Locally-Based Kernel PLS Smoothing to Non-parametric Regression Curve Fitting

Roman Rosipal[1,2], Leonard J Trejo[1], 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, Slovak Republic

Abstract

~~We present a novel smoothing approach to non-parametric regression curve fitting.~~ This is based on kernel partial least squares (PLS) regression in reproducing kernel Hilbert space. It is our concern to apply the methodology for smoothing experimental data where some level of knowledge about the approximate shape, local inhomogeneities or points where the desired function changes its curvature is known a priori or can be derived based on the observed noisy data. We propose locally-based kernel PLS regression that extends the previous kernel PLS methodology by incorporating this knowledge. We compare our approach with existing smoothing splines, hybrid adaptive splines and wavelet shrinkage techniques on two generated data sets.

1 Introduction

There has been significant advancement in developing non-parametric regression techniques during the last several decades with the aim of smoothing observed data corrupted by some level of noise. A subset of these techniques is based on defining an appropriate dictionary of basis functions from which the final regression model is constructed. The model is usually defined to be a linear combination of functions selected from the dictionary. The widely used methods like smoothing splines and wavelet shrinkage belong to this category (Wahba, 1990; Donoho & Johnstone, 1995). These smoothing techniques have also been successfully applied to problems of signal de-noising which involves a wide area of research in the signal processing community.

In this setting we usually assume the signal of interest to be a linear combination of the selected basis functions $\psi_i(x) \in \mathcal{D}$

$$g(x) = \sum_{i=1}^{p} w_i \psi_i(x)$$

where $\mathcal{D}$ represents a dictionary (family) of functions and $\{w_i\}_{i=1}^{p}$ are weighting coefficients. The main problem associated with this approach is the appropriate definition of $\mathcal{D}$ and the selection of a subset of basis functions used for the final model. Using a fixed dictionary of several basis functions, e.g., all polynomials up to the order d or several trigonometric functions, may provide an easier selection among basis functions, but in general may not guarantee the possibility to closely approximate the desired signal of interest. On the other side defining our solution in a "rich" functional space may guarantee exact functional approximation of the signal of interest, however in a noisy scenario we may have a bigger problem of finding an adequate final estimate of the signal of interest. Smoothing splines and closely related support vector machines (SVM) are examples of this second approach (Wahba, 1990; Vapnik, 1998; Cristianini & Shawe-Taylor, 2000; Schölkopf & Smola, 2002). The solution is defined to lay in a rich functional space and as such it can be expressed in the basis of the space (possibly infinite). However to avoid exact fitting of a measured noisy signal we need to incorporate some a priori assumptions about the smoothness of the desired signal of interest which is usually achieved through different forms of regularization. The appropriate functional space and regularization form selection for different types of noise distribution and types of signals are the main drawbacks of these methods.

In this paper we propose a novel approach which tries to combine both of these strategies. We consider our solution to be in a reproducing kernel Hilbert space (RKHS) (Aronszajn, 1950; Saitoh, 1988). A straightforward connection between a RKHS and the corresponding feature space representation allows us to define the desired solution in a form of penalized linear regression model. More specifically, we consider a kernel partial least squares (PLS) regression in a feature space $\mathcal{F}$ (Rosipal & Trejo, 2001). The basis functions $\psi_i(x)$ are taken to be components obtain by kernel PLS, which may be seen as the estimates of an orthogonal basis in $\mathcal{F}$ defined by the measured signal and the kernel function used. These estimates are sequentially obtained using the existing correlations between nonlinearly mapped input data into $\mathcal{F}$ and the measured noisy signal, that is, extracted basis functions closely reflect existing input-output dependencies. This methodology differs from the existing approaches in the sense that the construction of the basis functions is given by the measured signal itself and is not a priori pre-defined without respect to the measured signal. The second stage of the proposed approach reduces to the problem of setting the number of basis functions p. The sequential structure of the extracted basis functions with respect to increasing description of the overall variance of the measured signal motivates our use of the Vapnik-Chervonnekis (VC) based model selection criterion (Cherkassky et al., 1999; Cherkassky & Shao, 2001).

Finally, we extend this methodology of kernel PLS smoothing by assuming a set of locally-based kernel PLS models which allows us to more effectively deal with possible local inhomogeneities in the signal of interest. It is our intention to include a priori information about an approximate location of points of change of the signal curvature, discontinuities or other local inhomogeneities occurring in the signal. This approach is proposed for problems where this kind of information is known a priori or can be derived from the experimental data. The spatial localization of individual kernel PLS models is achieved by incorporating weight functions reflecting the local areas of interest. Depending on weight function selection this allows us to construct soft or hard thresholding regions where kernel PLS regression models are constructed. Final regression estimate consists of the weighted summation of individual local kernel PLS regression models.

We compared our methodology of kernel PLS and locally-based kernel PLS smoothing with the state-of-the-art wavelet based signal de-noising, smoothing splines and hybrid adaptive splines on heavisine and simulated human event-related potentials (ERP) distributed over individual scalp areas. We investigate the situations with different levels of additive uncorrelated or spatio-temporal correlated noise added to these simulated signals. The use of wavelet shrinkage and hybrid adaptive splines is motivated by the fact that the both methods are design to deal with local inhomogeneities in the signal.

The rest of this paper is organized as follows. First, in Section 2 a basic definition of a RKHS and formulation of the Representer theorem are given. In the next section the kernel PLS and locally-based kernel PLS regression models are described. This section is closed with a formulation of smoothing splines in RKHS and with a description of the hybrid adaptive splines approach. Section 3 describes the construction of used data sets. The experimental results are given in Section 4. Section 5 provides a short discussion and concludes the paper.

2 Methods

2.1 Basic formulation

Consider the regression problem

$$y_i = g(x_i) + \epsilon_i \tag{1}$$

where $\{y_i\}_{i=1}^n$ represent observations at equidistant design points $\{x_i\}_{i=1}^n$, $a < x_1 < x_2 < \ldots < x_n < b$ in $[a, b]$ and $\{\epsilon_i\}_{i=1}^n$ are errors not restricted to be uncorrelated or to be drawn from a pre-specified probability distribution. In this paper we consider non-parametric estimation of the function $g(.)$. We assume that $g(.)$ is a *smooth* function in a functional space $\mathcal{H}$. To restrict our estimate of $g(.)$ in $\mathcal{H}$ to be a function with the desired smoothness property we consider an estimate $\hat{g}$ to be obtained as

$$\hat{g}(.) = \arg\min_{g \in \mathcal{H}} \left[\frac{1}{n} \sum_{i=1}^{n} (y_i - g(x_i))^2 + \xi \Omega(g) \right] \tag{2}$$

In this formulation ξ is a positive number (regularization coefficient or constant) to control the trade-off between approximating properties and the smoothness of $g(.)$ imposed by the penalty functional $\Omega(g)$. Further we will assume that $\mathcal{H}$ is a RKHS which provides a finite dimensional solution of (2) in spite of the fact that (2) is defined over an infinite-dimensional space. Kernel partial least squares regression and smoothing splines described below fall into this framework.

A RKHS is uniquely defined by a positive definite kernel function $K(x, y)$, i.e., a symmetric function of two variables satisfying the Mercer theorem conditions (Mercer, 1909; Cristianini & Shawe-Taylor, 2000).[1] The fact that for any such positive definite kernel there exists a unique RKHS is well established by the *Moore-Aronszjan theorem* (Aronszajn, 1950). The form $K(x, y)$ has the following *reproducing property*

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

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

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

$$K(x, y) = \sum_{i=1}^{S} \lambda_i \phi_i(x) \phi_i(y) \quad S \leq \infty \tag{3}$$

[1] We consider one-dimensional input space, however, the following theoretical results are equally valid for a higher dimensional scenario.

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

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

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

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

and the norm

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

Define the *feature vector* $\Phi(.) = (\sqrt{\lambda_1}\phi_1(.), \sqrt{\lambda_2}\phi_2(.), \ldots, \sqrt{\lambda_S}\phi_S(.))^T$. Rewriting (3) in the form

$$K(x,y) = \sum_{i=1}^S \sqrt{\lambda_i}\phi_i(x)\sqrt{\lambda_i}\phi_i(y) = (\Phi(x).\Phi(y)) = \Phi(x)^T\Phi(y) \tag{4}$$

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

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

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

Now we can return to the solution of (2) and consider a general case when a penalty functional $\Omega(g)$ is designed to not penalize some components of $\mathcal{H}$, i.e., there will be a subspace of $\mathcal{H}$ of not penalized functions called the *null space*. Kimeldorf and Wahba (1971) have shown that in this case the solution of (2) leads to a general finite dimensional form also known as the *Representer theorem*:

$$\hat{g}(x) = \sum_{i=1}^n d_i K(x_i, x) + \sum_{j=1}^l e_j v_j(x) \tag{5}$$

where the functions $\{v_j(.)\}_{j=1}^l$ span the null space of $\mathcal{H}$ and the coefficients $\{d_i\}_{i=1}^n$, $\{e_j\}_{j=1}^l$ are given by the data. It is worth noting that in the case we consider a positive definite kernel K and the functional $\Omega(g) = \|g\|_{\mathcal{H}}^2$ there is an empty null space and we have the solution $\hat{g}(x) = \sum_{i=1}^n c_i K(x_i, x)$ also known as regularization network (Girosi, Jones, & Poggio, 1995). The Representer theorem also provides a very effective way to connect regularization networks, smoothing splines and recently highly developed methodology of SVM for regression (Vapnik, 1998; Cristianini & Shawe-Taylor, 2000; Schölkopf & Smola, 2002). The theory of SVM also motivated our construction of kernel PLS regression technique which we describe in the next subsection.

2.2 Kernel Partial Least Squares Regression

PLS Regression belongs to the class of regularized regression techniques. The core of the technique is the PLS method which creates orthogonal components by using the existing correlations between input (explanatory) variables and corresponding outputs while also trying to keep most of the variance of both, input and output data spaces (Frank & Friedman, 1993). For a moment we consider a general setting of linear PLS regression model where $\mathbf{x} \in \mathcal{R}^N$ denotes an N-dimensional vector of input variables and similarly we consider $\mathbf{y} \in \mathcal{R}^M$ denotes a vector of output variables. PLS decomposes the $(n \times N)$ matrix of zero-mean explanatory variables $\mathbf{X}$ and the $(n \times M)$ matrix of zero-mean outputs $\mathbf{Y}$ into the form

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

where the $\mathbf{T}$, $\mathbf{U}$ are $(n \times p)$ matrices of the extracted p orthogonal components (scores, latent variables), the $(N \times p)$ matrix $\mathbf{P}$ and the $(M \times p)$ matrix $\mathbf{Q}$ represent matrices of loadings and the $(n \times N)$ matrix $\mathbf{E}$ and the $(n \times M)$ matrix $\mathbf{F}$ are the matrices of residuals. We further assume that the X-scores variables $\{\mathrm{t}_i\}_{i=1}^p$ are good predictors of $\mathbf{Y}$. We also assume a linear inner relation between the scores of $\mathbf{t}$ and $\mathbf{u}$, that is,

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

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

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

which defines the considered linear PLS regression model

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

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

Consider now again $N = 1$, that is $x \in \mathcal{R}$, and assume a nonlinear transformation of x into a feature space $\mathcal{F}$. Using the straightforward connection between a RKHS and $\mathcal{F}$ we have extended the linear PLS regression model into its nonlinear (kernel) form (Rosipal & Trejo, 2001). Effectively this extension represents the construction of linear PLS model in $\mathcal{F}$. Denote $\mathbf{\Phi}$ the $(n \times S)$ matrix of mapped input data $\Phi(x)$ into an S-dimensional feature space $\mathcal{F}$. Instead of explicit mapping of the data we can use property (4) and write

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

where $\mathbf{K}$ represents the $(n \times n)$ *kernel Gram matrix* of the cross dot products between all input data points $\{\Phi(x)\}_{i=1}^n$, i.e., $\mathrm{K}_{ij} = K(x_i, x_j)$ where $K(.,.)$ is a selected kernel function. At the beginning of the section we assumed a zero-mean regression model. To centralize the mapped data in a feature space $\mathcal{F}$ we can simply apply the following procedure (Schölkopf, Smola, & Müller, 1998; Wu, Massarat, & de Jong, 1997)

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

where $\mathbf{I}$ is an n-dimensional identity matrix and $\mathbf{1}_n$ represent the $(n \times 1)$ vector with elements equal to one. The PLS method which in its classical form is based on the nonlinear iterative partial least squares (NIPALS) algorithm finds weight vectors $\mathbf{a}, \mathbf{b}$ such that

$$[cov(\mathbf{t}, \mathbf{u})]^2 = [cov(\mathbf{\Phi a}, \mathbf{Yb})]^2 = max_{|\mathbf{r}|=|\mathbf{s}|=1}[cov(\mathbf{\Phi r}, \mathbf{Ys})]^2$$

where $cov(\mathbf{t}, \mathbf{u}) = \mathbf{t}^T\mathbf{u}/n$ denotes the sample covariance between the two score vectors (components). However, it can be shown (Höskludsson, 1988; Manne, 1987; Rännar et al., 1994) that we can directly estimate the score vector $\mathbf{t}$ as first eigenvector of the following eigenvalue problem[2]

