Nonlinear Partial Least Squares: An Overview

Roman Rosipal
Department of Medical Cybernetics and Artificial Intelligence
Center for Brain Research
Medical University of Vienna, Austria
&
Pacific Development and Technology, LLC
Palo Alto, CA, USA

Abstract

In many areas of research and industrial situations, including many data analytic problems in chemistry, a strong nonlinear relation between different sets of data may exist. While linear models may be a good simple approximation to these problems, when nonlinearity is severe they often perform unacceptably. The nonlinear partial least squares (PLS) method was developed in the area of chemical data analysis. A specific feature of PLS is that relations between sets of observed variables are modeled by means of latent variables usually not directly observed and measured. Since its introduction, two methodologically different concepts of fitting existing nonlinear relationships initiated development of a series of different nonlinear PLS models. General principles of the two concepts and representative models are reviewed in this chapter. The aim of the chapter is two-fold i) to clearly summarize achieved results and thus ii) to motivate development of new computationally efficient nonlinear PLS models with better performance and good interpretability.

Keywords

partial least squares, nonlinear mapping, kernel learning

Introduction

Two-block linear partial least squares (PLS) has been proven to be a valuable method for modeling relationships between two data sets (data blocks). This method was developed in chemometrics and has received a great deal of attention in the fields of analytical chemistry, organic and bio-organic chemistry, medicinal chemistry and chemical engineering. PLS has also been successfully applied in other scientific areas including bioinformatics (Boulesteix & Strimmer, 2007), food research (Martens & Martens, 1986), medicine (Worsley, 1997), pharmacology (Leach & Gillet, 2003;

Nilsson, Jong, & Smilde, 1997), social sciences (Hulland, 1999), physiology and neuro-physiology (Lobaugh, West, & McIntosh, 2001; Trejo, Rosipal, & Matthews, 2006), to name a few.

PLS models relationships between sets of observed variables by means of latent variables. It can serve for regression and classification tasks as well as dimension reduction techniques and modeling. The underlying assumption of all PLS methods is that the observed data is generated by a system or process which is driven by a small number of latent (not directly observed or measured) variables. This projection of the observed data onto a subspace of usually orthogonal latent variables has been shown to be a powerful technique when observed variables are highly correlated, noisy and the ratio between the number of observations (data samples) and observed variables is low. The basic assumption of linear PLS is that the studied relation between observed data sets is linear and the same assumption of linearity then holds for modeling the relation in the projected subspace; that is, between latent variables.

However, in many areas of research and industrial situations a strong nonlinear relation between sets of data may exist. Although linear PLS can be used to approximate this nonlinearity, in many situations such approximation may not be adequate and the use of a nonlinear model is needed.

This chapter introduces the main concepts of nonlinear PLS and provides an overview of its application to different data analysis problems. The aim is to present a concise introduction that is a valuable guide for anyone who is concerned with nonlinear data analysis.

Background

The concept of nonlinear PLS modeling was introduced by S. Wold, Kettaneh-Wold, and Skagerberg (1989). Already in this seminal work, the authors distinguished and described two basic principles for modeling curved relationships between sets of observed data. The first principle, here denoted as Type I, is well-known and used in mathematical statistics and other research fields. The principle applies first a nonlinear transformation to observed variables. In the new representation a linear model is constructed. This principle can be easily applied to PLS, and indeed several different nonlinear PLS models were proposed and applied to real data sets. The first nonlinear PLS models in this category were constructed by using simple polynomial transformations of the observed data (Berglund & Wold, 1997, 1999). However, the proposed polynomial transformation approach possesses several computational and generalization limitations. To overcome these limitations, a computationally elegant kernel PLS method was proposed by Rosipal and Trejo (2001). The powerful concept of a kernel mapping function allows to construct highly flexible but still computationally simple nonlinear PLS models. However, in spite of the ability of kernel PLS to fit highly complex nonlinear data relationships, the model represents a 'black-box' with limited possibility to interpret the results with respect to the original data.

It is the second, here denoted as Type II, general principle for constructing nonlinear PLS models which overcomes the problem of loss of interpretability, but this is achieved at the expense of computational cost and optimization complexity. In contrast to the Type I principle, a nonlinear relation between latent variables is assumed and modeled, while the extracted latent vectors themselves are kept to be a linear combination of the original, not transformed, data. A quadratic function was used to fit relationship between latent variables in the first Type II nonlinear PLS approaches (Höskuldsson, 1992; S. Wold et al., 1989). Later, smoothing splines (Frank, 1990; S. Wold, 1992), artificial neural networks (Baffi, Martin, & Morris, 2000; Qin & McAvoy, 1992), radial basis neural networks (Wilson, Irwin, & Lightbody, 1997) or genetic programming methods (Hiden, McKay, Willis, & Montague, 1998) were used to fit more complex nonlinear relationships. Computational and optimization difficulties of the approach arise at the point when initially estimated weights for projecting observed data into latent vectors need to be corrected. The initial weights are estimated in the first step of the approach when a linear relationship between latent vectors is assumed. However, this linear assumption is violated by considering a nonlinear relation between the extracted latent vectors in the second step. Several methods were proposed to iteratively update both, the initial weights estimate and the latent variables relation fitting function, but a simple optimization and computational methodology is still missing (Baffi, Martin, & Morris, 1999; Searson, Willis, & Montague, 2007; S. Wold et al., 1989; Yoshida & Funatsu, 1997).

Before directly jumping into the area of nonlinear PLS, a section devoted to fundamentals of linear PLS is provided. The description uses nomenclature and originates from a recently published detailed survey of the variants of linear PLS and advances in the domain (Rosipal & Krämer, 2006). For readers interested in detailed understanding of linear PLS the survey can be a good starting point before reading this chapter. Next, an overview of the nonlinear PLS concepts is presented. Special emphasis is placed on kernel learning and the kernel PLS description. Detailed procedural and algorithmic implementation of each method mentioned goes beyond the scope of the chapter, but the limitation is compensated by an extensive literature survey. The technical overview of the nonlinear PLS approaches is followed by the section discussing selected positive and negative aspects of the presented methods. Ideas for further comparison, evaluation and extension of the current nonlinear PLS development status are also briefly sketched. A few concluding remarks close the chapter.

Linear Partial Least Squares

Consider the general setting of a linear PLS algorithm to model the relation between two data sets (blocks of observed variables). Denote by $\mathbf{x} \in \mathcal{X} \subset \mathcal{R}^N$ an N-dimensional vector of variables in the first set of data and similarly by $\mathbf{y} \in \mathcal{Y} \subset \mathcal{R}^M$ a vector of variables from the second set. PLS models the relationship between the two data sets by means of latent vectors (score vectors, components). Denote by n the number of data samples and let $\mathbf{X}$ be the $(n \times N)$ matrix of centered (zero-mean) variables sampled from the $\mathcal{X}$-space. Similarly, let the $\mathcal{Y}$-space data are represented by the $(n \times M)$ zero-mean matrix $\mathbf{Y}$. PLS decomposes the $\mathbf{X}$ and $\mathbf{Y}$ matrices into the form

$$\begin{aligned} \mathbf{X} &= \mathbf{T}\mathbf{P}^T + \mathbf{E} \\ \mathbf{Y} &= \mathbf{U}\mathbf{Q}^T + \mathbf{F} \end{aligned} \tag{1}$$

where $\mathbf{T}$ and $\mathbf{U}$ are the $(n \times p)$ matrices of the p extracted score (latent) vectors, the $(N \times p)$ matrix $\mathbf{P}$ and the $(M \times p)$ matrix $\mathbf{Q}$ represent matrices of loadings and the $(n \times N)$ matrix $\mathbf{E}$ and the $(n \times M)$ matrix $\mathbf{F}$ are the matrices of residuals. The PLS method, which in its classical form is based on the nonlinear iterative partial least squares (NIPALS) algorithm (H. Wold, 1975), finds weight vectors $\mathbf{w}, \mathbf{c}$ such that

$$[cov(\mathbf{t}, \mathbf{u})]^2 = [cov(\mathbf{X}\mathbf{w}, \mathbf{Y}\mathbf{c})]^2 = max_{|\mathbf{r}|=|\mathbf{s}|=1}[cov(\mathbf{X}\mathbf{r}, \mathbf{Y}\mathbf{s})]^2 \tag{2}$$

where $cov(\mathbf{t}, \mathbf{u}) = \mathbf{t}^T\mathbf{u}/n$ denotes the sample covariance between the score vectors $\mathbf{t}$ and $\mathbf{u}$. The NIPALS algorithm starts with random initialization of the $\mathcal{Y}$-space score vector $\mathbf{u}$ and repeats a sequence of the following steps until convergence

$$\begin{array}{ll} 1)\ \mathbf{w} = \mathbf{X}^T\mathbf{u}/(\mathbf{u}^T\mathbf{u}) & 4)\ \mathbf{c} = \mathbf{Y}^T\mathbf{t}/(\mathbf{t}^T\mathbf{t}) \\ 2)\ \|\mathbf{w}\| \to 1 & 5)\ \|\mathbf{c}\| \to 1 \\ 3)\ \mathbf{t} = \mathbf{X}\mathbf{w} & 6)\ \mathbf{u} = \mathbf{Y}\mathbf{c} \end{array} \tag{3}$$

where $\|.\| \rightarrow 1$ denotes transformation of a vector to unit norm. Once the score vectors $\mathbf{t}$ and $\mathbf{u}$ are extracted, the vectors of loadings $\mathbf{p}$ and $\mathbf{q}$ from (1) can be computed by regressing $\mathbf{X}$ on $\mathbf{t}$ and $\mathbf{Y}$ on $\mathbf{u}$, respectively

$$\mathbf{p} = \mathbf{X}^T\mathbf{t}/(\mathbf{t}^T\mathbf{t}) \quad \text{and} \quad \mathbf{q} = \mathbf{Y}^T\mathbf{u}/(\mathbf{u}^T\mathbf{u})$$

Note that different numerical techniques can be used to extract weight vectors and corresponding score vectors, some of which can be more efficient than NIPALS (De Jong, 1993; Höskuldsson, 1988).

