# Review of Sparse Methods in Regression and Classification with Application to Chemometrics

**Peter Filzmoser*, Moritz Gschwandtner and Valentin Todorov**

Sparse statistical methods lead to parameter estimates that contain exact zeros. This has advantages especially in the analysis of high-dimensional data, because the contribution of single variables—potential noise variables—is set to zero. We review recent proposals for sparse methods in the context of regression and classification, and compare the performance of these methods with their non-sparse counterparts, using several data examples from chemometrics.

# Review of Sparse Methods in Regression and Classification with Application to Chemometrics

**Peter Filzmoser$^{a,*}$, Moritz Gschwandtner$^{a}$ and Valentin Todorov$^{b}$**

$^{*}$ Correspondence to: P. Filzmoser
Institute of Statistics and Probability Theory, Vienna University of Technology,
Wiedner Hauptstrasse 8-10, A-1040 Vienna, Austria
E-mail:P.Filzmoser@tuwien.ac.at

$^{a}$ P. Filzmoser, M. Gschwandtner
Institute of Statistics and Probability Theory, Vienna University of Technology,
Wiedner Hauptstrasse 8-10, A-1040 Vienna, Austria
E-mail:P.Filzmoser@tuwien.ac.at; moritz.gschwandtner@chello.at

$^{b}$ V. Todorov
United Nations Industrial Development Organization (UNIDO), Vienna International Centre, P.O. Box 300, A-1400 Vienna, Austria
E-mail: V.Todorov@unido.org

The views expressed herein are those of the authors and do not necessarily reflect the views of the United Nations Industrial Development Organization.

**Abstract**

High-dimensional data often contain many variables that are irrelevant for predicting a response or for an accurate group assignment. The inclusion of such variables in a regression or classification model leads to a loss in performance, even if the contribution of the variables to the model is small. Sparse methods for regression and classification are able to suppress these variables. This is possible by adding an appropriate penalty term to the objective function of the method.

An overview of recent sparse methods for regression and classification is provided. The methods are applied to several high-dimensional data sets from chemometrics. A comparison with the non-sparse counterparts allows to getting an insight into their performance.

**Keywords:** Sparse methods, high-dimensional data, partial least squares regression, discriminant analysis, principal component analysis

# 1 Introduction

The field of chemometrics was very early faced with the problem of high-dimensional data. Spectral analysis or molecular descriptors resulted in data sets with hundreds of variables, and methods like partial least squares (PLS) regression turned out to be very useful in this context. In recent years, another area–the field of bioinformatics–came up with even more extreme data configurations, where the number of variables reaches several thousands or even more. In the analysis of gene expression data, for instance, it is then of interest to find the most important genes (variables) that allow to discriminate among two patient groups. This kind of variable selection in the high-dimensional space is a challenging task for statistics. Since traditional methods for variable selection can no longer be used because of the complexity of the data, various new techniques and methods have been developed. One important stream of methods is summarized with the term *sparse methods*. The expression "sparse" should not be mixed up with techniques for sparse data, containing many zero entries. Here, sparsity refers to the estimated parameter vector, which is forced to contain many zeros. For example, in the regression context a sparse regression method will produce a vector of estimated regression coefficients which contains many zeros and only few non-zero entries. In that way, only few variables contribute to explaining the response variable, and thus the sparse method can also be viewed as a method for variable selection.

In general, sparse methods are designed for high-dimensional data where they are particularly useful. In this contribution we focus on sparse methods in the context of linear regression and classification. The main idea of sparse methods for high-dimensional data is to reduce the noise contained in irrelevant variables for explaining a response or discriminating among groups. For example, in the context of discriminant analysis, often only few variables are relevant for the group discrimination. The discriminative power of the method would be weakened if noise variables would enter the discriminant rule. Sparse methods are setting the contribution of noise variables to zero, and thus this undesired effect is avoided. On the other hand, non-sparse methods will usually not set the contribution exactly to zero but only to a small (absolute) value. Especially in the high-dimensional case there are potentially many variables with very small contributions, but overall their influence is considerable. In total they generate a lot of noise and lower the discriminative power of the method. The same effect happens in the context of regression, where the predictive ability of a regression model containing noise variables is lower than a model containing only the variables being relevant for explaining the response.

In this article we present an overview of sparse methods for regression and classification that are relevant to the field of chemometrics. The concept of sparseness is introduced in Section 2 in the context of regression, and it is extended to discrimination problems. Section 3 presents a comparison between sparse and non-sparse methods, using a small simulation study, and several data sets from chemometrics. The final Section 4 concludes.

## 2 The Concept of Sparseness

The concept of sparseness has been introduced in the setting of the multiple linear regression model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{1}$$

where $\boldsymbol{y}$ is a vector of length $n$ representing the responses, $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ is a matrix of independent variables, and $\boldsymbol{\beta}$ is a vector of (unknown) coefficients. $\boldsymbol{\epsilon} \in \mathbb{R}^{n \times 1}$ is an error term. In ordinary least squares (OLS) regression one minimizes the sum of squares between $\boldsymbol{X\beta}$ and $\boldsymbol{y}$,

$$SS(\boldsymbol{\beta}) = \|\boldsymbol{y} - \boldsymbol{X\beta}\|_2^2,$$

with the L$_2$ norm $\|.\|_2$

$$\|\boldsymbol{v}\|_2 = \sqrt{\sum_{j=1}^{n} v_j^2}.$$

The solution for the regression coefficients is given by

$$\hat{\boldsymbol{\beta}}_{OLS} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\, SS(\boldsymbol{\beta}) = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{y}.$$

Especially for high-dimensional data, and in particular if the number of observations is smaller than the number of regressors, the inverse of $\boldsymbol{X}^\top \boldsymbol{X}$ cannot be computed, and the OLS estimator does not exist. Several solutions have been proposed to this problem, like variable selection, shrinkage methods, or methods using derived input directions [1]. The latter two methodologies are treated in the next sections.

## 2.1 Ridge Regression

If $n < p$, the matrix $\boldsymbol{X}^\top \boldsymbol{X}$ is singular and the OLS solution cannot be computed. But even in the case $n > p$ it can happen that some of the regressor variables are highly correlated, leading to *multicollinearity*, i.e. to a near singular matrix $\boldsymbol{X}^\top \boldsymbol{X}$. A consequence is that the estimated OLS regression coefficients become unstable and increase artificially in absolute number [see, e.g., 1]. This was the primary motivation for the introduction of *Ridge Regression* [2]. Instead of minimizing the sum of squared residuals, a penalized sum of squares criterion is minimized:

$$PSS_2(\boldsymbol{\beta}, \lambda) = \|\boldsymbol{y} - \boldsymbol{X\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2$$

Here we assume that $\boldsymbol{X}$ is centered, and therefore no intercept $\beta_0$ is included in $\boldsymbol{\beta}$. Otherwise, the intercept coefficient is not penalized. The last term is controlled by the penalty parameter $\lambda > 0$. The larger $\lambda$ is chosen, the more large values (in absolute number) of $\boldsymbol{\beta}$ are penalized. We say that the coefficients in $\boldsymbol{\beta}$ are shrunken towards zero. Because of the analytical properties of the L$_2$ norm, the solution can be computed directly through differentiation:

$$\hat{\boldsymbol{\beta}}_{Ridge}(\lambda) = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\, PSS_2(\boldsymbol{\beta}, \lambda) = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1}\boldsymbol{X}^\top \boldsymbol{y} \tag{2}$$

The matrix $\boldsymbol{I}$ denotes the $p \times p$ identity matrix. The addition of $\lambda \boldsymbol{I}$ results in an invertible matrix $\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I}$. It thus results in a better numerical stability, shrinks the regression coefficients, but does in general not lead to coefficients of exactly zero—thus not in a sparse solution.

## 2.2 The Lasso

The Lasso [3] adds an $L_1$ penalty term to the sum of squares criterion:

$$PSS_1(\boldsymbol{\beta}, \lambda) = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1,$$

where $\|.\|_1$ denotes the $L_1$ norm

$$\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{p} |\beta_j|.$$

The resulting regression estimator is defined by

$$\hat{\boldsymbol{\beta}}_{Lasso}(\lambda) = \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\, PSS_1(\boldsymbol{\beta}, \lambda), \tag{3}$$

and this minimization problem can only numerically be solved. A major advantage of the Lasso is the sparseness of the solutions $\hat{\boldsymbol{\beta}}_{Lasso}$: Due to the properties of the $L_1$ norm, several coefficients in the resulting vector $\hat{\boldsymbol{\beta}}_{Lasso}$ are exactly zero (if $\lambda$ is chosen large enough). In literature, this is sometimes referred to as *exact-zero property*. On one hand, this can be regarded as variable selection technique, as the most important variables are returned by the Lasso. On the other hand, this leads to solutions which are much easier to interpret, as the response is a linear combination of only few explanatory variables. Noise variables with minor predictive ability are suppressed, which usually improves the model and the prediction quality.

