

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

Nicole Krämer* Anne-Laure Boulesteix †‡ Gerhard Tutz§

Abstract

We propose a novel framework that combines penalization with Partial Least Squares (PLS). Starting with a generalized additive model, we expand each additive component in terms of a generous amount of B-Splines basis functions. In order to prevent overfitting, we estimate the model by applying a penalized version of PLS. This new method can be computed virtually as fast as PLS. Furthermore, we prove a close connection of penalized PLS to preconditioned linear systems. The proposed approach is very general and can be applied to other problems. In particular, we show its benefit for noisy functional data.

Keywords: generalized additive model, dimension reduction, nonlinear regression, conjugate gradient, Krylov spaces

AMS classification: 62J02 62G05 65F10

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 dimension 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 dimension reduction technique Partial Least Squares with a penalization framework. Our motivation stems from two important applications, namely the estimation of generalized additive models and the regularization of functional data.

Nonlinear regression effects may be modeled via additive models of the form

$$Y = \beta_0 + f_1(X_1) + \dots + f_p(X_p) + \varepsilon. \quad (1)$$

where the functions f_1, \dots, f_p have unspecified functional form. An approach which allows flexible representation of the functions f_1, \dots, f_p is the expansion in basis functions (Hastie and Tibshirani, 1990). 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,

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

The basis functions B_{kj} are chosen adaptively by a selection procedure. The second approach (that is outlined in Section 3) circumvents the problem of basis function selection. Instead, we

*Department of Electrical Engineering and Computer Science, TU Berlin, Franklinstr. 28/29, 10587 Berlin, Germany, tel: +49 30 314 78627, fax: +49 30 314 78622, nkraemer@cs.tu-berlin.de

†Sylvia Lawry Centre for MS Research, Hohenlindenerstr. 1, 81677 Munich, Germany

‡Department of Medical Statistics and Epidemiology, TU Munich, Ismaningerstr. 22, 81675 Munich, Germany

§Department of Statistics, University of Munich, Akademiestr. 1, 80799 Munich, Germany

allow a generous amount $K_j \gg 1$ of basis functions in the expansion (2). As this usually leads to high-dimensional and highly correlated data, we penalize the coefficients β_{jk} in the estimation process (Eilers and Marx, 1996).

Quite generally, a different approach to deal with high dimensionality is to use dimension reduction techniques such as Partial Least Squares (PLS) (Wold, 1975; Wold et al., 1984). The main idea is to build a few components from the predictor variables and to regress \mathbf{y} onto these components. A short overview on PLS can be found in Section 2. As a linear approach, PLS probably fails to yield high prediction accuracy in the case of nonlinear relationships between predictors and responses as in (1). In order to incorporate nonlinear structures, it might be advisable to transform the original predictors preliminarily to a PLS regression. This approach has been proposed in two different variants. The first method (Durand and Sabatier, 1997) is based on a variant of PLS that is computed via an iterative algorithm. This approach incorporates splines transformations of the predictors within each iteration of the iterative algorithm. In contrast, the method proposed by Durand (2001) is global. The predictors are first transformed using splines 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. In the present article, we suggest an alternative approach based on the penalty strategy of Eilers and Marx (1996). As described in Section 3, we transform the initial data matrix nonlinearly using B-splines basis functions. Our new method, which we call penalized PLS, is based on the following principle. The equivalent of penalizing the (higher order) differences of adjacent B-splines coefficients is, in the framework of dimension reduction, the penalization of (higher order) differences of adjacent weights.

In Section 4, we introduce an adaptation of the principle of penalization to PLS. Although the motivation stems from its use for B-splines transformed data, the proposed approach is very general and can be adapted to other penalty terms or to other dimension reduction techniques such as Principal Components Analysis. It turns out that the new method shares a lot of properties of PLS and that its computation requires virtually no extra costs. 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 (Hestenes and Stiefel, 1952) 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. Furthermore, we show that this new technique is closely related to the so-called kernel trick. We prove that penalized PLS is equivalent to ordinary PLS using a generalized inner product that is defined by the penalty term. In Section 5, we conduct a simulation study to establish an efficient model selection strategy, and we compare PLS with the method proposed by Durand (2001).

Next, we address regression problems for functional data (Ramsay and Silverman, 1997). We speak of functional data if the variables that we observe are (discrete observations of) curves. In this setting, it is often beneficial to regularize the estimation process by imposing smoothness conditions, e.g. by penalizing the curvature of the functions. In Section 6, we apply penalized Partial Least Squares to functional data that is derived from near infrared spectroscopy. In particular, we illustrate that in the case of noisy observations, penalized PLS leads to a lower prediction error compared to PLS.

In the rest of the paper, we restrict ourselves to a univariate response. In Section 7, we stress that the extension of our method to a multivariate response is straightforward.

2 Partial Least Squares Regression

Let us consider the general linear regression problem. We want to predict a univariate response variable Y using p predictor variables X_1, \dots, X_p based on a finite set

$$\{(y_i, \mathbf{x}_i) = (y_i, x_{i1}, \dots, x_{ip}), i = 1, \dots, n\}$$

of observations. We set $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^T \in \mathbb{R}^{n \times p}$ and $\mathbf{y} = (y_1, \dots, y_n) \in \mathbb{R}^n$. For simplicity of notation we require that both \mathbf{X} and \mathbf{y} are centered. If we assume that the relationship between predictors and response is linear, this relationship can be represented in compact form by

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}. \quad (3)$$

Here, $\boldsymbol{\beta}$ is the p -dimensional vector of regression coefficients and $\boldsymbol{\epsilon}$ is the vector of residuals. When $n < p$, the usual regression tools such as ordinary least squares (OLS) regression cannot be applied to estimate $\boldsymbol{\beta}$ since the $p \times p$ covariance matrix $(1/n)\mathbf{X}^T\mathbf{X}$ (which has rank at most $n - 1$) is singular. From a technical point of view, this may be solved by replacing the inverse of the covariance matrix by a generalized inverse. However, for $n < p$, OLS usually fits the training data perfectly and one cannot expect the method to perform well on a new data set. Partial Least Squares (PLS) (Wold, 1975; Wold et al., 1984) 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 for analyzing chemical data (Martens and Naes, 1989), and in recent years, the success of PLS has led to applications in other scientific fields such as physiology (Rosipal et al., 2003) or bioinformatics (Boulesteix and Strimmer, 2007).

The main idea of PLS is to build orthogonal components $\mathbf{t}_1, \dots, \mathbf{t}_m$ from the original data \mathbf{X} and to use them as predictors in a least squares regression. 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 and we focus on two of them. An overview on different forms of PLS can be found in Rosipal and Krämer (2006). A component is a linear combination of the original predictors that hopefully reflects the relevant structure of the data. PLS is similar to Principal Components Regression (PCR). The difference is that PCR extracts components that explain the variance in the predictor variables whereas PLS extracts components that have a large covariance with \mathbf{y} . We now formalize this concept. A latent component \mathbf{t} is a linear combination $\mathbf{t} = \mathbf{X}\mathbf{w}$ of the predictor variables. The vector \mathbf{w} is usually called the weight vector. We want to find a component with maximal covariance to \mathbf{y} , that is, for the first component $\mathbf{t}_1 = \mathbf{X}\mathbf{w}_1$ we maximize the empirical squared covariance

$$\mathbf{w}_1 = \arg \max_{\mathbf{w}} \frac{\text{cov}^2(\mathbf{X}\mathbf{w}, \mathbf{y})}{\mathbf{w}^T\mathbf{w}} = \arg \max_{\mathbf{w}} \frac{\mathbf{w}^T\mathbf{X}^T\mathbf{y}\mathbf{y}^T\mathbf{X}\mathbf{w}}{\mathbf{w}^T\mathbf{w}}. \quad (4)$$

The solution of (4) is unique up to a scalar and equals $\mathbf{w}_1 = \mathbf{X}^T\mathbf{y}$. The normalization of the weight vectors \mathbf{w}_i is not essential for the PLS algorithm and PLS algorithms differ in the way they scale the weight vectors and components. In this paper, we present all algorithms without the scaling of the vectors, in order to keep the notation as simple as possible.

