

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)

Preprint submitted to Analytica Chimica Acta

28 juin 2021

35 without any outlier.

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

37 1. Introduction

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

59 In this article, the attention is focused on methods intended for the
 60 calibration of PLS1 models in presence of potential outliers. This means
 61 that the methods weight the samples through the PLSR in order to reduce
 62 the impact of outliers on model calibration. In that sense, only a few robust
 63 methods were proposed along with an available algorithm.
 64 One of the first methods was proposed in [10]. This method carries out a
 65 robust least square regression for each explanatory variable. This means
 66 that the method considers independent variables with this procedure. This
 67 aspect was particularly argued in [17] because this process does not capture
 68 the multidimensional aspect of outliers.
 69 To address to this problem, [18], developed the Partial Robust M-regression
 70 (PRM) method. PRM is frequently studied and used in chemometrics. PRM
 71 is based on the NIPALS algorithm trained on the iteratively reweighted
 72 matrices (representing the explanatory variables and responses). PRM
 73 consists of weighting the samples on the basis of a PLSR model with a
 74 predefined number of latent variables (LVs). This means that the weights
 75 are defined for a specific model (*i.e.* PLSR with K latent variables). To
 76 determine the $k < K$ models, weights must be specifically recomputed for
 77 each given k , as opposed to PLSR where each 1 to K LVs model can
 78 be deduced at once from a K model. In PRM, an outlier is defined by a
 79 combination of the leverage estimation (*i.e.* the Euclidean distance between
 80 scores and the median of scores) and Y-residuals. A limitation of this
 81 method, is that outliers are detected using a PLSR model with a number of
 82 latent variables that is defined beforehand. In [10], this limitation is lifted
 83 by weighting the samples independently of the number of latent variables.

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

91 The link between PLS and gradient boosting has already been studied
92 and resulted in implementations for the processing of chemical data [19–
93 23] Essentially, these approaches use numerous weak learners, computed
94 sequentially from different sub-samples. Each new weak learner is computed
95 from the previous ones using a loss function. Finally, the weak learners are all
96 combined in a weight function according to their predictive potential. As for
97 the RoBoost-PLSR framework, it proposes to apply the basic idea of gradient
98 boosting : *i.e.* combining an ensemble of weak learners. The weak learners
99 are defined here as weighted one-latent variable PLSR models. The weights
100 are defined iteratively in order to reduce the contribution of outliers on the
101 calculated model. The weak learners are then combined using an unweighted
102 sum of the predictions of each weak learner.

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

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

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

115 **2. Theoretical background of the RoBoost-PLSR method**

116 *2.1. Notations*

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

124 *2.2. Principle of the method*

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

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

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

- 137 — X -residuals
- 138 — Y -residuals
- 139 — Leverage

140 2.3. Algorithm

141 Let \mathbf{X} be an $[n \times m]$ matrix containing n samples described by m variables.

142 Let \mathbf{y} be a response vector containing n samples. In this article \mathbf{y} is always
143 considered as a vector, *i.e.* the response is univariate.

144 For a definite number of K latent variables, the algorithm proceeds as
145 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} :

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

4: Derive the weighted means :

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

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

5: Center the data :

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

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

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

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

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 :

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

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

9: Derive the Y-residuals (\mathbf{f}), X-residuals (\mathbf{E}), leverage estimation (\mathbf{l})

corresponding to the current k^{th} latent variable :

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

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

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

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

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

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

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

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

With s_α , s_β , s_γ being respectively the median of $\{\|\mathbf{e}_i\|\}_n$, $\{f_i\}_n$ 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 :

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

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

12: while $k < K$

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

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

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

End Calibration

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

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

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 :

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

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

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

146 2.4. Method properties

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

150 Firstly, estimating the weights of samples independently for each latent
151 variable provides a simpler estimation of leverage points. Indeed, in usual
152 robust PLSR algorithms, leverage is computed either thanks to Euclidean
153 or Mahalanobis distances between the scores and the centre of the model.

154 In high dimensional spaces (numerous LVs), this estimation is not so trivial.
155 As a matter of fact, in the case of a Euclidean distance, the latest LVs have
156 only a minor contribution to the leverage value. This is naturally due to
157 the decreasing magnitude of scores. Nevertheless, the predictive potential
158 of these latest LVs is not necessarily lesser. In the case of a Mahalanobis
159 distance, the contributions of all LVs become equal in the computation of
160 the leverage value. This can be equally detrimental, since the predictive

161 potentials of the LVs are most oftenly uneven.

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

166 Thirdly, the method does not provide regression coefficients. Contrary to
167 other robust methods such as PRM, in this case, it is not trivial to compute
168 them. Indeed, the proposed algorithm for RoBoost-PLSR does not allow an
169 estimation of the rotation matrix \mathbf{R} . Models can nevertheless be interpreted
170 by analysing the loadings, although it is less convenient. Indeed, it is possible
171 to observe the loadings and derive the most influential variables within each
172 1LV model. However, unlike in conventional PLSR, it is not possible yet to
173 determine the relative influences of variables at the scale of the whole K-LV
174 model

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

179 **3. Material and methods**

180 *3.1. Data and software*

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

185 was developed on the basis of “[rnirs](#)” functions. The functions and data
186 associated with RoBoost-PLSR are available on Github “[RoBoost-PLSR](#)”.
187 PRM was implement with the “prms” function available in the “[sprm](#)”
188 package.

