Penalized Partial Least Squares with Applications to B-Spline Transformations and Functional Data

Nicole Krämer [a,*]

[a] *Department of Electrical Engineering and Computer Science, TU Berlin, Franklinstr. 28/29, 10587 Berlin, Germany*

Anne-Laure Boulesteix [b,c]

[b] *Sylvia Lawry Centre for MS Research, Munich, Germany*

[c] *Department of Medical Statistics and Epidemiology, TU Munich, Germany*

Gerhard Tutz [d]

[d] *Department of Statistics, University of Munich, Germany*

Abstract

We propose a novel framework that combines penalization techniques with Partial Least Squares (PLS). We focus on two important applications. (1) We combine PLS with a roughness penalty to estimate high-dimensional regression problems with functional predictors and scalar response. (2) Starting with an additive model, we expand each variable in terms of a generous number of B-Spline basis functions. To prevent overfitting, we estimate the model by applying a penalized version of PLS. We gain additional model flexibility by incorporating a sparsity penalty. Both applications can be formulated in terms of a unified algorithm called Penalized Partial Least Squares, which can be computed virtually as fast as PLS using the kernel trick. Furthermore, we prove a close connection of penalized PLS to preconditioned linear systems. In experiments, we show the benefits of our method to noisy functional data and to sparse nonlinear regression models.

Key words: NIR spectroscopy, additive model, dimensionality reduction, nonlinear regression, conjugate gradient, Krylov spaces

* Corresponding Author
Email address: nkraemer@cs.tu-berlin.de (Nicole Krämer).

1 Introduction

The problem of high dimensionality in statistical data analysis has been tackled in many ways. Two generic strategies are (a) the reduction of the dimensionality of the data by selecting variables or derived components and (b) the regularization of the estimation process by imposing penalty terms that incorporate additional knowledge about the data. In this paper, we propose a combination of the dimensionality reduction technique Partial Least Squares with a penalization framework. Our motivation stems from two important applications, namely the smoothing of functional data and the estimation of additive models.

We speak of functional data [30] if the observed predictors are (discrete observations of) curves. Throughout this paper, we consider the case of functional predictor variables and scalar response variables. An important example in the context of chemometrics is near infra red (NIR) spectroscopy. The spectrum of a sample can be interpreted as a discretized function of the wavelength. Typically, the task is to predict a continuous response (e.g. the amount of fat) based on the spectrum of a sample. The number of wavelengths is typically in the range of a few hundreds, thus yielding a very high-dimensional regression problem. Consequently, some sort of regularization is needed. The standard approach in functional data analysis is to regularize the estimation process by imposing smoothness conditions, e.g. by penalizing the curvature of the functions. This penalization strategy typically yields smooth regression coefficients and is particularly beneficial if the measurements of the curves are noisy or if the observations are not measured at equidistant points. It is also common to represent the spectra in terms of basis functions as B-Splines or wavelets [22,4,26] before applying regression techniques [32,36]. A different approach is to use dimensionality reduction techniques such as Partial Least Squares (PLS) [42,43]. The main idea is to build a few components from the predictor variables and to regress $\boldsymbol{y}$ on these components. As an additional benefit, the derived components can be used for visualization and interpretation.

In this paper, we propose a combination of the penalization approach and PLS by adding a multiplicative penalty term to the optimization criterion of PLS. This is an extension of functional principal component analysis [39] to a supervised setting. The new method shares a lot of properties of PLS and its computation requires virtually no extra costs. In particular, we derive a so-called kernel representation of the method, that scales with the number of observations and not with the number of variables. More precisely, we prove that penalized PLS is equivalent to ordinary PLS using a generalized inner product that is defined by the penalty term.

A combination of PLS with penalty terms was first proposed in Goutis and Fearn [10] for data derived from NIR spectroscopy. More precisely, they suggest to incorporate an additive penalty term that leads to an eigenvalue problem for each PLS component. Compared to our approach, this is computationally

less efficient, moreover, it is – to our knowledge – not possible to derive a kernel representation. Goutis and Fearn [10] report that in experiments, the incorporation of a penalty term does not increase the performance of PLS on spectral data. This originates from the fact that the data considered in [10] are measured at equidistant points and smooth. In section 5, we illustrate in a simulation study that if these conditions do not hold, penalized PLS leads to considerably better predictions than PLS.
We highlight the close connection between penalized PLS and preconditioned linear systems. It is already known that PLS is equivalent to the conjugate gradient method [12] applied to the set of normal equations associated to a linear regression problem. We prove that penalized PLS corresponds to a conjugate gradient method for a preconditioned set of normal equations, where the preconditioner depends on the penalty term.
The second important application of our novel framework combining PLS dimensionality reduction with regularization is the estimation of additive models. Nonlinear regression effects may be modeled via additive models of the form

$$\mathsf{Y} = \beta_0 + f_1(\mathsf{X}_1) + \ldots + f_p(\mathsf{X}_p) + \varepsilon\,, \tag{1}$$

where the functions $f_1, \ldots, f_p$ are represented in terms of basis functions [11]. To prevent overfitting, there are two general approaches. In the first approach, each function f_j is the sum of only a small set of basis functions,

$$f_j(x) = \sum_{k=1}^{K_j} \beta_{kj} B_{kj}(x)\,. \tag{2}$$

The basis functions B_{kj} are chosen adaptively by a selection procedure. In the second approach, we allow a generous number $K_j \gg 1$ of basis functions in (2). As this usually leads to high-dimensional and highly correlated data, we penalize the coefficients β_{jk} in the estimation process [9]. An efficient variant is introduced in [44,45]. The smoothing parameters are estimated via a gradient descent on a generalized cross-validation score.
As a linear approach, PLS probably fails to yield high prediction accuracy in the case of nonlinear relationships as in (1). Therefore, it is necessary to transform the original predictors prior to a PLS regression. This approach has been proposed in two different variants. The first method [8] is based on a variant of PLS that is computed via an iterative algorithm. This approach incorporates spline transformations of the predictors within each iteration of the iterative algorithm. In contrast, the method proposed by Durand [7] is global. The predictors are first transformed using spline basis functions as a preliminary step, then PLS regression is performed on the transformed data matrix. The choice of the degree of the polynomial pieces and of the number of knots is performed by an either ascending or descending search procedure

that is not automatic.
For large numbers of variables, this search procedure is computationally infeasible and might overfit the data. As a second application of the penalized PLS methodology presented in this article, we suggest to combine it with the penalty strategy of Eilers and Marx [9] in the context of additive regression models. We transform the initial data matrix nonlinearly using B-spline basis functions. We then apply penalized PLS to the transformed data by penalizing the (higher order) differences of weight vectors. As the estimated regression coefficients are linear combinations of the smoothed weight vectors, the obtained function is smooth as well. We introduce more flexibility by adding a sparsity constraint, which leads to nonlinear variable selection for the additive model (1). We illustrate the usefulness of this new method on a data set from sensometrics.
The proposed methods are implemented in the `R`-package `ppls` [17] that is publicly available at `http://cran.r-project.org`.

2 Background

In this section, we briefly recapitulate the main techniques that are needed in the rest of the paper. We start with the smoothing based approaches for functional data (Subsection 2.1) corresponding to our first application and regression splines (Subsection 2.2) corresponding to our second application, and then introduce the Partial Least Squares approach (Subsection 2.3).

2.1 Roughness Penalties for Functional Data

In a nutshell, we speak of functional data [30] if the variables that we observe are discrete observations of curves. We focus on the case that the predictor variables $\mathsf{X}_1, \ldots, \mathsf{X}_p$ are measurements of curves $x : T \to \mathbb{R}$ at p distinct points $t_1 < \ldots < t_p$ in the interval T. The n observed functions $x_i : T \to \mathbb{R}$ (with $i = 1, \ldots, n$) are then represented by vectors $\boldsymbol{x}_i \in \mathbb{R}^p$ via

$$\boldsymbol{x}_i = (x_i(t_1), \ldots, x_i(t_p))^\top \,. \tag{3}$$

For each curve x_i, we observe a scalar response $y_i \in \mathbb{R}$. The corresponding linear regression model is given by

$$\mathsf{Y}_i = \beta_0 + \int_T \beta(t) x_i(t) dt + \varepsilon_i \,,$$

with $\beta : T \to \mathbb{R}$. We can transform this into a multiple regression problem by estimating $\beta(t)$ at the discrete points $t_1, \ldots, t_p$, i.e. we estimate $\boldsymbol{\beta} = (\beta(t_1), \ldots, \beta(t_p))^\top$. As this leads to a high-dimensional regression problem, the ordinary least squares criterion

$$\widehat{\boldsymbol{\beta}}_{\text{ols}} = \text{argmin}_{\boldsymbol{\beta}} \|\boldsymbol{y} - \beta_0 \mathbf{1}_n - \boldsymbol{X}\boldsymbol{\beta}\| \tag{4}$$

(with $\boldsymbol{y} = (y_1, \ldots, y_n)$ and $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$) is regularized by imposing a roughness penalty on $\beta(t)$. Typically, the curvature of β, i.e the squared second derivative of $\beta(t)$, is penalized. This penalty term can be approximated in terms of $\boldsymbol{\beta}$ by computing the second order differences of the coefficients. For points $t_1 < \ldots < t_p$, the penalty is given as

$$P(\beta) \approx \lambda \boldsymbol{\beta}^\top \left(\boldsymbol{D}_{p-2}\boldsymbol{D}_{p-1}\right)^\top \left(\boldsymbol{D}_{p-2}\boldsymbol{D}_{p-1}\right)\boldsymbol{\beta} = \lambda \boldsymbol{\beta}^\top \boldsymbol{P}\boldsymbol{\beta}\,, \quad \lambda \geq 0\,.$$

Here, the $(K-1) \times K$ matrix $\boldsymbol{D}_K$

$$\boldsymbol{D}_K = \begin{pmatrix} h_1 & -h_1 & . & . & . \\ . & h_2 & -h_2 & . & . \\ . & . & . & . & . \\ . & . & . & h_{K-1} & -h_{K-1} \end{pmatrix}, \; h_j = \frac{1}{t_j - t_{j+1}} \tag{5}$$

defines the first order difference operator. Note that for equidistant measurements, h_j does not depend on j and in this case, we assume that $h_j = 1$. We remark that the penalty matrix $\lambda\boldsymbol{P}$ is positive semi-definite. The roughness penalty $\lambda\boldsymbol{\beta}^\top\boldsymbol{P}\boldsymbol{\beta}$ is added to the ordinary least squares criterion (4). The scalar $\lambda \geq 0$ controls the amount of smoothing. The penalization strategy is particularly beneficial if the observations (3) of the curves are noisy or if the discrete observations are not measured at equidistant points. To retrieve a function $\widehat{\beta} : T \to \mathbb{R}$ from the discrete estimates $\widehat{\boldsymbol{\beta}}$, one typically uses smoothing techniques.