Subsequent components $\mathbf{t}_2, \mathbf{t}_3, \dots$ are chosen such that they maximize (4) and that all components \mathbf{t}_i are mutually orthogonal. In PLS, there are different techniques to extract subsequent components, and each technique gives rise to a variant of PLS. We briefly introduce two of them. In the method called SIMPLS (de Jong, 1993), one computes for the i th component,

$$\arg \max_{\mathbf{X}\mathbf{w} \perp \mathbf{t}_j, j < i} \frac{\mathbf{w}^T\mathbf{X}^T\mathbf{y}\mathbf{y}^T\mathbf{X}\mathbf{w}}{\mathbf{w}^T\mathbf{w}}.$$

Alternatively, one can deflate the original predictor variables \mathbf{X} . That is, we only consider the part of \mathbf{X} that is orthogonal onto all components $\mathbf{t}_j, j < i$. For any matrix \mathbf{V} , let us denote by

$\mathcal{P}_{\mathbf{V}}$ the orthogonal projection onto the space that is spanned by the columns of \mathbf{V} . In matrix notation, we have $\mathcal{P}_{\mathbf{V}} = \mathbf{V}(\mathbf{V}^T\mathbf{V})^+ \mathbf{V}^T$. The deflation of \mathbf{X} with respect to the components $\mathbf{t}_1, \dots, \mathbf{t}_{i-1}$ is defined as

$$\mathbf{X}_i = \mathbf{X} - \mathcal{P}_{\mathbf{t}_1, \dots, \mathbf{t}_{i-1}} \mathbf{X} = \mathbf{X}_{i-1} - \mathcal{P}_{\mathbf{t}_{i-1}} \mathbf{X}_{i-1}. \quad (5)$$

For the computation of the i th component, \mathbf{X} is replaced by \mathbf{X}_i in (4). This method is called the NIPALS algorithm (Wold, 1975). The two methods are equivalent if \mathbf{y} is univariate in the sense that we end up with the same components \mathbf{t}_i (de Jong, 1993). In this paper, we use the NIPALS algorithm, that is summarized in algorithm 1. With $\mathbf{T} = (\mathbf{t}_1, \dots, \mathbf{t}_m)$ denoting the collection of

Algorithm 1 NIPALS algorithm

Input: $\mathbf{X}_1 = \mathbf{X}, \mathbf{y}, m$
for $i=1, \dots, m$ **do**
 $\mathbf{w}_i = \mathbf{X}_i^T \mathbf{y}$ (weight vector)
 $\mathbf{t}_i = \mathbf{X}_i \mathbf{w}_i$ (component)
 $\mathbf{X}_{i+1} = \mathbf{X}_i - \mathcal{P}_{\mathbf{t}_i} \mathbf{X}_i$ (deflation)
end for

components, the fitted response is given by

$$\hat{\mathbf{y}}_m = \mathbf{T}(\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{y} = \mathcal{P}_{\mathbf{T}}\mathbf{y}. \quad (6)$$

In order to obtain the response for new observations, we have to determine the vector of regression coefficients $\hat{\mathbf{y}}_m = \mathbf{X}\hat{\boldsymbol{\beta}}_m$. This can be done efficiently by exploiting the bidiagonality of the matrix $\mathbf{T}^T\mathbf{X}$ ($\mathbf{w}_1, \dots, \mathbf{w}_m$) (Manne, 1987). In Section 4, we extend this result to penalized PLS and present an algorithm for the derivation of the regression coefficients of penalized PLS.

3 Penalized Regression Splines

In this section, we motivate the utilization of penalization techniques in the context of nonlinear regression problems. In Section 6, we discuss a related approach for functional data.

The fitting of generalized additive models by use of penalized regression splines has become a widely used tool in statistics. Starting with the seminal paper by Eilers and Marx (1996), the approach has been extended and applied in various publications (Ruppert, 2002; Wood, 2000, 2006). The basic concept is to expand the additive component of each variable X_j in basis functions as in (2) and to estimate the coefficients by penalization techniques. As suggested in Eilers and Marx (1996), B-splines are used as basis functions yielding so-called P-splines (for penalized B-splines). Splines are one-dimensional piecewise polynomial functions. The points at which the pieces are connected are called knots or breakpoints. We say that a spline is of order d if all polynomials are of degree $\leq d$ and if the spline is $(d-1)$ times continuously differentiable at the breakpoints. A particular efficient set of basis functions are B-splines (de Boor, 1978). The number of basis functions depends on the order of the splines and the number of breakpoints. For a given variable X_j , we consider a set of corresponding B-splines basis functions B_{1j}, \dots, B_{Kj} . These basis functions define a nonlinear map $\Phi_j(x) = (B_{1j}(x), \dots, B_{Kj}(x))^T$. By performing such a transformation on each of the variables X_1, \dots, X_p , the observation vector \mathbf{x}_i turns into a vector

$$\mathbf{z}_i = (B_{11}(x_{i1}), \dots, B_{m1}(x_{i1}), \dots, B_{1p}(x_{ip}), \dots, B_{mp}(x_{ip}))^T = \Phi(\mathbf{x}_i) \quad (7)$$

of length pK . Here Φ is the function defined by the B-splines. The resulting data matrix obtained by the transformation of \mathbf{X} is denoted by \mathbf{Z} in the rest of the paper. Cubic splines (i.e. $d = 3$) are the most widely used splines.

The estimation of (1) is transformed into the estimation of the pK -dimensional vector that consists of the coefficients β_{jk} :

$$\boldsymbol{\beta}^T = (\beta_{11}, \dots, \beta_{K1}, \dots, \beta_{12}, \dots, \beta_{Kp}) = (\boldsymbol{\beta}_{(1)}^T, \dots, \boldsymbol{\beta}_{(p)}^T).$$

As explained above, the vector $\boldsymbol{\beta}$ determines a nonlinear, additive function

$$f(\mathbf{x}) = \beta_0 + \sum_{j=1}^p f_j(x_j) = \beta_0 + \sum_{j=1}^p \sum_{k=1}^K \beta_{kj} B_{kj}(x_j) = \beta_0 + \boldsymbol{\Phi}(\mathbf{x})^T \boldsymbol{\beta} = \beta_0 + \mathbf{z}^T \boldsymbol{\beta}.$$

As \mathbf{Z} is usually high-dimensional, the estimation of $\boldsymbol{\beta}$ by minimizing the squared error usually leads to overfitting. Following Eilers and Marx (1996), we use for each variable many basis functions, say $K_j \approx 20$, and estimate by penalization. The idea is to penalize the second derivative of the function f . Eilers and Marx (1996) show that the following difference penalty term is a good approximation of the penalty on the second derivative of f ,

$$P(\boldsymbol{\beta}) = \sum_{j=1}^p \sum_{k=3}^m \lambda_j (\Delta^2 \beta_{kj})^2.$$

These are also called the second-order differences of adjacent parameters. The difference operator $\Delta^2 \beta_{kj}$ has the form

$$\Delta^2 \beta_{kj} = (\beta_{kj} - \beta_{k-1,j}) - (\beta_{k-1,j} - \beta_{k-2,j}) = \beta_{kj} - 2\beta_{k-1,j} + \beta_{k-2,j}.$$

The coefficients $\lambda_j \geq 0$ control the amount of penalization. This penalty term can be expressed in terms of a penalty matrix \mathbf{P} . We denote by \mathbf{D}_K the $(K-1) \times K$ matrix

$$\mathbf{D}_K = \begin{pmatrix} 1 & -1 & . & . & . \\ . & 1 & -1 & . & . \\ . & . & . & . & . \\ . & . & . & 1 & -1 \end{pmatrix} \quad (8)$$

that defines the first order difference operator. Setting $\mathbf{K}_2 = (\mathbf{D}_{K-1} \mathbf{D}_K)^T \mathbf{D}_{K-1} \mathbf{D}_K$, we conclude that the penalty term equals

$$P(\boldsymbol{\beta}) = \sum_{j=1}^p \lambda_j \boldsymbol{\beta}_{(j)}^T \mathbf{K}_2 \boldsymbol{\beta}_{(j)} = \boldsymbol{\beta}^T (\boldsymbol{\Delta}_\lambda \otimes \mathbf{K}_2) \boldsymbol{\beta} = \boldsymbol{\beta}^T \mathbf{P} \boldsymbol{\beta}. \quad (9)$$

Here $\boldsymbol{\Delta}_\lambda$ is the $p \times p$ diagonal matrix containing $\lambda_1, \dots, \lambda_p$ 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. We simply replace \mathbf{K}_2 by $\mathbf{K}_q = (\mathbf{D}_{K-q+1} \dots \mathbf{D}_K)^T (\mathbf{D}_{K-q+1} \dots \mathbf{D}_K)$. Note furthermore that \mathbf{P} is a symmetric matrix that is positive semidefinite.