PLS is an iterative process. After the extraction of the score vectors $\mathbf{t}$ and $\mathbf{u}$, the matrices $\mathbf{X}$ and $\mathbf{Y}$ are deflated by subtracting their rank-one matrix approximations. Different forms of deflation exist, and each form defines a certain variant of PLS (for example, see Rosipal and Krämer (2006)). The most frequently used variant of linear PLS is based on two assumptions i) the score vectors $\{\mathbf{t}_i\}_{i=1}^p$ are good predictors of $\mathbf{Y}$, and ii) a linear *inner relation* between the scores vectors $\mathbf{t}$ and $\mathbf{u}$ exists; that is,

$$\mathbf{U} = \mathbf{TD} + \mathbf{H} \tag{4}$$

where $\mathbf{D}$ is the $(p \times p)$ diagonal matrix and $\mathbf{H}$ denotes the matrix of residuals. The asymmetric assumption of the input–output (predictor–predicted) variables relation is transformed into a deflation scheme where the input space score vectors $\{\mathbf{t}_i\}_{i=1}^p$ are good predictors of $\mathbf{Y}$. The score vectors are then used to deflate $\mathbf{Y}$; that is, a component of the regression of $\mathbf{Y}$ on $\mathbf{t}$ is removed from $\mathbf{Y}$ at each iteration of PLS

$$\mathbf{X} \leftarrow \mathbf{X} - \mathbf{t}\mathbf{t}^T\mathbf{X}/(\mathbf{t}^T\mathbf{t}) \quad \text{and} \quad \mathbf{Y} \leftarrow \mathbf{Y} - \mathbf{t}\mathbf{t}^T\mathbf{Y}/(\mathbf{t}^T\mathbf{t}) = \mathbf{Y} - \mathbf{t}\mathbf{c}^T \tag{5}$$

where $\mathbf{c}$ is the weight vector defined in step 4 of the NIPALS algorithm (3).

PLS Regression and Classification

By considering the assumption (4) of a linear relation between the scores vectors $\mathbf{t}$ and $\mathbf{u}$, the decomposition of the $\mathbf{Y}$ matrix in (1) can be rewritten as

$$\mathbf{Y} = \mathbf{TDQ}^T + (\mathbf{HQ}^T + \mathbf{F})$$

and this defines the linear PLS regression model

$$\mathbf{Y} = \mathbf{TC}^T + \mathbf{F}^* \tag{6}$$

where $\mathbf{C}^T = \mathbf{DQ}^T$ denotes the $(p \times M)$ matrix of regression coefficients and $\mathbf{F}^* = \mathbf{HQ}^T + \mathbf{F}$ is the residual matrix. Equation (6) is simply the decomposition of $\mathbf{Y}$ using ordinary least squares regression with orthogonal predictors $\mathbf{T}$; that is, the estimate of $\mathbf{C}$ is given as $\mathbf{C} = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{Y}$. The PLS regression model (6) can also be expressed using the originally observed data $\mathbf{X}$ and written as (Höskuldsson, 1988; Manne, 1987; Rännar, Lindgren, Geladi, & Wold, 1994; Rosipal & Krämer, 2006)

$$\mathbf{Y} = \mathbf{XB} + \mathbf{F}^* \tag{7}$$

where the estimate of $\mathbf{B}$ has the following form

$$\mathbf{B} = \mathbf{X}^T\mathbf{U}(\mathbf{T}^T\mathbf{X}\mathbf{X}^T\mathbf{U})^{-1}\mathbf{T}^T\mathbf{Y}$$

The PLS classification model closely follows the regression model (6). However, in the classification scenario the $\mathbf{Y}$ matrix is a class membership matrix uniquely coding each class of data. The close connection between Fischer discriminant analysis, canonical correlation analysis (CCA) and PLS has been discussed in Barker and Rayens (2003); De Bie, Cristianini, and Rosipal (2005); Rosipal, Trejo, and Matthews (2003); Rosipal and Krämer (2006).

The major focus of the chapter is the nonlinear extraction of the PLS score vectors and the followed up regression or classification step using the extracted vectors is somehow irrelevant to us. Moreover, variants of PLS similar to CCA for modeling symmetric relationships between sets of data exist (see Rosipal and Krämer (2006) or Wegelin (2000) for overview). There is no theoretical limitation to apply the principles of nonlinear PLS to these variants as well.

Nonlinear Partial Least Squares

Several different nonlinear PLS methods have been proposed. In principle these methods can be categorized into two groups. The Type I group of approaches consists of models where the observed **X** matrix of independent variables is projected onto a nonlinear surface. The inner relation (4) between score vectors **t** and **u** is kept linear. In contrast, it is the Type II group of approaches where the nonlinear relation between **X** and **Y** data sets is modeled by replacing the linear relation (4) with a nonlinear form. In what follows, both types of nonlinear PLS are described in detail.

Type I: Nonlinear projection of X

Methods of this group of nonlinear PLS are based on a principle of mapping the original data by means of a nonlinear function to a new representation (data space) where linear PLS is applied. A simple example would be the extension of the **X** matrix by considering component-wise square terms x_i^2 and cross-terms $x_i x_j$ of the input vector $\mathbf{x}$. It has been shown by Gnanadesikan (1977) and pointed out by S. Wold et al. (1989) that such an extension corresponds to the projection of the original data space onto a quadratic surface. In fact, this result was used by S. Wold et al. (1984) for PLS response surface modeling. To better understand this idea consider a simple binary classification problem depicted in the left part of Figure 1. An ellipsoidal boundary between two classes depicted by circles and crosses would be impossible to properly model with a linear line. Now consider a simple transformation of the original 2D data into a 3D data space denoted by $\mathcal{F}$ and defined by the following mapping

$$\begin{aligned} \Phi: \quad & \mathcal{X} = \mathcal{R}^2 \rightarrow \mathcal{F} = \mathcal{R}^3 \\ & \mathbf{x} = (x_1, x_2) \rightarrow \Phi(\mathbf{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2) \end{aligned} \tag{8}$$

It can be easily observed (right part of Figure 1) that the original nonlinear classification problem in 2D was transformed into a linear problem in 3D.

However, in practice it might be very difficult to find such a simple nonlinear transformation or extension of the original data space into a new space where the original nonlinear problem becomes linear. This is mainly due to the reason that we do not know the exact shape of the nonlinear boundary in the classification scenario or we do not know the exact form of the nonlinear relationship between predictor and predicted space in regression. Nevertheless, following the idea that in a higher dimensional space, where original data are mapped, the nonlinear problem becomes easier to solve or a less complex boundary can be found, many very useful techniques and methods have been developed in different research communities. Polynomial regression (for example see Seber and Lee (2003)), generalized additive models (Hastie & Tibshirani, 1990), projection pursuit regression (Friedman & Stuetzle, 1981), higher order splines or multivariate adaptive regression splines (Friedman, 1991; Wahba, 1990) are a few of the popular models developed in the statistical community. Regularization networks (Girosi, Jones, & Poggio, 1995), artificial neural networks

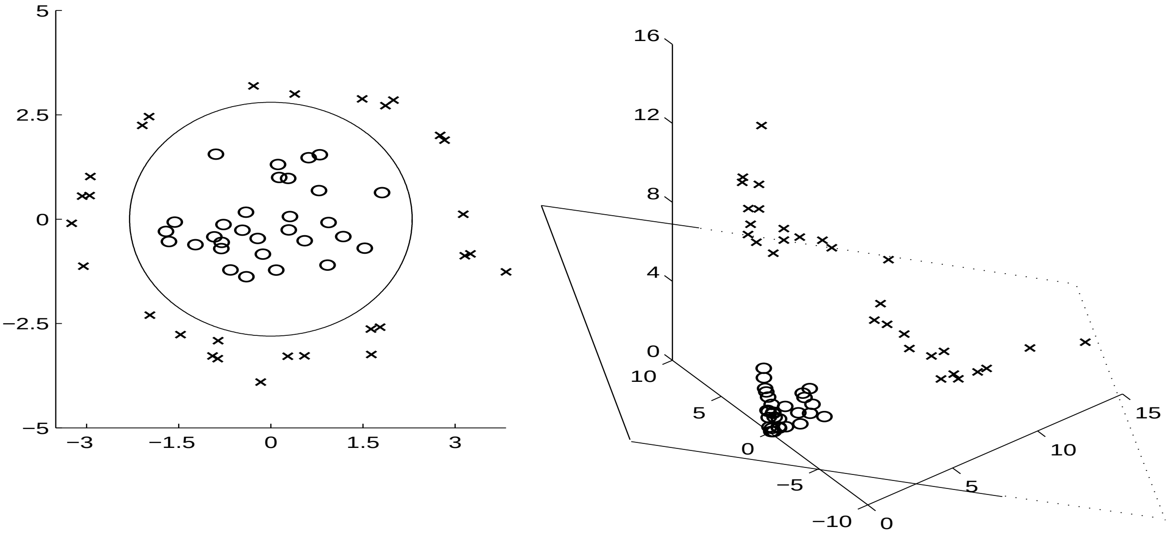

Figure 1: An example of transformation of a nonlinear classification problem with an ellipsoidal decision boundary (*left*) to a problem where a linear hyperplane can be used to separate two classes (*right*). Two classes are defined by circles and crosses, respectively. The original 2D data (*left*) were transformed into a 3D space (*right*) using the nonlinear mapping defined in eq. (8).

(Haykin, 1999) or recently developed theory and algorithms of kernel learning and support vector machines are examples of nonlinear model developments in the machine learning community (Cristianini & Shawe-Taylor, 2000; Schölkopf & Smola, 2002; Shawe-Taylor & Cristianini, 2004).

