Kernel Partial Least Squares for Nonlinear Regression and Discrimination

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 nonlinear kernel-based PLS regression model has proven to be competitive with other regularized regression methods in RKHS. In this paper the use of kernel PLS for discrimination is discussed. A new methodology for classification is then proposed. This is based on kernel PLS dimensionality reduction of the original data space followed by a support vector classifier. Good results using this method on a two-class classification problem are reported here.

Keywords: kernel-based learning, partial least squares, support vector machines

1 Introduction

The partial least squares (PLS) method [18, 19] has been a popular modeling, regression, discrimination and classification technique in its domain of origin—chemometrics. In its general form PLS creates orthogonal score vectors (components, latent vectors) 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 proven to be useful in situations where the number of observed variables is much 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, 13] the nonlinear kernel-based PLS methodology was proposed in [10]. This paper summarizes these results and show how the kernel PLS approach can be used for modeling relations between sets of observed variables,

[*] 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

regression and discrimination in a feature space defined by the selected nonlinear mapping—kernel function. Further, a new algorithm for classification is proposed. This is based on a combination of the kernel PLS method with a support vector machine for classification (SVC) [15, 13]. The advantage of using kernel PLS for 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 $K(\mathbf{x},\mathbf{y})$; that is, a symmetric function of two variables satisfying the Mercer theorem conditions [6, 13]. 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-Aronszajn theorem* [1]. The form $K(\mathbf{x},\mathbf{y})$ has the following *reproducing property*

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

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 $K(\mathbf{x},\mathbf{y})$ defined on a compact domain $\mathcal{X} \times \mathcal{X}$ can be written in the form

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

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

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

and $\{\lambda_i > 0\}_{i=1}^{S}$ are the corresponding positive eigenvalues. Rewriting (1) in the form

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

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

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

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

3 Kernel Partial Least Squares

Because the PLS technique is not widely known, first, a description of linear PLS is provided. This will simplify the next description of its nonlinear kernel-based variant [10].

Consider a general setting of the 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 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 the relations between these two blocks by means of score vectors. 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 score vectors (components, 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 $\mathbf{F}$ 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 score vectors $\mathbf{t}$ and $\mathbf{u}$. The NIPALS algorithm starts with random initialization of the Y-score vector $\mathbf{u}$ and repeats a sequence of the following steps until convergence:

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

However, it can be shown that the weight vector $\mathbf{w}$ also corresponds to the first eigenvector of the following eigenvalue problem [4]

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

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

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

Similarly, eigenvalue problems for the extraction of $\mathbf{t}$,$\mathbf{u}$ and $\mathbf{c}$ estimates can be derived [4]. The nonlinear kernel PLS method is based on mapping the original input data into a high-dimensional feature space $\mathcal{F}$. In this case the vectors $\mathbf{w}$ and $\mathbf{c}$ cannot be usually computed. Thus, the NIPALS algorithm needs to be reformulated into its kernel variant [5, 10]. Alternatively, the score vectors $\mathbf{t}$ can be directly estimated as the first eigenvector of the following eigenvalue problem [4, 8] (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 then 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 $\mathcal{F}$, Rosipal and Trejo [10] have extended the linear PLS model into its nonlinear kernel form. 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, property (2) can be used resulting in

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

where $\mathbf{K}$ represents the $(n \times n)$ *kernel Gram matrix* of the cross dot products between all input data points $\{\Phi(\mathbf{x}_i)\}_{i=1}^n$; that is, $\mathrm{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ where $K(.,.)$ is a selected kernel function. 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 $\mathcal{F}_1$. Analogous to $\mathbf{K}$ define the $(n \times n)$ kernel Gram matrix $\mathbf{K}_1$

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

given by the kernel function $K_1(.,.)$. Using this notation the estimates of $\mathbf{t}$ (6) and $\mathbf{u}$ (7) can be reformulated into its nonlinear kernel variant