4 Penalized Partial Least Squares Regression

We now introduce a general framework to combine PLS with penalization terms. We remark that this is not limited to spline transformed variables or to the special shape of the penalty matrix \mathbf{P} that is defined in (9). For this reason, we present the new method in terms of the original data matrix \mathbf{X} and only demand that \mathbf{P} is a symmetric positive semidefinite matrix.

4.1 The Algorithm

We modify the optimization criterion (4) of PLS in the following way. The first component $\mathbf{t}_1 = \mathbf{X}\mathbf{w}_1$ is defined by the solution of the problem

$$\arg \max_{\mathbf{w}} \frac{\mathbf{w}^T \mathbf{X}^T \mathbf{y} \mathbf{y}^T \mathbf{X} \mathbf{w}}{\mathbf{w}^T \mathbf{w} + \mathbf{w}^T \mathbf{P} \mathbf{w}}. \quad (10)$$

We obtain $\mathbf{w}_1 = \mathbf{M} \mathbf{X}^T \mathbf{y}$ with $\mathbf{M} = (\mathbf{I}_p + \mathbf{P})^{-1}$. Subsequent weight vectors and components are computed by deflating \mathbf{X} as described in (5) and then maximizing (10) with \mathbf{X} replaced by \mathbf{X}_i . In particular, we can compute the weight vectors and components of penalized PLS by simply replacing $\mathbf{w}_i = \mathbf{X}_i^T \mathbf{y}$ by $\mathbf{w}_i = \mathbf{M} \mathbf{X}_i^T \mathbf{y}$ in algorithm 1. Below, we derive an efficient algorithm for the computation of the regression coefficients of penalized PLS. Alternatively, we can adapt the SIMPLS algorithm by maximizing (10) under the orthogonality constraints on the components \mathbf{t}_j . We explain at the end of the next subsection that these two methods are equivalent.

In Figure 1, we give a geometric intuition of penalized PLS. We consider a linear regression model (3) with two correlated predictors. This is achieved by drawing 60 observations \mathbf{x}_i from a multivariate normal distribution

$$\mathbf{X} = (X_1, X_2)^T \sim N \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & -0.8 \\ -0.8 & 1 \end{pmatrix} \right).$$

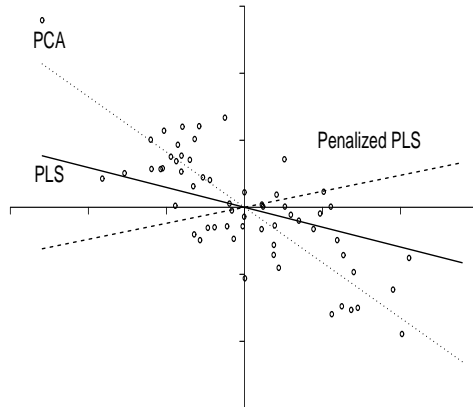


Figure 1: Comparison of the first latent components of PCA (dotted line), PLS (straight line) and penalized PLS (dashed line).

The true regression vector is $\boldsymbol{\beta} = (1, 0.5)^T$ and the variance of the noise term is set to 0.5. The observations $\mathbf{x}_1, \dots, \mathbf{x}_{60}$ are plotted in Figure 1. The first PCA component (dotted line) corresponds to the subspace of maximal variance in \mathbf{X} . As the PLS component (straight line) maximizes the covariance between \mathbf{X} and \mathbf{y} , it shifts the PCA components to directions that also explain the response \mathbf{y} . Now suppose that we want to penalize the difference $\beta_1 - \beta_2$ of the coefficients of the regression vector. This corresponds to the penalty matrix

$$\mathbf{P} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}.$$

As the penalty term $\boldsymbol{\beta}^t \mathbf{P} \boldsymbol{\beta}$ is zero for $\beta_1 = \beta_2$, the first penalized PLS component (dashed line) shifts the PLS component to the direction defined by $\beta_1 = \beta_2$. The higher the value of λ in the

penalty term $\lambda\beta^t\mathbf{P}\beta$, the closer is the penalized PLS component to this line.

We now present results on penalized PLS that allow us to compute its regression vectors efficiently. Note that all results on penalized PLS also hold for ordinary PLS if we choose $\mathbf{P} = \mathbf{0}$. Let

$$\mathbf{T} = (\mathbf{t}_1, \dots, \mathbf{t}_m) \quad \text{and} \quad \mathbf{W} = (\mathbf{w}_1, \dots, \mathbf{w}_m)$$

denote the matrices of components and weight vectors respectively.

Proposition 1. *The matrix*

$$(\mathbf{R}_m =) \mathbf{R} = \mathbf{T}^T \mathbf{X} \mathbf{W}$$

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

$$\mathbf{XW} = (\mathbf{T}\tilde{\mathbf{D}}) (\tilde{\mathbf{D}}\mathbf{R}). \quad (11)$$

In particular, the columns of \mathbf{T} and the columns of \mathbf{XW} span the same space.

This is an extension of a result for ordinary PLS that can be found e.g. in Manne (1987). Note that (11) is in fact the QR-decomposition of \mathbf{XW} .

Corollary 2. *The Penalized PLS regression vector obtained after m steps is*

$$\hat{\beta}_m = \mathbf{WR}^{-1}\mathbf{T}^T\mathbf{y}.$$

Proof. We deduce from (11) that $\mathbf{T}\tilde{\mathbf{D}} = (\mathbf{XW})\mathbf{R}^{-1}\tilde{\mathbf{D}}^{-1}$. Using the orthormality of the columns of $\mathbf{T}\tilde{\mathbf{D}}$, we have

$$\hat{\mathbf{y}}_m = \mathcal{P}_{\mathbf{T}\tilde{\mathbf{D}}}\mathbf{y} = (\mathbf{T}\tilde{\mathbf{D}}) (\mathbf{T}\tilde{\mathbf{D}})^T \mathbf{y} = \mathbf{XW} (\tilde{\mathbf{D}}\mathbf{R})^{-1} \tilde{\mathbf{D}}^T \mathbf{T}^T \mathbf{y} = \mathbf{XWR}^{-1}\mathbf{T}^T \mathbf{y}$$

which concludes the proof. □ □

This result is beneficial for two reasons. First, the inverse of \mathbf{R} can be computed very fast as the matrix is bidiagonal. Second, for all PLS components $i \leq m$ the inverse of \mathbf{R}_i is simply the submatrix of the inverse of \mathbf{R}_m that constitutes of the first i rows and columns. Combining this result with the PLS algorithm 1, we obtain the penalized PLS algorithm 2.