The nonlinear PLS model with quadratic projection of **X** was firstly discussed by S. Wold et al. (1989), but the principle of extending **X** with nonlinear terms was elaborated later by Berglund and Wold (1997). In their approach, called implicit nonlinear latent variables regression (INLR), Berglund and Wold proposed to extend the **X** matrix with squared x_i^2, cubic x_i^3 or higher order polynomial terms while ignoring cross-terms. To support this idea they showed that by "expanding the **X**-block with square terms in some ways corresponds to using a model that includes not only the squares of the latent variables, $\mathbf{t}_1^2, \mathbf{t}_2^2, \mathbf{t}_3^2$, etc., but also the cross-products of these, $\mathbf{t}_1\mathbf{t}_2, \mathbf{t}_1\mathbf{t}_3, \mathbf{t}_2\mathbf{t}_3$, etc." (Berglund & Wold, 1997).

On the other hand, their expansion of the **X** matrix with squared or higher order terms only, was strongly motivated by scaling limits of the full expansion. In the case of a high-dimensional predictor data and a small number of observations, that is, $n << N$, expanding **X** with both squared and cross-terms would result into extreme increase of the number of new variables in comparison to the existing number of observations. For an N-dimensional input space, there exist $\frac{(d+N-1)!}{d!(N-1)!}$ different the d-th order products (monomials) of the elements x_i.

For example, consider a standard problem from near-infrared reflectance spectroscopy where the predictor space consists of 250 absorbances at different wavelengths. Using the second-order polynomial expansion would result in a new expanded space with 31375 variables. Cubic expansion ($d = 3$) would increase this number to more than 2.6 million monomial terms and the problem quickly becomes computationally intractable.

The other problem associated with such an expansion is related to the one known as 'curse of dimensionality'. The curse of dimensionality is the term associated with the problem of an exponential increase of parameters which need to be estimated when mapping original data into a high-dimensional space, while the number of observed samples remains unchanged. For example, a regression model fitted by ordinary least squares method would result in an unbiased estimate of regression coefficients, but the variance of the estimate can be unacceptably high. Thus, although such a model will fit training data well, in the sense of sum of square errors, its generalization ability to fit previously unobserved (testing) data can be very poor. To avoid the problem, different forms of regularization are usually applied. One approach that is often used is based on diminishing the influence of regression coefficients with high absolute values. This can be done by penalizing a properly selected norm of the vector of estimated regression coefficients.

However, the new concept of statistical learning theory developed in its origin by Vapnik and Chervonenkis (Vapnik, 1998, 1995) provided new insights and perspectives into the problems of curse of dimensionality and bad generalization of learning machines. Based on the theory new support vector machines and kernel learning algorithms and concepts were developed. Among other models the nonlinear kernel PLS method was proposed. The basic principles of kernel learning are described in the following subsection.

Kernel Learning and Kernel PLS

Statistical learning theory was introduced by Vapnik and Chervonenkis in the early 1970's (Vapnik, 1998; Vapnik & Chervonenkis, 1974). However, it was not until the middle of the 1990's when this theory inspired development of new types of learning algorithms. The core principle of the new algorithms is the mapping of originally observed data into a high-dimensional feature space where simple linear models are constructed. The generalization abilities of the constructed models are controlled by considering theoretical results of the structural risk minimization (SRM) principle. SRM is an inductive principle where a trade-off between good-model fitting and the complexity of the model is appropriately balanced. In other words, having a finite set of data, a model selection step defined by the principle consists of two aspects which needs to be considered in parallel i) the quality of fitting training data (empirical error) and ii) the complexity of the hypothesis space where the final model is constructed. Over-complex models would fit training data perfectly well, but they may show high prediction or classification errors when dealing with new, previously unobserved, testing data. Vapnik and Chervonenkis (1971) introduced the concept of the VC dimension (for Vapnik-Chervonenkis dimension) which can be used as a measure of the complexity of a hypothesis space in which learning machines are constructed. Motivated by this concept of the VC dimension very powerful support vector machines (SVM) for classification were constructed (Cristianini & Shawe-Taylor, 2000; Schölkopf & Smola, 2002; Vapnik, 1998). For real-valued function approximations, including regression tasks, the SRM principle motivated the development of new, and renewal of previous theoretical and practical results of regularization theory (Girosi et al., 1995). Regularization theory was proposed in the 1960's by Tichonov and Ivanov (Ivanov, 1976; Tikhonov, 1963). Later, Kimeldorf and Wahba (1971) proved the important *representer theorem*, which also applies to kernel learning algorithms, including SVM. Using a less rigorous mathematical language, the representer theorem defines the form of an approximation or regression function $f(\mathbf{x})$ constructed in a functional space $\mathcal{H}$ and minimizing the following functional form

$$\min_{f \in \mathcal{H}} R_{reg} = \sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i)) + \xi \|f\|^2_{\mathcal{H}} \tag{9}$$

where $L(y_i, f(\mathbf{x}_i))$ is a loss function measuring point-wise difference between observed values $\{y_i\}_{i=1}^n$ and their approximations $\{f(\mathbf{x}_i)\}_{i=1}^n$. A typical example of L would be the squared loss $(y_i - f(\mathbf{x}_i))^2$. The regularization term ξ is a positive number controlling the trade-off between approximating properties and the smoothness of f. The squared norm $\|f\|^2_{\mathcal{H}}$ is sometimes called the 'stabilizer'. The representer theorem states that the solution of (9) can be always written in the form

$$f(\mathbf{x}) = \sum_{i=1}^{n} d_i k(\mathbf{x}, \mathbf{x}_i) \tag{10}$$

where $\{d_i\}_{i=1}^n$ are real-value coefficients and $k(\mathbf{x}, \mathbf{x}_i)$ is a symmetric positive definite function of two variables called the *kernel function.* Next, the properties and specific relations between the space $\mathcal{H}$ and the kernel function k will be discussed.

First, return to the mapping defined in (8) and consider the mappings $\Phi(\mathbf{x})$, $\Phi(\mathbf{y}) \in \mathcal{F}$ of the two points $\mathbf{x}, \mathbf{y} \in \mathcal{X}$

$$\begin{array}{lcl} \mathbf{x} = (x_1, x_2) & \rightarrow & \Phi(\mathbf{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2) \\ \mathbf{y} = (y_1, y_2) & \rightarrow & \Phi(\mathbf{y}) = (y_1^2, \sqrt{2}y_1y_2, y_2^2) \end{array}$$

Now compute a dot product between these two mappings

$$\begin{aligned} \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle &= (x_1^2, \sqrt{2}x_1x_2, x_2^2)^T (y_1^2, \sqrt{2}y_1y_2, y_2^2) \\ &= ((x_1, x_2)^T (y_1, y_2))^2 \\ &= \langle \mathbf{x}, \mathbf{y} \rangle^2 \end{aligned}$$

It becomes clear that $\langle \mathbf{x}, \mathbf{y} \rangle^2$ corresponds to a canonical (Euclidean) dot product in the space $\mathcal{F}$ where the input data were transformed by Φ. This is an important concept where computation of a dot product between two vectors in the feature space $\mathcal{F}$ can be replaced by evaluating the function $\langle \mathbf{x}, \mathbf{y} \rangle^2$ instead; that is, by computing a square value of a dot product between the two original points $\mathbf{x}, \mathbf{y}$ in $\mathcal{X}$. In our previous example with the 250-dimensional input space of absorbances this would mean to evaluate square values of the dot product between two 250-dimensional vectors instead of computing the dot product of two 31375-dimensional vectors in the mapped space. More interestingly the function $\langle \mathbf{x}, \mathbf{y} \rangle^2$ satisfies the Mercer theorem conditions (Cristianini & Shawe-Taylor, 2000; Mercer, 1909) and it is a valid kernel function known as the second order polynomial kernel $k(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle^2$. The Moore-Aronszajn theorem establishes the fact that for any kernel function, there exists a unique functional space, called a reproducing kernel Hilbert space (RKHS) (Aronszajn, 1950). Now, return to the representer theorem and define $\mathcal{H}$ to be a RKHS corresponding to a kernel function k. The link between the RKHS space $\mathcal{H}$ defined by k and a space of mapped features vectors $\mathcal{F}$ follows from Mercer's theorem. Any kernel function can be written in the form

$$k(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{\mathcal{D}_{\mathcal{H}}} \alpha_i \psi_i(\mathbf{x}) \psi_i(\mathbf{y}) = \Phi(\mathbf{x})^T \Phi(\mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle$$

where $\{\psi_i\}_{i=1}^{\mathcal{D}_{\mathcal{H}}}$ is a sequence of linearly independent functions, $\{\alpha_i\}_{i=1}^{\mathcal{D}_{\mathcal{H}}}$ are positive numbers and $\mathcal{D}_{\mathcal{H}} \leq \infty$ is the dimension of the space $\mathcal{H}$. Following this relation the feature map Φ can be written as

$$\begin{array}{rlcl} \Phi : & \mathcal{X} & \rightarrow & \mathcal{F} \\ & \mathbf{x} & \rightarrow & \Phi(\mathbf{x}) = (\sqrt{\alpha_1}\psi_1(\mathbf{x}), \sqrt{\alpha_2}\psi_2(\mathbf{x}), \ldots, \sqrt{\alpha_{\mathcal{D}_{\mathcal{H}}}}\psi_{\mathcal{D}_{\mathcal{H}}}(\mathbf{x})) \end{array}$$

Thus, if we are only interested in the computation of dot products in $\mathcal{F}$, it does not matter how $\mathcal{F}$ was constructed and simply all dot products can be replaced by a unique kernel function associated with $\mathcal{F}$. This is important to note because different feature spaces associated with the same kernel function can be constructed (Schölkopf & Smola, 2002). In literature, this replacement of a dot product with the kernel function value is known as the *kernel trick* method.

The polynomial kernel function $k(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle^d$ is a simple extension of the above mentioned second order polynomial mapping by considering a feature space of all monomials of d-th order.

