymbol{\mu}}_{raw,k_{max}}$ and $\hat{\boldsymbol{\Sigma}}_{raw,k_{max}}$ for MCD-regression. In Section 3.3 we explain which h -subsets are selected.
5. Now, according to (8), we need the LTS/MCD-regression estimates in the linear model

$$\mathbf{y}_m = \mathbf{a}_{0,k}^{-i} + \mathbf{A}_{k,q}^{-i} \mathbf{t}_{m,k}^{-i} + \mathbf{f}_m$$

for every $k = 1, \dots, k_{max}$ and $m \neq i$. For this, we first take $\mathbf{t}_{m,k}^{-i}$ as the first k components of $\mathbf{t}_{m,k_{max}}^{-i}$. Then we derive approximate LTS/MCD-regression estimates. For LTS-regression, we start by using $\hat{\mathbf{a}}_{0,k_{max}}^{-i}$ and by taking the first k components of $\hat{\mathbf{A}}_{k_{max},q}^{-i}$ as initial estimates. Note that in classical least squares regression, this approach would yield the exact $\hat{\mathbf{A}}_{k,q}$ estimates because of the orthogonality of the scores matrix $T_{n,k_{max}}$. Then we apply C-steps until convergence and perform the reweighting as in the FAST-LTS algorithm. For MCD-regression, the details are explained in the appendix. The resulting estimates are denoted as $\hat{\mathbf{a}}_{0,k}^{-i}$ and $\hat{\mathbf{A}}_{k,q}^{-i}$.

6. After transforming $\hat{\mathbf{a}}_{0,k}^{-i}$ and $\hat{\mathbf{A}}_{k,q}^{-i}$ to $\hat{\mathbf{b}}_{0,k}^{-i}$ and $\hat{\mathbf{B}}_k^{-i}$, we can compute the cross-validated residual $\mathbf{r}_{-i,k}$.

3.3 Computation of the weights

Let us now describe how we can efficiently compute the weights $w_{i,k}$ that are needed to draw the R-RMSECV curve. Their computation is similar to the outline of Table 2, but now the

samples do not need to be removed successively. The structure of the procedure is depicted in Table 3.

[Table 3 here]

The following steps are performed:

1. First we fix $k = k_{max}$.
2. Then, we apply ROBPCA with $k = k_{max}$ on the whole data set. This yields k_{max} -dimensional scores $\mathbf{t}_{i,k_{max}}$ for each observation $i = 1, \dots, n$. Moreover, we retain three h -subsets that are very likely to contain only regular observations. With the notation of [18], these are the subsets H_0 , H_1 and H_{freq} . The subset H_0 is obtained from the projection pursuit stage in ROBPCA and contains the h observations with smallest outlyingness. The set H_1 is the optimal h -subset resulting from the computation of the MCD estimator in ROBPCA, whereas H_{freq} are the h most frequently selected observations in that resampling algorithm.
3. Next, we apply LTS/MCD-regression on the $(\mathbf{t}_{i,k_{max}}, \mathbf{y}_i)$. The raw LTS regression estimates are denoted by $\hat{\mathbf{a}}_{0,k_{max}}$ and $\hat{\mathbf{A}}_{k_{max},q}$, whereas applying the raw MCD on (\mathbf{t}, \mathbf{y}) yields $\hat{\boldsymbol{\mu}}_{raw,k_{max}}$ and $\hat{\boldsymbol{\Sigma}}_{raw,k_{max}}$. The resampling algorithm for LTS- or MCD-regression again yields H_{freq} as the h most frequently selected observations.
4. Now, we need the LTS/MCD-regression estimates in the linear model (8) for every $k = 1, \dots, k_{max}$. As before, we first take $\mathbf{t}_{i,k}$ as the first k components of $\mathbf{t}_{i,k_{max}}$. Then we derive approximate LTS/MCD-regression estimates. For LTS-regression, we start by using $\hat{\mathbf{a}}_{0,k_{max}}$ and by taking the first k components of $\hat{\mathbf{A}}_{k_{max},q}$ as initial estimates. Then we apply C-steps until convergence and perform the reweighting as in the FAST-LTS algorithm. For MCD-regression, the details are again explained in the appendix. The resulting estimates are denoted as $\hat{\mathbf{a}}_{0,k}$, $\hat{\mathbf{A}}_{k,q}$ and $\hat{\boldsymbol{\Sigma}}_{e,k}$. During the computations, we retain H_{opt} for each k .
5. After transforming the regression estimates to $\hat{\mathbf{b}}_{0,k}$ and $\hat{\mathbf{B}}_k$, we can compute the residual $\mathbf{r}_{i,k}$, the residual distance $\text{ResD}_{i,k}$ and the weight $w_{i,k}$.