Algorithm 2 Penalized PLS algorithm

Input: $\mathbf{X}_1 = \mathbf{X}$, \mathbf{y} , m , \mathbf{P}

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

for $i=1, \dots, m$ **do**

$\mathbf{w}_i = \mathbf{M}\mathbf{X}_i^T\mathbf{y}$ (weight vector)

$\mathbf{t}_i = \mathbf{X}_i\mathbf{w}_i$ (component)

$\mathbf{X}_{i+1} = \mathbf{X}_i - \mathcal{P}_{\mathbf{t}_i}\mathbf{X}_i$ (deflation)

end for

$\mathbf{L} = (\mathbf{T}^T \mathbf{XW})^{-1}$ (inverse of \mathbf{R}_m)

for $i=1, \dots, m$ **do**

\mathbf{L}_i (first i rows and columns of \mathbf{L})

$\hat{\beta}_i = (\mathbf{w}_1, \dots, \mathbf{w}_i) \mathbf{L}_i (\mathbf{t}_1, \dots, \mathbf{t}_i)^T \mathbf{y}$ (regression vector)

end for

We now illustrate the influence of the penalty term and the number of components on a small example with only one predictor. The BOD (biochemical oxygen demand) data set (Marske, 1967) consists of six measurements. The predictor value is the time of measurement, the response variable is the biochemical oxygen demand. This data set is part of the R software (R

Development Core Team, 2005). We fix the number of knots to 25 and choose cubic splines, i.e. $d = 3$. More information on the choice of these parameters can be found in Section 5. In Figure 2 we plot the fitted functions obtained from penalized PLS for different number 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 values of λ (i.e. from

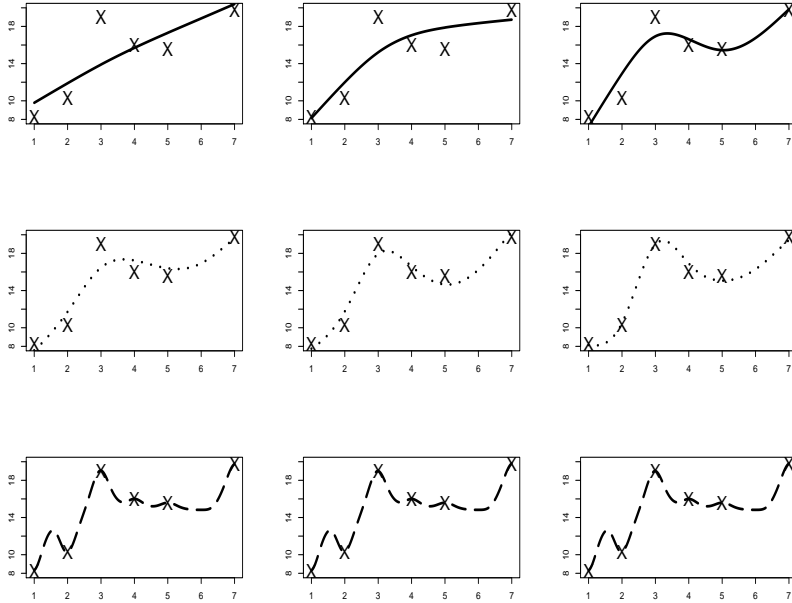


Figure 2: Penalized PLS for different numbers of components (from left to right: 1; 2; 3) and different values of the smoothing parameter λ (from top to bottom: 2000; 20; 0).

top to bottom) we see that the penalty term indeed controls the the curvature of the functions. 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.

4.2 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

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2. \quad (12)$$

This is equivalent to finding the solution of the associated normal equation

$$\mathbf{A}\boldsymbol{\beta} = \mathbf{b} \quad (13)$$

with $\mathbf{b} = \mathbf{X}^T \mathbf{y}$ and $\mathbf{A} = \mathbf{X}^T \mathbf{X}$. If the matrix \mathbf{A} is invertible, the solution of the normal equations is the OLS estimator $\hat{\boldsymbol{\beta}} = \mathbf{A}^{-1} \mathbf{b}$. If \mathbf{A} is singular, the solution of (13) with minimal Euclidean norm is $\mathbf{A}^+ \mathbf{b}$. We already mentioned in Section 2 that in the case of high dimensional data, the matrix \mathbf{A} is often (almost) singular and that the OLS estimator performs poorly on new data sets. A popular strategy is to regularize the least squares criterion (12) in the hope of improving the

performance of the estimator. This often corresponds to finding approximate solutions of (13). For example, Ridge Regression corresponds to the solution of the modified normal equations $(\mathbf{A} + \lambda \mathbf{I}_p) \boldsymbol{\beta} = \mathbf{b}$. Here $\lambda > 0$ is the Ridge parameter. Principal Components Regression uses the eigen decomposition of \mathbf{A} and approximates \mathbf{A}^+ and \mathbf{b} via the first m eigenvectors of \mathbf{A} .

It can be shown that the PLS estimators are equal to the approximate solutions of the conjugate gradient method (Hestenes and Stiefel, 1952). This is a procedure that iteratively computes approximate solutions of (13) by minimizing the quadratic function

$$\phi(\boldsymbol{\beta}) = \frac{1}{2} \boldsymbol{\beta}^T \mathbf{A} \boldsymbol{\beta} - \boldsymbol{\beta}^T \mathbf{b} = \frac{1}{2} \langle \boldsymbol{\beta}, \mathbf{A} \boldsymbol{\beta} \rangle - \langle \boldsymbol{\beta}, \mathbf{b} \rangle \quad (14)$$

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 (Lanczos, 1950). The latter is a method for approximating eigenvalues. The connection between PLS and these methods is well-elaborated in Phatak and de Hoog (2003).

We now establish a similar connection between penalized PLS and the above mentioned methods. Set $\mathbf{A}_M = \mathbf{M} \mathbf{A}$ and $\mathbf{b}_M = \mathbf{M} \mathbf{b}$. Recall that \mathbf{M} is a symmetric and positive definite matrix that is determined by the penalty term \mathbf{P} . We now illustrate that penalized PLS finds approximate solutions of the preconditioned normal equation

$$\mathbf{A}_M \boldsymbol{\beta} = \mathbf{b}_M. \quad (15)$$

Let us denote the space spanned by the sequence $\mathbf{b}_M, \mathbf{A}_M \mathbf{b}_M, \dots, \mathbf{A}_M^{m-1} \mathbf{b}_M$ as the Krylov space \mathcal{K}_m of \mathbf{A}_M and \mathbf{b}_M .

Lemma 3. *The space spanned by the weight vectors $\mathbf{w}_1, \dots, \mathbf{w}_m$ of penalized PLS equals \mathcal{K}_m .*

This is the generalization of a result for ordinary PLS. Note that it follows from lemma 3 and the fact that \mathbf{T} and $\mathbf{X} \mathbf{W}$ span the same space that the penalized PLS estimator is the solution of the optimization problem (12) with the constraint $\boldsymbol{\beta} \in \mathcal{K}_m$.

We now present the conjugate gradient method for the equation

$$\mathbf{A}_M \boldsymbol{\beta} = \mathbf{b}_M. \quad (16)$$

The Conjugate gradient method is normally applied if the involved matrix is symmetric. Note that in general, the matrix \mathbf{A}_M is not symmetric with respect to the canonical inner product, but with respect to the inner product $\langle \mathbf{x}, \tilde{\mathbf{x}} \rangle_{M^{-1}} = \mathbf{x}^T \mathbf{M}^{-1} \tilde{\mathbf{x}}$ defined by \mathbf{M}^{-1} . We can rewrite the quadratic function ϕ defined in (14) as

$$\phi(\boldsymbol{\beta}) = \frac{1}{2} \langle \boldsymbol{\beta}, \mathbf{A}_M \boldsymbol{\beta} \rangle_{M^{-1}} - \langle \boldsymbol{\beta}, \mathbf{b}_M \rangle_{M^{-1}}.$$

We replace the canonical inner product by the inner product defined by \mathbf{M}^{-1} and minimize this function iteratively along directions that are \mathbf{A}_M -orthogonal.