189 3.2. Simulated Data

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

195 The simulations were based on a combination of pure artificial spectra and
196 controlled noises. The aim of each simulation was to reproduce the common
197 external disturbances that can occur when calibrating a predictive model. It
198 consisted of adding to the dataset an additional set of predefined outliers that
199 have a negative effect on the performance of the models. The first simulation
200 introduced pure Y outliers. The second simulation introduced contaminant
201 induced outlier *i.e.* X -outliers occurring when an external substance pollutes
202 the calibration samples. These individuals are strong outliers because they
203 can be easily distinguished from inliers (*e.g.* by a spectra plot). The third
204 simulation introduced slight X -outliers. For all simulations, 900 inliers and
205 100 outliers were simulated. Descriptions of the simulation are available in
206 the [appendix](#) in table form. The differences between simulated inliers and
207 outliers are highlighted in bold in the tables.

208 3.3. Real dataset

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

218 3.4. Evaluation strategies

219 The purpose is to evaluate the behaviour of the newly introduced
220 RoBoost-PLSR methods in presence of outliers during calibration. The
221 calibrated model is then evaluated on a validation set. The reference against
222 which all models were compared was a PLSR calibrated on a dataset without
223 outliers (and will be designated as such). Roboost-PLSR was evaluated and
224 compared with two standard regression algorithms : PLSR and PRM.
225 In the case of the simulations, the weight parameters of PRM and
226 RoBoost-PLSR were optimised according to the validation set. Only the
227 results of the optimal (*i.e.* the parameters that provide the minimum value
228 of RMSEP) parameters of RoBoost-PLSR and PRM were presented in the
229 following section. The calibration sets were generated from 500 samples
230 (400 inliers and 100 outliers). The resulting models were studied with
231 validation sets containing 500 inliers. The prediction performance of the
232 RoBoost-PLSR method was studied also as a function of the proportion of

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

236

237 In the case of the real dataset, the weights parameters of PRM (using
238 the Hampel function) and RoBoost-PLSR were optimised according to the
239 validation set. Only the results of the optimal parameters of RoBoost-PLSR
240 and PRM were presented in the following section. The calibration set was
241 composed of 457 soybean protein (TTS) samples and 100 animal-protein
242 (ANF) samples that represent the outliers. The validation was conducted on
243 50 additional samples of soybean and results were evaluated through Root
244 Mean Square Error of Prediction (RMSEP).

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

248 4. Results and discussion

249 4.1. Simulation 1 : pure Y -outliers

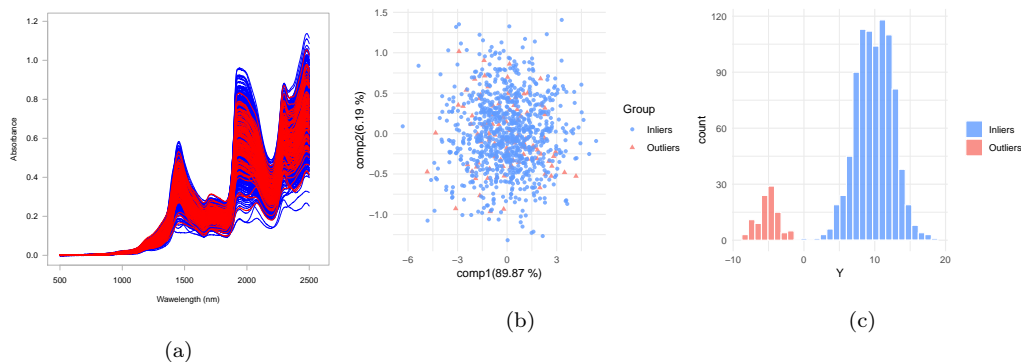


FIGURE 1: Simulated dataset 1. (a) Spectra, (b) PCA projection of spectra and (c) distributions of Y -responses. Inliers are represented in blue, while outliers are in red.

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

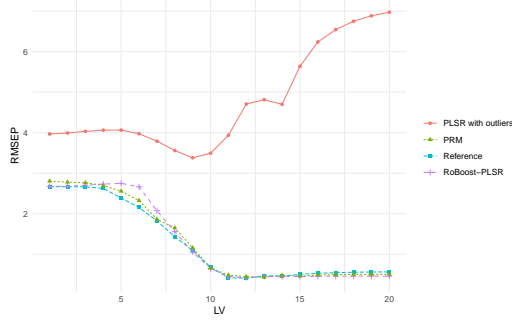


FIGURE 2: Evolution of the RMSEP as a function of latent variables for the reference, PLSR without outliers, PRM and RoBoost-PLSR for the dataset 1

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

275 number of LVs.

276

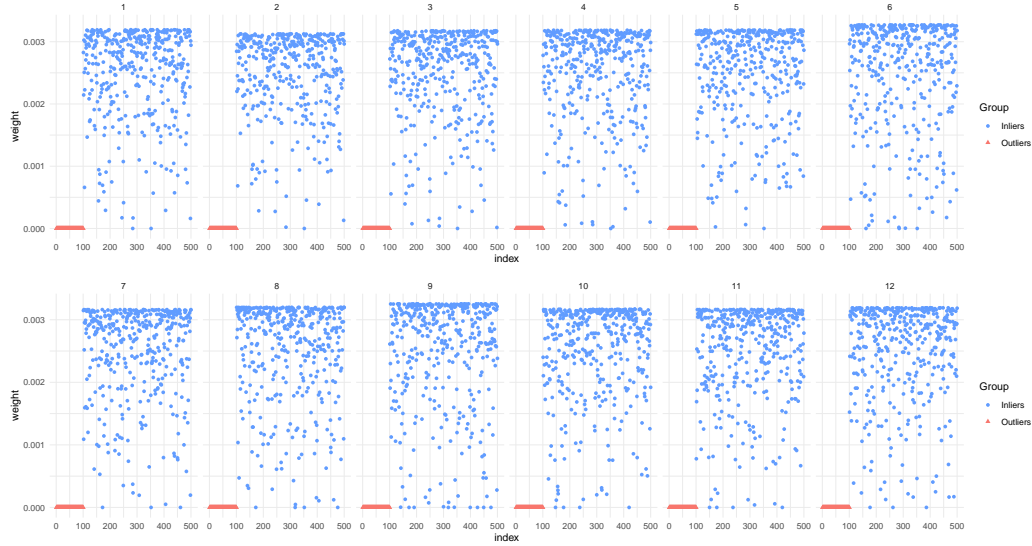


FIGURE 3: Repartition of the weights attributed to outliers (red) and inliers (blue) during the calibration of RoBoost-PLSR over 12 latent variables

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

287 are considered relevant from RoBoost-PLSR.

