Received: 14 April 2021 | Revised: 7 September 2021 | Accepted: 24 September 2021

DOI: 10.1002/sam.11558

WILEY

RESEARCH ARTICLE

# Data-driven sparse partial least squares

Hadrien Lorenzo[1,2,3] | Olivier Cloarec[2] | Rodolphe Thiébaut[1,4,5,6] | Jérôme Saracco[1,3]

[1]Inria, IMB, UMR 5251, Talence, France

[2]Corporate Research Advanced Data Analytics, Sartorius, Aubagne, France

[3]University of Bordeaux, CNRS, Bordeaux INP, IMB, UMR 5251, Talence, France

[4]SISTM, INSERM - U1219 BPH, University of Bordeaux, Bordeaux, France

[5]Vaccine Research Institute, Créteil, France

[6]CHU de Bordeaux, Service d'information médicale, Bordeaux, France

**Correspondence**
Hadrien Lorenzo, Inria, 200 Avenue de la Vieille Tour, 33405 Talence Cedex, France.
Email: hadrien.lorenzo@inria.fr

**Abstract**

In the supervised high dimensional settings with a large number of variables and a low number of individuals, variable selection allows a simpler interpretation and more reliable predictions. That subspace selection is often managed with supervised tools when the real question is motivated by variable prediction. We propose a partial least square (PLS) based method, called data-driven sparse PLS (ddsPLS), allowing variable selection both in the covariate and the response parts using a single hyperparameter per component. The subspace estimation is also performed by tuning a number of underlying parameters. The ddsPLS method is compared with existing methods such as classical PLS and two well established sparse PLS methods through numerical simulations. The observed results are promising both in terms of variable selection and prediction performance. This methodology is based on new prediction quality descriptors associated with the classical $R^2$ and $Q^2$, and uses bootstrap sampling to tune parameters and select an optimal regression model.

**KEYWORDS**

multiblock data, PLS regression, soft thresholding, supervised learning, variable selection

## 1 | INTRODUCTION

While recent biotechnologies offer more and more descriptors (denoted as $p$) for monitoring biological processes, the number of observations (denoted as $n$) does not increase, which implies the ill-posed problem $n \ll p$ in supervised settings (such as regression framework). Different methods have been developed in the past century to tackle this high dimensional problem and, partial least squares (PLS), particularly gathers attention since its introduction by Wold [27]. Indeed, this methodology has firstly become a standard in chemometrics calibration [13] and is now used in biology [29], animal genetics [10], in human genetics,[16] and in many other areas. However, to keep the ease of regression models' interpretation as $p$ grows further, variable selection has also started to be needed by PLS users.

Following the $L_1$-penalization of the regression coefficients introduced by Tibshirani [23], models that handle variable selection solutions (denoted as *sparse* methodologies in the following) and produce model regularization allowing to keep confidence in the built models have been introduced by Lê Cao et al. [11]. Even if Chun and Keleş [4]. Even if Chun and Keleş [4] proved that PLS consistency is lost as $n$ shrinks and $p$ grows, the PLS and sparse PLS methods have allowed authors to analyze real datasets meaningfully.

The PLS estimation goes through $R \in \mathbb{N}^{\star}$ identical processes which can be generally described as follows.

**(a)** Estimate the covariance matrix between covariate (x) and response (y) parts.

**(b)** Estimate the singular-space associated with the largest singular-value of the previous matrix.

**(c)** Project the covariate and response parts on the previously defined subspace.
**(d)** Estimate the linear regression matrix of the response part on the covariate part in the subspace.
**(e)** Remove the information carried by the current subspace from the covariate and response parts.

The step (a) corresponds to the estimation of a covariance matrix between the response and the covariate parts. The chosen estimator is very often $\mathbf{M} = 1/n\sum_{i=1}^{n} \left(\mathbf{y}_i - \bar{\mathbf{y}}\right) \left(\mathbf{x}_i - \bar{\mathbf{x}}\right)'$, for a data-set $(x_i, y_i)_{i=1..n}$ of size $n$ with empirical means $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$. This estimator is very sensitive to the curse of dimensionality, when the sample size is small and/or the number of variables is large, see for example [1]. However, the works previously mentioned only apply regularization methods in step (b), trying to overcome the failure to estimate the covariance matrix.

Therefore, the present work focuses on a regularized estimation of the covariance matrix directly in step (a). More precisely, the considered solution and its associated optimization problem are

$$S_\lambda(\mathbf{M}) = \underset{\mathbf{\Sigma}\in\mathbb{R}^{q\times p}}{\arg\min} ||\mathbf{M} - \mathbf{\Sigma}||^2 + 2\lambda|\mathbf{\Sigma}|, \tag{1}$$

where the soft-thresholding operator, an element-wise matrix operator, is defined as

$$\begin{aligned}\forall \mathbf{X} \in \mathbb{R}^{p,q},\ \ S_\lambda(\mathbf{X}) &= \left(S_\lambda\left(x_{i,j}\right)\right)_{(i,j)\in[\![1,p]\!]\times[\![1,q]\!]}\\ &= \left(\operatorname{sign}\left(x_{i,j}\right)\max(0,|x_{i,j}| - \lambda)\right)_{(i,j)\in[\![1,p]\!]\times[\![1,q]\!]},\end{aligned}$$

$||\bullet||$ and $|\bullet|$ are, respectively, the Frobenius and the $L_1$ norms. Note that if $\lambda = 0$, the chosen estimator is the empirical estimator of the variance–covariance matrix.

So far, no theoretical development has been conducted to study the behavior of this estimator but for the case of a variance matrix, that is, when $\mathbf{x} = \mathbf{y}$. Indeed, this case has been widely analyzed for the soft-thresholding operator but also for different thresholding operators. Johnstone and Lu [9] have explored the pertinence of thresholding the diagonal of the empirical covariance matrices in the objective of building sparse versions of the principal components analysis (PCA). Bickel and Levina [1] have put forward theoretical and numerical arguments to various types of thresholding operators. Also, Deshpande and Montanari [6] have shown the interest of such an idea where the whole empirical covariance matrix is thresholded, and the diagonal is removed.

In the PLS context, the number of components must be tuned, which is generally performed using prediction error. For sparse models, other parameters must also be adjusted by cross-validation and, more specifically, the leave-one-out method in the context of PLS. Recently, a sparse PLS approach (denoted as sgPLS for sparse group PLS and introduced in Refs. [18, 12]) has used bootstrap approach in Ref. [2] to efficiently overcome the difficulty of the "large $p$, small $n$" multiblock framework to select optimal models.

Recently, a meta-analysis of sparse PLS methodologies was performed by Mehmood et al. [14] over 16 different types of strategies divided in three classes which are "filter," "wrapper," and "embedded" methods. The first class "filter" corresponds to a filter which is applied on the output of a PLS model (e.g., on the regression coefficients) and allows to classify variables according to their importance. The second class "wrapper" considers methodologies where the result of a "filter" is used to refit a PLS model. This operation is produced a certain amount of times. The most classical methods in this context are the backward (and also forward even if this has not been studied in the PLS case yet) variable selection methodologies. It also takes into account randomized methodologies regarding to the studied variables or subsamples. This list is non-exhaustive and we encourage the reader to refer to the section "3.2 Wrapper methods" of [14] to appreciate the richness of the solutions devised. The third class "embedded" corresponds to context where variable selection is performed in the PLS model building.

In the following, the proposed sparse PLS model, based on empirical covariance thresholding, soft-thresholding more specifically, is studied. The thresholds and the number of components are tuned thanks to bootstrap operations. Following the classification of sparse PLS methods described in Ref. [14], the proposed methodology is close to the third class "embedded." Yet this methodology modifies the PLS algorithm working on the covariance matrix before the PLS model is actually built, this closeness to the data gave its name: *data-driven sparse PLS* (ddsPLS). This ddsPLS methodology opens a new class of sparse PLS methods.

In Section 2, the underlying statistical model associated with latent variables is introduced. Section 3 describes the PLS (NIPALS-PLS/PLS2) algorithm and details the existing sparse PLS solutions. Section 4 introduces the proposed ddsPLS solution and the associated model selection algorithm. Based on numerical studies, Section 5 reports the quality of the proposed methodology and compares it to the above-presented methods through simulation designs. Finally, Section 6 provides some concluding remarks.

## 2 | A LATENT VARIABLE MODEL

Let $\mathbf{x}$ (resp. $\mathbf{y}$) be a $p$-dimensional (resp. $q$-dimensional) random variable. Let assume that the variables of $\mathbf{x}$ and of $\mathbf{y}$ are centered and of unit variance. Let $n \in \mathbb{N}^\star$ be the sample size. Let $\mathbf{X}$ and $\mathbf{Y}$ be $n \times p$ and $n \times q$ random matrices where, for each row $i \in [[1, n]]$, $\mathbf{x}_i \sim \mathbf{x}$, and $\mathbf{y}_i \sim \mathbf{y}$. Let us further assume that the $\mathbf{x}_i$'s, respectively, the $\mathbf{y}_i$'s, are independent to each other. Different methods such as PCA and PLS are based on the principle of generative latent vector common to $\mathbf{x}$ and $\mathbf{y}$, denoted by $\boldsymbol{\phi}$ hereafter, a $\mathcal{R}$-dimensional random variable. The corresponding latent variable model can be written as

$$\begin{cases} \mathbf{x} = \mathbf{A}'\boldsymbol{\phi} + \varepsilon \text{ where } \mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_\mathcal{R}]' \in \mathcal{M}_{\mathcal{R}\times p}(\mathbb{R}) \\ \quad \text{with } \|\mathbf{a}_r\| \neq 0, \\ \mathbf{y} = \mathbf{D}'\boldsymbol{\phi} + \xi \text{ where } \mathbf{D} = [\mathbf{d}_1, \cdots, \mathbf{d}_\mathcal{R}]' \in \mathcal{M}_{\mathcal{R}\times q}(\mathbb{R}) \\ \quad \text{with } \|\mathbf{d}_r\| \neq 0, \\ \quad \text{with } \mathrm{var}(\boldsymbol{\phi}) = \mathbb{I}_\mathcal{R}, \mathbb{E}\boldsymbol{\phi} = \mathbf{0}_\mathcal{R}, \mathbb{E}\varepsilon = \mathbf{0}_p, \mathbb{E}\xi = \mathbf{0}_q, \\ \mathrm{cov}(\boldsymbol{\phi}, \varepsilon) = \mathbf{0}_{\mathcal{R}\times q}, \mathrm{cov}(\boldsymbol{\phi}, \xi) = \mathbf{0}_{\mathcal{R}\times q}, \mathrm{cov}(\varepsilon, \xi) = \mathbf{0}_{p\times q}. \end{cases} \tag{2}$$

This model implies that $\mathrm{var}(\mathbf{x}) = \mathbf{A}'\mathbf{A} + \mathrm{var}(\varepsilon)$ and $\mathrm{var}(\mathbf{y}) = \mathbf{D}'\mathbf{D} + \mathrm{var}(\xi)$. Furthermore, if the components of the random errors $\varepsilon$ and $\xi$ are independent and share the same variance, denoted by $\sigma^2$, it comes $\mathrm{var}(\mathbf{x}) = \mathbf{A}'\mathbf{A} + \sigma^2\mathbb{I}$ and $\mathrm{var}(\mathbf{y}) = \mathbf{D}'\mathbf{D} + \sigma^2\mathbb{I}$. In all cases, the relation holds $\mathrm{cov}(\mathbf{y}, \mathbf{x}) = \mathbf{D}'\mathbf{A}$.

**Remark 1.** Helland and Almøy [8] have studied the number R of latent variables shared by $\mathbf{y}$ and $\mathbf{x}$. They specified in which measure R is different from $\mathcal{R}$ in the general case. They have shown that the dimension R is equal to the largest number of eigen-vectors of $\mathbf{A}'\mathbf{A} = \mathrm{var}(\mathbf{x})$, which have non-null projections on $\mathbf{A}'\mathbf{D} = \mathrm{cov}(\mathbf{x}, \mathbf{y})$. In the PLS framework, R is the dimension of interest and is retrieved in the methodology as the estimated number of components.

The underlying latent variable model (2) intuitively implies the following regression model

$$\mathbf{y} = \mathbf{B}'\mathbf{x} + \mathbf{e}, \tag{3}$$

where $\mathbf{B} \in \mathbb{R}^{p\times q}$ is a non-stochastic matrix, $\mathbf{e} \in \mathbb{R}^q$ is a centered residual $q$-dimensional random vector, and $\mathbf{e}$ and $\mathbf{x}$ are independent. Matrix $\mathbf{B}$ must satisfy

$$(\mathbf{AB})'\boldsymbol{\phi} = \mathbf{D}'\boldsymbol{\phi} \underset{\mathbb{E}\cdot\boldsymbol{\phi}'}{\Rightarrow} \mathbf{AB} = \mathbf{D}. \tag{4}$$

In the general case, without any additional hypothesis, $\mathbf{B}$ is not unique, but for the trivial cases or when $\mathbf{AA}'$ is invertible, the ordinary least-squares estimator of $\mathbf{B}$ only uses Equation (3), minimizing the $l_2$-norm of $\mathbf{e}$, and provides $\widehat{\mathbf{B}} = \mathbf{X}^+\mathbf{Y}$, which is known to be unstable in the case of multicollinearity in the $\mathbf{X}$ matrix. Let us notice that the latent variable model (2) is also not identifiable. Indeed, for any matrix $\mathbf{G} \in \mathcal{GL}_\mathcal{R}(\mathbb{R})$, denoting by $\mathbf{t} = \mathbf{G}'\boldsymbol{\phi}$, $\mathbf{P} = \mathbf{G}^{-1}\mathbf{A}$ and $\mathbf{C} = \mathbf{G}^{-1}\mathbf{D}$, model (2) implies the following latent decomposition used by most of the authors.

$$\begin{cases} \mathbf{x} = \mathbf{P}'\mathbf{t} + \varepsilon, \\ \mathbf{y} = \mathbf{C}'\mathbf{t} + \xi, \\ \text{with var}\left(\mathbf{t}', \varepsilon', \xi'\right)' \text{diagonal}, \end{cases} \tag{5}$$

where $\mathbf{t}$ is called the score and $\mathbf{P}$ and $\mathbf{C}$ are called the loadings. Since the score and loadings are not identifiable, the PLS method concentrates on building a subspace that covers the space drawn by $\boldsymbol{\phi}$ through an unknown matrix $\mathbf{G}$. Note that, only when numerical simulations are carried out, Equation (4) allows to screen the quality of structural reconstruction of the current estimator $\widehat{\mathbf{B}}$ of $\mathbf{B}$ thanks to the metrics (e.g.)

$$\frac{\|\mathbf{A}\widehat{\mathbf{B}} - \mathbf{D}\|^2}{\|\mathbf{D}\|^2}, \tag{6}$$

which is as close to 0 as the estimated model is close to the theoretical one.

