A novel robust PLS regression method inspired from boosting principles: RoBoost-PLSR

Maxime Metz, Florent Abdelghafour, Jean-Michel Roger, Matthieu Lesnoff

A novel robust PLS regression method inspired from boosting principles : RoBoost-PLSR

Maxime Metz[a,b], Florent Abdelghafour[a,b], Jean-Michel Roger[a,b], Matthieu Lesnoff[b,c,d]

[a] *ITAP, Univ Montpellier, INRAE, Institut Agro, Montpellier, France*
[b] *ChemHouse Research Group, Montpellier, France*
[c] *CIRAD, UMR SELMET, Montpellier, France*
[d] *SELMET, Univ Montpellier, CIRAD,INRA, Institut Agro, Montpellier,France*

Abstract

The calibration of Partial Least Square regression (PLSR) models can be disturbed by outlying samples in the data. In these cases the models can be unstable and their predictive potential can be depreciated. To address this problem, some robust versions of the PLSR algorithm were proposed. These algorithms rely on the downweighting of these outliers during calibration. To this end, it is necessary to estimate an inconsistency measurement between the samples and the model. However, this estimation is not trivial in high dimensions. This paper proposes a novel robust PLSR algorithm inspired from the principles of boosting : RoBoost-PLSR. This method consists of realising a series of one latent variable weighted PLSR. RoBoost-PLSR is compared with the PLSR algorithm calibrated with and without outliers and also with Partial Robust M-regression (PRM), a reference robust method. This evaluation is conducted on the basis of three simulated datasets and a real dataset. Finally Roboost-PLSR proves to be resilient to the tested outliers, and can achieve the performances of the reference PLSR calibrated

Email address: maxime.metz@inrae.fr (Maxime Metz)

without any outlier.

Keywords: Partial least squares, Outliers, Robustness, Boosting ;

1. Introduction

Partial Least Square Regression (PLSR) [1] is a usual data analysis method and a well-established tool in analytical chemistry. PLSR is particularly relevant for the processing of high dimensional data, especially when the number of explanatory variables exceeds the number of samples. The successful processing of these data is partly conditioned by the fact that the samples can be assimilated to a well-defined distribution. However, if some samples do not share the properties of this distribution, the PLSR model can be disturbed and its predictive quality depreciated [2]. These samples are designated as outliers in comparison with the other ones called inliers. In order to deal with the presence of outliers, numerous strategies have been developed in chemometrics [3–15]. This type of methods are called robust methods. Robust methods place confidence in the main mass the of data. These methods must be parsimonious so as not to exclude major samples who contribute strongly to the good predictive quality of the model. According to [16], "*For high-dimensional data this would result in a severe loss of information as long as the outliers still contain some valuable information, and thus intelligent robust methods adapt the weights according to the outlyingness or inconsistency of the observations.*". In fact, a major difficulty is therefore to determine relevant outlying measurements in order to give low importance to outliers (*e.g.* through weighting), while retaining some of their relevant properties.

In this article, the attention is focused on methods intended for the calibration of PLS1 models in presence of potential outliers. This means that the methods weight the samples through the PLSR in order to reduce the impact of outliers on model calibration. In that sense, only a few robust methods were proposed along with an available algorithm.

One of the first methods was proposed in [10]. This method carries out a robust least square regression for each explanatory variable. This means that the method considers independent variables with this procedure. This aspect was particularly argued in [17] because this process does not capture the multidimensional aspect of outliers.

To address to this problem, [18], developed the Partial Robust M-regression (PRM) method. PRM is frequently studied and used in chemometrics. PRM is based on the NIPALS algorithm trained on the iteratively reweighted matrices (representing the explanatory variables and responses). PRM consists of weighting the samples on the basis of a PLSR model with a predefined number of latent variables (LVs). This means that the weights are defined for a specific model (*i.e.* PLSR with K latent variables). To determine the $k < K$ models, weights must be specifically recomputed for each given k, as opposed to PLSR where each 1 to K LVs model cand be deduced at once from a K model. In PRM, an outlier is defined by a combination of the leverage estimation (*i.e.* the Euclidean distance between scores and the median of scores) and Y-residuals. A limitation of this method, is that outliers are detected using a PLSR model with a number of latent variables that is defined beforehand. In [10], this limitation is lifted by weighting the samples independently of the number of latent variables.

Considering these perspectives, authors propose a new robust PLSR algorithm that combines principles of gradient boosting within a modified framework derived from [10] : RoBoost-PLSR. Boosting is a statistical and machine learning principle consisting in assembling a series of weak models (*i.e.* partially explanatory models) that are adjusted between them. Finally, the prediction by the strong model is the sum of the predictions of each weak model.

The link between PLS and gradient boosting has already been studied and resulted in implementations for the processing of chemical data [19–23] Essentially, these approaches use numerous weak learners, computed sequentially from different sub-samples. Each new weak learner is computed from the previous ones using a loss function. Finally, the weak learners are all combined in a weight function according to their predictive potential. As for the RoBoost-PLSR framework, it proposes to apply the basic idea of gradient boosting : *i.e.* combining an ensemble of weak learners. The weak learners are defined here as weighted one-latent variable PLSR models. The weights are defined iteratively in order to reduce the contribution of outliers on the calculated model. The weak learners are then combined using an unweighted sum of the predictions of each weak learner.

This strategy enables the weighting of samples in the calibration set independently of the number of latent variables (LVs) while considering the multivariate nature of the samples.

The objective of this paper is to provide a study of the proposed new RoBoost-PLSR method using simulated and real data. These data represent different types of outliers that could be present in spectral databases.

The first section presents the theoretical principles of RoBoost-PLSR and the associated algorithm. The following section presents the data and the methods used to evaluate and compare RoBoost-PLSR with standard PLSR and PRM. Finally, the last section presents applications for the calibration and prediction performances of RoBoost-PLSR on the basis of simulated and real data.

2. Theoretical background of the RoBoost-PLSR method

2.1. Notations

Capital bold characters will be used for matrices, *e.g.* $\mathbf{X}$; small bold characters for column vectors, *e.g.* $\mathbf{x}_j$ will denote the j^{th} column of $\mathbf{X}$; row vectors will be denoted by the transpose notation, *e.g.* $\mathbf{x}_i^{\mathrm{T}}$ will denote the i^{th} row of $\mathbf{X}$; italicised characters will be used for scalars, *e.g.* matrix elements x_{ij} or indices i. Constant scalars will be denoted with italicised characters, *e.g.* number of samples n. $\mathbb{1}$ will represent a column vector of ones, of proper dimension.

2.2. Principle of the method

RoBoost-PLSR consists in achieving a series of K unidimensional (1 LV) iteratively reweighted PLSR [24] models. The weigthed PLSR algorithm used is weighted-NIPALS [25] (steps 6-8,12). Each $K+1$ model is calibrated with the residuals ($\mathbf{X}$ and $\mathbf{Y}$) of the previous K models. Sample weights are defined thanks to a Bisquare function [26]. This weight function requires the optimisation of a hyperpameter. This optimisation can be done through a cross-validation procedure or an optimisation on an external validation set.

The more the samples deviate from the model, the closer the weights must be to 0. Iteratively, models are updated according to the weights previously attributed until convergence to a stable solution.

Within each PLSR model. Weights are computed according to a combination of three measurements :

— X-residuals

— Y-residuals

— Leverage

2.3. Algorithm

Let $\mathbf{X}$ be an $[n \times m]$ matrix containing n samples described by m variables. Let $\mathbf{y}$ be a response vector containing n samples. In this article $\mathbf{y}$ is always considered as a vector, *i.e.* the response is univariate.