2.2 Penalized Regression Splines

The basic concept of penalized regression splines is to expand each predictor variable X_j in basis functions as in (2) and to estimate the coefficients by penalization techniques. As suggested by Eilers and Marx [9], B-splines [6] are used as basis functions yielding so-called P-splines (for penalized B-splines). Splines are one-dimensional piecewise polynomial functions of a certain degree

d that are joined at a set of knots. For a given variable X_j, we consider a set of K corresponding B-spline basis functions $B_{1j}, \ldots, B_{Kj}$. These basis functions define a nonlinear map $\Phi_j(x) = (B_{1j}(x), \ldots, B_{Kj}(x))^\top$. By performing such a transformation on each of the variables $\mathsf{X}_1, \ldots, \mathsf{X}_p$, an observation vector $\widetilde{\boldsymbol{x}}_i$ of the original variables turns into a vector

$$\boldsymbol{x}_i = (B_{11}(\widetilde{x}_{i1}), \ldots, B_{K1}(\widetilde{x}_{i1}), \ldots, B_{1p}(\widetilde{x}_{ip}), \ldots, B_{Kp}(\widetilde{x}_{ip}))^\top = \Phi(\widetilde{\boldsymbol{x}}_i) \qquad (6)$$

of length pK. Here $\Phi = (\Phi_1, \ldots, \Phi_p)$ is the function defined by the B-splines. The resulting data matrix obtained by the transformation of the original observations is denoted by $\boldsymbol{X} \in \mathbb{R}^{n \times (pK)}$. Cubic splines (i.e. $d = 3$) are the most widely used splines.
The estimation of (1) is transformed into the estimation of the intercept β_0 and the pK-dimensional vector that consists of the coefficients β_{jk}:

$$\boldsymbol{\beta}^\top = (\beta_{11}, \ldots, \beta_{K1}, \ldots, \beta_{1p}, \ldots, \beta_{Kp}) = \left(\boldsymbol{\beta}_{(1)}^\top, \ldots, \boldsymbol{\beta}_{(p)}^\top\right) .$$

Hence, the nonlinear additive function in (1) can be written as $f = \beta_0 + \boldsymbol{x}^\top \boldsymbol{\beta}$. As the transformed data are usually high-dimensional, the estimation of $\boldsymbol{\beta}$ by minimizing the squared error (4) typically leads to overfitting. Following [9], we use for each variable many basis functions, say $K \approx 25$, and estimate by penalizing the squared second derivative of the function f. Eilers and Marx [9] show that the following difference penalty term is a good approximation of the penalty on the second derivative of f_j,

$$P_j\left(\boldsymbol{\beta}_{(j)}\right) = \lambda_j \boldsymbol{\beta}_{(j)}^\top (\boldsymbol{D}_{K-1}\boldsymbol{D}_K)^\top \boldsymbol{D}_{K-1}\boldsymbol{D}_K \boldsymbol{\beta}_{(j)} .$$

The matrix $\boldsymbol{D}_K$ of first order differences of adjacent parameters is defined in (5) with $h_j = 1$. Hence, we penalize the second order differences for each vector $\boldsymbol{\beta}_{(j)}$. The penalty term P_j coincides with the roughness penalty term for functional data introduced in Section 2.1 in the case of equidistant measurements.
Setting $\boldsymbol{K}_2 = (\boldsymbol{D}_{K-1}\boldsymbol{D}_K)^\top \boldsymbol{D}_{K-1}\boldsymbol{D}_K$, we conclude that the penalty term equals

$$P(\boldsymbol{\beta}) = \sum_{j=1}^{p} P_j\left(\boldsymbol{\beta}_{(j)}\right) = \boldsymbol{\beta}^\top (\boldsymbol{\Delta}_{\boldsymbol{\lambda}} \otimes \boldsymbol{K}_2)\boldsymbol{\beta} = \boldsymbol{\beta}^\top \boldsymbol{P}\boldsymbol{\beta} . \qquad (7)$$

Here $\boldsymbol{\Delta}_{\boldsymbol{\lambda}}$ is the $p \times p$ diagonal matrix containing $\lambda_1, \ldots, \lambda_p$ on its diagonal and $\otimes$ is the Kronecker product. The generalization of this method to higher-order differences of the coefficients of adjacent B-splines is straightforward. Note furthermore that $\boldsymbol{P}$ is a block-diagonal and symmetric matrix that is positive semi-definite.

If the number n of observations is small compared to the number p of predictor variables, ordinary least squares (OLS) regression usually fits the training data perfectly and one cannot expect the fitted model to perform well on a new data set. Partial Least Squares (PLS) [42,43] is an alternative regression tool which is more appropriate in the case of highly correlated predictors and high-dimensional data. PLS is a standard tool e.g. for analyzing chemical data [23], and the success of PLS has lead to applications in other scientific fields such as chemoinformatice, physiology or bioinformatics [37,35,2].
The main idea of PLS is to build orthogonal components $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_m$ from the data $\boldsymbol{X}$ and to use them as predictors in a least squares fit (4). There are different PLS techniques to extract these components, and each of them gives rise to a different variant of PLS. It is not our aim to explain all variants (an we refer to an overview of different forms of PLS in [33]). A component is a linear combination of the original predictors that hopefully reflects the relevant structure of the data, and PLS extracts components that have a large covariance with $\boldsymbol{y}$. We now formalize this concept. A latent component $\boldsymbol{t}$ is a linear combination $\boldsymbol{t} = \boldsymbol{X}\boldsymbol{w}$ of the predictor variables. The vector $\boldsymbol{w}$ is called the weight vector. We want to find a component with maximal covariance to $\boldsymbol{y}$, that is, for the first component $\boldsymbol{t}_1 = \boldsymbol{X}\boldsymbol{w}_1$ we maximize the empirical squared covariance

$$\boldsymbol{w}_1 = \operatorname{argmax}_{\boldsymbol{w}} \frac{\operatorname{cov}^2(\boldsymbol{X}\boldsymbol{w}, \boldsymbol{y})}{\boldsymbol{w}^\top \boldsymbol{w}} = \operatorname{argmax}_{\boldsymbol{w}} \frac{\boldsymbol{w}^\top \boldsymbol{X}^\top \boldsymbol{y}\boldsymbol{y}^\top \boldsymbol{X}\boldsymbol{w}}{\boldsymbol{w}^\top \boldsymbol{w}}\,. \tag{8}$$

The solution of (8) is unique up to a scalar and equals $\boldsymbol{w}_1 = \boldsymbol{X}^\top \boldsymbol{y}$. Subsequent components $\boldsymbol{t}_2, \boldsymbol{t}_3, \ldots$ are chosen such that they maximize (8) subject to mutual orthogonality of all components $\boldsymbol{t}_i$. This can be achieved by deflating the original predictor variables $\boldsymbol{X}$. That is, we only consider the part of $\boldsymbol{X}$ that is orthogonal to all components $\boldsymbol{t}_j, j < i$. For any matrix $\boldsymbol{V}$, let us denote by $\mathcal{P}_{\boldsymbol{V}}$ the orthogonal projection to the space that is spanned by the columns of $\boldsymbol{V}$. In matrix notation, we have $\mathcal{P}_{\boldsymbol{V}} = \boldsymbol{V}\left(\boldsymbol{V}^\top \boldsymbol{V}\right)^+ \boldsymbol{V}^\top$. Here, the superscript "+" denotes the Moore-Penrose inverse. The deflation of $\boldsymbol{X}$ with respect to the components $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{i-1}$ is defined as

$$\boldsymbol{X}_i = \boldsymbol{X} - \mathcal{P}_{\boldsymbol{t}_1,\ldots,\boldsymbol{t}_{i-1}}\boldsymbol{X} = \boldsymbol{X}_{i-1} - \mathcal{P}_{\boldsymbol{t}_{i-1}}\boldsymbol{X}_{i-1}\,. \tag{9}$$

For the computation of the ith component, $\boldsymbol{X}$ is replaced by $\boldsymbol{X}_i$ in (8). This method is called the NIPALS algorithm [42], and is summarized in algorithm 1. In order to obtain the response for new observations, we have to determine the vector of regression coefficients $\widehat{\boldsymbol{\beta}}_m$ that are defined via

$$\widehat{\boldsymbol{y}}_m = \mathcal{P}_{\boldsymbol{t}_1,\ldots,\boldsymbol{t}_m}\boldsymbol{y} = \boldsymbol{X}\widehat{\boldsymbol{\beta}}_m\,.$$

Algorithm 1 NIPALS algorithm

Input: $\boldsymbol{X}_1 = \boldsymbol{X}$, $\boldsymbol{y}$, m
for i=1,...,m **do**
(a) $\boldsymbol{w}_i = \boldsymbol{X}_i^\top \boldsymbol{y}$ (weight vector) (b) $\boldsymbol{w}_i = \boldsymbol{w}_i / \|\boldsymbol{w}_i\|$ (normalization)
(a) $\boldsymbol{t}_i = \boldsymbol{X}_i \boldsymbol{w}_i$ (component) (b) $\boldsymbol{t}_i = \boldsymbol{t}_i / \|\boldsymbol{t}_i\|$ (normalization)
$\boldsymbol{X}_{i+1} = \boldsymbol{X}_i - \mathcal{P}_{\boldsymbol{t}_i} \boldsymbol{X}_i$ (deflation)
end for

This can be done efficiently [20]. More details are discussed in Section 3.
We note that de Jong [14] introduced the SIMPLS algorithm for PLS which avoids the explicit deflation step (9). The optimization criterion for the weight vectors $\boldsymbol{w}_j$ is

$$\boldsymbol{w}_j = \text{argmax}_{\boldsymbol{w}} \frac{\text{cov}^2 (\boldsymbol{X}\boldsymbol{w}, \boldsymbol{y})}{\boldsymbol{w}^\top \boldsymbol{w}} \quad \text{subject to } \boldsymbol{X}\boldsymbol{w}_j \perp \boldsymbol{X}\boldsymbol{w}_i \,,\, i < j \,. \tag{10}$$

For univariate response vectors $\boldsymbol{y}$, both algorithms – NIPALS and SIMPLS – are equivalent.

