Available online at www.sciencedirect.com
SCIENCE DIRECT®
Neurocomputing ▮ (▮▮▮▮) ▮▮▮–▮▮▮
NEUROCOMPUTING
www.elsevier.com/locate/neucom

Subset based least squares subspace regression in RKHS

L. Hoegaerts*, J.A.K. Suykens, J. Vandewalle, B. De Moor

Katholieke Universiteit Leuven, Department of Electrical Engineering, ESAT-SCD-SISTA Kasteelpark Arenberg 10, B-3001 Leuven (Heverlee), Belgium

Received 27 October 2003; received in revised form 19 February 2004; accepted 23 April 2004

Abstract

Kernel based methods suffer from exceeding time and memory requirements when applied on large datasets since the involved optimization problems typically scale polynomially in the number of data samples. As a remedy, some least squares methods on one hand only reduce the number of parameters (for fast training), on the other hand only work on a reduced set (for fast evaluation). Departing from the Nyström based feature approximation, via the fixed-size LS-SVM model, we propose a general regression framework, based on restriction of the search space to a subspace and a particular choice of basis vectors in feature space. In the general model both reduction aspects are unified and become explicit model choices. This allows to accommodate kernel Partial Least Squares and kernel Canonical Correlation analysis for regression with a sparse representation, which makes them applicable to large data sets, with little loss in accuracy.

Keywords: Kernel methods; Regression; Reduced set methods; Subspaces; Eigenspaces; Reproducing kernel Hilbert space; Nyström approximation; Fixed-size least squares support vector machine; Kernel principal component analysis; Kernel partial least squares; Kernel canonical correlation analysis

*Corresponding author. Tel.: +32-16-32-85-40; fax: +32-16-32-19-70.

E-mail addresses: luc.hoegaerts@esat.kuleuven.ac.be (L. Hoegaerts), johan.suykens@esat.kuleuven.ac.be (J.A.K. Suykens).

0925-2312/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2004.04.013

1. Introduction

Over the last years one can see many learning algorithms being transferred to a kernel representation [28,31]. The benefit lies in the fact that nonlinearity can be allowed, while avoiding to solve a nonlinear optimization problem. In this paper we focus on least squares regression models in the kernel context. By means of a nonlinear map into a reproducing kernel Hilbert space (RKHS) [34] the data are projected to a high-dimensional space.

Kernel methods typically operate in this RKHS. The high-dimensionality which could pose problems for proper parameter estimation is circumvented by the kernel trick, which brings the dimensionality to the number of training instances n and at the same time allows excellent performance in classification and regression tasks. Yet, for large data sets this dimensionality in n means a serious bottleneck, since the corresponding training methods scale polynomially in n. Downsizing the system in dimensions to size $m \ll n$ is therefore needed. On the one hand, low-rank approximations (e.g. [7,35]) offer reduction to smaller $m \times m$ matrices. On the other hand reduced set methods work with a thin tall $n \times m$ system (e.g. [17,18,25,30,34]). This work connects at the latter perspective.

We start from the Nyström approximation, delivering a direct approximation of features, which, applied in an ordinary least squares model, leads to kernel principal component regression (KPCR) and fixed-size LS-SVMs [31]. Compared to KPCR, FS-LSSVM also only needs a small number of parameters, but additionally the regression is done in the primal space leading to a *sparse representation*, unlike estimation in the dual space as done in Gaussian Processes without having a sparse representation [35,36]. Such a parsimonious model is highly desirable when dealing with large data sets since it implies a substantial reduction in training and evaluation time.

By formalizing the structure of the FS-LSSVM model, we then obtain a least squares regression framework in which (i) we explicitly restrict the regression coefficients to a subspace and (ii) we express the subspace in the basis formed by mapped training data points. These model constraints respectively allow to control the number of regression parameters and the number of kernels. Our general model formulation introduces kernels in a natural manner into the model, yields a complementary, unifying viewpoint on some other kernel methods and it comprises a class of models. We especially focus on the expression in a *subset* $m \ll n$ of features of the RKHS. This scheme of LS subspace regression delivers a linear system that consequently only needs polynomial training times in m.

Like KPCR was extended by FS-LSSVM with a sparse representation, we now accommodate here also kernel partial least squares (KPLS) (together with some variants [11]) and kernel canonical correlation analysis (KCCA) with a sparse representation in a natural and efficient manner (different from [19], where sparsification is induced via a multi-step adaptation of the algorithm with extra computational burdens). Hereby we make their application possible for large scale data sets. In the subspace regression model, the extension consists of using alternative eigenspaces constructed by optimization of other (co)variance criteria. In some examples we finally show that these models perform well with little loss of accuracy and can effectively manage large data sets.

This paper is organized as follows. In Section 2 we present some minimal background on kernel methods in relation to reproducing kernel Hilbert spaces. In Section 3 we deal with the Nyström approximation for features to recognize three least squares models. In Section 4 we introduce subset based least squares subspace restricted regression in feature space. In Section 5 we give an overview of some alternative eigenspaces with which we can endow the model. In Section 6 we illustrate the different methods by some results on an artificial and a three real-world datasets, including a large scale example.

2. Reproducing kernel Hilbert space

The central idea for kernel algorithms within the learning theory context is to change the representation of a data point into a higher-dimensional mapping in a reproducing kernel Hilbert space (RKHS) H_k by means of a kernel function. When appropriately chosen, the kernel function with arguments in the original space corresponds to a dot product with arguments in the RKHS. This allows to circumvent working explicitly with the new representation as long as one can express the computation in the RKHS as inner products.

Assume data $\{\{(\mathbf{x}_i, y_i)\}_{i=1}^n \in \mathbb{R}^p \times \mathbb{R}\}$ have been given. A kernel k provides a similarity measure between pairs of data points

$$k : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R} : (\mathbf{x}_i, \mathbf{x}_j) \mapsto k(\mathbf{x}_i, \mathbf{x}_j). \tag{1}$$

Once a kernel is chosen, one can associate to each $\mathbf{x} \in \mathbb{R}^p$ a mapping $\varphi : \mathbb{R}^p \to H_k : \mathbf{x} \mapsto k(\mathbf{x}, \cdot)$, which can be evaluated at $\mathbf{x}'$ to give $\varphi(\mathbf{x})(\mathbf{x}') = k(\mathbf{x}, \mathbf{x}')$. One obtains a RKHS on $\mathbb{R}^p$ under the condition that the kernel is positive definite [1]. Remark that the kernel function coincides with an inner product function and $\langle k(\mathbf{x}, \cdot), k(\mathbf{x}', \cdot)\rangle = k(\mathbf{x}, \mathbf{x}')$, so it reproduces itself. It is also called a representer of f at $\mathbf{x}$ because $f(\mathbf{x}) = \langle f, k(\mathbf{x}, \cdot)\rangle$.

The RKHS has the property that every evaluation operator and norm of any element in H_k is bounded [34]. This makes the elements of a RKHS well-suited to interpolate pointwise known functions that must be smooth. In the context of regularization one tries to do this by minimizing a pointwise cost function $c(\cdot)$ over data and monotonic smoothness function $g(\cdot)$:

$$\min_{f \in H_k} c(\mathbf{x}_i, y_i, f(\mathbf{x}_i)) + g(\|f\|). \tag{2}$$

The representer theorem [15] states that the solution is constrained to the subspace spanned by the mapped data points:

$$f(\mathbf{x}) = \sum_{i=1}^{n} w_i \varphi(\mathbf{x}_i)(\mathbf{x}) = \sum_{i=1}^{n} w_i k(\mathbf{x}_i, \mathbf{x}). \tag{3}$$

Thus the solution can always be expanded as a linear function of dot products. The w_i coefficients are typically found by solving the optimization problem that involves the specific regularization functional.

The Mercer–Hilbert–Schmidt theorem reveals more about the nature of φ by stating that for each positive definite kernel there exists an orthonormal set $\{\phi_i\}_{i=1}^d$ with non-negative λ_i such that we have following spectral decomposition:

$$k(\mathbf{x},\mathbf{x}') = \sum_{i=1}^{d} \lambda_i \phi_i(\mathbf{x})\phi_i(\mathbf{x}') = \langle \varphi(\mathbf{x}), \varphi(\mathbf{x}')\rangle, \tag{4}$$

where $d \leqslant \infty$ is the dimension of the RKHS, and λ_i and ϕ_i are the eigenvalues and eigenvectors of the kernel operator, defined by the integral equation

$$\int k(\mathbf{x},\mathbf{x}')\phi_i(\mathbf{x})p(\mathbf{x})\,\mathrm{d}x = \lambda_i\phi_i(\mathbf{x}'), \tag{5}$$

on $L_2(\mathbb{R}^p)$. This allows to formulate the inner product in terms of expansion coefficients:

$$\langle f, g\rangle = \left\langle \sum_{i=1}^{d} a_i\phi_i, \sum_{j=1}^{d} b_j\phi_j \right\rangle = \sum_{i=1}^{d} \frac{a_i b_i}{\lambda_i}, \tag{6}$$

where $\langle \phi_i, \phi_j\rangle = \delta_{ij}/\lambda_i$. A proper scaling of the basis vectors ϕ_i with factor $\sqrt{\lambda_i}$ will transform the inner product to its most simple canonical form of an Euclidean dot product so that $k(\mathbf{x},\mathbf{x}') = \varphi(\mathbf{x})^{\mathrm{T}}\varphi(\mathbf{x}')$. Furthermore one can express each feature $\varphi(\mathbf{x})$ in this basis so that the φ mapping can be identified with a $d \times 1$ *feature* vector

$$\varphi(\mathbf{x}) = [\sqrt{\lambda_1}\phi_1(\mathbf{x}) \quad \sqrt{\lambda_2}\phi_2(\mathbf{x}) \quad \ldots \quad \sqrt{\lambda_d}\phi_d(\mathbf{x})]^{\mathrm{T}}. \tag{7}$$

For regression purposes, just like one builds up a $n \times p$ data matrix X from the $\mathbf{x}_i$ inputs, one constructs with the mapped data points $\varphi(\mathbf{x})$, an $n \times d$ *feature matrix*:

$$\Phi = \begin{pmatrix} \varphi^{\mathrm{T}}(\mathbf{x}_1) \\ \varphi^{\mathrm{T}}(\mathbf{x}_2) \\ \vdots \\ \varphi^{\mathrm{T}}(\mathbf{x}_n) \end{pmatrix} = \begin{pmatrix} \sqrt{\lambda_1}\phi_1(\mathbf{x}_1) & \sqrt{\lambda_2}\phi_2(\mathbf{x}_1) & \ldots & \sqrt{\lambda_n}\phi_d(\mathbf{x}_1) \\ \sqrt{\lambda_1}\phi_1(\mathbf{x}_2) & \sqrt{\lambda_2}\phi_2(\mathbf{x}_2) & \ldots & \sqrt{\lambda_n}\phi_d(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \sqrt{\lambda_1}\phi_1(\mathbf{x}_n) & \sqrt{\lambda_2}\phi_2(\mathbf{x}_n) & \ldots & \sqrt{\lambda_n}\phi_d(\mathbf{x}_n) \end{pmatrix}. \tag{8}$$

It is a difficulty that in general the elements of this feature matrix Φ are unknown because the explicit expression for φ is not available. The common approach to deal with this issue is then to make use of the so-called kernel trick [28]. The need for a direct expression is typically avoided by an ad hoc substitution such that scalar products are formed and consequently kernels come into play. Thus, the element $\varphi(\mathbf{x})$ may be not explicitly known, but the projection on any other $\varphi(\mathbf{x}')$ is simply $k(\mathbf{x},\mathbf{x}')$, and then the unknown elements are eliminated.

Here we follow another approach in which we start from an approximate explicit expression for $\varphi(\mathbf{x})$, via the Nyström method. We will show how it leads to a least squares regression model, FS-LSSVM, with a downsized number of parameters and a sparse kernel expansion. Such parsimonious models are necessary when dealing with large data sets.

3. The Nyström approximation

The Nyström method dates back from the late 1920s [21]. It offers an approximate solution to integral [3,23]. It became recently of interest in the kernel machine learning community via Gaussian processes [35,36] and recognized as implicitly present in the projection of features in the eigenspace produced by Kernel principal component analysis (KPCA), introduced in [29]. More specifically, the technique yields a simple approximation of the features for a predetermined subset of $m \ll n$ training points. These are then used on their turn as an interpolative, but quite accurate, formula for the features in the remaining training points.

