# Kernel Partial Least Squares for Nonlinear Regression and Dicrimination

Roman Rosipal*

**Abstract**

This paper summarizes recent results on applying the method of partial least squares (PLS) in a reproducing kernel Hilbert space (RKHS). A previously proposed kernel PLS regression model was proven to be competitive with other regularized regression methods in RKHS. The family of nonlinear kernel-based PLS models is extended by considering the kernel PLS method for discrimination. Theoretical and experimental results on a two-class discrimination problem indicate usefulness of the method.

## 1 Introduction

The partial least squares (PLS) method [18, 19] has been a popular modeling, regression and discrimination technique in its domain of origin—Chemometrics. PLS creates orthogonal components (scores, latent variables) by using the existing correlations between different sets of variables (blocks of data) while also keeping most of the variance of both sets. PLS has proved to be useful in situations where the number of observed variables is significantly greater than the number of observations and high multicollinearity among the variables exists. This situation is also quite common in the case of kernel-based learning where the original data are mapped to a high-dimensional feature space corresponding to a reproducing kernel Hilbert space (RKHS). Motivated by the recent results in kernel-based learning and support vector machines [15, 3, 13] the nonlinear kernel-based PLS methodology was proposed in [11]. In this paper we summarize these results and show how the kernel PLS approach can be used for modeling relations between sets of observed variables, regression and discrimination in a feature space defined by the selected nonlinear mapping—kernel function. We further propose a new form of discrimination based on a combination of the kernel PLS method for discrimination with state-of-the-art support vector machine classifier (SVC) [15, 3, 13]. The advantage of using kernel PLS for

*Roman Rosipal, NASA Ames Research Center, Computational Sciences Division, Moffett Field, CA 94035; Department of Theoretical Methods, Slovak Academy of Sciences, Bratislava 842 19, Slovak Republic, E-mail:rrosipal@mail.arc.nasa.gov

dimensionality reduction in comparison to kernel principal components analysis (PCA) [14, 13] is discussed in the case of discrimination problems.

## 2 RHKS - basic definitions

A RKHS is uniquely defined by a positive definite kernel function $K(\mathbf{x}, \mathbf{y})$; i.e. a symmetric function of two variables satisfying the Mercer theorem conditions [7, 3]. Consider $K(.,.)$ to be defined on a compact domain $\mathcal{X} \times \mathcal{X}$; $\mathcal{X} \subset R^N$. The fact that for any such positive definite kernel there exists a unique RKHS is well established by the *Moore-Aronszjan theorem* [1]. The form $K(\mathbf{x}, \mathbf{y})$ has the following *reproducing property*

$$f(\mathbf{y}) = \langle f(\mathbf{x}), K(\mathbf{x}, \mathbf{y}) \rangle_{\mathcal{H}} \quad \forall f \in \mathcal{H} ,$$

where $\langle ., . \rangle_{\mathcal{H}}$ is the scalar product in $\mathcal{H}$. The function $K$ is called *a reproducing kernel* for $\mathcal{H}$.

It follows from Mercer's theorem that each positive definite kernel $K(x, y)$ defined on a compact domain $\mathcal{X} \times \mathcal{X}$ can be written in the form

$$K(x, y) = \sum_{i=1}^{S} \lambda_i \phi_i(x) \phi_i(y) \quad S \leq \infty , \tag{1}$$

where $\{\phi_i(.)\}_{i=1}^{S}$ are the eigenfunctions of the integral operator $\Gamma_K : L_2(\mathcal{X}) \rightarrow L_2(\mathcal{X})$

$$(\Gamma_K f)(\mathbf{x}) = \int_{\mathcal{X}} K(x, y) f(y) dy \quad \forall f \in L_2(\mathcal{X})$$

and $\{\lambda_i > 0\}_{i=1}^{S}$ are the corresponding positive eigenvalues. The sequence $\{\phi_i(.)\}_{i=1}^{S}$ creates an orthonormal basis of $\mathcal{H}$ and we can express any function $f \in \mathcal{H}$ as $f(x) = \sum_{i=1}^{M} a_i \phi_i(x)$ for some $a_i \in \mathcal{R}$. This allows to define a scalar product in $\mathcal{H}$ as