For a definite number of K latent variables, the algorithm proceeds as described below :

Algorithm RoBoost-PLSR for K LV

Calibration$(\mathbf{X}, \mathbf{y}, K)$

1: Set $k = 1$

2: Set $\mathbf{X}_0 = \mathbf{X}$

3: Initialise the $[n \times n]$ weight matrix $\mathbf{D}$:

$\mathbf{D} = diag(d_1, d_2, ..., d_n)$ such as $\forall i \in [1, n], d_i = \frac{1}{n}$

4: Derive the weighted means :

$\bar{\mathbf{x}}_k^{\mathrm{T}} = \mathbb{1}^{\mathrm{T}}\mathbf{D}\mathbf{X}_{k-1}$

$\bar{y}_k = \mathbb{1}^{\mathrm{T}}\mathbf{D}\mathbf{y}_{k-1}$

5: Center the data :

$\mathbf{X}_k = \mathbf{X}_{k-1} - \mathbb{1}\bar{\mathbf{x}}_k^{\mathrm{T}}$

$\mathbf{y}_k = \mathbf{y}_{k-1} - \mathbb{1}\bar{y}_k$

6: Derive the k^{th} weighted loading's weights :

$$\mathbf{w}_k = \frac{\mathbf{X}_k^{\mathrm{T}}\mathbf{D}\mathbf{y}_k}{||\mathbf{X}_k^{\mathrm{T}}\mathbf{D}\mathbf{y}_k||}$$

7: Derive the k^{th} scores :

$\mathbf{t}_k = \mathbf{X}_k\mathbf{w}_k$

8: Derive the k^{th} weighted *loading vectors* of $\mathbf{X}_k$ and the k^{th} regression coefficient vector :

$$\mathbf{p}_k = \frac{\mathbf{X}_k^{\mathrm{T}}\mathbf{D}\mathbf{t}_k}{\mathbf{t}_k^{\mathrm{T}}\mathbf{D}\mathbf{t}_k}$$

$$q_k = \frac{\mathbf{y}_k^{\mathrm{T}}\mathbf{D}\mathbf{t}_k}{\mathbf{t}_\mathbf{k}^{\mathrm{T}}\mathbf{D}\mathbf{t}_k}$$

9: Derive the Y-residuals ($\mathbf{f}$), X-residuals ($\mathbf{E}$), leverage estimation ($\mathbf{l}$) corresponding to the current k^{th} latent variable :

$\mathbf{E} = \mathbf{X}_k - \mathbf{t}_k \mathbf{p}_k^{\mathrm{T}}$

$\mathbf{f} = \mathbf{y}_k - \mathbf{t}_k q_k$

$\mathbf{l} = \mathbf{t}_k$

10: Estimate and update the weights for each $i \in [1, n]$ sample

$\alpha_i = B(\frac{||\mathbf{e}_i||}{c_\alpha \times s_\alpha})$

$\beta_i = B(\frac{f_i}{c_\beta \times s_\beta})$

$\gamma_i = B(\frac{l_i}{c_\gamma \times s_\gamma})$

$d_i = \frac{1}{n} \times \alpha_i \times \beta_i \times \gamma_i$

With s_α , s_β , s_γ being respectively the median of $\{||\mathbf{e}_i||\}_n$, $\{|\mathbf{f}_i|\}_n$ and $\{|l_i|\}_n$ $\forall i \in [1, n]$. c respectively denotes fixed constants in each weight function. In this case the weight function B is the Bisquare function defined as :

$B(x) = (1 - x^2)^2$, for $|x| < 1$, B(x) = 0, for |x| > 1

11: Go back to step (4) until convergence of successive q's.

12: while $k < K$

$\mathbf{X}_{k+1} = \mathbf{X}_k - \mathbf{t}_k \mathbf{p}_k^{\mathrm{T}}$

$\mathbf{y}_{k+1} = \mathbf{y}_k - \mathbf{t}_k q_k$

set $k = k + 1 \rightarrow$ then go to step (3)

End Calibration

Prediction($\mathbf{x}^*$, fitted model)

Fitted model $\{ [q_1, q_2, ..., q_K], [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_K], [\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_K], [\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, ..., \bar{\mathbf{x}}_K] \}$

The estimation of $\hat{y}^*$ for a given new sample $\mathbf{x}^*$ is :

$\hat{y}^* = \sum_{k=1}^{K} \hat{y}_k^*$

The computation of $\hat{y}_k^*$ is given by :

$\hat{y}_k^* = \mathbf{t}_k q_k$ with,

$\mathbf{t}_k = \mathbf{x}_k^* \mathbf{w}_k$ and,

$\mathbf{x}_k^* = (\mathbf{x}_{k-1}^* - \bar{\mathbf{x}}_k^{\mathrm{T}}) - (\mathbf{x}_{k-1}^* - \bar{\mathbf{x}}_k^{\mathrm{T}}) \mathbf{w}_k \mathbf{p}_k^{\mathrm{T}}$

2.4. Method properties

The RoBoost-PLSR framework is designed foremost to facilitate the estimation of the samples weights, *i.e.* estimating the deviation from a model in large dimensions (a large number of latent variables).

Firstly, estimating the weights of samples independently for each latent variable provides a simpler estimation of leverage points. Indeed, in usual robust PLSR algorithms, leverage is computed either thanks to Euclidean or Mahalanobis distances between the scores and the centre of the model. In high dimensional spaces (numerous LVs), this estimation is not so trivial. As a matter of fact, in the case of a Euclidean distance, the latest LVs have only a minor contribution to the leverage value. This is naturally due to the decreasing magnitude of scores. Nevertheless, the predictive potential of these latest LVs is not necessarily lesser. In the case of a Mahalanobis distance, the contributions of all LVs become equal in the computation of the leverage value. This can be equally detrimental, since the predictive

potentials of the LVs are most oftenly uneven.

Secondly, the proposed method considers X-residuals, which is not the case in usual robust PLSR methods. The inclusion of these residuals provides additional information that cannot be expressed solely by leverage and Y-residuals.

Thirdly, the method does not provide regression coefficients. Contrary to other robust methods such as PRM, in this case, it is not trivial to compute them. Indeed, the proposed algorithm for RoBoost-PLSR does not allow an estimation of the rotation matrix $\mathbf{R}$. Models can nevertheless be interpreted by analysing the loadings, although it is less convenient. Indeed, it is possible to observe the loadings and derive the most influential variables within each 1LV model. However, unlike in conventional PLSR, it is not possible yet to determine the relative influences of variables at the scale of the whole K-LV model

Fourthly, like PLSR, RoBoost-PLSR makes it possible to deduce any of the 1 to K LVs models from the calibration of a single K LVs model. This preserves the operability during the validation and parameterisation process of the RoBoost-PLSR method.

3. Material and methods

3.1. Data and software

RoBoost-PLSR was evaluated on three simulated datasets and one real dataset. Simulations were used to introduce controlled disturbances while the real dataset was used to confirm and support the simulations results. The algorithms were developed using the R software packages. RoBoost-PLSR

was developed on the basis of "rnirs" functions. The functions and data associated with RoBoost-PLSR are available on Github "RoBoost-PLSR". PRM was implement with the "prms" function available in the "sprm" package.

3.2. Simulated Data

The three simulations were generated according to the generic framework proposed by [27]. Contrary to the simulation strategies usually used to evaluate robust methods, the data were not simulated from a real model. The data are simulated from a combination of spectral signatures, some of which are related to one or more variables to be predicted ($\mathbf{Y}$ matrix).

The simulations were based on a combination of pure artificial spectra and controlled noises. The aim of each simulation was to reproduce the common external disturbances that can occur when calibrating a predictive model. It consisted of adding to the dataset an additional set of predefined outliers that have a negative effect on the performance of the models. The first simulation introduced pure Y outliers. The second simulation introduced contaminant induced outlier *i.e.* X-outliers occurring when an external substance pollutes the calibration samples. These individuals are strong outliers because they can be easily distinguished from inliers (*e.g.* by a spectra plot). The third simulation introduced slight X-outliers. For all simulations, 900 inliers and 100 outliers were simulated. Descriptions of the simulation are available in the appendix in table form. The differences between simulated inliers and outliers are highlighted in bold in the tables.

3.3. Real dataset

The real dataset consisted of NIR spectral samples acquired from two types of feed materials : soybean and meat and bone meal. Each sample-spectrum was associated with its Y-response *i.e.* the chemical reference measurement of its protein content. The spectra were measured with a *Foss* spectrometer in the spectral range $[1100-2498$ nm] with a 2 nm spectral resolution. These data were extracted from the "PROT" database provided by the CRA-W (Agronomic Research Centre of Wallonia, Belgium). This database was already used for the development and comparison of local methods [28].