We see that the computation of the weights, which in fact is done before the cross-validation starts, yields $k_{max} + 4$ subsets of size h . These are exactly the subsets which are used in the fast cross-validation to perform a fast LTS/MCD-regression (see Section 3.2, step 4). Instead of applying C-steps on 500 h -subsets, C-steps are only started from this limited number of h -subsets.

3.4 Selection of k_{max}

We notice that our approximate algorithm extensively makes use of the maximal number of components that is studied k_{max} . The larger k_{max} is taken, the more rough the approximation may become (see also [19]). Hence, we recommend to select a not too large value of k_{max} . A reasonable value can sometimes be known a priori. If not, we should keep in mind that any regression estimator will reach a very large variance if there are too few observations compared to their dimension. To avoid this curse of dimensionality, it is often recommended that the number of cases be larger than 5 times the number of variables [20]. Hence, as a general rule, we advice to take k_{max} such that $n/(k_{max} + q)$ is at least 5. In our programs, we also by default do not take k_{max} larger than 10, as in practical examples more than 10 components are rarely needed.

3.5 Examples

3.5.1 The Octane data

The octane data [21] set measures the octane number of 39 production gasoline samples which are registered by means of NIR absorbance spectra with a range of 226 wavelengths from 1102nm to 1552nm. The response variable corresponds with the octane number, so $q = 1$. It is well-known that this data set contains 6 outliers (cases 25, 26, 36, 37, 38 and 39), as alcohol has been added to them. Hence, to compute the R-RMSECV curves, we have set $w_i = 0$ for these six samples, and $w_i = 1$ for the others. Figure 2 shows the fast and the naive R-RMSECV curves. We have set $k_{max} = 6$ following our rule of Section 3.4. We see that both curves almost collapse.

Moreover, there is a large decrease in computation time. Whereas the naive cross-validation takes 1061 seconds, we only need 12 seconds to draw the fast R-RMSECV curve.

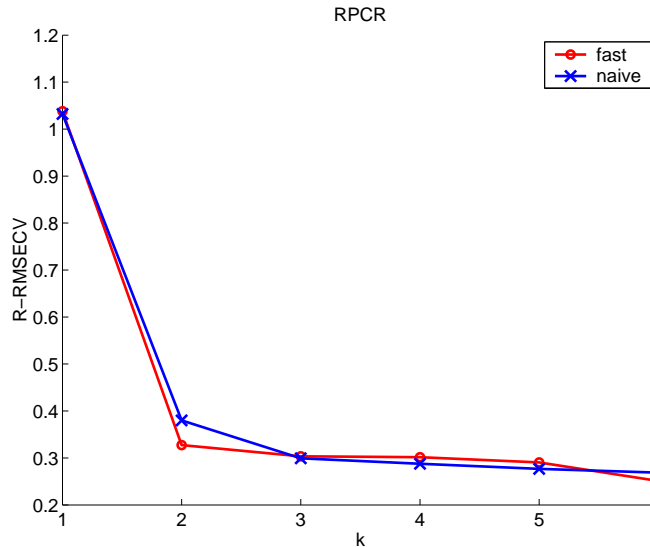


Figure 2: The R-RMSECV values for the octane data for RPCR.

3.5.2 The Egg data

This multivariate data set is kindly provided by Bart Kemps from the Faculty of Agricultural and Applied Biological Sciences of the Katholieke Universiteit Leuven [22]. The data set contains for 80 eggs NIR transmission spectra over $p = 753$ wavelengths from 507 nm to 933 nm. For each egg the haugh unit and the pH-value are measured, so $q = 2$. Also for this example the R-RMSECV curves in Figure 3 indicate that the fast methods are almost as accurate as the naive ones. A comparison of the computation time favors the fast procedure significantly (51 seconds versus 8045 seconds).

4 Robust Partial Least Squares Regression

4.1 The RSIMPLS algorithm

The RSIMPLS method is fully described in [12]. It is a robustification of the SIMPLS algorithm [23], hence it follows the main steps of SIMPLS.

Very shortly described, RSIMPLS starts by applying ROBPCA on the joint (x, y) variables, and then yields robust scores $\mathbf{t}_{i,k}$ by deflating the resulting cross-covariance matrix $\hat{\Sigma}_{xy}$.