Another kernel function widely used in practice is the Gaussian kernel function $k(\mathbf{x},\mathbf{y}) = e^{(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\delta})}$, where $\delta > 0$ determines the width of the function. Different kernel functions have been used and constructed (Cristianini & Shawe-Taylor, 2000; Saitoh, 1997; Schölkopf & Smola, 2002; Shawe-Taylor & Cristianini, 2004). Interestingly, a linear kernel $k(\mathbf{x},\mathbf{y}) = \langle \mathbf{x},\mathbf{y} \rangle = \mathbf{x}^T\mathbf{y}$ is also an admissible kernel function. Thus, the linear kernel principal component analysis (PCA) and linear kernel PLS methods (Rännar et al., 1994; Wu, Massarat, & De Jong, 1997), previously developed to reduce computational costs in the case where the input data dimension exceeds the number of observed samples ($n < N$), can be considered belonging to the framework of kernel learning.

The recently developed theory of kernel learning has also been applied to PLS. The nonlinear kernel PLS methodology for modeling relations between sets of observed variables, regression and classification problems was proposed by Rosipal and Trejo (2001) and Rosipal et al. (2003). The idea of kernel PLS is based on a nonlinear mapping of the original data from $\mathcal{X}$ into a high-dimensional feature space $\mathcal{F}$ where a linear PLS model is constructed.

Define the Gram matrix $\mathbf{K}$ of the cross dot products between all mapped input data points, that is, $\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^T$, where $\mathbf{\Phi}$ denotes the matrix of the mapped $\mathcal{X}$-space data $\{\Phi(\mathbf{x}_i) \in \mathcal{F}\}_{i=1}^n$. The kernel trick implies that the elements i, j of $\mathbf{K}$ are equal to the values of the kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$. Further consider that the mapped data were centered; that is, $\mathbf{\Phi}$ is a zero-mean matrix. This can be easily achieved by directly centering $\mathbf{K}$ and explicit manipulation of $\mathbf{\Phi}$ is not needed (Schölkopf, Smola, & Müller, 1998; Rosipal & Trejo, 2001). Denote by $\mathbf{I}_n$ an n-dimensional identity matrix and by $\mathbf{1}_n$ a vector of ones with the length of n. The centered Gram matrix can be then computed as $\mathbf{K} \leftarrow (\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T)\mathbf{K}(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T)$.

Now, consider a modified version of the NIPALS algorithm where steps 1 and 3 are merged and the score vectors $\mathbf{t}$ and $\mathbf{u}$ are scaled to unit norm instead of scaling the weight vectors $\mathbf{w}$ and $\mathbf{c}$. The obtained kernel form of the NIPALS algorithm is as follows (Rosipal & Trejo, 2001)[1]

$$\begin{array}{ll} 1)\ \mathbf{t} = \mathbf{\Phi}\mathbf{\Phi}^T\mathbf{u} = \mathbf{K}\mathbf{u} & \quad 4)\ \mathbf{u} = \mathbf{Y}\mathbf{c} \\ 2)\ \|\mathbf{t}\| \rightarrow 1 & \quad 5)\ \|\mathbf{u}\| \rightarrow 1 \\ 3)\ \mathbf{c} = \mathbf{Y}^T\mathbf{t} & \end{array}$$

Although step 2 guarantees orthonormality of the score vectors, the score vectors can be rescaled to follow the standard linear NIPALS algorithm with the unit norm weight vectors $\mathbf{w}$ (Rännar et al., 1994). In the following the unit norm orthonormal score vectors will be considered; that is, $(\mathbf{T}^T\mathbf{T})^{-1} = \mathbf{I}$.

Note that steps 3 and 4 can be further merged which may become useful in applications where an analogous kernel mapping $\mathbf{\Psi}$ of the $\mathcal{Y}$-space data is considered; that is, the Gram matrix $\mathbf{K}_y = \mathbf{\Psi}\mathbf{\Psi}^T$ of the cross dot products between all mapped output data is constructed. Then, the kernel NIPALS algorithm consists of the following four steps

$$\begin{array}{ll} 1)\ \mathbf{t} = \mathbf{K}\mathbf{u} & \quad 3)\ \mathbf{u} = \mathbf{K}_y\mathbf{t} \\ 2)\ \|\mathbf{t}\| \rightarrow 1 & \quad 4)\ \|\mathbf{u}\| \rightarrow 1 \end{array}$$

This form of kernel PLS can be useful when one is interested in modeling symmetric relationships between two sets of data. This is known as the PLS Mode A method (Rosipal & Krämer, 2006; Wegelin, 2000; H. Wold, 1985). The above mentioned kernel form then represents the kernel PLS Mode A form which has a close connection to the kernel CCA method (De Bie et al., 2005).

The important part of the iterative PLS algorithm is the deflation step. In the kernel PLS approach the elements of a feature space $\mathcal{F}$ where data are mapped are usually not accessible and

[1] In the case of the one-dimensional $\mathcal{Y}$-space computationally more efficient kernel PLS algorithms have been proposed (Momma, 2005; Rosipal et al., 2003).

the deflation scheme (5) needs to be replaced by its kernel variant (Rosipal & Trejo, 2001)

$$\mathbf{K} \leftarrow \mathbf{K} - \mathbf{t}\mathbf{t}^T\mathbf{K} - \mathbf{K}\mathbf{t}\mathbf{t}^T + \mathbf{t}\mathbf{t}^T\mathbf{K}\mathbf{t}\mathbf{t}^T = (\mathbf{I} - \mathbf{t}\mathbf{t}^T)\mathbf{K}(\mathbf{I} - \mathbf{t}\mathbf{t}^T)$$

Now continue with the kernel analogy of the linear PLS model defined in the previous section. While the kernel from of the model (6) remains the same,[2] the kernel variant of the model (7) has the following form

$$\mathbf{Y} = \mathbf{\Phi}\mathbf{B} + \mathbf{F}^*$$

where the estimate of $\mathbf{B}$ is now

$$\mathbf{B} = \mathbf{\Phi}^T\mathbf{U}(\mathbf{T}^T\mathbf{K}\mathbf{U})^{-1}\mathbf{T}^T\mathbf{Y}$$

Denote by $\mathbf{d}^m = \mathbf{U}(\mathbf{T}^T\mathbf{K}\mathbf{U})^{-1}\mathbf{T}^T\mathbf{y}^m$, $m = 1, \ldots, M$, where the $(n \times 1)$ vector $\mathbf{y}^m$ represents the m-th output variable. Then the kernel PLS regression estimate of the m-th output for a given input sample $\mathbf{x}$ can be written in the form

$$\hat{\mathbf{y}}^m = \Phi(\mathbf{x})^T\mathbf{\Phi}^T\mathbf{d}^m = \sum_{i=1}^{n} d_i^m k(\mathbf{x}, \mathbf{x}_i)$$

which resembles the solution (10) of the representer theorem. However, the form of regularization in PLS differs from the penalized regression models where a direct penalization of regression coefficients is the part of an optimized formula. Therefore, in the case of KPLS regression, the functional form (9) of the representer theorem cannot be straightforwardly formulated. Penalization proprieties of PLS have been discussed and compared with other penalized regression models elsewhere (for example see Lingjærde & Christophersen, 2000; Rosipal & Krämer, 2006).

By considering the kernel variant of the model (6), the kernel PLS regression estimate $\hat{\mathbf{y}}^m$ can be also written as

$$\hat{\mathbf{y}}^m = c_1^m t_1(\mathbf{x}) + c_2^m t_2(\mathbf{x}) + \ldots + c_p^m t_p(\mathbf{x}) = \sum_{i=1}^{p} c_i^m t_i(\mathbf{x})$$

where $\mathbf{c}^m = \mathbf{T}^T\mathbf{y}^m$ is the estimate of a vector of regression coefficients for the m-th regression model. The notation $\{t_i(\mathbf{x})\}_{i=1}^p$ stresses the fact that the score vectors can now be understood as nonlinear functions sampled at the data points $\mathbf{x}$. The following example demonstrates this point.

In Figure 2 an example of kernel PLS regression is depicted. One hundred uniformly spaced points in the range [0, 3.25] were taken and the corresponding values of the function $z(\mathrm{x}) = 4.26(e^{-\mathrm{x}} - 4e^{-2\mathrm{x}} + 3e^{-3\mathrm{x}})$ were computed. The function was used by Wahba (1990) to demonstrate smoothing spline properties. An additional sample of one hundred points representing noise was generated following the Gaussian distribution with zero-mean and variance equal to 0.04. These points were added to the computed values of z and subsequently the values were centered by the mean. The Gaussian kernel with the width parameter δ equal to 1.8 was used. The extracted score vectors plotted as a function of the input points x are depicted in Figure 3.

Note the following very important aspect about the number of kernel PLS score vectors. In contrast to linear PLS or Type II nonlinear PLS described next, the number of possible kernel PLS score vectors is given by the rank of $\mathbf{K}$, not by the rank of $\mathbf{X}$. The rank of $\mathbf{K}$ is either given by n, the number of different sample points, or by the dimensionality $\mathcal{D}_{\mathcal{H}}$ of $\mathcal{F}$ if $n < \mathcal{D}_{\mathcal{H}}$. However, in practice this is usually not the case because data are mapped in a way that $n << \mathcal{D}_{\mathcal{H}}$. Centering of

[2] Now considering the matrices of score vectors $\mathbf{T}$ and regression coefficients $\mathbf{C}$ to be computed in $\mathcal{F}$.