$$\langle f(x), h(x) \rangle_{\mathcal{H}} = \langle \sum_{i=1}^{S} a_i \phi_i(x), \sum_{i=1}^{S} b_i \phi_i(x) \rangle_{\mathcal{H}} \stackrel{\text{def}}{=} \sum_{i=1}^{S} \frac{a_i b_i}{\lambda_i}$$

and the norm

$$\|f\|_{\mathcal{H}}^2 \stackrel{\text{def}}{=} \sum_{i=1}^{S} \frac{a_i^2}{\lambda_i}.$$

Rewriting (1) in the form

$$K(x, y) = \sum_{i=1}^{S} \sqrt{\lambda_i} \phi_i(x) \sqrt{\lambda_i} \phi_i(y) = (\Phi(x).\Phi(y)) = \Phi(x)^T \Phi(y) , \tag{2}$$

it becomes clear that any kernel $K(x, y)$ also corresponds to a canonical (Euclidean) dot product in a possibly high-dimensional space $\mathcal{F}$ where the input data are mapped by

$$\begin{aligned} \Phi: \quad & \mathcal{X} \to \mathcal{F} \\ & \mathbf{x} \to (\sqrt{\lambda_1}\phi_1(x), \sqrt{\lambda_2}\phi_2(x), \ldots, \sqrt{\lambda_S}\phi_S(x)) \ . \end{aligned}$$

The space $\mathcal{F}$ is usually denoted as a *feature space* and $\{\{\sqrt{\lambda_i}\phi_i(x)\}_{i=1}^{S}, x \in \mathcal{X}\}$ as *feature mappings*. The number of basis functions $\phi_i(.)$ also defines the dimensionality of $\mathcal{F}$. It is worth noting that we can also construct a RKHS and a corresponding feature space by choosing a sequence of linearly independent functions (not necessarily orthogonal) $\{\zeta_i(x)\}_{i=1}^{S}$ and positive numbers $\alpha_i$ to define a series (in the case of $S = \infty$ absolutely and uniformly convergent) $K(x, y) = \sum_{i=1}^{S} \alpha_i \zeta_i(x)\zeta_i(y)$.

# 3 Kernel Partial Least Squares

Because the PLS technique is not widely known we first provide a description of linear PLS which will simplify our next description of its nonlinear kernel-based variant [11].

Consider a general setting of the linear PLS algorithm to model the relation between two data sets (blocks of observed variables) $\mathcal{X}$ and $\mathcal{Y}$. Denote by $\mathbf{x} \in \mathcal{X} \subset \mathcal{R}^N$ an $N$-dimensional vector of variables in the first block of data and similarly $\mathbf{y} \in \mathcal{Y} \subset \mathcal{R}^M$ denotes a vector of variables from the second set. PLS models relations between these two blocks by means of latent variables. Observing $n$ data samples from each block of variables, PLS decomposes the $(n \times N)$ matrix of zero-mean variables $\mathbf{X}$ and the $(n \times M)$ matrix of zero-mean variables $\mathbf{Y}$ into the form

$$\begin{aligned} \mathbf{X} &= \mathbf{T}\mathbf{P}^T + \mathbf{F} \\ \mathbf{Y} &= \mathbf{U}\mathbf{Q}^T + \mathbf{G} \end{aligned} \tag{3}$$

where the $\mathbf{T}$, $\mathbf{U}$ are $(n \times p)$ matrices of the extracted $p$ orthogonal components (scores, latent variables), 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 $\mathbf{F}$ and the $(n \times M)$ matrix $\mathbf{G}$ are the matrices of residuals. The PLS method, which in its classical form is based on the nonlinear iterative partial least squares (NIPALS) algorithm [18], finds weight vectors $\mathbf{w}, \mathbf{c}$ such that

