Nonlinear Kernel-Based Chemometric Tools: a Machine Learning Approach

Roman Rosipal,[1,2] **Leonard J Trejo**,[1] **Bryan Matthews**,[3] **and Kevin Wheeler**[1]

[1]NASA Ames Research Center, Computational Sciences Division, Moffett Field, CA 94035
[2]Department of Theoretical Methods, Slovak Academy of Sciences, Bratislava 842 19, Slovakia
[3]QSS Group Inc., NASA Ames Research Center, Moffett Field, CA 94035

Abstract

This paper provides a short introduction to support vector machines and other nonlinear kernel-based methods recently developed in machine learning research. We describe principles of construction of the nonlinear kernel-based variants of linear methods, which have been widely used in the domain of chemometrics. These include nonlinear kernel forms of the partial least squares, canonical correlation analysis, principal component analysis, principal component regression and ridge regression methods.

1 Introduction

Recently, there has been much interest in the machine learning community in theoretical and practical developments of kernel-based learning. In this paper we address the utility of these methods for applications in chemometrics. Strongly rooted in theoretical principles of statistical learning theory as elaborated by Vapnik [24] and others (see [6, 19] and references therein), several new classification, regression and unsupervised learning algorithms have been developed. This includes methodology called support vector machines (SVM), which have also influenced the construction of other kernel-based methods. As in chemometrics, kernel-based learning is usually associated with situations where the number of observed variables is much more greater than the number of observations and high multicollinearity exists among the variables. This motivated development of nonlinear kernel-based forms of principal components analysis (PCA), principal components regression (PCR), canonical correlation analysis (CCA) and ridge regression. A nonlinear kernel form of partial least squares (PLS) was also proposed [16, 17].

SVM as well as other nonlinear kernel-based methods have been very successfully applied to many problems coming from different domains of research. A few examples of these problems are time-series prediction, text categorization, handwritten digit recognition, face recognition, analysis of electroencephalograms, clustering and classification of microarray gene expression profiles, and DNA or protein analysis. More examples with detailed references can be found in [6, 19, 13]. In the domain of chemistry SVM has been used as an alternative modeling tool for classification and regression problems in quantitative structure property/activity relationships (QSPR/QSAR). Examples of this include studies of pharmaceutical data analysis and drug design [5, 27]. In [2], the SVM method was used to classify polymers by means of their mid-infrared spectra.

A common aspect of all of these new algorithms is the principle of nonlinear transformation of data into a high-dimensional feature space in which linear models are considered. The models are derived based on a straightforward connection between a reproducing kernel Hilbert space (RKHS) and the corresponding feature space representation to which the input

data are mapped. In this paper we review basic principles of this approach. We do not attempt to provide a full detailed description of all described methods but will try to sketch the main principles of their construction. Where appropriate, we will provide references to more complete treatment of the methods.

2 RKHS and Kernel Feature Spaces - 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 conditions of Mercer's theorem [12, 19]. 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 eigen-expansion 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 $\{\phi_i(.)\}_{i=1}^{S}$ and $\{\lambda_i > 0\}_{i=1}^{S}$ are the eigenfunctions and eigenvalues of the corresponding integral operator. By rewriting (1) into the form

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

it becomes clear that any kernel $K(\mathbf{x}, \mathbf{y})$ also corresponds to a canonical (Euclidean) dot product (here denoted as $\langle ., . \rangle$) in a 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 $\{\{\sqrt{\lambda_i}\phi_i(\mathbf{x})\}_{i=1}^{S}, \mathbf{x} \in \mathcal{X}\}$ are referred to as *feature mappings*. The replacement of a canonical dot product by a kernel function (2) is usually called the "kernel trick" and it makes any algorithm depending only on a dot product between its input data computationally tractable even for very high-dimensional spaces $\mathcal{F}$.

To help understand the idea of kernel functions and application of the kernel trick we use the simple and often considered example of the mapping (see Fig. 1)

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

Now, we consider a dot product between two mappings $\Phi(\mathbf{x})$ and $\Phi(\mathbf{y})$ in $\mathcal{F}$

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

where $K(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle^2$ is second order polynomial kernel function. Polynomial kernel functions $K(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle^d$ are then a simple extension of the proposed idea by considering mapping of the original data to a feature space of all monomials of dth order. Another kernel function widely used in practice is the Gaussian kernel $K(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}\right)$, where $\sigma > 0$ determines the width of the Gaussian function. Different kernel functions have been