3.4. Evaluation strategies

The purpose is to evaluate the behaviour of the newly introduced RoBoost-PLSR methods in presence of outliers during calibration. The calibrated model is then evaluated on a validation set. The reference against which all models were compared was a PLSR calibrated on a dataset without outliers (and will be designated as such). Roboost-PLSR was evaluated and compared with two standard regression algorithms : PLSR and PRM.
In the case of the simulations, the weight parameters of PRM and RoBoost-PLSR were optimised according to the validation set. Only the results of the optimal (*i.e.* the parameters that provide the minimum value of RMSEP) parameters of RoBoost-PLSR and PRM were presented in the following section. The calibration sets were generated from 500 samples (400 inliers and 100 outliers). The resulting models were studied with validation sets containing 500 inliers. The prediction performance of the RoBoost-PLSR method was studied also as a function of the proportion of

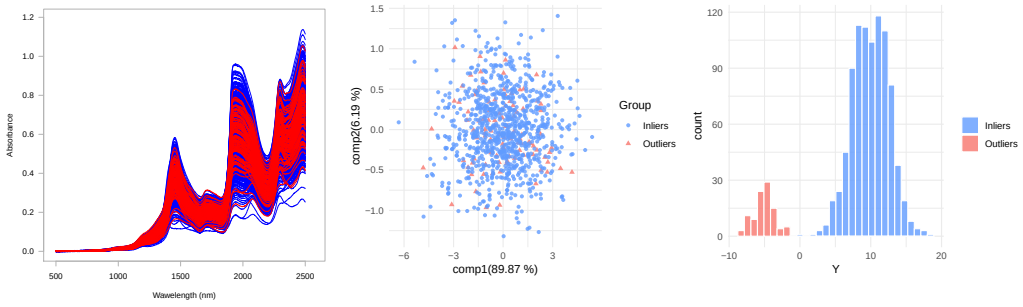
outliers . It varied from 10% to 40%. These performances were compared to the reference model (PLSR without outliers). This study was carried out with the three simulated datasets.

In the case of the real dataset, the weights parameters of PRM (using the Hampel function) and RoBoost-PLSR were optimised according to the validation set. Only the results of the optimal parameters of RoBoost-PLSR and PRM were presented in the following section. The calibration set was composed of 457 soybean protein (TTS) samples and 100 animal-protein (ANF) samples that represent the outliers. The validation was conducted on 50 additional samples of soybean and results were evaluated through Root Mean Square Error of Prediction (RMSEP).

The evaluation strategy also aimed at assessing the weights attributed to each sample. Weights are evaluted for the number of latent variable resulting in the minimum RMSEP respectively for PRM and Roboost.

4. Results and discussion

4.1. Simulation 1 : pure Y-outliers



(a) (b) (c)

FIGURE 1: Simulated dataset 1. (a) Spectra, (b) PCA projection of spectra and (c) distributions of Y-responses. Inliers are represented in blue, while outliers are in red.

Figure 1 presents the properties of the simulated dataset with Y-outliers. Figure 1a shows that inliers spectra (in red) blend perfectly with the rest of the population. Likewise, there is no separation of the two populations of spectra when projected on the two first principal components of a PCA (see Figure 1b). The same behaviour is observed up to the 10^{th} component. In this simulation, the outliers are simulated to display significant differences in terms of Y. Figure 1c shows that the distribution of the outliers is not similar to the distribution of inliers. The samples are distinguished only by their response values ($\mathbf{y}$) and not by their explanatory variables ($\mathbf{X}$).

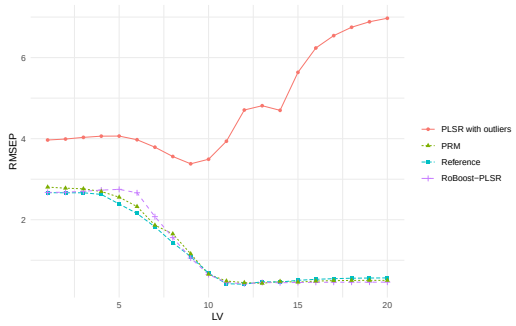


FIGURE 2: Evolution of the RMSEP as a function of latent variables for the reference, PLSR without outliers, PRM and RoBoost-PLSR for the dataset 1

Figure 2 presents four curves showing the RMSEP evolution as a function of the number of LVs, for PLSR calibrated with outliers, the reference, PRM and RoBoost-PLSR, both calibrated with outliers. This figure shows that pure Y-outliers have an impact on the calibration of a PLSR model. In this case, the standard PLSR model calibrated on data including outliers (red curve) achieves very poor prediction performances compared to the reference model (blue curve).

Figure 2 also shows that PRM achieves similar performances with the reference. The behaviour of the RMSEP curve of PRM is similar to the one of the reference along the LVs.

When RoBoost-PLSR (purple curve) is calibrated with Y-outliers, it achieves also similar performances with the reference along the LVs. This means that RoBoost-PLSR attributes low weights to outliers and reaches the best performance of the reference. The behaviour of the RoBoost-PLSR RMSEP curve is close to the reference. This means that the attribution of a weights close to 0 to the outliers for RoBoost-PLSR is independent of the selected

number of LVs.

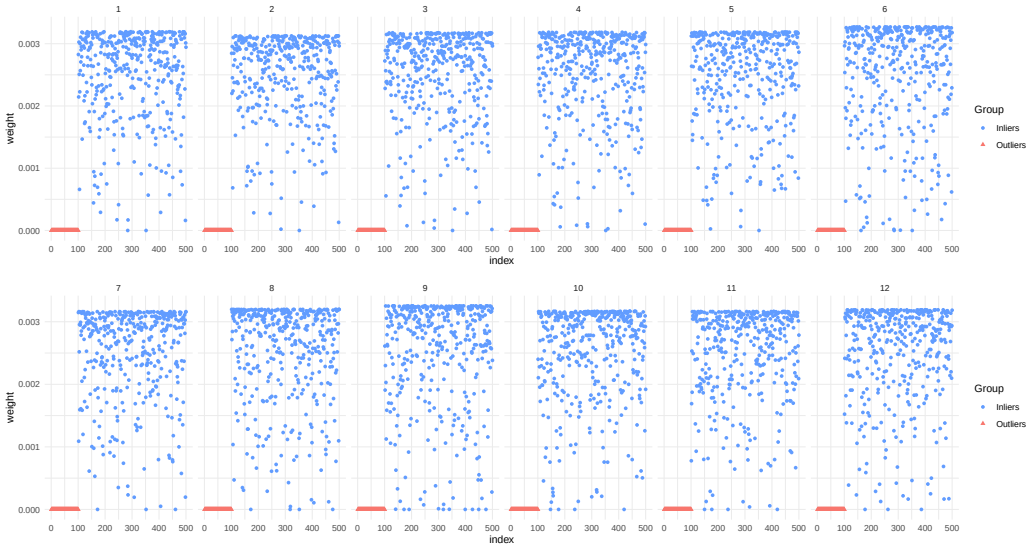


FIGURE 3: Repartition of the weights attributed to outliers (red) and inliers (blue) during the calibration of RoBoost-PLSR over 12 latent variables

Figure 3 shows the weights attributed by RoBoost-PLSR to outliers and inliers for the best performing model (12 LVs). Since the first LV, the outliers weights are close to 0. Few inliers are also erroneously assigned low weights during calibration. However, in this simulation, this distortion has no impact on prediction performance of RoBoost-PLSR, as shown by Figure 2. It can be observed that there are differences in ceiling values for the weights between LVs. This is due to the normalization of the weights. Actually, the weight of a given sample varies for each LV. At some point, it is possible that an increasing amount of samples are attributed high weights. Therefore, due to normalisation, the maximum value of the weights decreases as more samples

287 are considered relevant from RoBoost-PLSR.