3 Penalized Partial Least Squares

We now introduce a general framework to combine PLS with penalization terms. Functional data analysis and additive models with splines are the two main motivating applications which we consider in this article. However, our method is not limited to these particular cases. For this reason, we only assume that $\boldsymbol{P}$ is a symmetric positive semi-definite matrix.

3.1 General Framework

We modify the optimization criterion (8) of PLS in the following way. The first component $\boldsymbol{t}_1 = \boldsymbol{X}\boldsymbol{w}_1$ is defined by the solution of the problem

$$\text{argmax}_{\boldsymbol{w}} \frac{\boldsymbol{w}^\top \boldsymbol{X}^\top \boldsymbol{y}\boldsymbol{y}^\top \boldsymbol{X}\boldsymbol{w}}{\boldsymbol{w}^\top \boldsymbol{w} + \boldsymbol{w}^\top \boldsymbol{P}\boldsymbol{w}} \,. \tag{11}$$

We obtain $\boldsymbol{w}_1 = \boldsymbol{M}\boldsymbol{X}^\top \boldsymbol{y}$ with $\boldsymbol{M} = (\boldsymbol{I}_p + \boldsymbol{P})^{-1}$. Subsequent weight vectors and components are computed by deflating $\boldsymbol{X}$ as described in (9) and then maximizing (11) with $\boldsymbol{X}$ replaced by $\boldsymbol{X}_i$. In particular, we can compute the weight vectors and components of penalized PLS by simply replacing $\boldsymbol{w}_i = \boldsymbol{X}_i^\top \boldsymbol{y}$ by $\boldsymbol{w}_i = \boldsymbol{M}\boldsymbol{X}_i^\top \boldsymbol{y}$ in algorithm 1. This leads to the penalized PLS algorithm 2.

Algorithm 2 Penalized PLS algorithm

Input: $\boldsymbol{X}_1 = \boldsymbol{X}$, $\boldsymbol{y}$, m, $\boldsymbol{P}$
$\boldsymbol{M} = (\boldsymbol{I} + \boldsymbol{P})^{-1}$
for i=1,...,m **do**
(a) $\boldsymbol{w}_i = \boldsymbol{M}\boldsymbol{X}_i^\top \boldsymbol{y}$ (weight vector) (b) $\boldsymbol{w}_i = \boldsymbol{w}_i / \|\boldsymbol{w}_i\|$ (normalization)
(a) $\boldsymbol{t}_i = \boldsymbol{X}_i \boldsymbol{w}_i$ (component) (b) $\boldsymbol{t}_i = \boldsymbol{t}_i / \|\boldsymbol{t}_i\|$ (normalization)
$\boldsymbol{X}_{i+1} = \boldsymbol{X}_i - \mathcal{P}_{\boldsymbol{t}_i} \boldsymbol{X}_i$ (deflation)
end for

Let $\boldsymbol{T} = (\boldsymbol{t}_1, \ldots, \boldsymbol{t}_m)$ and $\boldsymbol{W} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_m)$ denote the matrices of components and weight vectors that are defined by the penalized PLS algorithm 2. The following proposition shows how to compute the vector of regression coefficients for penalized PLS.

Proposition 1 *The matrix*

$$(\boldsymbol{R}_m =)\, \boldsymbol{R} = \boldsymbol{T}^\top \boldsymbol{X}\ \boldsymbol{W}$$

is upper bidiagonal, that is $r_{ij} = \boldsymbol{t}_i^\top \boldsymbol{X} \boldsymbol{w}_j = 0$ *if* $i > j$ *or* $i + 1 < j$. *The matrix* $\boldsymbol{R}$ *is invertible. Furthermore, setting* $\widetilde{\boldsymbol{D}} = \mathrm{diag}(1/\|\boldsymbol{t}_1\|, \ldots, 1/\|\boldsymbol{t}_m\|)$, *we have*

$$\boldsymbol{X}\boldsymbol{W} = \left(\boldsymbol{T}\widetilde{\boldsymbol{D}}\right)\left(\widetilde{\boldsymbol{D}}\boldsymbol{R}\right). \tag{12}$$

In particular, the columns of $\boldsymbol{T}$ *and the columns of* $\boldsymbol{X}\boldsymbol{W}$ *span the same space, and the penalized PLS regression vector obtained after* m *steps is*

$$\widehat{\boldsymbol{\beta}}_m = \boldsymbol{W}\boldsymbol{R}^{-1}\boldsymbol{T}^\top \boldsymbol{y}. \tag{13}$$

This is an extension of a result for ordinary PLS that can be found e.g. in [20]. Note that (12) is in fact the QR-decomposition of $\boldsymbol{X}\boldsymbol{W}$.
This result is beneficial for two reasons. First, the inverse of $\boldsymbol{R}$ can be computed very fast as the matrix is upper-triangular. Second, for all PLS components $i \leq m$ the inverse of $\boldsymbol{R}_i$ is simply the submatrix of the inverse of $\boldsymbol{R}_m$ that consists of the first i rows and columns. Combining this result with the PLS algorithm 2, we obtain the penalized PLS algorithm 3. Furthermore, (13) shows that the regression vector is a linear combination of the weight vectors $\boldsymbol{w}_i$. Hence a smoothing of the weight vectors leads to smooth estimates of $\boldsymbol{\beta}$.

It is also possible to derive penalized PLS with the help of the SIMPLS algorithm by replacing the covariance by the penalized covariance in criterion (10). As we consider a univariate response, both methods can be shown to be equivalent [16].
We now illustrate the influence of the penalty term $\lambda\boldsymbol{P}$ and the number m

Algorithm 3 Penalized PLS algorithm

Input: $\boldsymbol{X}_1 = \boldsymbol{X}$, $\boldsymbol{y}$, m, $\boldsymbol{P}$
$\boldsymbol{M} = (\boldsymbol{I}_p + \boldsymbol{P})^{-1}$
for i=1,...,m **do**
  $\boldsymbol{w}_i = \boldsymbol{M}\boldsymbol{X}_i^\top \boldsymbol{y}$ (weight vector)
  $\boldsymbol{t}_i = \boldsymbol{X}_i \boldsymbol{w}_i$ (component)
  $\boldsymbol{X}_{i+1} = \boldsymbol{X}_i - \mathcal{P}_{\boldsymbol{t}_i} \boldsymbol{X}_i$ (deflation)
end for
$\boldsymbol{L} = \left(\boldsymbol{T}^\top \boldsymbol{X} \boldsymbol{W}\right)^{-1}$ (inverse of $\boldsymbol{R}_m$)
for i=1,...,m **do**
  $\boldsymbol{L}_i$ (first i rows and columns of $\boldsymbol{L}$)
  $\widehat{\boldsymbol{\beta}}_i = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_i)\, \boldsymbol{L}_i\, (\boldsymbol{t}_1, \ldots, \boldsymbol{t}_i)^\top \boldsymbol{y}$ (regression vector)
end for

of components for B-Spline transformations with one predictor. The BOD (biochemical oxygen demand) data set [21] consists of six measurements. The predictor variable is the time of measurement, the response variable is the biochemical oxygen demand. This data set is part of the R software [29]. We use penalized PLS on the B-spline transformed data. We fix the number of knots to 25 and choose cubic splines, i.e. $d = 3$. In Figure 1 we plot the fitted functions obtained from penalized PLS for different numbers of components (from left to right: $1, 2, 3$) and different values of the smoothing parameter λ (from top to bottom: $\lambda = 2000; 20; 0$). If we compare the results for different

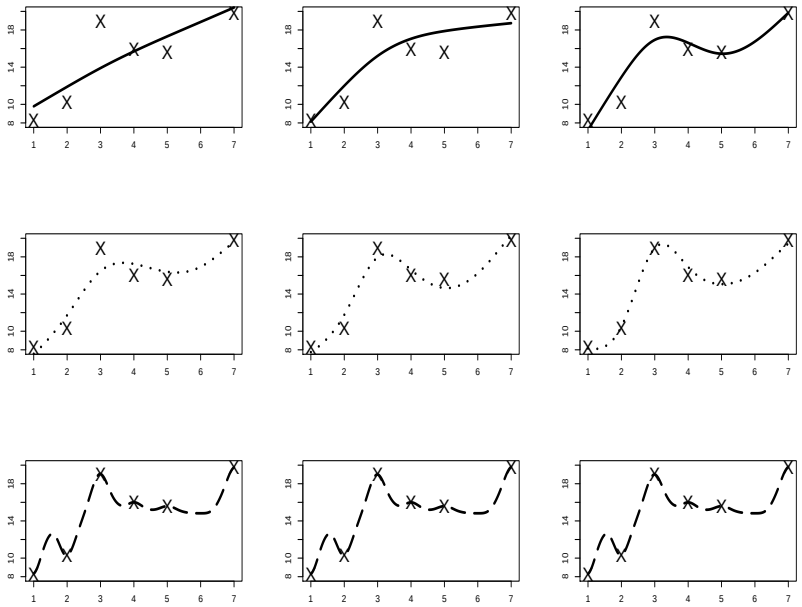

Fig. 1. Penalized PLS for different numbers of components (from left to right: $1; 2; 3$) and different values of the smoothing parameter λ (from top to bottom: $2000; 20; 0$).

values of λ (i.e. from top to bottom) we see that the penalty term indeed con-

trols the curvature of the functions. With no penalization (bottom line), the model already overfits for one PLS component. Moreover, the number of PLS components also controls the smoothness of the estimated functions. For small values of m, the obtained functions are very smooth. For higher values of m, they adapt themselves more and more to the data, which leads to overfitting. To summarize, the two model parameters influence the shape of the functions in opposite directions. High values of λ and low values of m lead to smooth functions.