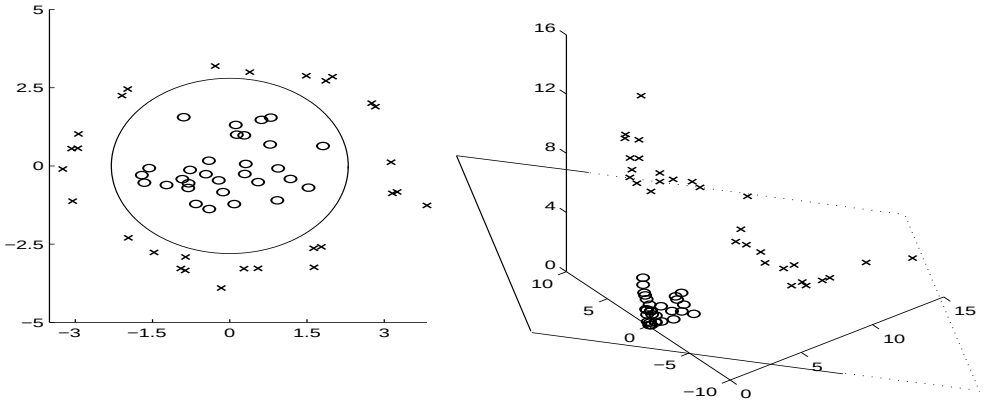

Figure 1: An example of transformation of a nonlinear classification problem with an ellipsoidal decision boundary (*left*) to a problem where a linear hyperplane can be used to separate two classes (*right*). Two classes are defined by circles and crosses, respectively. The original 2D data (*left*)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

also used and constructed [6, 19]. 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 PCA and linear kernel PLS methods [31, 14], previously developed to reduce computational costs in the case where number of input data dimensions (number of features, predictor variables) significantly exceeds number of observed samples, can be considered to belong to the framework of kernel-based learning.

The idea of the kernel trick was used in [4, 24] to construct a nonlinear SVM classifier. In the next section we start with a description of SVM for classification and regression, which significantly motivated development of other nonlinear kernel-based methods described later in the paper.

3 Support Vector Machines

In this section we provide a brief overview of the main principles used in the construction of SVM. In our description of SVM we skip several technical details. Readers interested in the theory of SVM can find detailed derivations in [24, 6, 19, 21, 25].

3.1 Classification

The construction of linear SVM for classification (SVC) of two classes of data has been motivated by the geometric idea of creating a separating hyperplane maximizing the margin; that is, the minimal distance of any data point to the hyperplane. Consider a set of training data $\{\mathbf{x}_i\}_{i=1}^n \in \mathcal{X} \subseteq \mathcal{R}^J$ with a corresponding set of labels $\{y_i\}_{i=1}^n \in \{-1, 1\}$ indicating assignment to one of two classes. We consider the separating hyperplane in a dot product space $\mathcal{X}$

$$\langle \mathbf{w}, \mathbf{x} \rangle + b = 0$$

and the corresponding decision function

$$f(\mathbf{x}) = sgn(\langle \mathbf{w}, \mathbf{x} \rangle + b)$$

where $\mathbf{w} \in \mathcal{X}$ is a vector of weights and $b \in \mathcal{R}$ is an intercept. Further, we consider so-called canonical representation of the separating hyperplane; that is, the form where $\mathbf{w}$ and b are rescaled such that the points closest to the hyperplane satisfy $|\langle \mathbf{w}, \mathbf{x} \rangle + b| = 1$ (see Fig. 2). Now, to construct the separating hyperplane maximizing the margin $1/\|\mathbf{w}\|$ we have to solve

$$\begin{array}{ll} \min_{\mathbf{w},b} & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to} & y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1, \quad i = 1, \ldots, n \end{array} \tag{4}$$

in which using the concept of Lagrange multipliers and primal-dual forms results in a quadratic optimization problem with the consequent final solution

$$f(\mathbf{x}) = sgn(\sum_{i=1}^{n} y_i \alpha_i \langle \mathbf{x}_i, \mathbf{x} \rangle + b) \tag{5}$$

where $\alpha_i \geq 0$ are Lagrange multipliers. The data points $\mathbf{x}_i$ for which $\alpha_i \neq 0$ are called the *support vectors* and they fully define the final decision function $f(\mathbf{x})$. Moreover, similar to (5) the final quadratic optimization problem which has to be solved to obtain $\{\alpha_i\}_{i=1}^n$ can be expressed in terms of the dot product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ between the training data. A direct extension of SVC into its nonlinear form is provided by substituting $\Phi(\mathbf{x}_i) \in \mathcal{F}$ for each $\mathbf{x}_i \in \mathcal{X}$ and considering a kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$. This implies considering the decision function $f(\mathbf{x}) = sgn(\langle \mathbf{w}, \Phi(\mathbf{x}) \rangle + b)$ and solving for a linear SVC model in a feature space $\mathcal{F}$.