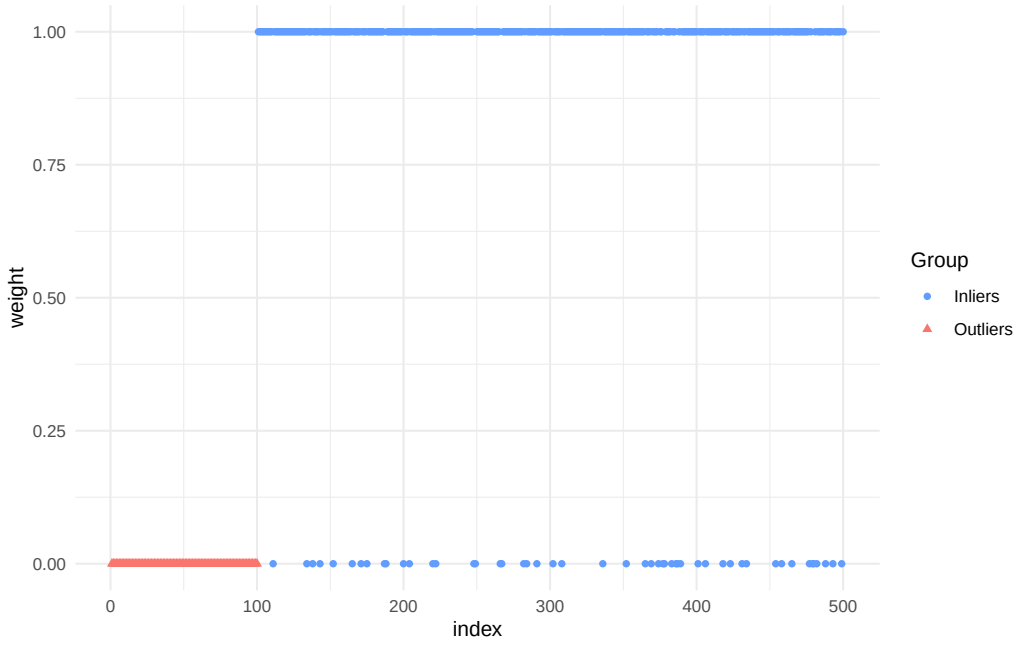
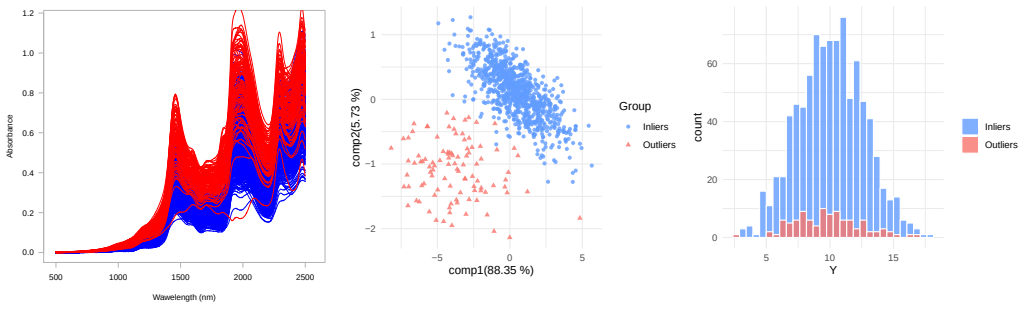


FIGURE 4: Repartition of the weights attributed to outliers (red) and inliers (blue) during the calibration of PRM for respectively 13 LVs

288 Figures 4 show the weights attributed to outliers and inliers in the
289 calibration set for 13 LVs (the best performing model)
290 Figure 4 shows a clear separation between outliers and inliers weigths with
291 a 13 LVs PRM model. Some inlier samples have a weight of 0 but the vast
292 majority of inlier samples have a weight of 1. As the performance curves
293 of PRM and the reference are almost similar, this does not disturb model
294 calibration.

295 *4.2. Simulation 2 : contaminant induced outliers*



(a) (b) (c)

FIGURE 5: Simulated dataset 2. (a) Spectra, (b) PCA projection of spectra and (c) distributions of Y-responses. Inliers are represented in blue, while outliers are in red.

296 Dataset 2 introduces X-outliers. The purpose is to simulate the impact
297 of a contamination of samples during spectral measurements without any
298 anomaly for the reference measures $\mathbf{y}$. Figure 5a shows that such outliers
299 (in red) overlap with standard observations. The difference between the
300 two groups is very faintly apparent on the spectra plot. Figure 5b shows
301 a separation of outliers from inliers on a projection onto the two first
302 principal components of PCA. The two groups are contiguous though,
303 which implies that some outliers could be confounded with inliers. Figure
304 5c shows the distribution of Y reponses for both outliers and inliers. Data
305 are simulated so that the outliers responses match the same distribution as
306 inliers. In practice, this situation corresponds to the possibility of conducting
307 rigorous reference measurements in controlled laboratory conditions for
308 chemical measures, while the spectral measurements are high-throughput
309 and possibly conducted in outdoor or uncontrolled conditions. In these cases

the extraction of information related to the spectra is more complex and probably requires additional LVs. Figure 5 therefore shows that the samples are distinguished only by their explanatory variables ($\mathbf{X}$) and not by their responses ($\mathbf{y}$).

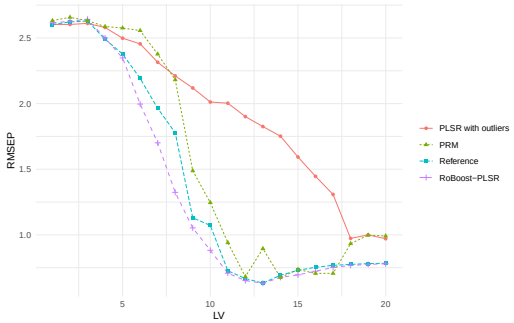


FIGURE 6: Evolution of the RMSEP as a function of latent variables for the reference, PLSR with outliers, PRM and RoBoost-PLSR for the dataset 2

Figure 6 shows that the PLSR with outliers (red curve) is less performant than the reference (blue curve). Indeed, the minimal RMSEP with outliers is $\simeq 1$ for 19 LVs whereas minimal RMSEP without outliers is $\simeq 0.4$ for 13 LVs. This means that the PLSR is sensitive to these outliers. In addition, Figure 6 shows that the number of LVs necessary to achieve the best performances is considerably higher (19 LVs vs. 13 LVs for the reference). This means that outliers add a detrimental information that requires the calculation of a PLSR model with a larger number of LVs [27].

Figure 6 shows that the PRM performance curve is close to the reference curve. This means that PRM can handle the presence of these outliers in the calibration set

Figure 6 shows that the RoBoost-PLSR curve reaches a minimum error close

to the reference. RoBoost-PLSR has a behaviour similar to the reference with the minimum RMSEP at 12 LVs. This means that RoBoost-PLSR attributes very low weights to the outliers but also to some inliers.

Both PRM and RoBoost-PLSR prove to be robust to "contaminant induced" which are simple X-outliers. RoBoost-PLSR seems to perform well and have the same behaviour as the reference.

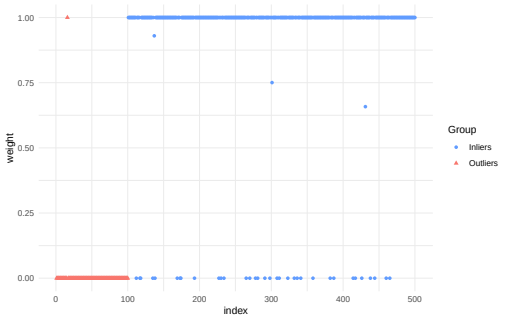


FIGURE 7: Repartition of the weights attributed to outliers (red) and inliers (blue) during the calibration of PRM for respectively 12 LVs

Figure 7 shows that the majority of inliers weights are 1 and the outliers weigths 0 for 12 LVs.

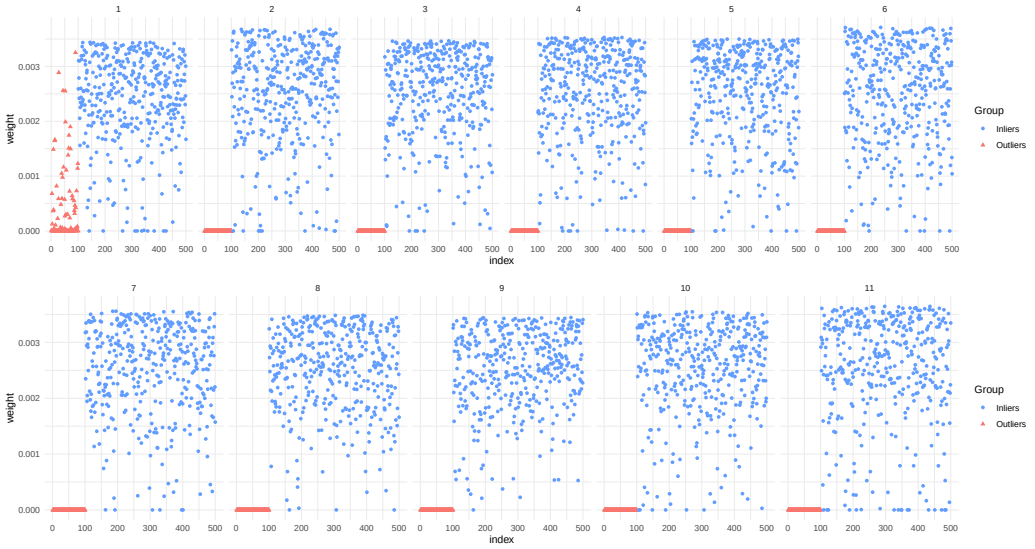
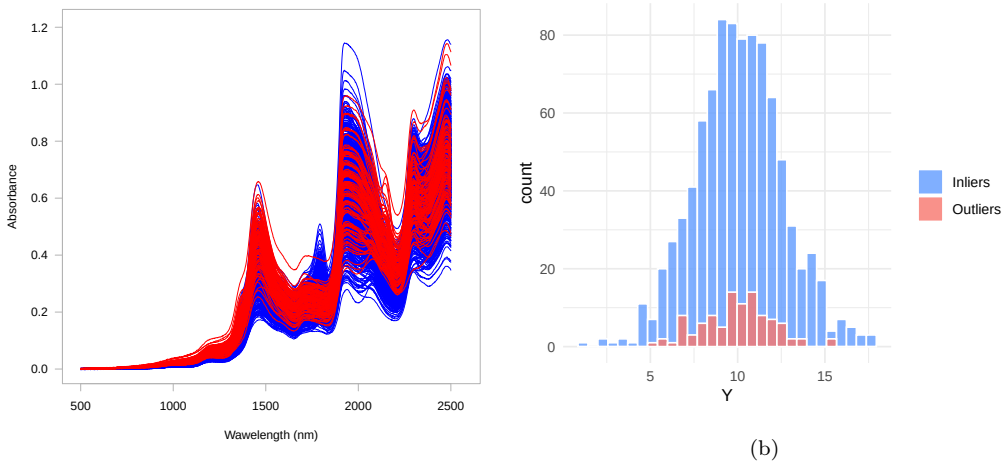