3.2 Kernel Representation of Penalized Partial Least Squares

The computation of the penalized PLS estimator as presented in algorithm 3 involves matrices and vectors of dimension $p \times p$ and p respectively. If the number of predictors p is very large, this leads to high computational costs. In this subsection, we show that we can represent this algorithm in terms of a so-called kernel matrix (of dimension $n \times n$) and $\boldsymbol{y}$. This strategy is known as the kernel trick [38,25]. Note that kernel versions of PLS have been derived in [31,34].
We define the $n \times n$ kernel matrix $\boldsymbol{K_M} = (\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle_{\boldsymbol{M}}) = \boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^\top$. It consists of the inner product of observations $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, where the inner product is defined via the positive definite matrix $\boldsymbol{M}$. The key is to find a representation $\widehat{\boldsymbol{\beta}}_m = \boldsymbol{M}\boldsymbol{X}^\top \widehat{\boldsymbol{\alpha}}_m$ of the regression vector in terms of kernel coefficients $\widehat{\boldsymbol{\alpha}}_m \in \mathbb{R}^n$. This can be accomplished by noting that

$$\boldsymbol{w}_i = \boldsymbol{M}\boldsymbol{X}_i^\top \boldsymbol{y} = \boldsymbol{M}\boldsymbol{X}^\top \left(\boldsymbol{I}_n - \mathcal{P}_{\boldsymbol{T}}\right) \boldsymbol{y} = \boldsymbol{M}\boldsymbol{X}^\top \left(\boldsymbol{y} - \widehat{\boldsymbol{y}}_{i-1}\right) = \boldsymbol{M}\boldsymbol{X}^\top \boldsymbol{u}_i$$

with $\boldsymbol{u}_i = \boldsymbol{y} - \widehat{\boldsymbol{y}}_{i-1}$ defined as the residuals in each step. Furthermore, it follows from the bidiagonality of $\boldsymbol{R}$ that

$$\boldsymbol{t}_i = \boldsymbol{X}_i \boldsymbol{w}_i = \left(\boldsymbol{I}_n - \mathcal{P}_{t_{i-1}}\right) \boldsymbol{X}\boldsymbol{w}_i = \left(\boldsymbol{I}_n - \mathcal{P}_{t_{i-1}}\right) \boldsymbol{K_M}\boldsymbol{u}_i \,.$$

Finally, we have $\boldsymbol{R} = \boldsymbol{T}^\top \boldsymbol{X}\boldsymbol{W} = \boldsymbol{T}^\top \boldsymbol{K_M}\boldsymbol{U}$ with $\boldsymbol{U} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m)$. We can now derive algorithm 4 for the kernel coefficients.

The kernel algorithm 4 reveals that penalized PLS equals ordinary PLS with the canonical inner product replaced by the inner product $\langle \boldsymbol{x}, \boldsymbol{z} \rangle_{\boldsymbol{M}} = \boldsymbol{x}^\top \boldsymbol{M}\boldsymbol{z}$. Why is this a sensible inner product? Let us consider the eigen decomposition of the penalty matrix, $\boldsymbol{P} = \boldsymbol{S}\boldsymbol{\Theta}\boldsymbol{S}^\top$. We prefer direction $\boldsymbol{s}$ such that $\boldsymbol{s}^\top \boldsymbol{P}\boldsymbol{s}$ is small, as these directions are smooth. Hence, we prefer directions that are defined by eigenvectors $\boldsymbol{s}_i$ of $\boldsymbol{P}$ with a small corresponding eigenvalue θ_i. If we represent the vectors $\boldsymbol{x} = \boldsymbol{S}\boldsymbol{x}'$ and $\boldsymbol{z} = \boldsymbol{S}\boldsymbol{z}'$ in terms of the eigenvectors of

Algorithm 4 Kernel Penalized PLS algorithm

Input: $\boldsymbol{X}_1 = \boldsymbol{X}$, $\boldsymbol{y}$, m, $\boldsymbol{P}$
$\boldsymbol{M} = (\boldsymbol{I}_p + \boldsymbol{P})^{-1}$, $\boldsymbol{K_M} = \boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^\top$, $\widehat{\boldsymbol{y}}_0 = \boldsymbol{t}_0 = \boldsymbol{0}$
for i=1,...,m **do**
  $\boldsymbol{u}_i = \boldsymbol{y} - \widehat{\boldsymbol{y}}_{i-1}$ (residuals)
  $\boldsymbol{t}_i = \left(\boldsymbol{I}_n - \mathcal{P}_{t_{i-1}}\right) \boldsymbol{K_M}\boldsymbol{u}_i$ (component)
  $\widehat{\boldsymbol{y}}_i = \widehat{\boldsymbol{y}}_{i-1} + \mathcal{P}_{\boldsymbol{t}_i}\boldsymbol{y}$ (fitted values)
end for
$\boldsymbol{L} = \left(\boldsymbol{T}^\top \boldsymbol{K_M} \boldsymbol{U}\right)^{-1}$ (inverse of $\boldsymbol{R}_m$)
for i=1,...,m **do**
  $\boldsymbol{L}_i$ = first i rows and columns of $\boldsymbol{L}_m$
  $\widehat{\boldsymbol{\alpha}}_i = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_i)\, \boldsymbol{L}_i\, (\boldsymbol{t}_1, \ldots, \boldsymbol{t}_i)^\top \boldsymbol{y}$ (kernel coefficients)
end for

$\boldsymbol{P}$, we conclude that

$$\langle \boldsymbol{x}, \boldsymbol{z} \rangle_{\boldsymbol{M}} = (\boldsymbol{x}')^\top \left(\boldsymbol{I}_p + \boldsymbol{\Theta}\right)^{-1} \boldsymbol{z}' = \sum_{i=1}^{p} \frac{1}{1+\theta_i} \boldsymbol{x}_i' \boldsymbol{z}_i' \,.$$

This implies that directions $\boldsymbol{s}_i$ with a small eigenvalue θ_i receive a higher weighting than directions with a large eigenvalue.

3.3 Penalized Partial Least Squares and Krylov Subspaces

It is well-known that PLS is closely connected to Krylov subspaces and conjugate gradient methods. Quite generally, linear regression problems can be transformed into algebraic problems in the following way. The OLS estimator is the solution of the minimization problem (4). This is equivalent to finding the solution of the associated normal equation

$$\boldsymbol{A}\boldsymbol{\beta} = \boldsymbol{b} \tag{14}$$

with $\boldsymbol{b} = \boldsymbol{X}^\top \boldsymbol{y}$ and $\boldsymbol{A} = \boldsymbol{X}^\top \boldsymbol{X}$. If the matrix $\boldsymbol{A}$ is invertible, the solution of the normal equations is the OLS estimator $\widehat{\boldsymbol{\beta}} = \boldsymbol{A}^{-1}\boldsymbol{b}$. If $\boldsymbol{A}$ is singular, the solution of (14) with minimal Euclidean norm is $\boldsymbol{A}^{+}\boldsymbol{b}$. In the case of high dimensional data, the matrix $\boldsymbol{A}$ is often (almost) singular and the OLS estimate performs poorly on new data sets. A popular strategy is to regularize the least squares criterion (4) in the hope of improving the performance of the estimator. This often corresponds to finding approximate solutions of (14). For example, Ridge Regression corresponds to the solution of the modified normal equations $(\boldsymbol{A} + \lambda \boldsymbol{I}_p)\, \boldsymbol{\beta} = \boldsymbol{b}$. Here $\lambda > 0$ is the Ridge parameter. Principal

Components Regression uses the eigen decomposition of $\boldsymbol{A}$ and approximates $\boldsymbol{A}^+$ and $\boldsymbol{b}$ via the first m eigenvectors of $\boldsymbol{A}$.
It can be shown that the PLS estimators are equal to the approximate solutions of the conjugate gradient method [12]. This is a procedure that iteratively computes approximate solutions of (14) by minimizing the quadratic function

$$\phi(\boldsymbol{\beta}) = \frac{1}{2}\boldsymbol{\beta}^\top \boldsymbol{A}\boldsymbol{\beta} - \boldsymbol{\beta}^\top \boldsymbol{b} = \frac{1}{2}\langle \boldsymbol{\beta}, \boldsymbol{A}\boldsymbol{\beta}\rangle - \langle \boldsymbol{\beta}, \boldsymbol{b}\rangle \tag{15}$$

along directions that are $\mathbf{A}$-orthogonal. The approximate solution obtained after m steps is equal to the PLS estimator obtained after m iterations. The conjugate gradient algorithm is in turn closely related to Krylov subspaces and the Lanczos algorithm [19]. The latter is a method for approximating eigenvalues. The connection between PLS and these methods is well-elaborated in [28].
We now establish a similar connection between penalized PLS and the above mentioned methods. Set $\boldsymbol{A_M} = \boldsymbol{M}\boldsymbol{A}$ and $\boldsymbol{b_M} = \boldsymbol{M}\boldsymbol{b}$. Recall that $\boldsymbol{M}$ is the symmetric and positive definite matrix

$$\boldsymbol{M} = (\boldsymbol{I}_p + \boldsymbol{P})^{-1} \ .$$

We now illustrate that penalized PLS finds approximate solutions of the preconditioned normal equation

$$\boldsymbol{A_M}\boldsymbol{\beta} = \boldsymbol{b_M} \, . \tag{16}$$

Let us denote the space spanned by the sequence $\boldsymbol{b_M}, \boldsymbol{A_M}\boldsymbol{b_M}, \ldots, \boldsymbol{A_M}^{m-1}\boldsymbol{b_M}$ as the Krylov space $\mathcal{K}_m$ of $\boldsymbol{A_M}$ and $\boldsymbol{b_M}$.