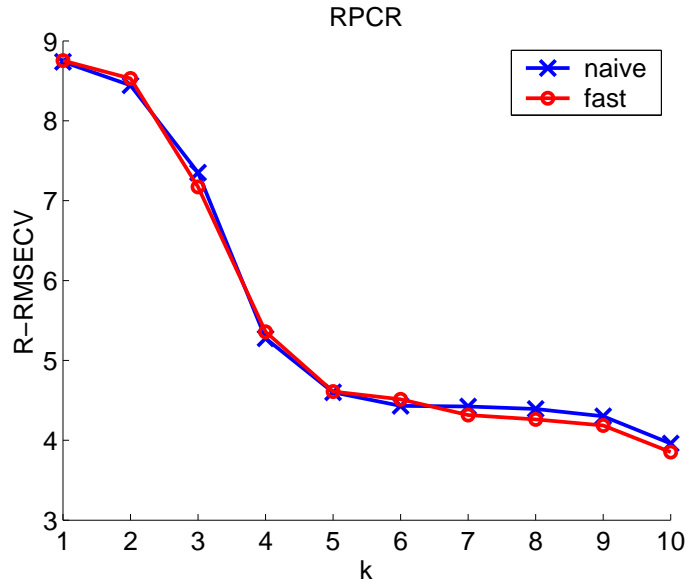


Figure 3: The R-RMSECV values for the egg data for RPCR.

Note that if k PLS components are requested, $k_0 = k + q$ eigenvectors are computed with ROBPCA.

To perform the regression of y on t , as in equation (8), a modified MCD-regression is applied (see the appendix for the details on MCD-regression). The so-called ROBPCA-regression also starts with an estimate of the center and scatter of the joint (t, y) variables. But instead of using the reweighted MCD-estimate, a weighted mean and covariance matrix is used, with weights derived from the application of ROBPCA on the (x, y) . The rest of the procedure is then similar to MCD-regression.

4.2 Fast cross-validation

To speed up cross-validation for RSIMPLS we can use similar ideas as used for RPCR. An overview is given in Table 4.

[Table 4 here]

1. First we fix $k = k_{max}$.
2. For every sample i , we remove it from the data set.

3. We apply the fast CV algorithm for ROBPCA with $k_0 = k_{max} + q$ on the joint (x, y) -variables. This yields scores $\mathbf{t}_{m, k_{max}}^{-i}$ and weights v_m^{-i} for all $m \neq i$.
4. Then we compute the weighted mean $\hat{\boldsymbol{\mu}}_{k_{max}}^{-1}$ and covariance matrix $\hat{\boldsymbol{\Sigma}}_{k_{max}}^{-i}$ of the $(\mathbf{t}_{m, k_{max}}^{-i}, \mathbf{y}^{-i})$ with weights v_m^{-i} .
5. For all $k = 1, \dots, k_{max}$ we derive $\hat{\boldsymbol{\mu}}_k^{-1}$ and $\hat{\boldsymbol{\Sigma}}_k^{-i}$ (as described in the appendix for MCD-regression). From these estimates, we compute $\hat{\mathbf{a}}_k^{-i}$ and $\hat{\mathbf{A}}_{k, q}^{-i}$ as in ROBPCA-regression.
6. Transforming $\hat{\mathbf{a}}_k^{-i}$ and $\hat{\mathbf{A}}_{k, q}^{-i}$ to $\hat{\mathbf{b}}_k^{-i}$ and $\hat{\mathbf{B}}_k^{-i}$ yields $\mathbf{r}_{-i, k}$.

The computation of the weights that are needed for the R-RMSECV curve are obtained in an equivalent way as described in Table 4, but without removing each observation at a time. The corresponding scheme is depicted in Table 5. As ROBPCA-regression is not based on resampling, we do not need to retain certain h -subsets as in Table 3.

[Table 5 here]

4.3 Examples

For the octane data set, we obtain the R-RMSECV curves of Figure 4, whereas the egg data set yields the curves exposed in Figure 5. In the latter plot we see some deviances between the two curves, but the R-RMSECV curve obtained with the fast algorithm is much smoother than the naive one. We thus have the impression that in this example the naive cross-validation yields a more variable curve than the fast CV. This situation can occur as all the resampling methods discussed in this paper in most cases only attain a local minimum. With the naive method, it is hence possible that the local minimum found at the full and at the reduced data set differ a lot. Our approximate method on the other hand looks for a solution which is close to the local minimum attained at the full data set.

As the fast R-RMSECV curve of the egg data is smooth (as we also observed in Figure 3 for RPCR), it is difficult to decide on the optimal k , solely based on this plot. Hence, we recommend to look at the RCS curve of Figure 6. From this plot we would decide to select 5, or eventually 6, components as it seems to offer the best balance between a sparse model

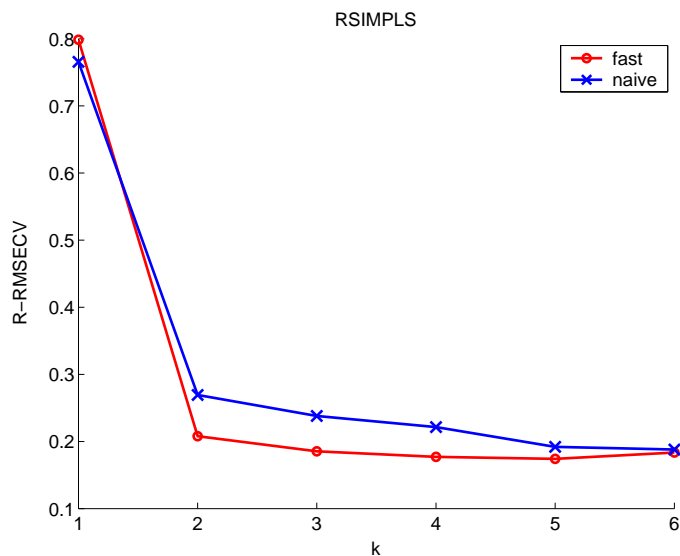


Figure 4: The R-RMSECV values of the octane data for RSIMPLS.