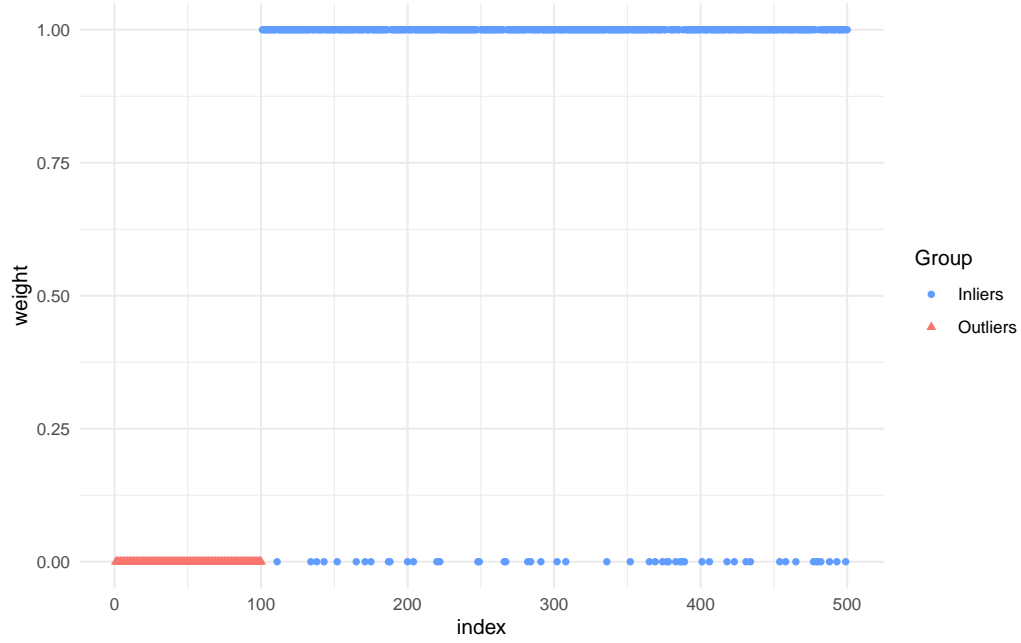


FIGURE 4: Repartition of the weights attributed to outliers (red) and inliers (blue) during the calibration of PRM for respectively 13 LVs

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 weights 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

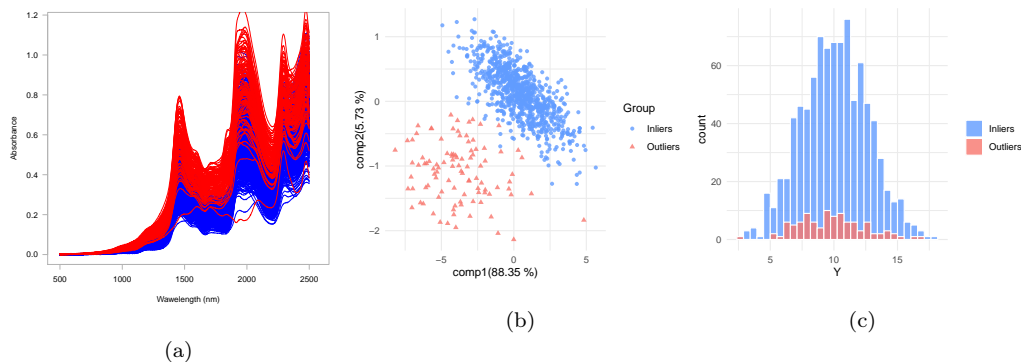


FIGURE 5: Simulated dataset 2. (a) Spectra, (b) PCA projection of spectra and (c) distributions of Y -responses. Inliers are represented in blue, while outliers are in red.

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}).

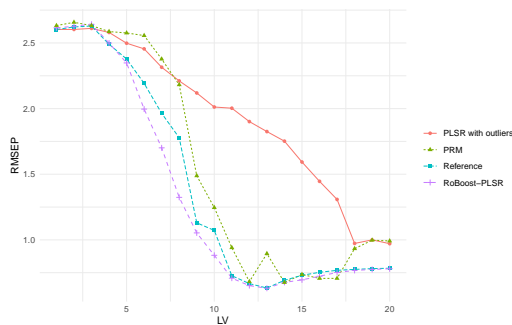


FIGURE 6: Evolution of the RMSEP as a function of latent variables for the reference, PLSR with outliers, PRM and RoBoost-PLSR for the dataset 2

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

326 to the reference. RoBoost-PLSR has a behaviour similar to the reference with
 327 the minimum RMSEP at 12 LVs. This means that RoBoost-PLSR attributes
 328 very low weights to the outliers but also to some inliers.
 329 Both PRM and RoBoost-PLSR prove to be robust to “contaminant induced”
 330 which are simple X-outliers. RoBoost-PLSR seems to perform well and have
 331 the same behaviour as the reference.