3.1. Approximation of eigenfunctions

One can discretize the integral of Eq. (5) on a finite set of evaluation points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, with simple equal weighing, yielding a system of equations:

$$\frac{1}{n}\sum_{j=1}^{n} k(\mathbf{x}, \mathbf{x}_j)\phi_i(\mathbf{x}_j) = \lambda_i^{(n)}\phi_i^{(n)}(\mathbf{x}), \tag{9}$$

which can be structured as a matrix eigenvalue problem:

$$KU_n = U_n\Lambda_n, \tag{10}$$

where $K_{ij} = \langle\varphi(\mathbf{x}_i), \varphi(\mathbf{x}_j)\rangle = k(\mathbf{x}_i, \mathbf{x}_j)$ are the elements of the kernel Gram matrix, U_n a $n \times n$ matrix of eigenvectors of K and Λ_n a $n \times n$ diagonal matrix of non-negative eigenvalues in non-increasing order. (We emphasized the dimensions of matrices by a subscript and for vectors it is indicated in the superscript, between brackets to avoid confusion with powers.) Expression (9) delivers direct approximations of the eigenvalues and eigenfunctions for the $\{\mathbf{x}_j\}_{j=1}^{n}$ points:

$$\phi_i(\mathbf{x}_j) \approx \sqrt{n}\,\mathbf{u}_{ji}^{(n)}, \tag{11}$$

$$\lambda_i \approx \frac{1}{n}\,\lambda_i^{(n)}. \tag{12}$$

It was a key observation of Nyström to backsubstitute these approximations in (9) to obtain an approximate of an eigenfunction evaluation in new points $\mathbf{x}'$:

$$\phi_i(\mathbf{x}') = \frac{\sqrt{n}}{\lambda_i^{(n)}}\sum_{j=1}^{n} k(\mathbf{x}', \mathbf{x}_j)\mathbf{u}_{ji}^{(n)} = \frac{\sqrt{n}}{\lambda_i^{(n)}}(\mathbf{k}^{(n)}(\mathbf{x}'))^{\mathrm{T}}\mathbf{u}_i^{(n)}, \tag{13}$$

where $\mathbf{k}^{(n)}(\mathbf{x}) = \Phi\varphi(\mathbf{x}) = [k(\mathbf{x}_1, \mathbf{x}), k(\mathbf{x}_2, \mathbf{x}), \ldots, k(\mathbf{x}_n, \mathbf{x})]^{\mathrm{T}}$. As can be seen from (7), the entries of features in a RKHS match the eigenfunctions, apart from a scaling factor, so an approximate eigenfunction yields us an approximate feature.

3.2. Feature approximation based on the complete training data set

Maximally n approximate eigenfunctions ϕ_i can be computed, since we have only information in the given data points. The Nyström approximation allows to obtain an explicit expression for the entries of the approximated feature vector:

$$\begin{aligned}\varphi_i(\mathbf{x}) &= \sqrt{\lambda_i}\;\phi_i(\mathbf{x}) && (14)\\ &= (\lambda_i^{(n)})^{-1/2}\sum_{j=1}^{n} k(\mathbf{x},\mathbf{x}_j)\mathbf{u}_{ji}^{(n)} && (15)\\ &= (\lambda_i^{(n)})^{-1/2}(\mathbf{k}^{(n)}(\mathbf{x}))^{\mathrm{T}}\mathbf{u}_i^{(n)}. && (16)\end{aligned}$$

The approximation to a feature becomes a $n \times 1$ vector:

$$\varphi(\mathbf{x}) \approx \Lambda_n^{-1/2} U_n^{\mathrm{T}} \mathbf{k}^{(n)}(\mathbf{x}). \tag{17}$$

At the same time we obtain also a $n \times n$ matrix approximation Φ_n of the $n \times d$ *regressor matrix* (8):

$$\Phi = \begin{pmatrix} \varphi^{\mathrm{T}}(\mathbf{x}_1) \\ \varphi^{\mathrm{T}}(\mathbf{x}_2) \\ \vdots \\ \varphi^{\mathrm{T}}(\mathbf{x}_n) \end{pmatrix} \approx K U_n \Lambda_n^{-1/2} =: \Phi_n. \tag{18}$$

If we use this matrix in a least squares setting, we obtain a linear model in feature space. More specifically we are then performing PCR, which has the advantage of reducing the number of parameters if enough components are left out, but a sparse kernel expansion is not obtained. In next section we show that this extra requirement can be added.

3.3. Feature approximation based on a training data subset

In order to introduce parsimony in the number of kernels, we choose a subset of $m \ll n$ data points and then a likewise m-approximation can be made:

$$\lambda_i \approx \frac{1}{m}\lambda_i^{(m)}, \tag{19}$$

$$\phi_i(\mathbf{x}) \approx \frac{\sqrt{m}}{\lambda_i^{(m)}} \sum_{j=1}^{m} k(\mathbf{x},\mathbf{x}_j)u_{ji} = \frac{\sqrt{m}}{\lambda_i^{(m)}}(\mathbf{k}^{(m)}(\mathbf{x}))^{\mathrm{T}}\mathbf{u}_i. \tag{20}$$

Again one can obtain an explicit expression for the entries of the approximated feature vector:

$$\begin{aligned}\varphi_i(\mathbf{x}) &= \sqrt{\lambda_i}\;\phi_i(\mathbf{x}) && (21)\\ &\approx (\lambda_i^{(m)})^{-1/2}\sum_{j=1}^{m} k(\mathbf{x}_j,\mathbf{x})u_{ji}. && (22)\end{aligned}$$

The approximation to the feature vector now becomes a $m \times 1$ vector:

$$\varphi(\mathbf{x}) \approx \Lambda_m^{-1/2} U_m^{\mathrm{T}} \mathbf{k}^{(m)}(\mathbf{x}). \tag{23}$$

It yields the following approximation of the feature matrix:

$$\Phi_n = K_{nm} U_m \Lambda_m^{-1/2}. \tag{24}$$

If this approximation is directly used in a LS model we obtain

$$f(\mathbf{x}) = (\mathbf{w}^{(m)})^{\mathrm{T}} \varphi(\mathbf{x}) + b \tag{25}$$

$$= \sum_{j=1}^{m} w_j \left(\sum_{i=1}^{m} (\lambda_j^{(m)})^{-1/2} u_{ij}^{(m)} k(\mathbf{x}_i, \mathbf{x}) \right) + b \tag{26}$$

$$\equiv \sum_{i=1}^{m} \beta_i k(\mathbf{x}_i, \mathbf{x}) + b, \tag{27}$$

where $\beta_i = \sum_{j=1}^{m} w_j (\lambda_j^{(m)})^{-1/2} u_{ij}^{(m)}$. This model, in conjunction with an entropy-based selection procedure has been named Fixed-Size LS-SVM and proposed in [31]. It delivers a *sparse kernel expansion* and at the same time the number of parameters of $\mathbf{w}$ is kept equal to m.

3.3.1. Eigenvector/value approximations

Next to the above two models that resulted from the Nyström approximation, yet another model can be identified. For this, consider the eigenvalues/vectors of the m-points approximation and how they relate to the n-points approximation [36]:

$$\lambda_i^{(n)} = \frac{n}{m} \lambda_i^{(m)}, \tag{28}$$

$$\mathbf{u}_i^{(n)} = \sqrt{\frac{n}{m}} \frac{1}{\lambda_i^{(m)}} K_{nm} \mathbf{u}_i^{(m)}, \tag{29}$$

where $i = 1, \ldots, m$. And once again one can obtain an explicit expression for the m entries of the approximate feature vector:

$$\varphi_i(\mathbf{x}) = \sqrt{\lambda_i} \phi_i(\mathbf{x}) \tag{30}$$

$$\approx (\lambda_i^{(n)})^{-1/2} \sum_{j=1}^{n} k(\mathbf{x}, \mathbf{x}_j) u_{ji}^{(n)} \tag{31}$$

$$= (\lambda_i^{(m)})^{-3/2} \sum_{j=1}^{n} k(\mathbf{x}, \mathbf{x}_j) \sum_{l=1}^{m} k(\mathbf{x}_j, \mathbf{x}_l) u_{li}^{(m)} \tag{32}$$

$$= (\lambda_i^{(m)})^{-3/2} (\mathbf{k}^{(n)}(\mathbf{x}))^{\mathrm{T}} K_{nm} \mathbf{u}_i^{(m)}. \tag{33}$$

This approximation leads to a $m \times 1$ feature vector

$$\varphi(\mathbf{x}) \approx \Lambda_m^{-3/2} U_m^{\mathrm{T}} K_{nm}^{\mathrm{T}} \mathbf{k}^{(n)}(\mathbf{x}) \tag{34}$$

and feature matrix

$$\Phi = \begin{pmatrix} \varphi^{\mathrm{T}}(\mathbf{x}_1) \\ \varphi^{\mathrm{T}}(\mathbf{x}_2) \\ \vdots \\ \varphi^{\mathrm{T}}(\mathbf{x}_n) \end{pmatrix} \approx K K_{nm} U_m \Lambda_m^{-3/2} = K \tilde{U}_{nm} \Lambda_m^{-1/2}, \tag{35}$$

where

$$\tilde{U}_{nm} = \begin{bmatrix} U_m \\ K_{(n-m)m} U_m \Lambda_m^{-1} \end{bmatrix} \tag{36}$$

is the approximation to the first m columns of the eigenspace of the kernel matrix K, but is not a true left eigensubspace, since its columns are not orthogonal to each other.

If this approximation is directly used in the least squares model we obtain the kernel expansion:

$$\begin{aligned} f(\mathbf{x}) &= (\mathbf{w}^{(m)})^{\mathrm{T}} \varphi(\mathbf{x}) + b && (37) \\ &= \sum_{j=1}^{m} w_j \left((\lambda_j^{(m)})^{-3/2} \sum_{i=1}^{n} k(\mathbf{x}, \mathbf{x}_i) \sum_{l=1}^{m} k(\mathbf{x}_i, \mathbf{x}_l) u_{lj}^{(m)} \right) + b && (38) \\ &= \sum_{j=1}^{m} w_j \left((\lambda_j^{(m)})^{-3/2} \sum_{i=1}^{n} k(\mathbf{x}, \mathbf{x}_i) K_{nm} \mathbf{u}_j^{(m)} \right) + b && (39) \\ &\equiv \sum_{i=1}^{n} \beta_i \, k(\mathbf{x}_i, \mathbf{x}) + b && (40) \end{aligned}$$

where $\beta_i = \sum_{j=1}^{m} w_j (\lambda_j^{(m)})^{-1/2} \tilde{\mathbf{u}}_j^{(n)} = \sum_{j=1}^{m} w_j (\lambda_j^{(m)})^{-3/2} K_{nm} \mathbf{u}_j^{(m)}$. We do not obtain here a sparse kernel expansion any more, but we do keep the sparseness in the number of regression parameters of $\mathbf{w}$.

With the above three models we are now in a position to generalize and identify in the following section a least squares regression model with specific restrictions. The properties of sparse kernel expansion and reduced number of parameters will become explicit model choices and these three models will be just special cases.

4. The subspace regression model

The general goal is to obtain a parsimonious model (in both parameters and kernels) of the underlying function between a set of independent variables and an dependent variable, based on pointwise information: the data input/output instances $\{\{(\mathbf{x}_i, y_i)\}_{i=1}^{n} \in \mathbb{R}^p \times \mathbb{R}\}$. Specifically we will firstly explicitize how to obtain reduction of the model parameters and secondly how to obtain a reduction of the kernel expansion. In this section we discuss the regression model, confine it to a subspace and restrain the choice of basis vectors.

4.1. Least squares regression

Consider the standard simple linear regression model in feature space

$$\mathbf{y} = \Phi \mathbf{w}^{(d)} + \mathbf{e}, \tag{41}$$

where $\mathbf{y}$ represents a $n \times 1$ vector of observations of the dependent variable, Φ is a $n \times d$ matrix of regressors $\{\varphi(\mathbf{x}_i)\}_{i=1}^{n}$, $\mathbf{w}^{(d)}$ is the unknown $d \times 1$ vector of regression coefficients and $\mathbf{e}$ is a $n \times 1$ vector of errors with zero-mean Gaussian i.i.d. values of equal variance σ^2 (unknown). We will assume all mapped data variables have been mean-centered.