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

where $cov(\mathbf{t}, \mathbf{u}) = \mathbf{t}^T\mathbf{u}/n$ denotes the sample covariance between the two score vectors (components). The NIPALS algorithm starts with random initialization

of the Y-score vector $\mathbf{u}$ and repeats a sequence of the following steps until convergence:

| | |
|---|---|
| 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) $\mathbf{u} = \mathbf{Y}\mathbf{c}/(\mathbf{c}^T\mathbf{c})$ |
| 3) $\mathbf{t} = \mathbf{X}\mathbf{w}$ | 6) repeat steps 1. – 5. until convergence |

However, it can be shown [5] that we can directly estimate the weight vector $\mathbf{w}$ as the first eigenvector of the following eigenvalue problem

$$\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}\mathbf{w} = \lambda\mathbf{w} \tag{4}$$

The X-scores $\mathbf{t}$ are than given as

$$\mathbf{t} = \mathbf{X}\mathbf{w} \tag{5}$$

We can similarly derive an eigenvalue problem for the extraction of $\mathbf{t}$,$\mathbf{u}$ and $\mathbf{c}$ estimates [5] and solve one of them for the computation of the other vectors. The nonlinear kernel PLS method is based on mapping the original input data into a high-dimensional feature space $\mathcal{F}$. In this case we usually cannot compute the vectors $\mathbf{w}$ and $\mathbf{c}$. Thus, we need to reformulate the NIPALS algorithm into its kernel variant [6, 11]. Alternatively, we can directly estimate the score vector $\mathbf{t}$ as the first eigenvector of the following eigenvalue problem [5, 9] (this can be easily shown by multiplying both sides of (4) by $\mathbf{X}$ matrix and using (5))

$$\mathbf{X}\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{t} = \lambda\mathbf{t} \tag{6}$$

The Y-scores $\mathbf{u}$ are than estimated as

$$\mathbf{u} = \mathbf{Y}\mathbf{Y}^T\mathbf{t} \tag{7}$$

Now, consider a nonlinear transformation of $\mathbf{x}$ into a feature space $\mathcal{F}$. Using the straightforward connection between a RKHS and $\mathcal{F}$ we have extended the linear PLS model into its nonlinear kernel form [11]. Effectively this extension represents the construction of a linear PLS model in $\mathcal{F}$. Denote $\mathbf{\Phi}$ as the $(n \times S)$ matrix of mapped $\mathcal{X}$-space data $\Phi(\mathbf{x})$ into an $S$-dimensional feature space $\mathcal{F}$. Instead of an explicit mapping of the data we can use property (2) and write

$$\mathbf{\Phi}\mathbf{\Phi}^T = \mathbf{K}$$

where $\mathbf{K}$ represents the $(n \times n)$ *kernel Gram matrix* of the cross dot products between all input data points $\{\Phi(\mathbf{x})\}_{i=1}^n$; i.e. $\mathrm{K}_{ij} = K(x_i, x_j)$ where $K(.,.)$ is a selected kernel function. We can similarly consider a mapping of the second set of variables $\mathbf{y}$ into a feature space $\mathcal{F}_1$ and denote by $\mathbf{\Psi}$ the $(n \times S_1)$ matrix of mapped $\mathcal{Y}$-space data $\Psi(\mathbf{y})$ into an $S_1$-dimensional feature space $\mathcal{F}_1$. We can write

$$\mathbf{\Psi}\mathbf{\Psi}^T = \mathbf{K}_1$$

where $\mathbf{K}_1$ similar to $\mathbf{K}$ represents the $(n \times n)$ *kernel Gram matrix* given by the kernel function $K_1(.,.)$. Using this notation we can reformulate the estimates of $\mathbf{t}$ (6) and $\mathbf{u}$ (7) into its nonlinear kernel variant

$$\begin{aligned} \mathbf{K}\mathbf{K}_1\mathbf{t} &= \lambda\mathbf{t} \\ \mathbf{u} &= \mathbf{K}_1\mathbf{t} \end{aligned} \tag{8}$$

At the beginning of this section we assumed a zero-mean regression model. To centralize the mapped data in a feature space $\mathcal{F}$ we can simply apply the following procedure [14, 11]

$$\mathbf{K} \leftarrow (\mathbf{I} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T)\mathbf{K}(\mathbf{I} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T)$$

where $\mathbf{I}$ is an $n$-dimensional identity matrix and $\mathbf{1}_n$ represent the $(n \times 1)$ vector with elements equal to one. The same is true for $\mathbf{K}_1$.