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

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

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

After the extraction of new scores vectors $\mathbf{t}, \mathbf{u}$ the matrices $\mathbf{K}$ and $\mathbf{L} \overset{\text{def}}{=} \mathbf{YY}^T$ are deflated. The deflation of these matrices takes the form (Rosipal & Trejo, 2001)

$$\mathbf{K} \leftarrow (\mathbf{I} - \mathbf{tt}^T)\mathbf{K}(\mathbf{I} - \mathbf{tt}^T) \quad ; \quad \mathbf{L} \leftarrow (\mathbf{I} - \mathbf{tt}^T)\mathbf{L}(\mathbf{I} - \mathbf{tt}^T)$$

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

Finally, taking into account normalized scores $\mathbf{t}$ we define the estimate of the PLS regression model in $\mathcal{F}$ as (Rosipal & Trejo, 2001)

$$\hat{\mathbf{Y}} = \mathbf{KU}(\mathbf{T}^T\mathbf{KU})^{-1}\mathbf{T}^T\mathbf{Y} = \mathbf{TT}^T\mathbf{Y} \tag{8}$$

It is worth noting that different scalings of the individual Y-score vectors $\{\mathbf{u}_i\}_{i=1}^p$ do not influence this estimate. Denote $\mathbf{d}^m = \mathbf{U}(\mathbf{T}^T\mathbf{KU})^{-1}\mathbf{T}^T\mathbf{Y}^m$, $m = 1, \ldots, M$ where the $(n \times 1)$ vector $\mathbf{Y}^m$ represents the m-th output variable. Then we can rewrite the solution of the kernel PLS regression (8) for the m-th output variable as

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

which agrees with the solution of the regularized formulation of regression (2) given by the Representer theorem (5). Using equation (8) we may also interpret the kernel PLS model as a linear regression model of the form

$$\hat{g}^m(x, \mathbf{c}^m) = c_1^m t_1(x) + c_2^m t_2(x) + \ldots + c_p^m t_p(x) = \sum_{i=1}^{p} c_i^m t_i(x) \tag{9}$$

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

Although we define the scores $\{\mathbf{t}_i\}_{i=1}^p$ to be vectors in an S-dimensional feature space $\mathcal{F}$ we may equally represent the scores to be functions of the original input data x. Thus, the

[2] In Rosipal and Trejo (2001) we have also proposed the nonlinear (kernel) modification of the classical NIPALS algorithm.

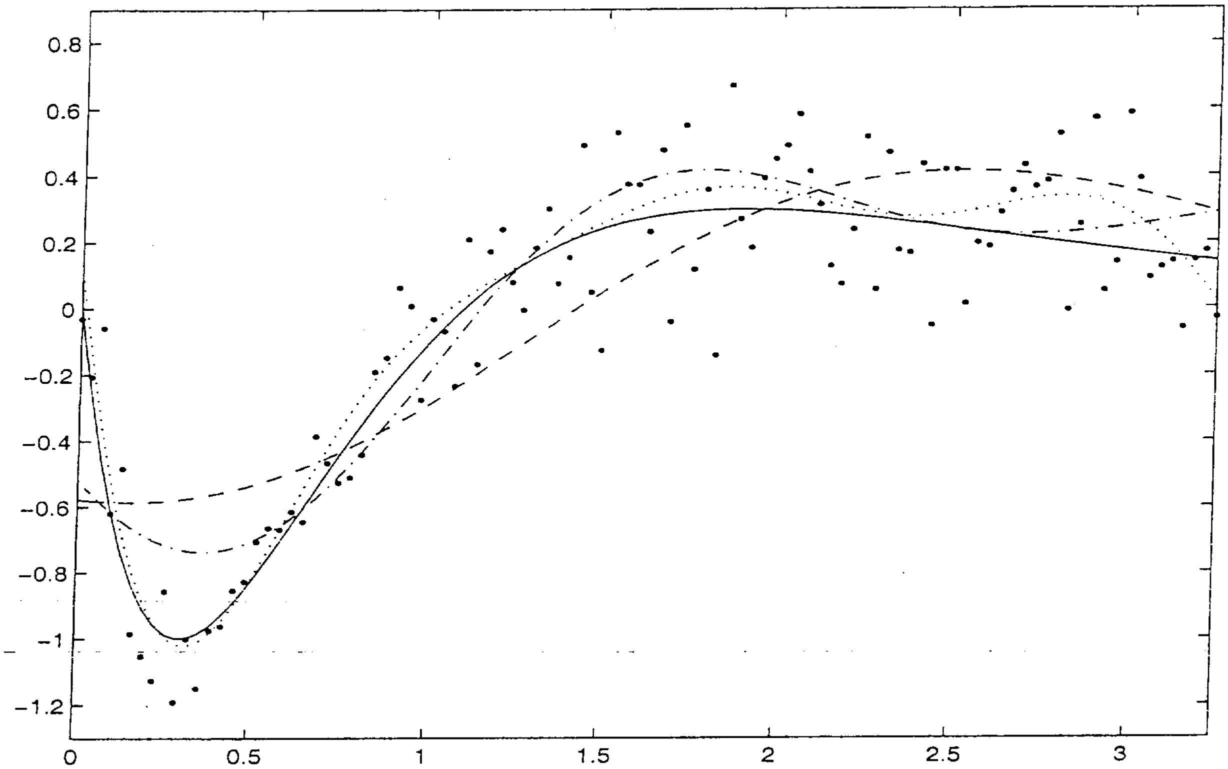

Figure 1: Smoothing of noisy $g(x) = 4.26(e^{-x} - 4e^{-2x} + 3e^{-3x})$ function. Comparison of the kernel PLS (KPLS) regression models using different numbers of components. Dotted line - KPLS with the first component (describing 64.0% of variance in input space and 66.3% variance in output space). Dash-dotted line - KPLS with the first four components (describing 99.7% of variance in input space and 77.9% variance in output space). Dotted line - KPLS with the first eight components (describing almost 100% of variance in input space and 86.7% variance in output space). The clean function is shown solid line. A noisy signal generated by adding white Gaussian noise with standard deviation equal to 0.2 is represented by dots.

proposed kernel PLS regression technique can be seen as a method of sequential construction of a basis of orthogonal functions $\{t_i(x)\}_{i=1}^{p}$ which are evaluated at the discretized locations $\{x_i\}_{i=1}^{n}$. It is important to note that the scores are sequentially extracted such that they describe overall variance in the input data space and more interestingly also describe the overall variance of the observed output data samples. In Fig.1 we demonstrate this fact on example taken from Wahba (1990). We compute the function $g(x) = 4.26(e^{-x} - 4e^{-2x} + 3e^{-3x})$ at $n = 101$ equally spaced data points x_i in the interval $\langle 0, 3.25 \rangle$ and add independently, identically distributed noise samples ϵ_i generated according to $\mathcal{N}(0, 0.04)$.

This example suggest that to more precisely model the first negative part of the generated $g(.)$ function we need to use components which generally reflect higher frequency parts of the investigated signal (see also Fig. 4). On contrary this may potentially lead to lowering the accuracy of our estimate over "smoother" parts of the signal. In our case second positive part of the $g(.)$ function. However, in many practical situations we may observe data segments where the observed function changes dramatically its curvature or even more we may have prior information about the approximate locations where the investigated function $g(.)$ changes its curvature. In the next subsection we describe the construction of locally-based kernel PLS regression models which incorporate this information.

2.3 Locally-Based Kernel Partial Least Squares Regression

To start this section we first demonstrate how the locally-based approach to kernel PLS described below may improve our estimate on the previous example. Consider that to model the first part of the generated function $g(.)$ we use the data in the range $\langle 0, 1.5 \rangle$ and remaining data will be used to model second part of the function. To avoid discontinuities on the common edge of these two models we will introduce a form of soft clustering, i.e., to construct both models we use all available data points, however, we introduce different weighting of individual data points. The used weighting functions are depicted at the top of Fig. 2. We may observe a smoother estimate over the second part of the generated g(.) function using this locally-based PLS approach in comparison to the results obtained with the global kernel PLS regression. Both estimates over the first part of the function provides comparable results. However, in the case of locally-based kernel PLS regression we used only four components extracted in each locally-based kernel PLS model in comparison to eight components (selected based on minimum mean square error on clean signal) used in the global kernel PLS model. The overall mean squared error in the case of locally-based kernel PLS regression decreased by a factor of two, as compared to the global PLS regression.

Now, we provide a more rigorous description of the method. First, we consider the soft or hard clustering of the input data and their associated outputs. We introduce a weighting function $r(x)$ which reflects importance of the point x in a kernel PLS model. The points having very small or zero values of the function $r(.)$ will effectively be excluded from the construction of orthogonal PLS basis (PLS component's extraction) for a regression model and vice-versa. The weighting functions are defined is such a way that the overall PLS model is decomposed into several local sub-models where local orthogonal PLS bases are constructed.[3] The final model is then based on a composition of the individual locally-based kernel PLS models.

Let the $(n \times 1)$ vector $\mathbf{r}$ represent the values of the weighting function $r(.)$ at the training data points $\{x_i\}_{i=1}^n$. The centralization of the $(n \times S)$ matrix of the mapped data vector $\mathbf{\Phi}$ is then given as

$$\mathbf{\Phi}_r = \mathbf{R}_d(\mathbf{\Phi} - \mathbf{1}_n \frac{\mathbf{r}^T\mathbf{\Phi}}{r_s})$$

where $r_s = \sum_{i=1}^n r_i$, $\mathbf{R}_d$ is $(n \times n)$ diagonal matrix with elements on diagonal equal to r_i, $\mathbf{I}$ and $\mathbf{1}_n$ is the identity matrix and vector of ones as defined in the previous section. This in fact represents weighted centralization of the data given by the weight vector $\mathbf{r}$. Consequently, the centralized Gram matrix will have the form

$$\mathbf{K}_r = \mathbf{\Phi}_r\mathbf{\Phi}_r^T = \mathbf{R}_d(\mathbf{I} - \frac{\mathbf{1}_n\mathbf{r}^T}{r_s})\mathbf{K}(\mathbf{I} - \frac{\mathbf{1}_n\mathbf{r}^T}{r_s})^T\mathbf{R}_d$$

Similarly, we do weighted centralization of the output data

$$\mathbf{Y}_r = \mathbf{R}_d(\mathbf{Y} - \mathbf{1}_n \frac{\mathbf{r}^T\mathbf{Y}}{r_s})$$

[3] This is analogous to the strategy used to construct the mixture of probabilistic principal components analyzers (Tipping & Bishop, 1999) where the function $r(.)$ represents a posterior *responsibilities* for generating data points x.

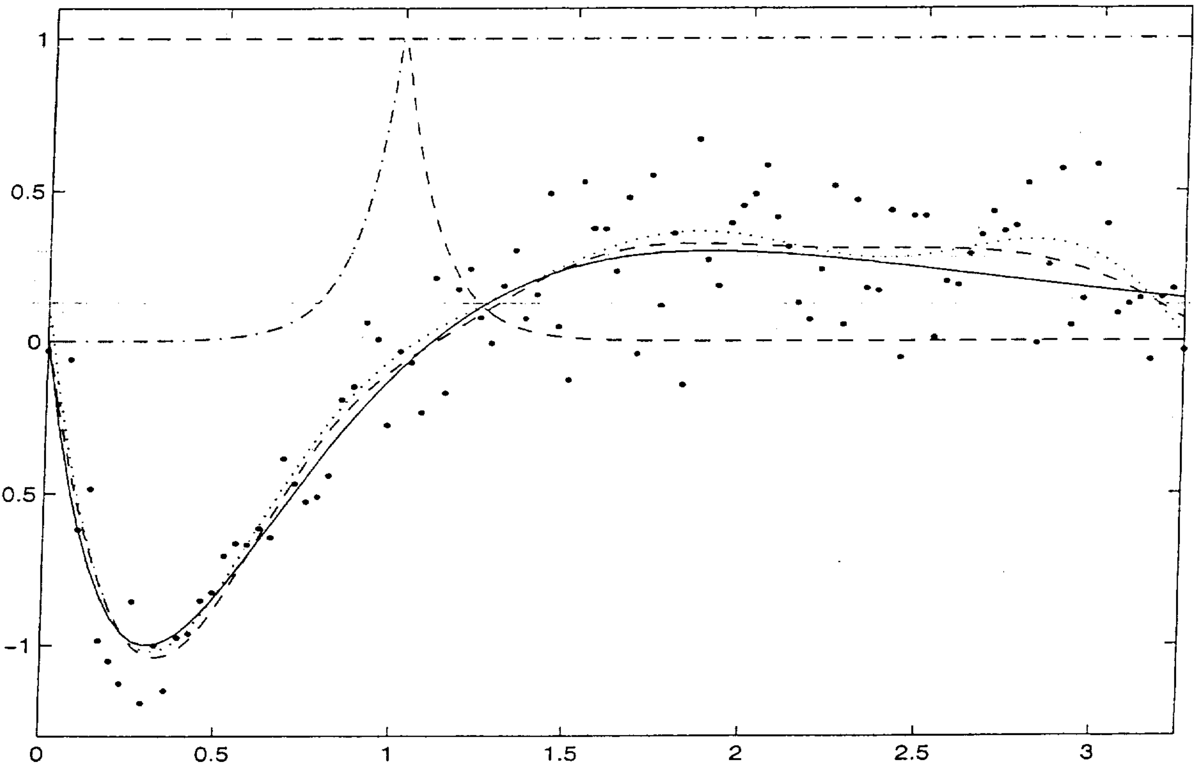

Figure 2: Comparison of the kernel PLS (KPLS) and locally-based kernel PLS (lKPLS) regression models on noisy function described in Fig. 1. Dashed line - lKPLS with the first four components in each model, upper dashed and dash-dotted lines represent the used weighting functions. Mean squared error (MSE) on clean function was equal to $1.9e^{-3}$. Dotted line - KPLS with the first eight components. MSE on clean function was equal to $4.3e^{-3}$. The clean function is shown solid line. Noisy signal generated by adding white Gaussian noise with standard deviation equal to 0.2 is represented by dots.