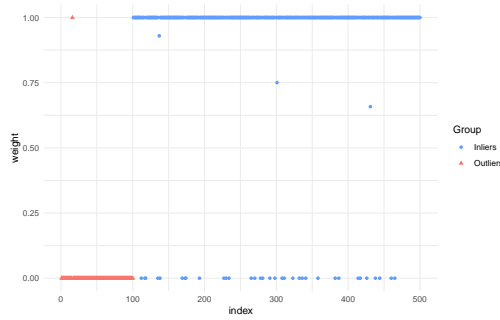


FIGURE 7: Repartition of the weights attributed to outliers (red) and inliers (blue) during the calibration of PRM for respectively 12 LVs

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

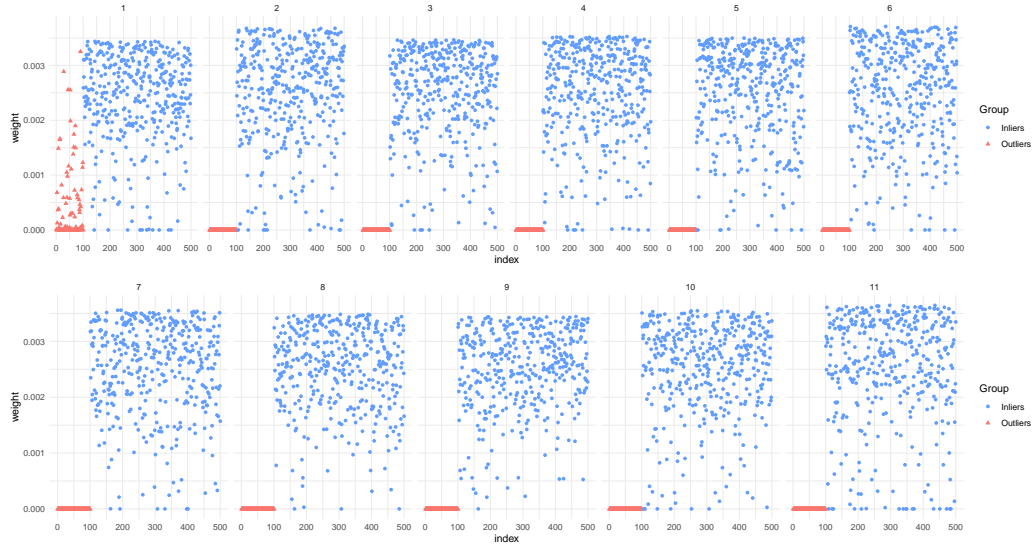


FIGURE 8: Repartition of the weights attributed to outliers (red) and inliers (blue) during the calibration of RoBoost-PLSR over 12 latent variables

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 2nd 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.

341 4.3. *Simulation 3 : X-outliers induced by microvariations of the measuring*
342 *environment*

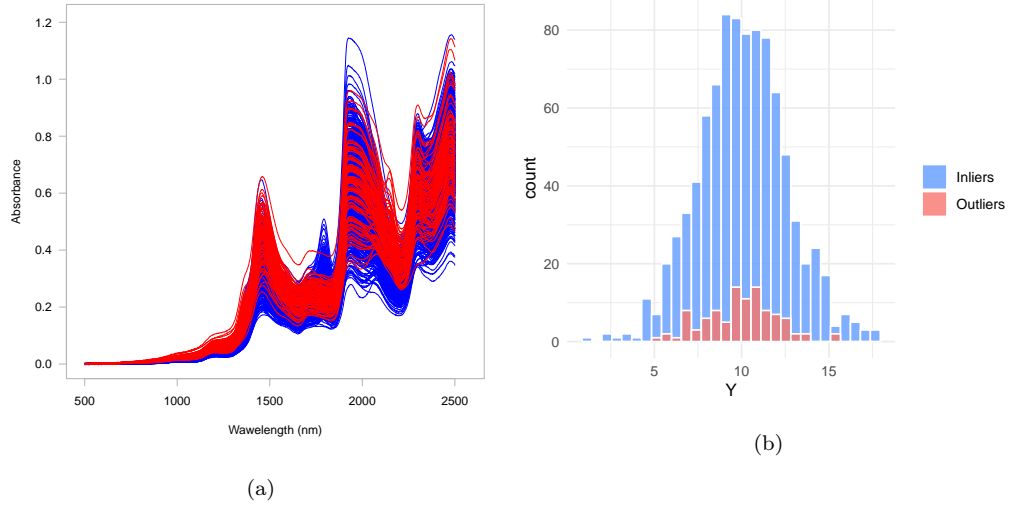


FIGURE 9: Simulated dataset 3. (a) Spectra, (b) distributions of Y-responses. Inliers are represented in blue, while outliers are in red.

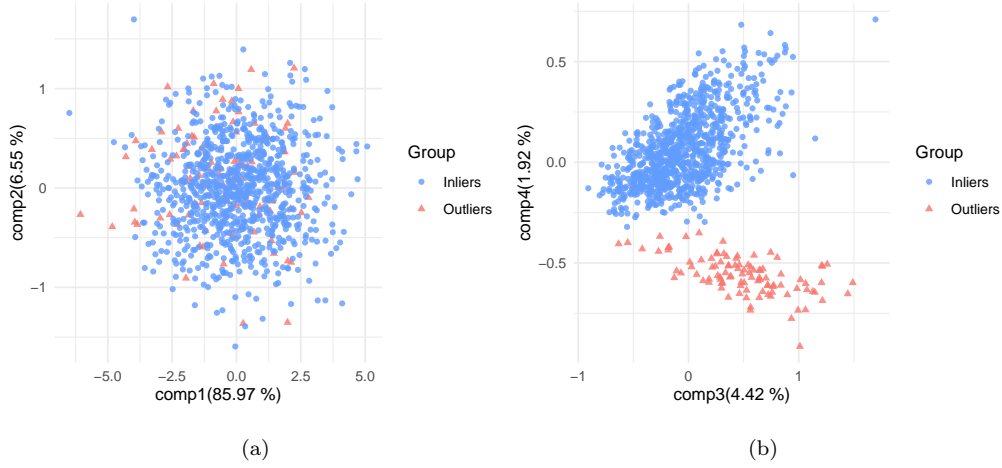


FIGURE 10: Simulated dataset 3. (a) PCA projection of spectra onto components 1 and 2, (b) PCA projection of spectra onto components 3 and 4. Inliers are represented in blue, while outliers are in red.