FIGURE 8: Repartition of the weights attributed to outliers (red) and inliers (blue) during the calibration of RoBoost-PLSR over 12 latent variables

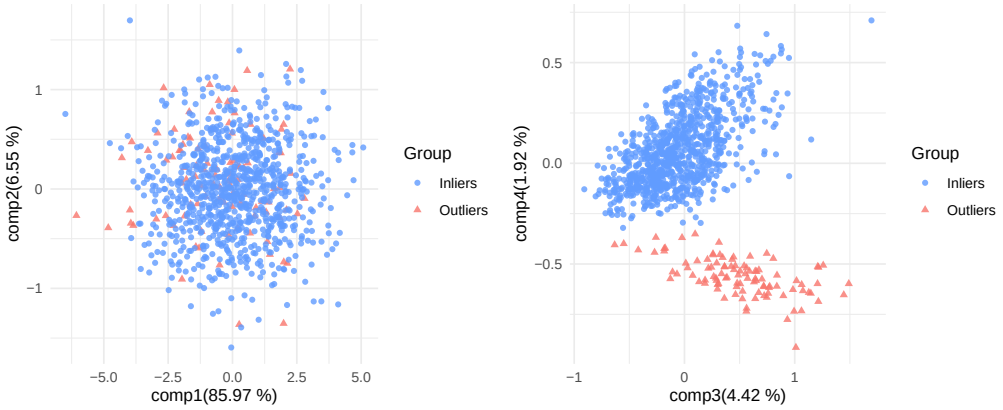
Figure 8 compares the weights assigned to outliers and inliers during the calibration process for RoBoost-PLS. It shows that for each LV, RoBoost-PLSR assigns to outliers a weight close to 0. As soon as the 2^{nd} latent variable, all outliers have a 0 weight. This result is due to the fact that the simulated spectra (outliers and inliers) have a first common source of variability and that, for the first LV, outliers are not detrimental to the model.

4.3. Simulation 3 : X-outliers induced by microvariations of the measuring environment



(a) (b)

FIGURE 9: Simulated dataset 3. (a) Spectra, (b) distributions of Y-responses. Inliers are represented in blue, while outliers are in red.



(a) (b)

FIGURE 10: Simulated dataset 3. (a) PCA projection of spectra onto components 1 and 2, (b) PCA projection of spectra onto components 3 and 4. Inliers are represented in blue, while outliers are in red.

Dataset 3 introduces further X-outliers. The purpose is to simulate the effect of microvariations of the measurement environment, such as temperature or hygrometry shifts *e.g.* when there is a timelapse between spectral measurements. The occurrence of such minor disturbances can alter the resulting spectra in imperceptible ways, yet, sufficiently to deteriorate PLSR models. Figure 9 shows the similarities between the outliers and the inliers. Spectra overlap so that the two populations are indistinguishable. Figure 10 presents their projection PCA axis. The first two components cannot help to differentiate outliers. It is only from the fourth component that the two groups are discriminated. However, this axis represents less than 2% of the total variability which describes the difficulty to determine the presence of such outliers beforehand. In terms of Y-responses, the distribution of outliers is simulated to match the inliers, hence the visible

overlay on figure 9. Finally, these samples could have been detected through the appropriate analysis. For instance, some outliers can be distinguished on PCA axes in this case. Nevertheless it is difficult to justify the removal of such samples from the presented graphs. Inherently, a sample should be discarded if it is detrimental to the prediction quality. To determine that, more elaborate methods should be considered, e.g. using a PLS model to detect the samples that diverge from the model. These types of approaches are very useful to understand the phenomena generating these outliers, but require considerable time to study the data. To reduce the time needed to detect outliers, it would be therefore relevant to use automated robust methods.

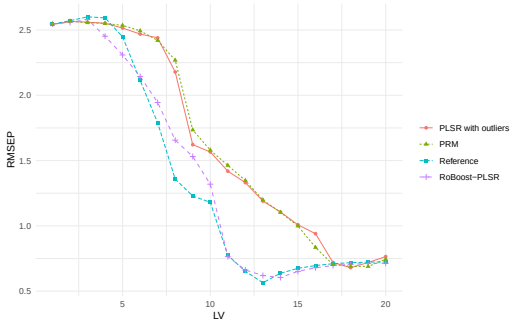


FIGURE 11: Evolution of the RMSEP as a function of latent variables for the reference, PLSR with outliers, PRM and RoBoost-PLSR for the dataset 3

Figure 11 shows that the PLSR with outliers (red curve) is less performant than the reference (blue curve). Indeed, the minimal RMSEP for the PLSR with outliers is $\simeq 0.7$ for 18 LVs whereas the minimal RMSEP of the reference is $\simeq 0.55$ for 13 LVs. This means that PLSR is sensitive to these outliers.

Figure 11 shows that the PRM performance curve is close to the PLSR with outliers curve. This means that PRM does not completely capture the nature of these outliers. It is fair to conjecture that PRM will perform much better for these data if based on a reweighting scheme that accounts for the residuals in the X-space as well

Figure 11 shows that the RoBoost-PLSR curve reaches a minimum error with 14 LVs, which is close to the reference. RoBoost-PLSR has a behaviour very similar to that of the reference. The minimum RMSEP of RoBoost-PLSR curve (14 LVs) is higher than the minimum RMSEP of the reference (13 LVs). This means that RoBoost-PLSR attributes a 0 weights to the outliers but also to some inliers. This leads to an increase in the number of LVs for a higher minimum RMSEP than the minimum RMSEP of the reference.

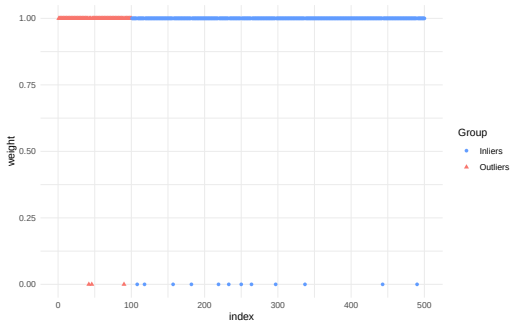


FIGURE 12: Repartition of the weights attributed to outliers (red) and inliers (blue) during the calibration of PRM for 18 latent variables

Figure 12 does not show a clear separation between the majority of outlier weights and inlier weights with a 18 LVs PRM model. This is due to the fact that outliers are not detected by PRM. This limitation of PRM

could be explained by the absence of X-residuals in the computation of weights. This also could be explained by the fact that outliers are weighted using a model with a predefined number of LVs.