Consider that we define Z clusters based on which Z locally-based kernel PLS models are constructed. Define centralized Gram matrix $\mathbf{K}_r^z$ constructed using the weight function $r^z(.)$ associated with the z-th cluster. Similarly define matrix $\mathbf{L}_r^z \overset{\text{def}}{=} \mathbf{Y}_r^z(\mathbf{Y}_r^z)^T$. Following (6) and (7) the scores $\mathbf{t}^z, \mathbf{u}^z$ of the z-th kernel PLS model are then given by solving the eigenvalue problem

$$\mathbf{K}_r^z \mathbf{L}_r^z \mathbf{t}^z = \lambda^z \mathbf{t}^z$$

and by

$$\mathbf{u}^z = \mathbf{L}_r^z \mathbf{t}^z$$

After each step we have to deflate $\mathbf{K}_r^z$ and $\mathbf{L}_r^z$ matrices in the same way as described in the previous section. Denoting by $\mathbf{T}^z$ and $\mathbf{U}^z$ the matrices with columns consisting from the extracted $\mathbf{t}^z, \mathbf{u}^z$ scores the kernel PLS regression estimate for the z-th cluster is given as

$$\hat{\mathbf{Y}}_r^z = \mathbf{T}^z(\mathbf{T}^z)^T \mathbf{Y}_r^z$$

To express this estimate in the original not centralized variables we can write

$$\hat{\mathbf{Y}}^z = \mathbf{R}_d^{-1} \hat{\mathbf{Y}}_r^z + \mathbf{1}_n \frac{(\mathbf{r}^z)^T \mathbf{Y}}{r_s^z}$$

where r_s^z is the sum of the elements of the weighting vector $\mathbf{r}^z$ defined for the z-th cluster by the weight function $r^z(.)$. To be consistent with our previous notation we denote by

$$\hat{g}^m(x_i)^z \overset{\text{def}}{=} (\hat{Y}_i^z)^m \,;i = 1,\ldots,n \,;m = 1,\ldots,M \,;z = 1,\ldots,Z$$

the locally-based kernel PLS estimate for the z-th cluster for the m-the output variable at the data point x_i.

The final locally-based kernel PLS regression model consists of the weighted summation of Z individual local kernel PLS regression estimates. This estimate for the input point x_i is given as

$$\hat{g}^m(x_i) = \sum_{z=1}^{Z} r_i^z \hat{g}^m(x_i)^z / \sum_{z=1}^{Z} r_i^z \;\; ;i = 1,\ldots,n \;\; ;m = 1,\ldots,M$$

where $\{r_i^z\}_{i=1}^n$ are the elements of the weighting vector $\mathbf{r}^z$.

Finally, let us make several comments on the proposed locally-based kernel PLS methodology:

a) First we have to stress that we defined the weighting function $r(.)$ based on our prior knowledge, visual inspection of the noisy data, detection of the segments with significant change of the curvature, etc. The obvious question which may occur is how the segmentation (clustering) of the input data will be "transformed" into the clustering of the data in a feature space $\mathcal{F}$. This is an important issue due to the fact that we consider local PLS in $\mathcal{F}$, not in the original input space. We may believe that we can invoke good clustering in a feature space $\mathcal{F}$ if the nonlinear mapping to $\mathcal{F}$ will be smooth and will preserve topological property of the original input data. The Euclidian distance between two points in $\mathcal{F}$ is given as

$$\|\Phi(x_i) - \Phi(x_j)\|^2 = \langle \Phi(x_i), \Phi(x_i) \rangle - 2\langle \Phi(x_i), \Phi(x_j) \rangle + \langle \Phi(x_j), \Phi(x_j) \rangle$$

If we consider a Gaussian kernel function $K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{h}\right)$ we may write

$$\|\Phi(x_i) - \Phi(x_j)\|^2 = 2 - 2\exp\left(-\frac{\|x_i - x_j\|^2}{h}\right)$$

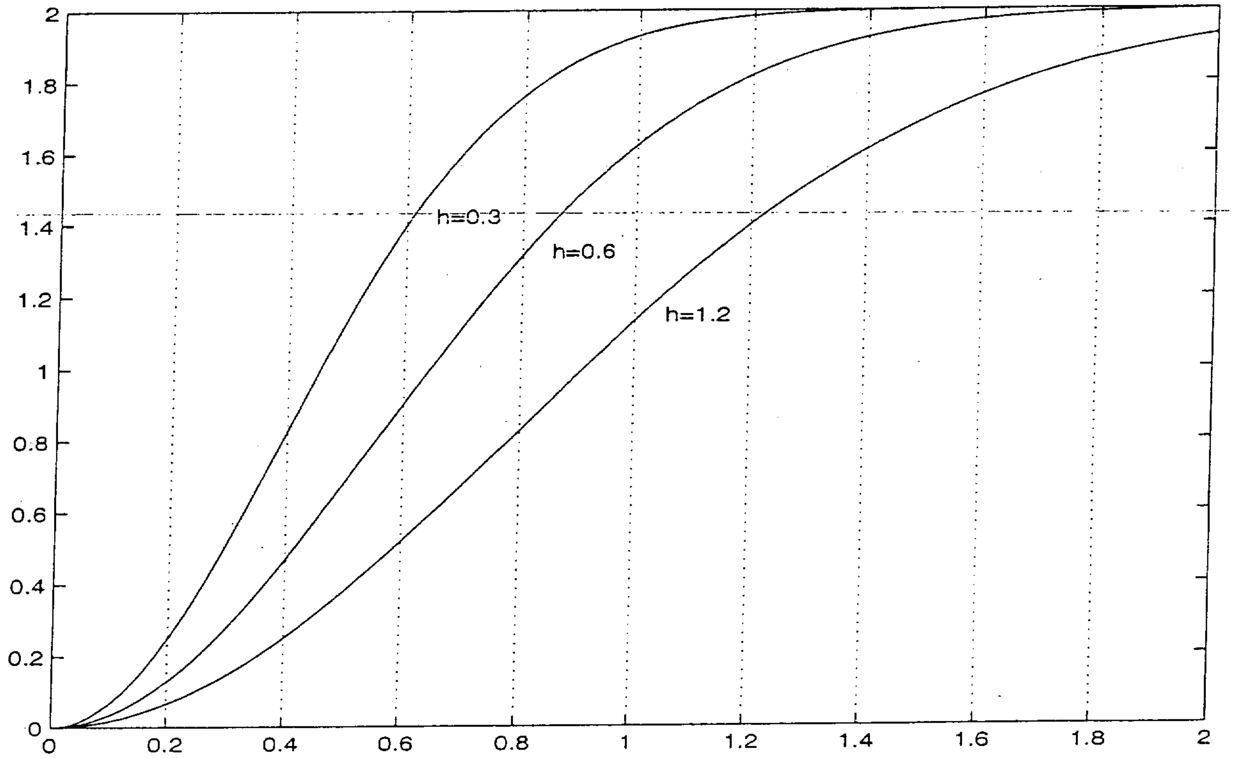

Figure 3: Dependence of the Euclidian distances between the mapped data points on the input space distances. Three curves represent the cases with different widths (h) of the Gaussian kernel used.

and we can plot this distance as the function of the Euclidian distance between the input points $\|x_i - x_j\|$ and the width of Gaussian kernel h (Fig. 3).

This plot suggest that using a wider Gaussian kernel function we can not successfully localize very small segments of input data. Data corresponding to small distances in input space will be "flocked" together in $\mathcal{F}$. In opposite for a smaller width of the Gaussian kernel function the input data with greater distances will be too "spread" in $\mathcal{F}$ and intended localization may be lost. The graphs suggest that for the width of the Gaussian kernel equal to 0.6 (this value have been used in the experiments described in the Section 4) segmentation of the input data with distances greater then 0.1 and less then 1.4 should result in appropriate data clustering in $\mathcal{F}$.

b) As we might already observe the (locally-based) kernel PLS method iteratively extracts orthogonal components describing overall variance of the input as well as output data. This effectively creates a sequence of orthogonal functions with increasing "complexity". Here we define the complexity in the sense that the first kernel PLS scores will pick up the trend of the output function and will represent rather smooth slowly-varying functions. In contrast higher scores will represent higher frequency components of the output signal or noise. We note that this hypothesis in general will depend on the selected kernel function. However, in our setting we have used the Gaussian kernel function which has a nice smoothing property based on the fact that higher frequency components are suppressed (Girosi, Jones, & Poggio, 1995; Schölkopf & Smola, 2002). We may hypothesize that the above argument will be true for such a class of "smooth" kernel functions. In Fig. 4 we demonstrate this argument using first nine score functions extracted in previous example where we have used Gaussian kernel with a width h equal to 1.8.[4]

c) Described (local) kernel PLS regression is well defined for univariate as well as multivariate outputs scenario. In the case of multivariate outputs this approach opens the possibility for spatio-temporal modeling of a collection of signal of interest. This can be useful in the case that observed signals represent different time realizations or measurements at different spatial locations. Or also in the situations where both, different time and spatial measurements of the same or similar signals of interest $g(.)$ are collected. We simply arrange the output matrix $\mathbf{Y}$ to be the matrix with columns representing these different temporal or spatial signal(s) of interest. The extracted components will represent common features of these different realizations. Finally, we would like to note that the "kernel trick" (4) allows us to easily extend the methodology to the case of multidimensional inputs and also for the case of not equally sampled data.

d) As we noticed in the previous example the locally-based kernel PLS approach provided (in terms of MSE) a better estimate of the generated function $g(.)$. This was achieved with smaller number of four different components used in individual local kernel PLS models. On several different data sets we experimentally observed that to properly smooth the noisy data over the segments with low curvature of $g(.)$ individual locally-based kernel PLS needed less then 8-10 components. In many case less than five components provided best results. This was observed on different smoothing problems described in the paper and other non-published data sets as well. However, results in Rosipal and Trejo (2001) indicate that if the regression task with smaller level of output noise is of interest this number may increase. In the current

[4] We just remind the reader that the kernel PCA decomposition without any connection to the output function would lead to the extraction of the components similar to a trigonometric series expansion for a Gaussian kernel (Schölkopf & Smola, 2002; Rosipal & Trejo, 2001).

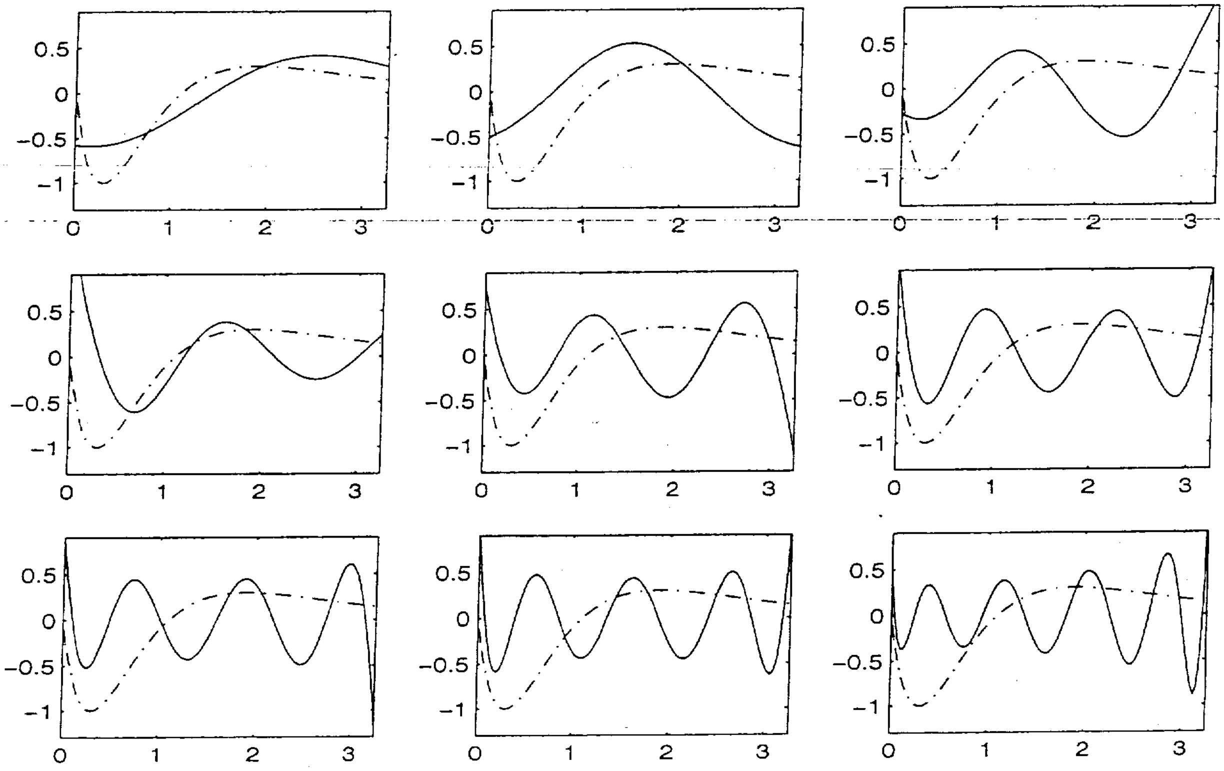

Figure 4: The first nine components (X-space scores) computed from noisy signal described in Fig. 1. The clean function is shown dash-dotted.

paper we provide results where the maximum number of components was restricted to be the first four components. The final number $p \leq 4$ of components was selected using the model selection criterion described in the next section.