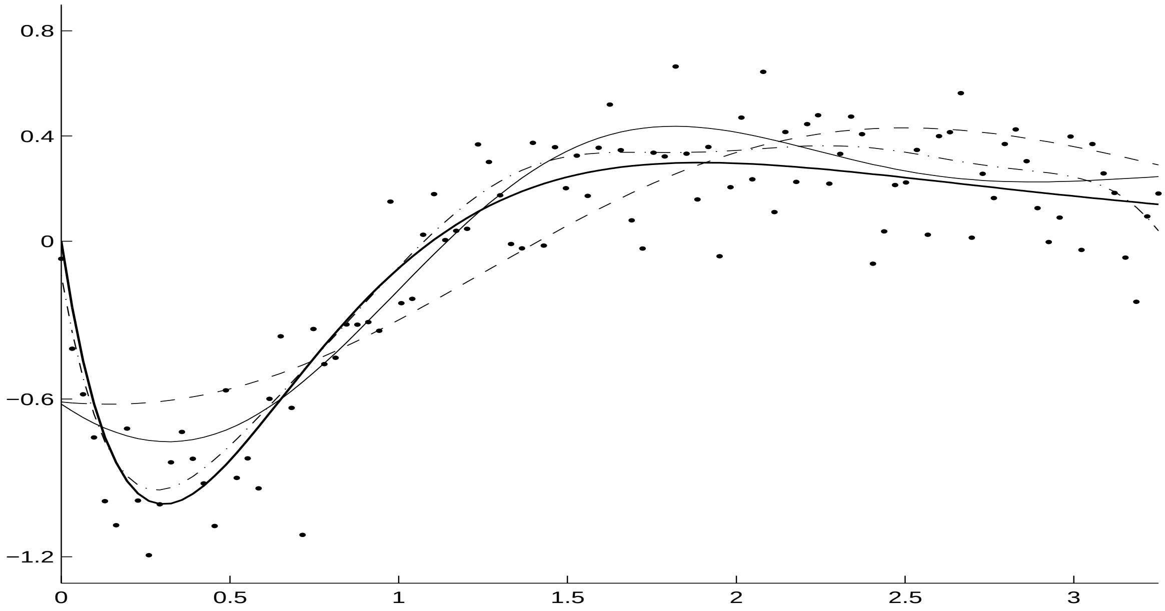

Figure 2: An example of kernel PLS regression. The generated function $z(\mathrm{x}) = 4.26(e^{-\mathrm{x}} - 4e^{-2\mathrm{x}} + 3e^{-3\mathrm{x}})$ is shown as a thick solid line. Dot markers represent noisy form of z used as training output points in kernel PLS regression. Kernel PLS regression using the first one, four and nine score vectors is shown as a dashed, thin solid, and dash-dotted line, respectively. The individual score vectors are plotted in Figure 3.

$\mathbf{K}$ will remove one degree of freedom, therefore the rank of $\mathbf{K}$ is in general equal to $\min(\mathcal{D}_{\mathcal{H}}, n-1)$. In the particular example depicted in Figure 2, the original input space dimension is equal to one and linear or Type II nonlinear PLS can work with one PLS score vector only. However, the kernel PLS method can in theory extract up to 99 nonlinear PLS score vectors. Indeed, to illustrate this point the first five and the ninth score vectors are depicted in Figure 3. This important aspect of finding a 'finer' data decomposition and representation is similar to the properties of kernel PCA as already discussed in (Schölkopf et al., 1998).

Returning to Figure 3, it can be observed that while the first three to four score vectors follow the shape of the approximated function itself, the higher number score vectors start to model the 'wiggling' noisy part of the investigated function. This becomes evident in the last (bottom right) subplot where the ninth score vector is depicted. Note that the score vectors are extracted such that they increasingly describe overall variance in the input data space and more interestingly they also describe the overall variance of the observed output data samples. Smola, Schölkopf, and Müller (1998) has theoretically shown that by choosing the *flattest* function in a feature space $\mathcal{F}$, conditioned by 'good' smoothing properties of the selected kernel function, a smooth function in the input space can be obtained. The term flat function needs to be understood as a linear regression model where the penalization of, in absolute value, large regression coefficients is applied. Shrinkage (or regularized least-square) regression methods like ridge regression, principal component regression or PLS belong to this category, although each of these methods applies a specific principle of penalization (Butler & Denham, 2000; Frank & Friedman, 1993; Goutis, 1996; Krämer, 2007; Lingjærde & Christophersen, 2000; Rosipal & Krämer, 2006).

Finally, I discuss the smoothing proprieties of a kernel mapping. By using different kernel functions feature spaces with different approximation properties are induced. Using the cubic

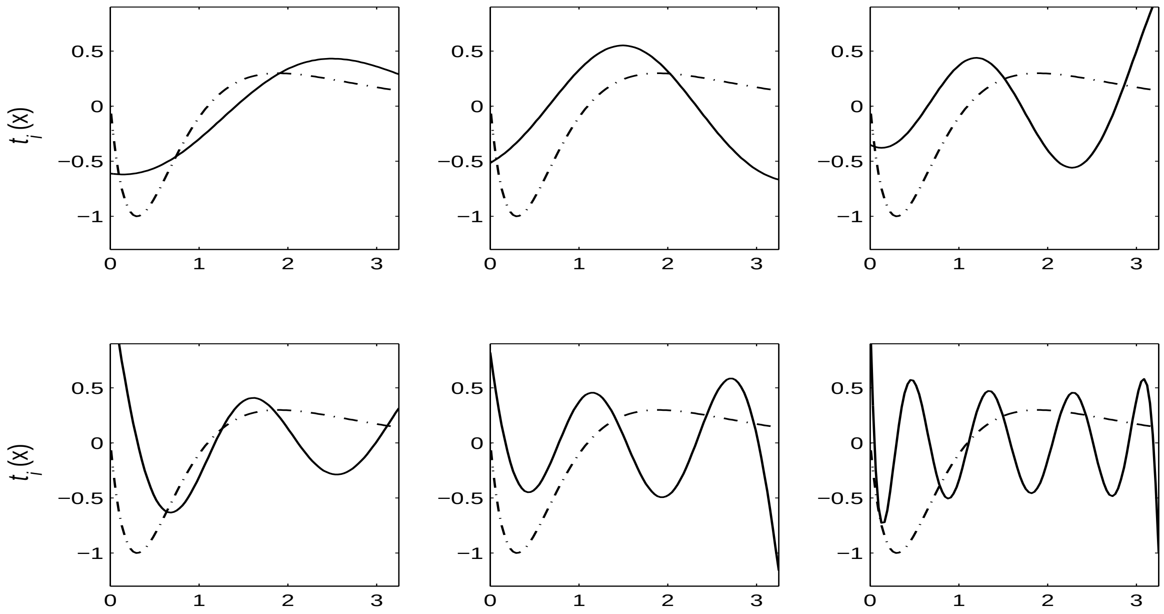

Figure 3: The first five and the ninth (from top left to bottom right) $\mathcal{F}$-space score vectors $t_i(\mathrm{x}), i = 1, \ldots, 5, 9$ computed from noisy signal described in Figure 2. The generated function z without noise is shown dash-dotted.

polynomial kernel will induce a feature space with higher approximation abilities to fit polynomial functions of the third or higher orders in comparison to the second order polynomial kernel. Similarly, by selecting the Gaussian kernel with different values of the width term δ corresponds to choosing feature spaces with different smoothing properties (Girosi, 1998). In general, wider Gaussian kernels will more strongly penalize higher frequency components resulting in smoother estimates. This is demonstrated in Figure 4. Thus, the kernel PLS approach is associated not only with a proper selection of the final number of score vectors but also with a proper selection of the kernel function. This needs to be balanced through the model selection method used. Although cross-validation is usually applied, recently, estimating the effective degrees of freedom in kernel PLS, Krämer and Braun (2007), have discussed and compared several other model selection criteria.

Type II: Nonlinear inner relation

S. Wold et al. (1989) were the first to extend the linear PLS model to its nonlinear form. They have done this by replacing the linear inner relation (4) between the score vectors $\mathbf{t}$ and $\mathbf{u}$ by a nonlinear model

$$\mathbf{u} = g(\mathbf{t}) + \mathbf{h} = g(\mathbf{X}, \mathbf{w}) + \mathbf{h} \tag{11}$$

where g represents a continuous nonlinear function. Again, $\mathbf{h}$ denotes a vector of residuals. The relation between each pair of latent variables is modeled separately; that is, in general a different form of the nonlinear function g can be used for each pair $\mathbf{t}$ and $\mathbf{u}$. Polynomial functions, smoothing splines, artificial neural networks or radial basis function networks have been used to model g. Importantly, in contrast to the Type I nonlinear PLS models, the assumption that the score vectors $\mathbf{t}$ and $\mathbf{u}$ are linear projections of the original variables is kept. This has a significant consequence towards the modification of the original NIPALS algorithm described in the previous linear PLS section. It can be observed that the vector of weights $\mathbf{w}$, computed in the first step of the NIPALS