The PLS algorithm is a multilinear regression method able to estimate regression matrices even in the context of degenerate datasets, that is, when $\mathbf{X}'\mathbf{X}$ is singular. The principle of this algorithm is to iteratively estimate, here for the $r$th-iteration, the first right and left singular vectors, $\mathbf{u}_r$ and $\mathbf{v}_r$, of the empirical covariance matrix $\mathbf{M}^{(r)} = \mathbf{Y}^{(r)'}\mathbf{X}^{(r)}/(n-1)$ where $\mathbf{Y}^{(r)}$ and $\mathbf{X}^{(r)}$ are the residual matrices of the previous steps, see technical details in next section.

## 3 | NIPALS-PLS AND SPARSE PLS SOLUTIONS

This section recalls three different PLS algorithms. First the original PLS-Nipals algorithm, denoted by NIPALS-PLS [27] hereafter, is presented. Then the sparse PLS algorithm using $L_1$-weights regularization, denoted by sPLS [11] hereafter, is detailed. Finally, a modification of the previous sparse solution to control the non-convex part of the optimization problem, denoted by SPLS [4] hereafter, is provided.

### 3.1 | NIPALS-PLS (PLS2) solution

The NIPALS-PLS [27] algorithm, often referred to as PLS2 algorithm in the literature, solves the optimization problem given at the first row of Table 1. The $r$th iteration

**TABLE 1** Core optimization problem of each PLS-based, where $\mathbf{M}^{(r)} = \mathbf{Y}^{(r)\prime}\mathbf{X}^{(r)}/n - 1$ and $\mathbf{S}^{X,(r)} = \mathbf{X}'\mathbf{Y}^{(r)}\mathbf{Y}^{(r)\prime}\mathbf{X}$

| Method | Optimization problem | Constraints | Parameters |
|---|---|---|---|
| NIPALS-PLS [27] | $\max_{\mathbf{u},\mathbf{v}} \ \mathbf{v}'\mathbf{M}^{(r)}\mathbf{u}$ | $\mathbf{u}'\mathbf{u} = \mathbf{v}'\mathbf{v} = 1$ | $R$ |
| sPLS [11] | $\min_{\mathbf{u},\mathbf{v}} \ \left\|(n-1)\mathbf{M}^{(r)} - \mathbf{v}\mathbf{u}'\right\|^2 + \lambda_u^{(r)}\lvert\mathbf{u}\rvert + \lambda_v^{(r)}\lvert\mathbf{v}\rvert$ | $\mathbf{u}'\mathbf{u} = \mathbf{v}'\mathbf{v} = 1$ | $R, \left(\lambda_u^{(r)}, \lambda_v^{(r)}\right)_r$ |
| SPLS [4] | $\min_{\mathbf{w},\mathbf{c}} - \kappa\mathbf{w}\mathbf{S}^{X,(r)}\mathbf{w} + (1-\kappa)(\mathbf{c}-\mathbf{w})'\mathbf{S}^{X,(r)}(\mathbf{c}-\mathbf{w}) + \lambda_1\lvert\mathbf{c}\rvert + \lambda_2\left\|\mathbf{c}\right\|$ | $\mathbf{w}'\mathbf{w} = 1$ | $R, \kappa, \lambda_1, \lambda_2$ |
| ddsPLS | $\arg\max_{\mathbf{u},\mathbf{v}} \ \mathbf{v}'S_{\lambda_r}\left(\mathbf{M}^{(r)\prime}\right)\mathbf{u}$ | $\mathbf{u}'\mathbf{u} = \mathbf{v}'\mathbf{v} = 1$ | $R, (\lambda_r)_r$ |

of this algorithm is described in (7): using the initializations $\mathbf{X}^{(1)} = \mathbf{X}$ and $\mathbf{Y}^{(1)} = \mathbf{Y}$,

$$\forall r \in [[1, R]] \begin{cases} (\mathbf{a})\, \mathbf{u}_r = \overrightarrow{\text{RSV}}\left(\mathbf{M}^{(r)}\right), \mathbf{v}_r = \overrightarrow{\text{RSV}}\left(\mathbf{M}^{(r)\prime}\right), \\ (\mathbf{b})\, \mathbf{t}_r = \mathbf{X}^{(r)}\mathbf{u}_r, \\ (\mathbf{c})\, \mathbf{p}_r = \mathbf{X}^{(r)\prime}\mathbf{t}_r/\mathbf{t}_r'\mathbf{t}_r, \\ (\mathbf{d})\, \mathbf{c}_r = \mathbf{Y}^{(r)\prime}\mathbf{t}_r/\mathbf{t}_r'\mathbf{t}_r, \\ (\mathbf{e})\, \mathbf{X}^{(r+1)} = \mathbf{X}^{(r)} - \mathbf{t}_r\mathbf{p}_r', \quad \mathbf{Y}^{(r+1)} \\ = \mathbf{Y}^{(r)} - \mathbf{t}_r\mathbf{c}_r', \end{cases} \tag{7}$$

where $\overrightarrow{\text{RSV}}$ (for Right-Singular-Vector) is the function that returns the first right singular vector of its argument. In this algorithm, step **(a)** estimates the weights, step **(b)** estimates the scores, steps **(c)** and **(d)** estimate the linear regression matrices of $\mathbf{X}^{(r)}$ on $\mathbf{t}_r$ and $\mathbf{Y}^{(r)}$ on $\mathbf{t}_r$, and finally step **(e)** applies deflation. The regression matrix estimation is performed using the relation

$$\widehat{\mathbf{B}} = \mathbf{U}\left(\mathbf{P}'\mathbf{U}\right)^{-1}\mathbf{C}' \tag{8}$$

where $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_R]$, $\mathbf{P} = \left[\mathbf{p}_1, \dots, \mathbf{p}_R\right]$, and $\mathbf{C} = [\mathbf{c}_1, \dots, \mathbf{c}_R]$. If one wants to build the scores from an $\mathbf{x}$-matrix, potentially different from the one used to build the model, it must not use the matrix $\mathbf{U}$ directly but the matrix $\mathbf{U}(\mathbf{P}'\mathbf{U})^{-1}$: $\mathbf{X}\mathbf{U}(\mathbf{P}'\mathbf{U})^{-1} = (\mathbf{t}_1, \dots, \mathbf{t}_R) = \left(\mathbf{X}^{(1)}\mathbf{u}_1, \dots, \mathbf{X}^{(R)}\mathbf{u}_R\right)$.

**Remark 2.** In its original form, the PLS2/NIPALS-algorithm details the step (a) alternating estimations of $\mathbf{u}_r$ and $\mathbf{v}_r$ until convergence of $\mathbf{u}_r$ such as

$$\begin{aligned} \mathbf{u}_r &= \mathbf{X}^{(r)\prime}\mathbf{s}_r/\|\mathbf{X}^{(r)\prime}\mathbf{s}_r\| \\ \mathbf{t}_r &= \mathbf{X}^{(r)}\mathbf{u}_r/\left(\mathbf{u}_r'\mathbf{u}_r\right) \\ \mathbf{c}_r &= \mathbf{Y}^{(r)\prime}\mathbf{t}_r/\left(\mathbf{t}_r'\mathbf{t}_r\right) \\ \mathbf{s}_r &= \mathbf{Y}^{(r)}\mathbf{c}_r/\left(\mathbf{c}_r'\mathbf{c}_r\right) \end{aligned}$$

where $\mathbf{s}_r$ is initialized as the first column of $\mathbf{Y}^{(r)}$, see Ref. [19] for example. This alternating algorithm and the $\overrightarrow{\text{RSV}}$ operators both build $\mathbf{u}_r$ and the form associated to the operator will be kept in the following due to its compact aspect.

## 3.2 | Two sparse PLS solutions

Variable selection can be performed during the computation of the components used to construct the PLS models. Based on $L_1$-penalization, two sparse PLS solutions have been developed, denoted sPLS [11] and SPLS [4] hereafter. More precisely, the NIPALS-PLS maximization criterion is then modified such as detailed in the second and third rows, respectively, of Table 1.

The sPLS [11] introduces sparsity both in the $\mathbf{X}$ and in the $\mathbf{y}$ parts through $2R+1$ parameters. The underlying idea is to construct a solution using the soft threshold operator $S_\lambda(\cdot)$. The $r$th step of the associated algorithm is detailed in Equation (9): using the initializations $\mathbf{X}^{(1)} = \mathbf{X}$ and $\mathbf{Y}^{(1)} = \mathbf{Y}$,

$$\forall r \in [[1, R]] \begin{cases} \left(a^\dagger\right) \text{ Initialize } \mathbf{v}_r \text{ normalized, iterate until} \\ \quad \text{convergence of the two steps}: \\ \quad \left(\text{i}^\dagger\right) \mathbf{u}_r = S_{\lambda_u^{(r)}}\left(\mathbf{M}^{(r)\prime}\mathbf{v}_r\right), \mathbf{u}_r = \mathbf{u}_r/\|\mathbf{u}_r\|, \\ \quad \left(\text{ii}^\dagger\right) \mathbf{v}_r S_{\lambda_v^{(r)}}\left(\mathbf{M}^{(r)}\mathbf{u}_r\right), \mathbf{v}_r = \mathbf{v}_r/\|\mathbf{v}_r\|, \\ (\text{b})\, \mathbf{t}_r = \mathbf{X}^{(r)}\mathbf{u}_r, \\ (\text{c})\, \mathbf{p}_r = \mathbf{X}^{(r)\prime}\mathbf{t}_r/\mathbf{t}_r'\mathbf{t}_r, \\ (\text{d})\, \mathbf{c}_r = \mathbf{Y}^{(r)\prime}\mathbf{t}_r/\mathbf{t}_r'\mathbf{t}_r, \\ (\text{e})\, \mathbf{X}^{(r+1)} = \mathbf{X}^{(r)} - \mathbf{t}_r\mathbf{p}_r', \quad \mathbf{Y}^{(r+1)} = \mathbf{Y}^{(r)} - \mathbf{t}_r\mathbf{c}_r', \end{cases} \tag{9}$$

where the regression matrix estimation is performed using the Relation (8). One might notice that $\lambda_v^{(r)}$ fixes the sparsity of $\mathbf{v}_r$. However, since Equation **(e)** in Algorithm (7) does not explicitly impose sparsity in the $\mathbf{y}$ part, then the estimated matrix $\widehat{\mathbf{B}}$ is not constrained to be sparse in $\mathbf{Y}$. In the implementation, the search for parameters $\left(\lambda_u^{(r)}, \lambda_v^{(r)}\right)$ is driven, in an equivalent way, by two new parameters $\left(\text{keep}_X, \text{keep}_Y\right)_r$ which respectively correspond to the number of selected variables in both $\mathbf{x}$ and $\mathbf{y}$ parts for the $r$th component.

The sPLS [4] concentrates on the **x** part sparsity and needs only four parameters to be tuned. Looking at the optimization problem of SPLS [4], the sparse solution **c**, relying on a $L_1$-penalization via the parameter $\lambda_1$, is not directly sought. Instead the algorithm defines **c** as a direction relatively close to **w**, the "associated PLS-based solution." This distance is measured according to the $\mathbf{S}^{X,(r)}$-metric and the importance of each of these two terms in the overall optimization problem is tuned by a parameter $\kappa \in [0,1]$. If $\kappa$ is high (equal 1 at the limit), the problem yields to the PLS [27] solution, while a low $\kappa$ means that **c** and **w** must be very close to each other but not necessarily to the classical PLS solution. According to Ref. [4], $\lambda_2$ can be tuned to $+\infty$ as to preserve the soft-thresholding solution and $\kappa < 1/2$ avoids local solution issues. There are now only three parameters $(R, \kappa, \lambda_1)$ left to be tuned. Algorithm (10) is the associated algorithm in the case of so-called NIPALS deflation, as described by the authors where NIPALS − PLS(.) is a function that takes a covariate and a response matrix for the two first arguments and a number of components in third and sends the NIPALS-PLS regression matrix such as described in Algorithm (7). In the following, the regression matrix is estimated as $\hat{\mathbf{B}} = \mathbf{B}^{(R)}$, also $\mathcal{A}$ is the index set of active variables, updated at each iteration, such as

$$\forall r \in [\![1,R]\!] : \left\{ \begin{array}{c} \mathbf{Z}^{(r)} = \mathbf{X}'\mathbf{Y}^{(r)} \text{ and } \mathbf{S}^{X,(r)} = \mathbf{Z}^{(r)}\mathbf{Z}^{(r)'} \\ \text{Initialize } \mathbf{c}^{(r)}, \text{iterate until convergence} \\ \text{of the two steps :} \\ \mathbf{w}^{(r)} = \underset{\mathbf{w},\mathbf{w}'\mathbf{w}=1}{\arg\min} - \kappa \mathbf{w}'\mathbf{S}^{X,(r)}\mathbf{w} \\ + (1-\kappa)\left(\mathbf{c}^{(r)} - \mathbf{w}\right)'\mathbf{S}^{X,(r)}\left(\mathbf{c}^{(r)} - \mathbf{w}\right), \\ \mathbf{c}^{(r)} = \underset{\mathbf{c}}{\arg\min}\left(\mathbf{Z}^{(r)'}\mathbf{c} - \mathbf{Z}^{(r)'}\mathbf{w}^{(r)}\right)' \\ \left(\mathbf{Z}^{(r)'}\mathbf{c} - \mathbf{Z}^{(r)'}\mathbf{w}^{(r)}\right) + \lambda_1 \mid \mathbf{c} \mid, \\ \mathcal{A} = \left\{ i \in [\![1,p]\!] \mid \mathbf{w}_i^{(r)} \neq 0 \vee \mathbf{B}_{i,.}^{(r-1)} = \mathbf{0} \right\}, \\ \mathbf{B}_{\mathcal{A}}^{(r)} = \text{NIPALS} - \text{PLS}\left(\mathbf{X}_{\mathcal{A}}, \mathbf{Y}^{(r)}, r\right) \\ \text{and } \mathbf{B}_{-\mathcal{A}}^{(r)} = \mathbf{0}, \\ (\text{e}') \quad \mathbf{Y}^{(r+1)} = \mathbf{Y}^{(r)} - \mathbf{X}\mathbf{B}^{(r)}. \end{array} \right. \tag{10}$$

The deflation in sPLS [11] is performed as in the classical NIPALS-PLS algorithm (see row **(e)** of Algorithm (7)), while only the **Y** part is deflated in the SPLS [4] algorithm (using row **(e′)** of Algorithm (10), instead of row **(e)** of Algorithm (7)).