Lemma 2 *The space spanned by the weight vectors $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_m$ of penalized PLS equals $\mathcal{K}_m$.*

This is the generalization of a result for ordinary PLS and can be shown via induction. Note that it follows from lemma 2 and the fact that $\boldsymbol{T}$ and $\boldsymbol{XW}$ span the same space that the penalized PLS estimator is the solution of the optimization problem (4) with the constraint $\boldsymbol{\beta} \in \mathcal{K}_m$.
We now present the conjugate gradient method for the equation

$$\boldsymbol{A_M}\boldsymbol{\beta} = \boldsymbol{b_M} \, . \tag{17}$$

Note that in general, the matrix $\boldsymbol{A_M}$ is not symmetric with respect to the canonical inner product, but with respect to the inner product $\langle \boldsymbol{x}, \widetilde{\boldsymbol{x}}\rangle_{\boldsymbol{M}^{-1}} =$

$\boldsymbol{x}^\top \boldsymbol{M}^{-1}\widetilde{\boldsymbol{x}}$ defined by $\boldsymbol{M}^{-1}$. We can rewrite the quadratic function ϕ defined in (15) as

$$\phi(\boldsymbol{\beta}) = \frac{1}{2}\left\langle \boldsymbol{\beta}, \boldsymbol{A_M}\boldsymbol{\beta}\right\rangle_{\boldsymbol{M}^{-1}} - \left\langle \boldsymbol{\beta}, \boldsymbol{b_M}\right\rangle_{\boldsymbol{M}^{-1}} .$$

We replace the canonical inner product by the inner product defined by $\boldsymbol{M}^{-1}$ and minimize this function iteratively along directions that are $\boldsymbol{A_M}$-orthogonal. We start with an initial guess $\boldsymbol{\beta}_0 = \boldsymbol{0}$ and define $\boldsymbol{d}_0 = \boldsymbol{r}_0 = \boldsymbol{b_M} - \boldsymbol{A_M}\boldsymbol{\beta}_0 = \boldsymbol{b_M}$. The quantity $\boldsymbol{d}_m$ is the search direction and $\boldsymbol{r}_m = \boldsymbol{b_M} - \boldsymbol{A_M}\boldsymbol{\beta}_{m-1}$ is the residual. For a given direction $\boldsymbol{d}_m$, we have to determine the optimal step size $a_m \in \mathbb{R}$ that minimized $\phi\left(\boldsymbol{\beta}_m + a_m\boldsymbol{d}_m\right)$. It is straightforward to check that

$$a_m = \frac{\left\langle \boldsymbol{d}_m, \boldsymbol{r}_m\right\rangle_{\boldsymbol{M}^{-1}}}{\left\langle \boldsymbol{d}_m, \boldsymbol{A_M}\boldsymbol{d}_m\right\rangle_{\boldsymbol{M}^{-1}}} .$$

The new approximate solution is then

$$\boldsymbol{\beta}_{m+1} = \boldsymbol{\beta}_m + a_m\boldsymbol{d}_m . \tag{18}$$

After updating the residuals via

$$\boldsymbol{r}_{m+1} = \boldsymbol{b_M} - \boldsymbol{A_M}\boldsymbol{\beta}_{m+1},$$

we define a new search direction $\boldsymbol{d}_{m+1}$ that is $\boldsymbol{A}_M$-orthogonal to the previous search directions. This is ensured by projecting the residual $\boldsymbol{r}_m$ to the space that is $\boldsymbol{A_M}$-orthogonal to $\boldsymbol{d}_0, \ldots, \boldsymbol{d}_m$. We obtain

$$\boldsymbol{d}_{m+1} = \boldsymbol{r}_{m+1} - \sum_{i=0}^{m} \frac{\left\langle \boldsymbol{r}_{m+1}, \boldsymbol{A_M}\boldsymbol{d}_i\right\rangle_{\boldsymbol{M}^{-1}}}{\left\langle \boldsymbol{d}_i, \boldsymbol{A_M}\boldsymbol{d}_i\right\rangle_{\boldsymbol{M}^{-1}}}\boldsymbol{d}_i .$$

Theorem 3 *The penalized PLS algorithm is equal to the conjugate gradient algorithm for the preconditioned system (17), that is $\boldsymbol{\beta}_m$ defined in (18) equals the penalized PLS estimator $\widehat{\boldsymbol{\beta}}_m$.*

The presentation of the conjugate gradient method above and the proof of its equivalence to penalized PLS are an extension of the corresponding results for PLS that is given in [28].
We conclude this subsection by remarking that the correspondence between penalized PLS and approximate solutions of the preconditioned equations (17) implies that after at most p iterations, the penalized PLS estimator equals $\boldsymbol{A}_{\boldsymbol{M}}^{+}\boldsymbol{b_M}$. If $\boldsymbol{A}$ is non-singular, this equals the OLS estimate.

4 Variable Selection in the Additive Model

For the estimation of additive models (1), penalized PLS depends on two types of model parameters (assuming that the number of knots is fixed). First, the number m of components and second, the degree of penalization λ_j for each variable X_j. Optimizing all $p+1$ model parameters is computationally infeasible for large values of p and might lead to overfitting. Therefore, we define a global penalty parameter $\lambda = \lambda_1 = \ldots = \lambda_p$. This of course restricts the flexibility of the model. Moreover, by definition of penalized PLS, the obtained model is not sparse. In general, all estimated functions f_j in (1) are non-zero. To overcome this restriction, we propose to add a sparsity constraint to penalized PLS.

Recall that the matrix $\boldsymbol{X}$ of B-spline transformed data (6) consists of p blocks $\boldsymbol{X}^{(j)}$ of p_j columns – with the jth block corresponding to the transformation of the jth variable. Moreover, the penalty matrix is a block-diagonal matrix, with each block $\boldsymbol{P}_j$ penalizing the (higher order) differences of the jth block of variables. This implies that the weight vector $\boldsymbol{w}$ for penalized PLS can be decomposed into p blocks, and the jth block maximizes the penalized covariance of $\boldsymbol{y}$ to the jth block of variables:

$$\begin{aligned}\boldsymbol{w}_i &= \left(\boldsymbol{w}_i^{(1)}, \ldots, \boldsymbol{w}_i^{(p)}\right) \\ \boldsymbol{w}_i^{(j)} &= (\boldsymbol{I} + \boldsymbol{P}_j)^{-1} \left(\boldsymbol{X}_i^{(j)}\right)^\top \boldsymbol{y}\end{aligned}$$

A sparse additive model (1) corresponds to weight vectors $\boldsymbol{w}_i$ that are *blockwise* sparse, i.e. some of the vectors $\boldsymbol{w}_i^{(j)}$ are equal to $\boldsymbol{0}$. Therefore, in each iteration step, we propose to select the block j^* that maximizes the penalized covariance to $\boldsymbol{y}$ and to define the penalized PLS vector only in terms of this *one* block:

$$\begin{aligned}j^* &= \operatorname{argmax}_{\|\boldsymbol{w}^{(j)}\|=1} \frac{1}{p_j} \operatorname{cov}^2\left(\boldsymbol{X}_i^{(j)} \boldsymbol{w}^{(j)}, \boldsymbol{y}\right) \\ \boldsymbol{w}_i &= \left(\boldsymbol{0}, \ldots, \boldsymbol{0}, (\boldsymbol{I} + \boldsymbol{P}_{j^*})^{-1} \left(\boldsymbol{X}_i^{(j^*)}\right)^\top \boldsymbol{y}, \boldsymbol{0}, \ldots, \boldsymbol{0}\right)\end{aligned} \tag{19}$$

In this way, only one block enters the model in each iteration step. The computation of the latent components $\boldsymbol{t}_i = \boldsymbol{X}_i \boldsymbol{w}_i$ and the deflation step remain unchanged. We note that the term $1/p_j$ in (19) is added to remove the bias introduced by different sizes of the blocks $\boldsymbol{X}^{(j)}$.

It is straightforward to show that all results of proposition 1 still hold, except for the fact that the matrix $\boldsymbol{R}$ is only upper-triangular, not upper bidiagonal. In particular, (13) shows that the linear combination of the block-wise sparse weight vectors $\boldsymbol{w}_i$ leads to a block-wise sparse regression vector $\boldsymbol{\beta}$. The sparsity is controlled by the number of components. In the next section, we discuss

an application of this method to a data set from sensometrics.

5 Experiments and Discussion

In this section, we assess the performance of penalized PLS. The first two Subsections 5.1 and 5.2 correspond to the modeling of functional data, and the last Subsection 5.3 corresponds to additive models.
In all experiments, the model parameters are optimized via cross-validation. We use a two-dimensional grid defined by a range of penalty terms and a range of components. An `R`-implementation of penalized PLS including model selection is available [17]. As penalized PLS is equal to Kernel PLS for a generalized inner product (see Subsection 3.2), it is possible to derive an unbiased estimate of its degrees of freedom [18]. Hence, model selection strategies based on information criteria are possible as well.

5.1 Simulation Study: Noisy Functional Data

First, we investigate the effect of noise to the prediction performance of PLS and penalized PLS. The importance of the effect of noise in the predictor variables is studied e.g. in [26]. The following example [1] is taken from [27] and is also discussed in [3]. The data consist of a training set of size 39 and a test set of size 31. The task is to predict with high accuracy the amount of fat in biscuit dough based on its NIR spectra. For each of the $n = 39 + 31 = 70$ observations of biscuit dough, the amount of fat and the reflectance of NIR light for $p = 700$ equidistant wavelengths in the range from 1100 to 2398 nanometers are measured. For each example, we obtain a discretized function $\boldsymbol{x}_i$ of the reflectance, which is called a spectrum.
The simulation set-up is as follows. For a fixed level $\sigma = 0, 0.01, \ldots, 0.09, 0.1$, we add noise to each sample

$$\boldsymbol{x}_i^{\text{noisy}} = \boldsymbol{x}_i + \varepsilon_i \in \mathbb{R}^{700} \qquad \varepsilon_i \sim \mathcal{N}\left(0, \sigma^2 I_{700}\right) .$$

Figure 2 displays a noisy observation for three different values of σ.