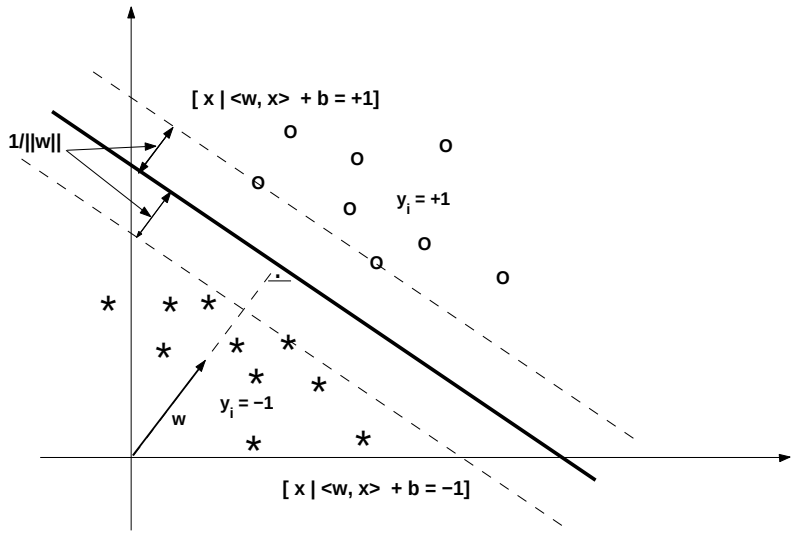

Figure 2: A binary classification problem: the decision boundary (separating hyperplane) $\{\mathbf{x} \in \mathcal{X} | \langle \mathbf{w}, \mathbf{x} \rangle + b = 0\}$ is shown as a solid line. We consider the weight vector $\mathbf{w}$ and b to be rescaled such that the points closest to the hyperplane satisfy $|\langle \mathbf{w}, \mathbf{x} \rangle + b| = 1$. The margin (the distance of the closest point to the hyperplane) is then equal to $1/\|\mathbf{w}\|$. The picture was adapted from [19].

Until now we have considered the case where both classes are linearly separable without errors; that is, so-called hard-margin classification. However, the separating hyperplane may not exist where there is overlap between the classes. Under these conditions to obtain a good generalization of SVC we need to allow some training errors. This can be done by introducing slack variables. Considering a nonlinear SVC model, we can reformulate the problem (4) into

$$\begin{array}{ll} \min_{\mathbf{w},b} & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \\ \text{subject to} & y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n \end{array} \tag{6}$$

where the constant $C > 0$ determines the trade-off between margin maximization and training error minimization. Similar to the previous hard-margin classification case, the final solution is of the form (5) and $\alpha_i \geq 0$ can be obtained by solving a quadratic optimization problem. Interestingly, to obtain a solution of (6) we can consider the optimization problem [26]

$$\min_{\mathbf{w},b} \sum_{i=1}^{n} [1 - y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle + b)]_+ + \eta \|\mathbf{w}\|^2 \tag{7}$$

which represents a familiar regularized paradigm of function estimation. The first term, where the subscript "+" indicates positive part, represents the loss function. This loss function is usually called *hinge* loss and it represents a reasonable choice of loss in the case of two-class classification. The second part of (7) is the penalty term with a penalization coefficient η. This formulation of SVC provides us a direct connection to SVM for regression (SVR).

3.2 Regression

Generally, the SVR problem can be defined as the determination of a function $f(\mathbf{x})$ which approximates an unknown desired regression function and has the form

$$f(\mathbf{x}) = \langle \boldsymbol{\omega}, \Phi(\mathbf{x}) \rangle + b$$

where similar to the classification case b is an unknown bias term and $\boldsymbol{\omega} \in \mathcal{F}$ is a vector of unknown coefficients. The problem of estimating unknown parameters $\boldsymbol{\omega}$ and b can be similar to (7) formulated through the minimization of the regularized risk functional

$$\min_{\boldsymbol{\omega},b} \sum_{i=1}^{n} V(y_i, f(\mathbf{x}_i)) + \eta \|\boldsymbol{\omega}\|^2 \tag{8}$$

where $V(.)$ is a loss function. The nature of the regression problem differs from the classification in the sense that $\{y_i\}_{i=1}^n$ now represents continuous output values and loss functions with the argument of the form $(y_i - f(\mathbf{x}_i))$ are usually considered. The choice of different loss functions in relation to a particular noise distribution was intensively studied by Huber and gave rise to the so-called *robust regression* [10]. Motivated by this result, for the class of densities "close" to the uniform distribution, Vapnik [23] introduced a ϵ-insensitive loss function of the form