## 2.3 The Elastic Net

The Elastic Net introduced by [4] can be seen as a compromise between Ridge and Lasso regression. It selects variables like Lasso, and shrinks the coefficients according to Ridge. The objective function of the Elastic Net is defined as

$$PSS_{EN}(\boldsymbol{\beta}, \lambda_1, \lambda_2) = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda_1 \|\boldsymbol{\beta}\|_1 + \lambda_2 \|\boldsymbol{\beta}\|_2^2,$$

and the resulting regression estimator

$$\hat{\boldsymbol{\beta}}_{EN}(\lambda_1, \lambda_2) = (1 + \lambda_2) \operatorname*{argmin}_{\boldsymbol{\beta}} PSS_{EN}(\boldsymbol{\beta}, \lambda_1, \lambda_2) \tag{4}$$

is obtained by numerical optimization. The Elastic Net is more flexible, and for $\lambda_2 = 0$, it gives the Lasso solution.

There is another advantage of the Elastic Net over Lasso: For $n < p$, the Lasso estimator can select at most $n$ variables, i.e. it results in at most $n$ non-zero coefficients. For many applications where $n$ is very low, typically in genomics, one wants to obtain a higher number of non-zero coefficients (select more than $n$ genes). By using this combination of Ridge and Lasso, the Elastic Net solves this problem.

## 2.4 Sparse Principal Component Regression

The idea of using an $\mathrm{L}_1$ penalty as in the objective function of the Lasso can be extended to other statistical methods. One of them is *principal component regression* (PCR). The idea of PCR is to compute principal components (PCs) of the original regressor variables, and to use them for regression. These "derived input directions" reduce the dimensionality in the regression problem, and generate orthogonal regressor variables, hereby avoiding the multicollinearity problem [1].

Principal component analysis (PCA) searches for orthogonal directions $\boldsymbol{a}$, for which the variance of the projected data $\boldsymbol{Xa}$ is a maximum. Denote the sample covariance matrix of $\boldsymbol{X}$ by $\mathbb{V}\mathrm{ar}(\boldsymbol{X}) = \hat{\boldsymbol{\Sigma}}$, then

$$\mathbb{V}\mathrm{ar}(\boldsymbol{Xa}) = \boldsymbol{a}^\top \, \mathbb{V}\mathrm{ar}(\boldsymbol{X}) \, \boldsymbol{a} = \boldsymbol{a}^\top \hat{\boldsymbol{\Sigma}} \boldsymbol{a}.$$

The criterion for the $j$-th PC direction is

$$\hat{\boldsymbol{a}}_{\boldsymbol{j}} = \operatorname*{argmax}_{\boldsymbol{a}, \|\boldsymbol{a}\|_2 = 1} \boldsymbol{a}^\top \hat{\boldsymbol{\Sigma}} \boldsymbol{a} \quad s.t. \ \boldsymbol{a} \perp \hat{\boldsymbol{a}}_{\boldsymbol{i}} \ \ \forall i \in \{1, \ldots, j-1\}. \tag{5}$$

The regression of $\boldsymbol{y}$ is then done on the first $K$ score vectors $\boldsymbol{X}\hat{\boldsymbol{a}}_{\boldsymbol{i}}, \ i \in \{1, \ldots, K\}$. The number $K$ of PCs has to be chosen individually according to a prediction quality criterion, and usually it is much smaller than $p$.

The objective function in (5) can be easily modified to include an $\mathrm{L}_1$ penalty, inducing sparseness of the PCA directions. The resulting criterion is called SCoTLASS criterion [5]:

$$\hat{\boldsymbol{a}}_{\boldsymbol{j}} = \operatorname*{argmax}_{\boldsymbol{a}, \|\boldsymbol{a}\|_2 = 1} \boldsymbol{a}^\top \hat{\boldsymbol{\Sigma}} \boldsymbol{a} - \lambda_j \|\boldsymbol{a}\|_1 \quad s.t. \ \boldsymbol{a} \perp \hat{\boldsymbol{a}}_{\boldsymbol{i}} \ \ \forall i \in \{1, \ldots, j-1\} \tag{6}$$

Note that here we want to maximize the objective function, so we subtract the penalty term. Sparse PCA can thus be seen as a compromise between variance maximization and sparseness of the PCA directions: the more sparseness is desired, the less variance is explained, and vice versa.

With PCR and sparse PCR one obtains score vectors $\boldsymbol{X}\hat{\boldsymbol{a}}_{\boldsymbol{j}}$ for regression that have been derived without using any information of the response $\boldsymbol{y}$. This obvious drawback is considered in Partial Least Squares (PLS) regression and its sparse counterpart, described in the next section.

## 2.5 Sparse Partial Least Squares Regression

In this section we consider the multivariate linear regression model

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{B} + \boldsymbol{E}, \tag{7}$$

where the response variable $\boldsymbol{y}$ from (1) extends to an $n \times q$ matrix $\boldsymbol{Y}$ of responses. We assume that both $\boldsymbol{X}$ and $\boldsymbol{Y}$ are centered. The matrix of regression coefficients $\boldsymbol{B}$ is of dimension $p \times q$, and the error matrix $\boldsymbol{E}$ has the same dimension as $\boldsymbol{Y}$.

Partial least squares (PLS) was developed in [6]. Similar to PCR, it performs a dimensionality reduction to the original regressor variables $\boldsymbol{X}$ by searching for directions $\boldsymbol{w}$. Contrary to PCR, the objective is to maximize the covariance between the scores $\boldsymbol{X}\boldsymbol{w}$ and a linear projection of the responses $\boldsymbol{Y}$. This ensures that the newly derived regressor variables contain relevant information for the prediction of the responses.

There are different models and estimators for the PLS regression problem [see, e.g., 7]. Here we will follow the proposal of [8] in order to continue with their version of sparse PLS.

The idea of PLS regression is a decomposition of the predictor matrix $\boldsymbol{X}$ and the response matrix $\boldsymbol{Y}$:

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^\top + \boldsymbol{E}_{\boldsymbol{X}} \tag{8}$$

$$\boldsymbol{Y} = \boldsymbol{T}\boldsymbol{Q}^\top + \boldsymbol{E}_{\boldsymbol{Y}} \tag{9}$$

where $\boldsymbol{T} = \boldsymbol{X}\boldsymbol{W} \in \mathbb{R}^{n \times K}$ is a score matrix, and $\boldsymbol{W} = (\boldsymbol{w}_{\boldsymbol{1}}, \ldots, \boldsymbol{w}_{\boldsymbol{K}}) \in \mathbb{R}^{p \times K}$ is a matrix of direction (loading) vectors. The equations (8) and (9) can be regarded as ordinary least squares problems, so $\boldsymbol{P} \in \mathbb{R}^{p \times K}$ and $\boldsymbol{Q} \in \mathbb{R}^{q \times K}$ are matrices of coefficients, whereas $\boldsymbol{E_X} \in \mathbb{R}^{n\times p}$ and $\boldsymbol{E_Y} \in \mathbb{R}^{n\times q}$ are matrices of random errors. Again, $K$ denotes the number of components, with $K \leq \min\{n, p, q\}$.

If we rewrite equation (9),

$$\boldsymbol{Y} = \boldsymbol{T}\boldsymbol{Q}^\top + \boldsymbol{E_Y} = \boldsymbol{X}\boldsymbol{W}\boldsymbol{Q}^\top + \boldsymbol{E_Y}, \tag{10}$$

we see that $\boldsymbol{W}\boldsymbol{Q}^\top \in \mathbb{R}^{p\times q}$ is a matrix of coefficients that relates $\boldsymbol{Y}$ to the original data $\boldsymbol{X}$ according to the original model (7).