The method proposed in the next section is a new sparse PLS methodology based on a soft-thresholding operation of the empirical covariance matrix, where the associated hyperparameters are adjusted via bootstrap.

## 4 | DATA-DRIVEN SPARSE PLS

This section details the ddsPLS method. First, Section 4.1 provides the underlying algorithm. Then Section 4.3 introduces the main quality descriptors generally used in PLS regression and their adaptations to the bootstrap context with slight modifications. Section 4.4 explains how to tune the different hyperparameters of the ddsPLS method, and an illustration based on an "easy" example is given in Section 4.6. Finally Section 4.2 provides the extension of the ddsPLS approach to the case of multiblock **x** datasets.

**Remark 3.** Since the underling latent variable model (2) assumes centered and scaled **x** and **y** variables, the data are (marginally) standardized on the overall dataset before applying the ddsPLS methodology. So data are standardized by subtracting the empirical mean and dividing by the empirical standard deviation. This standardization step makes the ddsPLS methodology sensitive to potential extreme observations' adverse effects and thus to outliers.

### 4.1 | ddsPLS algorithm

As noticed in Ref. [14], sPLS [11], and SPLS [4] are embedded sparse methodologies. But those methodologies are based on the penalization of the **x** parts from the eigen decomposition of $\mathbf{Y}'\mathbf{X}$ (and also of the **y** part for sPLS [11], but through a different regularization coefficient). The idea of the ddsPLS method is to soft-threshold directly the empirical covariance matrix right, using the $S_\lambda(.)$ operator, before building the weights. This operation naturally tends to remove non-interesting variables since their associated soft-thresholded empirical covariance coefficients are equal to 0, for a threshold $\lambda$ large enough, and the associated weight goes to 0. This regularization of the "data itself" (actually the empirical covariance matrix) and not of the associated weights (which are first eigenvectors of the empirical covariance matrices) led to the name "data-driven sparse PLS (ddsPLS)" to the proposed method. The corresponding optimization problem is provided in the last row of Table 1. One might notice that if $\lambda_r = 0$, it corresponds to the usual NIPALS-PLS [27] solution. The associated algorithm of the ddsPLS method is detailed below. Note that, to introduce sparsity in the response in the regression matrix $\hat{\mathbf{B}}$, the step **(d)** of the Algorithm (7) is modified such as

$$\mathbf{c}_r = \left(\mathbf{Y}^{(r)}\Pi_r\right)'\mathbf{t}_r/\mathbf{t}_r'\mathbf{t}_r$$

where $\Pi_r = \text{diag}\left(\left\{\delta_{\neq 0}\left(\mathbf{v}_r\right)_j\right\}_{j\in[\![1,q]\!]}\right)$ is sparse since "$\mathbf{v}_r$" is sparse for $\lambda$ large enough. Here also, if $\lambda = 0$, the result is equal to the solution offered by the NIPALS-PLS [27] solution.

The algorithm of ddsPLS is then

$$\forall r \in [\![1,R]\!] \begin{cases} (\mathbf{a}^\star)\ \mathbf{u}_r = \overrightarrow{\mathrm{RSV}}\left(S_{\lambda^{(r)}}\left(\mathbf{M}^{(r)}\right)\right), \mathbf{v}_r \\ = \overrightarrow{\mathrm{RSV}}\left(S_{\lambda^{(r)}}\left(\mathbf{M}^{(r)\prime}\right)\right), \\ (\mathbf{b})\,\mathbf{t}_r = \mathbf{X}^{(r)}\mathbf{u}_r, \\ (\mathbf{c})\ \mathbf{p}_r = \mathbf{X}^{(r)\prime}\mathbf{t}_r/\mathbf{t}_r'\mathbf{t}_r, \\ (\mathbf{d}^\star) \begin{cases} \Pi_r = \mathrm{diag}\left(\left\{\delta_{\neq 0}\left(\mathbf{v}_r\right)_j\right\}_{j\in[\![1,q]\!]}\right) \\ \mathbf{c}_r = \underset{\mathbf{V}}{\arg\min}\left\|\mathbf{Y}^{(r)}\Pi_r - \mathbf{t}_r\mathbf{V}'\right\|^2 \\ = \left(\mathbf{Y}^{(r)}\Pi_r\right)'\mathbf{t}_r/\left(\mathbf{t}_r'\mathbf{t}_r\right). \end{cases} \\ (\mathbf{e})\quad \mathbf{X}^{(r+1)} = \mathbf{X}^{(r)} - \mathbf{t}_r\mathbf{p}_r', \quad \mathbf{Y}^{(r+1)} \\ = \mathbf{Y}^{(r)} - \mathbf{t}_r\mathbf{c}_r', \end{cases} \tag{11}$$

where the regression matrix estimation is also performed using the Relation (8).

Notice that $R+1$ parameters need to be tuned to obtain an optimal solution: the number $R$ of components and the associated regularization coefficient $\lambda^{(r)}$ for each component. Based on different well-known quality descriptors used in PLS and on new appealing ones introduced in our ddsPLS context (see next subsection), a way to calibrate these different parameters is provided in Section 4.4.

## 4.2 | Adaptation to the multiblock structure

A multiblock generalization of the latent variable model (2) is

$$\forall t \in [\![1,T]\!],\ \ \mathbf{x}_t = \mathbf{A}_t'\boldsymbol{\phi} + \varepsilon_t \ \text{and}\ \ \mathbf{y} = \mathbf{C}'\boldsymbol{\phi} + \xi, \tag{12}$$

where $T$ is the total number of blocks. The corresponding underlying regression model can be written as

$$\mathbf{y} = \sum_{t=1}^{T}\mathbf{B}_t'\mathbf{b}_t + \mathbf{e}. \tag{13}$$

Let us introduce the following notation:

$$\mathbf{x} = \left(\mathbf{x}_1', \ldots, \mathbf{x}_T'\right)', \mathbf{A} = [\mathbf{A}_1, \ldots, \mathbf{A}_T] \\ \text{and}\,\mathbf{B} = \left[\mathbf{B}_1', \ldots, \mathbf{B}_T'\right]'. \tag{14}$$

Model (12) can be rewritten in the form of Model (2), thus it is possible to use the ddsPLS methodology as previously in this multiblock framework.

**Remark 4.** The adaptation of the PLS method to the multiblock context has firstly been described in Ref. [28] and finally defined in Ref. [24] to take the form

$$\begin{aligned} &\max_{\mathbf{w}_t,\mathbf{v},c_t} \mathbf{v}'\mathbf{Y}^{(r)\prime}\sum\nolimits_{t=1}^{T}\left(c_t\mathbf{X}_t^{(r)}\mathbf{w}_t\right), \\ &\text{s.t.} \qquad \|\mathbf{w}_t\| = \|\mathbf{v}\| = 1 \\ &\text{and} \qquad \sum\nolimits_{t=1}^{T} c_t^2 = 1. \end{aligned} \tag{15}$$

If $\mathbf{u}$ is associated to a solution of any of the previous algorithms (except for SPLS [4]) in steps **(a)**, **(i$^\dagger$)** or **(a$^\star$)**, and $\mathbf{u}_t$ is the extracted matrix from $\mathbf{u}$ associated with block $t$, then by identification

$$\mathbf{w}_t = \mathbf{u}_t/\|\mathbf{u}_t\| \ \text{and}\ c_t = \|\mathbf{u}_t\|. \tag{16}$$

**Remark 5.** For the deflation steps (see steps **(e)** and **(e′)**), the "mbPLS" algorithm (for multiblock PLS) uses the super-score deflation where the super-score gathers information from the $T$ different blocks: $\mathbf{t}_r = \sum_{t=1}^{T} c_t\mathbf{X}_t^{(r)}\mathbf{w}_t$. Hence, in the following, when NIPALS-PLS [27] or sPLS [11] are used in the multiblock framework, the super-score deflation is always used. Moreover, for the multiblock version of the ddsPLS algorithm, the super-score deflation strategy is also used.

Note different authors consider that deflating each block on the super-score might mix block information and, therefore, might also drive to senseless conclusions. For example, Westerhuis and Smilde [26] present specific designs where no deflation on the $\mathbf{X}$ blocks provides better prediction results. This corresponds to the solution chosen in the SPLS [4] algorithm.

**Remark 6.** When the response variable $\mathbf{y}$ has multiblocks, the proposed methodology remains the same where the different blocks are concatenated, as done in this section if $\mathbf{x}$ has multiblocks, see (14), and the ddsPLS methodology can again be applied on the resulting data-set.

## 4.3 | Model quality descriptors

The PLS methodology is known to provide the correct common subspaces between the $\mathbf{x}$ and the $\mathbf{y}$ parts if there is indeed information to catch. Whereas, if this cross structure is poor, the methodology sometimes over-fits focusing on the $\mathbf{x}$ part variance in particular when the dimension $p$ of $\mathbf{x}$ is large, see for example [5, 25]. Its side effect artificially increases the $R^2$ statistics, also denoted as explained variance or determination coefficient, which only describes the goodness of fit on the train dataset:

$$R^2 = 1 - \frac{\sum_{j=1}^{q}\sum_{i=1}^{n}\left(y_{i,j} - \hat{y}_{i,j}\right)^2}{\sum_{j=1}^{q}\sum_{i=1}^{n}\left(y_{i,j} - \bar{y}_j\right)^2}, \tag{17}$$

where $\bar{y}_j$ is the empirical mean of the $j$th-variable of the $\mathbf{y}$ part and $\hat{y}_{i,j}$ is the estimation of $y_{i,j}$ thanks to the current model. The numerator is often denoted as the residual sum of squares (RSS) and the denominator as the total sum of squares (TSS) and thus $R^2 = 1 - \text{RSS}/\text{TSS}$. This statistics can actually be interpreted as an estimator of

$$\gamma(\mathbf{p}) = 1 - \frac{\sum_{j=1}^{q}\text{var}\left(y_j - y_j^{(\mathbf{p})}\right)}{\sum_{j=1}^{q}\text{var}(y_j)}, \tag{18}$$

built on the (potentially stochastic) prediction model $\mathbf{p}$ where $y_j^{(\mathbf{p})}$ is an estimator of $y_j$ based on the prediction model $\mathbf{p}$. This metrics is as close to 1 as the prediction error is low. Its interest is to take into account the different scales of the variables $y_j$. As an interpretation, asymptotically (when $n \to +\infty$), if this metric is equal to 0, then the associated prediction model performs equivalently to the mean prediction model, and if it is below 0, it performs worse. In the oracle context (a prediction model which knows the true relations between the informational variables), the estimator of $\mathbf{y}$ by the oracle predictor model $\mathbf{p}^\star$ is $\mathbf{y}^\star = \mathbf{A}'\boldsymbol{\phi}$ according to Model (2), and the previous metrics can be written as

$$\gamma^\star = 1 - \frac{\sum_{j=1}^{q}\text{var}\left(\varepsilon_j\right)}{\sum_{j=1}^{q}\text{var}\left(y_j\right)}. \tag{19}$$

**Remark 7.** The two previous statistics are global in the sense of the $q$-dimensional response variable $\mathbf{y}$ and, therefore, do not allow for a marginal critique of the model. Another interesting statistics would be $\gamma_j(\mathbf{p}) = 1 - \text{var}\left(y_j - y_j^{(\mathbf{p})}\right)/\text{var}\left(y_j\right)$, for $j = 1 \dots q$, which exhibits the marginal effects of each variable on the response variable. However, the response variables have been standardized in order to be processed on the same scale (see Remark 3), which rationalizes the definition of the $\gamma$. Although this does not make it possible to identify marginally the variables of $\mathbf{y}$ that can potentially degrade the criterion, their deleterious effects remain observable. Moreover, in the case of marginal $\gamma_j$, it would have been necessary to define a transformation aggregating the $q$ marginal criteria making it possible to define a decision algorithm on the conservation or not of the current component. The definition of such a transformation is not trivial and, therefore, will not be considered in this work. For all these reasons, the global statistics $\gamma$ is used in the proposed methodology.

The $R^2$ estimator is subject to over-fitting, this is due to the fact that the quality of the model is assessed on the whole sample used to build the model.

To circumvent this over-fitting drawback, the criterion $Q^2$ has been introduced by Stone [17] and Geisser [7]. It is based on $F$-folds cross-validation such as

$$Q^2 = 1 - \frac{\sum_{j=1}^{q}\sum_{f=1}^{F}\sum_{i\notin \text{cv}_f}\left(y_{i,j} - \hat{y}_{i,j}^{(\text{cv}_f)}\right)^2}{\sum_{j=1}^{q}\sum_{f=1}^{F}\sum_{i\notin \text{cv}_f}\left(y_{i,j} - \bar{y}_j\right)^2}, \tag{20}$$

where $\hat{y}_{i,j}^{(\text{cv}_f)}$ is the estimation of $y_{i,j}$ thanks to the current model where the observations in "cv$_f$" are used as the train sample for fold "$f$." The numerator is often denoted as the PRediction Error Sum of Squares (PRESS) and thus $Q^2 = 1 - \text{PRESS}/\text{TSS}$.

Thanks to $R^2$ and $Q^2$, different rules have been used to select "optimal" models. In the general prediction context, the most popular maximizes $Q^2$, which is equivalent to minimize the mean squared error in prediction (MSEP) since the denominator of the right-hand side term of $Q^2$ does not use the model's prediction (but the mean estimator). In the context of PLS, a rule of thumb is to favor models for which the difference between $R^2$ and $Q^2$ is minimum, which is supposed to correspond to a minimum of over-fitting, see for example, Cloarec [5].