$$V(y, f(\mathbf{x})) = |y - f(\mathbf{x})|_\epsilon = \begin{cases} 0 & : \quad |y - f(\mathbf{x})| \leq \epsilon \\ |y - f(\mathbf{x})| - \epsilon & : \quad otherwise \end{cases}$$

and the notion of SVR is usually associated with this type of loss function. It has been shown that the regression estimate that minimizes (8) with ϵ-insensitive loss has the form

$$f(\mathbf{x}) = \sum_{i=1}^{n} (\gamma_i^* - \gamma_i) K(\mathbf{x}_i, \mathbf{x}) + b \tag{9}$$

where $\{\gamma_i, \gamma_i^*\}_{i=1}^n$ are Lagrange multipliers given by the solution of a corresponding quadratic optimization problem [24, 19].

The quadratic loss function $V(y_i, f(\mathbf{x}_i)) = (y_i - f(\mathbf{x}_i))^2$ is also widely used in practice. In this case the regression estimate is of the form

$$f(\mathbf{x}) = \sum_{i=1}^{n} a_i K(\mathbf{x}_i, \mathbf{x}) + b$$

The unknown parameters $\mathbf{a}$ and b are given by the solution of the linear equations

$$(\mathbf{K} + \eta \mathbf{I}_n)\mathbf{a} + \mathbf{1}b = \mathbf{y}$$
$$\mathbf{1}^T\mathbf{a} = 0$$

where $\mathbf{K}$ represents the $(n \times n)$ *Gram matrix* (or *kernel matrix*) of the cross dot products between all mapped 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. $\mathbf{I}_n$ is an n-dimensional identity matrix and $\mathbf{1}_n$ represents a $(n \times 1)$ vector with elements equal to one.

This regression model is known as kernel ridge regression [18], however the same regression model can be derived through the methodology of regularization networks [8, 7] or Gaussian processes [29, 6].

4 Nonlinear Kernel-Based PLS

We consider a general setting of the linear partial least squares (PLS) algorithm to model the relations between two data sets (blocks of observed variables). As in the previous settings we consider $\{\mathbf{x}\}_{i=1}^n \in \mathcal{X} \subseteq \mathcal{R}^J$ to be a set of J-dimensional vectors drawn from the first block of data. We also consider $\{\mathbf{y}\}_{i=1}^n \in \mathcal{Y} \subseteq \mathcal{R}^M$ to be a set of M-dimensional vectors drawn from the second set. PLS models the relations between these two blocks in terms of score vectors. Observing n data samples from each block of variables, PLS decomposes the $(n \times J)$ matrix of zero-mean variables $\mathbf{X}$ and the $(n \times M)$ matrix of zero-mean variables $\mathbf{Y}$ into the form

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{F}$$
$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^T + \mathbf{G}$$

where the $\mathbf{T}$, $\mathbf{U}$ are $(n \times p)$ matrices of the extracted p score vectors (components, latent vectors), the $(J \times p)$ matrix $\mathbf{P}$ and the $(M \times p)$ matrix $\mathbf{Q}$ represent matrices of loadings and the $(n \times J)$ matrix $\mathbf{F}$ and the $(n \times M)$ matrix $\mathbf{G}$ are the matrices of residuals. The PLS method finds weight vectors $\mathbf{w}, \mathbf{c}$ such that

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

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 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 usually be computed. However, Höskuldsson [9] has shown that the score vectors $\mathbf{t}$ can be directly estimated as the first eigenvector of the following eigenvalue problem

$$\mathbf{X}\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{t} = \lambda\mathbf{t} \tag{10}$$

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

$$\mathbf{u} = \mathbf{Y}\mathbf{Y}^T\mathbf{t} \tag{11}$$

As in the construction of SVM, we consider a nonlinear transformation of $\mathbf{x}$ into a feature space $\mathcal{F}$. Effectively this extension represents the construction of a linear PLS model in $\mathcal{F}$. We denote by $\mathbf{\Phi}$ the $(n \times S)$ matrix of mapped $\mathcal{X}$-space data $\{\Phi(\mathbf{x}_i)\}_{i=1}^n$ into an S-dimensional feature space $\mathcal{F}$ and we write

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

where $\mathbf{K}$ denotes the previously defined Gram matrix. Similarly, we can consider a mapping of the second set of variables $\mathbf{y}$ into a feature space $\mathcal{F}_1$. We denote by $\mathbf{\Psi}$ the $(n \times S_1)$ matrix of mapped $\mathcal{Y}$-space data $\{\Psi(\mathbf{y}_i)\}_{i=1}^n$ into an S_1-dimensional feature space $\mathcal{F}_1$. Analogous to $\mathbf{K}$ we define the $(n \times n)$ Gram matrix $\mathbf{K}_1$