After the extraction of new score vectors $\mathbf{t}, \mathbf{u}$ the matrices $\mathbf{K}$ and $\mathbf{K}_1$ are deflated by subtracting their rank-one approximations based on $\mathbf{t}$ and $\mathbf{u}$. The different forms of deflation correspond to different forms of PLS (see [17] for a review). The PLS Mode A is based on rank-one deflation of individual block matrices using corresponding score and loading vectors. This approach was originally design by H. Wold [18] to model the relation between the different blocks of data. Because (4) corresponds to the singular value decomposition of the transposed cross-product matrix $\mathbf{X}^T\mathbf{Y}$, computation of all eigenvectors from (4) at once involves implicit rank-one deflation of the overall transposed cross-product matrix. This form of PLS was used in [12] and in accordance with [17] we denote it as PLS-SB. The kernel analog of PLS-SB results from the computation of all eigenvectors of (8) at once. PLS1 (one of the blocks has single variable) and PLS2 (both blocks are multidimensional) as regression methods use a different form of deflation which we describe in the next section.

## 3.1 Kernel PLS Regression

In kernel PLS regression we estimate a linear PLS regression model in a feature space $\mathcal{F}$. The data set $\mathcal{Y}$ represents a set of dependent output variables and in this scenario we do not have reason to consider a nonlinear mapping of the $\mathbf{y}$ variables into a feature space $\mathcal{F}_1$. This simply means that we consider $\mathbf{K}_1 = \mathbf{Y}\mathbf{Y}^T$ and $\mathcal{F}_1$ to be the original Euclidian $\mathcal{R}^M$ space. In agreement with the standard linear PLS model we further assume the score variables $\{\mathbf{t}_i\}_{i=1}^p$ are good predictors of $\mathbf{Y}$. We also assume a linear inner relation between the scores of $\mathbf{t}$ and $\mathbf{u}$; i.e.

$$\mathbf{U} = \mathbf{T}\mathbf{B} + \mathbf{H}$$

where $\mathbf{B}$ is the $(p \times p)$ diagonal matrix and $\mathbf{H}$ denotes the matrix of residuals. In this case, we can rewrite the decomposition of the $\mathbf{Y}$ matrix (3) as

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^T + \mathbf{F} = (\mathbf{T}\mathbf{B} + \mathbf{H})\mathbf{Q}^T + \mathbf{F} = \mathbf{T}\mathbf{B}\mathbf{Q}^T + (\mathbf{H}\mathbf{Q}^T + \mathbf{F})$$

which defines the considered linear PLS regression model

$$\mathbf{Y} = \mathbf{T}\mathbf{C}^T + \mathbf{F}^*$$

where $\mathbf{C}^T = \mathbf{B}\mathbf{Q}^T$ now denotes the $(p \times M)$ matrix of regression coefficients and $\mathbf{F}^* = \mathbf{H}\mathbf{Q}^T + \mathbf{F}$ is the Y-residual matrix.

Taking into account normalized scores $\mathbf{t}$ we define the estimate of the PLS regression model in $\mathcal{F}$ as [11]

$$\hat{\mathbf{Y}} = \mathbf{K}\mathbf{U}(\mathbf{T}^T\mathbf{K}\mathbf{U})^{-1}\mathbf{T}^T\mathbf{Y} = \mathbf{T}\mathbf{T}^T\mathbf{Y} \,. \tag{9}$$