We then split the whole data set into a training set of size 39 and a test set of size 31. We estimate the optimal PLS and penalized PLS model on the training data using 10fold cross-validation. The quality of the two models are assessed by computing the mean squared error (mse) on the test data. This procedure is repeated 20 times for each value of σ. Figure 3 displays the results.

[1] available at: `http://www.stat.tamu.edu/~mvannucci/webpages/codes.html`

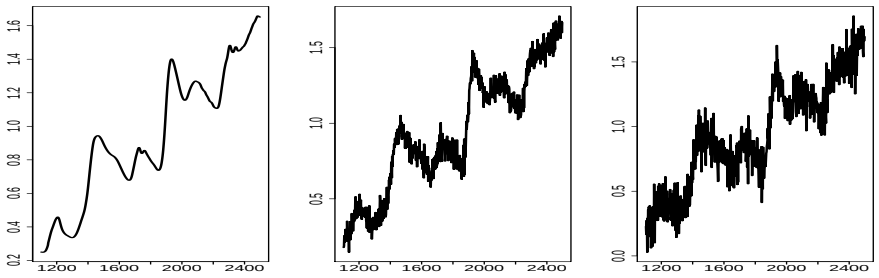

Fig. 2. One sample of the biscuit dough data set with with added noise. The noise levels are $\sigma = 0$ (left), $\sigma = 0.05$ (center) and $\sigma = 0.1$ (right)

It shows the median mse $\pm$ the median absolute deviation over the 20 runs. The figure indicates that the penalization approach significantly improves the

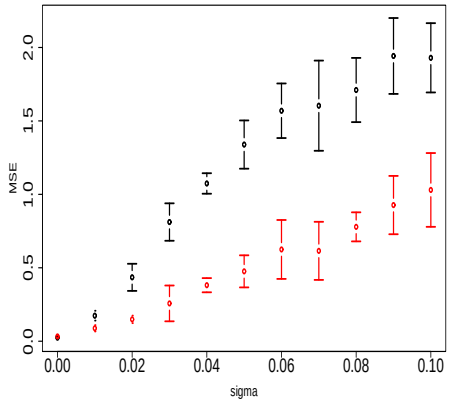

Fig. 3. Mean Square Error (mse) for PLS and penalized PLS. Boxplot of the mse for both methods as a function of the noise level σ. The upper curve (black) corresponds to PLS, the lower curve (red) corresponds to penalized PLS.

performance of PLS in the presence of noise. While penalization does not lead to better results in the noise-free scenario (which reproduces the findings in [10]), its increase in predictive performance becomes larger for higher values of σ.

5.2 Application: Derivatives of Spectra

Instead of predicting the amount of fat based on the spectrum itself, it is also common to consider (discrete approximations of the) derivatives of the spectrum. In Figure 4, we plot the spectrum and its first and second derivative for one of the 70 observations. While the spectrum itself is smooth, the approximate derivatives are not, and typically, smoothing techniques are needed to compensate. We now show that in the case of non-smooth spectra, penalized PLS outperforms PLS.

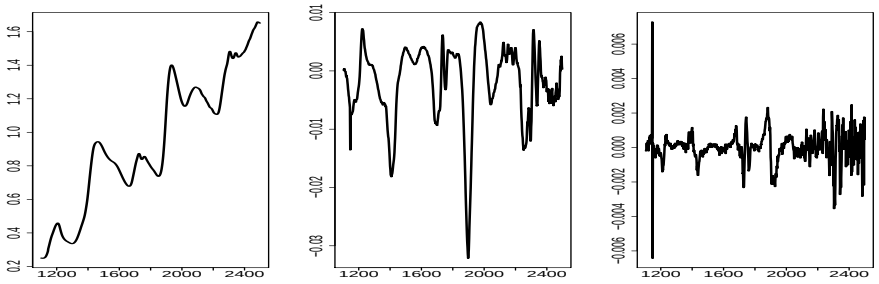

Fig. 4. NIR spectrum of biscuit dough and its derivatives. Left: Original spectrum. Center: First derivative. Right: Second derivative.

First, we derive two data sets from the original data $\boldsymbol{X}$ by computing the discretized first and second derivative using the difference operator (5), i.e. we transform $\boldsymbol{X}$ via $\boldsymbol{X}' = \left(\boldsymbol{D}_{700}\boldsymbol{X}^\top\right)^\top$ and $\boldsymbol{X}'' = \left(\boldsymbol{D}_{699}(\boldsymbol{X}')^\top\right)^\top$ respectively. We then compare PLS and penalized PLS on these three data sets. We randomly split the whole data set into a training set of size 39 and a test set of size 31. On the training set, we derive the optimal model parameters for PLS and penalized PLS via 10fold cross-validation. We then measure the performance of the two methods on the test set. This procedure is repeated 30 times. Table 1 displays the mean test error and their standard deviations. We conduct a Wilcoxon rank sum test to test the alternative hypothesis that the test error of penalized PLS is lower than the test error of PLS. The p-values can also be found in Table 1.

Table 1
Test error for the biscuit dough data set.

	original data	*1st derivative*	*2nd derivative*
PLS	0.181 ± 0.073	0.349 ± 0.103	3.319 ± 0.803
penalized PLS	0.208 ± 0.126	$\mathbf{0.161 \pm 0.041}$	$\mathbf{0.243 \pm 0.077}$
p-value	0.5484	3.685e-09	7.254e-12

The lowest test error is achieved on the first derivative of the data, i.e. in this example, the linear transformation $\boldsymbol{X} \to \boldsymbol{X}'$ indeed improves the performance. More importantly, penalized PLS leads to a significantly lower test set error compared to PLS on the two data sets $\boldsymbol{X}'$ and $\boldsymbol{X}''$ that correspond to non smooth spectra.

5.3 Application to additive models with spline transformations: Orange Juice Data

In the remainder of this section, we present a quantitative and qualitative analysis of the orange juice data that is discussed in [7]. The data consist of

24 samples of orange juice and 10 input variables that describe the mineralogical properties of the juices. These are the conductivity CON, the eight mineralogical characters $SiO_2, Na, K, Ca, Mg, Cl, SO_4, HCo_3$, and the SUM of the eight characters [7]. The information on the response variable is hidden due to confidentiality.

Table 2 displays the correlation matrix of the 10 predictors. In Figure 5, the

Table 2
Correlation matrix for the orange juice data set

	SiO2	Na	K	Ca	Mg	Cl	SO4	HCO3	Sum
CON	-0.10	0.04	0.10	0.98	0.97	-0.04	0.96	0.24	0.96
SiO2		0.26	0.84	-0.14	-0.11	0.07	-0.12	0.06	-0.07
Na			0.03	-0.11	-0.05	0.74	-0.08	0.09	-0.01
K				0.08	0.11	-0.08	0.11	-0.09	0.09
Ca					0.95	-0.16	0.99	0.20	0.97
Mg						-0.11	0.93	0.23	0.93
Cl							-0.14	0.13	-0.07
SO4								0.15	0.96
HCO3									0.41

10 predictor variables are plotted versus the response variable. The corre-

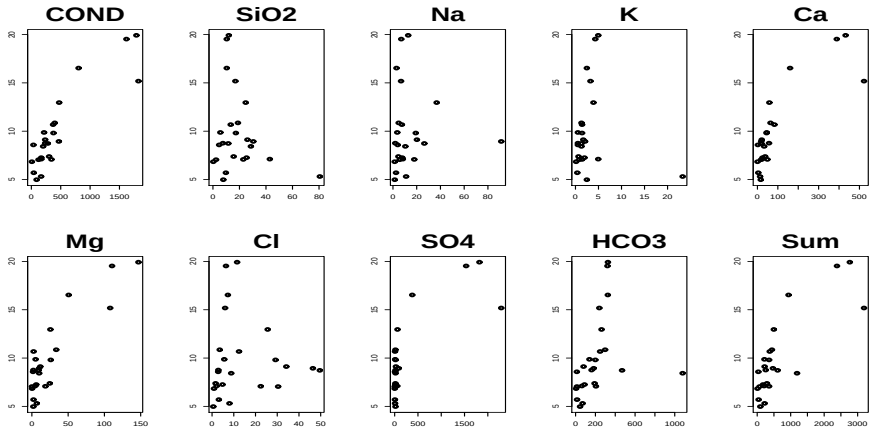

Fig. 5. Scatter plot of the orange juice data. The x-axis corresponds to the respective predictor variable, the y-axis corresponds to the response variable.

lation matrix in Table 2 reveals that some of the input variables are highly redundant, with correlation ≥ 0.95. Furthermore, [7] conclude from visual and quantitative analysis that the data contain nonlinear structure.

We fit several regression models to the data set. We apply two linear regression methods, (a) PLS and (b) Lasso [41]. As nonlinear methods, we compare (c) penalized PLS on B-spline transformations with variable selection (as described in Section 4), and (d) mgcv, an automatic estimation of the smoothing parameters for each variable [44,45] that is described in Section 1. The latter is the gold standard for fitting generalized additive models. For penalized PLS, we choose a range of possible λ's from 500 to 5000, which ensures that

the first component is very smooth and is in fact very close to a linear fit. Recall that the number of components influences the number of variables in the model, so a-priori, we should allow a generous number of components. For this data, we choose $m = 1, \ldots, 30$. For all methods, we compute the nested leave-one-out error. We compute the median over all 24 absolute test errors $\pm$ their median absolute deviation. The results are displayed in 3. In addition, we use a Wilcoxon rank sum test to test the alternative hypothesis that the absolute test errors of penalized PLS are lower. The p-values are given in the last column of the table. The nonlinear methods outperform the linear

Table 3
Test error for the juice data set.