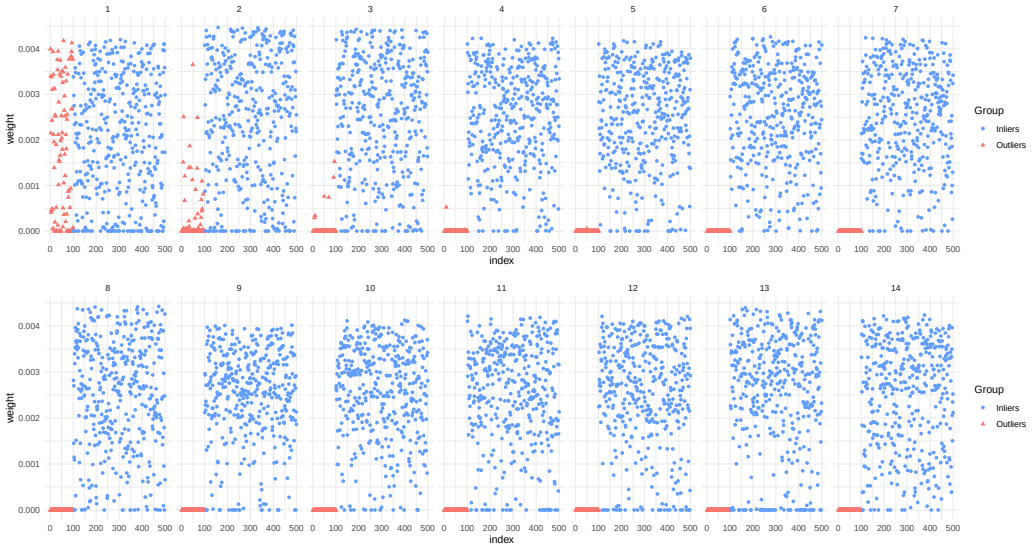


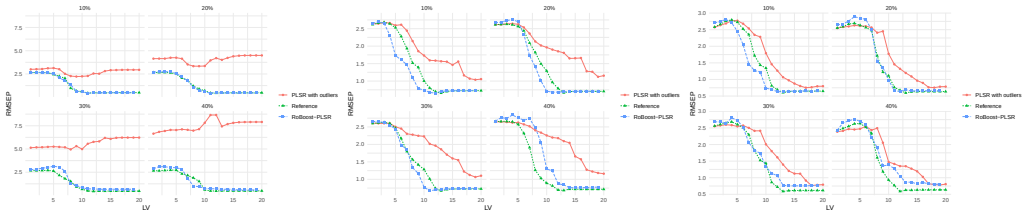
FIGURE 13: Repartition of the weights attributed to outliers (red) and inliers (blue) during the calibration of RoBoost-PLSR over 14 latent variables

Figure 13 shows the weights assigned to outliers and inliers by RoBoost-PLSR. It shows that RoBoost-PLSR begins to assign 0 weights to the outliers from the 3^{rd} LV. RoBoost-PLSR also attributes very low weights to a significant number of inliers while some outliers are attributed higher weights along the three first LVs. This means that some *a priori* informative samples are not necessarily favourable or even relevant for some LVs. It also means that outliers are not necessarily detrimental for the determination of all LVs. For example, the first LV can often be assimilated to baselines.

In these cases, outliers sharing a similar baseline are not detrimental while inliers with minor baseline shifts can be detrimental.

RoBoost-PLSR seems to be able to taking into account the variability of the beneficial samples and even sometimes the non-abnormal properties of outliers.

4.4. Influence of the proportion of outliers within calibration



(a) simulation1 (b) simulation2 (c) simulation3

FIGURE 14: RMSEP depending on the proportion of outliers in the calibration set

Figure 14 shows the prediction performances obtained with proportions of outliers varying between 10% and 40% for PLSR and RoBoost-PLSR. Figure 14 shows that the proportion of outliers affects the PLSR performance for each simulation. The higher the proportion of outliers, the lower the quality of outlier prediction.

Figure 14a shows that the proportion of outliers does not affect the performance of RoBoost-PLSR until 30% of outliers. This is due to the fact that the Y-distributions of the two groups of samples are differentiable (see Figure 1) and therefore the separation between outliers and inliers is easy. However, with 40% outliers, RoBoost-PLSR methods can not produce the same result as the PLSR method without outliers. Indeed, when the

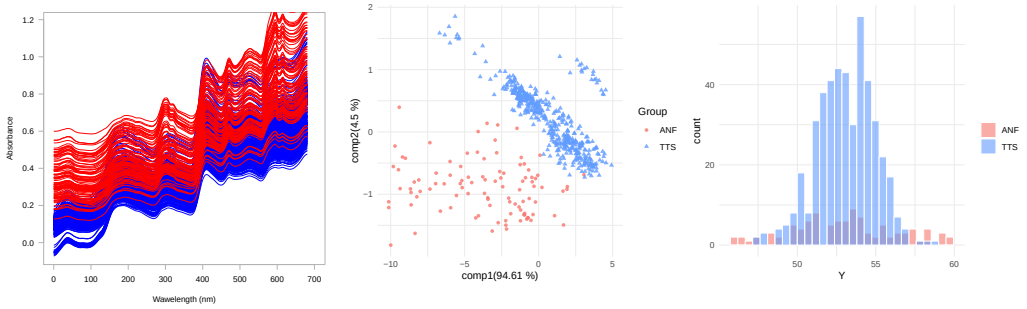
proportion of outliers is close to the proportion of inliers, it becomes really difficult to focus the model on the main mass. Despite this, RoBoost-PLSR method has a stable curve and is generally close to reference even with 40% outliers.

Figure 14b shows that the proportion of outliers has little effect on the performance of the RoBoost-PLSR method. This means that the outliers are correctly detected by RoBoost-PLSR even when the proportion of outliers is close to the inliers proportion.

Figure 14c shows that the proportion of outliers has little effect on the performance of the RoBoost-PLSR method until 30%. For 40%, the curve of RoBoost-PLSR is between the PLSR witout outliers and the PLSR with outliers. This means that the method detects some but not all outliers. In conclusion, the RoBoost-PLSR method supports these three types of outliers up to 30% with prediction performances approaching the reference.

4.5. Real dataset and application : prediction of protein content.

The present section intends to deal with real agronomic data, with the example of a common animal nutrition application : the prediction of the protein content of feed materials and the presence of incorrectly categorised samples. In this database the samples resulting from animal bonemeal (noted ANF) represent the outliers polluting the regular soyabean cakes (noted TTS).



(a) (b) (c)

FIGURE 15: Properties of TTS and ANF proteins. (a) Spectra, (b) PCA projection of spectra and (c) distributions of Y-responses. TTS Inliers are represented in blue, while ANF outliers are in red.

In the proposed application, the proteins contained in ANF outliers, present spectral similarities with soya proteins (TTS). Therefore, even in minor proportions, these outliers can alter PLSR models. Figure 15 shows the similarities between the outliers (ANF) and the inliers (TTS). Spectra overlap so that the two populations are indistinguishable. There is supposedly an overall difference in baselines, yet insufficient to separate the data into consistent clusters. Figure 15 presents the data projected on the first two PCA axes. On the basis of this projection, it is also difficult to attest the presence of two distinct groups. With the beforehand knowledge regarding the affiliation of samples, inliers (in blue) seem to follow a precise trajectory while the outliers (in red) form a sparse cloud. However without this knowledge, it would not represent a reliable clustering of data, all the more since inliers present a second marginal distribution, parallel to the main one. Therefore in practice it is not trivial to discard unknown outliers, accidentally introduced

in a calibration set. In terms of response, (see fig 15), the outliers also present a similar distribution (in red) overlapping the distribution of inliers responses (in blue). It is often the case in food applications where different raw materials can present comparable nutrient contents.

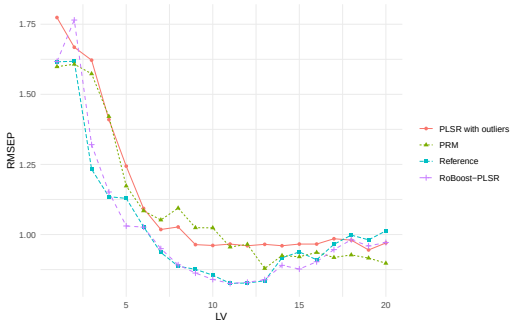


FIGURE 16: Evolution of the RMSEP as a function of latent variables for the reference, PLSR with outliers, PRM and RoBoost-PLSR for the real set

Figure 16 shows that the PLSR with ANF samples (red curve) is less performant than the reference (PLSR calibrated without ANF samples, blue curve). Indeed, the minimal RMSEP for the PLSR with ANF samples is $\simeq$ 0.95 for 19 LVs whereas the minimal RMSEP of the reference is $\simeq$ 0.83 for 11 LVs. This means that PLSR is sensitive to these ANF samples.

Figure 16 shows that the PRM performance curve is between the PLSR with and without ANF samples curves. At best, it achieves an RMSEP equal to 0.87 for 13 LVs.

Figure 16 shows that the RoBoost-PLSR curve reaches a minimum error with 11 LVs, that is the same as the reference (RMSEP = 0.83). RoBoost-PLSR has a behaviour very similar to that of the reference. This means that RoBoost-PLSR attributes a 0 weights to the ANF samples but also to some

TTS samples.