Those metrics $R^2$ and $Q^2$ are helpful to evaluate the quality of a complete model. Other metrics have been introduced to evaluate the quality of a current $r$th-component of the model and are denoted as $R_r^2$ and $Q_r^2$, see for example, Tenenhaus [19]. Let us suppose that the first $(r-1)$ components have already been built, then the quality of the $r$th-component can be screened thanks to

$$\begin{aligned} R_r^2 &= 1 - \frac{\sum_{j=1}^{q}\sum_{i=1}^{n}\left(y_{i,j} - \left(\hat{y}_{i,j}^{(r)} - \hat{y}_{i,j}^{(r-1)}\right) - \bar{y}_j\right)^2}{\sum_{j=1}^{q}\sum_{i=1}^{n}\left(y_{i,j} - \bar{y}_j\right)^2}, \\ Q_r^2 &= 1 - \frac{\sum_{j=1}^{q}\sum_{f=1}^{F}\sum_{i\notin \text{cv}_f}\left(y_{i,j} - \hat{y}_{i,j}^{(\text{cv}_f,r)}\right)^2}{\sum_{j=1}^{q}\sum_{f=1}^{F}\sum_{i\notin \text{cv}_f}\left(y_{i,j} - \hat{y}_{i,j}^{(\text{cv}_f,r-1)}\right)^2}, \end{aligned} \tag{21}$$

where $\hat{y}_{i,j}^{(r)}$, respectively, $\hat{y}_{i,j}^{\star,(r)}$, is the prediction of the $i$th-observation and the $j$th-response variable thanks to model based on components the first $r$ and all observations, respectively, all observations except the $i$th one. By convention, $\hat{y}_{i,j}^{(0)} = \bar{y}_j$ and $\hat{y}_{i,j}^{\star,(0)} = \frac{1}{n-1}\sum_{i'\neq i} y_{i',j}$. The metrics $R_r^2$ represents the proportion of variability which is explained by the new component $r$ and not by the $(r-1)$ previous ones, while the $Q_r^2$ does the same but based on out-of-fold samples from the cross-validation fold.

Rules of thumb exist to validate the current component based on the value of $Q_r^2$, for example $Q_r^2 \geq 0.0975$ following for instance [21, 11], or $Q_r^2 \geq 0.05$ according to [15, 20]) or $Q_r^2 \geq 0$ (or $\geq 0.05$ if $n \leq 100$) according to Ref. [22].

Based on these descriptors $Q_r^2$, another metrics which describes the overall quality of a $R$-components model is the cumulative $Q^2$ criterion, denoted as $Q^2_{R,(\text{cum})}$ hereafter, and defined as an aggregation of the $R$ criteria $\left(Q_1^2, \dots, Q_R^2\right)$ such as

$$Q^2_{R,(\text{cum})} = 1 - \prod_{r=1}^{R} \left(1 - Q_r^2\right).$$

This metric $Q^2_{R,(\text{cum})}$ is used in Ref. [19] with the rule that the current component $R$ is retained if $Q^2_{R,(\text{cum})} \gg Q^2_{R-1,(\text{cum})}$.

In the previous statistics $Q_r^2$ (and thus in the metric $Q^2_{R,(\text{cum})}$), the performances of numerous prediction models are simultaneously mixed, over the $F$-folds, in the numerator and in the denominator. One consequence of this is that model selection is driven by prediction models for which the errors are the most important. To circumvent this drawback, for $F$-folds cross-validation, one way is first to consider the $F$ different "marginal" $Q^2$-like statistics, denoted as

$$1 - \frac{\sum_{j=1}^{q} \sum_{i \notin \text{cv}_f} \left(y_{i,j} - \widehat{y}_{i,j}^{(\text{cv}_f)}\right)^2}{\sum_{j=1}^{q} \sum_{i \notin \text{cv}_f} \left(y_{i,j} - \overline{y}_{j}^{(\text{cv}_f)}\right)^2}, \text{for } f = 1, \dots, F, \quad (22)$$

and then to combine them a posteriori using the empirical mean. To the best of our knowledge, this approach has never been proposed in the PLS literature. However, when the sample size $n$ is small, using $F$-folds cross-validation is not relevant and the use of bootstrap is interesting to enrich the underlying information of the available data. This is the strategy chosen for the continuation of this work.

Based on $B$ bootstrap samples, the bootstrapped versions of $R^2$ and $Q^2$ are given by

$$\overline{R}_B^2 = \frac{1}{B}\sum_{b=1}^{B} R_b^2 \quad \text{and} \quad \overline{Q}_B^2 = \frac{1}{B}\sum_{b=1}^{B} Q_b^2 \quad (23)$$

with, for the current bootstrap sample $b$,

$$R_b^2 = 1 - \frac{\sum_{j=1}^{q} \sum_{i \in \text{IN}(b)} \left(y_{i,j} - \widehat{y}_{i,j}^{b}\right)^2}{\sum_{j=1}^{q} \sum_{i \in \text{IN}(b)} \left(y_{i,j} - \overline{y}_{j}^{b}\right)^2},$$

$$Q_b^2 = 1 - \frac{\sum_{j=1}^{q} \sum_{i \in \text{OOB}(b)} \left(y_{i,j} - \widehat{y}_{i,j}^{b}\right)^2}{\sum_{j=1}^{q} \sum_{i \in \text{OOB}(b)} \left(y_{i,j} - \overline{y}_{j}^{b}\right)^2}, \quad (24)$$

where IN($b$) and OOB($b$) are respectively the set of the in-bag observations and the set of the out-of-bag observations, $\widehat{y}_{i,j}^{b}$ is the prediction of the $i$th-observation and the $j$th-response variable and $\overline{y}_{j}^{b}$ is the empirical mean of the $j$th-response variable estimated on the bootstrap sample $b$.

In the same way, bootstrapped versions of $R_r^2$ and $Q_r^2$ are given by

$$\overline{R}_{B,r}^2 = \frac{1}{B}\sum_{b=1}^{B} R_{b,r}^2 \quad \text{and} \quad \overline{Q}_{B,r}^2 = \frac{1}{B}\sum_{b=1}^{B} Q_{b,r}^2 \quad (25)$$

where

$$R_{b,r}^2 = 1 - \frac{\sum_{j=1}^{q} \sum_{i \in \text{IN}(b)} \left(y_{i,j} - \left(\widehat{y}_{i,j}^{b,(r)} - \widehat{y}_{i,j}^{b,(r-1)}\right) - \overline{y}_{j}^{b}\right)^2}{\sum_{j=1}^{q} \sum_{i \in \text{IN}(b)} \left(y_{i,j} - \overline{y}_{j}^{b}\right)^2},$$

$$Q_{b,r}^2 = 1 - \frac{\sum_{j=1}^{q} \sum_{i \in \text{OOB}(b)} \left(y_{i,j} - \widehat{y}_{i,j}^{b,(r)}\right)^2}{\sum_{j=1}^{q} \sum_{i \in \text{OOB}(b)} \left(y_{i,j} - \widehat{y}_{i,j}^{b,(r-1)}\right)^2}. \quad (26)$$

**Remark 8.** Those metrics, $\overline{R}_B^2$, $\overline{Q}_B^2$, $\overline{R}_{B,r}^2$, and $\overline{Q}_{B,r}^2$, share the same philosophies as those discussed for Equations (17), (20), and (21), but they introduce two innovative ideas (except for the bootstrap analysis itself). Firstly, they are empirical means over all bootstrap replications. This allows not mixing bootstrap quadratic errors, which is a good feature since estimations can vary from one bootstrap sample to another. Secondly, the "$R^2$"-spirit metrics are based directly on the $b$th bootstrap replication but use the associated in-bag sample, while the "$Q^2$"-spirit metrics use the associated out-of-bag sample. This implies that the estimators $R_b^2$ and $R_{b,r}^2$ are even more subject to over-fitting and lead the difference "$\overline{R}_B^2 - \overline{Q}_B^2$" to be more discriminant and informative in the presence of over-fitting case.

The following section details how these metrics are used to adjust the hyperparameters of the ddsPLS method by following the spirit of their cross-validation versions.

## 4.4 | Selection of the hyperparameters

The ddsPLS algorithm detailed in Equation (11) and the other sparse PLS methods require hyperparameters to be adjusted. In the case of ddsPLS, the following regularization parameters, $R$ and $\Lambda_R = \{\lambda_r, r = 1, \dots, R\}$, must be optimally tuned. Due to the variance constraints of the variables $\mathbf{x}$ and $\mathbf{y}$, we know that $\lambda_r \in [0, 1]$ where

- $\lambda_r = 0$, corresponds to the NIPALS-PLS solution,

- $\lambda_r = 1$, leads to no variable selected and thus to the prediction model equal to empirical mean estimation.

This optimal hyperparameter selection is based on the minimization of $R_r^2 - Q_r^2$ (that might correspond to a minimum in over-fitting for the $r$th-component). To this end, the proposed ddsPLS strategy considers $\overline{R}^2_{B,r} - \overline{Q}^2_{B,r}$, the bootstrapped version of the corresponding metrics defined in (23).

More precisely, the selected model must satisfy that, for a current model built on $(r-1)$ components with estimated values for regularization parameters $\widehat{\Lambda}_{r-1} = \left\{\widehat{\lambda}_1, \cdots, \widehat{\lambda}_{r-1}\right\}$, the $r$th-component is acceptable if

1. the $r$th-component performs better than the mean estimation (i.e., $\overline{Q}^2_{B,r} > 0$),
2. the model built on $r$ components performs better than the one built on $(r-1)$ components (i.e., $\overline{Q}^2_B$ is increasing with the number of components),
3. the bootstrapped explained variance is close to the bootstrapped cross-validated explained variance (i.e., $\overline{R}^2_B - \overline{Q}^2_B$ is minimum on the set of hyperparameters).

This hyperparameter selection procedure is illustrated on a toy example in Section 4.6.

## 4.5 | A theoretical lower bound for the regularization coefficients

Among the sparse covariance estimators the solution developed in Ref. [3] uses a fully data-driven estimator for each element of the covariance matrix. Even if they consider the variance case (where $\mathbf{y} = \mathbf{x}$) and assume that random parts are normally distributed, they use the adaptive threshold

$$\lambda_{i,j} = \delta\sqrt{\frac{\widehat{\theta}_{i,j}\log p}{n}}, \tag{27}$$

where $\delta$ can be whether taken equal to 2, whether empirically chosen though cross-validation and $\theta_{i,j} = \text{var}\left((\mathbf{x}_i - \mathbb{E}\mathbf{x}_i)\left(\mathbf{x}_j - \mathbb{E}\mathbf{x}_j\right)\right)$, the variance of the covariance coefficient between $\mathbf{x}_i$ and $\mathbf{x}_j$.

As $n$ decreases/$p$ increases, the probability for a covariance coefficient to be larger than a given value increases. The value of $\lambda_{i,j}$ estimates the variability of the covariance coefficient below which non null coefficient can be considered as null with a high probability.

Inspired from the previous work, let us introduce the following value

$$\lambda_r^{(0)} = \frac{1}{pq}\sum_{j=1}^{q}\sum_{i=1}^{p}\sqrt{\frac{\theta^{(r)}_{(j,i)}\log(\max(p,q))}{n}}$$
$$\text{where } \theta^{(r)}_{(j,i)} = \frac{1}{n}\sum_{k=1}^{n}\left(x^{(r)}_{(k,i)}y^{(r)}_{(k,j)} - m^{(r)}_{(j,i)}\right)^2,$$

where $x^{(r)}_{(k,i)}, y^{(r)}_{(k,j)}$, and $m^{(r)}_{(j,i)}$ are the elements of the matrices $\mathbf{X}^{(r)}, \mathbf{Y}^{(r)}$, and $\mathbf{M}^{(r)}$, respectively, at positions $(k,i)$, $(k,j)$, and $(j,i)$ respectively. This value $\lambda_r^{(0)}$ is the lowest accessible value by the algorithm detailed in the previous section. It prevents from building components in which a large number of variables would be selected. This lower bound for $\lambda$ is implemented in the package.

## 4.6 | A toy example

Consider the latent variable model (Equation (2)) with the following "easy" data structure

$$\mathbf{A} = \sqrt{1-\sigma^2}\left(\mathbf{1}'_{50}\ \mathbf{0}'_{950}\right), \mathbf{D} = \sqrt{1-\sigma^2}\left(1\right), \tag{28}$$

where $1-\sigma^2 = 0.9025$ and

$$\boldsymbol{\psi} = \left(\phi', \varepsilon'_{1\ldots 50}/\sigma, \varepsilon'_{51\ldots 1000}, \xi/\sigma\right)' \sim \mathcal{N}\left(\mathbf{0}, \mathbb{I}_{1+1000+1}\right).$$

The dependent variable $\mathbf{y}$ is only associated with the first 50 variables of $\mathbf{x}$ while the last 950 variables of $\mathbf{x}$ are not linked to $\mathbf{y}$. According to Remark 1, the true value for the number of components is $R = 1$. Thus, in this simulation framework, one of the objectives of the ddsPLS method is to properly retain one component for which the first 50 variables of $\mathbf{x}$ are selected in the final model. Moreover the corresponding prediction squared error is expected around "$\gamma = 1 - \sigma^2 = 0.9025$," for a sample with a reasonable number $n$ of observations.

Figure 1 gathers results of three simulations based on three sample sizes, respectively, $n = 50{,}100{,}200$. For the bootstrap-based selection criterion, $B = 50$ bootstrap samples are used. For each of these samples, the proposed algorithm keeps only one component. The three graphs at the top of Figure 1 provide the plot of $\widehat{R}^2_{B,1}$ and $\widehat{Q}^2_{B,1}$ versus $\lambda$, for each sample size. The "optimal" $\widehat{\lambda}_1$'s (for which $\overline{R}^2_{B,1} - \overline{Q}^2_{B,1}$ are minimum) are also located in these graphs. One might notice that selecting model considering only $\overline{Q}^2_{B,1}$ would lead to bad conclusions, where its maximum (see the colored star on the curve of $\overline{Q}^2_{B,1}$) is around $\lambda = 0.8$ (for $n = 50$ or 200) or 0.3 (for $n = 100$) which is an over-fitting region according to the difference $\overline{R}^2_{B,1} - \overline{Q}^2_{B,1}$.