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

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

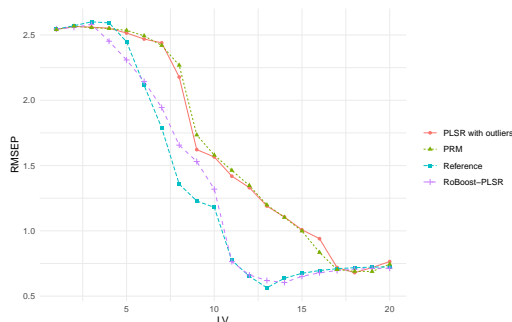


FIGURE 11: Evolution of the RMSEP as a function of latent variables for the reference, PLSR with outliers, PRM and RoBoost-PLSR for the dataset 3

367 Figure 11 shows that the PLSR with outliers (red curve) is less
 368 performant than the reference (blue curve). Indeed, the minimal RMSEP
 369 for the PLSR with outliers is $\simeq 0.7$ for 18 LVs whereas the minimal RMSEP
 370 of the reference is $\simeq 0.55$ for 13 LVs. This means that PLSR is sensitive to
 371 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.

384

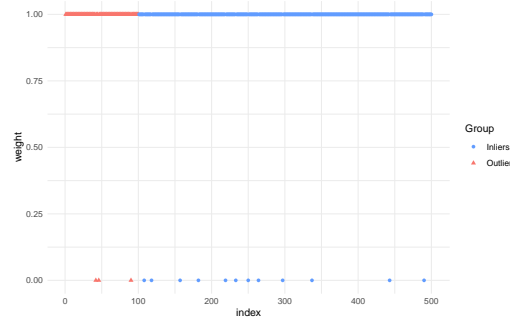


FIGURE 12: Repartition of the weights attributed to outliers (red) and inliers (blue) during the calibration of PRM for 18 latent variables

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.

391

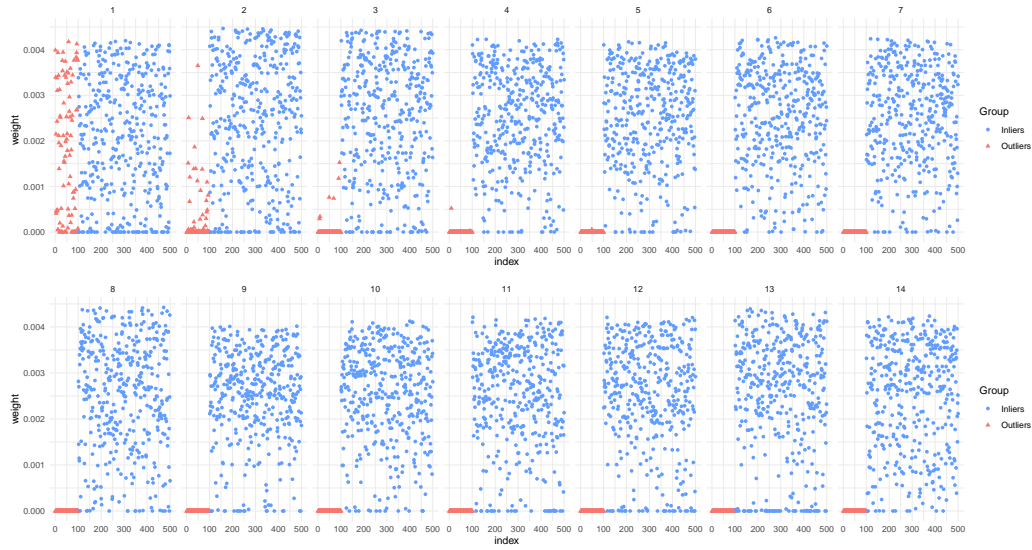


FIGURE 13: Repartition of the weights attributed to outliers (red) and inliers (blue) during the calibration of RoBoost-PLSR over 14 latent variables

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 3rd 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.

400 In these cases, outliers sharing a similar baseline are not detrimental while
 401 inliers with minor baseline shifts can be detrimental.
 402 RoBoost-PLSR seems to be able to taking into account the variability of
 403 the beneficial samples and even sometimes the non-abnormal properties of
 404 outliers.