In order to successively find direction vectors $\boldsymbol{w}$ that maximize the covariance between the explanatory variables and the responses, the SIMPLS criterion of [9] is used (we consequently replace the unknown population parameters by their corresponding sample estimates):

$$\boldsymbol{\hat{w}_j} = \underset{\boldsymbol{w}, \|\boldsymbol{w}\|_2=1}{\operatorname{argmax}} \ \boldsymbol{w}^\top \boldsymbol{M}\boldsymbol{w} \quad s.t.\ \boldsymbol{w}^\top \boldsymbol{\hat{\Sigma}}\boldsymbol{\hat{w}_i} = 0 \ \ \forall i \in \{1, \ldots, j-1\} \tag{11}$$

where $\boldsymbol{\hat{\Sigma}}$ is the sample covariance of $\boldsymbol{X}$, and $\boldsymbol{M} = \boldsymbol{X}^\top\boldsymbol{Y}\boldsymbol{Y}^\top\boldsymbol{X}$. Once the $K$ directions ($\boldsymbol{\hat{w}_1}$, ..., $\boldsymbol{\hat{w}_K}$) are found, $\boldsymbol{Q}$ can be estimated through OLS, and we finally get

$$\boldsymbol{\hat{\beta}}_{PLS} = \boldsymbol{\hat{W}}\boldsymbol{\hat{Q}}^\top. \tag{12}$$

If a penalty term is added to the criterion in the usual way, the resulting direction estimates tend to be not sparse enough when one requires a high percentage of explained variance [5]. [8] modified the objective function (11) in order to obtain an even stronger criterion for the sparseness of the direction vectors $\boldsymbol{w}$. The modified criterion is inspired by [10] and by the ideas of the Elastic Net [4]. The sparsity is imposed on a surrogate of vector $\boldsymbol{c}$ instead of the original vector $\boldsymbol{w}$:

$$\boldsymbol{\hat{w}} = \underset{\boldsymbol{c}, \boldsymbol{w}, \|\boldsymbol{w}\|_2=1}{\operatorname{argmin}} \ -\kappa\boldsymbol{w}^\top\boldsymbol{M}\boldsymbol{w} + (1-\kappa)(\boldsymbol{c}-\boldsymbol{w})^\top\boldsymbol{M}(\boldsymbol{c}-\boldsymbol{w}) + \lambda_1\|\boldsymbol{c}\|_1 + \lambda_2\|\boldsymbol{c}\|_2^2 \tag{13}$$

The formula consists of different parts: Similar to the original SIMPLS criterion, the first term $-\boldsymbol{w}^\top\boldsymbol{M}\boldsymbol{w}$ is responsible for a high covariance between the response and the regressor variables. The second term $(\boldsymbol{c}-\boldsymbol{w})^\top\boldsymbol{M}(\boldsymbol{c}-\boldsymbol{w})$ assures that $\boldsymbol{c}$ and $\boldsymbol{w}$ are kept close to each other. A compromise between these two terms is controlled by the parameter $\kappa$. Finally, the $\mathrm{L}_1$ penalty term imposes sparsity on $\boldsymbol{c}$, while the $\mathrm{L}_2$ penalty takes care of the shrinkage of the parameters. The problem of solving equation (13) for $\boldsymbol{c}$ and fixed $\boldsymbol{w}$ is equivalent to the Elastic Net. The criterion coincides with SCoTLASS, if $\boldsymbol{w} = \boldsymbol{c}$ and $\boldsymbol{M} = \boldsymbol{\hat{\Sigma}}$.

[8] pointed out that in practice, the problem of finding the four parameters $K$, $\kappa$, $\lambda_1$, and $\lambda_2$ can be reduced to a two parameter search. This is due to the fact that $\lambda_2$ is commonly chosen very large and it can thus be set to infinity, and for univariate $\boldsymbol{Y}$ the solution does not depend on $\kappa$. The remaining Lasso penalty problem can then be converted to a soft thresholding of the original PLS direction vectors,

$$\tilde{w}_i = \max(0, |\hat{w}_i| - \eta \max_{1 \leq j \leq p} |\hat{w}_j|) \cdot \operatorname{sign}(\hat{w}_i) \tag{14}$$

where $\hat{\boldsymbol{w}} = (\hat{w}_1, \ldots, \hat{w}_p)^\top$ are the estimated PLS direction vectors, and $\eta$ plays the role of $\lambda_1$, with $0 \leq \eta \leq 1$. This form of thresholding sets those components of the direction vectors to zero, which are smaller than a given proportion of the largest component. This formulation of sparse partial least squares will be used in the following.

## 2.6 Sparse Partial Least Squares Discriminant Analysis

PLS regression has become popular in chemometrics also for classification tasks. The resulting method is then called PLS discriminant analysis (PLSDA) [see, e.g., 11]. This method is especially useful for high-dimensional data, where classical discrimination methods like linear discriminant analysis (LDA) have numerical difficulties because of singularity issues. This is the reason why PLSDA is used not only in chemometrics, but also frequently in bioinformatics [e.g. 12, 13, 14, 15, 16].

For PLSDA, the same model (7) as in the previous section is used, and also the decomposition of $\boldsymbol{X}$ and $\boldsymbol{Y}$ into scores and loadings as in (8) and (9) is done. However, the difference here is that $\boldsymbol{Y}$ contains the group information: Suppose that each observation belongs to one of $G$ groups. Then the matrix $\boldsymbol{Y}$ is constructed as an $n \times G$ matrix, with elements

$$y_{ij} = \begin{cases} 1 & \text{if the } i\text{-th observation belongs to the } j\text{-th group} \\ 0 & \text{otherwise,} \end{cases}$$

where $i = 1, \ldots, n$ and $j = 1, \ldots, G$. In case of $G = 2$ it is sufficient to work with only the first column of $\boldsymbol{Y}$. For alternative coding schemes see [13] and [14].

There are basically two possibilities for the group assignment of a new test set observation: Since after (sparse) PLS regression one obtains for the test set observation a group prediction for each of the $G$ groups, one can assign the observation to that group with the largest predicted value. This, however, is depending on the way how $\boldsymbol{X}$ is preprocessed [see, e.g., 11]. Another option is to estimate the scores $\hat{\boldsymbol{T}} = \boldsymbol{X}\hat{\boldsymbol{W}}$ using (sparse) PLS, and afterwards to employ LDA in this score space. Since the dimensionality of $\hat{\boldsymbol{T}}$ is usually much less than the original dimension $p$ of $\boldsymbol{X}$, LDA can be applied efficiently, and the different group sizes can even be incorporated with prior probabilities [see, e.g., 7]. We will consider this last approach in the numerical examples below. This approach has also been suggested in the context of sparse PLS discriminant analysis by [17], and it has been successfully applied in bioinformatics [e.g. 18, 19].

# 3 Examples

In may be difficult to provide an overall picture on how and when sparseness improves regression and classification models, in particular in applications to chemometrics. Many papers in the field of bioinformatics have demonstrated the usefulness of sparse estimation, but usually the dimensionality of the data sets is in the thousands, and even much higher, and the sample size is very low. Nevertheless, in this section we include simulated data and real data examples from chemometrics to compare sparse and non-sparse methods. In the following, sparse PLS is abbreviated by SPLS, and sparse PLSDA by SPLSDA.

## 3.1 PLSDA Versus SPLSDA for Simulated Data

We demonstrate the discriminative abilities of SPLSDA in a simulated data example, following the data generation suggested by [20]. A training set with dimension $200 \times p$ is generated, containing $G = 2$ groups of 100 observations each. Both groups are distributed according to a multivariate normal distribution, $\boldsymbol{X_i} \sim N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, with

$$\boldsymbol{\mu}_1 = (0, 2.9, 0, \dots, 0)^\top \qquad \boldsymbol{\mu}_2 = (0, -2.9, 0, \dots, 0)^\top$$

and

$$\boldsymbol{\Sigma}_i = \begin{pmatrix} 1 & 0.7 & 0 \dots 0 \\ 0.7 & 1 & 0 \dots 0 \\ 0 & 0 & 1 \dots 0 \\ \vdots & \vdots & \vdots \ddots \vdots \\ 0 & 0 & 0 \dots 1 \end{pmatrix}$$

for $i \in \{1, 2\}$. This means that the first and the second variable store the whole information about the group membership, whereas the remaining variables are uncorrelated noise.

Using this training set, PLSDA and SPLSDA are applied, in order to establish classification rules. For both methods, the optimal number of $K = 2$ components is chosen. Afterwards a test set of 100 observations (50 per group) is created and the misclassification rates of both methods are computed. The parameter $p$, controlling the dimension of the data, is varied from 100 to 20000. For each value of $p$, the procedure is repeated 10 times in order to obtain mean values of the misclassification rates, as well as corresponding standard errors.

The situation clearly favors the sparse version, as there are many noisy variables in the data. It is most likely that classical PLSDA includes many of them in the projected scores. In contrast, the sparse property of SPLSDA forces noise variables to be neglected. The larger the penalty parameter is chosen, the more variables are excluded. Therefore we can expect higher misclassification rates of the PLSDA method, especially for larger values of $p$.

Figure 1 demonstrates the superior abilities of the SPLSDA method. The thick lines denote the mean misclassification rates in the simulation (averaged over 10 simulations), whereas the thin lines are mean misclassification rate ± standard error. As the dimension $p$ increases, the more and more erroneous the classical PLSDA method gets (solid lines). In the case of $G = 2$, SPLSDA does not depend on the tuning parameter $\kappa$ but only on $\eta$, see Section 2.5. Here we did not optimize with respect to $\eta$, but present two solutions, for $\eta = 0.5$ (dashed lines) and for $\eta = 0.8$ (dotted lines). Especially for the latter choice, the performance of SPLSDA is impressive.