We start with an initial guess $\boldsymbol{\beta}_0 = \mathbf{0}$ and define $\mathbf{d}_0 = \mathbf{r}_0 = \mathbf{b}_M - \mathbf{A}_M \boldsymbol{\beta}_0 = \mathbf{b}_M$. The quantity \mathbf{d}_m is the search direction and \mathbf{r}_m is the residual. For a given direction \mathbf{d}_m , we have to determine the optimal step size, that is we have to find

$$a_m = \arg \min_a \phi(\boldsymbol{\beta}_m + a \mathbf{d}_m).$$

It is straightforward to check that

$$a_m = \frac{\langle \mathbf{d}_m, \mathbf{r}_m \rangle_{M^{-1}}}{\langle \mathbf{d}_m, \mathbf{A}_M \mathbf{d}_m \rangle_{M^{-1}}}.$$

The new approximate solution is then

$$\boldsymbol{\beta}_{m+1} = \boldsymbol{\beta}_m + a_m \mathbf{d}_m. \quad (17)$$

After updating the residuals via

$$\mathbf{r}_{m+1} = \mathbf{b}_M - \mathbf{A}_M \boldsymbol{\beta}_{m+1},$$

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

$$\mathbf{d}_{m+1} = \mathbf{r}_{m+1} - \sum_{i=0}^m \frac{\langle \mathbf{r}_{m+1}, \mathbf{A}_M \mathbf{d}_i \rangle_{M^{-1}}}{\langle \mathbf{d}_i, \mathbf{A}_M \mathbf{d}_i \rangle_{M^{-1}}} \mathbf{d}_i.$$

Theorem 4. *The penalized PLS algorithm is equal to the conjugate gradient algorithm for the preconditioned system (15), that is $\boldsymbol{\beta}_m$ defined in (refeq:cgbeta) 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 Phatak and de Hoog (2003).

We conclude this subsection by first remarking that the correspondence between penalized PLS and approximate solutions of the preconditioned equations (16) implies that after at most p iterations, the penalized PLS estimator equals $\mathbf{A}_M^+ \mathbf{b}_M$. We conclude that if the matrix \mathbf{A} is of full rank, the penalized PLS estimator equals the OLS estimator after at most p iterations.

Furthermore, it is straightforward to show (via induction) that the components that are defined by the penalized SIMPLS algorithm span $\mathbf{X}\mathcal{K}_m = \text{span}\{\mathbf{t}_1, \dots, \mathbf{t}_m\}$. The last equality follows from proposition 1 and lemma 3. Hence we conclude that that also the penalized versions of NIPALS and SIMPLS are equivalent.

4.3 Kernel Penalized Partial Least Squares

The computation of the penalized PLS estimator as presented in algorithm 2 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 \mathbf{y} .

To illustrate the concept of kernel based methods, let us consider the case of ordinary PLS on B-Splines transformed variables. Recall that in (7), we transform the original data \mathbf{X} using a nonlinear function Φ defined by the B-Splines. The key observation (Rännar et al., 1994; Rosipal and Trejo, 2001) is that the PLS algorithm can be represented in such a way that it only relies on inner products between observations and not on the observations themselves. This implies that we do not need to map the data points explicitly using the function Φ . It suffices to compute the function $k(\mathbf{x}, \tilde{\mathbf{x}}) = \langle \Phi(\mathbf{x}), \Phi(\tilde{\mathbf{x}}) \rangle$.

The function k is called a kernel, and the replacement of the usual inner product by a kernel is known as the kernel trick. Instead of defining a nonlinear map Φ , we define a valid kernel function $k(x, z)$. E.g., nonlinear relationships can be modeled via Gaussian kernels

$$k_d(\mathbf{x}, \tilde{\mathbf{x}}) = e^{-\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|^2}{\sigma^2}}, \sigma > 0.$$

Literature on the kernel trick and its applications is abundant. A detailed treatise of the subject can be found in Schölkopf and Smola (2002). A nonlinear version of PLS using the kernel trick is presented in Rosipal and Trejo (2001).

Let us return to penalized Partial Least Squares. We define the $n \times n$ matrix $\mathbf{K}_M = (\langle \mathbf{x}_i, \mathbf{x}_j \rangle_M) = \mathbf{X} \mathbf{M} \mathbf{X}^T$. The key is to find a representation $\widehat{\boldsymbol{\beta}}_m = \mathbf{M} \mathbf{X}^T \widehat{\boldsymbol{\alpha}}_m$ of the regression vector in terms of kernel coefficients $\widehat{\boldsymbol{\alpha}}_m$.

We start with the remark that $\mathbf{w}_i = \mathbf{M}\mathbf{X}^T\mathbf{u}_i$ with $\mathbf{u}_i = \mathbf{y} - \hat{\mathbf{y}}_i$ defined as the residuals in each step. Furthermore, it follows from the bidiagonality of \mathbf{R} that

$$\mathbf{t}_i = \mathbf{X}_i\mathbf{w}_i = (\mathbf{I}_n - \mathcal{P}_{t_{i-1}})\mathbf{X}\mathbf{w}_i = (\mathbf{I}_n - \mathcal{P}_{t_{i-1}})\mathbf{K}_M\mathbf{u}_i.$$

Finally, we have $\mathbf{R} = \mathbf{T}^T\mathbf{X}\mathbf{W} = \mathbf{T}^T\mathbf{K}_M\mathbf{U}$ with $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_m)$. We can now derive algorithm 3 for the kernel coefficients.

Algorithm 3 Kernel Penalized PLS algorithm

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

The kernel algorithm reveals that penalized PLS itself is closely connected to the kernel trick as well. More precisely, penalized PLS equals ordinary PLS with the canonical inner product replaced by the inner product $\langle \mathbf{x}, \mathbf{z} \rangle_M = \mathbf{x}^T\mathbf{M}\mathbf{z}$. This function is called a linear kernel. Why is this a sensible inner product? Let us consider the eigen decomposition of the penalty matrix, $\mathbf{P} = \mathbf{S}\boldsymbol{\Theta}\mathbf{S}^T$. We prefer direction \mathbf{s} such that $\mathbf{s}^T\mathbf{P}\mathbf{s}$ is small, that is we prefer directions that are defined by eigenvectors \mathbf{s}_i of \mathbf{P} with a small corresponding eigenvalue θ_i . If we represent the vectors $\mathbf{x} = \mathbf{S}\tilde{\mathbf{x}}$ and $\mathbf{z} = \mathbf{S}\tilde{\mathbf{z}}$ in terms of the eigenvectors of \mathbf{P} , we conclude that

$$\langle \mathbf{x}, \mathbf{z} \rangle_M = \tilde{\mathbf{x}}^T (\mathbf{I}_p + \boldsymbol{\Theta})^{-1} \tilde{\mathbf{z}} = \sum_{i=1}^p \frac{1}{1 + \theta_i} \tilde{\mathbf{x}}_i \tilde{\mathbf{z}}_i.$$

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

5 Model Selection and Performance

Our proposed method depends on several model parameters that have to be selected. If we consider the most general setting, there are three parameters for each variable (number of knots, degree of the splines and the smoothness penalty) and one global parameter (the number of PLS components). This yields $3p + 1$ model parameters. Optimizing this huge amount of parameters without any restriction is not only computationally infeasible, it for sure leads to overfitting. For this reason, we first reduce the complexity by imposing that the number of knots, the degree of smoothness and the smoothness parameters are the same for each variable. Still, this leaves us with four parameters to tune. In this section, we conduct simulation studies to explore the behavior of the different parameters. It turns out that it is sufficient to fix the degree of the splines and the numbers of knots and only optimize with respect to the number of components and the global penalty term.

The setting of the simulation study is as follows. The number of variables is $p = 20$, the number of training examples is $n = 50$. The input data \mathbf{X} is drawn from a multivariate uniform

distribution, i.e. $X_i \sim U[-1, 1]$. We consider the regression model

$$Y = \underbrace{\sum_{j=1}^5 f_j(X_j)}_{=F(X)} + \epsilon,$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is an error which is independent of X_1, \dots, X_p and $f_j(x) = a_j x + \sin(6b_j x)$. For every simulation round, we draw a_j and b_j from a uniform distribution $a_j, b_j \sim U[-1, 1]$. Note that the remaining $20 - 5 = 15$ variables X_6, \dots, X_{20} are irrelevant for prediction. The amount of noise is determined by setting the signal-to-noise ratio to $\text{var}(F(\mathbf{X}))/\sigma^2 = 16^2$. We now explore the behavior of penalized PLS if certain model parameters are fixed. We generate training data and compute the optimal penalized PLS model (i.e. the optimal combination of λ and m) using 10fold cross-validation. The performance of the chosen model is then assessed on a test set of size 200. This procedure is repeated 30 times.