It is worth noting that different scalings of the individual Y-score vectors $\{\mathbf{u}_i\}_{i=1}^p$ do not influence this estimate. The deflation in the case of PLS1 and PLS2 is based on rank-one reduction of the $\mathbf{\Psi}$ and $\mathbf{Y}$ matrices using a new extracted score vector $\mathbf{t}$ at each step. It can be written in the kernel form as follows [11]

$$\mathbf{K} \leftarrow (\mathbf{I} - \mathbf{t}\mathbf{t}^T)\mathbf{K}(\mathbf{I} - \mathbf{t}\mathbf{t}^T) \quad ; \quad \mathbf{K}_1 \leftarrow (\mathbf{I} - \mathbf{t}\mathbf{t}^T)\mathbf{K}_1(\mathbf{I} - \mathbf{t}\mathbf{t}^T)$$

This deflation is based on the fact that we decompose the $\mathbf{\Phi}$ matrix as $\mathbf{\Phi} \leftarrow \mathbf{\Phi} - \mathbf{t}\mathbf{p}^T = \mathbf{\Phi} - \mathbf{t}\mathbf{t}^T\mathbf{\Phi}$, where $\mathbf{p}$ is the vector of loadings corresponding to the extracted component $\mathbf{t}$. Similarly for the $\mathbf{Y}$ matrix we can write $\mathbf{Y} \leftarrow \mathbf{Y} - \mathbf{t}\mathbf{c}^T = \mathbf{Y} - \mathbf{t}\mathbf{t}^T\mathbf{Y}$.

Denote $\mathbf{d}^m = \mathbf{U}(\mathbf{T}^T\mathbf{K}\mathbf{U})^{-1}\mathbf{T}^T\mathbf{Y}^m$ , $m = 1, \ldots, M$ where the $(n \times 1)$ vector $\mathbf{Y}^m$ represents the $m$-th output variable. Then the solution of the kernel PLS regression (9) for the $m$-th output variable can be written as

$$\hat{g}^m(x, \mathbf{d}^m) = \sum_{i=1}^{n} d_i^m K(x, x_i)$$

which agrees with the solution of the regularized form of regression in RKHS given by the Representer theorem [16, 11]. Using equation (9) we may also interpret the kernel PLS model as a linear regression model of the form

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

where $\{t_i(\mathbf{x})\}_{i=1}^p$ are the projections of the data point $\mathbf{x}$ onto the extracted $p$ components and $\mathbf{c}^m = \mathbf{T}^T\mathbf{Y}^m$ is the vector of weights for the $m$-th regression model.

Although the scores $\{\mathbf{t}_i\}_{i=1}^p$ are defined to be vectors in an $S$-dimensional feature space $\mathcal{F}$ we may equally represent the scores to be functions of the original input data $\mathbf{x}$. Thus, the proposed kernel PLS regression technique can be seen as a method of sequential construction of a basis of orthogonal functions $\{t_i(\mathbf{x})\}_{i=1}^p$ which are evaluated at the discretized locations $\{\mathbf{x}_i\}_{i=1}^n$. It is important to note that the scores are extracted such that they increasingly describe overall variance in the input data space and more interestingly also describe the overall variance of the observed output data samples.

## 3.2 Kernel PLS Discrimination

Consider the ordinary least squares regression with outputs $\mathbf{Y}$ to be an indicator vector coding two classes with the values $+1$ and $-1$, respectively. The regression coefficient vector from the least squares solution is than proportional to the linear discriminant analysis (LDA) direction [4]. Moreover, if the number of samples in both classes is equal, the intercepts are the same resulting in the same decision rules. This close connection between LDA and least square regression motivates the use of PLS for discrimination. Moreover, a very close connection between Fisher's LDA (FDA) and PLS-SB methods for multi-class discrimination has been shown in [2]. Using the fact that PLS can be seen as a form of penalized canonical correlations analysis (CCA)[1]

$$[cov(\mathbf{t},\mathbf{u})]^2 = [cov(\mathbf{Xw},\mathbf{Yc})]^2 = var(\mathbf{Xw})[corr(\mathbf{Xw},\mathbf{Yc})]^2 var(\mathbf{Yc})$$

it was suggested [2] to remove the not meaningful $\mathcal{Y}$-space penalty $var(\mathbf{Yc})$ in the PLS discrimination scenario where the Y-block of data is coded in the following way

$$\mathbf{Y} = \begin{pmatrix} \mathbf{1}_{n_1} & \mathbf{0}_{n_1} & \dots & \mathbf{0}_{n_1} \\ \mathbf{0}_{n_2} & \mathbf{1}_{n_2} & \dots & \mathbf{0}_{n_2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{n_g} & \mathbf{0}_{n_g} & \dots & \mathbf{1}_{n_g} \end{pmatrix}$$