2.3.1 Model Selection

We have already noticed that kernel PLS extracts components with increasing complexity in the sense of remark b) in the previous section. This construction of a sequence of functions with increasing complexity also motivates our use of the idea of Vapnik's *structural risk minimization* approach for model selection, i.e., in our case this is the determination of the number of components p used in each locally-based kernel PLS regression model. Vapnik has shown that for regression problems with a squared loss function the following bound on an estimate of in-sample prediction error (PE) holds with probability $1-\eta$ (Vapnik, 1998)

$$PE \leq \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{g}(x_i))^2 \left(1 - c\sqrt{\frac{h(\ln(\frac{an}{h})+1) - \ln\eta}{n}}\right)_{+}^{-1} \tag{10}$$

where h is VC dimension of the set of approximating functions, c is a constant reflecting the "tails of the loss function distribution", a is a theoretical constant and

$$\begin{array}{rl} (x)_{+} = & x \ \ if \ \ x > 0 \\ & 0 \ \ otherwise \end{array}$$

The first term on right hand side of (10) represents *empirical error* while the second term is often called *penalization factor*, which for increasing complexity of the regression model inflates empirical error.

For practical use it is difficult to compute the exact VC dimension for an arbitrary set of functions, moreover, it can be infinite for some classes of functions. However, constructing a regression function to be a linear combination of a finite (fixed) set of basis functions, ordered based on the increasing complexity, Cherkassky et. al. (1999) and Cherkassky and Shao (2001) suggested to take the following heuristic penalization factor

$$\left(1 - \sqrt{v - v\ln v + \frac{\ln n}{2n}}\right)_{+}^{-1}$$

where $v = p/n$ with p representing VC dimension of the considered regression function (9) with p terms. To complete this replacement Cherkassky et. al. (1999) and Cherkassky and Shao (2001) set $\eta = 1/\sqrt{n}$ and they have considered parameters a, c to be equal one.[5] In comparison with other model selection criteria, it was demonstrated that the new model selection criterion motivated by the structural risk minimization theory and VC dimension may provide comparable or better results (Cherkassky et al., 1999; Cherkassky & Shao, 2001).

2.4 Univariate Polynomial Smoothing Splines

We again consider model (1) and we further assume that function $g(.)$ belongs to the Sobolev Hilbert space

$$W_2^c[0,1] = \{f : f^{(\eta)} \text{ abs. cont.}, \eta = 1, \ldots, c-1; \int_0^1 (f^{(c)}(x))^2 dx < \infty\}$$

[5] For more detail description of this reasoning see Cherkassky et al. (1999) and Cherkassky and Shao (2001).

which determines the smoothness properties of $g(.)$. Without lack of generality we consider input data to be in the interval $[0, 1]$. Wahba (1990) has shown that the same general framework for the solution of regularized functional (2) in a RKHS can also be applied for smoothing splines. More precisely, the RKHS is a Sobolev Hilbert space and the penalty functional $\Omega(g)$ is the semi-norm of the space. Thus the smoothing splines are the solution of regularized functional

$$\hat{g}(x) = \arg \min_{g \in W_2^c[0,1]} \left[\frac{1}{n} \sum_{i=1}^{n} (y_i - g(x_i))^2 + \gamma \int_0^1 (f^{(c)}(x))^2 dx \right] \tag{11}$$

and they have the following proprieties

$$\hat{g}(x) = \begin{cases} \pi^{c-1} & \text{for } x \in [0, x_1] \\ \pi^{2c-1} & \text{for } x \in [x_j, x_{j+1}] \\ \pi^{c-1} & \text{for } x \in [x_n, 1] \\ C^{2c-2} & x \in [0, 1] \end{cases}$$

where π^k are polynomials of degree k and C^k represents functions of k continuous derivatives. The smoothness of the solution is controlled through the parameter $\gamma > 0$. In this paper we consider $c = 2$ in which case (11) has the unique solution called *natural cubic splines*. Adapting theory of RKHS and the Representer theorem we can write the solution of (11) in the form

$$\hat{g}(x) = b_1 v_1(x) + b_2 v_2(x) + \sum_{i=1}^{n} c_i K(x, x_i) \tag{12}$$

where $v_1(x) = 1,\ v_2(x) = x - 1/2$ and

$$\begin{aligned} K(x, x_i) = \ & \tfrac{1}{4}((x - \tfrac{1}{2})^2 - \tfrac{1}{12})((x_i - \tfrac{1}{2})^2 - \tfrac{1}{12}) \\ & -((|x - x_i| - \tfrac{1}{2})^4 - \tfrac{1}{2}(|x - x_i| - \tfrac{1}{2})^2 + \tfrac{7}{240})/24 \end{aligned} \tag{13}$$

To estimate the vectors of unknown coefficients $\mathbf{b} = (b_1, b_2)^T, \mathbf{c} = (c_1, c_2, \ldots, c_n)^T$ we need to solve the following optimization problem

$$\arg \min_{\mathbf{b},\mathbf{c}} \left[\frac{1}{n} \|\mathbf{y} - (\boldsymbol{\Upsilon}\mathbf{b} + \mathbf{K}\mathbf{c})\|^2 + \gamma \mathbf{c}^T \mathbf{K} \mathbf{c} \right]$$

where $\mathbf{y} = (y_1, \ldots, y_n)^T, \mathbf{K}$ is the $(n \times n)$ Gram matrix with (i, j)-th entry to be equal $K(x_i, x_j)$ of (13) and $\boldsymbol{\Upsilon}$ is the $(n \times 2)$ matrix with (i, j)-th entry $v_j(x_i)$. It can can be further shown (Wahba, 1990; Green & Silverman, 1994; Hastie, Tibshirani, & Friedman, 2001) that we can express this solution in the form

$$\hat{\mathbf{g}} = \mathbf{S}_\gamma \mathbf{y}.$$

where $\hat{\mathbf{g}}$ denotes $(n \times 1)$ vector of fitted values $\hat{g}(x_i)$ at the training input data points $\{x_i\}_{i=1}^n$ and $\mathbf{S}_\gamma$ is the *smoother matrix* which depends only on $\{x_i\}_{i=1}^n$ and γ.

In the case of independent, identically normally distributed errors ϵ_i the minimum of *generalized cross-validation* (GCV) function was proved to provide a good estimate for γ (Wahba, 1990)

$$\mathrm{GCV}(\gamma) = \frac{1}{n} \sum_{i=1}^{n} [\frac{y_i - \hat{g}(x_i)}{1 - trace(\mathbf{S}_\gamma)/n}]^2$$

Trace of the smoother matrix $\mathbf{S}_\gamma$ defines *effective degrees of freedom* $df_\gamma = trace(\mathbf{S}_\gamma)$ (Hastie, Tibshirani, & Friedman, 2001). Having defined the df_γ of the smoothing spline we may numerically derive the corresponding γ parameter. This parametrization may provide us a more intuitive comparison of different regularization of smoothing splines in practice, e.g., in the case we would like to visually compare smoothness of the fit.

2.4.1 Hybrid Adaptive Splines

To deal with a possible local inhomogeneity in $g(.)$ function Luo and Wahba (1997) proposed new methodology arising from the solution (12). The methodology is based on approximation of the Gram matrix $\mathbf{K}$ by a set of basis functions which represent different columns of $\mathbf{K}$. The motivation to do this is based on the assumption that the regions of local inhomogeneity or data dense regions in the case of unequal sampling should be modeled with a higher number of basis functions in comparison to smooth or more uniformly sampled regions. The individual columns of $\mathbf{K}$ are sequentially added to the model based on their reduction in the residual sum of squares. The size of the model, i.e., the number of columns added, is determined using the following modified generalized cross-validation (mGCV) criterion

$$\mathrm{mGCV}(k) = \frac{1}{n}\sum_{i=1}^{n}[\frac{y_i - \hat{g}(x_i)}{(1-(2+(k-2)\mathrm{IDF})/n)}]^2$$

where IDF is the inflated degrees of freedom parameter (Luo & Wahba, 1997). Taking the k which minimizes mGCV the final HAS model will have the form

$$\hat{g}(x) = b_1 v_1(x) + b_2 v_2(x) + \sum_{l=1}^{k} c_i K(x, x_{i_l})$$

where index i_l represents the selected columns of $\mathbf{K}$. This methodology is close in its strategy of selection of a subset of basis functions defined by the columns of the Gram matrix $\mathbf{K}$ to the different approximations of regularization networks (Haykin, 1999; Chen, Cowan, & Grant, 1991; Orr, 1995; Platt, 1991) and also to the recently developed relevance vector machines (Tipping, 2001). As was pointed out by the authors if the signal does not have much local inhomogeneity smoothing splines with a global smoothing parameter should in general perform better than the HAS approximation. To find the vectors of coefficient $\mathbf{b} = (b_1, b_2)^T, \mathbf{c} = (c_1, c_2, \ldots, c_k)^T$ the authors suggest to used penalized regression due to the possible higher correlations among the selected columns of $\mathbf{K}$. This may numerically stabilize the least squares estimate of the regression coefficients and decrease their variance, however, it will also lead to the bias in the estimate and also to the potential further smoothing. Although this may provide a regression model with better generalization properties due to the lower variance of the regression coefficients, our goal here is to smooth measured signals and we do not have a reason to use this further penalization. We have used unpenalized ordinary least squares approach with pseudo-inverse to compute desired regression coefficients $\mathbf{b}, \mathbf{c}$.

3 Data Construction

3.1 Heavisine function

To examine the performance of locally-based kernel PLS method we selected a standard example of de-noising heavisine function taken from Donoho and Johnstone (1995)

$$g(x) = 4sin(4\pi x) - sign(x - 0.3) - sign(0.72 - x)$$

The function of a period one has two jumps at $x_1 = 0.3$ and $x_2 = 0.72$. The function scaled into interval $[-1, 1]$ is depicted in Fig. 8b. We computed heavisine function on five sets of equally spaced data points with number of samples 124, 256, 512, 1024 and 2048, respectively. We added Gaussian noise of two levels such that signal-to-noise ratio (SNR) was 1dB and 5dB, respectively.[6] We created 50 different replicates of the noisy heavisine function for each set of different lengths and noise levels.

3.2 Event Related Potentials

We simulated brain event related potentials (ERP) and ongoing electroencephalogram (EEG) activity using the dipole simulator program of BESA software package.[7] In this scenario we consider EEG to represent spatially and temporally correlated noise added to the ERP. This simulation provides a reasonable but simplified model of real-world ERP measurements.

The sources of generated ERP were represented by five dipoles with different spatial location and orientation. The placement of the dipoles with their sample activation function is presented in Fig. 5. The dipoles that contribute to the ERP produce a composite waveform with four prominent peaks: N1 (negative peak with the latencies in the range 100-140 ms), P2 (positive peak with the latencies in the range 200-240 ms), N2 (negative peak with the latencies in the range 250-300 ms) and P3 (positive peak with the latency in the range 340-400 ms) as observed on C_z electrode. These four peaks correspond to well known components of human auditory or visual ERP (Hillyard & Kutas, 1983; Naatanen & Picton, 1987; Naatanen & Picton, 1986; Parasuraman & Beatty, 1980; Picton et al., 1974). We generated 20 different realizations of ERP on scalp by random changing of the latency of peaks and amplitudes of the individual activation functions. These dipoles were used to generate a full scalp topography containing 19 data channels located based on International 10-20 System (Jasper, 1958). Average of the signal at mastoid electrodes A1 and A2 was used as reference signal for other electrodes. The sample of ERP for one of 20 different trials is also shown in Fig. 5. Individual data epochs were designed to be 800 ms long starting 100 ms before the event. We have used a sampling rate equal to 160Hz, 320Hz and 640Hz resulting into $128, 256$ and 512 data points long epochs. In the next step coherent noise modeling of ongoing EEG activity was added to the generated ERP. The noise data waveforms are summed over the contributions from 200 different sources (dipoles) with random locations and orientations. The noise is generated in a way that there is a high correlation between signal amplitudes from close electrodes. The frequency characteristic reflects characteristics of EEG spectra $(1/\sqrt{f+1})$ with added dominant α-band frequency about 10Hz. The weighting of the α-band in comparison to the other frequencies was changed in each realization but the proportion of the α-band stays in the range of $40\% - 60\%$. Two different levels of the amplitudes of the noise signal waveforms were set. This created two sets of noisy ERP with averaged SNR over the electrodes and trials to be equal 5dB and -1.5dB, respectively. The same sampling rates as described above were used. Finally, temporally and spatially uncorrelated Gaussian noise was added to the generated data to represent measurement noise and other non-biological sources. For the first set of the data the zero-mean noise with standard deviation equal to 0.5 was generated. In the second case standard deviation of the white Gaussian noise was increased to 1. The noise was generated for each channel individually and referenced to average of A_1 and A_2 electrodes.