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

given by the kernel function $K_1(.,.)$. Using this notation we reformulate the estimates of $\mathbf{t}$ (10) and $\mathbf{u}$ (11) into the corresponding nonlinear kernel variants

$$\begin{aligned} &\mathbf{K}\mathbf{K}_1\mathbf{t} = \lambda\mathbf{t} \\ &\mathbf{u} = \mathbf{K}_1\mathbf{t} \end{aligned} \tag{12}$$

As for linear PLS we assume a zero-mean nonlinear kernel PLS. To centralize the mapped data in a feature space $\mathcal{F}$ without explicit computation of $\mathbf{\Phi}$ we can use the following procedure [20, 16]

$$\mathbf{K} \leftarrow (\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T)\mathbf{K}(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T) \tag{13}$$

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 [28] for a review). Here we discuss PLS1 (one of the blocks has single variable) and PLS2 (both blocks are multidimensional) generally used as regression methods. The deflations in the cases of PLS1 and PLS2 are based on rank-one reduction of the input and output space matrices using the new extracted score vector $\mathbf{t}$ at each step. This is described in the next section.

4.1 Regression

In the case of regression the data set $\mathcal{Y}$ represents a set of dependent output variables. 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 $\mathcal{Y} \subseteq \mathcal{R}^M$ space. In agreement with the standard linear PLS model we assume that the score vectors $\{\mathbf{t}_i\}_{i=1}^p$ are good predictors of $\mathbf{Y}$. Further, we also assume a linear inner relation between the score vectors $\mathbf{t}$ and $\mathbf{u}$.

Taking into account the normalized score vectors $\{\mathbf{t}_i\}_{i=1}^p$ the estimate of the PLS regression model in $\mathcal{F}$ is defined as [16]

$$\hat{\mathbf{Y}} = \mathbf{K}\mathbf{U}(\mathbf{T}^T\mathbf{K}\mathbf{U})^{-1}\mathbf{T}^T\mathbf{Y} = \mathbf{T}\mathbf{T}^T\mathbf{Y} \tag{14}$$

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 of the $\mathbf{\Phi}$ matrix using a new extracted score vector $\mathbf{t}$ at each step has to be reformulated into its kernel form [16]

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

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 $\mathbf{t}$. The deflation of the $\mathbf{Y}$ matrix is given by $\mathbf{Y} \leftarrow \mathbf{Y} - \mathbf{t}\mathbf{c}^T = \mathbf{Y} - \mathbf{t}\mathbf{t}^T\mathbf{Y}$.

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 mth output variable. Then, for the mth output variable, we can write the solution of the kernel PLS regression (14) as

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

which agrees with the solution of the regularized form of regression in RKHS given by the Representer theorem [25]. Using equation (14) we can also interpret the kernel PLS model as a linear regression model of the form

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

where $\{t_i(\mathbf{x})\}_{i=1}^p$ 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 mth regression model.

4.2 Classification

In the case of classification we consider the outputs $\mathbf{Y}$ to be an indicator matrix coding individual g classes with different labels representing class membership

$$\mathbf{Y} = \begin{pmatrix} \mathbf{1}_{n_1} & \mathbf{0}_{n_1} & \ldots & \mathbf{0}_{n_1} \\ \mathbf{0}_{n_2} & \mathbf{1}_{n_2} & \ldots & \mathbf{0}_{n_2} \\ \vdots & \vdots & \ddots & \mathbf{1}_{n_{g-1}} \\ \mathbf{0}_{n_g} & \mathbf{0}_{n_g} & \ldots & \mathbf{0}_{n_g} \end{pmatrix}$$

where $\{n_i\}_{i=1}^g$ denotes the number of samples in each class, $\sum_{i=1}^g n_i = n$ and $\mathbf{0}_{n_i}$ is a $(n_i \times 1)$ vector of all zeros. A close connection between CCA, Fisher's linear discriminant analysis (LDA) and PLS, was recently shown by Barker and Rayens [1]. This motivates us to use the orthonormalized PLS method for discrimination. The kernel variant of this approach transforms (12) into the following equations

$$\begin{aligned} &\mathbf{K}\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{t} = \mathbf{K}\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\mathbf{t} = \lambda\mathbf{t} \\ &\mathbf{u} = \tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^T\mathbf{t} \end{aligned}$$

where $\tilde{\mathbf{Y}} = \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1/2}$ represents a matrix of uncorrelated and normalized original output variables. The use of this modified kernel PLS method is motivated by the fact that $\mathbf{K}\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T$ represents the among-classes sum-of-squares matrix [1]. Thus, unlike principal component analysis (PCA), which has previously served as a dimension reduction step for discrimination problems, orthonormalized PLS extracts score vectors possessing information about the distribution of class centers. In the final classification step we can consider SVC or other methods for classification (for example, LDA, logistic regression) to be applied on extracted kernel PLS score vectors [17].