To estimate the unknown model parameters, the elements of $\mathbf{w}^{(d)}$, we choose to minimize the squared errors $\mathbf{e}$ by ordinary least squares (OLS) regression

$$\min_{\mathbf{w}^{(d)} \in \mathbb{R}^d} \|\mathbf{y} - \Phi \mathbf{w}^{(d)}\|_2^2, \tag{42}$$

with the least squares estimate of the coefficient vector

$$\hat{\mathbf{w}}^{(d)}_{\mathrm{OLS}} = (\Phi^{\mathrm{T}} \Phi)^{-1} \Phi^{\mathrm{T}} \mathbf{y}. \tag{43}$$

Additionally, one can also choose to add a regularization parameter (with controlling coefficient γ) to the optimization problem, which is the well-known case of Ridge Regression (RR) [12]:

$$\min_{\mathbf{w}^{(d)} \in \mathbb{R}^d} \|\mathbf{y} - \Phi \mathbf{w}^{(d)}\|_2^2 + \frac{1}{\gamma} \|\mathbf{w}^{(d)}\|_2^2, \tag{44}$$

which introduces a bias, but may have the effect of variance reduction of the regression coefficients estimate

$$\hat{\mathbf{w}}^{(d)}_{\mathrm{RR}} = \left(\Phi^{\mathrm{T}} \Phi + \frac{1}{\gamma} I\right)^{-1} \Phi^{\mathrm{T}} \mathbf{y}. \tag{45}$$

4.2. Restriction to a subspace

A first difficulty in this feature space setup, is that d may be very large compared to the number of data points n, even $d = +\infty$ potentially. This implies that also an infinite number of regression coefficients will be necessary. But this infinite number of degrees of freedom is not practical, nor sensible, since one can approximate any function with an infinite number of parameters.

So in order to obtain a parsimonious model in the parameters one can reduce the search for the estimate to a subspace of $\mathbb{R}^{d \times 1}$, with finite dimension $s \ll d$. We can gather linearly independent vectors $\{\mathbf{v}_i\}_{i=1}^{s}$ in the columns of a $d \times s$ transformation matrix V, such that they span the subspace. This column space, denoted by $\mathrm{range}(V) = \{\mathbf{w}^{(d)} | \mathbf{w}^{(d)} = V \boldsymbol{\alpha}^{(s)} \text{ for any } \boldsymbol{\alpha}^{(s)}\}$ forms then a confined search space for the regression vector estimate and our OLS optimization is adapted as

$$\min_{\mathbf{w}^{(d)} \in \text{range(V)}} \|\mathbf{y} - \Phi \mathbf{w}^{(d)}\|_2^2, \tag{46}$$

where the restricted OLS estimate is determined by s regression coefficients:

$$\hat{\mathbf{w}}_{\text{OLS}}^{(s)} = V(V^{\text{T}} \Phi^{\text{T}} \Phi V)^{-1} V^{\text{T}} \Phi^{\text{T}} \mathbf{y}. \tag{47}$$

Remark that this model comes down to a LS problem in transformed regressors. If we express the $\varphi(\mathbf{x}_i)$ in the new coordinates $\mathbf{z}(\mathbf{x}_i) = V^{\text{T}} \varphi(\mathbf{x}_i)^{\text{T}}$, so that in matrix notation $Z = \Phi V$, we obtain Z as a $n \times s$ matrix of transformed regressors. The (i, k)th element of Z is the value of the projection on the kth feature vector $\mathbf{v}_k$. The resulting estimate from the (unrestricted) regression in Z would be $\hat{\mathbf{w}}_{\text{OLS}}^{(s)}$, and be considered as a *primal* variable.

We also can modify the RR model with a penalisation of the magnitude of the regression parameter vector,

$$\min_{\mathbf{w}^{(d)} \in \text{range(V)}} \|\mathbf{y} - \Phi \mathbf{w}^{(d)}\|_2^2 + \frac{1}{\gamma} \|\mathbf{w}^{(d)}\|_2^2, \tag{48}$$

where the restricted RR estimate is determined by s regression coefficients

$$\hat{\mathbf{w}}_{\text{RR}}^{(d)} = V \left(V^{\text{T}} \Phi^{\text{T}} \Phi V + \frac{1}{\gamma} V^{\text{T}} V \right)^{-1} V^{\text{T}} \Phi^{\text{T}} \mathbf{y}. \tag{49}$$

This RR model is a LS problem in transformed regression coefficients, and not in the regressors, because for that, the term $(1/\gamma) V^{\text{T}} V$ would be $(1/\gamma) I$, which is plain RR on some set of transformed features. If $s = d$ then a mere basis change is performed, but when $s < d$, then the solution is confined to a proper subspace. The particular choice of the basis vectors will of course affect the quality of the model.

4.3. Choice of basis

A second difficulty is that in general the elements of matrix Φ are unknown because the explicit expression for φ is not available. An elegant and optimal approach is to make use of the so-called kernel trick [28]. The need for a direct expression is typically avoided by formal manipulation towards scalar products so that kernels come into play. The element $\varphi(\mathbf{x})$ may be not explicitly known, but the projection on any other $\varphi(\mathbf{x}')$ is simply $k(\mathbf{x}, \mathbf{x}')$.

In order to insert kernels in our setup, we may choose for the subspace any linear combination of $\varphi(\mathbf{x}_i)$. If we select a (sub)set of $m \leqslant n$ features, this can be achieved by introducing a subspace decomposition $V = \Phi_m^{\text{T}} A$ with Φ_m a feature matrix in the m input vectors. The $m \times s$ matrix A holds in fact the projections (scalar products or loadings) of the s basis vectors $\mathbf{v}_k$ onto the m (score) vectors $\varphi(\mathbf{x}_l)$. This expression of V in the column basis of Φ_m^{T} allows to write $Z = \Phi V = \Phi \Phi_m^{\text{T}} A = K_{nm} A$ and our model equation becomes

$$\mathbf{y} = Z \boldsymbol{\alpha}^{(s)} + \mathbf{e} = (K_{nm} A) \boldsymbol{\alpha}^{(s)} + \mathbf{e}. \tag{50}$$

Remark that the use of kernels requires in fact to represent the subspace spanning vectors in the basis of the mapped datapoints. As such, the determination of V is replaced by a determination of the $m \times s$ matrix A on the basis of the data. In next subsection we will address the subspaces and loading matrices.

Some more general remarks:

- Once computed the $s \times 1$ vector $\boldsymbol{\alpha}^{(s)}$ of regression coefficients, we can obtain an estimate of $\mathbf{y}$, and we can indeed express the linear model as an expansion of kernels:

$$f(\mathbf{x}) = \sum_{i=1}^{m} \left(\sum_{j=1}^{s} a_{ij} \alpha_j^{(s)} \right) k(\mathbf{x}_i, \mathbf{x}) \equiv \sum_{i=1}^{m} \beta_i \, k(\mathbf{x}_i, \mathbf{x}). \tag{51}$$

- If we wish to evaluate the function after training in other (test) points, then for a given test set $\{\{(\mathbf{x}_j, y_j)\}_{j=n+1}^{t}\}$ we have a $t \times d$ feature matrix $\Phi_t = (k(\mathbf{x}_i, \mathbf{x}_j))_{ij}$ and a corresponding $t \times n$ kernel matrix $K_t = \Phi_t \Phi_m^{\mathrm{T}}$ so that the prediction in new points is expressed as

$$\hat{\mathbf{y}}_t = (K_{tm} A) \boldsymbol{\alpha}^{(s)} + \mathbf{e}. \tag{52}$$

- Since we assumed that the data are centered, we need to adjust the expression $\varphi(\mathbf{x})$ with $\varphi(\mathbf{x}) - (1/n) \sum_{i=1}^{n} \varphi(\mathbf{x}_i)$ everywhere. For the kernel matrices this has the consequence of replacing K by [29]

$$K := \left(I_n - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^{\mathrm{T}} \right) K \left(I_n - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^{\mathrm{T}} \right), \tag{53}$$

  where I_n is the identity matrix and $\mathbf{1}_n$ a $n \times 1$ vector of ones.

4.4. A complementary view

In Table 1 it is summarized how the three LS models of previous section appear as special cases of the sparse restricted least squares kernel regression framework. But also on some other common kernel methods the concept of restricted regression offers a useful complementary view. As such, we do not present with our framework one kernel version of one specific algorithm, but in fact a whole general class of kernel models.

For example, if we confine the regression coefficients to a subspace $V = \Phi^{\mathrm{T}}$, then the restricted RR model yields

$$\hat{\mathbf{w}}_{\mathrm{RR}}^{(n)} = \Phi^{\mathrm{T}} \left(K + \frac{1}{\gamma} I \right)^{-1} \mathbf{y}, \tag{54}$$

where we made use of the fact that K is invertible and symmetric. In this estimate we may readily recognize the kernel RR model solution, originally proposed in [27], and which shows up also in Regularization Networks [6,8] and Gaussian Processes [20]

Table 1
Interpretation of different LS models with Nyström based features, as special cases in the sparse restricted least squares kernel regression framework

Method	Subspace		Sparsity	
	Subset	Loading	Kernels	Parameters
KPCR	Φ_n^{T}	$U_n \Lambda_n^{-1/2}$	No	Yes
FS-LSSVM	Φ_m^{T}	$U_m \Lambda_m^{-1/2}$	Yes	Yes
low-rank	Φ_m^{T}	$K_{mm}^{-1} K_{nm}^{\mathrm{T}}$	Yes	No

context. If the model is enhanced with an additional bias term then it is related to the Least Squares Support Vector Machine (LS-SVM) [31], which also emphasizes primal-dual formulations of the problem.

Furthermore, to induce kernel sparseness, we could choose a reduced set of basis vectors $V = \Phi_m^{\mathrm{T}}$ and then we find:

$$\hat{\mathbf{w}}_{\mathrm{RR}}^{(m)} = \Phi_m^{\mathrm{T}}\left(K_{nm}^{\mathrm{T}} K_{nm} + \frac{1}{\gamma} K_{mm}\right) K_{nm}^{\mathrm{T}} \mathbf{y}, \tag{55}$$

a model proposed in [25,30]. And if we set $V = \Phi^{\mathrm{T}}$ in the restricted OLS model with radial basis kernels, then we relate immediately to RBF-networks [22]. Some commonly applied kernel based (least squares) methods can be interpreted as implicitly restricting their regression coefficients to a particular subspace.

Another proposal for a subspace may turn out to be close to the Nyström approximation. Suppose we plan to restrict to a subset of m basis vectors, then it seems a reasonable idea to take those vectors in this subspace that are closest to the other $n - m$ training vectors. If we understand 'closest' -for example- in the least squares sense, this suggests that we consider the so called 'projection matrix' of the column space of Φ_m^{T},

$$P_m = \Phi_m^{\mathrm{T}} (\Phi_m \Phi_m^{\mathrm{T}})^{-1} \Phi_m, \tag{56}$$

and take as the subspace the projected data points:

$$V = P_m \Phi^{\mathrm{T}} = \Phi_m^{\mathrm{T}} (\Phi_m \Phi_m^{\mathrm{T}})^{-1} \Phi_m \Phi^{\mathrm{T}}. \tag{57}$$

From the subspace regression point of view, we thus employ a loading matrix $A = K_{mm}^{-1} K_{mn}$ in the model

$$\mathbf{y} = (K_{nm} A)\mathbf{w}^{(n)} + \mathbf{e} \tag{58}$$

and in this case we happen to arrive at the well-known Nyström low-rank approximation for K:

$$\tilde{K} = K_{nm} K_{mm}^{-1} K_{mn}. \tag{59}$$

This shows that the LS subspace regression view is a quite versatile framework which links and complements smoothly to various other methods in the kernel methods field. In specific, we have seen in previous section how KPCR and FS-LSSVM (with its favorable sparse kernel expansion) were related through the approximation of features. The framework of this section gives a more formal general view on those models where they appear to involve the choice of a particular subspace and basis. In the next section we review in short some other sortlike models, correspond to another choice of subspace, that we can then extend with a sparse representation.

5. Eigenvectors for the subspace

In this section we describe next to KPCA two other closely related methods that deliver meaningful vectors to span the subspace. They have in common that it are eigenvectors based on the data, constructed unsupervised or supervised. We only treat here in short the basics of the kernelized versions and for a short summary from the linear viewpoint we refer to Appendix A. From previous section it is clear then that by choosing the particular subset and basis, the proposed framework in fact directly allows to accommodate these with a sparse kernel representation in a natural manner.