We first fix the degrees of the splines to $d = 3$, which leads to the widely used cubic splines. We compare the performance of penalized PLS for different numbers of knots 10, 15, 20, 25, 30, 35, 40. The boxplot of the test errors are displayed in Figure 3. The test error is slightly higher if only a few knots are included in the model. For large number of knots, the curve of median test errors stays flat. In particular, allowing a generous amount of knots does not lead to overfitting. This behavior might be explained by the fact that the smoothness penalty λ compensates for the irregularity of the functions f_j that is introduced by the number of knots.

Taking into account these results, we now fix the number of knots to 25 and vary the degree of the splines by considering $d = 1, 2, 3, 4, 5, 6$. The boxplot of the test errors is displayed in Figure 3. Again, the test errors is slightly higher for low values of the model parameter and

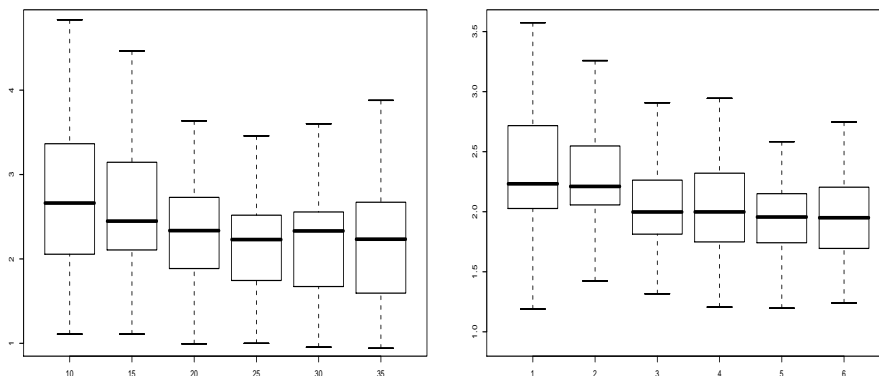


Figure 3: Test error of penalized PLS for different parameter settings. Left: Test error for different number of knots. Right: Test error for different degrees of the splines.

does not increase for higher values.

The experiments on the simulated data indicates that it is not necessary to vary all 4 model parameters. For this reason, we suggest to fix the number of knots to a high value, say 25 and fix the degrees of the splines to the default value of $d = 3$.

As indicated in the introduction, the approach by Durand (2001) also applies Partial Least Squares to B-splines transformed data. But instead of regularizing the PLS solution via penalization techniques, Durand (2001) vary the degrees of the splines and the numbers of knots. As there is no hint to a model selection strategy (and optimizing all of these parameters simultaneously is not feasible), we again impose that the degree and the number of knots is equal for all variables. This yields 3 model parameters, degree of the splines, number of knots and number of PLS components.

We now compare this approach to penalized PLS on the simulated data that is described above. The mean test errors and their standard deviations are 1.900 ± 0.614 for penalized PLS 2.152 ± 0.722 for Durand’s approach. Furthermore, we conduct a Wilcoxon rank sum test with the alternative hypothesis that the test error of penalized PLS is lower. The corresponding p -value is 0.083.

We remark that – apart from its lower test error in the simulation study – penalized PLS is also more efficient in terms of model selection. In Durand’s approach, the transformation (7) has to be computed for all possible values of d and K separately, whereas for penalized PLS, the transformation is only performed once.

6 Application to Functional Data

In this section, we extend the framework of penalized PLS to functional data (Ramsay and Silverman, 1997). In a nutshell, we speak of functional data if the variables that we observe are discrete observations of curves. Although a more general setting is possible, we focus on the case that only the predictor variables X_1, \dots, X_p are measurements of curves $x : T \rightarrow \mathbb{R}$ at different points $t_1 < \dots < t_p$ in the interval T . This implies that the observations \mathbf{x}_i are of the form

$$\mathbf{x}_i = (x_i(t_1), \dots, x_i(t_p))^T. \quad (18)$$

The corresponding linear regression model for functional data is given by

$$Y_i = \beta_0 + \int_T \beta(t)x_i(t)dt + \varepsilon_i$$

with $\beta : T \rightarrow \mathbb{R}$. We can transform this into a multivariate regression problem by estimating $\beta(t)$ at the discrete points t_1, \dots, t_p , i.e. we estimate $\boldsymbol{\beta} = (\beta(t_1), \dots, \beta(t_p))^T$. As this leads to a high-dimensional regression problem, the least squares criterion is regularized by imposing a roughness penalty on $\beta(t)$. Typically, the curvature of the function β is penalized. This penalty term can be approximated in terms of $\boldsymbol{\beta}$ by computing the second order differences of the coefficients. For equidistant points t_1, \dots, t_p , we yield

$$P(\beta) \approx \boldsymbol{\beta}^T (\mathbf{D}_{p-2}\mathbf{D}_{p-1})^T (\mathbf{D}_{p-2}\mathbf{D}_{p-1}) \boldsymbol{\beta} = \boldsymbol{\beta}^T \mathbf{P} \boldsymbol{\beta}.$$

The matrix \mathbf{D}_K is defined in (8). The extension to non-equidistant measurements or to differences of higher orders is straightforward. This penalization strategy is particularly beneficial if the observations (18) of the curves are noisy or if the discrete observations are not measured at equidistant points.

We now discuss the application of penalized PLS to functional data. In fact, a combination of PLS with penalty terms was first proposed in Goutis and Fearn (1996) for data derived from near infrared (NIR) spectroscopy. More precisely, they suggest to incorporate an additive penalty $\mathbf{w}^T \mathbf{X}^T \mathbf{y} \mathbf{y}^T \mathbf{X} \mathbf{w} - \mathbf{w}^T \mathbf{P} \mathbf{w}$. Here, \mathbf{P} penalizes the curvature of \mathbf{w} . The solution is the first eigenvector of the matrix $\mathbf{X}^T \mathbf{y} \mathbf{y}^T \mathbf{X} - \mathbf{P}$. Compared to our approach, this is computationally less efficient, moreover, it is – to our knowledge – not possible to apply the kernel trick. Goutis and Fearn (1996) report that in experiments, the incorporation of a penalty term does not increase the performance of PLS on spectral data. We conjecture that this originates from the fact that NIR data is typically measured at equidistant points and that the measurements are typically smooth. In this section, we illustrate that if these conditions do not hold, penalized PLS leads to considerably better predictions than PLS.

The following example is taken from Osborne et al. (1994). The data can be downloaded from <http://www.stat.tamu.edu/~mvannucci/webpages/codes.html>. The data consists 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. As the direct measurement of fat is costly and time-consuming, NIR (near

infra red) spectroscopy is used instead. For each of the $n = 39 + 31 = 70$ observations of biscuit dough, the amount of fat and the reflectance of NIR light for different wavelengths is measured. In this example, $p = 700$ equidistant wavelengths in the range from 1100 to 2398 nanometers are used. For each example, we obtain a function of the reflectance, which is called a spectrum. The task is to predict the amount of fat of a new sample after observing its spectrum. Instead of

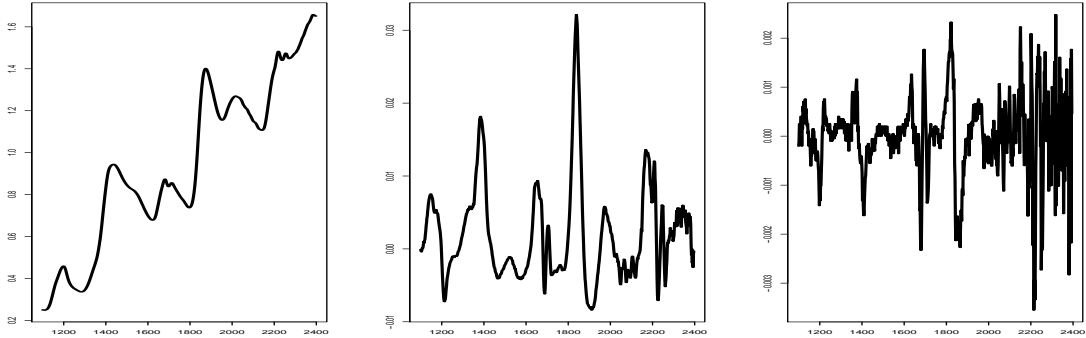


Figure 4: NIR spectrum of biscuit dough and its derivatives. Left: Original spectrum. Center: First derivative. Right: Second derivative.