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

Similar to linear PLS, a zero mean nonlinear kernel PLS model is assumed. To centralize the mapped data in a feature space $\mathcal{F}$ the following procedure must be applied [14, 10]

$$\mathbf{K} \leftarrow (\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T)\mathbf{K}(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T) \tag{9}$$

where $\mathbf{I}_n$ is an n-dimensional identity matrix and $\mathbf{1}_n$ represents a $(n \times 1)$ 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 relations 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 a sequence of implicit rank-one deflations of the overall cross-product matrix. Although the weight vectors $\{\mathbf{w}_i\}_{i=1}^p$ will be mutually orthogonal the corresponding score vectors $\{\mathbf{t}_i\}_{i=1}^p$, in general, will not be mutually orthogonal. The same is true for the weight vectors $\{\mathbf{c}_i\}_{i=1}^p$ and the score vectors $\{\mathbf{u}_i\}_{i=1}^p$. This form of PLS was used by Sampson et al. [12] and in accordance with [17] it is denoted 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) generally used as regression methods use a different form of deflation. This is described in the next section.

3.1 Kernel PLS Regression

In kernel PLS regression a linear PLS regression model in a feature space $\mathcal{F}$ is considered. The data set $\mathcal{Y}$ represents a set of dependent output variables and in this scenario there is no reason to nonlinearly map $\mathbf{y}$ variables into a feature space $\mathcal{F}_1$. This simply means that $\mathbf{K}_1 = \mathbf{Y}\mathbf{Y}^T$ and $\mathcal{F}_1$ is the original Euclidian $\mathcal{R}^M$ space. In agreement with the standard linear PLS model it is assumed that the score variables $\{\mathbf{t}_i\}_{i=1}^p$ are good predictors of $\mathbf{Y}$. Further, a linear inner relation between the scores of $\mathbf{t}$ and $\mathbf{u}$ is also assumed; that is,

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

where $\mathbf{B}$ is the $(p \times p)$ diagonal matrix and $\mathbf{H}$ denotes the matrix of residuals. In this case, the decomposition of the $\mathbf{Y}$ matrix (3) can be written 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 $\mathbf{t}$ the estimate of the PLS regression model in $\mathcal{F}$ is defined as [10]

$$\hat{\mathbf{Y}} = \mathbf{K}\mathbf{U}(\mathbf{T}^T\mathbf{K}\mathbf{U})^{-1}\mathbf{T}^T\mathbf{Y} = \mathbf{T}\mathbf{T}^T\mathbf{Y} \tag{10}$$

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 $\mathbf{\Phi}$ 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 [10]

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

This deflation is based on the fact that the $\mathbf{\Phi}$ matrix is decomposed 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 score vector $\mathbf{t}$. Similarly for the $\mathbf{Y}$ matrix it can be written $\mathbf{Y} \leftarrow \mathbf{Y} - \mathbf{t}\mathbf{c}^T = \mathbf{Y} - \mathbf{t}\mathbf{t}^T\mathbf{Y}$. In the case of PLS1 and PLS2 score vectors $\{\mathbf{t}_i\}_{i=1}^p$ are mutually orthogonal. In general, this is not true for $\{\mathbf{u}_i\}_{i=1}^p$ [4].

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 (10) for the m-th output variable can be written as

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

which agrees with the solution of the regularized form of regression in RKHS given by the Representer theorem [16, 10]. Using equation (10) the kernel PLS model can also be interpreted 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 score vectors and $\mathbf{c}^m = \mathbf{T}^T\mathbf{Y}^m$ is the vector of weights for the m-th regression model.

It is worth noting that the score vectors $\{\mathbf{t}_i\}_{i=1}^p$ may be represented as functions of the original input data $\mathbf{x}$. Then, 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 also 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 two different labels representing class membership. The regression coefficient vector from the least squares solution is then proportional to the linear discriminant analysis (LDA) direction [3]. 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 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

$$\mathbf{Y} = \left(\begin{array}{cccc} \mathbf{1}_{n_1} & \mathbf{0}_{n_1} & \ldots & \mathbf{0}_{n_1} \\ \mathbf{0}_{n_2} & \mathbf{1}_{n_2} & \ldots & \mathbf{0}_{n_2} \\ \vdots & \vdots & \ddots & \mathbf{1}_{n_{g-1}} \\ \mathbf{0}_{n_g} & \mathbf{0}_{n_g} & \ldots & \mathbf{0}_{n_g} \end{array} \right)$$