405 4.4. Influence of the proportion of outliers within calibration

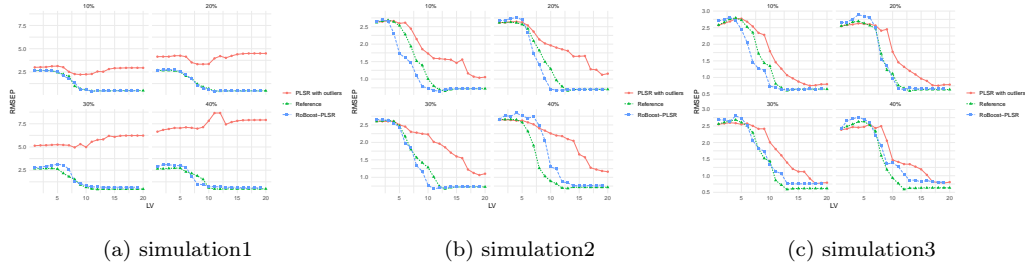


FIGURE 14: RMSEP depending on the proportion of outliers in the calibration set

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

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

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

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

425 Figure 14c shows that the proportion of outliers has little effect on the
426 performance of the RoBoost-PLSR method until 30%. For 40%, the curve
427 of RoBoost-PLSR is between the PLSR without outliers and the PLSR with
428 outliers. This means that the method detects some but not all outliers.
429 In conclusion, the RoBoost-PLSR method supports these three types of
430 outliers up to 30% with prediction performances approaching the reference.

431 4.5. Real dataset and application : prediction of protein content.

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

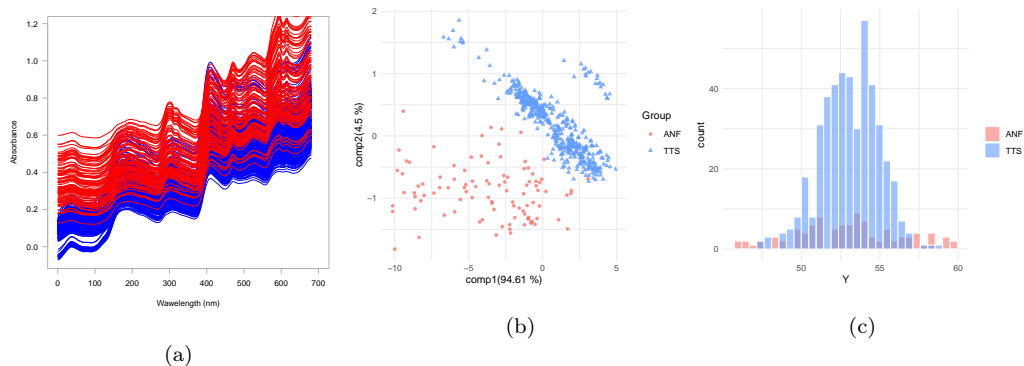


FIGURE 15: Properties of TTS and ANF proteins. (a) Spectra, (b) PCA projection of spectra and (c) distributions of Y-responses. TTS Inliers are represented in blue, while ANF outliers are in red.

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

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

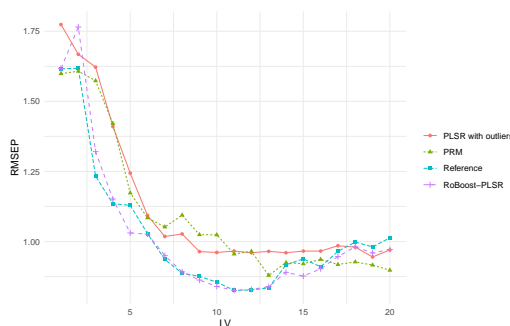


FIGURE 16: Evolution of the RMSEP as a function of latent variables for the reference, PLSR with outliers, PRM and RoBoost-PLSR for the real set

456 Figure 16 shows that the PLSR with ANF samples (red curve) is less
 457 performant than the reference (PLSR calibrated without ANF samples, blue
 458 curve). Indeed, the minimal RMSEP for the PLSR with ANF samples is \simeq
 459 0.95 for 19 LVs whereas the minimal RMSEP of the reference is \simeq 0.83 for
 460 11 LVs. This means that PLSR is sensitive to these ANF samples.

461 Figure 16 shows that the PRM performance curve is between the PLSR with
 462 and without ANF samples curves. At best, it achieves an RMSEP equal to
 463 0.87 for 13 LVs.

464 Figure 16 shows that the RoBoost-PLSR curve reaches a minimum error with
 465 11 LVs, that is the same as the reference (RMSEP = 0.83). RoBoost-PLSR
 466 has a behaviour very similar to that of the reference. This means that
 467 RoBoost-PLSR attributes a 0 weights to the ANF samples but also to some

468 TTS samples.

469

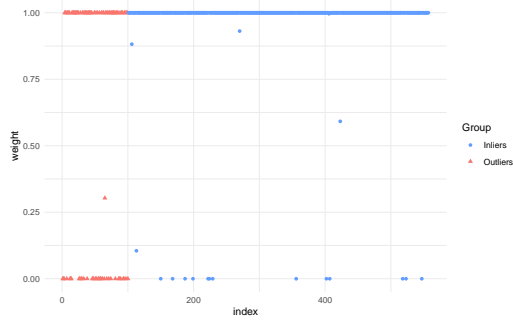


FIGURE 17: Repartition of the weights attributed to outliers (red) and inliers (blue) during the calibration of PRM for 13 latent variables with the best weights constant setting