Here, $\{n_i\}_{i=1}^{g}$ denotes the number of samples in each class. This modified PLS method is than based on eigen solutions of the between classes scatter matrix which connects this approach to CCA or equivalently to FDA [4, 2]. More interestingly, in the case of two classes the direction of only one PLS component will be identical with the first PLS component found by the PLS1 method with the Y-block represented by the vector with dummy variables coding two classes. However, in the case of PLS1 we can extract additional components each possessing the same similarity with directions computed with CCA on deflated X-block matrices. This provides a more principled dimensionality reduction in comparison to standard PCA based on the criterion of maximum data variation in the $\mathcal{X}$-space alone.

On several classification problems the use of kernel PCA for dimensionality reduction and/or de-noising followed by linear SVC computed on this reduced $\mathcal{X}$-space data representation has shown good results in comparison to nonlinear SVC using the original data representation [13, 14]. However, previous theoretical results suggest to replace the kernel PCA data preprocessing step with the more principled kernel PLS. The advantage of using linear SVC as the follow up step is motivated by the construction of an *optimal separating hyperplane* in the sense of maximizing of the distance to the closest point from either class

[1] In agreement with previous notation $var(.)$ and $corr(.,.)$ denotes the sample variance and correlation, respectively.

| Method | KPLS-SVC | SVC | KFDA | RBF |
|---|---|---|---|---|
| avg. error [%] | $10.6 \pm 0.4$ | $11.5 \pm 0.7$ | $10.8 \pm 0.5$ | $10.8 \pm 0.6$ |

Table 1: Comparison of results between kernel PLS with $\nu$-SVC (KPLS-SVC), C-Support Vector Classifier (SVC), kernel Fisher's LDA (KFDA) and Radial Basis Functions classifier (RBF). The results represents average and standard deviation of the misclassification error using 100 different test sets.

[15, 3, 13]. In comparison to nonlinear kernel FDA [8, 13] this may become more suitable in the situation of non-Gaussian class distribution in a feature space $\mathcal{F}$. Moreover, when the data are not separable the SVC approach provides a way to control the extent of this overlap.

# 4 Experiments

On an example of a two-class discrimination problem (Fig. 1(left)) we demonstrate good results using the proposed combined method of nonlinear kernel-based PLS components extraction and the subsequent linear $\nu$-SVC [13] (denote this method KPLS-SVC). We have used the banana data set obtained via `http://www.first.gmd.de/~raetsch`. This data repository provides the complete 100 partitions of training and testing data used in previous experiments [10, 8, 13]. The repository also provides the value of the Gaussian kernel $K(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2/h)$ width parameter ($h$) found by 5-fold cross-validation (CV) on the first five training data partitions and used by the C-SVC classifier [13] and kernel FDA methods [8], respectively (on this data set the 5-fold CV method results in the same value of the width for both of the methods, $h = 1$). Thus, in all experiments we have used the Gaussian kernel with the same width and we have applied the same CV strategy for the selection of the number of used kernel PLS components and the values of $\nu$ parameter for $\nu$-SVC. The final number of components and $\nu$ value was set to be equal to the median of the five different estimates.