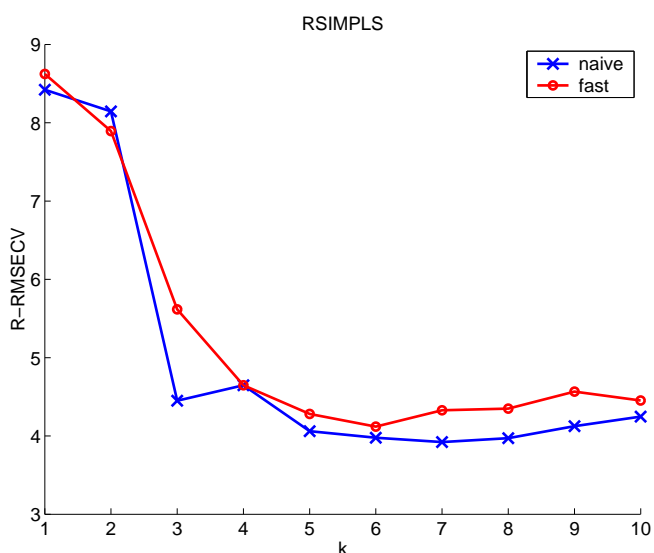


Figure 5: The R-RMSECV values of the egg data for RSIMPLS.

on one hand, and on the other hand a model with low mean squared errors and optimal predictive power.

Also here, the computation times are drastically reduced: 5 seconds versus 106 seconds for the octane data, and 43.59 seconds versus 4316 seconds for the egg data.

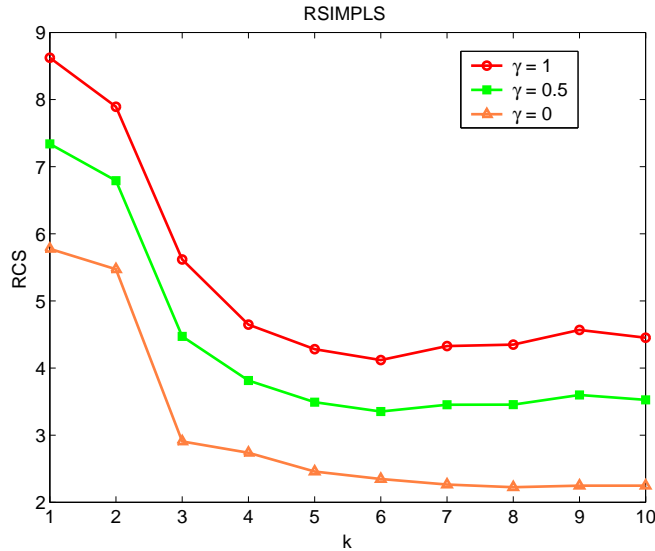


Figure 6: The RCS plot of the egg data for RSIMPLS.

5 Conclusions

In this paper we have proposed fast algorithms for performing cross-validation with robust PCR and PLSR methods. These methods are especially useful to draw RMSECV curves within reasonable time. We have also proposed a new robust selection criterion (RSC) by combining the robust RMSECV value with the robust root mean squared error.

The new algorithms are part of our Matlab toolbox ‘LIBRA: Library for Robust Analysis’ [24], available at <http://www.wis.kuleuven.ac.be/stat/robust.html>.

Appendix

The MCD estimator

The objective of the raw MCD estimator [13] is to find $h > \frac{n}{2}$ observations out of n whose covariance matrix has the smallest determinant. Its breakdown value is $\frac{[n-h+1]}{n}$, hence the number h determines the robustness of the estimator. Ideally the value of h corresponds with the number of regular points in the data. The larger h is, the more efficient the method, but the less robust and vice versa. In this paper we have always set h equal to 75% of the

number of samples (this is similar to the default setting in the LIBRA toolbox [24]). For its computation, the FAST-MCD algorithm [25] can be used. It roughly proceeds as follows :