5.1. Kernel principal component analysis

Instead of the data points $\mathbf{x}_i$ and the design matrix X from PCA, we just deal with the features $\varphi(\mathbf{x}_i)$ and the feature matrix Φ in the kernel version. Starting from criterion (A.2) we need to solve the eigenproblem

$$\Phi^{\mathrm{T}}\Phi\mathbf{v} = \lambda\mathbf{v}. \tag{60}$$

Again by lack of explicit expression of Φ, one resorts to the application of the kernel trick to bring kernels into play. As was cleverly noted in the original paper [29], this is possible because the above solutions $\mathbf{v}$ are exactly linear combination of the $\varphi(\mathbf{x}_i)$ with coefficients $\Phi\mathbf{v}$. Thus we may state $\mathbf{v} = \Phi^{\mathrm{T}}\mathbf{u}$ and perform also a left multiplication with Φ to obtain

$$\Phi\Phi^{\mathrm{T}}\Phi\Phi^{\mathrm{T}}\mathbf{u} = \lambda\Phi\Phi^{\mathrm{T}}\mathbf{u}. \tag{61}$$

which yields, using the fact that K is positive definite, the central eigendecomposition of KPCA

$$K\mathbf{u} = \lambda\mathbf{u}. \tag{62}$$

For a more principled derivation, with a primal-dual approach, we refer to [32], where PCA was originally reported in a support vector machine formulation.

For the eigenvectors that should satisfy the PCA constraint $\mathbf{v}^{\mathrm{T}}\mathbf{v} = 1$ a normalization must be imposed:

$$\mathbf{v} = \lambda^{-1/2}\Phi^{\mathrm{T}}\mathbf{u}. \tag{63}$$

Although we do not know V explicitly, the projection on the eigenvectors, which serve as new variables, can be computed with kernels:

$$\mathbf{z}_k(\mathbf{x}) = \varphi(\mathbf{x})^{\mathrm{T}}\mathbf{v}_k = \lambda^{-1/2}\varphi(\mathbf{x})^{\mathrm{T}}\Phi^{\mathrm{T}}\mathbf{u}_k = \lambda^{-1/2}(\mathbf{k}(\mathbf{x}))^{\mathrm{T}}\mathbf{u}_k = \lambda^{-1/2}\sum_{i=1}^{n} k(\mathbf{x}_i, \mathbf{x})u_{ik}. \tag{64}$$

Remark that these entries coincide with the entries of the feature vector in the Nyström approximation based on the full training data set.

A natural way to parsimony in the LS regression with these KPCA features follows from observing the least squares estimate of the regression vector

$$\mathbf{w} = (Z^{\mathrm{T}}Z)^{-1}Z^{\mathrm{T}}\mathbf{y} = \Lambda^{-1}Z^{\mathrm{T}}\mathbf{y} = \Lambda^{-1}V^{\mathrm{T}}\Phi^{\mathrm{T}}\mathbf{y} = \sum_{i=1}^{n}\lambda_i^{-1}\mathbf{v}_i^{\mathrm{T}}\Phi^{\mathrm{T}}\mathbf{y} \tag{65}$$

and its corresponding variance–covariance matrix expression (assuming the output corrupted with Gaussian noise of variance σ^2)

$$\mathrm{cov}(\mathbf{w}) = \sigma^2 V(Z^{\mathrm{T}}Z)^{-1}V^{\mathrm{T}} = \sigma^2 V\Lambda^{-1}V^{\mathrm{T}} = \sigma^2\sum_{i=1}^{n}\lambda_i^{-1}\mathbf{v}_i\mathbf{v}_i^{\mathrm{T}}. \tag{66}$$

It is clear that the occurrence of small eigenvalues will introduce large variance of the estimate. Therefore the common remedy is to leave out the eigenvectors corresponding to the smallest eigenvalues. In the above expressions we take therefore $s \leqslant n$ components. The price is that it introduces a bias, but that is acceptable as long as it remains small in comparison to the variance.

As such the kernel PCA based model, or kernel principal components regression (KPCR), on n training points, extended with a bias term, and s components, becomes an expansion in kernels:

$$f(\mathbf{x}) = \sum_{j=1}^{s} w_j\mathbf{z}_j^{(n)} + b = \sum_{j=1}^{s} w_j\left(\sum_{i=1}^{n}\lambda_j^{-1/2}\mathbf{u}_{ij}\right)k(\mathbf{x}_i, \mathbf{x}) + b = \sum_{i=1}^{n}\alpha_i k(\mathbf{x}_i, \mathbf{x}) + b \tag{67}$$

where $\alpha_i = \sum_{j=1}^{s}\lambda_j^{-1/2}\mathbf{u}_{ij}$.

If we only take a subset of m training data points, we have that the subspace $V = \Phi_m^{\mathrm{T}}A$ is the $m \times s$ eigenspace of $\Phi_m^{\mathrm{T}}\Phi_m$ and that the loading matrix $A = U_{ms}\Lambda_s^{-1/2}$, the $n \times s$ eigenspace of $K_{mm} = \Phi_m\Phi_m^{\mathrm{T}}$, with an eigenvalue scaling. Taking all $s = m$ components corresponds to the Fixed Size LS-SVM model. In the next subsections we take the same approach, but consider some other alternative eigenspaces. We compute then these spaces on a reduced set to employ them likewise in the LS subspace regression model.

5.2. Kernel canonical correlation analysis

Starting from criterion (A.4), we can proceed likewise in feature space, but now with $\mathbf{v}$ and $\mathbf{w}$ as $d \times 1$ feature space vectors. To arrive at calculation with kernels instead of feature vectors one typically expands the new basis vectors as follows

$$\mathbf{v} = \sum_{i=1}^{n} \mathbf{a}_i \varphi(\mathbf{x}_i) = \Phi^{\mathrm{T}} A \tag{68}$$

$$\mathbf{w} = \sum_{i=1}^{n} \mathbf{b}_i \varphi(\mathbf{y}_i) = \Phi^{\mathrm{T}} B. \tag{69}$$

By substitution in (A.4) and use of some algebra one arrives at the following criterion

$$\max_{\mathbf{ab}} \frac{\mathbf{a}^{\mathrm{T}} K_{xx} K_{yy} \mathbf{b}}{\sqrt{\mathbf{a}^{\mathrm{T}} K_{xx} \mathbf{a}} \sqrt{\mathbf{b}^{\mathrm{T}} K_{yy} \mathbf{b}}} \tag{70}$$

where $[K_{xx}]_{ij} = \varphi(\mathbf{x}_i)^{\mathrm{T}} \varphi(\mathbf{x}_j)$ and $[K_{yy}]_{ij} = \varphi(\mathbf{y}_i)^{\mathrm{T}} \varphi(\mathbf{y}_j)$ are the kernel Gram matrices and $\mathbf{v}$ and $\mathbf{w}$ were divided by their norm. Taking then derivatives with respect to $\mathbf{a}$ and $\mathbf{b}$ and setting to zero leads to the following coupled system of equations

$$K_{yy}\mathbf{b} = \lambda K_{xx}\mathbf{a},$$

$$K_{xx}\mathbf{a} = \lambda K_{yy}\mathbf{b}. \tag{71}$$

Again, a principled primal-dual derivation was proposed in the context of primal-dual LS-SVM formulations, which leads to a more extended 'regularized' KCCA variant [31]:

$$K_{yy}\mathbf{b} = \lambda(v_1 K_{xx} + I)\mathbf{a},$$

$$K_{xx}\mathbf{a} = \lambda(v_2 K_{yy} + I)\mathbf{b}, \tag{72}$$

where v_1 and v_2 act as regularization parameters. A regularized KCCA was originally reported by [16] and [2], in an independent component analysis (ICA) context. The KCCA case where a linear kernel is used for the $\mathbf{y}$ variables was treated in [33] and here included in the tests as the KCCA1 variant.

5.3. Kernel partial least squares

By direct substitution of (68)–(69) in the PLS criterion (A.8) and use of some algebra one arrives at the following criterion

$$\max_{\mathbf{a},\mathbf{b}} \frac{\mathbf{a}^{\mathrm{T}} K_{xx} K_{yy} \mathbf{b}}{\sqrt{\mathbf{a}^{\mathrm{T}}\mathbf{a}} \sqrt{\mathbf{b}^{\mathrm{T}}\mathbf{b}}}. \tag{73}$$

Taking then derivatives with respect to $\mathbf{a}$ and $\mathbf{b}$ and equating to zero leads to the following coupled system of equations

$$\begin{aligned} K_{yy}\mathbf{b} &= \lambda\mathbf{a}, \\ K_{xx}\mathbf{a} &= \lambda\mathbf{b}. \end{aligned} \tag{74}$$

This variant is the nonlinear counterpart of PLS-SVD, for the other variants from the appendix Section A we have the following expressions:

- For KPLS-WA we can substitute X by Φ_1 and Y by Φ_2. But because of the unknown elements of Φ, we cannot directly obtain the SVD of $(\Phi_1^r)^{\mathrm{T}}\Phi_2^r$. The NIPALS-PLS algorithm [37] allows to circumvent this issue and delivers the $\mathbf{a}$ and $\mathbf{b}$ as first singular vectors of $K_{xx}^{(r)}K_{yy}^{(r)}$ and $K_{yy}^{(r)}K_{xx}^{(r)}$. Then the deflation expressions become for PLS-U at step r:

  $$\begin{aligned} K_{xx}^{(r+1)} &= K_{xx} - K_{xx}^2 A_r (A_r^{\mathrm{T}} K_{xx}^3 A_r)^{-1} A_r^{\mathrm{T}} K_{xx}^2 \\ K_{yy}^{(r+1)} &= K_{yy} - K_{yy}^2 B_r (B_r^{\mathrm{T}} K_{yy}^3 B_r)^{-1} B_r^{\mathrm{T}} K_{yy}^2. \end{aligned} \tag{75}$$

  The same is valid for KPLS-WA (which resembles the kernel version of PLS1 [26]) where we have:

  $$\begin{aligned} K_{xx}^{(r+1)} &= K_{xx}^{(r)} - \mathbf{a}_r\mathbf{a}_r^{\mathrm{T}}K_{xx}^{(r)} - K_{xx}^{(r)}\mathbf{a}_r\mathbf{a}_r^{\mathrm{T}} + \mathbf{a}_r\mathbf{a}_r^{\mathrm{T}}K_{xx}^{(r)}\mathbf{a}_r\mathbf{a}_r^{\mathrm{T}} \\ K_{yy}^{(r+1)} &= K_{yy}^{(r)} - \mathbf{b}_r\mathbf{b}_r^{\mathrm{T}}K_{yy}^{(r)} - K_{yy}^{(r)}\mathbf{b}_r\mathbf{b}_r^{\mathrm{T}} + \mathbf{b}_r\mathbf{b}_r^{\mathrm{T}}K_{yy}^{(r)}\mathbf{b}_r\mathbf{b}_r^{\mathrm{T}}. \end{aligned} \tag{76}$$

- The kernel version of PLS2 and PLS1 has been extensively studied in [26].
- For KPLSx and KPLSy we point out that its solutions can be considered in the optimization context as special cases of the regularized KCCA variant of (72) if one fixes the positive parameters (v_1, v_2) respectively as $(1,0)$ and $(0,1)$. In fact even KPLS-SVD is a subcase, with $(v_1, v_2) = (0,0)$.

So all these closely related methods deliver us meaningful $m \times m$ subspaces spanned by the column space of the loading matrix A (where we assume eigenvectors $\{\mathbf{a}_i\}_{i=1}^m$ ordered corresponding to nondecreasing values of the eigenvalues λ_i). For an overview of the eigenvalue criterion for the methods (K)PCA, (K)CCA and some (K)PLS variants, we refer to Tables 2 and 3.