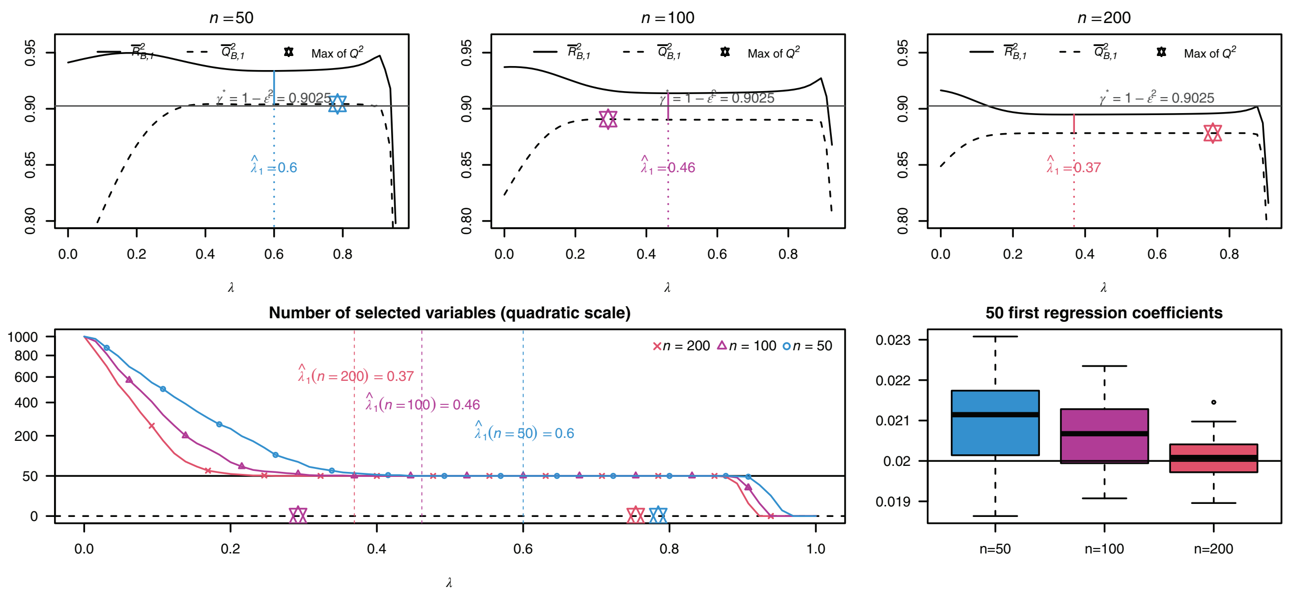


FIGURE 1 Simulation results for the toy example

The graph at the bottom left of Figure 1 shows the number of selected variables for each value of $\lambda$ and each of the three samples ($n = 50$, 100, and 200). These curves are decreasing until a first loss of the 950 non-informative variables of $\mathbf{x}$ which occurs naturally at lower $\lambda$ values when $n$ is large. Then one can observe a large plate, and finally, the curves become decreasing again, resulting in the loss of the 50 relevant variables of $\mathbf{x}$ in the model. Indeed, as $n$ increases, the minimum correlation to remove the 950 non-informative variables of $\mathbf{x}$ is small, and the maximum correlation keeping the 50 informative variables is high. For the three selected models (with the optimal $\widehat{\lambda}_1$ values), only the 50 relevant variables of $\mathbf{x}$ are selected. The right bottom graph of Figure 1 shows boxplots of the corresponding first 50 regression coefficients for the three sample sizes. As the variables share the same distribution and are independent, the regression coefficient associated with the first 50 variables of $\mathbf{x}$ should naturally be close to $1/50$, while the last 950 coefficients regression coefficient associated with the last 950 variables of $\mathbf{x}$ are equal to 0. One might notice from the three boxplots, the coefficients of the first 50 relevant variables get closer to this expected value $1/50$ as $n$ increases.

In the following, let us define the true positive rate (TPR) and the false positive rate (FPR) of variable (of $\mathbf{x}$) selection as $TPR = \mathrm{TP}/\mathrm{P}$ and $FPR = \mathrm{FP}/\mathrm{N}$ where TP is the number of selected variables among those that should be selected, P is the total number of variables that should be selected, FP is the number of selected variables among those that should not be selected, and $N$ is the total number of variables that should not be selected. Naturally, the optimal target is to have $TPR = 1$ and $FPR = 0$. In this simulation study, the TPR and FPR of variables of $\mathbf{x}$ selection are optimal for the three sample sizes. Note that the TPR and FPR are also optimal for the variable of $\mathbf{y}$ selection since the one-dimensional dependent variable is always retained in the final selected models.

Finally, according to Equation (6), the metrics $||\mathbf{A}\widehat{\mathbf{B}} - \mathbf{D}||^2/||\mathbf{D}||^2$ can be calculated in this simulation framework: its value is approximately equal to $0.002 \ll 1$ and so shows reliable results.

This "toy" example illustrated the principle of the selection criterion for the ddsPLS method. In Section 5, others simulations will lead to estimate its good numerical behavior in more complex contexts.

## 5 | SIMULATION STUDY

Different synthetic structures have been used to evaluate the numerical behavior of the proposed ddsPLS methodology. Since the multiblock framework can be reduced to a monoblock framework, only mono-block structure in the $\mathbf{x}$ part are considered in this simulation study via three specific designs.

The first design corresponds to two groups of variables in the $\mathbf{x}$ part, which are not correlated, and a three-dimensional response variable $\mathbf{y}$. The second design simulates spectroscopic data and will be detailed below.

For each design, 100 replications will be considered.

In this simulation study, various methodologies are compared with the proposed "**ddsPLS**" approach. The different methodologies are based on what has already been proposed by other authors and adapted to the bootstrap versions of $\gamma$. More precisely,

- "**sPLS 1**", "**sPLS 2**," and "**sPLS 3**" build sparse PLS models as detailed in Ref. [11]. The additional notation "**1**" refers to the solution minimizing the difference $\overline{R}^2_{B,r} - \overline{Q}^2_{B,r}$ for each component $r$. The notation "**2**" refers to the solution maximizing $\overline{Q}^2_{B,r}$ for each component $r$, which is the closest solution to what is done currently using this model, corresponding to minimizing the cross-validation error (here replaced with maximizing the out-of-bag normalized error). Finally the notation "**3**" refers to model which builds variables while there exists $(\text{keep}_X, \text{keep}_Y)$ such as $\overline{Q}^2_{R,(\text{cum})} > 0.0975$ (based on CV results) where the couple $(\text{keep}_X, \text{keep}_Y)_r$ is then choosing maximizing $\overline{Q}^2_{R,(\text{cum})}$. This is what the most popular methodology for this method, see for example [19].
- "**NIPALS-PLS**" corresponds to classical PLS algorithm where the number of components is chosen minimizing $\overline{R}^2_{B,r} - \overline{Q}^2_{B,r}$ for each component.
- "**SPLS**" builds sparse PLS models as detailed in Ref. [4]. The parameter $\kappa$ is left to the value "$\kappa = 0.5$" (knowing that $\kappa \in [[0,1]]$) which provides a balanced compromise between sparsity (low $\kappa$) and prediction results close to the classical PLS problem (high $\kappa$). Using a grid of values corresponding to $\eta = \{0.1,0.2, \dots ,0.9\}$ and $R = \{1, 2, \dots , 10\}$, the considered best model corresponds to the minimum of the mean squared prediction error.

For "**SPLS**" and "**sPLS 3**," the maximum number of possible components is fixed to 10. Also, for those two methodologies, the metric $Q^2$, based on the $F$-folds cross-validation, is used, with a number $F$ of folds equal to $F = (25, 50, 100, 100, 100)$ for $n = (25, 50, 100, 200, 400)$ respectively. Note that, when $n = (25, 50, 100)$, leave-one-out cross-validation is performed.

For the other approaches ("**sPLS 1**," "**sPLS 2**," "**NIPALS-PLS**," and "**ddsPLS**"), Bootstrap versions of $Q^2$ or $R^2$ metrics are used with a number $B$ of bootstrap samples adapted to the sample size $n$: more specifically $B = (1000, 500, 300, 100, 100)$ for $n = (25, 50, 100, 200, 400)$, respectively.

## 5.1 | Design 1: Simulated data

The first design considers

$$\mathbf{A} = \sqrt{1-\sigma^2}\begin{pmatrix} \alpha_3 \mathbf{1}_{3,50} & \mathbf{0}_{3,50} & \mathbf{0}_{3,900} \\ \mathbf{0}_{2,50} & \alpha_2 \mathbf{1}_{2,50} & \mathbf{0}_{2,900} \end{pmatrix}_{(5,1000)}$$

$$\text{and} \quad \mathbf{D} = \sqrt{1-\sigma^2}\begin{pmatrix} \alpha_3 \mathbf{1}_{3,1} & \mathbf{0}_{3,1} & \mathbf{0}_{3,1} \\ \mathbf{0}_{2,1} & \alpha_2 \mathbf{1}_{2,1} & \mathbf{0}_{2,1} \end{pmatrix}_{(5,3)}, \tag{29}$$

where $\forall k \in \mathbb{N}^\star$, $\alpha_k = 1/\sqrt{k}$, $\sqrt{1-\sigma^2} = 0.99$ and

$$\boldsymbol{\psi} = \left(\phi', \varepsilon'_{1\dots100}/\sigma, \varepsilon'_{101\dots1000}, \xi'_{1\dots2}/\sigma, \xi_3\right)' \sim \mathcal{N}\left(\mathbf{0}, \mathbb{I}_{5+1000+3}\right).$$

The corresponding latent variable model is then:

$$x_j = \begin{cases} \sqrt{1-\sigma^2}\left(\phi_1+\phi_2+\phi_3\right)/\sqrt{3} + \varepsilon_j & \text{for } j = 1 \dots 50 \\ \sqrt{1-\sigma^2}\left(\phi_4+\phi_5\right)/\sqrt{2} + \varepsilon_j & \text{for } j = 51 \dots 100 \\ \varepsilon_j \quad \text{for} \quad j = 101 \dots 1000 \end{cases}$$

$$\text{and} \begin{cases} y_1 = \sqrt{1-\sigma^2}\left(\phi_1+\phi_2+\phi_3\right)/\sqrt{3} + \xi_1 \\ y_2 = \sqrt{1-\sigma^2}\left(\phi_4+\phi_5\right)/\sqrt{2} + \xi_2 \\ y_3 = \xi_3 \end{cases} \tag{30}$$

An analysis of matrices **A** and **D** described in (29) shows that $\mathcal{R} = 5$, but there are multicolinearities in **A** and the real objective for a relevant number of components might be $R = 2$, which is proved according to Ref. [8], as noticed in Remark 1. Also, only 100 variables over the 1000 variables of the variable **x** should be selected, moreover two out of the three variables of the dependent variable **y** should be selected. The theoretical $\gamma$ is equal to $\gamma^* = 2\left(1-\sigma^2\right)/3 = 0.6534$ according to (19).

According to Figure 2 and for large $n$ (=200 or 400), all the methodologies almost always build the correct number of components, with the exception of the "**SPLS**" approach. For low $n$ (=25, 50, or 100), the "**NIPALS-PLS**" and "**sPLS 3**" methodologies mostly build far too many components, which is also the case for the "**sPLS 2**" and "**SPLS**" methodologies but in a mitigate manner with median values equal to 3. The "**ddsPLS**" methodology and "**sPLS 1**" almost always build the correct number of components with a clear advantage for the first-mentioned method.

According to Figure 3, when $n = 400$ (large sample size), all the methodologies correctly provide "$Q^2$" values close to $\gamma^*$. For all the cases, "**ddsPLS**" underestimates the value of $Q^2$. When $n = 25$, the bootstrap based versions of $Q^2$ seem to under-estimate $\gamma^*$ while the "**sPLS 3**" over-estimates it and "**SPLS**" shows high variability around this value. For "**NIPALS-PLS**," "**sPLS 3**," and "**SPLS**" methodologies, the bad performances are due to over-fitting because of the high values retained for $R$ for these approaches. However, the proposed $Q^2$ metric, $\overline{Q}^2_B$, used for "**NIPALS-PLS**" (and for "**sPLS 1**"

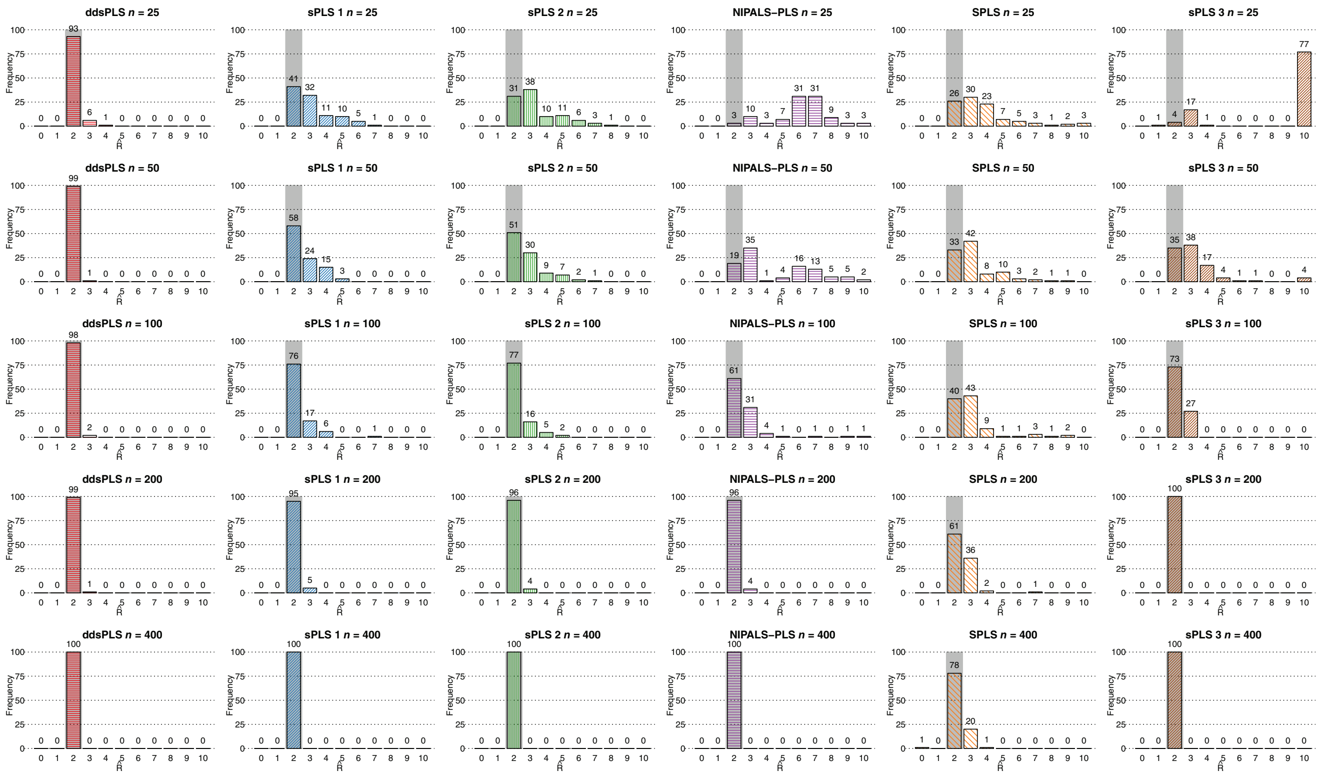


FIGURE 2 Number of components built for design 1, gray areas correspond to the objective, that is, $R = 2$ components