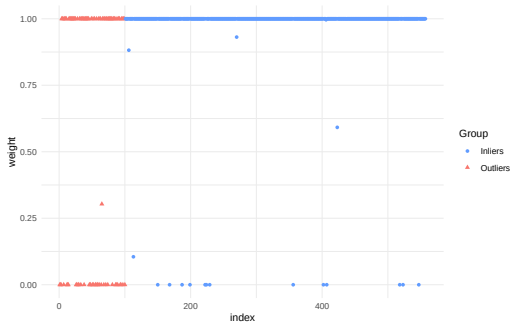


FIGURE 17: Repartition of the weights attributed to outliers (red) and inliers (blue) during the calibration of PRM for 13 latent variables with the best weights constant setting

Figure 17 presents the repartition of the weights attributed within the calibration of PRM. ANF samples weights are not distinguished from the TTS samples weights. This result explains the poor prediction performances of PRM on this real dataset.

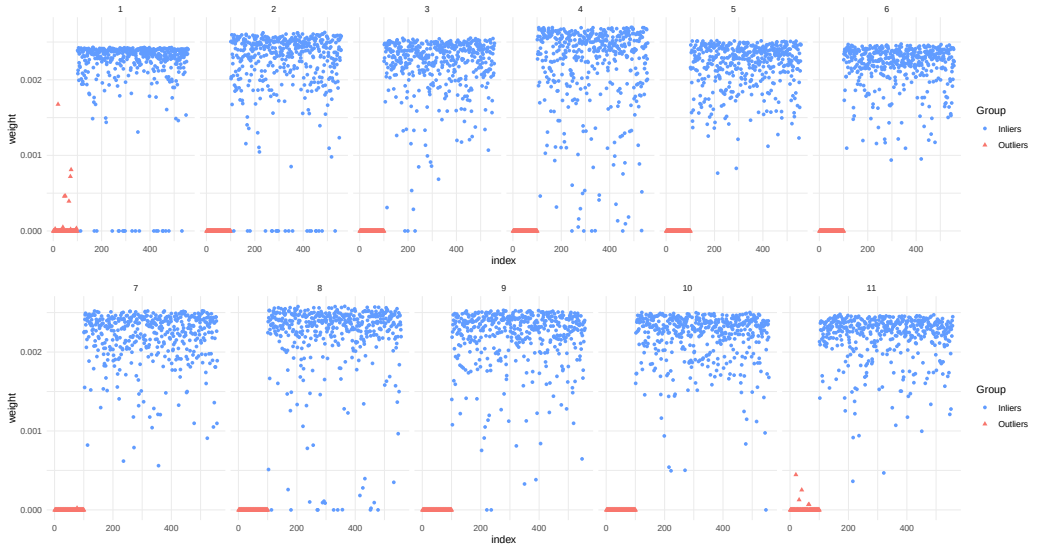


FIGURE 18: Repartition of the weights attributed to outliers (red) and inliers (blue) during the calibration of RoBoost-PLSR over 11 latent variables with the best weights constants settings

Figure 18 presents the weights attributed to samples for the eight considered LVs within the calibration of RoBoost-PLSR. From the first LV, most ANF samples are assigned null weights along with a few TTS samples. From the second latent variable, all ANF samples weights are close to 0.

5. Conclusion & Perspectives

This article showed the potential of the RoBoost-PLSR method. This method offers a relevant solution for the calibration of PLSR models in the presence of various types of outliers. At this stage the proposed algorithm is mainly based on weighting strategies within a of series unidimensional PLS. The method is designed to detect outliers within the calibration, through iterations where dissimilarity measurements take into account the hypothesis

of robust linear models. The four evaluated applications demonstrate that the introduced outliers are predominantly detected and discarded from the model. As a result, RoBoost-PLSR is able to attain performance on par with the reference.

One dataset was found to be particularly difficult to process by the PRM method. This dataset was the one with X-outliers. It would be interesting to integrate within PRM a weighting criterion related to the X-residuals (as in RoBoost-PLSR).This would enable to observe the benefit of considering X-residuals compared to the benefit of estimating weights based on the score space alone. Eventually, RoBoost-PLSR proved to be a promising framework to deal with various practical issues. Further studies should be carried out in practical context for more diverse applications ; Including smaller datasets, where it is yet undetermined if the estimation of weight criteria is still relevant / functional. Indeed, the observations carried out in this paper are based on large learning databases. This implies that it is potentially possible to apply stricter weights without degrading the prediction quality of the method.

To this end, the method requires further studies on the following issues : Firstly, a comprehensive study regarding the weight functions and their optimisation should be carried out, in order to better adjust the models. Indeed, the parametrisation of RoBoost-PLSR can be a difficult task (three parameters have to be optimised for each LV). In this paper, the constants were fixed for all the latent variables. However, it would be relevant to define specific constants optimised for each latent variable. It is not conceivable to manually optimise constants for each latent variable. Secondly, in real applications, outliers can be present both in the calibration and validation

sets. In this paper, the validation sets are not contaminated. To obtain a fully operational method, it should be completed with the development of a metric intended to determine the consistency of unknown samples with the model. This would enable to predict datasets containing potential outliers, and then only process data for which the model is calibrated for.

Thirdly, the interpretation of the RoBoost-PLSR model is complex. Indeed, the proposed algorithm does not provide an estimation of regression coefficients unlike approaches such as PRM. In order to allow better interpretability, it would be essential in future work to propose an algorithm that enables an estimation of the regression coefficients. Fourthly, the initial estimators (centering) and the estimation of the sample weights can be corrupted. In RoBoost-PLS, data are centred about the arithmetic mean. It is well known that the arithmetic mean is non-robust and can thereby provide distorted starting values for the algorithm. A potential solution is to replace these estimators with robust alternatives (*e.g.* robust multivariate location). As for the bisquare weight function, it uses the (coordinatewise) median, which can lie outside the convex hull of the data than breakdown. It would be relevant to consider other weight functions that take into account these aspects. Fifthly, the cross-validation of robust methods, for instance for the optimisation of hyperparameters, is not a straightforward procedure. In this paper, this limitation has been overcome by optimising and studying the behaviour of the methods on an unpolluted validation set. In future work it will be interesting to develop tools to cross-validate the RoBoost-PLSR method in order to allow the development of this method on real cases.

Finally, the robust multivariate methods have proven their reliable

predictive quality in classification issues [29]. RoBoost-PLSR could also be adapted to classification problems. This implies that RoBoost-PLSR should be adapted to multidimensional Y.

Acknowledgements

This work was supported by the French National Research Agency under the Investments for the Future Program, referred as ANR-16-CONV-0004. Authors care to thank Vincent Baeten and Pierre Dardenne from the CRA-W (Agronomic Research Centre of Wallonia, Belgium) for providing the real dataset used in this article. Authors wish to adress a special thanks to Gilbert Saporta from the CNAM for his thorough proofreading of the paper and his precious advice.

Referencse

[1] S. Wold, H. Martens, H. Wold, The multivariate calibration problem in chemistry solved by the pls method, in : B. Kågström, A. Ruhe (Eds.), Matrix Pencils, Springer Berlin Heidelberg, Berlin, Heidelberg, 1983, pp. 286–293.

[2] S. Serneels, C. Croux, P. J. Van Espen, Influence properties of partial least squares regression, Chemometrics and Intelligent Laboratory Systems 71 (1) (2004) 13–20. doi:https://doi.org/10.1016/j.chemolab.2003.10.009.

[3] P. Filzmoser, S. Höppner, I. Ortner, S. Serneels, T. Verdonck, Cellwise robust M regression, Computational Statistics & Data Analysis 147 (2020) 106944. doi:10.1016/j.csda.2020.106944.

[4] M. Griep, I. Wakeling, P. Vankeerberghen, D. Massart, Comparison of semirobust and robust partial least squares procedures, Chemometrics and Intelligent Laboratory Systems 29 (1) (1995) 37–50. doi:10.1016/0169-7439(95)80078-N.

[5] I. Stanimirova, S. Serneels, P. J. Van Espen, B. Walczak, How to construct a multiple regression model for data with missing elements and outlying objects, Analytica Chimica Acta 581 (2) (2007) 324–332. doi:10.1016/j.aca.2006.08.014.

[6] R. J. Pell, Multiple outlier detection for multivariate calibration using robust statistical techniques, Chemometrics and Intelligent Laboratory Systems 52 (1) (2000) 87–104. doi:10.1016/S0169-7439(00)00082-4.