6. Experiments

We perform some experiments on the LS subspace models on a reduced set of size m, with each time a different eigenspace, as outlined in the previous section. We assess whether there are severe differences in performance for prediction on an independent test set. We do these tests of an artificial and two real-world data sets. These data sets are rather of moderate size, instead of large, but it allows to compare with a standard regressor solved on the full set, as an absolute reference. With a large

Table 2
Independent test set performance in terms of mean square error (standard deviation in brackets) versus subset size for different subset based least squares subspace regression models on the Boston housing data set

Subset size	25	50	75	100	150	200	250	300
KPCR	0.476	0.407	0.271	0.251	0.196	0.187	0.173	0.161
	(0.039)	(0.054)	(0.078)	(0.054)	(0.026)	(0.027)	(0.027)	(0.028)
KPLS-SVD	0.505	0.437	0.327	0.303	0.209	0.195	0.194	0.192
	(0.028)	(0.050)	(0.085)	(0.041)	(0.037)	(0.040)	(0.040)	(0.040)
KPLSx	0.515	0.433	0.308	0.297	0.211	0.196	0.194	0.181
	(0.043)	(0.053)	(0.062)	(0.042)	(0.034)	(0.038)	(0.038)	(0.038)
KPLSy	0.504	0.418	0.329	0.297	0.211	0.201	0.192	0.192
	(0.045)	(0.046)	(0.094)	(0.046)	(0.034)	(0.036)	(0.038)	(0.036)
KPLS-WA	0.478	0.410	0.272	0.255	0.196	0.199	0.197	0.183
	(0.039)	(0.055)	(0.077)	(0.055)	(0.026)	(0.029)	(0.029)	(0.030)
KPLS-U	0.478	0.409	0.273	0.255	0.195	0.194	0.188	0.180
	(0.039)	(0.053)	(0.078)	(0.055)	(0.025)	(0.027)	(0.028)	(0.028)
KPLS1	0.476	0.407	0.271	0.252	0.195	0.187	0.173	0.161
	(0.039)	(0.054)	(0.078)	(0.054)	(0.026)	(0.026)	(0.028)	(0.027)
KCCA	0.503	0.420	0.316	0.291	0.203	0.198	0.194	0.192
	(0.031)	(0.054)	(0.091)	(0.043)	(0.034)	(0.035)	(0.037)	(0.035)
KCCA1	0.526	0.435	0.334	0.298	0.207	0.195	0.194	0.194
	(0.050)	(0.063)	(0.092)	(0.049)	(0.038)	(0.039)	(0.042)	(0.040)
LS-SVMsubset	0.646	0.543	0.422	0.388	0.289	0.268	0.220	0.219
	(0.078)	(0.070)	(0.102)	(0.039)	(0.045)	(0.045)	(0.047)	(0.046)

scale data set example we demonstrate that the sparse kernel framework indeed can manage where other methods fail.

6.1. Artificial data example

We applied the LS subspace model with the various subspace choices on a simple sinc function. We considered a domain dataset of 200 equally-spaced points in the interval $[-10, 10]$. The corresponding output values were centralized. We used the common Gaussian kernel (with width parameter h)

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{h^2}\right). \tag{77}$$

In Fig. 1 (top) a typical picture of the first three components qualitatively show a good correlation with the target function. Non-equally-spaced sampling causes the components to be more irregular and oscillatory, while prediction will be less performant in undersampled regions and near boundaries. In Fig. 1 (bottom) we show an approximation example on the same data, but with added Gaussian noise with standard deviation $\sigma = 0.2$. The other methods give similar component profiles and prediction results. This example shows that often with a very small subset very

Table 3
Independent test set performance in terms of mean square error (standard deviation in brackets) versus subset size for different subset based least squares subspace regression models on the Abalone data set

Subset size	25	50	75	100	150	200	250	300
KPCR	0.558	0.535	0.510	0.498	0.490	0.484	0.479	0.475
	(0.028)	(0.022)	(0.013)	(0.011)	(0.008)	(0.005)	(0.005)	(0.003)
KPLS-SVD	0.575	0.565	0.528	0.522	0.513	0.511	0.505	0.504
	(0.035)	(0.025)	(0.016)	(0.013)	(0.015)	(0.010)	(0.005)	(0.010)
KPLSx	0.582	0.565	0.528	0.522	0.517	0.506	0.503	0.502
	(0.034)	(0.028)	(0.016)	(0.012)	(0.014)	(0.013)	(0.007)	(0.007)
KPLSy	0.579	0.562	0.525	0.521	0.511	0.499	0.498	0.497
	(0.022)	(0.027)	(0.013)	(0.013)	(0.020)	(0.010)	(0.014)	(0.008)
KPLS-WA	0.570	0.544	0.515	0.504	0.494	0.490	0.483	0.481
	(0.029)	(0.026)	(0.013)	(0.014)	(0.009)	(0.005)	(0.005)	(0.005)
KPLS-U	0.577	0.545	0.522	0.508	0.496	0.490	0.484	0.479
	(0.029)	(0.023)	(0.016)	(0.017)	(0.009)	(0.008)	(0.005)	(0.007)
KPLS1	0.558	0.535	0.510	0.498	0.490	0.484	0.479	0.475
	(0.028)	(0.022)	(0.013)	(0.011)	(0.008)	(0.005)	(0.005)	(0.003)
KCCA	0.578	0.567	0.525	0.522	0.511	0.499	0.499	0.498
	(0.022)	(0.029)	(0.015)	(0.015)	(0.022)	(0.013)	(0.011)	(0.013)
KCCA1	0.578	0.567	0.526	0.524	0.510	0.498	0.498	0.498
	(0.022)	(0.022)	(0.022)	(0.014)	(0.018)	(0.011)	(0.017)	(0.019)
LS-SVMsubset	0.901	0.806	0.742	0.728	0.678	0.650	0.617	0.602
	(0.058)	(0.019)	(0.027)	(0.025)	(0.033)	(0.018)	(0.018)	(0.017)

good approximations can be made, with performance close to the one of methods that make use of information on the full training set.

We compared for the methods for different subset sizes with each 20 trials on sinc data sets (noise added with standard deviation $\sigma = 0.2$ and subset sizes $m = 1, 2, \ldots, 50$). Parameter $h^2 \in \{e^{-4}, e^{-3.5}, \ldots, e^{10}\}$ was determined by 10-fold cross-validation (CV). From Fig. 2 (top) we see only one representer for all methods because they took virtually the same value of mean square error (MSE) on an independent test data set, and have comparable variance.

For reference we included the solutions of a state-of-the-art regression solver, the standard LS-SVM for regression, using the LS-SVMlab software from http://www.esat.kuleuven.ac.be/sista/lssvmlab/. The LS-SVM was trained on the full set and also on the subset only, which corresponds to best case and worst case performances respectively. From a subset size $m = 6$ on, the difference with the full set solution vanishes. Intuitively, we might say that the more redundancy in the data, the quicker a reduced set method will reach the optimal performance.

As for the KCCA parameters v_1 and v_2 we conclude that the regression result is fairly insensitive to v_2, but that large values of v_1 cause overfitting, we further just took unity values. The use of other kernels, like the polynomial or the sigmoidal kernel, did not produce such good results as the Gaussian kernel.

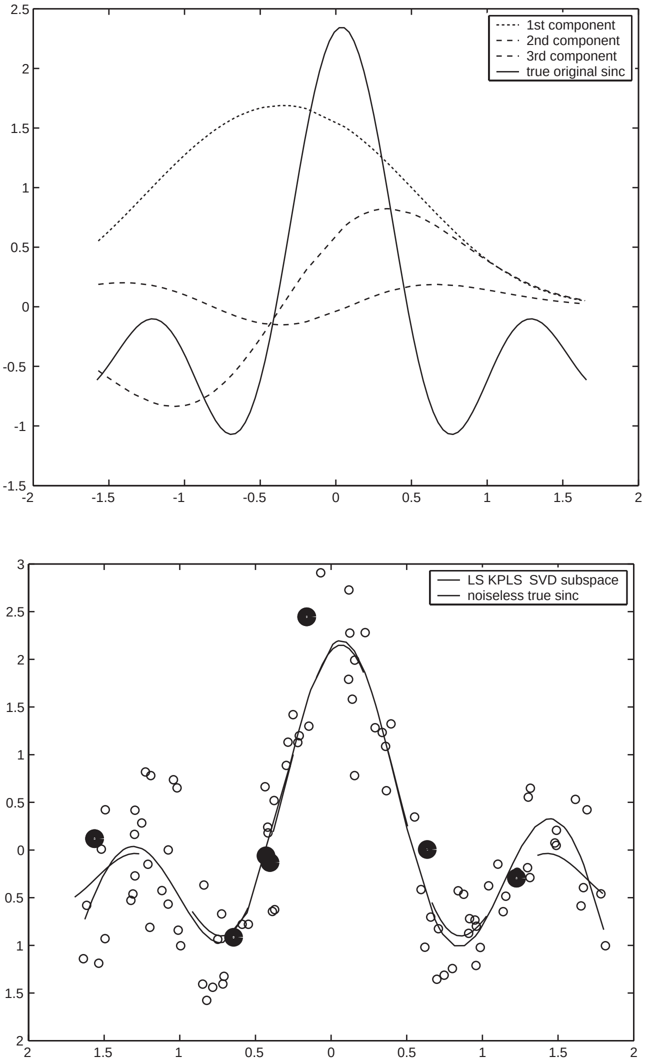
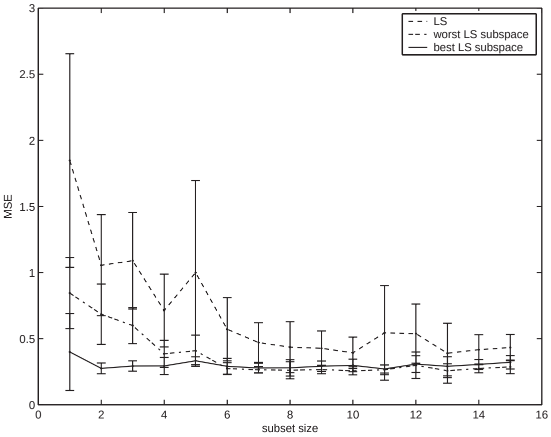
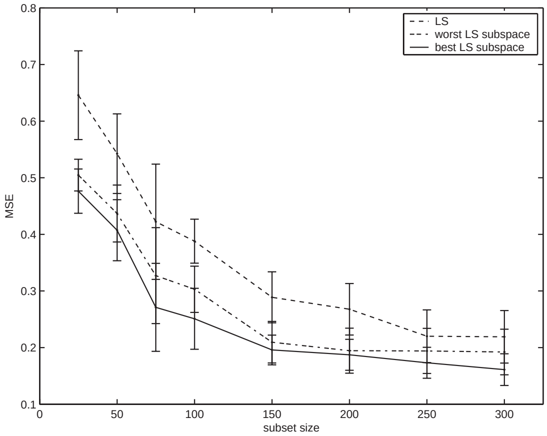


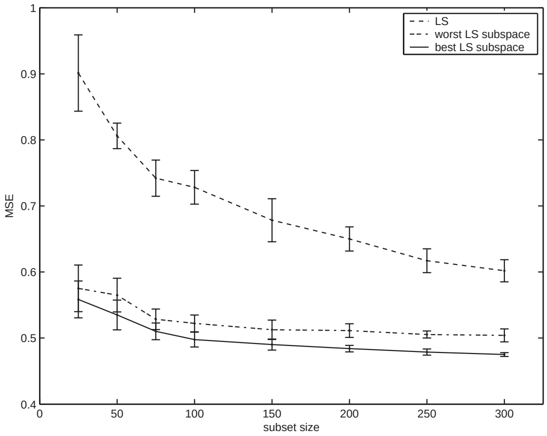
Fig. 1. (top): Visualisation of the features of the LS PLS-SVD subspace regression model on a sinc artificial data set sample. The subset consists of 5 points. It are linear combinations of these clearly nonlinear features that will fit the original curve; (bottom): Visualization of the LS KPCA subspace regression model, leading to a sparse representation, on sinc artificial data set sample. The subset consists of 5 points, marked with a '+' on the figure.

6.2. Real world data examples

The Boston Housing data set [10] consists of 506 cases having $p = 13$ input variables. The aim is to predict the housing prices. We standardized the data to zero mean and unit variance. We picked at random a training set of size $n = 400$ and a test set of size $t = 106$.