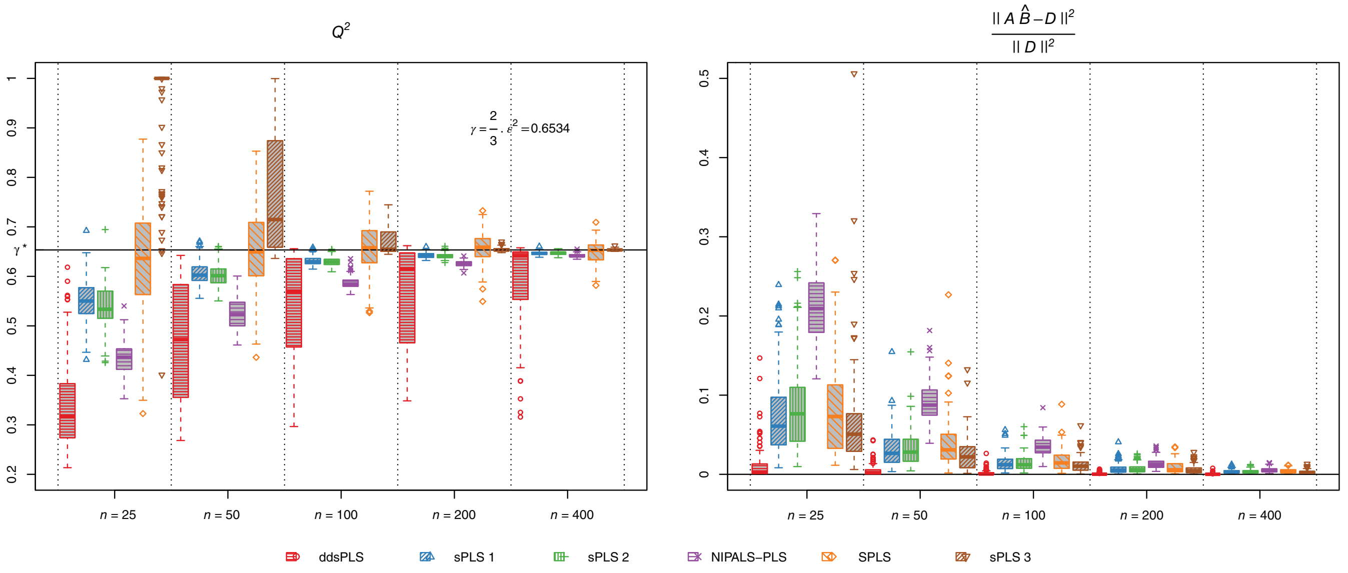


FIGURE 3 Main performance metrics for design 1. The horizontal line represents the value of $Q^2 = \gamma^\star = 2\epsilon^2/3 \approx 0.6534$ which corresponds to the objective, where $\epsilon = \sqrt{1-\sigma^2}$

and "**sPLS 2**"), detects over-fitting providing low $Q^2$ values, while "**sPLS 3**" provides high values of $Q^2$ which are often larger than the true expected value, this being essentially due to how $Q^2$ is calculated and not representative of the actual quality of the model. Finally, the "**sPLS 1**" and "**sPLS 2**" methodologies show slightly better results than "**ddsPLS**" in that low $n$ context. Even if this metric seems worse for "**ddsPLS**," this might be subject to over-fitting and other descriptors (such as $||\mathbf{A}\hat{\mathbf{B}} - \mathbf{D}||^2/||\mathbf{D}||^2$, or a prediction error on an independent data-set) should be used. These results have been observed and are accessible on the right part of the

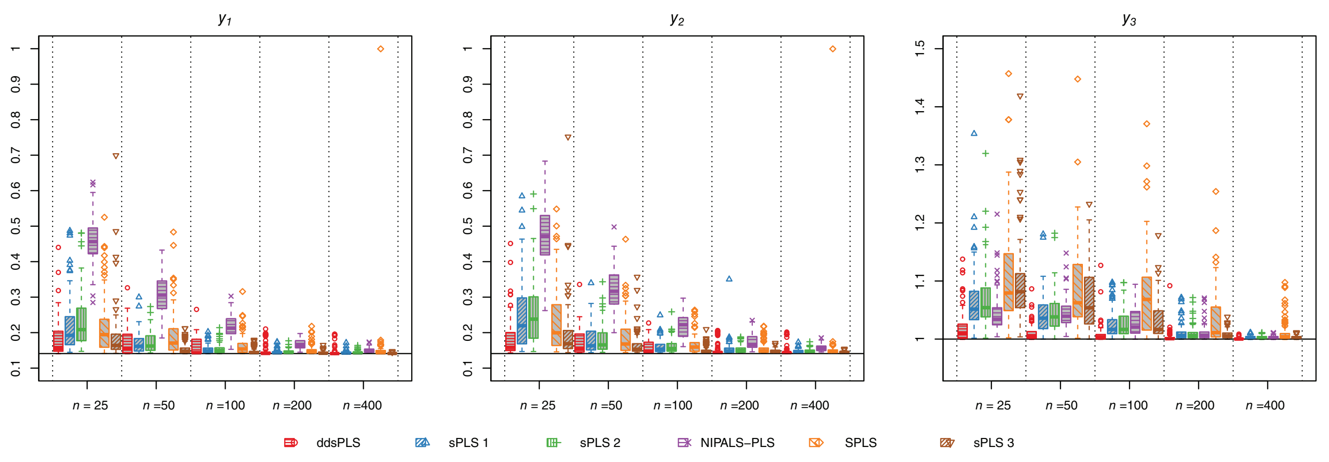


**FIGURE 4** Scaled RMSE for design 1, for each response variable, on an independent test sample of size $n_{\text{test}} = 1000$. Horizontal lines represent the objective values ≈ (0.141,0.141,1)

Figure 2 and in the Figure 3. Those figures are commented below.

In Figure 3, from the boxplots of $||\mathbf{A}\widehat{\mathbf{B}} - \mathbf{D}||^2/||\mathbf{D}||^2$, for each methodology and each sample size, the methodologies ranked in descending order of performance are: "**ddsPLS**," "**sPLS 3**," "**SPLS**," "**sPLS 1**," "**sPLS 2**," and "**NIPALS-PLS**" for low $n$. As $n$ increases, this order is maintained and all the methodologies get better results for this metric.

We define the scaled root mean-squared error (RMSE) such as

$$\forall j \in [\![1, q]\!],\ \text{scRMSE}_j = \sqrt{\frac{\sum_{i=1}^{n_{\text{test}}} \left(y_{i,j} - \widehat{y}_{i,j}\right)^2}{\sum_{i=1}^{n_{\text{test}}} \left(y_{i,j} - \bar{y}_j\right)^2}},$$

estimated on an independent sample of size $n_{\text{test}}$ where $\bar{y}_j$ is estimated on the independent test sample and $\widehat{y}_{i,j}$ is estimated according to the current prediction model. The Figure 4 shows the scaled RMSE for the different methodologies in all cases, for each response variable separately. Methodologies "**ddsPLS**" and "**sPLS 3**" show the best performances for low $y_1$ and $y_2$ while "**ddsPLS**" and "**NIPALS-PLS**" are the best for $y_3$. More interestingly, "**sPLS 3**" and "**SPLS**" are especially not precise for $y_3$, this is associated with the remark formulated in the previous paragraph associated with the potential over-fitting of those methodologies.

In terms of variable selection, the Figure 5 insures that the TPR of variable of $\mathbf{x}$ selection is mostly equal to 1 for all methodologies, except for "**ddsPLS**." On the contrary, the FPR is mostly equal to 0 only for "**ddsPLS**" and is closely equal to 0 for "**SPLS**." Other methodologies based on "sPLS" decrease their FPR with $n$ until $n$ reaches 100. More precisely, the FPR of "**sPLS 1**" indeed slowly decreases with $n$, "**sPLS 2**" and "**sPLS 3**" increase their FPR for $n \geq 100$. This shows that $Q^2$ maximization based methods (both for cross-validation and bootstrap versions) seem to build sparser models as the number $n$ of individuals grows. On the contrary, $Q^2 - R^2$ minimization based methods decrease their FPR as $n$ grows.

In the $\mathbf{y}$ part, regardless of $n$, all methodologies select always $y_1$ and $y_2$, and so the FPR equal 1 in all cases. According to Figure 5B, all methodologies naturally decreases the selection rate of $y_3$ as $n$ increases, except for "**sPLS 3**." Furthermore, among the bootstrapped methodologies, "**ddsPLS**" outperforms the others.

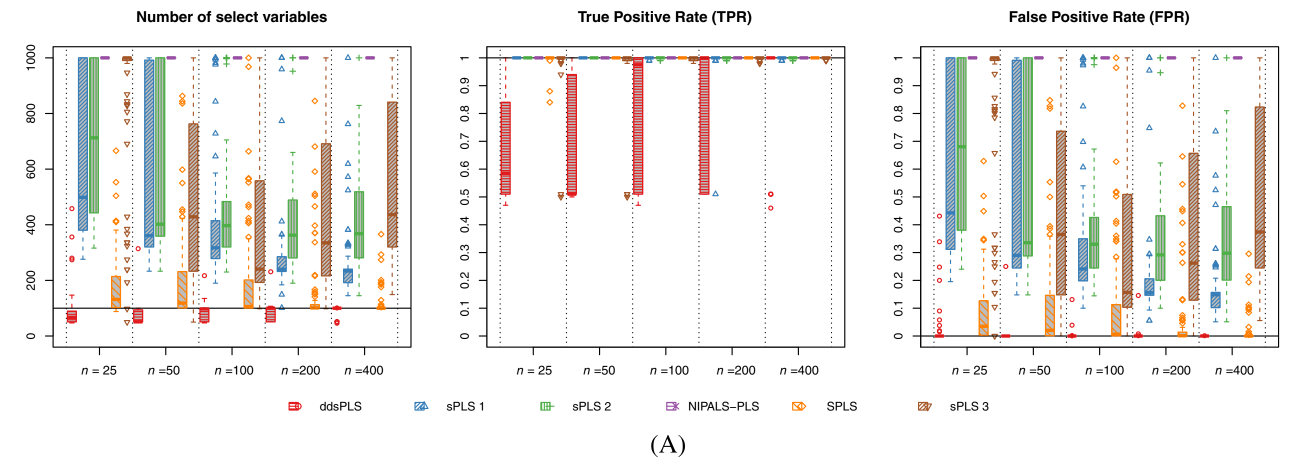
In terms of variable selection, this can lead to the conclusion that bootstrap-based versions provide better results than those based on cross-validation, and more precisely the one associated with the minimization of $\overline{R}^2_{B,r} - \overline{Q}^2_{B,r}$ which perform better than those based on the maximization of $\overline{Q}^2_{B,r}$, for both $\mathbf{x}$ and $\mathbf{y}$ parts.

## 5.2 | Design 1 with varying $q$

As to evaluate the performance of the proposed methodology if $q$ varies, it has been considered to add extra response variables, as to reach $q = \{10, 50, 100\}$. Each of these extra response variables is not associated nor with the other response variables nor with the other response variables.

As previously, $N = 100$ different data sets (with $n = 50$) have been generated to build different prediction models. Different information can be highlighted.

- All the models are built on two components.
- All the models select, in the response part, only the two relevant variables (the two first ones).
- The first row of Figure 6 gives the scaled RMSE of the $N$ models, for each scenario and for each response



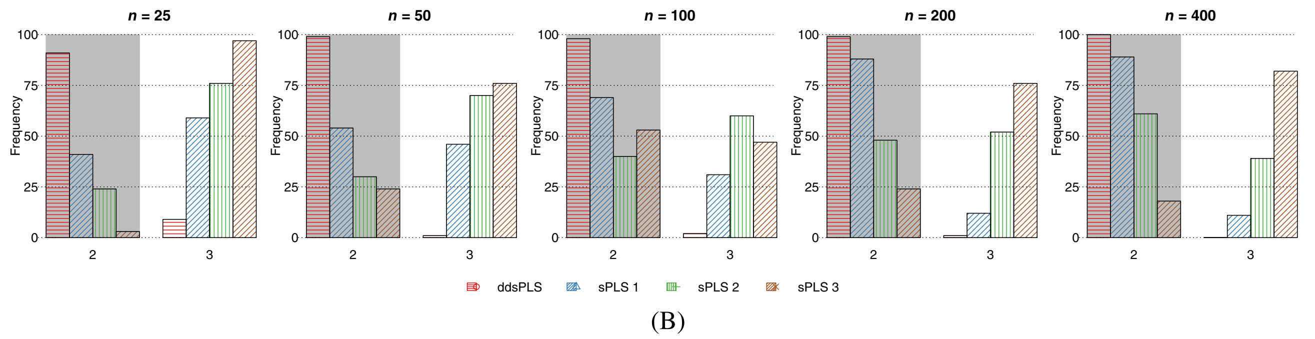


FIGURE 5 Variable selection performances in the **x** and **y** parts for design 1. (A) Variable selection performances in the **x** part. (B) Variable selection performance in the y part measured by the number of selected **y** variables where the gray areas correspond to theoretical objectives, that is two selected variables: $y_1$ and $y_2$ (which are always selected, providing a TPR equal to 1)

variable. An independent data set of size $n_{\text{test}} = 1000$ has been used to estimate those distributions. It is clear that all the models predict well the two selected variables ($y_1$ and $y_2$) and do not "predict" for others but returns the empirical mean values.

- The second row of Figure 6 shows the frequency of selection in the covariate part for each scenario. Only the 100 first covariates, associated with $y_1$ and $y_2$, are selected.

This experiment shows that the proposed "**ddsPLS**" methodology is robust to an increasing number $q$ of response variables.