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

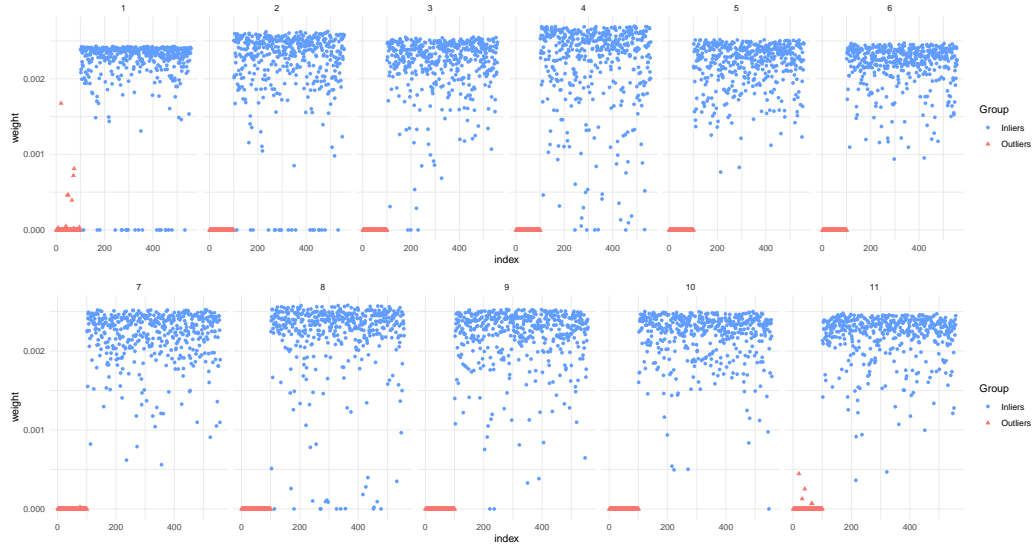


FIGURE 18: Repartition of the weights attributed to outliers (red) and inliers (blue) during the calibration of RoBoost-PLSR over 11 latent variables with the best weights constants settings

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

478 5. Conclusion & Perspectives

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

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

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

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

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

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

534 Finally, the robust multivariate methods have proven their reliable

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

538 Acknowledgements

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

546 Referencse

- 547 [1] S. Wold, H. Martens, H. Wold, The multivariate calibration problem in
548 chemistry solved by the pls method, in : B. Kågström, A. Ruhe (Eds.),
549 Matrix Pencils, Springer Berlin Heidelberg, Berlin, Heidelberg, 1983, pp.
550 286–293.
- 551 [2] S. Serneels, C. Croux, P. J. Van Espen, Influence properties of partial
552 least squares regression, Chemometrics and Intelligent Laboratory
553 Systems 71 (1) (2004) 13–20. doi:<https://doi.org/10.1016/j.chemolab.2003.10.009>.
554 [chemolab.2003.10.009](https://doi.org/10.1016/j.chemolab.2003.10.009).
- 555 [3] P. Filzmoser, S. Höppner, I. Ortner, S. Serneels, T. Verdonck, Cellwise
556 robust M regression, Computational Statistics & Data Analysis 147
557 (2020) 106944. doi:[10.1016/j.csda.2020.106944](https://doi.org/10.1016/j.csda.2020.106944).

- 558 [4] M. Griep, I. Wakeling, P. Vankeerberghen, D. Massart, Comparison of
559 semirobust and robust partial least squares procedures, *Chemometrics*
560 and *Intelligent Laboratory Systems* 29 (1) (1995) 37–50. [doi:10.1016/
561 0169-7439\(95\)80078-N](https://doi.org/10.1016/0169-7439(95)80078-N).
- 562 [5] I. Stanimirova, S. Serneels, P. J. Van Espen, B. Walczak, How to
563 construct a multiple regression model for data with missing elements
564 and outlying objects, *Analytica Chimica Acta* 581 (2) (2007) 324–332.
565 [doi:10.1016/j.aca.2006.08.014](https://doi.org/10.1016/j.aca.2006.08.014).
- 566 [6] R. J. Pell, Multiple outlier detection for multivariate calibration using
567 robust statistical techniques, *Chemometrics and Intelligent Laboratory*
568 *Systems* 52 (1) (2000) 87–104. [doi:10.1016/S0169-7439\(00\)00082-4](https://doi.org/10.1016/S0169-7439(00)00082-4).
- 569 [7] J. A. Gil, R. Romera, On robust partial least squares (PLS) methods,
570 *Journal of Chemometrics* 12 (6) (1998) 365–378. [doi:10.1002/\(SICI\)
571 1099-128X\(199811/12\)12:6<365::AID-CEM519>3.0.CO;2-G](https://doi.org/10.1002/(SICI)1099-128X(199811/12)12:6<365::AID-CEM519>3.0.CO;2-G).
- 572 [8] S. Acitas, P. Filzmoser, B. Senoglu, A new partial robust adaptive
573 modified maximum likelihood estimator, *Chemometrics and Intelligent*
574 *Laboratory Systems* 204 (2020) 104068. [doi:10.1016/j.chemolab.
575 2020.104068](https://doi.org/10.1016/j.chemolab.2020.104068).
- 576 [9] J. González, D. Peña, R. Romera, A robust partial least squares
577 regression method with applications, *Journal of Chemometrics* 23 (2)
578 (2009) 78–90. [doi:10.1002/cem.1195](https://doi.org/10.1002/cem.1195).
- 579 [10] I. N. Wakeling, H. J. H. Macfie, A robust PLS procedure, *Journal of*
580 *Chemometrics* 6 (4) (1992) 189–198. [doi:10.1002/cem.1180060404](https://doi.org/10.1002/cem.1180060404).