	PLS	Lasso	penalized PLS	mgcv
absolute error	1.613 ± 1.780	1.657 ± 1.334	0.890 ± 0.961	1.087 ± 1.200
p-value	5.379×10^{-2}	1.574×10^{-2}	–	3.317×10^{-1}

methods PLS and Lasso, which confirms that the data exhibits a nonlinear structure. Penalized PLS with variable selection is on par with mgcv.
Next, we estimate the optimal penalized PLS model on the whole data set with leave-one-out cross-validation. Figure 6 displays the fitted additive components for each variable. The plot reveals a linear relationship in most of the

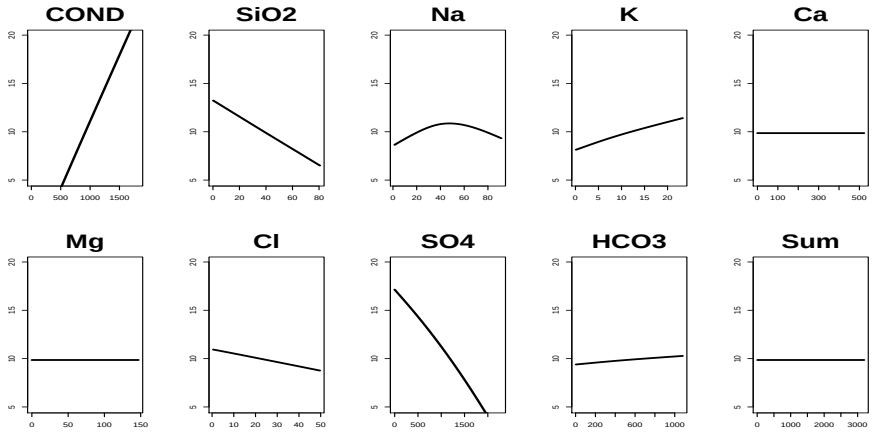

Fig. 6. Additive model fitted by penalized PLS. Each figure displays the nonlinear function $f_j(X_j)$ of the additive model (1).

components, except for Na. Note that the obtained model is sparse, as the 3 predictors Ca, Mg, Sum do not enter the model. Recall that Ca and Sum are highly correlated to $Cond$, and therefore might not carry much additional information.

6 Conclusion

In this work, we propose an extension of Partial Least Squares Regression using penalization techniques. Apart from its computational efficiency (it is virtually as fast as PLS), it also shares a lot of mathematical properties of PLS. Furthermore, a representation in terms of kernel matrices provides an intuitive geometric interpretation of the penalty term. Experiments on simulated and real world data show that penalized PLS is particularly successful for non-smooth and noisy observations.
Partial Least Squares iteratively minimizes a quadratic loss function, which is most suited for regression problems. Recent extensions of PLS that allow arbitrary convex loss functions [24] further increase its applicability. An extension of this framework that incorporates penalization is a promising research direction.
Penalty terms that control the roughness of the estimated functions are discussed in this paper. While these are the most prominent types of applications, our method might be used in a semi-supervised framework [5] as well. In a semi-supervised setting, a penalty term based on the graph Laplacian of the unlabeled data can be used to improve the prediction performance. Adding this type of penalty to PLS then leads to semi-supervised dimensionality reduction.
We highlighted the close connection between penalized PLS and preconditioned conjugate gradients (cg) . While cg and Krylov methods are commonly used in numerical linear algebra, their benefits for data analysis have not yet been exploited sufficiently. Only recently [13,15], they are utilized explicitly in a statistical framework. We strongly believe [1] that the interplay between numerical linear algebra and statistics will stimulate the analysis of data further.

Acknowledgement

This research was supported by the Deutsche Forschungsgemeinschaft (SFB 386, "Statistical Analysis of Discrete Structures"), and by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778.

A Proofs

We recall that for $k < i$

$$\boldsymbol{X}_i = \left(\boldsymbol{I}_n - \mathcal{P}_{\boldsymbol{t}_1,\ldots,\boldsymbol{t}_{i-1}}\right)\boldsymbol{X} = \left(\boldsymbol{I}_n - \mathcal{P}_{\boldsymbol{t}_k,\ldots,\boldsymbol{t}_{i-1}}\right)\boldsymbol{X}_k\,. \tag{A.1}$$

Proof of Proposition 1

We first proof a weaker version of 1, namely that the matrix $\boldsymbol{R}$ is upper triangular. We emphasize that this result does not depend on the form of the weight vectors. In particular, the result holds for the sparse penalized PLS algorithm introduced in Section 4 as well.
First note that (A.1) is equivalent to $\boldsymbol{X} = \boldsymbol{X}_j + \mathcal{P}_{\boldsymbol{t}_1,\ldots,\boldsymbol{t}_{j-1}}\boldsymbol{X}$. It follows that

$$\boldsymbol{X}\boldsymbol{w}_j = \boldsymbol{X}_j\boldsymbol{w}_j + \mathcal{P}_{\boldsymbol{t}_1,\ldots,\boldsymbol{t}_{j-1}}\boldsymbol{X}\ \boldsymbol{w}_j = \boldsymbol{t}_j + \sum_{i=1}^{j-1}\frac{\boldsymbol{t}_i^\top\boldsymbol{X}\boldsymbol{w}_j}{\boldsymbol{t}_i^\top\boldsymbol{t}_i}\boldsymbol{t}_i\,, \tag{A.2}$$

which proves (12). As all components $\boldsymbol{t}_i$ are mutually orthogonal, $\boldsymbol{t}_i^\top\boldsymbol{X}\boldsymbol{w}_j = 0$ for $i > j$ and $\boldsymbol{t}_i^\top\boldsymbol{X}\boldsymbol{w}_i = \boldsymbol{t}_i^\top\boldsymbol{t}_i \neq 0$. We conclude that $\boldsymbol{R}$ is an upper triangular matrix with all diagonal elements $\neq 0$. Plugging (12) into the formula for the projection operator, and using the orthonormality of the columns of $\boldsymbol{T}\widetilde{\boldsymbol{D}}$, we have

$$\widehat{\boldsymbol{y}}_m = \left(\boldsymbol{T}\widetilde{\boldsymbol{D}}\right)\left(\boldsymbol{T}\widetilde{\boldsymbol{D}}\right)^\top\boldsymbol{y} = \boldsymbol{X}\boldsymbol{W}\left(\widetilde{\boldsymbol{D}}\boldsymbol{R}\right)^{-1}\widetilde{\boldsymbol{D}}^\top\boldsymbol{T}^\top\boldsymbol{y} = \boldsymbol{X}\boldsymbol{W}\boldsymbol{R}^{-1}\boldsymbol{T}^\top\boldsymbol{y}\,.$$

This proves (13). To show the bidiagonality of $\boldsymbol{R}$, we note that the condition $i > j$ implies $\boldsymbol{X}_i\boldsymbol{w}_j = \boldsymbol{X}_j\boldsymbol{w}_j - \mathcal{P}_{\boldsymbol{t}_1,\ldots,t_{i-1}}\boldsymbol{X}_j\boldsymbol{w}_j = \boldsymbol{t}_j - \boldsymbol{t}_j = 0$. From this we can conclude directly that the weight vectors of penalized PLS are mutually $\boldsymbol{M}^{-1}$-orthogonal. This follows as for $i > j$

$$\langle\boldsymbol{w}_i,\boldsymbol{w}_j\rangle_{\boldsymbol{M}^{-1}} = \left\langle\boldsymbol{M}\boldsymbol{X}_i^\top\boldsymbol{y},\boldsymbol{w}_j\right\rangle_{\boldsymbol{M}^{-1}} = \boldsymbol{y}^\top\boldsymbol{X}_i\boldsymbol{w}_j = \boldsymbol{y}^\top\boldsymbol{0} = 0\,. \tag{A.3}$$

It follows from lemma 2 and the fact that $\boldsymbol{T}$ and $\boldsymbol{X}\boldsymbol{W}$ span the same space that $\boldsymbol{t}_i \in \boldsymbol{X}\mathcal{K}_m$. We can conclude that

$$\boldsymbol{M}\boldsymbol{X}^\top\boldsymbol{t}_i \in \boldsymbol{M}\boldsymbol{X}^\top\boldsymbol{X}\mathcal{K}_i = \boldsymbol{A}_{\boldsymbol{M}}\mathcal{K}_i \subset \mathcal{K}_{i+1} = \operatorname{span}\{\boldsymbol{w}_1,\ldots,\boldsymbol{w}_{i+1}\}\ .$$

In particular,

$$\boldsymbol{M}\boldsymbol{X}^\top\boldsymbol{t}_i = \sum_{k=1}^{i+1}\alpha_k\boldsymbol{w}_k\,. \tag{A.4}$$

Now recall (A.3). We conclude that for $j > i+1$

$$\boldsymbol{t}_i^\top\boldsymbol{X}\boldsymbol{w}_j = \langle\boldsymbol{M}\boldsymbol{X}^\top\boldsymbol{t}_i,\boldsymbol{w}_j\rangle_{\boldsymbol{M}^{-1}} \overset{(A.4)}{=} \langle\sum_{k=1}^{i+1}\alpha_k\boldsymbol{w}_k,\boldsymbol{w}_j\rangle_{\boldsymbol{M}^{-1}} \overset{(A.3)}{=} \sum_{k=1}^{i+1}\alpha_k 0 = 0\,. \qquad \square$$

Proof of Theorem 3

First note that it can be shown via induction that span $\{\boldsymbol{d}_0, \ldots, \boldsymbol{d}_{m-1}\} = \mathcal{K}_m$. It follows from the iterative definition of $\boldsymbol{\beta}_m$ that

$$\boldsymbol{\beta}_m = \sum_{i=0}^{m-1} \frac{\langle \boldsymbol{d}_i, \boldsymbol{b}_{\boldsymbol{M}} \rangle_{\boldsymbol{M}^{-1}}}{\langle \boldsymbol{d}_i, \boldsymbol{A}_{\boldsymbol{M}} \boldsymbol{d}_i \rangle_{\boldsymbol{M}^{-1}}} \boldsymbol{d}_i \,. \tag{A.5}$$

Hence, it suffices to show that $\langle \boldsymbol{d}_i, \boldsymbol{r}_i \rangle_{\boldsymbol{M}^{-1}} = \langle \boldsymbol{d}_i, \boldsymbol{b}_{\boldsymbol{M}} \rangle_{\boldsymbol{M}^{-1}}$. Note that

$$\boldsymbol{r}_i = \boldsymbol{b}_{\boldsymbol{M}} - \sum_{j=0}^{i-1} a_j \boldsymbol{A}_{\boldsymbol{M}} \boldsymbol{d}_j \,.$$