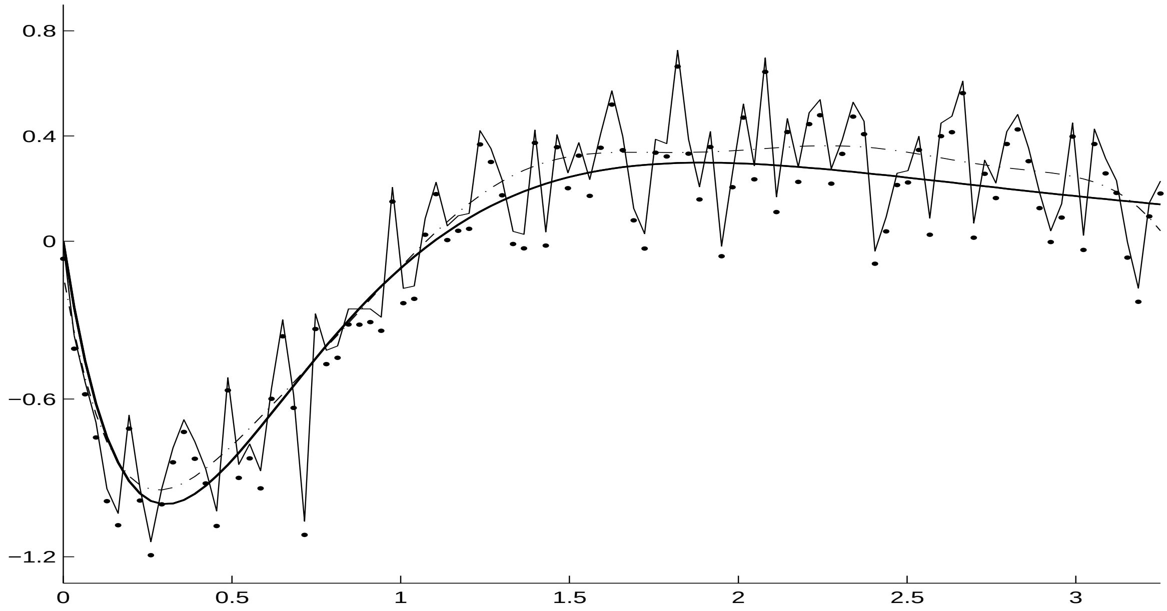

Figure 4: Examples of kernel PLS regression using different values δ of the Gaussian kernel function but the same number of nine score vectors. A dash-dotted line represents the model depicted in Figure 2 with the value of $\delta = 1.8$. A thin solid line represents the model with the extremely small value $\delta = 0.01$. Tendency of this model to follow noise elements represented by dot markers is clearly observed. The generated function without noise is shown as a thick solid line.

algorithm (3), represents the sample covariance between the output space score vector $\mathbf{u}$ and the input space data matrix $\mathbf{X}$. However, the use of a nonlinear model to relate the score vectors in the inner relation affects the computation of $\mathbf{w}$. Although $\mathbf{w}$ represents the association among variables of $\mathbf{X}$ and $\mathbf{u}$ also in nonlinear PLS, this association will be closely related to the covariance values only if the nonlinear mapping between latent variables is monotonic and slightly nonlinear (curved). If this is not the case, an update of $\mathbf{w}$ needs to be considered. Thus, two different approaches have been proposed and used.

In the first approach no update of $\mathbf{w}$ is applied. S. Wold et al. (1989) named this approach the *quick and dirty* nonlinear PLS algorithm and it consists of the standard NIPALS steps until the convergence and the subsequent nonlinear fitting of the inner relation between the extracted pair of the $\mathbf{t}$ and $\mathbf{u}$ vectors. Different nonlinear models were used to fit this relation: higher order polynomial regression (S. Wold et al., 1989), smoothing splines or different types of smoothing estimators (Frank, 1990, 1995), artificial neural networks (Qin & McAvoy, 1992) and radial basis function networks (Wilson et al., 1997).

In the second group of approaches a nonlinear function modeling the inner relation is used to update an initial linear PLS estimate of the weight vector $\mathbf{w}$. S. Wold et al. (1989) proposed to update $\mathbf{w}$ by means of a Newton-Raphson-like linearization of g. The procedure thus consists of a first-order Taylor series expansion of g, followed by the calculation of the correction term $\Delta\mathbf{w}$ which is used to update $\mathbf{w}$. So, consider the nonlinear inner relation (11) where $g(\mathbf{t}) = g(\mathbf{X}, \mathbf{w})$ is continuous and differentiable with respect to $\mathbf{w}$. The second-order Taylor expansion of (11) has the form

$$\hat{\mathbf{u}} = \mathbf{u}_{00} + \left.\frac{\partial g}{\partial \mathbf{w}}\right|_{00} \Delta\mathbf{w} \tag{12}$$

where $\mathbf{u}_{00} = g(\mathbf{t})$ is the value of g at the known value of $\mathbf{t}$. Similarly, $\frac{\partial g}{\partial \mathbf{w}}|_{00}$ stands for the partial derivatives of g numerically evaluated at the same known value of $\mathbf{t}$. The second term of (12) can be written element-wise as

$$\left.\frac{\partial g}{\partial \mathbf{w}}\right|_{00} \Delta\mathbf{w} = \sum_{i=1}^{N} \left.\frac{\partial g}{\partial w_i}\right|_{00} \Delta w_i$$

At this point several different methods to compute the correction $\Delta\mathbf{w}$ were proposed. To simplify further notation consider the matrix form of the linear approximation $\hat{\mathbf{u}}$

$$\hat{\mathbf{u}} = \mathbf{Z}\mathbf{v}$$

where $\mathbf{Z} = [\mathbf{u}_{00} \ \frac{\partial g}{\partial \mathbf{w}}|_{00}]$ and $\mathbf{v} = [1 \ \ \Delta\mathbf{w}]^T$. The following variants to compute $\Delta\mathbf{w}$ were suggested:

S. Wold et al. (1989) proposed to compute $\Delta\mathbf{w}$ by reflecting the first steps of the ordinary NIPALS algorithm

1. $\mathbf{v} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\hat{\mathbf{u}}$ (*S. Wold et al. (1989) original formula:* $\mathbf{v} = \mathbf{Z}^T\hat{\mathbf{u}}/(\hat{\mathbf{u}}^T\hat{\mathbf{u}})$)
2. $\|\mathbf{v}\| \to 1$
3. $\mathbf{s} = \mathbf{Z}\mathbf{v}$
4. $b = \mathbf{u}^T\mathbf{s}/(\mathbf{s}^T\mathbf{s})$
5. *for* $\Delta\mathbf{w}$ *take the corresponding elements of* $b\mathbf{v}$*; that is, elements at the* $2, \ldots, (N+1)$ *positions*

Correspondence of the first three steps of the original S. Wold et al. (1989) algorithm (for step 1 see formula in brackets) with the first steps of NIPALS can be easily observed by replacing $\mathbf{v}$ with $\mathbf{w}$ and $\mathbf{s}$ with $\mathbf{t}$. However, this original idea of the regression of $\mathbf{v}$ on $\mathbf{Z}$, that is, assuming the relation $\mathbf{Z} = \hat{\mathbf{u}}\mathbf{v}$ instead of $\hat{\mathbf{u}} = \mathbf{Z}\mathbf{v}$, was later questioned by Hasegawa, Kimura, Miyashita, and Funatsu (1996). In a personal communication the authors verified this error with S. Wold and they have applied the correct formula $\mathbf{v} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\hat{\mathbf{u}}$. This point and correction was later also repeated by Baffi et al. (1999).

Baffi et al. (1999), initiated by the S. Wold et al. (1989) idea of nonlinear PLS, proposed a different way for updating $\mathbf{w}$. The approach, denoted *error-based*, considers the difference $\hat{\mathbf{u}} - \mathbf{u}_{00} = \left.\frac{\partial g}{\partial \mathbf{w}}\right|_{00} \Delta\mathbf{w}$ at the first step. Next, the authors replace the Newton-Raphson linearization estimate $\hat{\mathbf{u}}$ by the actual value of $\mathbf{u}$ computed in the last step of the NIPALS loop as $\mathbf{u} = \mathbf{Y}\mathbf{c}$. This leads to the following definition of a mismatch $\mathbf{e}$

$$\mathbf{e} = \mathbf{u} - \mathbf{u}_{00} = \left.\frac{\partial g}{\partial \mathbf{w}}\right|_{00} \Delta\mathbf{w} = \mathbf{Z}_w\Delta\mathbf{w} \tag{13}$$

where the matrix $\mathbf{Z}_w = [\frac{\partial g}{\partial \mathbf{w}}|_{00}]$ consists of the partial derivatives $\{\frac{\partial g}{\partial w_i}|_{00}\}_{i=1}^N$. From this mismatch relation $\Delta\mathbf{w}$ can be directly computed by regressing the mismatch $\mathbf{e}$ on $\mathbf{Z}_w$, that is,

$$\Delta\mathbf{w} = (\mathbf{Z}_w^T\mathbf{Z}_w)^{-1}\mathbf{Z}_w^T\mathbf{e}$$

It needs to be mentioned that both S. Wold et al. (1989) and Baffi et al. (1999) considered the case where g is also dependent on variables different than $\mathbf{w}$. More precisely, they considered the quadratic relation $\mathbf{u} = b_0 + b_1\mathbf{t} + b_2\mathbf{t}^2 + \mathbf{h}$. In this case partial derivatives of g with respect to the elements of the vector $\mathbf{b} = [b_0, b_1, b_2]$ can be added to $\mathbf{Z}$ or $\mathbf{Z}_w$ matrix, respectively, and $\Delta\mathbf{b}$ to the vector $\mathbf{v}$.

Finally, the whole nonlinear PLS method consisting of the NIPALS steps and the steps for updating $\mathbf{w}$ is summarized below. The method starts with a random initialization of $\mathbf{u}$. The following steps are repeated until convergence

```
1) $\mathbf{w} = \mathbf{X}^T\mathbf{u}/(\mathbf{u}^T\mathbf{u})$
2) $\|\mathbf{w}\| \rightarrow 1$
3) $\mathbf{t} = \mathbf{X}\mathbf{w}$
4) $\textit{fit} \;\; g(.) \;\; \textit{using} \;\; \mathbf{u}, \mathbf{t}$
5) $\mathbf{u}_{00} = g(\mathbf{t})$
6) $\mathbf{c} = \mathbf{Y}^T\mathbf{u}_{00}/(\mathbf{u}_{00}^T\mathbf{u}_{00})$
7) $\|\mathbf{c}\| \rightarrow 1$
8) $\mathbf{u} = \mathbf{Y}\mathbf{c}$
9) $\textit{compute} \;\; \Delta\mathbf{w}$
10) $\mathbf{w} = \mathbf{w} + \Delta\mathbf{w}$
11) $\textit{go} \;\; \textit{to} \;\; \textit{step} \;\; 2)$
```

Detailed description of these steps can be found in S. Wold et al. (1989); Baffi et al. (1999) or S. Wold (1992).