5 Other Nonlinear Kernel-Based Methods

From the construction of SVM, it is straightforward to derive the nonlinear kernel variants of PCA, PCR and CCA. In this section we briefly review the main principles of these methods.

5.1 Kernel PCA, Multi-layer SVM and Kernel PCR

The PCA problem in high-dimensional feature space $\mathcal{F}$ can be formulated as the diagonalization of an n-sample estimate of the covariance matrix

$$\hat{\mathbf{C}} = \frac{1}{n}\sum_{i=1}^{n}\Phi(\mathbf{x}_i)\Phi(\mathbf{x}_i)^T$$

where again $\{\Phi(\mathbf{x}_i)\}_{i=1}^n$ are centered nonlinear mappings of the input variables $\{\mathbf{x}_i\}_{i=1}^n \in \mathcal{X} \subseteq \mathcal{R}^J$ (we use the same Gram matrix centralization (13)). The diagonalization represents a transformation of the original data to new coordinates defined by orthogonal eigenvectors $\mathbf{V}$. We have to find eigenvalues $\lambda \geq 0$ and non-zero eigenvectors $\mathbf{V} \in \mathcal{F}$ satisfying the eigenvalue equation

$$\lambda\mathbf{V} = \hat{\mathbf{C}}\mathbf{V}$$

Using the fact that all solutions $\mathbf{V}$ with $\lambda \neq 0$ lie in the span of mappings $\Phi(\mathbf{x}_1), \ldots, \Phi(\mathbf{x}_n)$, we can derive the equivalent eigenvalue problem [20] (the linear, kernel form of PCA was described in [31])

$$n\lambda\mathbf{v} = \mathbf{K}\mathbf{v}$$

where $\mathbf{v}$ denotes the column vector (sometimes called dual vector) with coefficients $v_1, \ldots, v_n$ such that

$$\mathbf{V} = \sum_{i=1}^{n} v_i\Phi(\mathbf{x}_i)$$

Normalizing the solutions $\mathbf{V}^k$ corresponding to the non-zero eigenvalues λ_k of the matrix $\mathbf{K}$, translates into the condition $\lambda_k\langle\mathbf{v}^k, \mathbf{v}^k\rangle = 1$ [20]. Finally, we can compute the kth nonlinear principal component of $\mathbf{x}$ as the projection of $\Phi(\mathbf{x})$ onto the eigenvector $\mathbf{V}^k$

$$\beta(\mathbf{x})_k = \langle\mathbf{V}^k, \Phi(\mathbf{x})\rangle = \sum_{i=1}^{n} v_i^k K(\mathbf{x}_i, \mathbf{x}) \tag{15}$$

We then select the first $p < n$ nonlinear principal components, for example, the directions which describe a desired percentage of data variance, and thus work in the p-dimensional subspace of feature space $\mathcal{F}$. This allows us to construct multi-layer support vector machines [20], where a preprocessing layer extracts features for the next regression or classification task. Combining the kernel PCA preprocessing step with SVC yields to multi-layer SVC in the following form [20, 15]

$$f(\mathbf{x}) = sgn(\sum_{i=1}^{n} y_i\alpha_i K_1(\boldsymbol{\beta}(\mathbf{x}_i), \boldsymbol{\beta}(\mathbf{x})) + b)$$

where components of vectors $\boldsymbol{\beta}$ are defined by (15). Using the SVR form (9) we can similarly define multi-layer SVR

$$f(\mathbf{x}) = \sum_{i=1}^{n}(\gamma_i^* - \gamma_i)K_1(\boldsymbol{\beta}(\mathbf{x}_i), \boldsymbol{\beta}(\mathbf{x})) + b$$

In general, we try to select an appropriate mapping of the input data to $\mathcal{F}$ with the aim of "linearizing" the considered regression or classification problem. This simply suggests we

consider a linear kernel $K_1(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle$. We are thus performing linear SVR or SVC on the p-dimensional sub-space of $\mathcal{F}$.