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. We now show that in the case of non-smooth spectra, penalized PLS outperforms PLS.

First, we derive two data sets from the original data \mathbf{X} by computing the discretized first and second derivative using the difference operator (8), i.e. we transform \mathbf{X} via $\mathbf{X}' = (\mathbf{D}_{700}\mathbf{X}^T)^T$ and $\mathbf{X}'' = (\mathbf{D}_{699}(\mathbf{X}')^T)^T$ 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. Again, 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.

	<i>original data</i>	<i>1st derivative</i>	<i>2nd derivative</i>
<i>PLS</i>	0.181 ± 0.073	0.349 ± 0.103	3.319 ± 0.803
<i>penalized PLS</i>	0.208 ± 0.126	0.161 ± 0.041	0.243 ± 0.077
<i>p-value</i>	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 $\mathbf{X} \rightarrow \mathbf{X}'$ indeed improves the performance. More importantly, penalized PLS leads to a significantly lower test set compared to PLS on the two data sets \mathbf{X}' and \mathbf{X}'' that correspond to non smooth spectra.

7 Concluding Remarks

In this work, we proposed 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 data and real world data show that efficient model selection is possible and that penalized PLS is particularly successful for non smooth and noisy observations.

The introduction of a penalty term can easily be adapted to other dimension reduction techniques. For example for Principal Components Analysis, the penalized optimization criterion is

$$\max_{\mathbf{w}} \frac{\text{var}(\mathbf{X}\mathbf{w})}{\mathbf{w}^T\mathbf{w} + \mathbf{w}^T\mathbf{P}\mathbf{w}}.$$

PLS can handle multivariate responses \mathbf{Y} . The natural extension of criterion (4) is

$$\max_{\mathbf{w}} \frac{\|\text{cov}(\mathbf{X}\mathbf{w}, \mathbf{Y})\|^2}{\mathbf{w}^T\mathbf{w}} = \max_{\mathbf{w}} \frac{\mathbf{w}^T \mathbf{X}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X} \mathbf{w}}{\mathbf{w}^T \mathbf{w}}.$$

The solution is the eigenvector of the matrix $\mathbf{X}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X}$ that corresponds to its largest eigenvalue. If we want to apply penalized PLS for multivariate responses, we compute

$$\max_{\mathbf{w}} \frac{\mathbf{w}^T \mathbf{X}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X} \mathbf{w}}{\mathbf{w}^T \mathbf{w} + \mathbf{w}^T \mathbf{P} \mathbf{w}}.$$

The solution fulfills

$$\mathbf{X}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X} \mathbf{w} = \gamma (\mathbf{I}_p + \mathbf{P}) \mathbf{w}, \gamma \in \mathbb{R}.$$

This is called a generalized eigenvalue problem or a matrix pencil. Note that for multivariate \mathbf{Y} , the equivalence of SIMPLS and NIPALS does not hold, so we expect the penalized versions of these methods to be different as well. There are kernel versions for PLS with multivariate \mathbf{Y} (Rännar et al., 1994; Rosipal and Trejo, 2001), hence we can also represent multivariate penalized PLS in terms of kernel matrices.

Acknowledgement

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

References

- Boulesteix, A.-L. and K. Strimmer (2007). Partial Least Squares: A Versatile Tool for the Analysis of High-Dimensional Genomic Data. *Briefings in Bioinformatics* 8(1), 32–44.
- de Boor, C. (1978). *A Practical Guide to Splines*. Springer.
- de Jong, S. (1993). SIMPLS: An Alternative Approach to Partial Least Squares Regression. *Chemometrics and Intelligent Laboratory Systems* 18, 251 – 263.
- Durand, J. F. (2001). Local Polynomial Additive Regression Through PLS and Splines: PLSS. *Chemometrics and Intelligent Laboratory Systems* 58, 235–246.
- Durand, J. F. and R. Sabatier (1997). Additive Splines for Partial Least Squares Regression. *Journal of the American Statistical Association* 92, 1546–1554.

- Eilers, P. and B. Marx (1996). Flexible Smoothing with B-Splines and Penalties. *Statistical Science* 11, 89–121.
- Goutis, C. and T. Fearn (1996). Partial Least Squares Regression on Smooth Factors. *Journal of the American Statistical Association* 91, 627–632.
- Hastie, T. and R. Tibshirani (1990). *Generalized Additive Models*. Chapman and Hall.
- Hestenes, M. and E. Stiefel (1952). Methods for Conjugate Gradients for Solving Linear Systems. *Journal of Research of the National Bureau of Standards* 49, 409–436.
- Lanczos, C. (1950). 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.
- Manne, R. (1987). Analysis of Two Partial-Least-Squares Algorithms for Multivariate Calibration. *Chemometrics and Intelligent Laboratory Systems* 2, 187–197.
- Marske, D. (1967). Biochemical oxygen demand data interpretation using sum of squares surface. *Master's thesis, University of Wisconsin-Madison*.
- Martens, H. and T. Naes (1989). *Multivariate Calibration*. Wiley, New York.
- Osborne, B., T. Fearn, A. Miller, and S. Douglas (1994). 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.
- Phatak, A. and F. de Hoog (2003). Exploiting the Connection between PLS, Lanczos, and Conjugate Gradients: Alternative Proofs of some Properties of PLS. *Journal of Chemometrics* 16, 361–367.
- R Development Core Team (2005). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.
- Ramsay, J. O. and B. W. Silverman (1997). *Functional Data Analysis*. Springer, New York.
- Rännar, S., F. Lindgren, P. Geladi, and S. Wold (1994). A PLS Kernel Algorithm for Data Sets with many Variables and Fewer Objects, Part I: Theory and Applications. *Journal of Chemometrics* 8, 111–125.
- Rosipal, R. and N. Krämer (2006). Overview and Recent Advances in Partial Least Squares. In *Subspace, Latent Structure and Feature Selection Techniques*, Lecture Notes in Computer Science, pp. 34–51. Springer.
- Rosipal, R. and L. Trejo (2001). Kernel Partial Least Squares Regression in Reproducing Kernel Hilbert Spaces. *Journal of Machine Learning Research* 2, 97–123.
- Rosipal, R., L. Trejo, and B. Matthews (2003). Kernel PLS-SVC for Linear and Nonlinear Classification. In *Proceedings of the Twentieth International Conference on Machine Learning*, Washington, DC, pp. 640–647.
- Ruppert, D. (2002). Selecting the Number of Knots for Penalized Splines. *Journal of Computational and Graphical Statistics* 11, 735–757.
- Schölkopf, B. and A. Smola (2002). *Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press.

Wold, H. (1975). Path Models with Latent Variables: The NIPALS Approach. In H. B. et al. (Ed.), *Quantitative Sociology: International Perspectives on Mathematical and Statistical Model Building*, pp. 307–357. Academic Press.

Wold, S., H. Ruhe, H. Wold, and W. D. III (1984). 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.

Wood, S. (2006). *Generalized Additive Models: An Introduction with R*. Chapman and Hall.