1. Many random $(p+1)$ -subsets are drawn, which are enlarged to initial h -subsets using a C-step as explained in the next step. Note that at least $p+1$ observations are required to obtain a non-singular covariance matrix. If it is computationally feasible, all possible $(p+1)$ -subsets are used. Else 500 $(p+1)$ -subsets are drawn.
2. Within each h -subset, two C-steps are performed. Basically a C-step consists of computing first the classical center $\hat{\boldsymbol{\mu}}_0$ and the classical covariance matrix $\hat{\boldsymbol{\Sigma}}_0$ of the h observations. Then the robust distance (which depends on $\hat{\boldsymbol{\mu}}_0$ and $\hat{\boldsymbol{\Sigma}}_0$) of each point is computed as:

$$\text{RD}_{\hat{\boldsymbol{\mu}}_0, \hat{\boldsymbol{\Sigma}}_0}(\mathbf{x}_i) = \sqrt{(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_0)' \hat{\boldsymbol{\Sigma}}_0^{-1} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_0)}. \quad (10)$$

A new h -subset is formed by the h observations with smallest robust distance.

3. For the 10 h -subsets with the best value for the objective function, C-steps are performed until convergence. Other time saving techniques can be applied and are described in [25].
4. The raw MCD estimates of location $\hat{\boldsymbol{\mu}}_{raw}$ and scatter $\hat{\boldsymbol{\Sigma}}_{raw}$ are the classical mean and classical covariance matrix (multiplied by a consistency factor) of the h -subset H_0 with lowest objective function.
5. Next, a reweighting step is applied for efficiency purposes. Every observation is multiplied by a weight based on its robust distance $\text{RD}_{\hat{\boldsymbol{\mu}}_{raw}, \hat{\boldsymbol{\Sigma}}_{raw}}(\mathbf{x}_i)$. When this squared distance is larger than the 0.975 quantile of the χ_k^2 distribution, the weight is set equal to 0 and else to 1. The classical mean and covariance matrix of the weighted observations are the final robust center $\hat{\boldsymbol{\mu}}_{MCD}$ and scatter matrix $\hat{\boldsymbol{\Sigma}}_{MCD}$.

Fast cross-validation for the MCD has been fully described in [18]. When one observation is removed, the cross-validated MCD estimator seeks for the optimal $(h-1)$ -subset (instead of h -subset) in order to obtain a breakdown value as close as possible to the breakdown value of the full estimator. The approximate algorithm proceeds as follows:

1. First the MCD algorithm is performed on the whole data set. The optimal h -subset H_0 , the center $\hat{\boldsymbol{\mu}}_{raw}$ and covariance matrix $\hat{\boldsymbol{\Sigma}}_{raw}$ before weighting are stored.

2. For each sample $i = 1, \dots, n$:
 - (a) Sample i is removed from the data set.
 - (b) Now the $(h - 1)$ -subset $H_{-i,0}$ with lowest objective function has to be found. Instead of obtaining it by resampling, an update of H_0 is made. When H_0 contains the i th observation, the remaining $(h - 1)$ points of H_0 are taken. On the other hand, when sample i does not belong to H_0 , the $(h - 1)$ points of H_0 with the smallest robust distance $\text{RD}_{\hat{\boldsymbol{\mu}}_{raw}, \hat{\boldsymbol{\Sigma}}_{raw}}(\mathbf{x}_i)$ are used to form $H_{-i,0}$.
 - (c) The mean and covariance matrix of the $(h - 1)$ points from $H_{-i,0}$ are computed, yielding $\hat{\boldsymbol{\mu}}_{-i,0}$ and the $\hat{\boldsymbol{\Sigma}}_{-i,0}$. Next, C-steps are applied and a reweighting step is performed, leading to $\hat{\boldsymbol{\mu}}_{-i,MCD}$ and $\hat{\boldsymbol{\Sigma}}_{-i,MCD}$.

The LTS-regression estimator

The LTS-estimator [26] provides a robust alternative for univariate least squares regression by seeking for the $\frac{n}{2} < h \leq n$ points for which the objective function:

$$\sum_{i=1}^h r_{(i)}^2 \quad (11)$$

is minimized, where $r_{(1)} \leq r_{(2)} \leq \dots \leq r_{(h)}$ are the ordered residuals. The LTS fit thus corresponds with the least squares fit on this h -subset. In [26] a fast algorithm for the LTS estimator has been described. It is very similar to the FAST-MCD procedure (described above). The differences are:

1. In the first step, many random p -subsets are drawn, and the regression fit through these data points is determined. Note that p points are usually sufficient to estimate the regression coefficients in a p -dimensional space.
2. A C-step is applied to a h -subset. It consists of executing least squares regression on the h observations, and computing the residuals of all the n samples. The h data points with smallest absolute residual are stored and define a new h -subset.
3. In the reweighting step, the weights are defined based on the absolute standardized residuals. Observations with absolute standardized residual smaller than $\sqrt{\chi_{1,0.975}^2}$

receive weight 1, and else 0. The final LTS estimates are the least squares coefficients based on the observations with weight 1.