[6] In all our experiments we define SNR to be $10\log_{10}\frac{S}{N}$, where $S = \sum_{i=1}^{n} g(x_i)^2$ is a sum of squared amplitudes of clean signal and $N = \sum_{i=1}^{n} \epsilon_i^2$ is the sum of squared amplitudes of noise, respectively.

[7] `http://www.besa.de`

This resulted in final averaged SNR over electrodes and trials to be equal 1.3dB and −4.6dB, respectively.

In our second set of noise ERP we added uncorrelated zero-mean Gaussian noise with standard deviation equal to 0.5, 1 and 2, respectively, to the clean ERP. This represents averaged SNR over the electrodes and trials to be equal 4.5dB, −1.6dB and −7.5dB.

4 Experiments

To compare our results we have used two measures of goodness of fit to the clean signal $g(.)$. Normalized root mean squares error (NRMSE) defined as

$$\text{NRMSE} = \sqrt{\frac{\sum_{i=1}^{n}(g(x_i) - \hat{g}(x_i))^2}{\sum_{i=1}^{n}(g(x_i) - \bar{g}(x))^2}}\ ,\ \bar{g}(x) = \frac{1}{n}\sum_{i=1}^{n} g(x_i)$$

The second measure we have used is the correlation coefficient (CC) between de-noised signal $\hat{g}(.)$ and clean generated signal $g(.)$ Although both measures can provide good intuition about the goodness of fit and similarity of the shapes between our estimate of the signal of interest and the signal itself, they reflect mainly overall global characteristics of the fit. Thus we have also visually inspected smoothness of the individual estimates $\hat{g}(.)$ and goodness of fit over the parts of spatial inhomogeneities in the signal $g(.)$.

4.1 Heavisine function

We compared the proposed locally-based kernel PLS (lKPLS) method with existing smoothing splines (SS), hybrid adaptive splines (HAS) and Wavelet Shrinkage (WS) methods.

In the case of lKPLS we used equally spaced data in the interval $[-1, 1]$. The theoretical value of two times the variance of uniformly distributed data in $[-1, 1]$ equals $0.\bar{6}$ and in all our experiments with the Gaussian kernel function this motivates our choice of the width h to be equal 0.6. Visual inspection of a sample of noisy heavisine function (Fig. 8a) suggests to set the weighting function $r(.)$ over four segments (intervals) $\mathcal{S}_1$ =([-1,-0.5],[-0.5,0],[0,0.5],[0.5,1]). These segments reflect four "bumps" visible on the noisy function over which individual local kernel PLS models are constructed. We set $r(.)$ to be equal to 1 over the individual segments. The segments are overlapped with exponentially decaying values in a way that it reaches values close to zero in the middle of the neighborhood segment. In the second case we assume that the approximate location of the local inhomogeneity close to the point 0.45 is known in advance. We rearrange four segments to be $\mathcal{S}_2$ =([-1,-0.5],[-0.5,0.4],[0.4,0.5],[0.5,1]) and set the weighting function $r(.)$ to be equal to 1 over the individual segments. We have used the same decay of $r(.)$ as in the first case to overlap the segments.

In our comparison we have investigated different WS methods as implemented in the Matlab Wavelet Toolbox.[8] Comparing the results in the terms of NRMSE and CC on recovery of the known clean heavisine function we examined different families of wavelets, different number of wavelet decomposition levels and different threshold selection rules as implemented in `wden` function of the Wavelet Toolbox. The best results were achieved and are reported using Daubechies 4 family of wavelets, heuristic SURE as threshold selection rule, threshold rescaling using a single estimation of level noise based on the first-level of coefficients and soft

[8] `http://www.mathworks.com/`

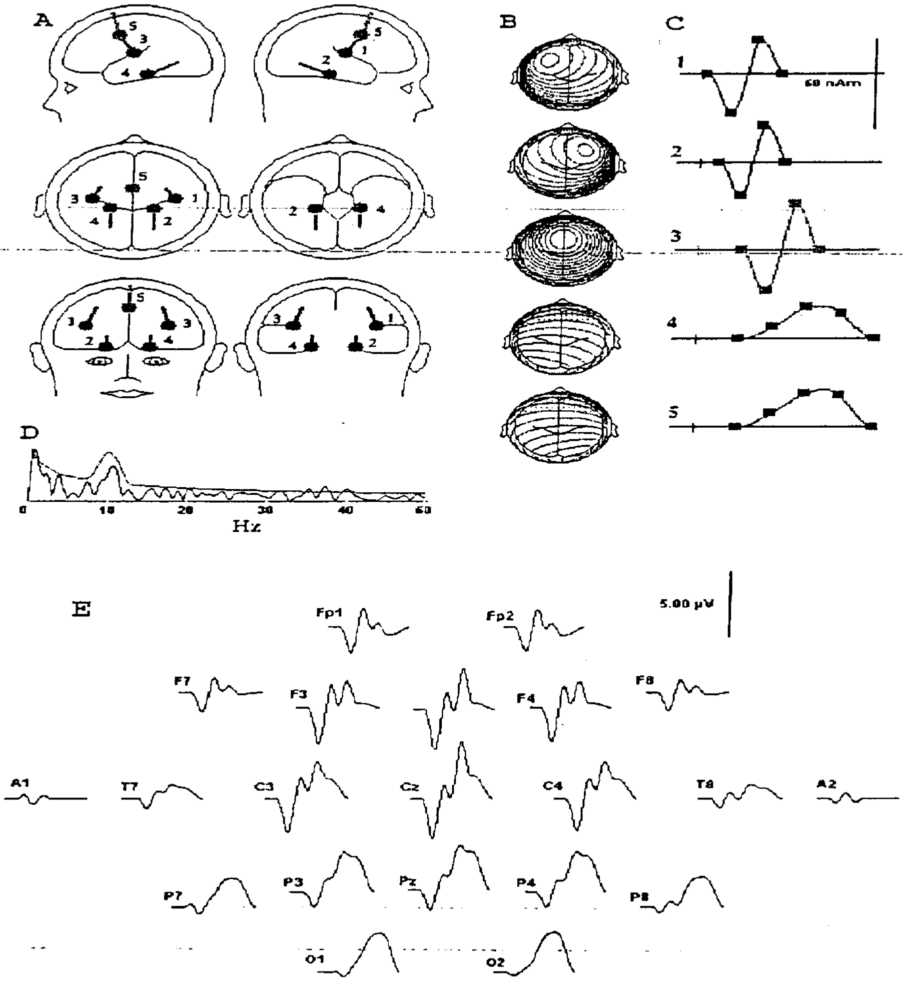

Figure 5: Simulation ERP data. A: Location and orientation of dipoles. B: Topographical scalps maps showing topography of each dipole C: Activation function of each dipole. D: The averaged (over electrodes) noise amplitude spectrum and the weighting function of an sample of noise added to ERP. The spectra simulate ongoing EEG activity with dominant α-band frequency to be 50% of the spectra. E: Simulated ERP waveshapes on different electrodes (scalp locations).

thresholding of details coefficients at level $N-1$, where N is maximal number of decomposition levels as defined by the number of data points used. These settings are in good agreements with findings of Donoho and Johstone (1995).

The computation of SS was done using the penalized smoothing splines (pspline) package as implemented in R software.[9] We set the order of splines to correspond to natural cubic splines. We observed a little difference between the use of GCV and ordinary CV criterion for the selection of smoothing parameter γ. We report the results using GCV criterion. HAS were implemented in Matlab based on the original paper of Luo and Wahba (1997). The individual basis functions were added based on minimizing mGCV with IDF equal to 1.2.

In Fig. 6 and Fig. 7 a boxplot with lines at the lower quartile, median, and upper quartile values and a whisker plot of the distribution of NRMSE over different runs corresponding to the case of SNR equal to 1dB is depicted. The triplets correspond to the results achieved with the use of different number of samples. The first boxplot in a triplet is for lKPLS, second for SS and third for WS. Fig. 6 represents results with $\mathcal{S}_1$ setting of segments in lKPLS. We may see that SS provides the best results followed by lKPLS. With increasing number of samples the differences among the methods are decreasing. We have to note that using more widely overlapping segments in the case of lKPLS improved results for the case of 128 and 256 samples and provided the results almost identical to SS. In the case of smaller number of samples individual lKPLS models are constructed using the segments of smaller number of data points with higher weights. Thus, increasing the segments of influential data points (slower decay of the tails of the weight function $r(.)$) will have tendency to provide more global smooth estimate, however, it also leads to loosing the tendency to cope with local inhomogeneities. The results achieved using the second set $\mathcal{S}_2$ of segments in the case of lKPLS are depicted in Fig. 7. We may see improvement of lKPLS results for all number of samples. For the higher number of samples (512,1024,2048) lKPLS provides slightly better results in terms of median, upper and lower quartiles than SS ans WS. Again using slower decay of the tails of the weight function $r(.)$ provided comparable results to SS in the case of of 128 and 256 samples used.

The comparison of the WS, SS and lKPLS methods for the case of SNR equal to 5dB provided qualitatively the same results as described for SNR equal to 1dB, that is, comparable performance of SS and lKPLS followed by slightly worse performance of WS. Intuitively, the values of NRMSE using individual methods were decreased.

In terms of CC we also observed qualitatively the same results as described for NRMSE. In the case of SNR equal to 1dB the median of CC for lKPLS and SS was in the range between $0.975 - 0.995$ with smaller values starting for the case of smaller number of samples used. Values of the median of CC for WS were for smaller number of samples used slightly smaller. In the case of SNR equal to 5dB the range of median values of CC for lKPLS and SS was between $0.985 - 0.997$ corresponding to the high "agreement of shapes" between estimates $\hat{g}(.)$ and the clean heavisine function $g(.)$.

In Fig. 8c we plotted an estimate of $\hat{g}(.)$ for which lKPLS fit $g(.)$ with NRMSE closest to median. Estimate using the same data sample however second set of the segments $\mathcal{S}_2$ is plotted in Fig. 8d. In Fig. 8e,f we have also plotted estimates of SS and WS with NRMSE closest to this lKPLS estimate. The visual inspection suggests that lKPLS with second set of segments $\mathcal{S}_2$ provides the best fit to the clean heavisine function with clear detection of the local inhomogeneity close to the point 0.45 and shows best overall smoothness of the estimate

[9] `http://cran.r-project.org/`

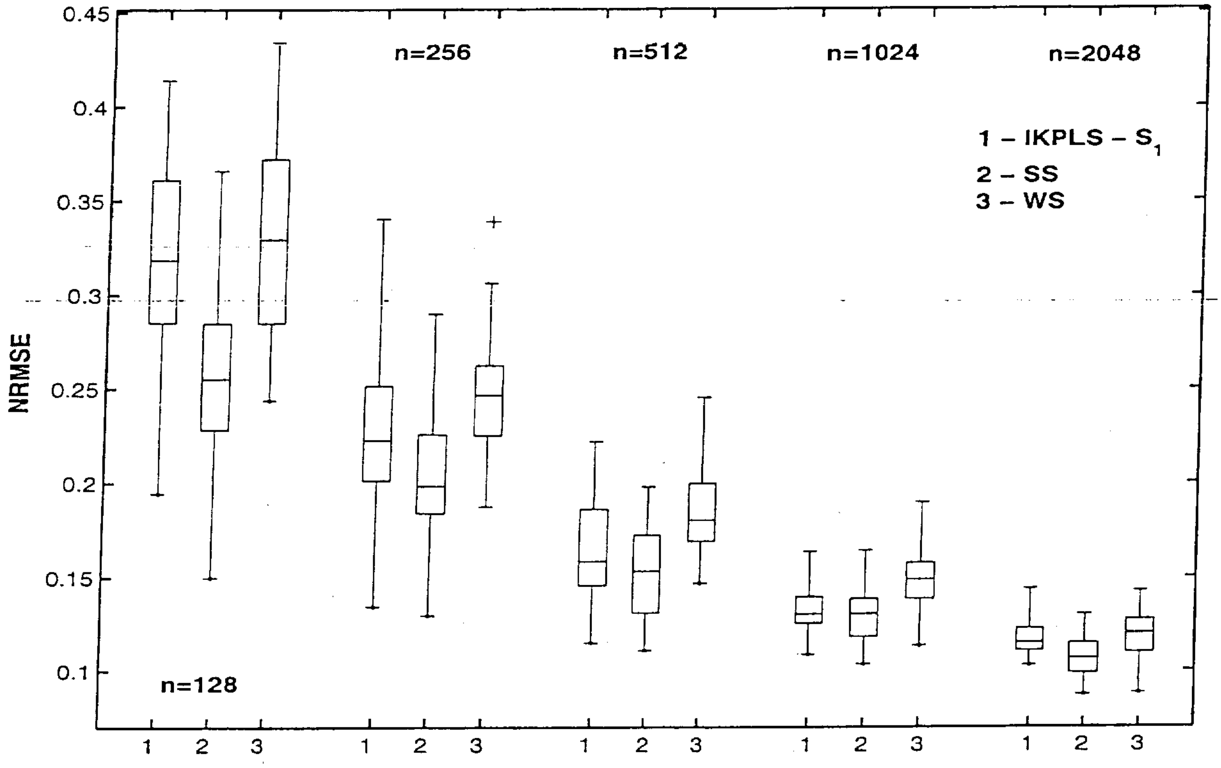

Figure 6: Results on heavisine function. Boxplots with lines at the lower quartile, median, and upper quartile values and whisker plot for three different nonparametric smoothing methods and different number (n) of samples used. The performance of locally-based kernel PLS using set of segments $\mathcal{S}_1$ (lKPLS-$\mathcal{S}_1$, left-hand boxplots in the individual triplets) is compared with smoothing splines (SS, middle boxplots in the triplets) and wavelet shrinkage (WS, right-hand boxplots in the triplets) in terms of normalized root mean squared error (NRMSE). The boxplots are computed on results from 50 different replicates of noisy heavisine function (SNR=1dB).