Figure 1 about here

In order to better understand the results, we further investigate the scores $\hat{\boldsymbol{T}} = \boldsymbol{X}\hat{\boldsymbol{W}}$. Since the number of score vectors is $K = 2$, they can easily be plotted. Figure 2 compares the scores of PLSDA (left) and SPLSDA with $\eta = 0.8$ (right), for $p = 100$ (top) and $p = 15000$ (bottom). The difference in the plots explains the trend of the misclassification rates: As the dimension increases, the non-sparse partial least squares method tends to overfit the data. This is clearly shown in the lower left plot, where the observations of the training set (dots) are perfectly separated, whereas possible test set observations (triangles) are overlapping. In contrast, the sparse version remains stable in the separation of both training and test set.

Figure 2 about here

In order to get a better idea of the described difference between PLSDA and SPLSDA, we investigate the loadings of the first component $\hat{\boldsymbol{w}}_{\mathbf{1}}$, which is a vector of length $p$. It connects the first component of the scores $\hat{\boldsymbol{T}}$ to the original data $\boldsymbol{X}$. Thus, a large absolute loading value means that the corresponding (original) variable is highly correlated with the response. Figure 3 depicts the number of loadings in $\hat{\boldsymbol{w}}_{\mathbf{1}}$ which are not equal to zero, depending on the dimension $p$. It can be seen that for PLSDA, all loadings are not equal to zero. This means that all the variables of $\boldsymbol{X}$, even the unnecessary noise, are included in the LDA, which results in overfitting. This becomes more and more problematic for discrimination as the dimension grows. SPLSDA includes only the most important variables and thus remains stable.

Figure 3 about here

## 3.2 PLS Versus SPLS for NIR Data

[21] provided a data set where 166 alcoholic fermentation mashes of different feedstock (rye, wheat and corn) were analyzed. The response variables are the concentrations of glucose and ethanol (in g/L) in substrates from the bioethanol processes. The 235 predictor variables contain the first derivatives of near infrared spectroscopy (NIR) absorbance values at 1115-2285 nm, measured in liquid samples. The data set is available in the R package **chemometrics** as data set `NIR` [22].

In the following we build separate models for glucose and ethanol, thus the response is univariate. We compute PLS and SPLS models using a range from 1 to 20 components. The optimal model is then selected according to repeated double cross validation (rdCV) [23] with 10 repetitions, 4 outer and 10 inner segments. Note that for SPLS also the optimal value of $\eta$ is selected with this validation scheme; we selected the optimal $\eta$ among values in the interval 0.1 to 0.9 in steps of 0.1. As a measure of performance we use the mean squared error of prediction

$$\mathrm{MSEP}_l = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i^l)^2,$$

where $\boldsymbol{y}_i$ are the univariate responses, and $\hat{\boldsymbol{y}}_i^l$ are the corresponding predictions within the rdCV scheme, using $l \in \{1, \ldots, 20\}$ components. For the determination of the optimal number of (S)PLS components we use the *two standard error rule* [1, 7], which takes the most parsimonious model for which the MSEP is still in the range of two standard errors of the overall minimal MSEP. The final optimal number of components after the rdCV procedure is denoted by $a_{opt}^P$ for PLS and $a_{opt}^S(\eta)$ for SPLS with the final choice of $\eta$.

The results are depicted in Figure 4, which is split up into four parts: The upper and lower part of the image correspond to the glucose and the ethanol response, respectively. The left column shows the MSEP values for both PLS and SPLS with the final best values of $\eta$. The vertical lines indicate the optimal number of components for PLS (solid) and SPLS (dashed). The right images plot the experimental $y_i$ values against the fitted values $\hat{y}_i$ $(i = 1, \ldots, n)$ for the final SPLS models. Since rdCV has been repeated ten times, ten predictions are available for each observation. These predictions are shown as gray crosses, whereas the corresponding means are painted in black.

For the response *glucose* (top row) the optimal number of components is the same for PLS and SPLS, but there is a remarkable difference in the resulting MSEP, as it is visible from the figure. For *ethanol* (bottom row), SPLS requires fewer components than PLS, which can be an advantage in terms of interpretation. However, the resulting MSEP is almost the same.

Figure 4 about here

## 3.3 PLS Versus SPLS for QSPR Data

This example contains data describing chemical-physical properties of chemical compounds, and thus belongs to the area of quantitative structure-property relationships (QSPR). The compounds are modeled by chemical structure data which have been drawn manually by the structure editor software Corina [24] and Dragon [25]. Here we consider 209 polycyclic aromatic compounds (PAC) which is the characterized by

467 molecular descriptors. The response variable is the gas chromatographic (GC) retention index for several substance classes. Since the descriptors cover a great diversity of chemical structures, still many of them may be irrelevant for predicting the response. The data set is available in the R package **chemometrics** as data set PAC.

Again, we used rdCV and the same approach as described in the NIR example above. The results are depicted in Figure 5 and are even more convincing. The superior predictive ability of SPLS is clearly shown in the left plot. Note that especially in the low component number sector, SPLS outperforms PLS by far.

Figure 5 about here

Several other regression techniques have been used for this data example in [7] (Section 4.9.1). However, the SPLS method presented here outperforms all of them: with the optimal number of 4 components the MSEP is 57.3.

## 3.4 PLSDA Versus SPLSDA for Spectral Data

[26] used a data set containing the spectra of three different cultivars of the same fruit. The three cultivars (groups) are named D, M and HA, and their sample sizes are 490, 106 and 500 observations, respectively. The spectra are measured at 256 wavelengths. The fruit data is thus a high-dimensional data set which was used by [26] to illustrate a new approach for robust linear discriminant analysis, and it was studied again by [27]. From these studies it is known that the first two cultivars D and M are relatively homogenous and do not contain atypical observations, but the third group HA contains a subgroup of 180 observations which were obtained with a different illumination system. Outlier detection and robust methods for discrimination are out of the scope of this work, and therefore we remove these 180 observations using the same procedure as [27], and we remain with 320 observations in the third group. This outlier removal is even necessary for applying (S)PLSDA for the three groups, since the outliers would have a strong influence on the classification rules. Figure 6 shows plots of the first and second PLSDA component when using the complete group HA (left) and the cleaned group HA (right). According to the plots, the group separation has improved when applying the non-sparse method to the cleaned data, and thus one can also expect an improvement of the discriminative

power for the sparse method.

Figure 6 about here

Next we apply PLSDA and SPLSDA to the reduced data set. We also compare with a PCA dimension reduction, followed by LDA. This procedure will be denoted as PCADA. For all these methods it is crucial to determine the optimal number of components. As an optimality criterion we use the misclassification rate, which is the proportion of misclassified test set observations. Since the complete data set contains a reasonable number of samples, the data could be randomly split into training (60%) and test (40%) data sets for the evaluation, as suggested by [26]. We prefer to use the rdCV approach of [7], which is applied with 10 repetitions, 4 outer and 10 inner segments. In the inner loop for SPLS not only the number of components is selected but also the optimal value of $\eta$ – we selected the optimal $\eta$ among values in the interval 0.1 to 0.9 in steps of 0.1 (minimizing the misclassification rate). Table 1 shows the resulting misclassification rates (in %) for PCADA, PLSDA, and SPLSDA. The results for PCADA are clearly worse—a consequence of not using the information of the response variable for dimension reduction. Both PLSDA and SPLSDA achieve already with 5 PLS components good discrimination performance below 3% and the misclassification rate falls below 1% when 10 or more components are selected. We report only the results for up to 10 components, since 10 was the optimal number for PLSDA and SPLSDA. It is obvious from the results that the *fruit* data set is not "sparse" (not many variables are highly correlated with each other, respectively there are no variables containing only noise) and SPLSDA cannot identify a parsimonious model. Except for 1 PLS component, always about 250 variables are selected (out of 256) and the results do not change much when we investigate the discrimination performance for 10 or even more PLS components.

Table 1 about here

In general, it is preferable to use SPLSDA instead of PLSDA because with not much more computational effort we could automatically take advantage of data sets in which not all variables are important for the discrimination of the groups.

## 3.5 PLSDA Versus SPLSDA: Data of Low Dimensionality

A popular data set often used for evaluating and comparing classification and discrimination algorithms is the wine data set, which contains results of chemical analyses of Italian wines from the same region but derived from three different cultivars. The data set is described in [28] and is available from the UCI repository of Machine Learning Databases [29]. It was used among others by [26] to demonstrate their robust SIMCA algorithm, see also [20]. The chemical analysis resulted in the quantities of 13 constituents in each of the three types of wines represented by 178 different samples. In this example we want to investigate the discrimination performance of SPLSDA with different numbers of selected variables. We split randomly the data set into a training data set containing 70% of the data and a test data set with the remaining observations. Using the training data set we build a model with 2 SPLS components, where each of the components is restricted to $l = 1, \ldots, 13$ non-zero coefficients. This means that here we do not directly use a tuning parameter for computing the SPLS models, but the number of variables that may contribute to each component is restricted. For each model we estimate the misclassification rate using the test data set. This exercise is repeated 100 times and the average misclassification rate for each selected number of variables is calculated. These average error rates are plotted against the number of selected variables in Figure 7.