Also fast cross-validation for LTS is similar to the approach applied for the MCD estimator. Let H_0 denote the optimal h -subset based on the full data set. If CV is performed for the i th datum and the i th case belongs to H_0 , then $H_{-i,0}$ contains the remaining $h - 1$ observations. Otherwise the observation with largest absolute residual is removed from H_0 .

The MCD-regression estimator

MCD-regression [14] is a robust method for multivariate regression, such as in model (8). It starts by computing the center and the covariance matrix of the joint t - and y -variables by means of the reweighted MCD estimator, yielding $\hat{\boldsymbol{\mu}}_{MCD} = (\hat{\boldsymbol{\mu}}_t, \hat{\boldsymbol{\mu}}_y)'$ and $\hat{\boldsymbol{\Sigma}}_{MCD}$:

$$\hat{\boldsymbol{\Sigma}}_{MCD} = \begin{pmatrix} \hat{\boldsymbol{\Sigma}}_t & \hat{\boldsymbol{\Sigma}}_{ty} \\ \hat{\boldsymbol{\Sigma}}_{yt} & \hat{\boldsymbol{\Sigma}}_y \end{pmatrix}.$$

Here, $\hat{\boldsymbol{\Sigma}}_t$ is an estimate of the scatter matrix of the t -variables, $\hat{\boldsymbol{\Sigma}}_y$ of the y -variables and $\hat{\boldsymbol{\Sigma}}_{ty}$ of the cross-covariance matrix between the t - and y -variables. Note that as the reweighted MCD estimates are used, they correspond with a weighted mean and weighted empirical covariance matrix (of the non-outliers found with the raw estimates). The raw MCD estimates of the (t, y) variables, denoted as $\hat{\boldsymbol{\mu}}_{raw}$ and $\hat{\boldsymbol{\Sigma}}_{raw}$ are not used to obtain regression estimates but they will be useful in the cross-validation.

The robust regression parameters are then given by:

$$\begin{aligned} \hat{\mathbf{A}}_{k,q} &= \hat{\boldsymbol{\Sigma}}_t' \hat{\boldsymbol{\Sigma}}_{ty} \\ \hat{\mathbf{a}}_0 &= \hat{\boldsymbol{\mu}}_y - \hat{\mathbf{A}}_{q,k}' \hat{\boldsymbol{\mu}}_t \\ \hat{\boldsymbol{\Sigma}}_f &= \hat{\boldsymbol{\Sigma}}_y - \hat{\mathbf{A}}_{q,k}' \hat{\boldsymbol{\Sigma}}_t \hat{\mathbf{A}}_{q,k} \end{aligned} \tag{12}$$

These initial regression estimates are followed by a one-step reweighting, similar to LTS-regression. Now, the residual distance $\sqrt{\mathbf{r}_i' \hat{\boldsymbol{\Sigma}}_f^{-1} \mathbf{r}_i}$ of each observation is calculated and a weight equal to 1 is assigned if this distance is smaller than $\sqrt{\chi_{q,0.975}^2}$. The final estimates then correspond with the least squares estimates of the weighted observations.

The key step in the fast computation of the R-RMSECV value for the MCD-regression is the extraction of the regression results for a model with k regression components from the parameters corresponding with k_{max} regressors (see for example step 4 in Section 3.2). This proceeds as follows :

1. Performing MCD-regression with k_{max} yields the raw estimates $\hat{\Sigma}_{raw,k_{max}}$ and $\hat{\boldsymbol{\mu}}_{raw,k_{max}}$:

$$\hat{\Sigma}_{raw,k_{max}} = \begin{pmatrix} \hat{\Sigma}_{t,k_{max}}^{raw} & \hat{\Sigma}_{ty,k_{max}}^{raw} \\ \hat{\Sigma}_{yt,k_{max}}^{raw} & \hat{\Sigma}_{y,k_{max}}^{raw} \end{pmatrix}$$

$$\hat{\boldsymbol{\mu}}_{raw,k_{max}} = \begin{pmatrix} \hat{\boldsymbol{\mu}}_{t,k_{max}}^{raw} & \hat{\boldsymbol{\mu}}_{y,k_{max}}^{raw} \end{pmatrix}'.$$

2. For k -dimensional scores t , preliminary estimates of the center $\hat{\boldsymbol{\mu}}_{raw,k}$ and scatter $\hat{\Sigma}_{raw,k}$ of the joint (t, y) -variables are then obtained by selecting the appropriate rows and columns from $\hat{\Sigma}_{raw,k_{max}}$ and $\hat{\boldsymbol{\mu}}_{raw,k_{max}}$. This means that:

$$\hat{\Sigma}_{raw,k} = \begin{pmatrix} \hat{\Sigma}_{t,k_{max}}^{raw}(1:k, 1:k) & \hat{\Sigma}_{ty,k_{max}}^{raw}(1:k, :) \\ \hat{\Sigma}_{yt,k_{max}}^{raw}(:, 1:k) & \hat{\Sigma}_{y,k_{max}}^{raw} \end{pmatrix}$$

and

$$\hat{\boldsymbol{\mu}}_{raw,k} = \begin{pmatrix} \hat{\boldsymbol{\mu}}_{t,k_{max}}^{raw}(1:k) & \hat{\boldsymbol{\mu}}_{y,k_{max}}^{raw} \end{pmatrix}'.$$

where for any matrix \mathbf{A} the notation $\mathbf{A}(1:k, :)$ indicates the first k rows of \mathbf{A} and $\mathbf{A}(:, 1:k)$ its first k columns.