In Table 1 we compare the achieved results with the results using different methods but with identical data partitioning [10, 8, 13]. We see very good results of the proposed KPLS-SVC method. We have further investigated the influence of the number of selected components on the overall accuracy of KPLS-SVC. For the fixed number of components the "optimal" value of the $\nu$ parameter was set using the same CV strategy as described above. Results in Fig. 1(right) show that when more than five PLS components are selected the method provides very consistent, low misclassification rates. Finally, in Fig. 2 we plot the projection of the data from both classes onto the direction found by kernel FDA, using the first component found by kernel PLS and the first component found by kernel

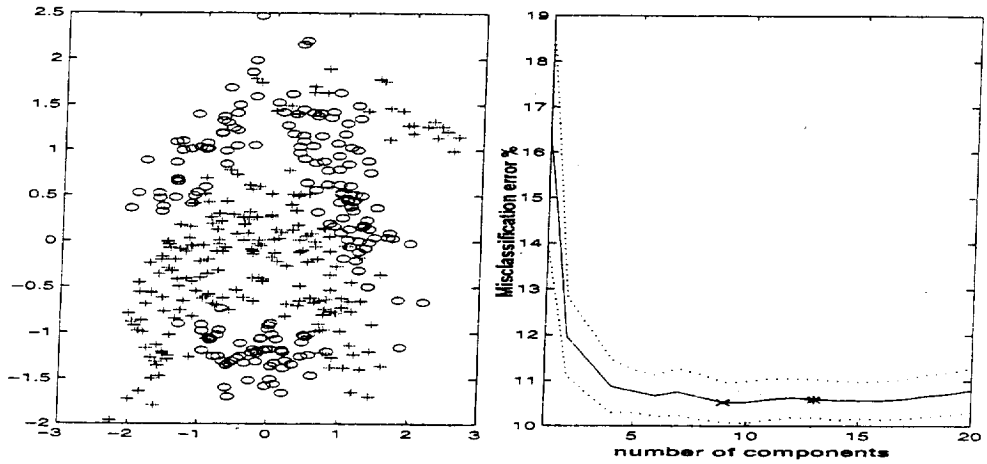


Figure 1: *left:* An example of training patterns (first training data partition was used). *right:* Dependence of the averaged misclassification error on a number of PLS components used. The standard deviation is represented by the dotted lines. For a fixed number of components cross-validation (CV) was used to set $\nu$ parameter for $\nu$-SVC. The cross point indicates the minimum misclassification error achieved. Star indicates a misclassification error when both, number of components and $\nu$ value were set by CV (see Table 1).

PCA, respectively. While we see similarity and nice separation of two classes in the case of kernel FDA and kernel PLS, the kernel PCA method fails to separate the data using the first principal component.

# 5 Conclusions

A summary of the kernel PLS methodology in RKHS was provided. We have shown that the method may be useful for the modeling of existing relations between blocks of variables. With specific arrangement of one of the blocks of variables we may use the technique for nonlinear regression or discrimination problems. We have shown that the proposed technique of combining dimensionality reduction by means of kernel PLS and discrimination of the classes using SVC methodology may result in performance comparable with the previously used classification techniques. Moreover, the projection of the high-dimensional feature space data onto a small number of necessary PLS components resulting in optimal or near optimal discrimination gives rise to the possibility of visual inspection of data separability providing more useful insight into the data structure. Following the theoretical and practical results reported in [2] we also argue that kernel PLS would be preferred to kernel PCA when a feature space dimensionality reduction with respect to data discrimination is employed. The