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

A Proofs

We recall that for $k < i$

$$\mathbf{X}_i = \prod_{j=k}^{i-1} (\mathbf{I}_n - \mathcal{P}_{\mathbf{t}_j}) \mathbf{X}_k = (\mathbf{I}_n - \mathcal{P}_{\mathbf{t}_k, \dots, \mathbf{t}_{i-1}}) \mathbf{X}_k \quad (19)$$

The last equality follows from the fact that the components \mathbf{t}_i are mutually orthogonal. In particular, we obtain

$$\mathbf{X}_i = (\mathbf{I}_n - \mathcal{P}_{\mathbf{t}_1, \dots, \mathbf{t}_{i-1}}) \mathbf{X}. \quad (20)$$

We first proof a weaker version of proposition 1.

Lemma 5. *The matrix \mathbf{R} defined in proposition 1 is upper triangular and invertible. Furthermore, the columns of \mathbf{T} and the columns of \mathbf{XW} span the same space.*

Proof. First note that (20) is equivalent to $\mathbf{X} = \mathbf{X}_j + \mathcal{P}_{\mathbf{t}_1, \dots, \mathbf{t}_{j-1}} \mathbf{X}$. It follows that

$$\mathbf{X}\mathbf{w}_j = \mathbf{X}_j\mathbf{w}_j + \mathcal{P}_{\mathbf{t}_1, \dots, \mathbf{t}_{j-1}} \mathbf{X}\mathbf{w}_j = \mathbf{t}_j + \sum_{i=1}^{j-1} \frac{\mathbf{t}_i^T \mathbf{X}\mathbf{w}_j}{\mathbf{t}_i^T \mathbf{t}_i} \mathbf{t}_i. \quad (21)$$

As all components \mathbf{t}_i are mutually orthogonal, $\mathbf{t}_i^T \mathbf{X}\mathbf{w}_j = 0$ for $i > j$ and $\mathbf{t}_i^T \mathbf{X}\mathbf{w}_i = \mathbf{t}_i^T \mathbf{t}_i \neq 0$. We conclude that \mathbf{R} is an upper triangular matrix with all diagonal elements $\neq 0$. Furthermore, it follows from (21) that all vectors $\mathbf{X}\mathbf{w}_j$ are linear combinations of the components $\mathbf{t}_1, \dots, \mathbf{t}_j$. This implies that the columns of \mathbf{XW} and the columns of \mathbf{T} span the same space. \square

Now note that the condition $i > j$ implies $\mathbf{X}_i\mathbf{w}_j = \mathbf{X}_j\mathbf{w}_j - \mathcal{P}_{\mathbf{t}_1, \dots, \mathbf{t}_{i-1}} \mathbf{X}_j\mathbf{w}_j = \mathbf{t}_j - \mathbf{t}_j = 0$. From this we can conclude directly that the weight vectors of penalized PLS are mutually \mathbf{M}^{-1} -orthogonal. This follows as for $i > j$

$$\langle \mathbf{w}_i, \mathbf{w}_j \rangle_{\mathbf{M}^{-1}} = \langle \mathbf{M}\mathbf{X}_i^T \mathbf{y}, \mathbf{w}_j \rangle_{\mathbf{M}^{-1}} = \mathbf{y}^T \mathbf{X}_i \mathbf{M} \mathbf{M}^{-1} \mathbf{w}_j = \mathbf{y}^T \mathbf{X}_i \mathbf{w}_j = \mathbf{y}^T \mathbf{0} = 0. \quad (22)$$

Proof of lemma 3. We use induction. For $m = 1$, $\mathbf{w}_1 = \mathbf{b}_M$. For a fixed $m > 1$, we conclude from the induction hypothesis and lemma 5 that every vector $\mathbf{s} \in \text{span}\{\mathbf{t}_1, \dots, \mathbf{t}_m\}$ is of the form $\mathbf{s} = \mathbf{X}\mathbf{v}$ with $\mathbf{v} \in \mathcal{K}_m$. We conclude that

$$\mathbf{w}_{m+1} = \mathbf{M}\mathbf{X}_{m+1}^T \mathbf{y} = \mathbf{M}\mathbf{X}^T \mathbf{y} - \mathbf{M}\mathbf{X}^T \mathcal{P}_{\mathbf{t}_1, \dots, \mathbf{t}_m} \mathbf{y} = \mathbf{b}_M - \mathbf{M}\mathbf{X}^T \mathbf{X} \mathbf{s} = \mathbf{b}_M - \mathbf{A}_M \mathbf{s} \in \mathcal{K}_m. \quad \square$$

Proof of proposition 1. It follows from lemma 3 and the fact that \mathbf{T} and \mathbf{XW} span the same space that $\mathbf{t}_i \in \mathbf{X}\mathcal{K}_m$. We can conclude that

$$\mathbf{MX}^t \mathbf{t}_i \in \mathbf{MX}^t \mathbf{X}\mathcal{K}_i = \mathbf{A}_M \mathcal{K}_i \subset \mathcal{K}_{i+1} = \text{span}\{\mathbf{w}_1, \dots, \mathbf{w}_{i+1}\}.$$

In particular,

$$\mathbf{MX}^t \mathbf{t}_i = \sum_{k=1}^{i+1} \alpha_k \mathbf{w}_k. \quad (23)$$

Now recall (22). We conclude that for $j > i + 1$

$$\mathbf{t}_i^t \mathbf{X} \mathbf{w}_j = \langle \mathbf{MX}^t \mathbf{t}_i, \mathbf{w}_j \rangle_{M^{-1}} \stackrel{(23)}{=} \left\langle \sum_{k=1}^{i+1} \alpha_k \mathbf{w}_k, \mathbf{w}_j \right\rangle_{M^{-1}} \stackrel{(22)}{=} \sum_{k=1}^{i+1} \alpha_k 0 = 0$$

Finally, (11) is equivalent to (21). \square

Proof of theorem 4. First note that it can be shown via induction that $\text{span}\{\mathbf{d}_0, \dots, \mathbf{d}_{m-1}\} = \mathcal{K}_m$. Next, we have

$$\boldsymbol{\beta}_m = \sum_{i=0}^{m-1} \frac{\langle \mathbf{d}_i, \mathbf{b}_M \rangle_{M^{-1}}}{\langle \mathbf{d}_i, \mathbf{A}_M \mathbf{d}_i \rangle_{M^{-1}}} \mathbf{d}_i. \quad (24)$$

This corresponds to the iterative definition of $\boldsymbol{\beta}_m$. We only have to show that $\langle \mathbf{d}_i, \mathbf{r}_i \rangle_{M^{-1}} = \langle \mathbf{d}_i, \mathbf{b}_M \rangle_{M^{-1}}$. Note that

$$\mathbf{r}_i = \mathbf{b}_M - \sum_{j=0}^{i-1} a_j \mathbf{A}_M \mathbf{d}_j.$$

As \mathbf{d}_i is \mathbf{A}_M -orthogonal onto all directions \mathbf{d}_j , $j < i$, (24) holds. Now, as \mathbf{T} and \mathbf{XW} span the same space, we have

$$\hat{\mathbf{y}}_m = \mathcal{P}_{\mathbf{XW}} \mathbf{y} = \mathbf{XW} (\mathbf{W}^T \mathbf{X}^T \mathbf{XW})^{-1} \mathbf{W}^T \mathbf{X}^T \mathbf{y} = \mathbf{X} \hat{\boldsymbol{\beta}}_m.$$

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

$$\hat{\boldsymbol{\beta}}_m = \mathbf{D} (\mathbf{D}^T \mathbf{A}_M \mathbf{D})^{-1} \mathbf{D}^T \mathbf{b} = \mathbf{D} (\mathbf{D}^T \mathbf{M}^{-1} \mathbf{A}_M \mathbf{D})^{-1} \mathbf{D}^T \mathbf{M}^{-1} \mathbf{b}_M = \sum_{i=0}^{m-1} \frac{\langle \mathbf{d}_i, \mathbf{b}_M \rangle_{M^{-1}}}{\langle \mathbf{d}_i, \mathbf{A}_M \mathbf{d}_i \rangle_{M^{-1}}} \mathbf{d}_i$$

and this equals (24). \square