### 5.3 | Design 2: simulated data inspired by real data

In this design, datasets with structures similar to spectroscopic data have been generated. Using the routine already used by Cloarec [5], the **x** part contains spectra observations and is divided into two blocks. The first one ($\mathbf{X}_1$) corresponds to spectroscopy with large overlapping peaks and a low number of variables, such as Ultraviolet, Visible and Near-Infrared spectrophotometry. The second block ($\mathbf{X}_2$) has narrow peaks and a high number of variables but with much less overlap between peaks. The blocks $\mathbf{X}_t, t = 1, 2$ are built using the following procedure:

$$\mathbf{X}_t = \sqrt{1-\sigma_t^2}\mathbf{C}_t\mathbf{S}_t + \sigma_t\mathbf{E}_t \tag{31}$$

with

$$\mathbf{S}_t = \left[\left(s_{t,r,j}\right)_{(r,j)}\right] \in \mathbb{R}^{\mathcal{R}\times p_t}, \quad \mathbf{C}_t = \left[\left(c_{t,i,r}\right)_{(i,r)}\right] \in \mathbb{R}^{n\times\mathcal{R}}$$
$$\text{and } \mathbf{E}_t = \left[\left(\mathbf{e}'_{t,i}\right)_i\right] \in \mathbb{R}^{n\times p_t}$$

where

$$\forall (r,j) \in [\![1,\mathcal{R}]\!] \times [\![1,p_t]\!], \quad s_{t,r,j} = \sum_{k=1}^{K_r} h_{r,k} \exp\left\{-\left(j-\pi_{r,k}\right)^2 / \left(2\omega_{r,k}^2\right)\right\},$$
$$\mathbf{e}_{t,i} \sim \mathcal{N}\left(\mathbf{0}_{p_t}, \mathbb{I}_{p_t}\right) \text{ and } \sigma_t = 0.05.$$

The score matrix is here denoted $\mathbf{C}_t$ for "**C**oncentration", and the matrix of the weights is denoted $\mathbf{S}_t$ for "**S**pectrum." Let $p_t$ be the number of variables in each block $t$ and $\mathcal{R}_t$ the

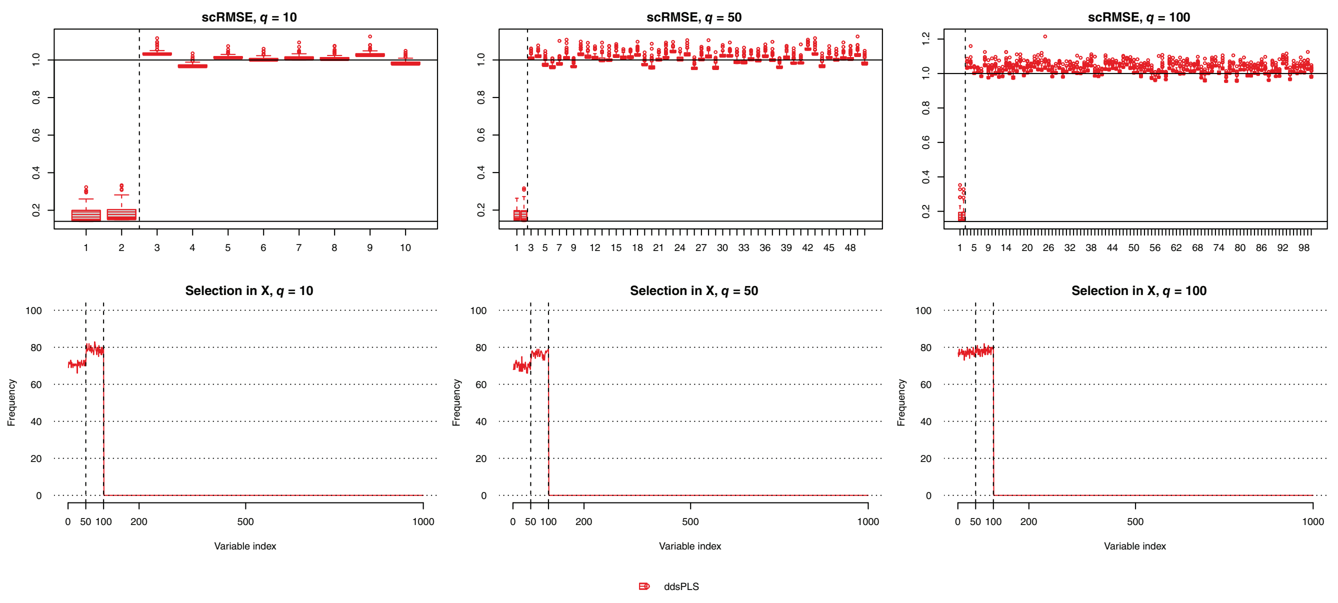


FIGURE 6 Scaled RMSE for each response variable and selection frequencies in the covariate part for the three scenarios $q = \{10, 50, 100\}$

number of latent variables ($\mathcal{R}_t = 10$ for both blocks). For each latent variable, $r \in [[1, \mathcal{R}_t]]$, the number $K_r$ of peaks is selected randomly from a uniform discrete random distribution $\mathcal{U}[[1,5]]$ in order to generate, $\forall k \in [[1, K_r]]$, a spectrum with the following characteristics:

- position of the peaks is sampled from a discrete uniform distribution $\pi_{r,k} \sim \mathcal{U}\left[[2, p_t]\right]$;
- relative peak heights are sampled from a continuous uniform distribution $h_{r,k} \sim \mathcal{U}(0,1)$;
- peak widths are sampled from a continuous uniform distribution $\omega_{r,k} \sim \mathcal{U}(2,3)$ and are proportional to the number of variables.

The previous parameters allow to build the matrix $\mathbf{S}_t$. To build the matrix $\mathbf{C}_t$ such as the intensity of each latent variable in each observation, the following Gaussian distribution is used: $c_{t,i,r} \sim \mathcal{N}\left(\mu_{t,r}, \sigma_{t,r}^2\right)$ where

- the expected intensity of each latent variable is randomly sampled from an exponential distribution: $\log\left(\mu_{t,r}\right) \sim \mathcal{U}(0,1)$,
- the variance of the intensity for each latent variable is randomly sampled from a uniform distribution $\sigma_{t,r}^2 \sim \mathcal{U}\left(0, \sigma_\mu \mu_{t,r}\right)$ where $\sigma_\mu$, when low enough, allows to generate positive values for $c_{t,i,r}$ with high probability.

Moreover, in order to get the correlation between the two blocks $\mathbf{X}_1$ and $\mathbf{X}_2$, the first five columns of $\mathbf{C}_2$ have been replaced by the first five columns of $\mathbf{C}_1$. Figure 7 provides a visualization of an extract from the $\mathbf{x}$ part for a dataset simulated according to the procedure.

The five-dimensional $\mathbf{y}$ part is generated from the model $\mathbf{Y} = \sqrt{1-\sigma_y^2}\mathbf{C}_y + \sigma_y \mathbf{E}_y$, where each row of $\mathbf{E}_y$ follows $\mathcal{N}(\mathbf{0}, \mathbb{I})$, $\sigma_y = 0.2$ corresponds to the standard deviation of this observation noise, and the matrix $\mathbf{C}_y$ is used to manage the link with the $\mathbf{x}$ part. More precisely, the first three columns of $\mathbf{C}_y$ are the standardized first three columns of $\mathbf{C}_1$. The last two columns of $\mathbf{Y}$ are generated from a standardized normal distribution. Hence only the first three variables of the $\mathbf{y}$ part are related to the $\mathbf{x}$ part.

According to the way the spectra are generated, the objective number $R$ of components of the underlying model is equal to three with a high probability.

Figures 8A,B respectively show, for 100 replications, the number $\widehat{R}$ of components retained in each model and the associated number of selected variables in the $\mathbf{x}$ part. Since the correct dimension is equal to $R = 3$, all the methodologies (apart from "**NIPALS-PLS**", which builds far too many components) succeed to get most of times models with three components, and only "**ddsPLS**" has a clear majority of models built on three components. Moreover, according to Figure 8B, "**sPLS 1**" and "**sPLS 2**" build often non-sparse models with very large numbers of selected variables. "**sPLS 3**" also builds models with various numbers of selected variables and also builds non sparse models. The methodologies "**ddsPLS**" and "**SPLS**" build the sparsest models, with a clear advantage for the first methodology even if "**ddsPLS**" sometimes builds non sparse models. Note that these graphs

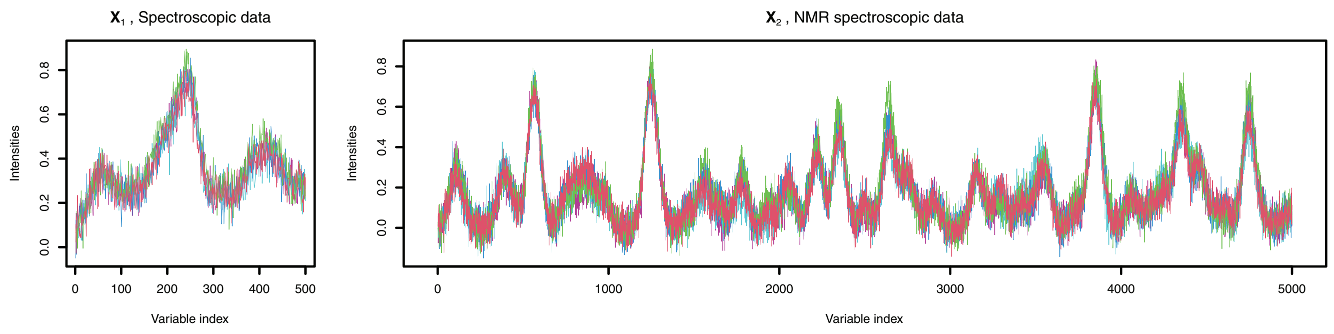


**FIGURE 7** First five observations of simulated datasets $\mathbf{X_1}$ and $\mathbf{X_2}$ from design 2

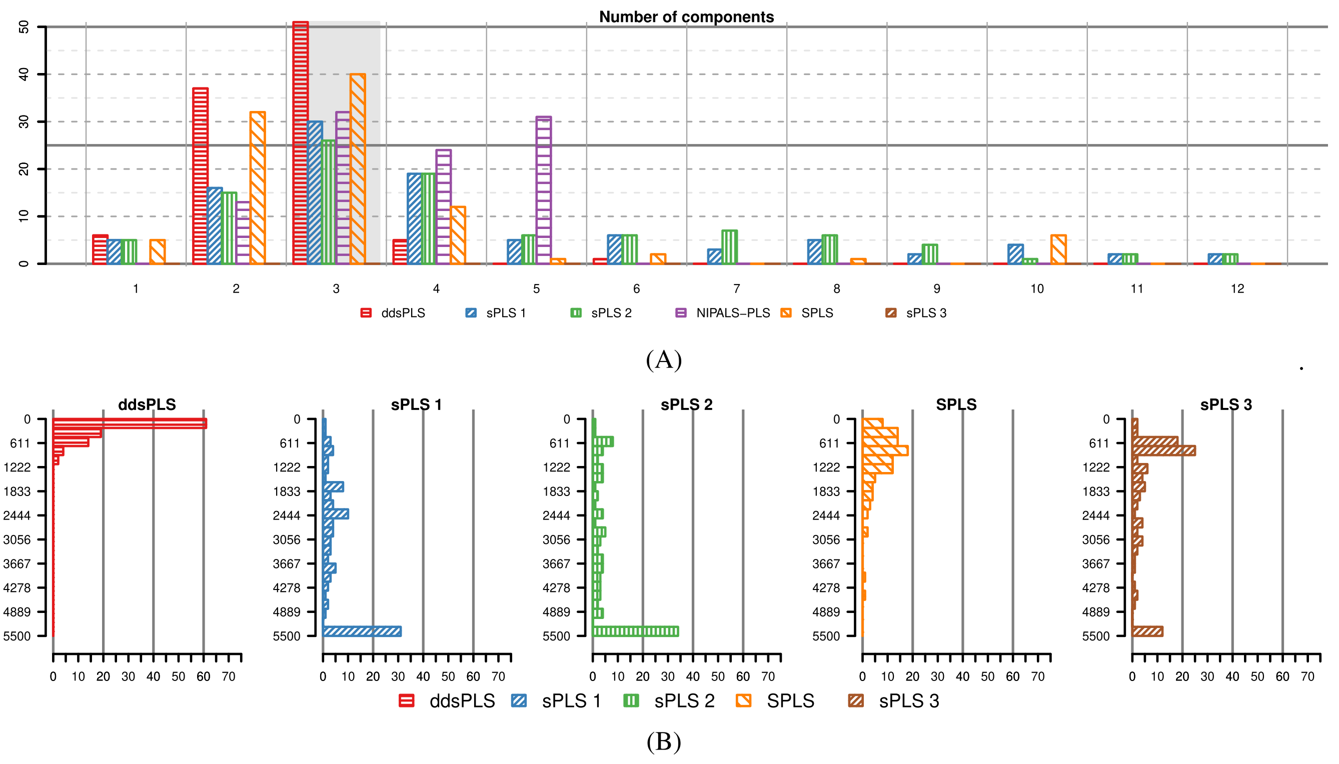


**FIGURE 8** Estimated number of components $\widehat{R}$ (top) and number of selected variables (bottom) for design 2. (A) Number $\widehat{R}$ of components built for each methodology for design 2. Gray area corresponds to the correct value $\boldsymbol{R} = 3$. (B) Number of selected variables in the **x** part for each methodology for design 2

do not make it possible to appreciate the quality of the selection procedure and the corresponding estimated model different methodologies; this point is discussed below.

Since the simulated spectra are not sparse, it is not possible to directly evaluate the relevance of the variable selections. However, the three spectra, which are common to $\mathbf{X}_1$, $\mathbf{X}_2$, and $\mathbf{Y}$ are known in this simulation framework. It is reasonable to assume that the relevance of a variable is associated with the intensity of this variable in these three spectra of interest. Then, the selected variables are described in terms of TPR and FPR by varying a threshold $\alpha$ over the maximum values of the spectra for each variable along with the three spectra of interest as to build receiver operating characteristic (ROC) curves and so create a common score of selectivity corresponding to the area under the curve (AUC). For a relevant analysis, a common distribution of $\alpha$ for all methodologies has been used.