Figure 7 about here

It can be seen that with the restriction of six variables (per component) the discriminative power of the sparse PLS model is already reasonably high (with a misclassification rate of 3.4%). In comparison, a PLS model with two components based on all available variables results in a misclassification rate on 2.8%, and using SPLS with 12 non-zero coefficients per component leads to a value of 2.5%.

During the computation of the sparse model we also recorded the frequencies with which the different variables were selected. The frequencies for the model with six selected variables per component are shown in Table 2, and this confirms that the importance of the variables for the SPLSDA model is stable and does not much depend on the random partitioning. The variables 6, 7, 11, 12 and 13 have (almost) always been selected for the first component, variables 1, 3, 10, 11 and 13 (almost) always for the second component. This not only suggests which variables are most important for the discrimination, but it would also allow the expert to interpret the resulting components.

Table 2 about here

# 4 Conclusions

In high-dimensional regression and classification problems, usually only few of the available variables are really informative for explaining the response or the grouping variable(s). Including also the uninformative "noise" variables in a model–even with a very small contribution–reduces the performance of the method. This is particularly the case in high-dimensional problems, where small noise contributions can sum up to very significant (non-)information. A solution to this problem is to use a Lasso ($L_1$) penalty [3] in the objective function of the corresponding method. Depending on the size of this penalty, some of the estimated coefficients are forced to take on values of zero, which reduces the influence of noise variables completely. The resulting method can then be seen as a variable selection tool for regression or classification.

Different forms of the penalty have been proposed in the literature. Since an $L_1$ penalty may result in too few non-zero coefficients, a combination of $L_1$ and $L_2$ penalty has been proposed [4], which is also used in the context of sparse PLS [8, 17].

A comparison between sparse methods and non-sparse counterparts shows that especially in high-dimension, sparseness can lead to an improvement of the prediction or classification performance. There is, however, no guarantee that sparse methods automatically give better results. It depends on the data structure if a sparse solution can be found, i.e. if the data contain noise variables that can be omitted by the sparse method.

Also the sample size plays an important role in the context of statistical estimation in high-dimensional problems. The smaller the sample size, the more likely it is to identify pure random noise variables as "good" predictors in a regression or classification problem, [see, e.g., 30, Figure 1]. This undesired artifact shows that the estimation problem is no longer managable, and a way out is to impose sparsity. One can thus conclude that sparse data (in the sense of only few observations in high dimension) even call for sparse estimation methods.

The only disadvantage of sparse methods is that additional tuning parameters need to be selected within a validation scheme, and this leads to a higher computational effort. An efficient implementation of the sparse method is thus a requirement in order to be practically useful, like the computationally efficient algorithm for SPLS proposed by [8] which is implemented in their R package **spls**. Especially in case of univariate response the algorithm is very fast: an average run including the tuning needs less then a minute for a sample size of $n = 100$ with $p = 5000$ predictors on a 64 bit Windows 7 machine with Intel Core i7 2600 (4x3.4 Ghz) CPU. An average run for the SPLSDA example in Section 3.4 with $n = 595$, $p = 56$ and three groups including the tuning by cross validation took less than three minutes on the same machine. The overall computation time, however, depends on the scheme for determining the optimal number of PLS components. In our examples we used repeated double cross validation [23] for this purpose, which is computationally much more demanding that an evaluation based on standard cross validation or information criteria.

There are several packages in R [22], implementing SPLS and SPLSDA. In most of the cases these functions are wrappers around the corresponding `pls` and `spls` functions. The most popular packages are **spls** [see 31], **mixomics** [see 32] and **caret** [see 33].

Note that the proposed procedure for SPLSDA is a two-step procedure (SPLS followed by LDA) where the objective function is devoted to obtain sparse regression coefficients, but not sparseness directly for the classification purpose. There exist other sparse techniques for discrimination which apply $L_1$ penalization or similar methods directly to the discriminant analysis framework: [34] implemented their method in the R package **sparseLDA** [see 35] and [36] implemented their proposal in the R package **penalizedLDA** [see 37].

# References

[1] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Verlag, New York, 2nd edition, 2009.

[2] A.E. Hoerl and R.W. Kennard. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

[3] R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 58:267–288, 1996.

[4] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society B*, 67:301–320, 2005.

[5] I. T. Jolliffe, N. T. Trendafilov, and M. Uddin. A modified principal component technique based on the Lasso. *Journal of Computational and Graphical Statistics*, 12:531–547, 2003.

[6] H. Wold. Soft modeling by latent variables: the non-linear iterative partial least squares approach. In J. Giani, editor, *Perspectives in probability and statistics, papers in honor of M.S. Bartlett*, pages 117–142, London, 1975. Academic Press.

[7] Kurt Varmuza and Peter Filzmoser. *Introduction to Multivariate Statistical Analysis in Chemometrics*. Taylor and Francis - CRC Press, Boca Raton, FL, 2009.

[8] Hyonho Chun and Sndz Keles. Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(1):3–25, 2010.

[9] S. de Jong. SIMPLS: An alternative approach to partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 18:251–263, 1993.

[10] Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15:265–286, 2006.

[11] R.G. Brereton. *Applied Chemometrics for Scientists*. Wiley, Chichester, United Kingdom, 2007.

[12] D. Nguyen and D. Rocke. Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics*, 18:39–50, 2002.

[13] D. Nguyen and D. Rocke. Muti-class cancer classification via partial least squares with gene expression profiles. *Bioinformatics*, 18:1216–1226, 2002.

[14] Anne-Laure Boulesteix. PLS dimension reduction for classification with microarray data. *Statistical Applications in Genetics and Molecular Biology*, 3(1):Article 33, 2004.

[15] Jian J. Dai, Linh Lieu, and David Rocke. Dimension reduction for classification with gene expression microarray data. *Statistical Applications in Genetics and Molecular Biology*, 5(1):Article 6, 2006.

[16] Anne-Laure Boulesteix, Christine Porzelius, and Martin Daumer. Microarray-based classification and clinical predictors: on combined classifiers and additional predictive value. *Bioinformatics*, 24(15):1698–1706, 2008.

[17] Dongjun Chung and Sunduz Keles. Sparse partial least squares classification for high dimensional data. *Statistical Applications in Genetics and Molecular Biology*, 9(1):Article 17, 2010.

[18] Kim-Anh Lê Cao, Debra Rossouw, Christle Robert-Grani, and Philippe Besse. A sparse pls for variable selection when integrating omics data. *Statistical Applications in Genetics and Molecular Biology*, 7(1):Article 35, 2008.

[19] Kim-Anh Lê Cao, Simon Boitard, and Philippe Besse. Sparse pls discriminant analysis: biologically relevant feature selection and graphical displays for multiclass problems. *BMC Bioinformatics*, 12(1):253, 2011.

[20] Zhihua Qiao, Lan Zhou, and Jianhua Z. Huang. Sparse linear discriminant analysis with applications to high dimensional low sample size data. *IAENG International Journal of Applied Mathematics*, 39(1):48–60, 2009.

[21] B. Liebmann, A. Friedl, and K. Varmuza. Determination of glucose and ethanol in bioethanol production by near infrared spectroscopy and chemometrics. *Analytica Chimica Acta*, 642:171–178, 2009.

[22] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.

[23] P. Filzmoser, B. Liebmann, and K. Varmuza. Repeated double cross validation. *Journal of Chemometrics*, 23:160–171, 2009.

[24] J. Sadowski, C.H. Schwab, and J. Gasteiger. *Software for the generation of high-quality three-dimensional molecular models.* Molecular Networks GmbH Computerchemie, Erlangen, Germany, 2004.

[25] R. Todeschini, V. Consonni, A. Mauri, and M. Pavan. *Software for the calculation of molecular descriptors*. Pavan M. Talete slr, Milan, Italy, 2004.

[26] M. Hubert and K. Van Driessen. Fast and robust discriminant analysis. *Computational Statistics & Data Analysis*, 45:301–320, 2004.

[27] Karlien Vanden Branden and Mia Hubert. Robust classification in high dimensions based on the SIMCA method. *Chemometrics and Intellegent Laboratory Systems*, 79:10–21, 2005.

[28] M. Forina, C. Armanino, M. Castino, and M. Ubigli. Multivariate data analysis as a discriminating method of the origin of wines. *Vitis*, 25:189–201, 1986.