LS
worst LS subspace
best LS subspace
MSE
0
0.5
1
1.5
2
2.5
3
0
2
4
6
8
10
12
14
16
subset size


LS
worst LS subspace
best LS subspace
MSE
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
50
100
150
200
250
300
subset size


LS
worst LS subspace
best LS subspace
MSE
0.4
0.5
0.6
0.7
0.8
0.9
1
0
50
100
150
200
250
300
subset size

We performed 10-fold CV to estimate the kernel width parameter. After CV the best model was evaluated on an independent test set. The randomization trials were repeated 20 times and the subset size over the values (25 : 25 : 300) is shown in Fig. 2 (middle). In the plot we show the MSE versus subset size. For the corresponding numeric overview of the MSE performances of the different methods, we refer to Table 2. The full set LS-SVM solution, achieved a best MSE= 0.1381 with regularization parameter $\gamma = 32.3517$ and kernel width $h^2 = 15.9694$, determined by CV.

On this data set the different methods yield more different results than on the sinc, and for clarity we show the worst LS subspace regression result and the best; all other methods take their values in this band. Yet, a Wilcoxon rank sum test [24] at a significance level of $\alpha = 0.05$ shows no difference of the mean values between the methods relative to the standard deviations. Apart from this nondifference, we must remark that the mean of the KPCA based model is approximately equal to the mean of the KPLS1 model.

The Abalone data set is another benchmark from the same UCI repository [4], consisting of 4177 cases, having $p = 7$ input variables. The aim is to predict the age of abalone fish from physical measurements. We picked at random a training set of size $n = 3000$ and a test set of size $t = 1117$. The same tests were repeated. The full set LS-SVM solution achieved a best MSE $= 0.4476$ with regularization parameter $\gamma = 9.768$ and kernel width $h^2 = 12.276$, both determined by CV.

For the numeric overview of the MSE performances of the different methods, we refer to Table 3. As can be seen from Fig. 2 (bottom), on this data set the methods start to perform with difference from around $m = 75$ on. Again we show a band of the worst (KPLSx), with in between the other methods (KPLS-SVD, KPLSy, KCCA, KCCA1), till the best (KPLS-U, KPLS-WA, KPLS1, KPCR). Yet, relative to the standard deviation, we may conclude that the difference is not too large in fact either. Again we note that KPLS1 coincides practically with the kernel principal component regression (KPCR) solution.

6.3. Large scale example

In general, from the computational side our approach achieves overall a much smaller $O(nm)$ memory cost, compared to the typical $O(n^2)$ and a computational complexity of $O(nm^3)$ compared to the typical $O(n^2)$.

The ADULT UCI data set [4] consists of 45 222 cases having 14 input variables. The aim is to classify if the income of a person is greater than 50K based on several

Fig. 2. (top): Plot of mean square error on independent test set versus subset size for the sinc artificial data set. It suffices to choose a subset of 6 data points to approximate the original function to a proper accuracy; (middle): Plot of mean square error on independent test set versus subset size for the Boston housing data set. In between the best and worst bounds, the other methods take their mean performance values on the randomization sets; (bottom): Plot of mean square error on independent test set versus subset size for the Abalone data set. In between the best and worst bounds, the other methods take their mean performance values on the randomization sets.

census parameters, such as age, education, marital status, etc. We standardized the data to zero mean and unit variance. We picked at random a training set of size $n = 33\,000$ and a test set of size $t = 12\,222$. We used the common Gaussian kernel with width parameter $h^2 = 29.97$ derived from preliminary experiments. We performed 10-fold cross-validation to estimate the optimal number of components. After cross-validation the best model was evaluated on the independent test set. The randomization trials were repeated 20 times, and this for subset sizes over different values in the range [25,1000]. Training and evaluation time per model is typically of the order of minutes (12 s to 5 m) for a modest number of components of order 100 (on a Pentium 2 GHz pc).

In Fig. 3 we show the averaged misclassification rate (in percent) versus subset size m on the test set. We show the two best results, namely FS-LSSVM and sparse KPLS1. The LS-SVM was trained and cross-validated each time only on the subset data of size m, because it is computationally not possible to take into account the information of the entire training set. Qualitatively both the FS-LSSVM and sparse KPLS1 outperform LS-SVM (on subset) with a considerable amount of at least 4%. A Wilcoxon rank sum test [24] at a significance level of $\alpha = 0.01$ shows no difference of the mean values between the two sparse methods relative to the standard deviations. Other tests on smaller benchmark datasets all confirm that KPLS equals FS-LSSVM in performance.

It is remarkable that already with a subset size of 75 a favorable classification is obtained. The lowest rate was reached at a subset size of $m = 400$, after which we found no further improvements. Compared with results from literature, a correct test

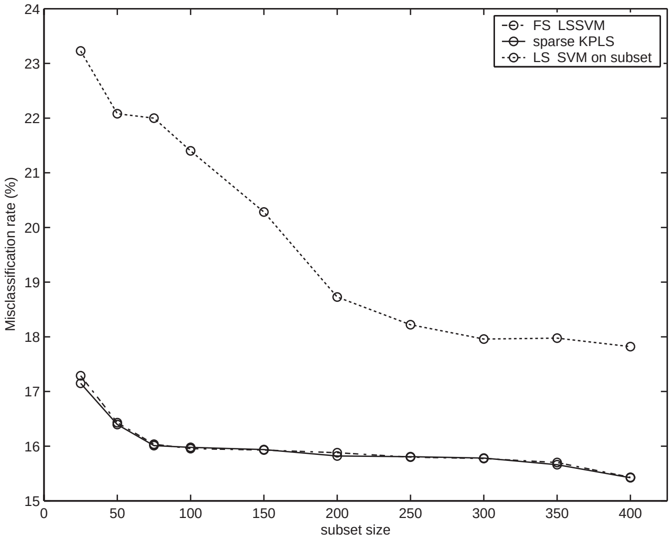


Fig. 3. Averaged misclassification rate (in %) on the independent test set versus subset size m for FS-LSSVM, sparse KPLS and LS-SVM on the subset. Note that few basis vectors are necessary to reach favorable performance at a low computational cost.

set classification rate of 84.7(±0.3)% ranks among the best of results delivered by other current state-of-the-art classifiers [31]. The use of other kernels, like the polynomial or the sigmoidal kernel, did not produce such good results as the Gaussian kernel.

7. Conclusions

In large scale applications a model must be able to process a large number of datapoints. Most kernel methods are simply not adequate for this task. In this paper we started from the Nyström feature approximation which bears in a least squares regression context the KPCR and the Fixed-Size LS-SVM model. The latter operates in primal space and has two important advantages: a small number of regression coefficients (which allows a fast training) and a sparse kernel expansion (which allows fast evaluation).

Extending FS-LSSVM with supervised counterparts, we presented a regression framework in which these two advantages are distinct model choices, through a restriction of the search space to a subspace and a particular choice of basis vectors in feature space. We showed that our model is not just a kernelization of a single model, but comprises a whole general class of least squares kernel models at once.

We focused then on restricted models using only a small subset of basis vectors and chose eigenspaces, derived from covariance optimization criteria, that have yielded feasible models in a non-reduced setting before, like KPLS and KCCA, plus some common variants. As such, we obtain sparse KPLS and sparse KCCA models, enhanced to deal with large data sets.

From experiments we may conclude that the feasibility carries over well to the reduced setting, with a limited loss of accuracy. Taking the Nyström or KPCA based subset model as reference, some methods, like KPLS1, KPLS-U, KPLS-WA show on some test cases to perform equally well, while others perform close to it. On a large scale example we show that a sparse restricted regression model is effectively capable of dealing with a large dataset.

In future work there are certainly some issues that need more investigation: how to choose or construct optimal subspaces, how to choose the right subspace size, what basis vectors should one choose, which kernel is best suited for the dependent variables, etc. Partially they are classical questions, partially they are specific to this nonlinear context. Nevertheless, the formulated framework of subspace regression with a reduced set of basis functions offers a more algebraically motivated, complementary view and renders sparse models, capable of dealing effectively with large scale data sets.

Acknowledgements

The authors would like to thank the anonymous reviewers for their comments and suggestions to help improve the manuscript. All scientific responsibility is assumed by the authors. This research work was carried out at the ESAT laboratory of the Katholieke Universiteit Leuven. It is supported by grants from several funding agencies and sources: Research Council KU Leuven: Concerted Research Action GOA-Mefisto 666 (Mathematical Engineering), IDO (IOTA Oncology, Genetic networks), several PhD/postdoc & fellow grants; Flemish Government: Fund for Scientific Research Flanders (several PhD/postdoc grants, projects G.0407.02 (support vector machines), G.0256.97 (subspace), G.0115.01 (bio-i and microarrays), G.0240.99 (multilinear algebra), G.0197.02 (power islands), research communities ICCoS, ANMMM), AWI (Bil. Int. Collaboration Hungary/ Poland), IWT (Soft4s (softsensors), STWW-Genprom (gene promotor prediction), GBOU-McKnow (Knowledge management algorithms), Eureka-Impact (MPC-control), Eureka-FLiTE (flutter modeling), several PhD grants); Belgian Federal Government: DWTC (IUAP IV-02 (1996–2001) and IUAP V-10-29 (2002–2006) (2002–2006): Dynamical Systems and Control: Computation, Identification & Modelling), Program Sustainable Development PODO-II (CP/40: Sustainibility effects of Traffic Management Systems); Direct contract research: Verhaert, Electrabel, Elia, Data4s, IPCOS. Luc Hoegaerts is a PhD student supported by the Flemish Institute for the Promotion of Scientific and Technological Research in the Industry (IWT). Johan Suykens is an associate professor at KU Leuven. Bart De Moor and Joos Vandewalle are full professors at KU Leuven, Belgium.

Appendix A. Subspace construction

Basically, we need to find directions in the variable space that form a new basis, upon which we can project. Several criteria can be chosen to arrive at a subspace to confine the regression. We take an overview of two important criteria and an intermediate one.

A.1. Minimization of within-space correlation

A more common name for within-space correlation is multicollinearity, the degree of covariance between the data vectors in x space. It occurs often when using multiple regression on data that one has collected that one has no full control over the design of the experiment. A high degree of multicollinearity produces unacceptable uncertainty (large variance) in regression coefficient estimates.

The commonly used tool for the purpose of multicollinearity reduction is principal component analysis (PCA). For a broad and thorough overview of this technique a standard reference is [13]. PCA is mostly stated as a problem in which one maximizes the variance of the new variables $\mathbf{s} = \mathbf{v}^{\mathrm{T}}\mathbf{x}$:

$$\max_{\mathbf{v}} \operatorname{var}(\mathbf{v}^{\mathrm{T}}\mathbf{x}) = \mathbf{v}^{\mathrm{T}} C_{xx} \mathbf{v} \tag{A.1}$$

subject to $\|\mathbf{v}\| = 1$ and $V^{\mathrm{T}}V = I_p$, with $C_{xx} = X^{\mathrm{T}}X$ the $p \times p$ sample covariance matrix. This involves a diagonalization procedure which requires solving an eigenvalue problem

$$C_{xx}\mathbf{v} = \lambda\mathbf{v}. \tag{A.2}$$

From the viewpoint of dimension reduction one can also describe PCA as the search for a best fitting subspace in a least squares sense. So PCA is equivalent to successive minimization of

$$J_{\mathrm{PCA}}(\mathbf{v}) = \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{v}\mathbf{v}^{\mathrm{T}}\mathbf{x}_i\|^2, \tag{A.3}$$

subject to the constraints.

A.2. Maximization of between-space correlation

On the other hand, the goal is maximization of between-space correlation. For the purpose of prediction one wishes to select subspace input vectors that have strong correlation with the target vectors.

The specific method that implements this criterion is canonical correlation analysis (CCA) [9]. Here, one considers the $p \times q$ sample covariance matrix $C_{xy} = X^{\mathrm{T}}Y$ between two spaces. To minimize the cross-covariances, again diagonal elements are maximized

$$\max_{\mathbf{v},\mathbf{w}} \operatorname{corr}(\mathbf{v}^{\mathrm{T}}\mathbf{x}, \mathbf{w}^{\mathrm{T}}\mathbf{y}) = \frac{\mathbf{v}^{\mathrm{T}}C_{xy}\mathbf{w}}{\sqrt{\mathbf{v}^{\mathrm{T}}C_{xx}\mathbf{v}}\sqrt{\mathbf{w}^{\mathrm{T}}C_{yy}\mathbf{w}}} \tag{A.4}$$

subject to $\operatorname{var}(\mathbf{v}^{\mathrm{T}}\mathbf{x}) = \mathbf{v}^{\mathrm{T}}C_{xx}\mathbf{v} = 1$ and $\operatorname{var}(\mathbf{w}^{\mathrm{T}}\mathbf{y}) = \mathbf{w}^{\mathrm{T}}C_{yy}\mathbf{w} = 1$. Essentially this requires the solution of the system

$$C_{xy}\mathbf{w} = \lambda C_{xx}\mathbf{v} \tag{A.5}$$

$$C_{yx}\mathbf{v} = \lambda C_{yy}\mathbf{w}. \tag{A.6}$$

The new basises in both spaces are chosen such that the vector components (projections) of all data maximally coincide. In CCA one successively minimizes

$$J_{\mathrm{CCA}}(\mathbf{v}, \mathbf{w}) = \sum_{i=1}^{n} \|\mathbf{v}^{\mathrm{T}}\mathbf{x}_i - w^{\mathrm{T}}\mathbf{y}_i\|^2 \tag{A.7}$$

subject to the constraints. Thus the difference between the cosine of angles of lines in both spaces is minimized.