The top right graph of the Figure 9 shows the AUC metrics computed for the models built over the 100 replications of datasets. It allows evaluating the performance of the methodologies in terms of variable selection. It seems that "**sPLS 1**" and "**sPLS 2**" are the least efficient methodologies to select the correct variables efficiently. The methodologies "**sPLS 3**" and "**SPLS**" get better AUC

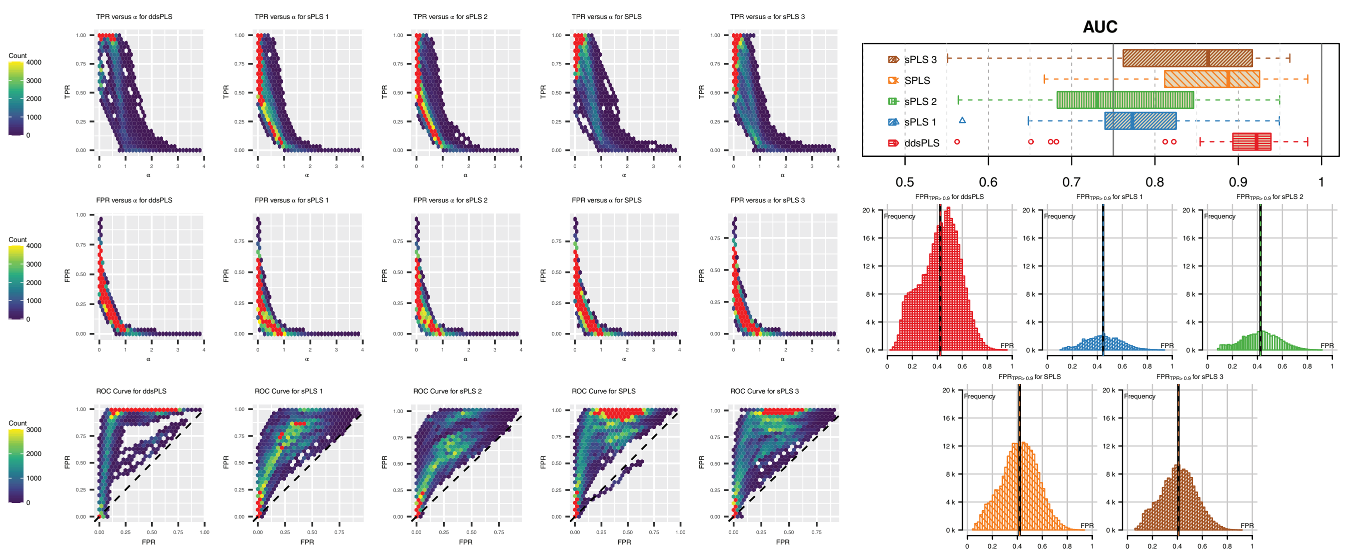


**FIGURE 9** Variable selection performances for design 2. Top left graphs: density plots for the five competing methodologies (in columns) of the TPR versus $\alpha$ (first row), of the FPR versus $\alpha$ (second row) and of FPR versus TPR (third row), red areas correspond to densities higher than 4000 for the first two rows and 3000 otherwise. Top right graph: boxplots of AUC for the five competing methodologies. Bottom right graphs: histograms of FPR given TPR > 0.9 for the five competing methodologies

but keep high variability. Finally, the proposed "**ddsPLS**" approach provides the best AUC.

Top left graphs (density plots) of the Figure 9 detail more precisely the quality of the selection procedure of the five competing methodologies (in columns), via the TPR versus $\alpha$ (first row), the FPR versus $\alpha$ (second row) and the FPR versus the TPR (third row).

- From the first row, one might notice that "**sPLS 1**" and "**sPLS 2**" present a huge loss of TPR for very low $\alpha$ and do not succeed to keep TPR close to 1 for low values of $\alpha$. "**sPLS 3**" also suffers from a fall although less important and even succeeds to keep TPR close to 1 for low values of $\alpha$. "**SPLS**" keeps this tendency, and "**ddsPLS**" performs even better.
- According to the second row, it seems that "**sPLS 1**" and "**sPLS 2**" quickly (in terms of $\alpha$) fall in FPR, which is encouraging in a certain way, while the three other methodologies keep it high for low values of $\alpha$. This actually, quite paradoxically, allows considering that "**ddsPLS**," "**SPLS**," and "**sPLS 3**" are, in fact, better methodologies because they do not tend to remove too quickly variables; feature that is often criticized in sparse methods.
- The third row shows the ROC curves in density and once again confirms the previous observations: "**ddsPLS**," "**SPLS**," and "**sPLS 3**," in decreasing order of performance, outperform the other methodologies since the high densities are concentrated at the top of their respective plot, corresponding to high TPR. Unfortunately, the precision is not sufficient to allow a correct differentiation of each distribution for high TPR's.

The five histograms in Figure 9 (bottom right graphs) show the conditional distribution of the FPR given $TPR >$ 0.9. "**ddsPLS**" clearly provides the heaviest distribution in that TPR area and gets the heaviest lower distribution tail, which are the two reasonable objectives of a selection method.

In terms of $Q^2$ results (not provided here), "**ddsPLS**," "**SPLS**," "**sPLS 1**," and "**sPLS 2**" are the most efficient methodologies (respectively, equal to 0.47, 0.46, 0.47, and 0.43 in median for the all simulated datasets) for $\hat{R} = 3$ components built models. Note that those four methodologies get lower $Q^2$ than expected ($\gamma = 3\epsilon_y^2/5 = 0.597$ in the considered simulation framework). Only "**sPLS 3**" gets higher median for $Q^2$, equal to 0.55. For more complex models built with $\hat{R} > 3$ components, the results also show that "**sPLS 3**" give a median $Q^2$ equal to 0.64 for $\hat{R} = 4$, 0.75 for $\hat{R} = 5$, 0.93 for $\hat{R} = 6$, 0.91 for $\hat{R} = 8$ and 0.99 for $\hat{R} = 10$, while the four other methodologies get decreasing values for median $Q^2$. Finally, in the context of that simulation design, it seems again that "**sPLS 3**" does not succeed to beat over-fitting.

Concerning the selection in the **y** part (which is only available for "**ddsPLS**," "**sPLS 1**," "**sPLS 2**," and "**sPLS 3**" methodologies), the following results were obtained. Below, the notation $(y_j, N_j)$ means that variable $y_j$ has been selected $N_j$ times over the 100 replications:

- “**ddsPLS**”: $(y_1, 87), (y_2, 87), (y_3, 81), (y_4, 18), (y_5, 19)$,
- “**sPLS 1**”: $(y_1, 83), (y_2, 86), (y_3, 74), (y_4, 37), (y_5, 36)$,
- “**sPLS 2**”: $(y_1, 88), (y_2, 84), (y_3, 82), (y_4, 53), (y_5, 51)$,
- “**sPLS 3**”: $(y_1, 89), (y_2, 91), (y_3, 81), (y_4, 32), (y_5, 33)$.

It is clear that “**ddsPLS**” outperforms other methodologies regarding the FPR’s in $\mathbf{y}$ part selection, while the TPR’s appear to be equivalent for the four methodologies.

## 6 | CONCLUSION

The proposed ddsPLS methodology is a sparse PLS method, both in $\mathbf{x}$ and $\mathbf{y}$, based on soft-thresholding using bootstrap operations for model selection. This solution fixes sparsity in both $\mathbf{x}$ and $\mathbf{y}$ parts using a single parameter, which is also interpretable in terms of minimal empirical covariance allowed to be selected in the model. Moreover this solution shows very good results both in prediction and in selection, through the various simulation models analyzed.

While most of the other sparse PLS are based on cross-validation approaches, our solution uses bootstrap. This allows building smooth curves of the chosen criterion even in the ill-conditioned contexts of low $n$ and large $p$ data.

Based on the proposed bootstrap version of sparse PLS, a new relevant definition of the $Q^2$ and the $R^2$ statistics denoted as $\overline{Q}_B^2$ and $\overline{R}_B^2$ where $B$ is the number of bootstrap samples, has been introduced. Those new versions of $Q^2$ and $R^2$ do not mix the errors of the different folds but aggregate the fold estimations through the bootstrap mean estimator.

Different aspects must be highlighted regarding those new statistics. The newly introduced $\gamma^\star$ is a theoretical limit for the $Q^2$ and $R^2$ statistics that our criterion respects while the tested competing methods violate it in the $n << p$ context even for simple simulation schemes. The chosen criterion $\overline{R}_B^2 - \overline{Q}_B^2$ allows to efficiently select sparse models.

Finally, the ddsPLS methodology can be applied to different data contexts such as biological data, omics data and chemometrics, where the sample size $n$ is indeed much lower than the number $p$ of variables.

An implementation of the proposed ddsPLS methodology is available at https://github.com/hlorenzo/ddsPLS2. A vignette is also available to facilitate the use of the corresponding R package and the associated functions for any user not necessarily specialized in R or statistical modeling. Note that this package uses the `foreach` (v.1.5.0) package to parallelize the computations associated with the different bootstrap samples. These codes are written in C$^{++}$ thanks to the packages `Rcpp` (v.1.0.5) and `RcppEigen` (v.0.3.3.7.0) in order to reduce the computation time and to ensure a portability toward other languages, Python in particular.

What’s more, all simulation codes (regarding **Toy Example**, **Design 1**, and **Design 2**) are accessible via the repository https://github.com/hlorenzo/simulation_ddspls, where two files (**README.md** and **simulation_ddspls.html**) help to use the different simulation functions.

## ACKNOWLEDGMENTS

We thank the editor and the two anonymous referees for their comments that led to an improved version of this article. Hadrien Lorenzo received the support of “Sartorius, Corporate Research Advanced Data Analytics” for this project.

## DATA AVAILABILITY STATEMENT

Data sharing not applicable - no new data generated, or the article describes entirely theoretical research.

## ORCID

*Hadrien Lorenzo* https://orcid.org/0000-0002-0253-8539

## REFERENCES

1. P. J. Bickel and E. Levina, *Regularized estimation of large covariance matrices*, Ann. Stat. 36(1) (2008), 199–227. https://doi.org/10.1214/009053607000000758
2. C. Broc, T. Truong, and B. Liquet, *Penalized partial least squares for pleiotropy*, BMC Bioinform. 22(1) (2021), 1–31.
3. T. Cai and W. Liu, *Adaptive thresholding for sparse covariance matrix estimation*, J. Am. Stat. Assoc. 106(494) (2011), 672–684. https://doi.org/10.1198/jasa.2011.tm10560
4. H. Chun and S. Keleş, *Sparse partial least squares regression for simultaneous dimension reduction and variable selection*, J. R. Stat. Soc.: Ser. B (Stat. Methodol.) 72(1) (2010), 3–25.
5. O. Cloarec, *Can we beat over–fitting?* J. Chemom. 28(8) (2014), 610–614.
6. Y. Deshpande and A. Montanari, *Sparse PCA via covariance thresholding*, J. Mach. Learn. Res. 17(141) (2016), 1–41.
7. S. Geisser, *A predictive approach to the random effect model*, Biometrika 61(1) (1974), 101–107. https://doi.org/10.1093/biomet/61.1.101
8. I. S. Helland and T. Almøy, *Comparison of prediction methods when only a few components are relevant*, J. Am. Stat. Assoc. 89(426) (1994), 583–591.
9. I. M. Johnstone and A. Y. Lu, *On consistency and sparsity for principal components analysis in high dimensions*, J. Am. Stat. Assoc. 104(486) (2009), 682–693.
10. A. F. Kautt, C. F. Kratochwil, A. Nater, G. Machado-Schiaffino, M. Olave, F. Henning, J. Torres-Dowdall, A. Härer, C. D. Hulsey, P. Franchini, M. Pippel, E. W. Myers, and A. Meyer, *Contrasting signatures of genomic divergence during sympatric speciation*, Nature 588(7836) (2020), 106–111.

11. K.-A. Lê Cao, D. Rossouw, C. Robert-Granié, and P. Besse, *A sparse PLS for variable selection when integrating omics data*, Stat. Appl. Genet. Mol. Biol. 7 (2008), 35.
12. B. Liquet, P. L. De Micheaux, B. P. Hejblum, and R. Thiébaut, *Group and sparse group partial least square approaches applied in genomics context*, Bioinformatics 32(1) (2016), 35–42.
13. H. T. N. Martens and T. Naes, *Multivariate calibration*, John Wiley & Sons, New York, 1992.
14. T. Mehmood, S. Sæbø, and K. H. Liland, *Comparison of variable selection methods in partial least squares regression*, J. Chemom. 34 (2020), e3226. https://doi.org/10.1002/cem.3226
15. M. Pérez-Enciso and M. Tenenhaus, *Prediction of clinical outcome with microarray data: A partial least squares discriminant analysis (pls-da) approach*, Hum. Genet. 112 (2003), 581–592.
16. A. Rechtien, L. Richert, H. Lorenzo, G. Martrus, B. Hejblum, C. Dahlke, R. Kasonta, M. Zinser, H. Stubbe, U. Matschl, A. Lohse, V. Krähling, M. Eickmann, S. Becker, VEBCON Consortium, R. Thiébaut, M. Altfeld, and M. Addo, *Systems vaccinology identifies an early innate immune signature as a correlate of antibody responses to the ebola vaccine rvsv-zebov*, Cell Rep. 20(9) (2017), 2251–2261. https://doi.org/10.1016/j.celrep.2017.08.023
17. M. Stone, *Cross-validatory choice and assessment of statistical predictions*, J. R. Stat. Soc. Ser. B (Methodol.) 36(2) (1974), 111–147.
18. M. Sutton, R. Thiébaut, and B. Liquet, *Sparse partial least squares with group and subgroup structure*, Stat. Med. 37(23) (2018), 3338–3356. https://doi.org/10.1002/sim.7821
19. M. Tenenhaus, *La régression PLS: théorie et pratique*, Editions technip, Paris, 1998.
20. M. Tenenhaus, *PLS regression and PLS path modeling for multiple table analysis*, in *Proceedings in COMPSTAT, Proceedings in Computational Statistics*, J. Antoch, Ed., Physica-Verlag HD, Heidelberg, 2004, 489–499.
21. P. Bastien, V. E. Vinzi, and M. Tenenhaus, *PLS generalised linear regression*, Computational Statistics & Data Analysis. 48(1) (2005), 17–46. http://dx.doi.org/10.1016/j.csda.2004.02.005.
22. E. A. Thévenot, ropls: Pca, pls (−da) and opls (−da) for multivariate analysis and feature selection of omics data. Bioconductor, 2016.
23. R. Tibshirani, *Regression shrinkage and selection via the lasso*, J. R. Stat. Soc. Ser. B Methodol. 58 (1996), 267–288.
24. M. Vivien, Approches PLS linéaires et non linéaires pour la modélisation de multi-tableaux. Théorie et applications. Ph.D. thesis, Université Montpellier I, 2002.
25. I. N. Wakeling and J. J. Morris, *A test of significance for partial least squares regression*, J. Chemom. 7(4) (1993), 291–304.
26. J. A. Westerhuis and A. K. Smilde, *Deflation in multiblock PLS*, J. Chemom. 15(5) (2001), 485–493.
27. S. Wold, H. Martens and H. Wold, The multivariate calibration problem in chemistry solved by the PLS method. In: Kågström B., Ruhe A. (eds) *Matrix Pencils*, Lecture Notes in Mathematics, vol 973. Springer, Berlin, Heidelberg.
28. S. Wold, Three PLS algorithms according to SW (Proc. Symp. MULDAST, Multivar. Anal. Sci. Technol.), 1984, 26–30.
29. Z. S. Zoong Lwe, R. Welti, D. Anco, S. Naveed, S. Rustgi, and S. Narayanan, *Heat stress elicits remodeling in the anther lipidome of peanut*, Sci. Rep. 10(1) (2020), 22163. https://doi.org/10.1038/s41598-020-78695-3

**How to cite this article:** H. Lorenzo, O. Cloarec, R. Thiébaut, and J. Saracco, *Data-driven sparse partial least squares*, Stat. Anal. Data Min.: ASA Data Sci. J. (2021), 1–19. https://doi.org/10.1002/sam.11558