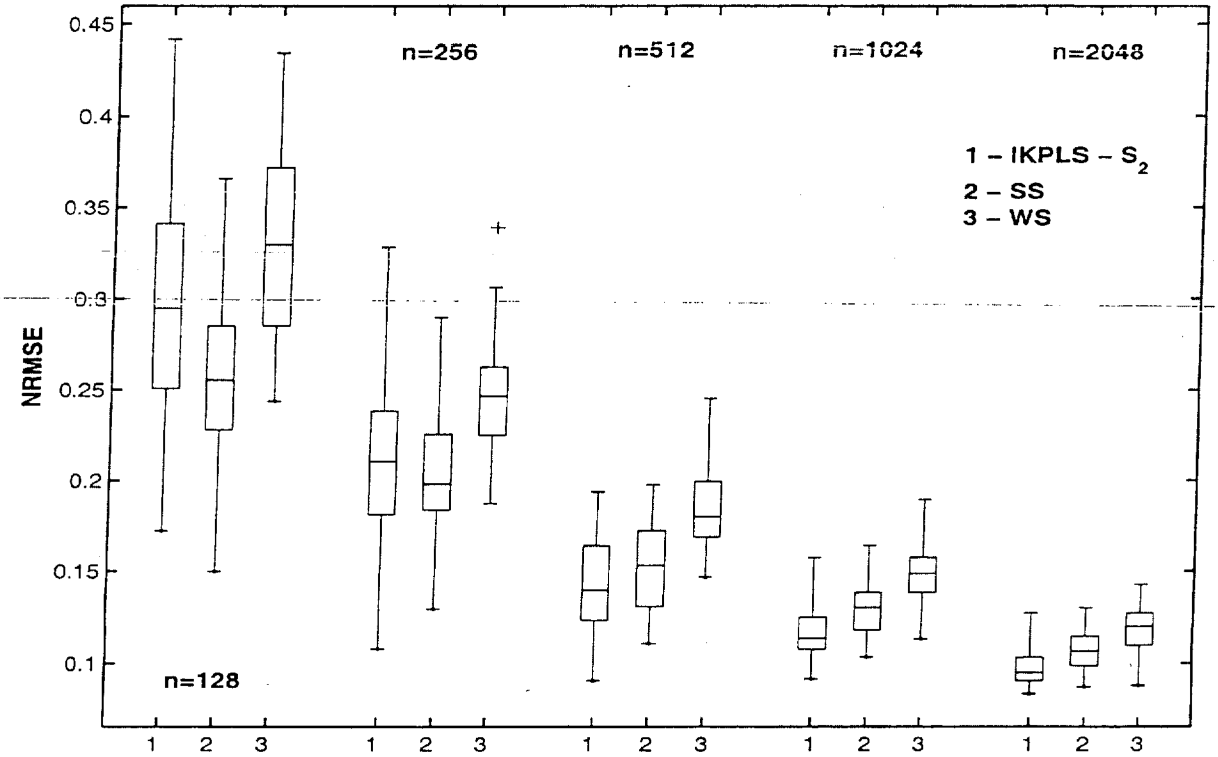

Figure 7: Results on heavisine function. Boxplots with lines at the lower quartile, median, and upper quartile values and whisker plot for three different nonparametric smoothing methods and different number (n) of samples used. The performance of locally-based kernel PLS using set of segments $\mathcal{S}_2$ (lKPLS-$\mathcal{S}_2$, left-hand boxplots in the individual triplets) is compared with smoothing splines (SS, middle boxplots in the triplets) and wavelet shrinkage (WS, right-hand boxplots in the triplets) in terms of normalized root mean squared error (NRMSE). The boxplots are computed on results from 50 different replicates of noisy heavisine function (SNR=1dB).

in comparison to SS and WS.

We further compared the SS and HAS methods. Overall performance in terms of median NRMSE and CC of HAS was worse in comparison to SS.

We visually inspected all 50 trials in the case of 1024 samples and both SNR rates. In the case of SNR equal to 5dB (1dB) we could visually confirm the detection of local inhomogeneity close to the point 0.45 in 23 (16) cases for HAS, 11 (8) cases for SS, 30 (12) cases for WS and 44 (34) cases for lKPLS using $\mathcal{S}_2$ set of segments. The overall smoothness of lKPLS and its fit to the clean heavisine signal has been evidently higher in comparison to HAS and WS. In terms of median NRMSE, lKPLS provided improvement of 10.2% (5dB) and 15.5% (1dB) in comparison to HAS and 14.7% and 23.6%, respectively, in comparison to WS. In Fig. 9 we plotted the trial with the value of NRMSE closest to the median NRMSE of HAS. We may see that HAS and lKPLS provides visually comparable results while SS fails to detect local inhomogeneity close to the point 0.45 on this trial.

Finally, the kernel PLS (KPLS) regression approach was used. In this case we observed that by fixing number of components to be in the range $8-15$ we may achieve comparable performance with other methods. This is demonstrated in Fig. 10 where a boxplot over the NRMSE distribution for KPLS and lKPLS (using both sets of segments) is depicted. Although the KPLS approach provided adequate smooth estimate of the desired heavisine function, similar to SS, it fails to detect a local inhomogeneity in the function (Fig. 9f). We have also observed that the VC-based model selection criterion described in subsection 2.3.1 has tendency to underestimate a number of selected components in the case of higher number of data points (512,1024,2048) and consequently KPLS regression resulted in decreased accuracy. In the case of lower number of data points used (128,256) this model selection criterion provided comparable or better results in comparison to the case where fixed number of components was used. The maximum number of components in these cases was set to 12.

4.2 Event Related Potentials

In the case of lKPLS we again used the equally spaced data in the interval $[-1,1]$ and the width h of Gaussian kernel equal to 0.6. To set weighting function $r(.)$ we created an average of the first five ERP trials on P_z electrode to visually set the segments over which the lKPLS regression was constructed. We also took into account existing knowledge about the shape of real-world ERP (Hillyard & Kutas, 1983; Naatanen & Picton, 1987; Naatanen & Picton, 1986; Parasuraman & Beatty, 1980; Picton et al., 1974) (see also Section 3.2). This motivates our choice of three segments ([-1,-0.3],[-0.3,0.1],[0.1,1]). We set $r(.)$ to be equal 1 over individual segments and then overlap the segments with exponentially decaying values of $r(.)$ reaching the values close to zero (less then $10e^{-5}$) on interval 0.4.

In the case of WS we have used thresholds rescaled by a level-dependent estimation of the noise level (Johnstone & Silverman, 1997). This provided better results in this case where temporally and spatially correlated noise was added to ERP. It was already observed that in the case of colored noise GCV or CV criteria fail to provide an appropriate estimate of the smoothing parameter γ in SS approach (Diggle & Hutchinson, 1989; Wang, 1998; Opsomer, Wang, & Yang, 2001). Although there exist several modifications of GCV in the case of colored noise usually some a priori knowledge about the covariance matrix of the correlated noise or its parametric specifications is needed (Diggle & Hutchinson, 1989; Wang, 1998; Opsomer, Wang, & Yang, 2001).

Signal averaging was also used to extract ERP embedded in noise. This approach was

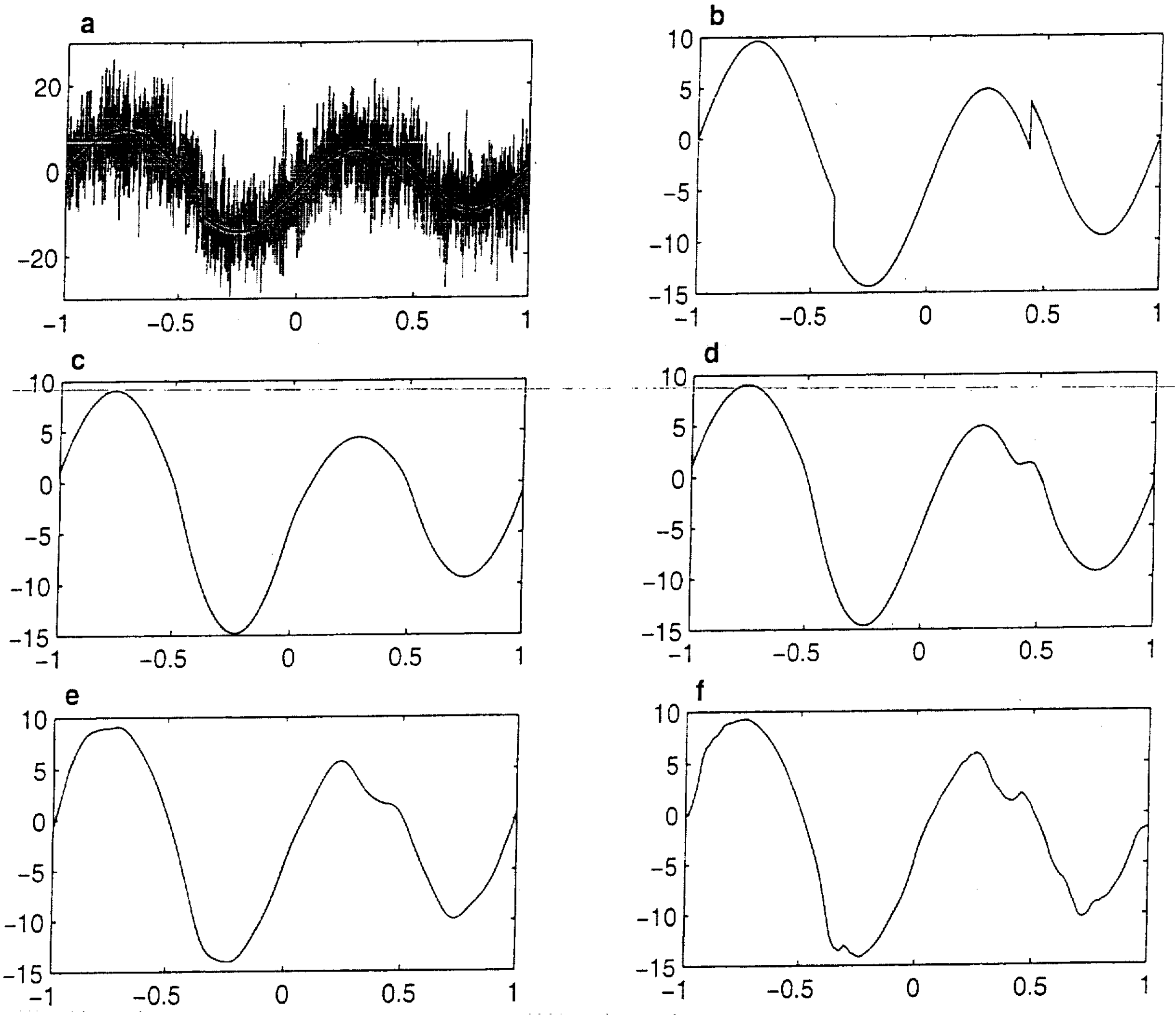

Figure 8: Results on heavisine function, SNR=1dB, number of samples n = 2048 a) example of noisy heavisine function b) clean heavisine function c) locally-based kernel PLS using set of segments $\mathcal{S}_1$, normalized root mean squared error (NRMSE) was equal to 0.115 d) locally-based kernel PLS using set of segments $\mathcal{S}_2$, NRMSE=0.087 e) smoothing splines, NRMSE=0.087 f) wavelet shrinkage, NRMSE=0.087.