[29] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[30] J. Fan and J. Lv. A selective overview of variable selection in high dimensional feature space. *Statistica Sinica*, 20:101–148, 2010.

[31] Dongjun Chung, Hyonho Chun, and Sunduz Keles. *spls: Sparse Partial Least Squares (SPLS) Regression and Classification*, 2009. R package version 2.1-0.

[32] Sebastien Dejean, Ignacio Gonzalez, Kim-Anh Lê Cao, and Pierre Monget. *mixOmics: Omics Data Integration Project*, 2011. R package version 2.9-4.

[33] Max Kuhn, Jed Wing, Steve Weston, Andre Williams, Chris Keefer, and Allan Engelhardt. *caret: Classification and Regression Training*, 2011. R package version 5.05.004.

[34] L. Clemmensen, T. Hastie, D. Witten, and B. Ersboll. Sparse discriminant analysis. *Technometrics*, 2011. To appear.

[35] Line Clemmensen and Max Kuhn. *sparseLDA: Sparse Discriminant Analysis*, 2009. R package version 0.1-5.

[36] Daniela M. Witten and Robert Tibshirani. Penalized classification using Fisher's linear discriminant analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, pages 753–772, 2011.

[37] Daniela Witten. *penalizedLDA: Penalized classification using Fisher's linear discriminant*, 2011. R package version 1.0.

Table 1: PCADA, PLSDA and SPLSDA for the fruit data set: misclassification rate in % for the first 10 components. The last two rows show the parameter $\eta$ for SPLSDA found by rdCV and the number of variables selected by SPLSDA for the corresponding number of PLS components. The last column shows the optimal number of components for the corresponding model. The total number of variables in the data set is 256.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Opt |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PCADA | 42.12 | 15.97 | 13.63 | 9.56 | 8.92 | 6.84 | 6.06 | **3.94** | 3.85 | 3.89 | 8 |
| PLSDA | 38.43 | 12.61 | 9.52 | 4.23 | 2.86 | 2.77 | 1.95 | 1.77 | 1.82 | **0.95** | 10 |
| SPLSDA | 36.30 | 12.37 | 9.71 | 4.23 | 2.77 | 2.57 | 2.19 | 1.62 | 1.53 | **0.93** | 10 |
| ($\eta$) | 0.9 | 0.2 | 0.2 | 0.2 | 0.4 | 0.6 | 0.6 | 0.7 | 0.7 | 0.7 | 0.7 |
| (VARS) | 26 | 252 | 255 | 256 | 254 | 248 | 249 | 247 | 247 | 251 | 250 |

Table 2: Frequencies of selection of the variables by SPLSDA for the wine data obtained by 100 random partitions of the data for two components, and 6 non-zero coefficients per component. The rows correspond to the first and second component, the columns to the 13 variables. The row sums are thus 600.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 13 | 0 | 45 | 0 | 100 | 100 | 21 | 28 | 1 | 99 | 100 | 93 |
| 2 | 100 | 35 | 99 | 11 | 64 | 0 | 0 | 0 | 0 | 100 | 91 | 0 | 100 |

## Legends of Figures

### Figure 1.

Misclassification rates (MCR) of the simulated data with growing dimension $p$. The thicker lines are the mean misclassification rates (averaged over 10 simulations). The thinner lines denote mean ± standard error.

### Figure 2.

Simulated data example: First versus second component of the scores $\hat{\boldsymbol{T}} = \boldsymbol{X}\hat{\boldsymbol{W}}$, derived from PLSDA (left) and SPLSDA (right), for $p = 100$ (top) and $p = 15000$ (bottom), respectively. The dots denote the training set, whereas the triangles are observations from a possible, independent test set. The colors correspond to the groups.

### Figure 3.

Simulated data example: Number of loadings in $\hat{\boldsymbol{w}}_{\boldsymbol{1}}$ not equal to zero, depending on the dimension $p$. For PLSDA, the resulting line indicates that all loadings are not equal to zero, whereas for SPLSDA only few loadings are.

### Figure 4.

NIR data. Top: Response glucose, Bottom: Response ethanol. Left: Mean squared error of prediction, measured through repeated double cross validation for different numbers of components. The vertical lines denote $a_{opt}$: The smallest number of components, for which the MSEP is still in range of two standard errors of the overall minimum MSEP. Right: Experimental values $y_i$ versus the fitted values $\hat{y}_i$ for the final SPLS model. The set of fitted values obtained through the repeated double cross validation is painted gray, their averages are painted black.

### Figure 5.

PAC data. Left: Mean squared error of prediction, measured through repeated double cross validation for different numbers of components. The vertical lines denote $a_{opt}$: The smallest number of components, for which the MSEP is still in range of two standard deviations of the overall minimum MSEP. Right: Experimental values $y_i$ versus the fitted values $\hat{y}_i$ for the final SPLS model. The set of fitted values obtained through the repeated double cross validation is painted gray, their averages are painted black.

**Figure 6.**

Visualization of the first two (non-sparse) PLSDA scores when using the complete group HA including the outliers (left) and the cleaned group HA (right).

**Figure 7.**

SPLSDA of Wine data. Average misclassification rate (from 100 random partitions of the data into training and test data sets) against the selected number of variables per SPLS component.

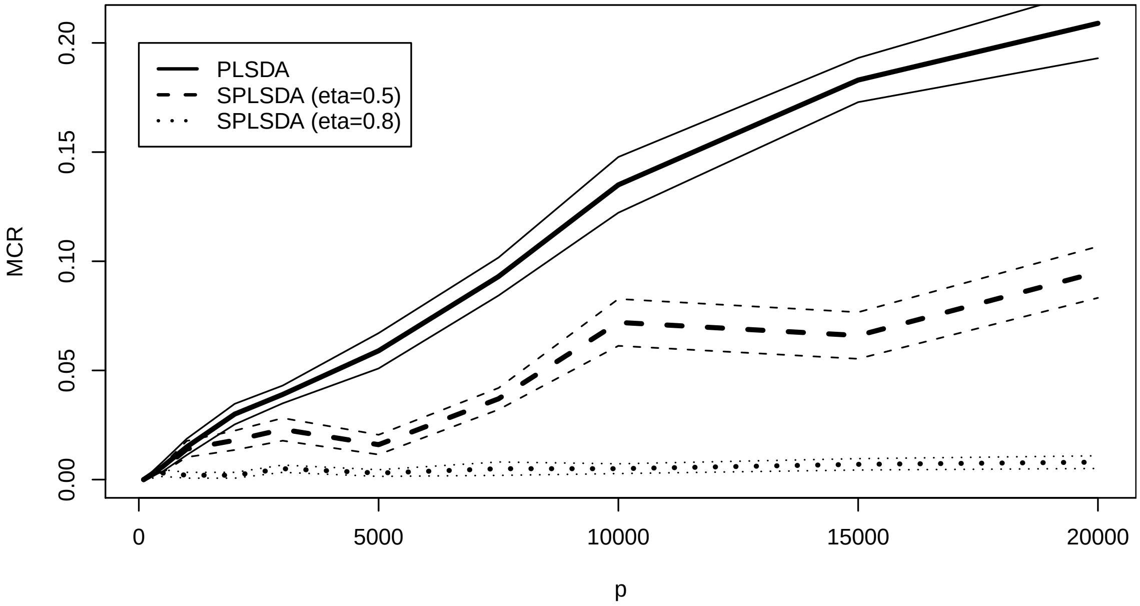

Figure 1: Misclassification rates (MCR) of the simulated data with growing dimension $p$. The thicker lines are the mean misclassification rates (averaged over 10 simulations). The thinner lines denote mean ± standard error.