3. Next, C-steps (as in the MCD algorithm) are performed starting with $\hat{\boldsymbol{\mu}}_{raw,k}$ and $\hat{\Sigma}_{raw,k}$ in order to ameliorate the estimate of the center and covariance matrix of the first k scores and the dependent variables.
4. Finally, the same reweighting step as in the original algorithm is performed, leading to the final cross-validated regression parameters.

References

- [1] A. D. R. McQuarrie and C. L. Tsai. *Regression and Time Series Model Selection*. World Scientific Publishing, Singapore, 1998.
- [2] J. Neter, M. H. Kutner, C. J. Nachtsheim, and W. Wasserman. *Applied Linear Statistical Models*. McGraw-Hill, Boston, 1996.
- [3] R. Bruggemann and H. Bartel. *J. Chem. Inf. Comput. Sci.*, 39:211–217, 1999.
- [4] R. Rajkó and K. Héberger. *Chemom. Intell. Lab. Syst*, 57:1–14, 2001.
- [5] K. Héberger and Rajkó R. *J. Chemometrics*, 16:436–443, 2002.
- [6] S. Wold. *Technometrics*, 20:397-405, 1978.
- [7] B. Li, J. Morris, and B. Martin. *Chemom. Intell. Lab. Syst*, 64:79–89, 2002.
- [8] E. Ronchetti, C. Field, and W. Blanchard. *J. Amer. Statist. Assoc*, 92:1017–1023, 1997.
- [9] M. Wasim and R. G. Brereton. *Chemom. Intell. Lab. Syst*, 72:133–151, 2004.
- [10] B. Mertens, T. Fearn, and M. Thompson. *Stat. Comput.*, 5:227–235, 1995.
- [11] M. Hubert and S. Verboven. *J. Chemometrics*, 17:438–452, 2003.
- [12] M. Hubert and K. Vanden Branden. *J. Chemometrics*, 17:537–549, 2003.
- [13] P. J. Rousseeuw. *J. Amer. Statist. Assoc*, 79:871–880, 1984.
- [14] P. J. Rousseeuw, S. Van Aelst, K. Van Driessen, and J. Agulló. *Technometrics*, 46:293–305, 2004.
- [15] L. Norgaard, A. Saudland, J. Wagner, J.P. Nielsen, L. Munck, and Engelsen S.B. *Applied Spectroscopy*, 54:413–419, 2000.
- [16] I.T. Joliffe. Springer, New York, 1986.
- [17] M. Hubert, P. J. Rousseeuw, and K. Vanden Branden. *Technometrics*, 2005. To appear.
- [18] M. Hubert and S. Engelen. Fast cross-validation for high-breakdown resampling algorithms for PCA, 2004. Submitted.

- [19] S. Engelen, M. Hubert, and K. Vanden Branden. *Austrian J. of Statistics*, 2005. To appear.
- [20] P. J. Rousseeuw and B. C. van Zomeren. *J. Amer. Statist. Assoc.*, 85:633–651, 1990.
- [21] K.H. Esbensen, S. Schönkopf, and T. Midtgaard. Camo, Trondheim, 1994.
- [22] B. J. Kemps, F. R. Bamelis, B. De Ketelaere, K. Mertens, B. Kamers, K. Tona, E. M. Decuypere, and J. G. De Baerdemaeker. 2004. Submitted.
- [23] S. de Jong. *Chemom. Intell. Lab. Syst.*, 18:251–263, 1993.
- [24] S. Verboven and M. Hubert. *Chemom. Intell. Lab. Syst.*, 2004. To appear.
- [25] P. J. Rousseeuw and K. Van Driessen. *Technometrics*, 41:212–223, 1999.
- [26] P. J. Rousseeuw and K. Van Driessen. An algorithm for positive-breakdown methods based on concentration steps. In W. Gaul, O. Opitz, and M. Schader, editors, *Data Analysis: Scientific Modeling and Practical Application*, pages 335–346, New York, 2000. Springer-Verlag.

List of tables

Table 1: Scheme to perform naive cross-validation for the RPCR method.