In the case of Gaussian noise the best approximation to the regression provides a least squares method with the quadratic loss function. Considering this loss function and using the first p nonlinear principal components (15) we can construct a nonlinear kernel-based analog of the linear PCR model. This kernel PCR model is then defined as [15]

$$f(\mathbf{x}) = \sum_{k=1}^{p} w_k \beta(\mathbf{x})_k + b = \sum_{k=1}^{p} w_k \sum_{i=1}^{n} v_i^k K(\mathbf{x}_i, \mathbf{x}) + b = \sum_{i=1}^{n} e_i K(\mathbf{x}_i, \mathbf{x}) + b$$

where the weight vector $\mathbf{w} = (w_1, \ldots, w_p)^T$ is the ordinary least squares solution to the regression problem with the predictor variables $\boldsymbol{\beta}(\mathbf{x}) = (\beta(\mathbf{x})_1, \ldots, \beta(\mathbf{x})_p)^T$ and where $\{e_i = \sum_{k=1}^{p} w_k \alpha_i^k\}_{i=1}^{n}$.

5.2 Kernel CCA

CCA is a method of relating two or more sets of variables using the existing relations among the sets. We consider, as in the PLS case, two data sets represented by the zero-mean data matrices $\mathbf{X}$ and $\mathbf{Y}$ of the size $(n \times J)$ and $(n \times M)$, respectively. In the case of $M = 1$, the CCA method reduces to the usual ordinary least squares regression. For $M > 1$, in contrast to the PLS2 method, there is no assumption of causal asymmetry in CCA. In this sense CCA is more closely related to the Mode A PLS method developed for modeling the relation phenomena between different sets of variables rather then for prediction [30, 28]. CCA finds a pair of linear transformations of each block of data with maximal correlation coefficient. This can be formally described as the maximization problem

$$\max_{|\mathbf{r}|=|\mathbf{s}|=1} [corr(\mathbf{Xr}, \mathbf{Ys})]^2 = [corr(\mathbf{Xa}, \mathbf{Yb})]^2 = [cov(\mathbf{Xa}, \mathbf{Yb})]^2 / [var(\mathbf{Xa}) var(\mathbf{Yb})]$$

where as in our previous notation the symbols *corr* and *var* denote the sample correlation and variance, respectively. Estimates of the weight vectors $\mathbf{a}$ and $\mathbf{b}$ are given as the solutions of the following eigenvalue problems [11]

$$(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{X}\mathbf{a} = \lambda\mathbf{a} \quad , \quad (\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}\mathbf{b} = \lambda\mathbf{b}$$

where the eigenvalues λ corresponds to the squared canonical correlation coefficient. As in the PCA and PLS methods the weight vectors $\mathbf{a}$ and $\mathbf{b}$ lie in the span of the observed samples of a particular space. In this case we can write $\mathbf{a} = \mathbf{X}^T\boldsymbol{\alpha}$ and $\mathbf{b} = \mathbf{Y}^T\boldsymbol{\beta}$, where $\boldsymbol{\alpha}, \boldsymbol{\beta}$ are dual vectors (weights). This allows us to formulate a kernel form of regularized CCA which is based on the solution of the following eigenproblems [3, 22]

$$\begin{aligned} (\mathbf{X}\mathbf{X}^T + \eta\mathbf{I}_n)^{-1}\mathbf{Y}\mathbf{Y}^T(\mathbf{Y}\mathbf{Y}^T + \eta\mathbf{I}_n)^{-1}\mathbf{X}\mathbf{X}^T\boldsymbol{\alpha} &= \lambda\boldsymbol{\alpha} \\ (\mathbf{Y}\mathbf{Y}^T + \eta\mathbf{I}_n)^{-1}\mathbf{X}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \eta\mathbf{I}_n)^{-1}\mathbf{Y}\mathbf{Y}^T\boldsymbol{\beta} &= \lambda\boldsymbol{\beta} \end{aligned} \tag{16}$$

where $\eta > 0$ represents a regularization term. The regularization is also a necessary step in the case where a dimension of the data space exceeds the number of samples and we need to deal with the singular sample estimates of the particular covariance matrices [3]. In the case of $M = 1$, the regularized CCA method is equivalent to ridge regression.

Finally, we consider nonlinear kernel mappings of the original data into features spaces $\mathcal{F}$ and $\mathcal{F}_1$. We obtain the nonlinear kernel CCA method by replacing the $\mathbf{X}\mathbf{X}^T$ and $\mathbf{Y}\mathbf{Y}^T$ matrices in (16) with the corresponding centralized Gram matrices $\mathbf{K}$ and $\mathbf{K}_1$, respectively. The centralization is again given by (13).