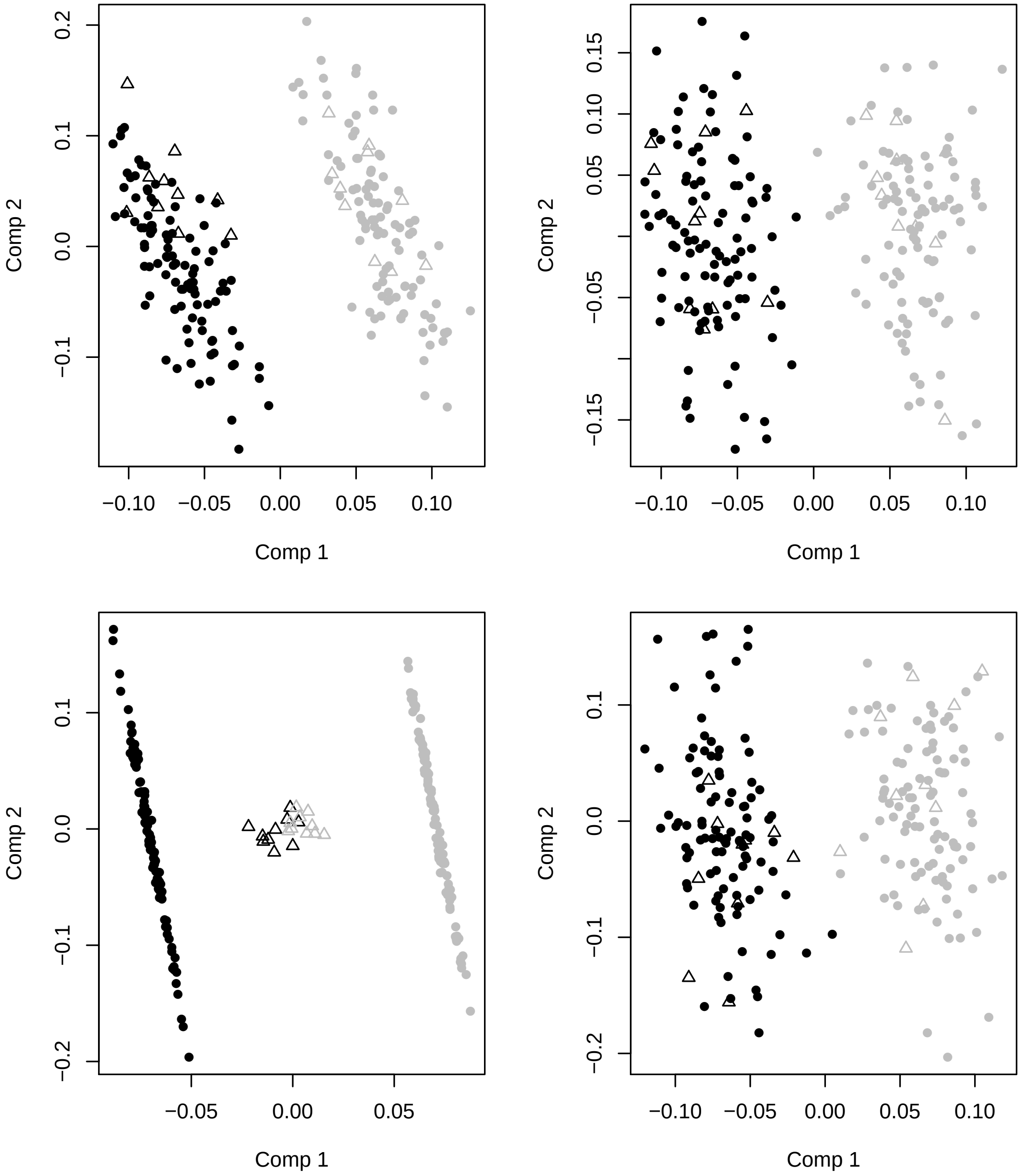

Figure 2: Simulated data example: First versus second component of the scores $\hat{\boldsymbol{T}} = \boldsymbol{X}\hat{\boldsymbol{W}}$, derived from PLSDA (left) and SPLSDA (right), for $p = 100$ (top) and $p = 15000$ (bottom), respectively. The dots denote the training set, whereas the triangles are observations from a possible, independent test set. The colors correspond to the groups.

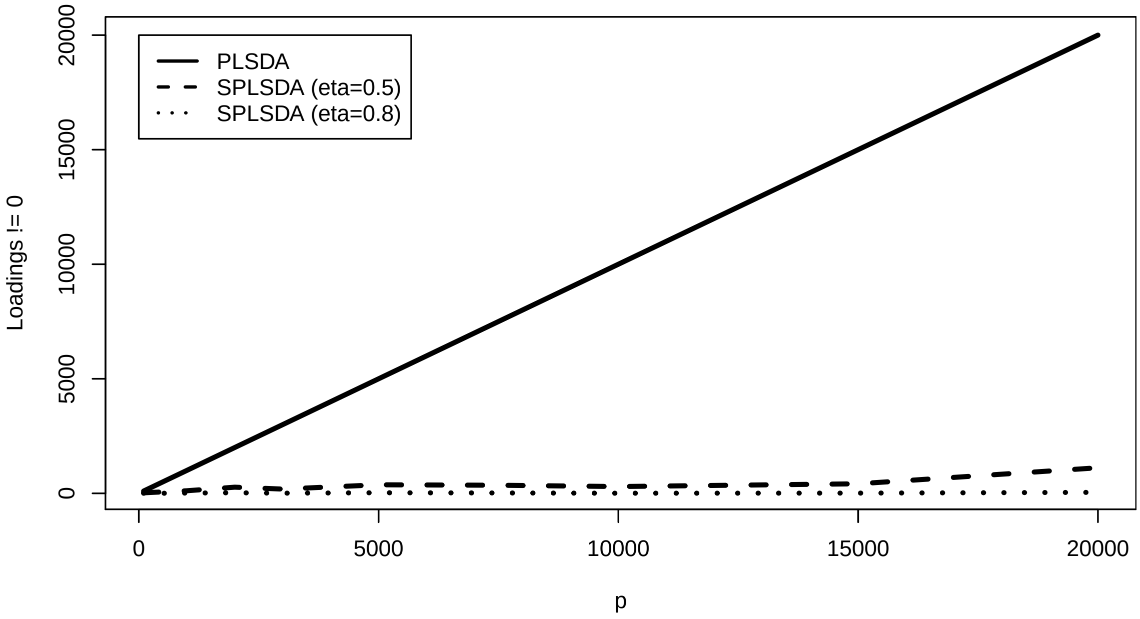

Figure 3: Simulated data example: Number of loadings in $\hat{\boldsymbol{w}}_{\mathbf{1}}$ not equal to zero, depending on the dimension $p$. For PLSDA, the resulting line indicates that all loadings are not equal to zero, whereas for SPLSDA only few loadings are.

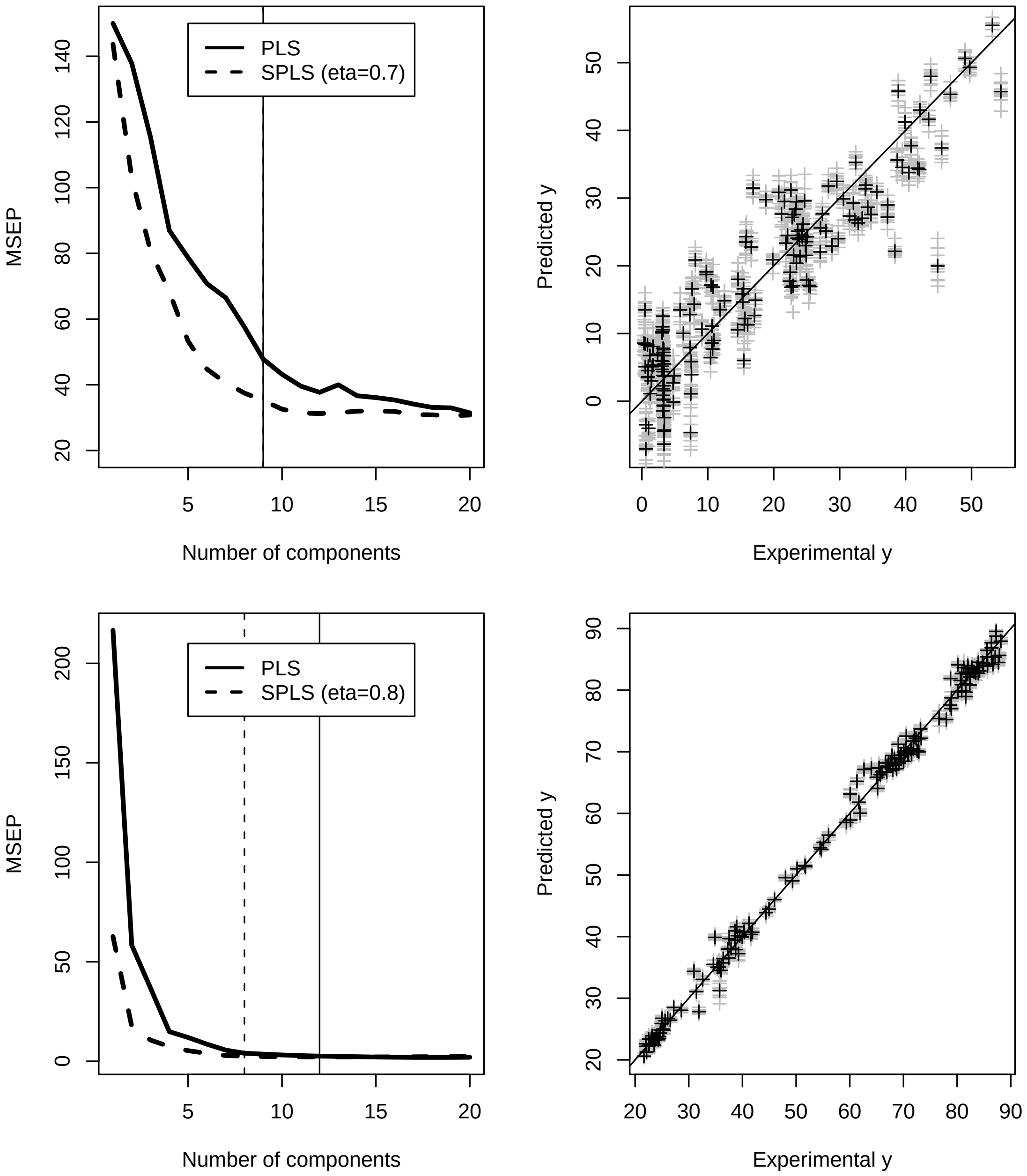

Figure 4: NIR data. Top: Response glucose, Bottom: Response ethanol. Left: Mean squared error of prediction, measured through repeated double cross validation for different numbers of components. The vertical lines denote $a_{opt}$: The smallest number of components, for which the MSEP is still in range of two standard errors of the overall minimum MSEP. Right: Experimental values $y_i$ versus the fitted values $\hat{y}_i$ for the final SPLS model. The set of fitted values obtained through the repeated double cross validation is painted gray, their averages are painted black.