[7] J. A. Gil, R. Romera, On robust partial least squares (PLS) methods, Journal of Chemometrics 12 (6) (1998) 365–378. doi:10.1002/(SICI)1099-128X(199811/12)12:6<365::AID-CEM519>3.0.CO;2-G.

[8] S. Acitas, P. Filzmoser, B. Senoglu, A new partial robust adaptive modified maximum likelihood estimator, Chemometrics and Intelligent Laboratory Systems 204 (2020) 104068. doi:10.1016/j.chemolab.2020.104068.

[9] J. González, D. Peña, R. Romera, A robust partial least squares regression method with applications, Journal of Chemometrics 23 (2) (2009) 78–90. doi:10.1002/cem.1195.

[10] I. N. Wakeling, H. J. H. Macfie, A robust PLS procedure, Journal of Chemometrics 6 (4) (1992) 189–198. doi:10.1002/cem.1180060404.

[11] J. Peng, S. Peng, Y. Hu, Partial least squares and random sample consensus in outlier detection, Analytica Chimica Acta 719 (2012) 24–29. doi:10.1016/j.aca.2011.12.058.

[12] P. Filzmoser, R. Maronna, M. Werner, Outlier identification in high dimensions, Computational Statistics & Data Analysis 52 (3) (2008) 1694–1711. doi:10.1016/j.csda.2007.05.018.

[13] M. Hubert, K. V. Branden, Robust methods for partial least squares regression, Journal of Chemometrics 17 (10) (2003) 537–549. doi:10.1002/cem.822.

[14] U. Kruger, Y. Zhou, X. Wang, D. Rooney, J. Thompson, Robust partial least squares regression : Part II, new algorithm and benchmark studies, Journal of Chemometrics 22 (1) (2008) 14–22, _eprint : https ://onlinelibrary.wiley.com/doi/pdf/10.1002/cem.1095. doi:10.1002/cem.1095.

[15] I. Hoffmann, S. Serneels, P. Filzmoser, C. Croux, Sparse partial robust M regression, Chemometrics and Intelligent Laboratory Systems 149 (2015) 50–59. doi:10.1016/j.chemolab.2015.09.019.

[16] P. Filzmoser, V. Todorov, Review of robust multivariate statistical methods in high dimension, Analytica Chimica Acta 705 (1-2) (2011) 2–14. doi:10.1016/j.aca.2011.03.055.

[17] S. F. Møller, J. v. Frese, R. Bro, Robust methods for multivariate data analysis, Journal of Chemometrics 19 (10) (2005) 549–563, _eprint :

https ://onlinelibrary.wiley.com/doi/pdf/10.1002/cem.962. doi:10.1002/cem.962.

[18] S. Serneels, C. Croux, P. Filzmoser, P. J. Van Espen, Partial robust M-regression, Chemometrics and Intelligent Laboratory Systems 79 (1) (2005) 55–64. doi:10.1016/j.chemolab.2005.04.007.

[19] J. Betzin, Pls-regression in the boosting framework, in : M. Vilares, M. Tenenhaus, P. Coelho, A. Morineau, V. Esposito Vinzi (Eds.), PLS and Related Methods, DECISIA, Levallois Perret, 2003, pp. 261–269.

[20] A.-L. Boulesteix, PLS Dimension Reduction for Classification with Microarray Data, Statistical Applications in Genetics and Molecular Biology 3 (1), publisher : De Gruyter Section : Statistical Applications in Genetics and Molecular Biology (Nov. 2004). doi:10.2202/1544-6115.1075.

[21] X. Shao, X. Bian, W. Cai, An improved boosting partial least squares method for near-infrared spectroscopic quantitative analysis, Analytica Chimica Acta 666 (1-2) (2010) 32–37. doi:10.1016/j.aca.2010.03.036.

[22] R. Rosipal, N. Krämer, Overview and Recent Advances in Partial Least Squares, in : C. Saunders, M. Grobelnik, S. Gunn, J. Shawe-Taylor (Eds.), Subspace, Latent Structure and Feature Selection, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2006, pp. 34–51. doi:10.1007/11752790_2.

[23] M. H. Zhang, Q. S. Xu, D. L. Massart, Boosting partial least squares, Analytical Chemistry 77 (5) (2005) 1423–1431, pMID : 15732927. doi:10.1021/ac048561m.

[24] D. J. Cummins, C. W. Andrews, Iteratively reweighted partial least squares : A performance analysis by monte carlo simulation, Journal of Chemometrics 9 (6) (1995) 489–507. doi:10.1002/cem.1180090607.

[25] S. Schaal, C. G. Atkeson, S. Vijayakumar, Scalable Techniques from Nonparametric Statistics for Real Time Robot Learning, Applied Intelligence 17 (1) (2002) 49–60. doi:10.1023/A:1015727715131.

[26] W. S. Cleveland, Robust Locally Weighted Regression and Smoothing Scatterplots, Journal of the American Statistical Association (1979) 9.

[27] M. Metz, A. Biancolillo, M. Lesnoff, J.-M. Roger, A note on spectral data simulation, Chemometrics and Intelligent Laboratory Systems 200 (2020) 103979. doi:10.1016/j.chemolab.2020.103979.

[28] M. Lesnoff, M. Metz, J.-M. Roger, Comparison of locally weighted PLS strategies for regression and discrimination on agronomic NIR data, Journal of Chemometrics 34 (5) (2020) e3209. doi:10.1002/cem.3209.

[29] I. Hoffmann, P. Filzmoser, S. Serneels, K. Varmuza, Sparse and robust PLS for binary classification, Journal of Chemometrics 30 (4) (2016) 153–162. doi:10.1002/cem.2775.

Appendix

TABLE .1: The different choices in the simulation 1

	Inliers	Outliers
$\mathbf{P}_u$	Pure spectrum of glucose	
$\mathbf{T}_u$	Folded-normal distribution	
$\mathbf{P}_d$	Pure spectrum of water Pure spectrum of ethanol Spectrum of water-ethanol Interaction 10 Artificial spectra	
$\mathbf{T}_d$	Folded-normal distribution Folded-normal distribution Product between T_{water} and $T_{ethanol}$ Folded-normal distribution	
$\mathbf{E}$	Gaussian distribution	
f	$Y = 10 * T_{glucose}$	$\mathbf{Y} = -\mathbf{5} * \mathbf{T_{glucose}}$
$\mathbf{F}$	Gaussian distribution	

u define useful space, d define detrimental space, E define the spectral noise and F the response noise.

TABLE .2: The different choices in the simulation 2

	Inliers	Outliers
$\mathbf{P}_u$	Pure spectrum of glucose	
$\mathbf{T}_u$	Folded-normal distribution	
$\mathbf{P}_d$	Pure spectrum of water Pure spectrum of ethanol Spectrum of water-ethanol Interaction 10 Artificial spectra	Pure spectrum of water Pure spectrum of ethanol Spectrum of water-ethanol Interaction 10 Artificial spectra **100 Artificial spectra**
$\mathbf{T}_d$	Folded-normal distribution Folded-normal distribution Product between T_{water} and $T_{ethanol}$ Folded-normal distribution	Folded-normal distribution Folded-normal distribution Product between T_{water} and $T_{ethanol}$ Folded-normal distribution **Folded-normal distribution**
$\mathbf{E}$	Gaussian distribution	
f	$Y = 10 * T_{glucose}$	
$\mathbf{F}$	Gaussian distribution	

u define useful space, d define detrimental space, E define the spectral noise and F the response noise.

TABLE .3: The different choices in the simulation 3

	Inliers	Outliers
$\mathbf{P}_u$	Pure spectrum of glucose	
$\mathbf{T}_u$	Folded-normal distribution	
$\mathbf{P}_d$	Pure spectrum of water Pure spectrum of ethanol Spectrum of water-ethanol Interaction 10 Artificial spectra	Pure spectrum of water Pure spectrum of ethanol Spectrum of water-ethanol Interaction 10 Artificial spectra **10 Artificial spectra**
$\mathbf{T}_d$	Folded-normal distribution Folded-normal distribution Product between T_{water} and $T_{ethanol}$ Folded-normal distribution	Folded-normal distribution Folded-normal distribution Product between T_{water} and $T_{ethanol}$ Folded-normal distribution **Folded-normal distribution**
$\mathbf{E}$	Gaussian distribution	
f	$Y = 10 * T_{glucose}$	
$\mathbf{F}$	Gaussian distribution	

u define useful space, d define detrimental space, E define the spectral noise and F the response noise.