Here, $\{n_i\}_{i=1}^g$ denotes the number of samples in each of the g classes and $\mathbf{0}_{n_i}$ is a $(n_i \times 1)$ vector of all zeros. This modified PLS method is then based on eigen solutions of the between classes scatter matrix which connects this approach to CCA or equivalently to FDA [3, 2]. Interestingly, in the case of two classes the direction of only one PLS-SB score vector will be identical with the first PLS score vector 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 additional score vectors each possessing the same similarity with directions computed with CCA on deflated X-block matrices can be extracted. 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 or de-noising followed by linear SVC computed on the reduced $\mathcal{F}$-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. In comparison to nonlinear kernel FDA [7, 13] this may become more suitable in the situation of non-Gaussian class distribution in a feature space $\mathcal{F}$. 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 [15, 13]. Moreover, when the data are not separable the SVC approach provides a way to control the extent of this overlap. Alternatively, other methods for classification (for example, LDA, logistic regression) applied on extracted PLS score vectors can be considered.

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

Method	KPLS-SVC	C-SVC	KFDA	RBF
avg. error [%]	10.6 ± 0.5	11.5 ± 0.7	10.8 ± 0.5	10.8 ± 0.6

Table 1: Comparison between kernel PLS with ν-SVC (KPLS-SVC), C-SVC, kernel FDA (KFDA) and a radial basis function classifier (RBF) on the Banana data set. The results represent average and standard deviation of the misclassification error using 100 different test sets.

4 Experiments

On an example of a two-class classification problem (Fig. 1(left)) good results are demonstrated using the proposed combined method of nonlinear kernel-based PLS1 score vectors extraction and the subsequent linear ν-SVC [13] (denote this method KPLS-SVC). The used Banana data set was 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 [9, 7]. 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 [13] and kernel FDA methods [7], 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 the Gaussian kernel with the same width has been used and the same CV strategy has been applied for the selection of the number of used kernel PLS score vectors and the values of ν parameter for ν-SVC. The final number of score vectors and ν value was set to be equal to the median of the five different estimates.

Table 1 compares the achieved results with the results using different methods but with identical data partitioning [9, 7]. Good results of the proposed KPLS-SVC method can be seen. Further, the influence of the number of selected score vectors on the overall accuracy of KPLS-SVC has been investigated. For the fixed number of score vectors extracted using the data from the whole training partition the "optimal" value of the ν parameter was set based on the same CV strategy as described above. Results in Fig. 1(right) show that when more than five kernel PLS score vectors are selected the method provides very consistent, low misclassification rates. Finally, Fig. 2 depicts the projection of the data from both classes onto the direction found by kernel FDA, using the first score vector found by kernel PLS and the first principal component found by kernel PCA, respectively. While similarity and nice separation of two classes in the case of kernel FDA and kernel PLS can be seen, the kernel PCA method fails to separate the data using the first principal component.