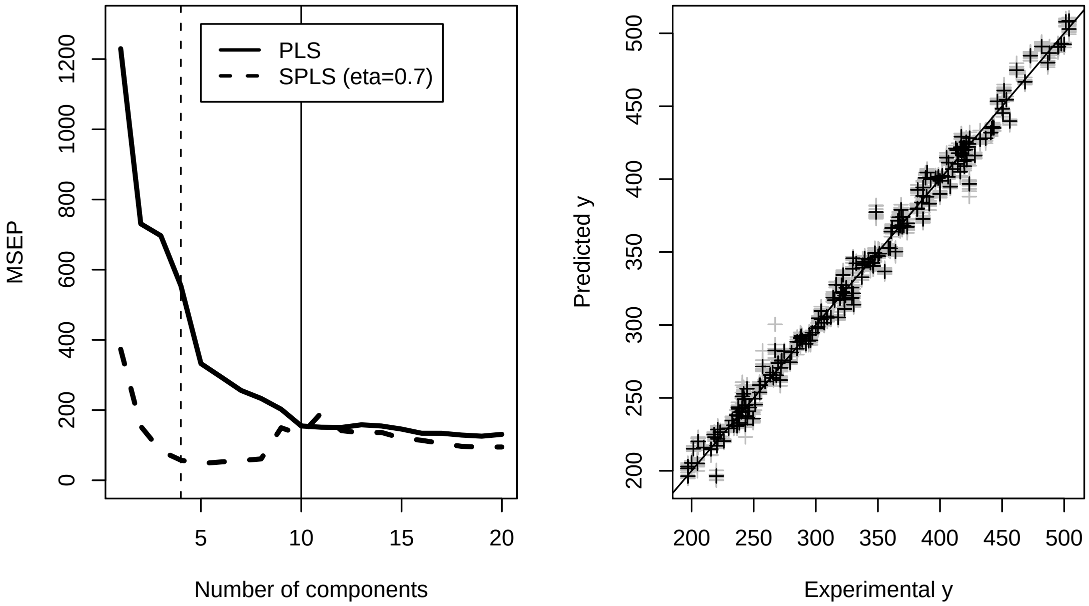

Figure 5: PAC data. Left: Mean squared error of prediction, measured through repeated double cross validation for different numbers of components. The vertical lines denote $a_{opt}$: The smallest number of components, for which the MSEP is still in range of two standard deviations of the overall minimum MSEP. Right: Experimental values $y_i$ versus the fitted values $\hat{y}_i$ for the final SPLS model. The set of fitted values obtained through the repeated double cross validation is painted gray, their averages are painted black.

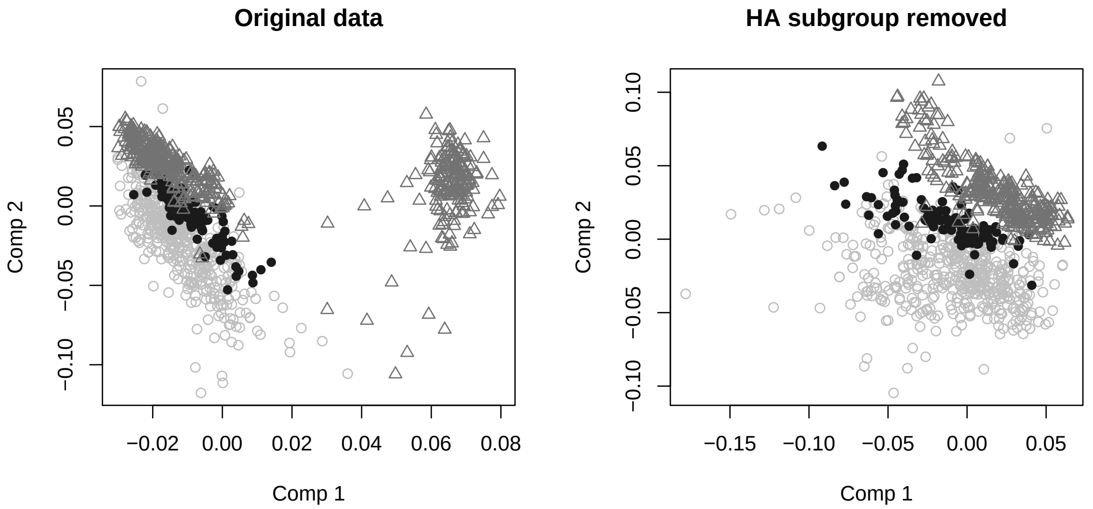

Figure 6: Visualization of the first two (non-sparse) PLSDA scores when using the complete group HA including the outliers (left) and the cleaned group HA (right).

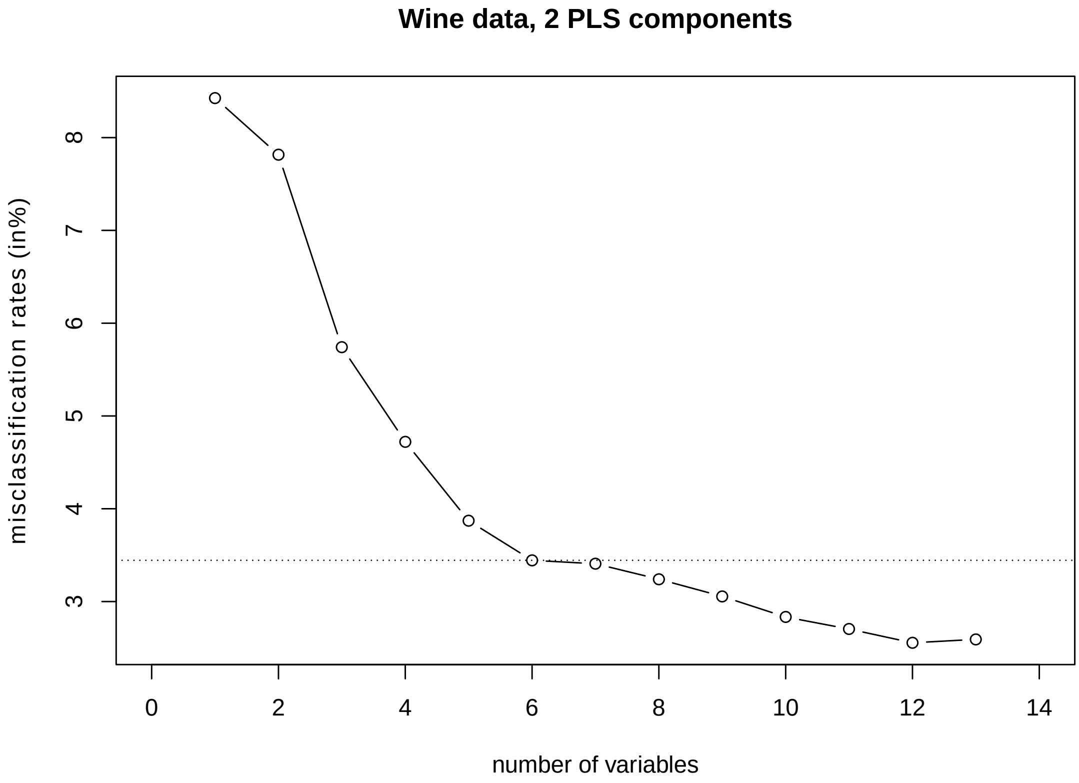

Figure 7: SPLSDA of Wine data. Average misclassification rate (from 100 random partitions of the data into training and test data sets) against the selected number of variables per SPLS component.