6 Conclusions

We summarized the construction of support vector machines and several related nonlinear kernel-based learning methods. For each method—SVM, kernel PLS, kernel PCA, kernel PCR, kernel ridge regression and kernel CCA—we described the main theoretical principles so as to provide guidance into this research area for readers unfamiliar with these developments. We also explained the close connections among all of these methods.

We focused our attention on the nonlinear kernel-based extensions of several regression and classification methods widely used in the domain of chemistry and chemometrics. These extensions may be very useful in the case where nonlinear relations between sets of data exist. If nonlinear relations cannot be adequately approximated using linear models, the nonlinear kernel-based methods offer a means to provide more precise results; for example, in the sense of prediction accuracy or lower classification errors. However, there is also a potential drawback associated with these new approaches—loss of interpretability. The interpretation of the final kernel models constructed in a feature space may become difficult with respect to the original, observed data. This remains an open area for further research which should profit from merging research in the machine learning and chemometrics communities.

Acknowledgments. Many thanks to Tijl De Bie for useful discussions on kernel CCA topics. This work was supported by funding from the NASA CICT/ITSR and IS/HCC programs.

References

[1] M. Barker and W.S. Rayens. Partial least squares for discrimination. *Journal of Chemometrics*, 17:166–173, 2003.

[2] A.I. Belousov, S.A. Verzakov, and J. von Frese. Applicational aspects of support vector machines. *Journal of Chemometrics*, 16(8–10):482–489, 2002.

[3] T. De Bie, N. Cristianini, and R. Rosipal. *Handbook of Computational Geometry for Pattern Recognition, Computer Vision, Neurocomputing and Robotics*, chapter Eigenproblems in Pattern Recognition. Springer Verlag, 2003 (in preparation).

[4] B.E. Boser, I.M. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proceeding of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh, 1992. ACM Press.

[5] R. Burbidge, M. Trotter, B. Buxton, and S. Holden. Drug design by machine learning: support vector machines for pharmaceutical data analysis. *Computers & Chemistry*, 26(1):5–14, 2001.

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

[7] T. Evgeniou, M. Pontil, and T. Poggio. Regularization Networks and Support Vector Machines. *Advances in Computational Mathematics*, 13:1–50, 2000.

[8] F. Girosi, M. Jones, and T. Poggio. Regularization Theory and Neural Network Architectures. *Neural Computation*, 7:219–269, 1995.

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

[10] P.J. Huber. *Robust Statistics*. John Wiley and Sons, New York, 1981.

[11] K.V. Mardia, J.T. Kent, and J.M. Bibby. *Multivariate Analysis*. Academic Press, 1997.

[12] 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.

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

[14] S. Rännar, F. Lindgren, P. Geladi, and S. Wold. 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, 1994.

[15] R. Rosipal, M. Girolami, L.J. Trejo, and A. Cichocki. Kernel PCA for Feature Extraction and De-Noising in Nonlinear Regression. *Neural Computing & Applications*, 10(3):231–243, 2001.

[16] 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.

[17] 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.

[18] C. Saunders, A. Gammerman, and V. Vovk. Ridge Regression Learning Algorithm in Dual Variables. In *Proceedings of the 15th International Conference on Machine Learning*, pages 515–521, Madison, Wisconsin, 1998.

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

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

[21] A.J. Smola and B. Schölkopf. A Tutorial on Support Vector Regression. Technical Report NC-TR-1998-030, NeuroColt, Royal Holloway College, 1998.

[22] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002.

[23] V.N. Vapnik. *Estimation of Dependencies Based on Empirical Data*. Springer-Verlag, Berlin, 1982.

[24] V.N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

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

[26] G. Wahba, Y. Lin, and H. Zhang. GACV for Support Vector Machines. In A.J. Smola, P.J. Bartlett, B. Schölkopf, and D. Schuurmans, editor, *Advances in Large Margin Classifiers*, pages 297–309. MIT Press, 2000.

[27] M.K. Warmuth, J. Liao, G. Rätsch, M. Mathieson, S. Putta, and C. Lemmen. Active Learning with Support Vector Machines in the Drug Discovery Process. *Journal of Chemical Information Sciences*, 43(2):667–673, 2003.

[28] 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.

[29] C.K.I. Williams. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In M.I. Jordan, editor, *Learning and Inference in Graphical Models*, pages 599–621. Kluwer, 1998.

[30] H. Wold. Partial least squares. In S. Kotz and N.L. Johnson, editors, *"Encyclopedia of the Statistical Sciences"*, pages 581–591. Wiley, 1985.

[31] W. Wu, D.L. Massarat, and S. de Jong. The kernel PCA algorithms for wide data. Part I: theory and algorithms. *Chemometrics and Intelligent Laboratory Systems*, 36:165–172, 1997.