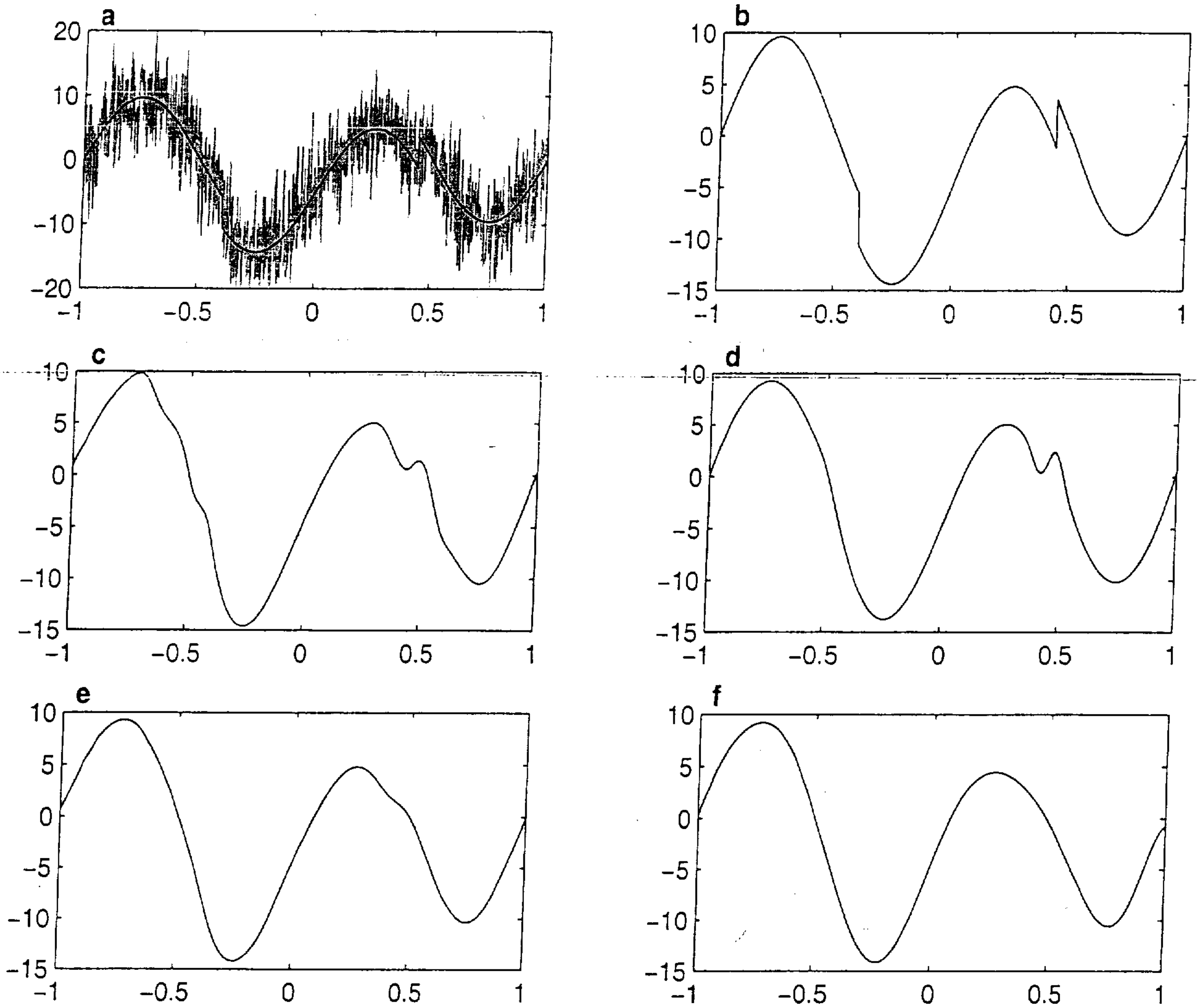

Figure 9: Results on heavisine function, SNR=5dB, number of samples n = 1024 a) example of noisy heavisine function b) clean heavisine function c) hybrid adaptive splines, normalized root mean squared error (NRMSE) was equal to 0.105 d) locally-based kernel PLS using set of segments $\mathcal{S}_2$, NRMSE=0.093 e) smoothing splines, NRMSE=0.096 f) kernel PLS, NRMSE=0.112.

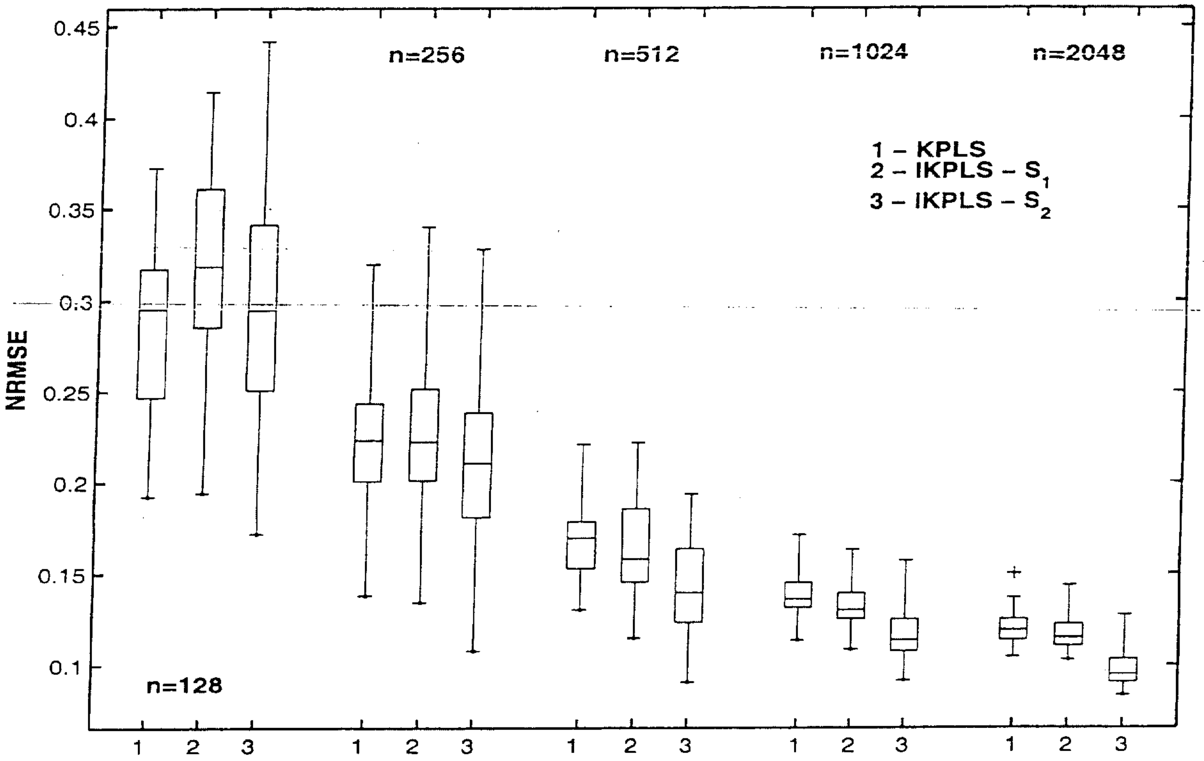

Figure 10: Results on heavisine function. Boxplots with lines at the lower quartile, median, and upper quartile values and whisker plot for three different nonparametric smoothing methods and different number (n) of samples used. The performance of kernel PLS (KPLS, left-hand boxplots in the individual triplets) is compared with locally-based kernel PLS using set of segments $\mathcal{S}_1$ (lKPLS-$\mathcal{S}_1$, middle boxplots in the triplets) and locally-based kernel PLS using set of segments $\mathcal{S}_2$ (lKPLS-$\mathcal{S}_2$, right-hand boxplots in the triplets) in terms of normalized root mean squared error (NRMSE). The boxplots are computed on results from 50 different replicates of noisy heavisine function (SNR=1dB).

proved to be quite useful due to the overlap of the ERP and noise spectra. Although the assumption of over the time stationary, not-varying ERP justify this method, it may provide a useful smooth estimate of ERP also in the case of slight variability among individual ERP realizations. However, in this case the information about the amplitudes and latencies differences over individual trials will be smeared out. The estimate of the ERP at each electrode was constructed to be the average of 20 different realizations of the ERP measured at the electrode. This was taken to represent the same de-noised signal for all trials and NRMSE and CC between this estimate and individual clean ERP realizations was then computed. We have to stress that in contrast to the averaging approach other smoothing methods described are single-trial oriented.

In the case of lKPLS we have also compared a univariate approach in which each electrode represents single lKPLS regression model with the approaches where spatial, temporal and spatio-temporal setting of the multivariate outputs was used. In the spatial setting individual columns are constructed using measurements at different electrodes while in temporal they are represented by different time measurements at the same electrode. In spatio-temporal setting we combine both approaches. We have to stress that this spatio-temporal information is only used to extract common PLS components while regression models are consequently built for each electrode individually. This allows us to obtained several different estimates for a particular electrode by using lKPLS models with different settings of the output matrix. Finally we may create a final estimate at the electrode by combining these individual estimates. We investigated different settings of local spatial distribution of the electrodes as well as more global settings where the measurements from spatially more distributed electrodes created the final multidimensional output matrix $\mathbf{Y}$. The similar modification of short term and long term temporal setting was investigated and mutually combined. We observed small differences among the results achieved over individual trials, however, in terms of averaged NRMSE and CC over all trials the differences were small. In the next we report the results where overall spatial information from all electrodes was used, that is, columns of the output matrix are measurements at individual electrodes over one trial. A number of selected components for the regression models corresponding to individual electrodes was set based on the minimum of the VC-based model selection criterion for the particular model.

The median values of NRMSE computed over 20 different trials on each of 19 electrodes for two levels of SNR used are plotted in Fig. 11 and 12. From this plot we may observe that lKPLS provides consistently better results in comparison to SS and WS. Although these results are worse than applying the signal averaging technique we may also observe that the same technique of averaging applied to smoothed, de-noised estimates of lKPLS outperformed averaging applied to the raw ERP. We may observe that for a smaller level of noise the median of CC is higher then the median of CC using raw noisy ERP data averaging technique over some of the electrodes. This suggests that single trial lKPLS smoothing may better follow latency and amplitude changes among individual ERP realizations. In Fig. 13 we plotted an example taken at C_4 electrode which partially supports this assumption. We may see that average taken over all 20 noisy ERP trials tend to by slightly shifted in the latency of the second negative peak and lower in the amplitude of the first positive peak. It is important to note that the results on some of the electrodes indicate median NRMSE greater than one and also lower median correlation coefficient ($CC < 0.8$). This occurs on electrodes with smaller SNR (frontal and temporal electrodes) and in these cases we cannot expect appropriate reconstruction of clean ERP.

Further, we were interested to see if a proper selection of smoothness in the case of SS may

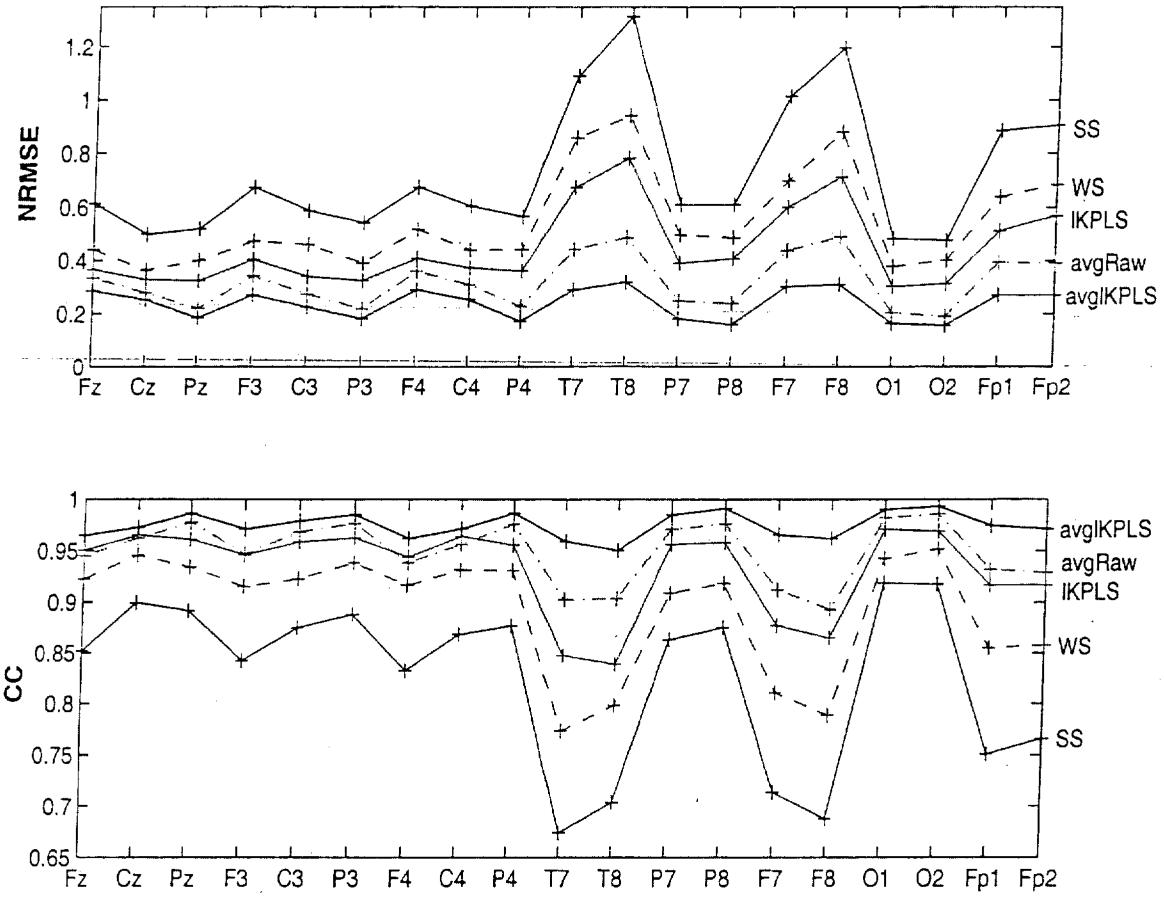

Figure 11: Results on noisy event related potentials (ERP)–20 different trials were used. Averaged SNR over the trials and electrodes was equal to 1.3dB and 512 samples were used. Comparison of different nonparametric smoothing methods in terms of median normalized root mean squared error (NRMSE) (upper graph) and correlation coefficient (CC) (lower graph). In the upper graph the plots from the top to the bottom represent smoothing splines (SS), wavelet shrinkage (WS), locally-based kernel PLS (lKPLS), averaged raw ERP (avgRaw), averaged smoothed curves as result of lKPLS (avglKPLS). The order of the plots is reversed for CC.