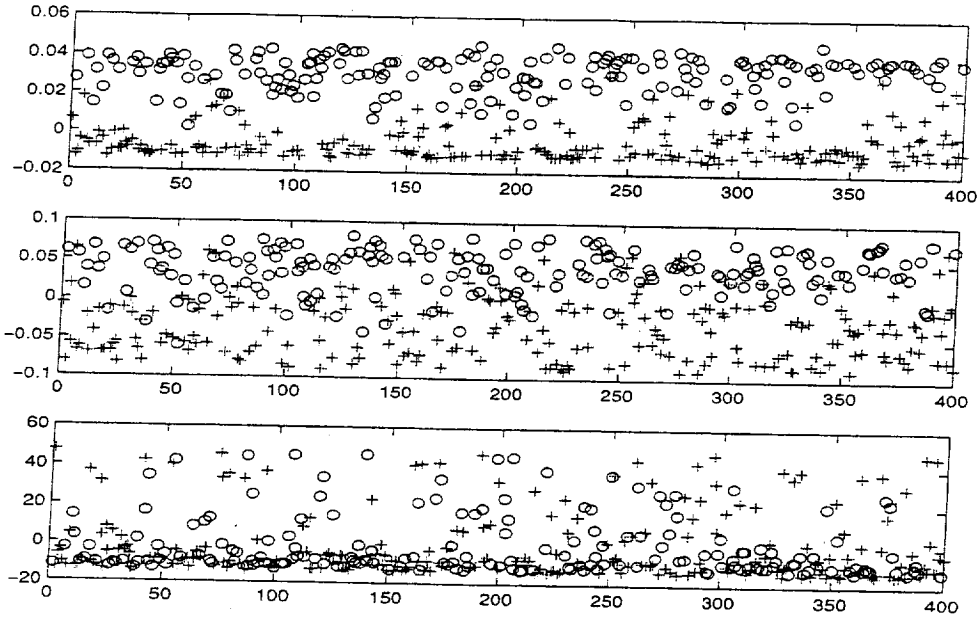


Figure 2: The values of *top*: data projected onto the direction found by kernel Fisher discriminant *middle*: the first kernel PLS component *bottom*: the first kernel PCA principal component. The data depicted in Fig. 1(left) were used.

proposed combination of kernel PLS with SVC can be useful in real world situations where we can expect overlaps among different classes with non-Gaussian distribution.

# References


[1] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

[2] M. Barker and W.S. Rayens. A partial least squares paradigm for discrimination. *to appear Journal of Chemometrics*, 2003.

[3] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines.* Cambridge University Press, 2000.

[4] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning.* Springer, 2001.

[5] A. Höskuldsson. PLS Regression Methods. *Journal of Chemometrics*, 2:211–228, 1988.

[6] P.J. Lewi. Pattern recognition, reflection from a chemometric point of view. *Chemometrics and Intelligent Laboratory Systems*, 28:23–33, 1995.

[7] J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions Royal Society London*, A209:415–446, 1909.

[8] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.R. Müller. Fisher discriminant analysis with kernels. In Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editor, *NNSP IX*, pages 41–48, 1999.

[9] S. Rännar, F. Lindgren, P. Geladi, and S. Wold. A PLS kernel algorithm for data sets with many variables and fewer objects. *Chemometrics and Intelligent Laboratory Systems*, 8:111–125, 1994.

[10] Rätsch, T. Onoda, and K.R. Müller. Soft margins for AdaBoost. *Machine Learning*, 42(3):287–320, 2001.

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

[12] P.D. Sampson, A. P. Streissguth, H.M. Barr, and F.L. Bookstein. Neurobehavioral effects of prenatal alcohol: Part II. Partial Least Squares analysis. *Neurotoxicology and tetralogy*, 11(5):477–491, 1989.

[13] B. Schölkopf and A. J. Smola. *Learning with Kernels -Support Vector Machines, Regularization, Optimization and Beyond.* The MIT Press, 2002.

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

[15] V.N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, New York, 2nd edition, 1998.

[16] G. Wahba. *Splines Models of Observational Data*, volume 59 of *Series in Applied Mathematics.* SIAM, Philadelphia, 1990.

[17] J.A. Wegelin. A survey of Partial Least Squares (PLS) methods, with emphasis on the two-block case. Technical report, Department of Statistics, University of Washington, Seattle, 2000.

[18] H. Wold. Soft Modeling by Latent Variables; the Nonlinear Iterative Partial Least Squares Approach. In J. Gani, editor, *Perspectives in Probability and Statistics*, pages 520–540. Academic Press, London, 1975.

[19] S. Wold, H. Ruhe, H. Wold, and W.J. Dunn III. The collinearity problem in linear regression. The PLS approach to generalized inverse. *SIAM Journal of Scientific and Statistical Computations*, 5:735–743, 1984.