A.3. An intermediate criterion

The two above criteria can be taken as two extremes, and an optimal subspace choice may involve a trade-off. The partial least squares (PLS) [37] method can be positioned in between these two. PLS is a multivariate technique that delivers an optimal basis in x-space for y onto x regression. Reduction to a certain subset of the basis introduces a bias, but reduces the variance.

In general, PLS is based on a maximization of the covariance between successive linear combinations in x and y space, $\langle\mathbf{v}, \mathbf{x}\rangle$ and $\langle\mathbf{w}, \mathbf{y}\rangle$, where coefficient vectors are normed to unity and constrained to be orthogonal in x space:

$$\max_{\mathbf{v},\mathbf{w}} \operatorname{cov}(\mathbf{v}^{\mathrm{T}}\mathbf{x}, \mathbf{w}^{\mathrm{T}}\mathbf{y}) = \mathbf{v}^{\mathrm{T}} C_{xy} \mathbf{w} \tag{A.8}$$

subject to $\|\mathbf{v}\| = 1 = \|\mathbf{w}\|$ and $V^{\mathrm{T}}V = I_p$. Solutions can be obtained by using Lagrange multipliers, which leads to solving the following system

$$C_{xy}\mathbf{w} = \lambda\mathbf{v}, \tag{A.9}$$

$$C_{yx}\mathbf{v} = \lambda\mathbf{w}. \tag{A.10}$$

As a least squares cost function, PLS turns out to be a sum of the least squares formulation of each of the above methods:

$$J_{\mathrm{PLS}}(\mathbf{v},\mathbf{w}) = J_{\mathrm{PCA}}(\mathbf{v}) + J_{\mathrm{CCA}}(\mathbf{v},\mathbf{w}) + J_{\mathrm{PCA}}(\mathbf{w}) \tag{A.11}$$

subject to the constraints. Indeed, by simplifying this expression, we obtain the covariance term only: the two variance minimizations of the CCA criterion are compensated by the PCA variance maximizations.

The above PLS version may be called PLS-SVD since the variables $\mathbf{v}, \mathbf{w}$ are the integral eigenspaces of $C_{xy}C_{yx}$. Imposing other constraints, results in other PLS variants. From the plethora of possible modified PLS constructs, we mention here a few transparant ones:

- The original version of Wold, PLS-WA, computes consecutively the first left and right singular vectors of $(X^{(r)})^{\mathrm{T}} Y^{(r)}$, after which each time the data matrices are deflated (projected into the complement of the space spanned by the previous found new variables, or scores):

$$X^{(r+1)} = X^{(r)} - \mathbf{b}_r(\mathbf{b}_r^{\mathrm{T}}\mathbf{b}_r)^{-1}\mathbf{b}_r^{\mathrm{T}} X^{(r)} \quad \text{with } \mathbf{b}_r = X^{(r)}\mathbf{v}_r$$

$$Y^{(r+1)} = Y^{(r)} - \mathbf{a}_r(\mathbf{a}_r^{\mathrm{T}}\mathbf{a}_r)^{-1}\mathbf{a}_r^{\mathrm{T}} Y^{(r)} \quad \text{with } \mathbf{a}_r = Y^{(r)}\mathbf{w}_r. \tag{A.12}$$

  As such the orthogonality of the scores is guaranteed in both spaces.
- The directly adapted most used variant of PLS-WA has resulted in PLS2 (multivariate) or PLS1 (univariate), where y-space is being deflated with the x-space score instead [14].
- We included PLS-U, which is PLS-WA, not with an orthogonality constraint, but more strongly, uncorrelatedness with the previously found coefficients:

$$V_r^{\mathrm{T}} C_{xx} V_r = I_p \tag{A.13}$$

$$W_r^{\mathrm{T}} C_{yy} W_r = I_q. \tag{A.14}$$

  By Lagrange multipliers one arrives after some lengthy calculation again at an eigenproblem with deflations:

$$X^{(r+1)} = X^{(r)} - (C_{xx}V_r)((C_{xx}V_r)^{\mathrm{T}}(C_{xx}V_r))^{-1}(C_{xx}V_r)^{\mathrm{T}} X^{(r)}$$

$$Y^{(r+1)} = Y^{(r)} - (C_{yy}W_r)((C_{yy}W_r)^{\mathrm{T}}(C_{yy}W_r))^{-1}(C_{yy}W_r)^{\mathrm{T}} Y^{(r)}. \tag{A.15}$$

- The PLS least squares interpretation allows to add immediately two more variants. Leaving out the compensating PCA term in x space, we obtain PLSx from CCA, with consequently $C_{xx} = I_p$. By symmetry, also a PLSy version can be obtained. These versions may be useful especially when dealing with many dummy variables, such that the PCA contribution does not make much sense.

For a generic overview, partly inspired by [5], of the criteria for the methods PCA, CCA and some PLS variants, we refer to Tables 4 and 5.

Table 4
PCA, CCA and 3 PLS-SVD variants summarized in their (co)variance and least squares formulation

	Covariance maximization	Least squares minimization
PCA	$\max_{\mathbf{v}} \frac{\mathrm{var}(\mathbf{v}^{\mathrm{T}}\mathbf{x})}{(\mathbf{v}^{\mathrm{T}}\mathbf{v})}$	$\min_{\mathbf{v}} \sum_{i=1}^{n} \Vert\mathbf{x}_i - \mathbf{v}\mathbf{v}^{\mathrm{T}}\mathbf{x}_i\Vert^2$
PLS	$\max_{\mathbf{v},\mathbf{w}} \frac{[\mathrm{cov}(\mathbf{v}^{\mathrm{T}}\mathbf{x},\mathbf{w}^{\mathrm{T}}\mathbf{y})]^2}{(\mathbf{v}^{\mathrm{T}}\mathbf{v})(\mathbf{w}^{\mathrm{T}}\mathbf{w})}$	$\min_{\mathbf{v},\mathbf{w}} \sum_{i=1}^{n} \Vert\mathbf{v}\mathbf{v}^{\mathrm{T}}\mathbf{x}_i - \mathbf{x}_i\Vert^2 + \Vert\mathbf{v}^{\mathrm{T}}\mathbf{x}_i - \mathbf{w}^{\mathrm{T}}\mathbf{y}_i\Vert^2 + \Vert\mathbf{w}\mathbf{w}^{\mathrm{T}}\mathbf{y}_i - \mathbf{y}_i\Vert^2$
PLS a	$\max_{\mathbf{v},\mathbf{w}} \frac{[\mathrm{cov}(\mathbf{v}^{\mathrm{T}}\mathbf{x},\mathbf{w}^{\mathrm{T}}\mathbf{y})]^2}{(\mathbf{v}^{\mathrm{T}}\mathbf{v})\mathrm{var}(\mathbf{w}^{\mathrm{T}}\mathbf{y})}$	$\min_{\mathbf{v},\mathbf{w}} \sum_{i=1}^{n} \Vert\mathbf{v}\mathbf{v}^{\mathrm{T}}\mathbf{x}_i - \mathbf{x}_i\Vert^2 + \Vert\mathbf{v}^{\mathrm{T}}\mathbf{x}_i - \mathbf{w}^{\mathrm{T}}\mathbf{y}_i\Vert^2$
PLS b	$\max_{\mathbf{v},\mathbf{w}} \frac{[\mathrm{cov}(\mathbf{v}^{\mathrm{T}}\mathbf{x},\mathbf{w}^{\mathrm{T}}\mathbf{y})]^2}{\mathrm{var}(\mathbf{v}^{\mathrm{T}}\mathbf{x})(\mathbf{w}^{\mathrm{T}}\mathbf{w})}$	$\min_{\mathbf{v},\mathbf{w}} \sum_{i=1}^{n} \Vert\mathbf{v}^{\mathrm{T}}\mathbf{x}_i - \mathbf{w}^{\mathrm{T}}\mathbf{y}_i\Vert^2 + \Vert\mathbf{w}\mathbf{w}^{\mathrm{T}}\mathbf{y}_i - \mathbf{y}_i\Vert^2$
CCA	$\max_{\mathbf{v},\mathbf{w}} \frac{[\mathrm{cov}(\mathbf{v}^{\mathrm{T}}\mathbf{x},\mathbf{w}^{\mathrm{T}}\mathbf{y})]^2}{\mathrm{var}(\mathbf{v}^{\mathrm{T}}\mathbf{x})\mathrm{var}(\mathbf{w}^{\mathrm{T}}\mathbf{y})}$	$\min_{\mathbf{v},\mathbf{w}} \sum_{i=1}^{n} \Vert\mathbf{v}^{\mathrm{T}}\mathbf{x}_i - \mathbf{w}^{\mathrm{T}}\mathbf{y}_i\Vert^2$

Table 5
(K)PCA, (K)CCA and 3 (K)PLS-SVD variants summarized in their generalized eigenvalue formulation, where $M[\mathbf{v};\mathbf{w}] = \lambda Q[\mathbf{v};\mathbf{w}]$ represents the linear version with solutions in primal space and $L[\mathbf{a};\mathbf{b}] = \lambda R[\mathbf{a};\mathbf{b}]$ the nonlinear kernel version, with solutions in dual space

	Linear (primal)		Kernel (dual)	
	M	Q	L	R
(K)PCA	S_{xx}	I	K_{xx}	I
(K)PLS	$\begin{pmatrix} 0 & C_{xy} \\ C_{yx} & 0 \end{pmatrix}$	$\begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix}$	$\begin{pmatrix} 0 & K_{xx}K_{yy} \\ K_{yy}K_{xx} & 0 \end{pmatrix}$	$\begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix}$
(K)PLSx	$\begin{pmatrix} 0 & C_{xy} \\ C_{yx} & 0 \end{pmatrix}$	$\begin{pmatrix} I & 0 \\ 0 & C_{yy} \end{pmatrix}$	$\begin{pmatrix} 0 & K_{xx}K_{yy} \\ K_{yy}K_{xx} & 0 \end{pmatrix}$	$\begin{pmatrix} I & 0 \\ 0 & K_{yy}^2 \end{pmatrix}$
(K)PLSy	$\begin{pmatrix} 0 & C_{xy} \\ C_{yx} & 0 \end{pmatrix}$	$\begin{pmatrix} C_{xx} & 0 \\ 0 & I \end{pmatrix}$	$\begin{pmatrix} 0 & K_{xx}K_{yy} \\ K_{yy}K_{xx} & 0 \end{pmatrix}$	$\begin{pmatrix} K_{xx}^2 & 0 \\ 0 & I \end{pmatrix}$
(K)CCA	$\begin{pmatrix} 0 & C_{xy} \\ C_{yx} & 0 \end{pmatrix}$	$\begin{pmatrix} C_{xx} & 0 \\ 0 & C_{yy} \end{pmatrix}$	$\begin{pmatrix} 0 & K_{xx}K_{yy} \\ K_{yy}K_{xx} & 0 \end{pmatrix}$	$\begin{pmatrix} K_{xx}^2 & 0 \\ 0 & K_{yy}^2 \end{pmatrix}$

References

[1] N. Aronszajn, Theory of reproducing kernels, Trans. Amer. Math. Soc. 686 (1950) 337–404.

[2] F. Bach, M. Jordan, Kernel independent component analysis, J. Machine Learning Res. 3 (2002) 1–48.

[3] C. Baker, The numerical treatment of integral equations, Oxford Clarendon Press, Oxford, 1977.