As $\boldsymbol{d}_i$ is $\boldsymbol{A}_{\boldsymbol{M}}$-orthogonal to all directions $\boldsymbol{d}_j$, $j < i$, (A.5) holds. Now, as $\boldsymbol{T}$ and $\boldsymbol{XW}$ span the same space, we have

$$\widehat{\boldsymbol{y}}_m = \mathcal{P}_{\boldsymbol{XW}} \boldsymbol{y} = \boldsymbol{XW} \left(\boldsymbol{W}^\top \boldsymbol{X}^\top \boldsymbol{XW} \right)^{-1} \boldsymbol{W}^\top \boldsymbol{X}^\top \boldsymbol{y} = \boldsymbol{X} \widehat{\boldsymbol{\beta}}_m \,.$$

Finally, as the search directions $\boldsymbol{d}_i$ span the Krylov space $\mathcal{K}_m$, we can replace the matrix $\boldsymbol{W}$ in this equation by $\boldsymbol{D} = (\boldsymbol{d}_0, \ldots, \boldsymbol{d}_{m-1})$. As the search directions are $\boldsymbol{A}_{\boldsymbol{M}}$-orthogonal, we have

$$\begin{aligned}
\widehat{\boldsymbol{\beta}}_m &= \boldsymbol{D} \left(\boldsymbol{D}^\top \boldsymbol{AD} \right)^{-1} \boldsymbol{D}^\top \boldsymbol{b} \\
&= \boldsymbol{D} \left(\boldsymbol{D}^\top \boldsymbol{M}^{-1} \boldsymbol{A}_{\boldsymbol{M}} \boldsymbol{D} \right)^{-1} \boldsymbol{D}^\top \boldsymbol{M}^{-1} \boldsymbol{b}_{\boldsymbol{M}} \\
&= \sum_{i=0}^{m-1} \frac{\langle \boldsymbol{d}_i, \boldsymbol{b}_{\boldsymbol{M}} \rangle_{\boldsymbol{M}^{-1}}}{\langle \boldsymbol{d}_i, \boldsymbol{A}_{\boldsymbol{M}} \boldsymbol{d}_i \rangle_{\boldsymbol{M}^{-1}}} \boldsymbol{d}_i \,,
\end{aligned}$$

and this equals (A.5). □

References

[1] A.-L. Boulesteix, A. Kondylis, and N. Krämer. Comments on "Augmenting the Bootstrap to Analyze High-Dimensional Genomic Data". *TEST*, 17:31–35, 2008. invited discussion.

[2] A.-L. Boulesteix and K. Strimmer. Partial Least Squares: A Versatile Tool for the Analysis of High-Dimensional Genomic Data. *Briefings in Bioinformatics*, 8(1):32–44, 2007.

[3] P.J. Brown, T. Fearn, and M. Vannucci. Bayesian Wavelet Regression on Curves with Application to a Spectroscopic Calibration Problem. *Journal of the American Statistical Association*, 96:398–408, 2001.

[4] H. Cardot and F. Ferraty and P. Sarda. Spline Estimators for the Functional Linear Model. *Statistica Sinica*, 13:571–591, 2003.

[5] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-supervised Learning*. MIT Press, 2006.

[6] C. de Boor. *A Practical Guide to Splines*. Springer, 1978.

[7] J. F. Durand. Local Polynomial Additive Regression Through PLS and Splines: PLSS. *Chemometrics and Intelligent Laboratory Systems*, 58:235–246, 2001.

[8] J. F. Durand and R. Sabatier. Additive Splines for Partial Least Squares Regression. *Journal of the American Statistical Association*, 92:1546–1554, 1997.

[9] P. Eilers and B.D. Marx. Flexible Smoothing with B-Splines and Penalties. *Statistical Science*, 11:89–121, 1996.

[10] C. Goutis and T. Fearn. Partial Least Squares Regression on Smooth Factors. *Journal of the American Statistical Association*, 91:627–632, 1996.

[11] T.J. Hastie and R.J. Tibshirani. *Generalized Additive Models*. Chapman and Hall, 1990.

[12] M. Hestenes and E. Stiefel. Methods for Conjugate Gradients for Solving Linear Systems. *Journal of Research of the National Bureau of Standards*, 49:409–436, 1952.

[13] T. Ide and K. Tsuda. Change Point Detection Using Krylov Subspace Learning. In *Proceedings of the SIAM International Conference on Data Mining*, pp. 515 – 520, 2007.

[14] S. de Jong. SIMPLS: An Alternative Approach to Partial Least Squares Regression. *Chemometrics and Intelligent Laboratory Systems*, 18:251–263, 1993.

[15] A. Kondylis and J. Whittaker. Spectral Preconditioning of Krylov Spaces: Combining PLS and PC Regression. *Computational Statistics & Data Analysis*, 52(5):2588–2603, 2008.

[16] N. Krämer. Analysis of High-Dimensional Data with Partial Least Squares and Boosting. *PhD Thesis*, TU Berlin, 2006.

[17] N. Krämer and A.-L. Boulesteix. *ppls: Penalized Partial Least Squares*, 2008. R package version 1.01.

[18] N. Krämer and M.L. Braun. Kernelizing PLS, Degrees of Freedom, and Efficient Model Selection. In Z. Ghahramani, editor, *Proceedings of the 24th International Conference on Machine Learning*, pages 441 – 448, 2007.

[19] C. Lanczos. An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators. *Journal of Research of the National Bureau of Standards*, 45:225–280, 1950.

[20] R. Manne. Analysis of Two Partial-Least-Squares Algorithms for Multivariate Calibration. *Chemometrics and Intelligent Laboratory Systems*, 2:187–197, 1987.

[21] D. Marske. Biochemical Oxygen Demand Data Interpretation Using Sum of Squares Surface. *Master's thesis, University of Wisconsin-Madison*, 1967.

[22] B. Marx and P. Eilers. Generalized Linear Regression on Samples Signals and Curves: a P-Spline Approach. *Technometrics*, 41:1–13, 1999.

[23] H. Martens and T. Naes. *Multivariate Calibration*. Wiley, New York, 1989.

[24] M.Momma and K.P. Bennett. Constructing Orthogonal Latent Features for Arbitrary Loss. *Feature Extraction, Foundations and Applications*, Springer, 2006.

[25] K.R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An Introduction to Kernel-based Learning Algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, 2001.

[26] B. Nadler, R.R. Coifman The Prediction Error in CLS and PLS: the Importance of Feature Selection Prior to Multivariate Calibration. *Journal of Chemometrics*, 19 (2):107–118, 2005.

[27] B.G. Osborne, T. Fearn, A.R. Miller, and S. Douglas. Application of Near Infrared Reflectance Spectroscopy to Compositional Analysis of Biscuits and Biscuits Dough. *Journal of the Science of Food and Agriculture*, 35:99–105, 1994.

[28] A. Phatak and F. de Hoog. Exploiting the Connection between PLS, Lanczos, and Conjugate Gradients: Alternative Proofs of some Properties of PLS. *Journal of Chemometrics*, 16:361–367, 2003.

[29] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. `http://www.R-project.org`.

[30] J. O. Ramsay and B. W. Silverman. *Functional Data Analysis*. Springer, New York, 2nd edition, 2005.

[31] S. Rännar, F. Lindgren, P. Geladi, and S. Wold. A PLS Kernel Algorithm for Data Sets with many Variables and Fewer Objects, Part I: Theory and Applications. *Journal of Chemometrics*, 8:111–125, 1994.

[32] P. Reiss and R. Odgen. Functional Principal Component Regression and Functional Partial Least Squares. *Journal of the American Statistical Association*, 102:984–996, 2007.

[33] R. Rosipal and N. Krämer. Overview and Recent Advances in Partial Least Squares. In C. Saunders, M. Grobelnik, S. Gunn, and J. Shawe-Taylor, editors, *Subspace, Latent Structure and Feature Selection Techniques*, Lecture Notes in Computer Science, pages 34–51. Springer, 2006.

[34] R. Rosipal and L.J. Trejo. Kernel Partial Least Squares Regression in Reproducing Kernel Hilbert Spaces. *Journal of Machine Learning Research*, 2:97–123, 2001.

[35] R. Rosipal, L.J. Trejo, and B. Matthews. Kernel PLS-SVC for Linear and Nonlinear Classification. In T. Fawcett and N. Mishra, editors, *Proceedings of the Twentieth International Conference on Machine Learning*, pages 640–647, Washington, DC, 2003.

[36] W. Saeys, B. de Ketelaere, and P. Darius. Potential Applications of Functional Data Analysis in Chemometrics. *Journal of Chemometrics*, 22:335–344, 2008.

[37] H. Saigo, N. Krämer, and K. Tsuda. Partial Least Squares Regression for Graph Mining. *14th International Conference on Knowledge Discovery and Data Mining*, to appear, 2008.

[38] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10(5):1299–1319, 1998.

[39] B.W. Silverman. Smoothed Functional Principal Components Analysis by Choice of Norm. *The Annals of Statistics*, 24(1):1–24, 1996.

[40] C.H. Spiegelman, M.J. McShane, M.J. Goetz, M. Motamedi, Q.L. Yue, G.L Coté. Theoretical Justification of Wavelength Selection in PLS Calibration: Development of a New Algorithm. *Analytical Chemistry*, 70(1):35–44, 1998

[41] R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[42] H. Wold. Path Models with Latent Variables: The NIPALS Approach. In H.M. Blalock et al., editor, *Quantitative Sociology: International Perspectives on Mathematical and Statistical Model Building*, pages 307–357. Academic Press, 1975.

[43] S. Wold, H. Ruhe, H. Wold, and W.J. Dunn III. The Collinearity Problem in Linear Regression. The Partial Least Squares (PLS) Approach to Generalized Inverses. *SIAM Journal of Scientific and Statistical Computations*, 5:735–743, 1984.

[44] S. N. Wood. Modelling and Smoothing Parameter Estimation with Multiple Quadratic Penalties. *Journal of the Royal Statistical Society B*, 62(2):413–428, 2000.

[45] S.N. Wood. Stable and Efficient Multiple Smoothing Parameter Estimation for Generalized Additive Models. *Journal of the American Statistical Association*, 99(467):673–687, 2004.