$\forall i = 1, \dots, n$	remove case i	
	$\forall k = 1, \dots, k_{max}$	apply ROBPCA $\Rightarrow \mathbf{t}_{m,k}^{-i}$ for all $m \neq i$
		apply LTS/MCD-regression \Rightarrow $\hat{\mathbf{a}}_{0,k}^{-i}$ and $\hat{\mathbf{A}}_{k,q}^{-i}$
		compute $\mathbf{r}_{-i,k}$

Table 2: Scheme to perform fast cross-validation for the RPCR method.

set $k = k_{max}$		
$\forall i = 1, \dots, n$	remove case i	
	apply fast CV for ROBPCA $\Rightarrow \mathbf{t}_{m, k_{max}}^{-i}$ for all $m \neq i$	
	apply fast CV for LTS/MCD-regression $\Rightarrow \hat{\mathbf{a}}_{0, k_{max}}^{-i}$ and $\hat{\mathbf{A}}_{k_{max}, q}^{-i}$ (LTS) $\Rightarrow \hat{\boldsymbol{\mu}}_{k_{max}}^{-i}$ and $\hat{\boldsymbol{\Sigma}}_{k_{max}}^{-i}$ (MCD)	
	$\forall k = 1, \dots, k_{max}$	compute $\hat{\mathbf{a}}_{0, k}^{-i}$ and $\hat{\mathbf{A}}_{k, q}^{-i}$ compute $\mathbf{r}_{-i, k}$

Table 3: Scheme to compute the weights for the R-RMSECV curve for the RPCR method.

set $k = \hat{k}_{max}$		
apply ROBPCA $\Rightarrow \mathbf{t}_{i, \hat{k}_{max}}$ for all i		retain H_0, H_1 and H_{freq}
apply LTS/MCD-regression $\Rightarrow \hat{\mathbf{a}}_{0, \hat{k}_{max}}$ and $\hat{\mathbf{A}}_{\hat{k}_{max}, q}$ (LTS), $\hat{\boldsymbol{\mu}}_{raw, \hat{k}_{max}}$ and $\hat{\boldsymbol{\Sigma}}_{raw, \hat{k}_{max}}$ (MCD)		retain H_{freq}
$\forall k = 1, \dots, \hat{k}_{max}$	compute $\hat{\mathbf{a}}_{0, k}, \hat{\mathbf{A}}_{k, q}$ and $\hat{\boldsymbol{\Sigma}}_{\mathbf{e}, k}$	retain H_{opt}
	compute $\text{ResD}_{i, k}$ and $w_{i, k}$	

Table 4: Scheme to perform fast cross-validation for the RSIMPLS method.

set $k = k_{max}$		
$\forall i = 1, \dots, n$	remove case i	
	apply fast CV for ROBPCA $\Rightarrow \mathbf{t}_{m, k_{max}}^{-i}$ and v_m^{-i} for all $m \neq i$	
	compute weighted mean and covariance of the $(\mathbf{t}_{m, k_{max}}^{-i}, \mathbf{y}^{-i}) \Rightarrow \hat{\boldsymbol{\mu}}_{k_{max}}^{-i}$ and $\hat{\boldsymbol{\Sigma}}_{k_{max}}^{-i}$	
	$\forall k = 1, \dots, k_{max}$	derive $\hat{\boldsymbol{\mu}}_k^{-i}$ and $\hat{\boldsymbol{\Sigma}}_k^{-i}$
		compute $\hat{\mathbf{a}}_{0, k}^{-i}$ and $\hat{\mathbf{A}}_{k, q}^{-i}$
		compute $\mathbf{r}_{-i, k}$

Table 5: Scheme to compute the weights for the R-RMSECV curve of RSIMPLS.

set $k = k_{max}$	
apply ROBPCA $\rightarrow \mathbf{t}_{i,k_{max}}$ and v_i for all i	
compute weighted mean and covariance of the $(\mathbf{t}_{i,k_{max}}, \mathbf{y}) \Rightarrow \hat{\boldsymbol{\mu}}_{k_{max}}$ and $\hat{\boldsymbol{\Sigma}}_{k_{max}}$	
$\forall k = 1, \dots, k_{max}$	derive $\hat{\boldsymbol{\mu}}_k$ and $\hat{\boldsymbol{\Sigma}}_k$
	compute $\hat{\mathbf{a}}_{0,k}$, $\hat{\mathbf{A}}_{k,q}$ and $\hat{\boldsymbol{\Sigma}}_{\mathbf{e},k}$
	compute $\text{ResD}_{i,k}$ and $w_{i,k}$

Legends of the figures

Figure 1: The RCS plot for the beer data for RSIMPLS.

Figure 2: The R-RMSECV values for the octane data for RPCR.

Figure 3: The R-RMSECV values for the egg data for RPCR.

Figure 4: The R-RMSECV values of the octane data for RSIMPLS.

Figure 5: The R-RMSECV values of the egg data for RSIMPLS.

Figure 6: The RCS plot of the egg data for RSIMPLS.