[4] C. Blake, C. Merz, Uci repository of machine learning databases, (1998). URL http://www.ics.uci.edu/~mlearn/MLRepository.html.

[5] M. Borga, Learning multidimensional signal processing, Ph.D. Thesis, Department of Electrical Engineering, Linkoping University, 1998.

[6] T. Evgeniou, M. Pontil, T. Poggio, Regularization networks and support vector machines, Adv. Computat. Math. 13 (2000) 1–50.

[7] S. Fine, K. Scheinberg, Efficient SVM training using low-rank kernel representations, J. Machine Learning Res. 2 (2) (2002) 243–264.

[8] F. Girosi, M. Jones, T. Poggio, Regularization theory and neural networks architectures, Neural Comput. 7 (2) (1995) 219–269.

[9] R. Gittins, Canonical Analysis, Biomathematics Edition, Vol. 12, Springer, Berlin, 1985.

[10] D. Harrison, D. Rubinfeld, Hedonic prices and the demand for clean air, J. Environ. Econom. Manage. 5 (1978) 81–102.

[11] L. Hoegaerts, J.A.K. Suykens, J. Vandewalle, B. De Moor, Kernel pls variants for regression, in: Proc. of the 11th European Symposium on Artificial Neural Networks (ESANN2003), Bruges, Belgium, 2003, pp. 203–208.

[12] A. Hoerl, R. Kennard, Ridge regression: biased estimation for nonorthogonal problems, Technometrics 12 (3) (1970) 55–67.

[13] I. Jolliffe, Principal Component Analysis, Springer, Berlin, 1986.

[14] S. de Jong, A. Phatak, Partial least squares regression, in: S.V. Huffel (Ed.), Recent Advances in Total Least Squares Techniques and Errors-in-Variables Modeling, SIAM, Philadelphia, 1997, pp. 311–338.

[15] G. Kimeldorf, G. Wahba, Tchebycheffian spline functions, J. Math. Ana. Applic. 33 (1971) 82–95.

[16] P.L. Lai, C. Fyfe, Kernel and nonlinear canonical correlation analysis, Internat. J. Neural Systems 10 (5) (2000) 365–377.

[17] Y. Lee, O.L. Mangasarian, RSVM: reduced support vector machines, First SIAM International Conference on Data Mining, Chicago, July, 2000. Technical Report 00-07, 2001.

[18] K.-M. Lin, C.-J. Lin, A study on reduced support vector machines, IEEE Trans. Neural Networks 14 (6) (2003) 1449–1459.

[19] M. Momma, K. Bennett, Sparse kernel partial least squares regression, in: Proceedings of Conference on Learning Theory (COLT 2003), 2003, pp. 216–230.

[20] R. Neal, Bayesian learning for neural networks, Ph.D. Thesis, Graduate Dept. of Computer Science, Univ. of Toronto, 1995.

[21] E.J. Nyström, Uber die praktische auflosung von linearen integralgleichungen mit anwendungen auf randwertaufgaben der potentialtheorie, Comment. Phys.-Math. 4 (15) (1928) 1–52.

[22] T. Poggio, F. Girosi, Networks for approximation and learning, Proc. IEEE 78 (9) (1990) 1481–1497.

[23] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes: The Art of Scientific Computing, 1st Edition, Cambridge University Press, Cambridge (UK) and New York, 1986.

[24] F. Ramsey, D. Schafer, The Statistical Sleuth: A Course in Methods of Data Analysis, Wadsworth Publishing Company, Belmont, CA, 1997.

[25] R. Rifkin, G. Yeo, T. Poggio, Advances in learning theory: methods, models and applications, in: J.A.K. Suykens, G. Horvath, S. Basu, C. Micchelli, J. Vanderwalle (Eds.), Computer and Systems Sciences, NATO Science Series III, Vol. 190, IOS Press, Amsterdam, 2003.

[26] R. Rosipal, L.J. Trejo, Kernel partial least squares regression in reproducing kernel hilbert space, J. Machine Learning Res. 2 (2001) 97–123.

[27] G. Saunders, A. Gammerman, V. Vovk, Ridge regression learning algorithm in dual variables, in: Proc. 15th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA, 1998, pp. 515–521.

[28] B. Schölkopf, A. Smola, Learning With Kernels, MIT Press, Cambridge, 2002.

[29] B. Schölkopf, A. Smola, K. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (5) (1998) 1299–1319.

[30] A.J. Smola, B. Schölkopf, Sparse greedy matrix approximation for machine learning, in: Proc. 17th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA, 2000, pp. 911–918.

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

[32] J.A.K. Suykens, T. Van Gestel, J. Vandewalle, B. De Moor, A support vector machine formulation to pca analysis and its kernel version, IEEE Trans. Neural Networks 14 (2) (2003) 447–450.

[33] T. Van Gestel, J.A.K. Suykens, J. De Brabanter, B. De Moor, J. Vandewalle, Kernel canonical correlation analysis and least squares support vector machines, in: Proc. of the International Conference on Artificial Neural Networks (ICANN 2001), Vienna, Austria, 2001, pp. 381–386.

[34] G. Wahba, Spline Models for Observational Data, in: CBMS-NSF Regional Conference Series in Applied Mathematics, Vol. 59, SIAM, Philadelphia, PA, 1990.

[35] C. Williams, C. Rasmussen, A. Schwaighofer, V. Tresp, Observations on the Nyström method for Gaussian processes, Tech. rep. of Institute for Adaptive and Neural Computation, Division of Informatics, University of Edinburgh, 2002.

[36] C. Williams, M. Seeger, Using the Nyström method to speed up kernel machines, in: T.K. Leen, T.G. Diettrich, V. Tresp (Eds.), Advances in Neural Information Processing Systems, Vol. 13, MIT Press, New York, 2001, pp. 682–688.

[37] H. Wold, Estimation of principal components and related models by iterative least squares, in: P.R. Krishnaiah (Ed.), Multivariate Analysis, Academic Press, New York, 1966, pp. 391–420.

Luc Hoegaerts was born in January 23, 1975, in Bonheiden, Belgium. He received the M.Sc. degree in Physics in 1998 and a complementary degree in Informatics in 1999, from the Katholieke Universiteit Leuven. He is currently pursuing a Ph.D. at the KULeuven in the faculty of Applied Sciences, department of Electrical Engineering, in the SISTA/SCD laboratory. His research interests include machine learning, statistical inference, data modeling, data mining, neural networks, kernel methods, support vector machines.

Johan A.K. Suykens was born in Willebroek Belgium, May 18, 1966. He received the degree in Electro-Mechanical Engineering and the Ph.D. degree in Applied Sciences from the Katholieke Universiteit Leuven, in 1989 and 1995, respectively. In 1996 he has been a Visiting Postdoctoral Researcher at the University of California, Berkeley. He has been a Postdoctoral Researcher with the Fund for Scientific Research FWO Flanders and is currently an associate professor at K.U.Leuven. His research interests are mainly in the areas of the theory and application of neural networks and nonlinear systems. He is author of the books "Artificial Neural Networks for Modelling and Control of Non-linear Systems" (Kluwer Academic Publishers) and "Least Squares Support Vector Machines" (World Scientific) and editor of the books "Nonlinear Modeling: Advanced Black-Box Techniques" (Kluwer Academic Publishers) and "Advances in Learning Theory: Methods, Models and Applications" (IOS Press). In 1998 he organized an International Workshop on Nonlinear Modelling with Time-series Prediction Competition. He has served as associate editor for the IEEE Transactions on Circuits and Systems-I (1997–1999) and since 1998 he is serving as associate editor for the

IEEE Transactions on Neural Networks. He received an IEEE Signal Processing Society 1999 Best Paper (Senior) Award and several Best Paper Awards at International Conferences. He is a recipient of the International Neural Networks Society INNS 2000 Young Investigator Award for significant contributions in the field of neural networks. He has served as Director and Organizer of a NATO Advanced Study Institute on Learning Theory and Practice taking place Leuven July 2002.

Joos Vandewalle was born in Kortrijk, Belgium, in August 1948. He obtained the electrical engineering degree and a doctorate in applied sciences, both from the Katholieke Universiteit Leuven, Belgium in 1971 and 1976 respectively. From 1976 to 1978 he was Research Associate and from July 1978 to July 1979, he was Visiting Assistant Professor both at the University of California, Berkeley. Since July 1979 he is back at the Department of Electrical Engineering (ESAT) of the Katholieke Universiteit Leuven, Belgium where he is Full Professor since 1986 and the head of the SCD division at ESAT, that has more than 120 researchers. He is an Academic Consultant since 1984 at IMEC (Interuniversity Microelectronics Center, Leuven). From August 1996 to August 1999 he was Chairman of the Department of Electrical Engineering and from August 1999 till July 2002 he was the vice-dean of the Faculty of Engineering, both at the Katholieke Universiteit Leuven. In the second semester of 2002–2003 he is on sabbatical leave ath the I3S laboratory of CNRS Sophia Antipolis, France. He teaches courses in linear algebra, linear and nonlinear system and circuit theory, signal processing and neural networks. His research interests are mainly in mathematical system theory and its applications in circuit theory, control, signal processing, cryptography and neural networks. His recent research interests are in nonlinear methods (support vector machines, multilinear algebra) for data processing. He has authored or coauthored more than 200 international journal papers in these areas. He is the co-author of 4 books and co-editor of 5 books. He is a member of the editorial board of the International Journal of Circuit Theory and its Applications, Neurocomputing, Neural Networks and the Journal of Circuits Systems and Computers. From 1989 till 1991, he was associate editor of the IEEE Transactions on Circuits and Systems. He was elected fellow of IEEE in 1992 for contributions to nonlinear circuits and systems. In 1991–1992 he held the Francqui chair on Artificial Neural Networks at the University of Lige (Belgium) and in 2001–2002 he held this chair on Advanced Data Processing techniques at the Free University of Brussels. He is also Fellow of the IEE (UK). Since 2001 he is a member of the Advisory Board of the International Journal on Information Security (IJIS). Since January 2001 he is co-editor-in-chief of Journal A, the Benelux journal on Automation. He is Deputy Editor-in-chief of the IEEE Transactions on Circuits and Systems part I Fundamental theory and applications (since January 2002). He received several best paper awards and research awards. He is a member of the Academia Europaea and of the Belgian Academy of sciences and of 2 committees of the Fonds voor Wetenschappelijk Onderzoek Vlaanderen (Belgium).

Bart De Moor was born Tuesday July 12, 1960 in Halle, Belgium. He is married and has three children. In 1983, he obtained his Master (Engineering) Degree in Electrical Engineering at the Katholieke Universiteit Leuven, Belgium, and a PhD in Engineering at the same university in 1988. He spent 2 years as a Visiting Research Associate at Stanford University (1988–1990) at the departments of EE (ISL, Prof. Kailath) and CS (Prof. Golub). Currently, he is a full professor at the Department of Electrical Engineering (http://www.esat.kuleuven.ac.be) of the K.U.Leuven. His research interests are in numerical linear algebra and optimization, system theory and identification, quantum information theory, control theory, data-mining, information retrieval and bio-informatics, areas in which he has (co)authored several books and hundreds of research papers (consult the publication search engine at http://www.esat.kuleuven.ac.be/sista-cosic-docarch/template.php). Currently, he is leading a research group of 39 PhD students and 8 postdocs and in the recent past, 16 PhDs were obtained under his guidance. He has been teaching at and been a member of PhD jury's in several universities in Europe and the US. He is also a member of several scientific and professional organizations. His work has won him several scientific awards (Leybold-Heraeus Prize (1986), Leslie Fox Prize (1989), Guillemin-Cauer best paper Award of the IEEE Transaction on Circuits and Systems (1990), Laureate of the Belgian Royal Academy of Sciences (1992), bi-annual Siemens Award (1994), best paper award of Automatica (IFAC, 1996), IEEE Signal Processing Society Best Paper Award (1999). He is an associate editor of several scientific journals. From 1991–1999 he was the chief advisor on Science and Technology of several ministers of the Belgian Federal Government and the Flanders Regional Governments. He was and/or is in the board of 3 spin-off companies (www.ipcos.be, www.data4s.com, www.tml.be), of the Flemish Interuniversity Institute for Biotechnology (www.vib.be), the Study Center for Nuclear Energy (www.sck.be) and several other scientific and cultural organizations. Full details on his CV can be found at www.esat.kuleuven.ac.be/ demoor.