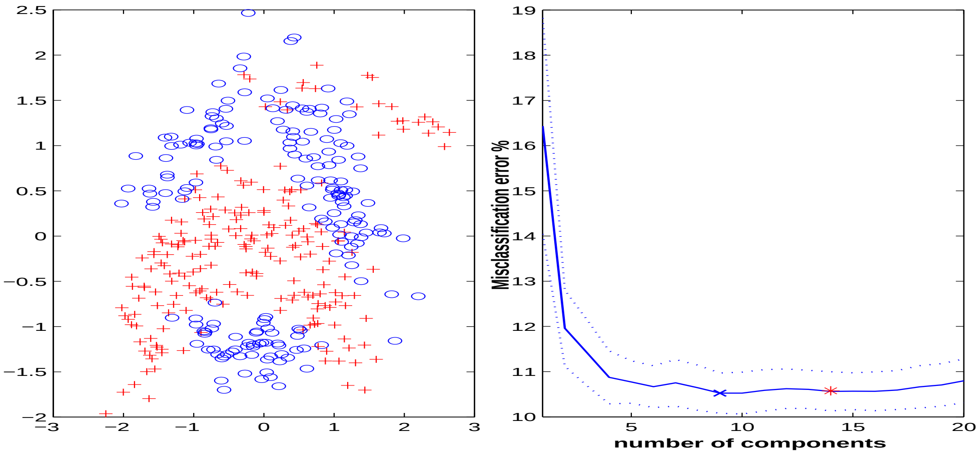

Figure 1: *left:* An example of training patterns (the first training data partition was used). *right*: Dependence of the averaged test set misclassification error on a number of PLS score vectors used. The standard deviation is represented by the dotted lines. For a fixed number of score vectors cross-validation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

5 Conclusions

A summary of the kernel PLS methodology in RKHS was provided. It has been shown that the method may be useful for modeling of the existing relations between blocks of observed variables. With specific arrangement of one of the blocks of variables the technique can be used for nonlinear regression or discrimination problems. It also has been shown that the proposed technique of combining dimensionality reduction by means of kernel PLS with classification using SVC methodology may result in performance comparable with the previously used classification techniques. More detail experimental study confirming this observation was recently published in [11]. Moreover, the projection of the high-dimensional feature space data onto a small number of necessary PLS score vectors 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] it is also argued that kernel PLS would be preferred to kernel PCA when a feature space dimensionality reduction with respect to data discrimination is employed.

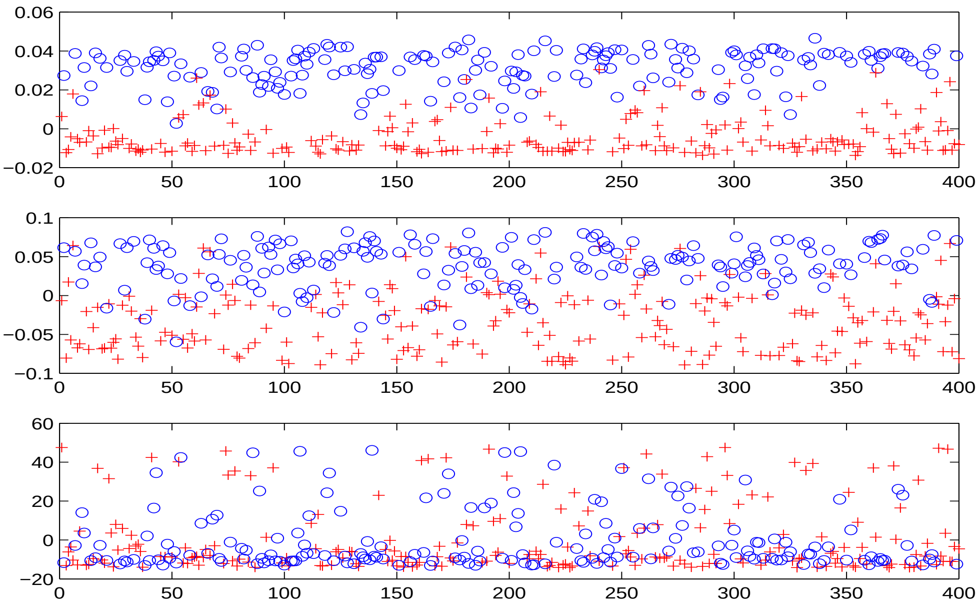

Figure 2: The values of *top*: data projected onto the direction found by kernel Fisher discriminant *middle*: the first kernel PLS score vector *bottom*: the first kernel PCA principal component. The data depicted in Fig. 1(left) were used.

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. Partial least squares for discrimination. *Journal of Chemometrics*, 17:166–173, 2003.

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

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

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

[6] 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.

[7] 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, *Neural Networks for Signal Processing IX*, pages 41–48, 1999.

[8] 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.

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

[10] 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.

[11] R. Rosipal, L.J. Trejo, and B. Matthews. Kernel PLS-SVC for Linear and Nonlinear Classification. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, Washington DC, 2003.

[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. *Statistical Learning Theory.* Wiley, New York, 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.