Discussion and Future Research Directions

Both Type I and Type II represent two general principles for constructing nonlinear PLS models. Many variants of nonlinear PLS can be constructed by applying the wide variety of existing nonlinear methods developed outside of the chemometrics research field. An example is the GIFI approach to nonlinear PLS (Berglund et al., 2001). The method developed in mathematical statistics consists of quantizing the original $\mathbf{X}$ variables into bins. A subsequent representation of each bin with a 1/0 dummy variable then reflects whether the observed continuous value falls into a bin or not (Michailidis & De Leeuw, 1998). The method fits into Type I nonlinear PLS where the quantization represent a mapping of the observed variables into a dummy representation. Similarly, different nonlinear models can be used in Type II nonlinear PLS to represent the inner relation g function. When considering quick and dirty nonlinear PLS, that is, nonlinear PLS without updating of $\mathbf{w}$, almost any nonlinear model can be used. This is different from the situation when $\mathbf{w}$ is updated by any of the rules described in the previous subsection. In this case the Newton-Raphson procedure requires that the nonlinear function g is continuous and differentiable with respect to $\mathbf{w}$. Moreover, a simultaneous mathematical optimization of the inner relation function and computation of new $\mathbf{w}$ from a linearization of the function is not always efficient and convergence to an optimal solution is sometimes not achieved. To relieve these limits Yoshida and Funatsu (1997) and Li, Mei, and Cong (1999) proposed to use a genetic algorithm based optimization technique to simultaneously update g and $\mathbf{w}$. Interestingly, in the work of Yoshida and Funatsu (1997) this was done without a linearization of g. Therefore, a wider set of nonlinear functions to represent g can be used because the criterion of continuity and differentiability can be dropped. In theory, recently extensively used and now-popular concept of kernel learning cannot only by used in the quick and dirty nonlinear PLS scenario, but also in the case where an appropriate update of $\mathbf{w}$ would be advantageous. An example can be a support vector regression model for g. Finally, Hiden et al. (1998) and Searson et al. (2007) extended the concept of genetic programming to represent and estimate the inner relation model itself.

Comparing the two types of nonlinear PLS it is difficult to define the favorable methodology. While Type I is easily implementable, often computationally less demanding and capable to model complex nonlinear relations, it usually leads to a loss of the interpretability of the results with respect to the original data.[3] On the other hand it is not difficult to construct data situations where the Type II approach of keeping latent variables to be linear projections of the original data

[3] One needs to be very careful about proper balancing interpretability and prediction ability of the used model. If the model fits and predicts data poorly interpretation of the observed relations can often be misleading.

may not be adequate. Consider the situation depicted in the left part of Figure 1. Taking any one-dimensional linear projection would create a score vector where both classes would be mixed up with a low level of separability. Thus, a better choice can be an appropriate initial nonlinear mapping of the data. However, consider the same task but add to the problem several dimensions with random variables possessing no separability proprieties with respect to the two classes. In this case an appropriate projection to a lower, optimally two-dimensional original problem space, followed by a nonlinear mapping of the obtained score vectors can be a better choice. In practice a researcher needs to decide about the adequacy of using a particular method based on the problem in hands and requirements like simplicity of the solution, implementation difficulties or interpretation of the results.

Since the work of S. Wold et al. (1989), there have been twenty years of development of the nonlinear PLS concepts, which have resulted in a wide variety of different models. Extensive experimental work has been carried out to compare the studied nonlinear PLS models with the existing concepts of nonlinear modeling developed outside of chemometrics. However, much less research has been carried out to compare the nonlinear PLS models themselves. Within the area of Type II models, Baffi et al. (1999) numerically compared the approach of S. Wold et al. (1989) with their error-based approach. On a synthetic and a real data set they demonstrated better prediction abilities of the error-based approach; in addition, this was achieved with a lower number of latent variables. Similarly, in the area of the Type I models, the kernel PLS method was compared with approaches within the area of kernel learning. The method was proved to be competitive with the classification and regression approaches like SVM, kernel ridge regression or kernel Fischer discriminant analysis (Rosipal et al., 2003; Rosipal & Trejo, 2001). However, a rigorous and thorough experimental and theoretical comparison of different Type I and Type II nonlinear PLS models is lacking.

Finally, there is no restriction on combining the Type I and Type II nonlinear PLS concepts. This can be done by mapping observed data first (Type I) and applying a Type II nonlinear PLS model on transformed data. Would this concept of applying two nonlinear maps be beneficial? In a specific task of smoothing electroencephalographic (EEG) signals it has been observed that better results can be obtained by applying locally weighted PLS regression in a feature space, that is, the nonlinear kernel PLS method was modified by applying a nonlinear PLS model in the mapped data space (Rosipal & Trejo, 2004). Recall that locally weighted PLS regression approximates nonlinearity by a series of local linear models (Cleveland, 1979; Næs & Isaksson, 1992).

Conclusion

There is no doubt that nonlinear PLS modeling represent an important concept for analysis of data sets with nonlinear relationships. However, until recently and in contrast to linear PLS, nonlinear PLS has been mainly used in the chemical data analysis domain. It was the new concept of nonlinear kernel PLS, representing an elegant way of dealing with nonlinear aspects of measured data, which has considerably extended the applicability of nonlinear PLS into a wider area of research fields (Hardoon, Ajanki, Puolamaki, Shawe-Taylor, & Kaski, 2007; Lee, Wu, Huntbatch, & Yang, 2007; Mu, Nandi, & Rangayyan, 2007; Saunders, Hardoon, Shawe-Taylor, & Widmer, 2008; Trejo et al., 2006). The main reason is the fact that the kernel PLS method keeps computational and implementation simplicity of linear PLS while providing a powerful modeling, regression, discrimination or classification tool. Moreover, kernel PLS has been proven to be competitive with the state-of-the-art SVM and other kernel regression and classification methods. However, it would be a big mistake to prefer the kernel method over the other nonlinear PLS approaches, especially

Type II nonlinear PLS. The PLS method projects original data onto a more compact space of latent variables. Among many advantages of such an approach is the ability to analyze and interpret the importance of individual observed variables. The feature which is somehow lost in kernel learning, where usually not easily interpretable nonlinear kernel mapping is applied. The interpretabillity is an important factor not only in chemical data analysis but also in other research and application domains. For example, in an experimental design where many insignificant terms are measured, PLS results can guide the practitioner into more compact experimental settings with a significant cost reduction and without a high risk associated with the 'blind' variables deletion.

The chapter reviewed two main concepts of nonlinear PLS. The aim of the review was to provide fundamental and unifying insights into the understanding of individual methods. Hopefully, this review will draw more attention to developing new nonlinear PLS methods that may overcome drawbacks of the existing approaches.

At the time of writing, I was not aware of a comprehensive software covering all of the described nonlinear PLS methods. A set of Matlab® routines for kernel PLS is available upon request.

Acknowledgments
The author's research was supported by the funds from the Austrian Science Fund FWF project P19857 carried out at the Medical University of Vienna, Vienna, Austria and by the EC project BAMOD, No LSHC-CT-2005-019031, Workpackage 6: Statistical Algorithms, carried out at the Institute of Measurement Science, Slovak Academy of Sciences, Bratislava, Slovakia. Partial support for the production of this review was provided by Pacific Development and Technology, LLC, Palo Alto, CA, USA. I also wish to acknowledge constructive comments by Dr. Leonard J. Trejo.

References

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society*, *68*, 337-404.

Baffi, G., Martin, E., & Morris, A. (1999). Non-linear projection to latent structures revisited: the quadratic PLS algorithm. *Computers and Chemical Engineering*, *23*, 395–411.

Baffi, G., Martin, E., & Morris, A. (2000). Non-linear dynamic projection to latent structures modelling. *Chemometrics and Intelligent Laboratory Systems*, *52*, 5–22.

Barker, M., & Rayens, W. (2003). Partial least squares for discrimination. *Journal of Chemometrics*, *17*, 166–173.

Berglund, A., Kettaneh, N., Uppgåd, L., Wold, S., Bendwell, N., & Cameron, D. (2001). The GIFI approach to non-linear PLS modeling. *Journal of Chemometrics*, *15*, 321–336.

Berglund, A., & Wold, S. (1997). INLR, Implicit Non-linear Latent Variable Regression. *Journal of Chemometrics*, *11*(2), 141–156.

Berglund, A., & Wold, S. (1999). A Serial Extension of Multiblock PLS. *Journal of Chemometrics*, *13*, 461–471.

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

Butler, N., & Denham, M. (2000). The peculiar shrinkage properties of partial least squares regression. *Journal of the Royal Statistical Society: B*, *62*, 585–593.

Cleveland, W. (1979). Robust Locally Weighted Regression and Smoothing Scatterplots. *Journal of the American Statistical Association*, *74*(368), 829–836.

Cristianini, N., & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methodsq.* Cambridge University Press.

De Bie, T., Cristianini, N., & Rosipal, R. (2005). Eigenproblems in Pattern Recognition. In E. Bayro-Corrochano (Ed.), *Handbook of Geometric Computing: Applications in Pattern Recognition, Computer Vision, Neuralcomputing, and Robotics* (pp. 129–170). Springer.

De Jong, S. (1993). SIMPLS: an alternative approach to partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, *18*, 251–263.

Frank, I. (1990). A nonlinear PLS model. *Chemolab*, *8*, 109–119.

Frank, I. (1995). Modern nonlinear regression methods. *Chemometrics and Intelligent Laboratory Systems*, *27*, 1–9.

Frank, I., & Friedman, J. (1993). A Statistical View of Some Chemometrics Regression Tools. *Technometrics*, *35*, 109–147.

Friedman, J. (1991). Multivariate Adaptive Regression Splines (with discussion). *The Annals of Statistics*, *19*, 1–141.

Friedman, J., & Stuetzle, W. (1981). Projection pursuit regression. *Journal of the American Statistical Association*, *76*(376), 817–823.

Girosi, F. (1998). An Equivalence Between Sparse Approximation and Support Vector Machines. *Neural Computation*, *10*(6), 1455–1480.