- 581 [11] J. Peng, S. Peng, Y. Hu, Partial least squares and random sample
582 consensus in outlier detection, *Analytica Chimica Acta* 719 (2012) 24–29.
583 [doi:10.1016/j.aca.2011.12.058](https://doi.org/10.1016/j.aca.2011.12.058).
- 584 [12] P. Filzmoser, R. Maronna, M. Werner, Outlier identification in high
585 dimensions, *Computational Statistics & Data Analysis* 52 (3) (2008)
586 1694–1711. [doi:10.1016/j.csda.2007.05.018](https://doi.org/10.1016/j.csda.2007.05.018).
- 587 [13] M. Hubert, K. V. Branden, Robust methods for partial least squares
588 regression, *Journal of Chemometrics* 17 (10) (2003) 537–549. [doi:10.](https://doi.org/10.1002/cem.822)
589 [1002/cem.822](https://doi.org/10.1002/cem.822).
- 590 [14] U. Kruger, Y. Zhou, X. Wang, D. Rooney, J. Thompson, Robust
591 partial least squares regression : Part II, new algorithm and benchmark
592 studies, *Journal of Chemometrics* 22 (1) (2008) 14–22, _eprint :
593 [https ://onlinelibrary.wiley.com/doi/pdf/10.1002/cem.1095](https://onlinelibrary.wiley.com/doi/pdf/10.1002/cem.1095). [doi:10.](https://doi.org/10.1002/cem.1095)
594 [1002/cem.1095](https://doi.org/10.1002/cem.1095).
- 595 [15] I. Hoffmann, S. Serneels, P. Filzmoser, C. Croux, Sparse partial robust
596 M regression, *Chemometrics and Intelligent Laboratory Systems* 149
597 (2015) 50–59. [doi:10.1016/j.chemolab.2015.09.019](https://doi.org/10.1016/j.chemolab.2015.09.019).
- 598 [16] P. Filzmoser, V. Todorov, Review of robust multivariate statistical
599 methods in high dimension, *Analytica Chimica Acta* 705 (1-2) (2011)
600 2–14. [doi:10.1016/j.aca.2011.03.055](https://doi.org/10.1016/j.aca.2011.03.055).
- 601 [17] S. F. Møller, J. v. Frese, R. Bro, Robust methods for multivariate data
602 analysis, *Journal of Chemometrics* 19 (10) (2005) 549–563, _eprint :

- 603 <https://onlinelibrary.wiley.com/doi/pdf/10.1002/cem.962>. [doi:10.](#)
604 [1002/cem.962](#).
- 605 [18] S. Serneels, C. Croux, P. Filzmoser, P. J. Van Espen, Partial robust
606 M-regression, *Chemometrics and Intelligent Laboratory Systems* 79 (1)
607 (2005) 55–64. [doi:10.1016/j.chemolab.2005.04.007](#).
- 608 [19] J. Betzin, Pls-regression in the boosting framework, in : M. Vilares,
609 M. Tenenhaus, P. Coelho, A. Morineau, V. Esposito Vinzi (Eds.), *PLS*
610 *and Related Methods*, DECISIA, Levallois Perret, 2003, pp. 261–269.
- 611 [20] A.-L. Boulesteix, PLS Dimension Reduction for Classification with
612 Microarray Data, *Statistical Applications in Genetics and Molecular*
613 *Biology* 3 (1), publisher : De Gruyter Section : *Statistical Applications in*
614 *Genetics and Molecular Biology* (Nov. 2004). [doi:10.2202/1544-6115.](#)
615 [1075](#).
- 616 [21] X. Shao, X. Bian, W. Cai, An improved boosting partial least squares
617 method for near-infrared spectroscopic quantitative analysis, *Analytica*
618 *Chimica Acta* 666 (1-2) (2010) 32–37. [doi:10.1016/j.aca.2010.03.](#)
619 [036](#).
- 620 [22] R. Rosipal, N. Krämer, Overview and Recent Advances in Partial Least
621 Squares, in : C. Saunders, M. Grobelnik, S. Gunn, J. Shawe-Taylor
622 (Eds.), *Subspace, Latent Structure and Feature Selection*, *Lecture Notes*
623 *in Computer Science*, Springer, Berlin, Heidelberg, 2006, pp. 34–51.
624 [doi:10.1007/11752790_2](#).

- 625 [23] M. H. Zhang, Q. S. Xu, D. L. Massart, Boosting partial least squares,
626 Analytical Chemistry 77 (5) (2005) 1423–1431, pMID : 15732927. doi:
627 [10.1021/ac048561m](https://doi.org/10.1021/ac048561m).
- 628 [24] D. J. Cummins, C. W. Andrews, Iteratively reweighted partial least
629 squares : A performance analysis by monte carlo simulation, Journal of
630 Chemometrics 9 (6) (1995) 489–507. doi:[10.1002/cem.1180090607](https://doi.org/10.1002/cem.1180090607).
- 631 [25] S. Schaal, C. G. Atkeson, S. Vijayakumar, Scalable Techniques from
632 Nonparametric Statistics for Real Time Robot Learning, Applied
633 Intelligence 17 (1) (2002) 49–60. doi:[10.1023/A:1015727715131](https://doi.org/10.1023/A:1015727715131).
- 634 [26] W. S. Cleveland, Robust Locally Weighted Regression and Smoothing
635 Scatterplots, Journal of the American Statistical Association (1979) 9.
- 636 [27] M. Metz, A. Biancolillo, M. Lesnoff, J.-M. Roger, A note on spectral
637 data simulation, Chemometrics and Intelligent Laboratory Systems 200
638 (2020) 103979. doi:[10.1016/j.chemolab.2020.103979](https://doi.org/10.1016/j.chemolab.2020.103979).
- 639 [28] M. Lesnoff, M. Metz, J.-M. Roger, Comparison of locally weighted PLS
640 strategies for regression and discrimination on agronomic NIR data,
641 Journal of Chemometrics 34 (5) (2020) e3209. doi:[10.1002/cem.3209](https://doi.org/10.1002/cem.3209).
- 642 [29] I. Hoffmann, P. Filzmoser, S. Serneels, K. Varmuza, Sparse and robust
643 PLS for binary classification, Journal of Chemometrics 30 (4) (2016)
644 153–162. doi:[10.1002/cem.2775](https://doi.org/10.1002/cem.2775).

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	$Y = 10 * T_{glucose}$	$\mathbf{Y} = -5 * \mathbf{T}_{glucose}$
\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.