Girosi, F., Jones, M., & Poggio, T. (1995). Regularization theory and neural networks architectures. *Neural Computation*, *7*(2), 219–269.

Gnanadesikan, R. (1977). *Methods for Statistical Data Analysis of Multivariate Observations.* Wiley, New York.

Goutis, C. (1996). Partial least squares yields shrinkage estimators. *The Annals of Statistics*, *24*, 816–824.

Hardoon, D., Ajanki, A., Puolamaki, K., Shawe-Taylor, J., & Kaski, S. (2007). Information Retrieval by Inferring Implicit Queries from Eye Movements. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics (AISTATS).*

Hasegawa, K., Kimura, T., Miyashita, Y., & Funatsu, K. (1996). Nonlinear Partial Least Squares Modeling of Phenyl Alkylamines with the Monoamine Oxidase Inhibitory Activities. *Journal of Chemical Information and Computer Sciences*, *36*(5), 1025–1029.

Hastie, T., & Tibshirani, R. (1990). *Generalized Additive Models.* Chapman & Hall/CRC.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation* (2nd ed.). Prentice-Hall.

Hiden, H., McKay, B., Willis, M., & Montague, G. (1998). Non-linear partial least squares using genetic programming. In J. Koza (Ed.), *Genetic Programming: Proceedings of the Third Annual Conference.* Morgan Kaufmann.

Höskuldsson, A. (1988). PLS Regression Methods. *Journal of Chemometrics*, *2*, 211–228.

Höskuldsson, A. (1992). Quadratic PLS regression. *Journal of Chemometrics*, *6*(6), 307–334.

Hulland, J. (1999). Use of partial least squares (PLS) in strategic management research: A review of four recent studies. *Strategic Management Journal*, *20*, 195–204.

Ivanov, V. (1976). *The Theory of Approximate Methods and Their Application to the Numerical Solution of Singular Integral Equations.* Nordhoff International.

Kimeldorf, G., & Wahba, G. (1971). Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, *33*, 82–95.

Krämer, N. (2007). An overview on the shrinkage properties of partial least squares regression. *Computational Statistics*, *22*(2), 249–273.

Krämer, N., & Braun, M. (2007). Kernelizing PLS, Degrees of Freedom, and Efficient Model Selection. In Z. Ghahramani (Ed.), *Proceedings of the 24th International Conference on Machine Learning* (pp. 441–448).

Leach, A., & Gillet, V. J. (2003). *An Introduction to Chemoinformatics.* Springer.

Lee, S., Wu, Q., Huntbatch, A., & Yang, G. (2007). Predictive K-PLSR Myocardial Contractility

Modeling with Phase Contrast MR Velocity Mapping. In N. Ayache, S. Ourselin, & A. Maeder (Eds.), *Medical Image Computing and Computer-Assisted Intervention MICCAI 2007* (pp. 866–873). Springer.

Li, T., Mei, H., & Cong, P. (1999). Combining nonlinear PLS with the numeric genetic algorithm for QSAR. *Chemometrics and Intelligent Laboratory Systems*, *45*, 177–184.

Lingjærde, O., & Christophersen, N. (2000). Shrinkage Structure of Partial Least Squares. *Scandinavian Journal of Statistics*, *27*, 459–473.

Lobaugh, N., West, R., & McIntosh, A. (2001). Spatiotemporal analysis of experimental differences in event-related potential data with partial least squares. *Psychophysiology*, *38*, 517–530.

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

Martens, M., & Martens, H. (1986). Partial Least Squares Regression. In J. Piggott (Ed.), *Statistical Procedures in Food Research* (pp. 293–359). Elsevier Applied Science, London.

Mercer, J. (1909). Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London*, *Series A 209*, 415–446.

Michailidis, G., & De Leeuw, J. (1998). The Gifi System of Descriptive Multivariate Analysis. *Statistical Science*, *13*(4), 307-336.

Momma, M. (2005). Efficient Computations via Scalable Sparse Kernel Partial Least Squares and Boosted Latent Features. In *Proceedings of SIGKDD International Conference on Knowledge and Data Mining* (pp. 654–659). Chicago, IL.

Mu, T., Nandi, A., & Rangayyan, R. (2007). Classification of breast masses via nonlinear transformation of features based on a kernel matrix. *Medical and Biological Engineering and Computing*, *45*(8), 769–780.

Næs, T., & Isaksson, T. (1992). Locally Weighted Regression in Diffuse Near-Infrared Transmittance Spectroscopy. *Applied Spectroscopy*, *46*(1), 34–43.

Nilsson, J., Jong, S. de, & Smilde, A. (1997). Multiway Calibration in 3D QSAR. *Journal of Chemometrics*, *11*, 511–524.

Qin, S., & McAvoy, T. (1992). Non-linear PLS modelling using neural networks. *Computers & Chemical Engineering*, *16*(4), 379–391.

Rännar, S., Lindgren, F., Geladi, P., & Wold, S. (1994). A PLS kernel algorithm for data sets with many variables and fewer objects. Part 1: Theory and algorithm. *Chemometrics and Intelligent Laboratory Systems*, *8*, 111–125.

Rosipal, R., & Krämer, N. (2006). Overview and Recent Advances in Partial Least Squares. In C. Saunders, M. Grobelnik, S. Gunn, & J. Shawe-Taylor (Eds.), *Subspace, Latent Structure and Feature Selection Techniques.* Springer.

Rosipal, R., & Trejo, L. (2001). Kernel Partial Least Squares Regression in Reproducing Kernel Hilbert Space. *Journal of Machine Learning Research*, *2*, 97–123.

Rosipal, R., & Trejo, L. (2004). Kernel PLS Estimation of Single-trial Event-related Potentials. *Psychophysiology*, *41*, S94. (Abstracts of The 44th Society for Psychophysiological Research Annual Meeting, Santa Fe, NM)

Rosipal, R., Trejo, L., & Matthews, B. (2003). Kernel PLS-SVC for Linear and Nonlinear Classification. In *Proceedings of the Twentieth International Conference on Machine Learning* (pp. 640–647). Washington, DC.

Saitoh, S. (1997). *Integral Transforms, Reproducing Kernels and Their Applications.* Addison Wesley Longman.

Saunders, C., Hardoon, D., Shawe-Taylor, J., & Widmer, G. (2008). Using String Kernels to Identify Famous Performers from their Playing Style. *Intelligent Data Analysis*, *12*(4), 425–440.

Schölkopf, B., Smola, A., & Müller, K. (1998). Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, *10*, 1299–1319.

Schölkopf, B., & Smola, A. J. (2002). *Learning with Kernels – Support Vector Machines, Regularization, Optimization and Beyond.* The MIT Press.

Searson, D., Willis, M., & Montague, G. (2007). Co-evolution of non-linear PLS model components. *Journal of Chemometrics*, *21*(12), 592–603.

Seber, G. A. F., & Lee, A. J. (2003). *Linear Regression Analysis* (2nd ed.). Wiley-Interscience.

Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel Methods for Pattern Analysis.* Cambridge University Press.

Smola, A., Schölkopf, B., & Müller, K. (1998). The connection between regularization operators and support vector kernels. *Neural Networks*, *11*, 637–649.

Tikhonov, A. (1963). On solving ill-posed problem and method of regularization. *Doklady Akademii Nauk USSR*, *153*, 501–504.

Trejo, L., Rosipal, R., & Matthews, B. (2006). Brain-Computer Interfaces for 1-D and 2-D Cursor Control: Designs using Volitional Control of the EEG Spectrum or Steady-State Visual Evoked Potentials. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *14*(2), 225–229.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory.* Springer-Verlag.

Vapnik, V. (1998). *Statistical Learning Theory.* New York: Wiley.

Vapnik, V., & Chervonenkis, A. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, *16*(2), 264–280.

Vapnik, V., & Chervonenkis, A. (1974). *Theory of Pattern Recognition [in Russian].* Nauka, Moscow. ((German Translation: W.N. Vapnik and A.J. Cherwonenkis (1979). Theorie der Zeichenerkennung. Akademia-Verlag, Berlin))

Wahba, G. (1990). *Splines Models of Observational Data* (Vol. 59). Philadelphia: SIAM.

Wegelin, J. (2000). *A survey of Partial Least Squares (PLS) methods, with emphasis on the two-block case* (Tech. Rep.). Department of Statistics, University of Washington, Seattle.

Wilson, D., Irwin, G., & Lightbody, G. (1997). Nonlinear PLS modeling using radial basis functions. In *American Control Conference.* Albuquerque, New Mexico .

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, H. (1985). Partial least squares. In S. Kotz & N. Johnson (Eds.), *Encyclopedia of the Statistical Sciences* (Vol. 6, pp. 581–591). John Wiley & Sons.

Wold, S. (1992). Nonlinear partial least squares modeling. II. Spline inner relation. *Chemolab*, *14*, 71–84.

Wold, S., Albano, C., Wold, H., Dunn III, W., Edlund, U., Esbensen, K., et al. (1984). Multivariate data analysis in chemistry. In B. Kowalski (Ed.), *Chemometrics. Mathematics and Statistics in Chemistry.* Reidel, Dordrecht.

Wold, S., Kettaneh-Wold, N., & Skagerberg, B. (1989). Nonlinear PLS Modeling. *Chemometrics and Intelligent Laboratory Systems*, *7*, 53–65.

Worsley, K. (1997). An overview and some new developments in the statistical analysis of PET and fMRI data. *Human Brain Mapping*, *5*, 254–258.

Wu, W., Massarat, D., & De Jong, S. (1997). The kernel PCA algorithms for wide data. Part I: theory and algorithms. *Chemometrics and Intelligent Laboratory Systems*, *36*, 165–172.

Yoshida, H., & Funatsu, K. (1997). Optimization of the Inner Relation Function of QPLS Using Genetic Algorithm. *Journal of Chemical Information and Modeling*, *37*(6), 1115–1121.