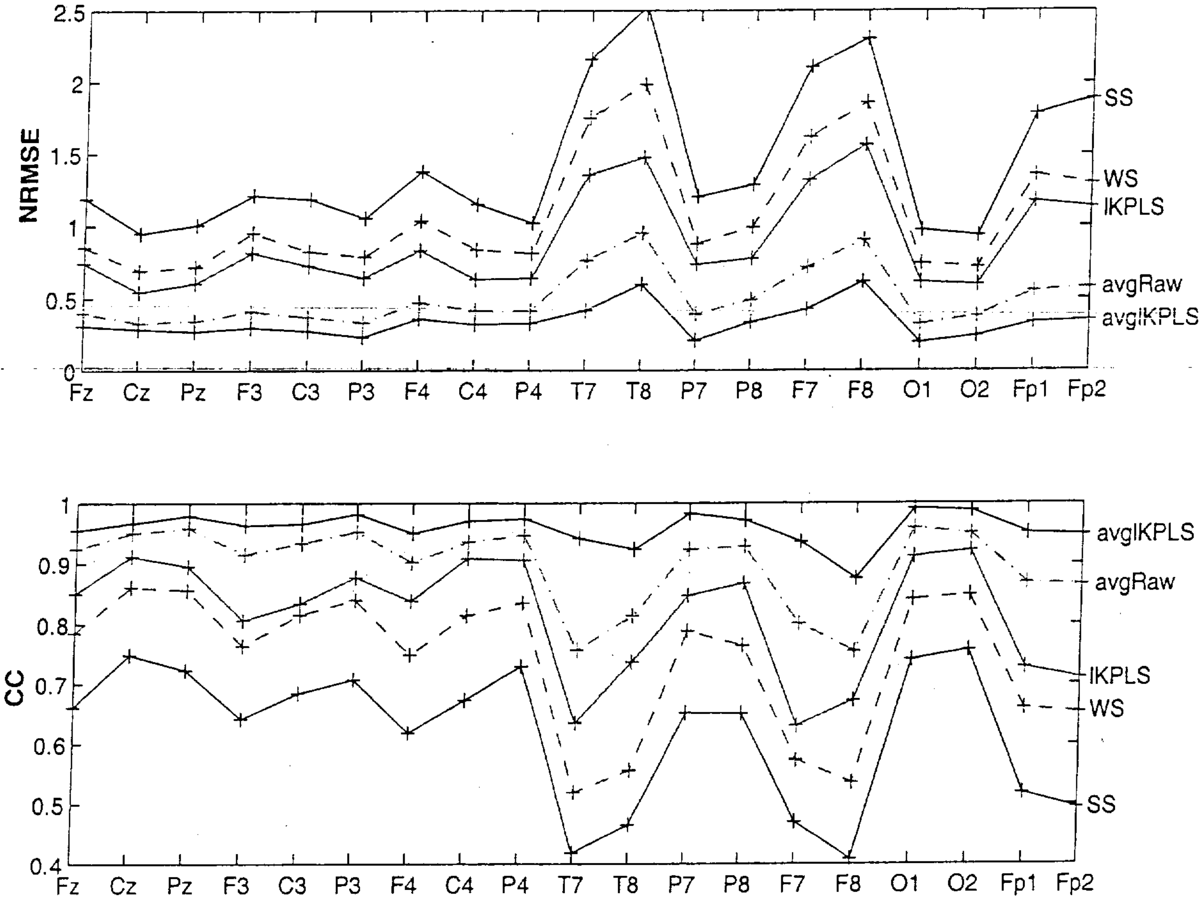

Figure 12: Results on noisy event related potentials (ERP)–20 different trials were used. Averaged SNR over the trials and electrodes was equal to -4.6dB and 512 samples were used. Comparison of different nonparametric smoothing methods in terms of median normalized root mean squared error (NRMSE) (upper graph) and correlation coefficient (CC) (lower graph). In the upper graph the plots from the top to the bottom represent smoothing splines (SS), wavelet shrinkage (WS), locally-based kernel PLS (lKPLS), averaged raw ERP (avgRaw), averaged smoothed curves as result of lKPLS (avglKPLS). The order of the plots is reversed for CC.

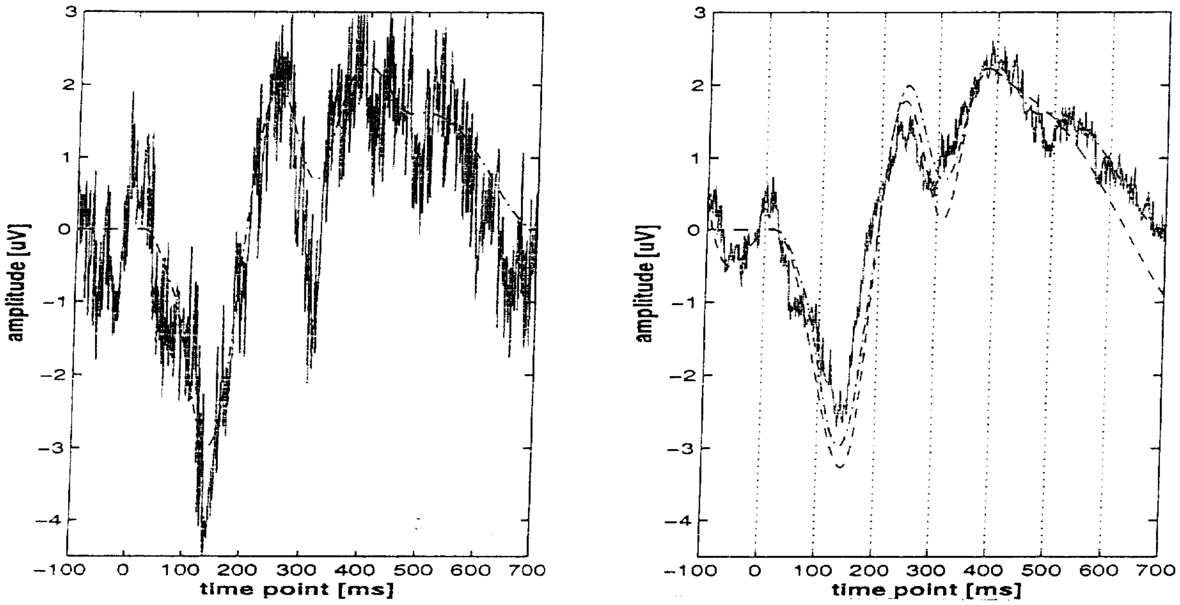

Figure 13: Example of smoothing a trial of ERP on C_4 electrode, SNR=4.5dB, number of samples $n = 512$ a) left graph: solid line–clean ERP, dash-dotted line–noisy ERP b) right graph: dash-dotted line–clean ERP, dashed line–smoothed noisy ERP using locally-based kernel PLS, solid line–average of 20 different noisy ERP trials on C_4 electrode.

provide results which would be better in comparison to that achieved with lKPLS. Because the pspline software package allows us to define df_γ, that is, degrees of freedom parameter, we may vary this parameter and investigate performance of SS. We have to stress that this is only possible due to the fact that we know the clean ERP signals. We observed that changing df in the range of $8 - 16$ provided very comparable results with lKPLS in terms of median NRMSE and CC (Fig. 14).

Using the KPLS approach we have observed the values of median NRMSE and CC to be between the values achieved with WS and lKPLS (see Fig. 11,12). The VC-based model selection criterion using the maximum number of components equal to 12 was used. Similar to lKPLS overall spatial information from all electrodes was used to extract the components and a final number of selected components was again set based on the minimum of the VC-based model selection criterion for the particular model.

In the case where only white Gaussian noise was added to the generated ERP we observed comparable results of SS and lKPLS in terms of NRMSE and CC, closely followed by the KPLS, WS and HAS methods. Although this experimental setting further justified adequacy of the use of the lKPLS and KPLS methods on wider set of smooth ERP signals, it has only low practical value due to the fact that in real measurements of ERP we can not assume spatially and temporally uncorrelated noise superimposed to the ERP waves. In terms of median NRMSE and CC we observed better performance of SS method in comparison to HAS.

Finally we note that all reported results were consistent for different numbers of samples used, i.e., 128, 256 and 512, respectively.

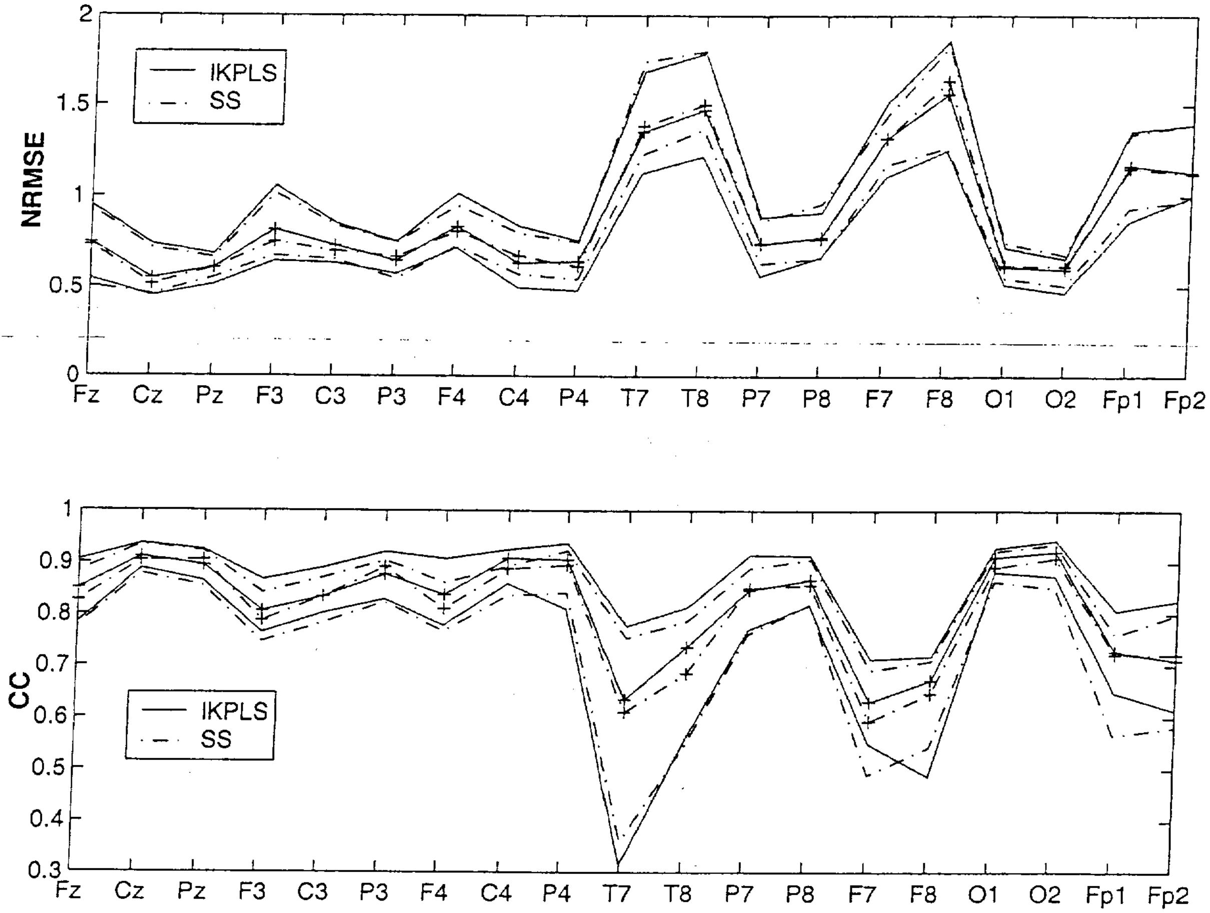

Figure 14: Results on noisy event related potentials (ERP)–20 different trials were used. Averaged SNR over the trials and electrodes was equal to -4.6dB and 512 samples were used. Comparison of smoothing splines ($df_\gamma = 14$) (dash-dotted lines) and locally-based kernel PLS (solid lines) in terms of median, upper and lower quantiles of normalized root mean squared error (NRMSE) (upper graph) and correlation coefficient (CC) (lower graph).

5 Discussion and Conclusion

We have described a new smoothing technique based on kernel PLS regression. On two different data sets we have shown that the proposed methodology may provide comparable or better results when compared with the existing state-of-the-art, namely smoothing splines, hybrid adaptive splines and wavelet shrinkage techniques. By expressing smoothing splines in RKHS we could directly see existing connections with our kernel PLS regression method. These close connections between smoothing splines and recently elaborated methods of regularization networks and support vector machines, which also motivated our kernel PLS methodology, was already pointed out in Smola, Schölkopf, and Müller (1998), Girosi (1998), Wahba (1999) and Evgeniou, Pontil, and Poggio (2000). Recent interest in the use and development of different types of kernel functions in kernel-based learning gives hope to extend the methodology of nonparametric curve fitting discussed in this paper. An example of this may be the recent theoretical work on statistical asymptotic properties of Gaussian and a periodic Gaussian kernels (Lin & Brown, 2002). In agreement with theoretical results, the authors have shown that periodic Gaussian kernel may provide better results in the case of very smooth functions. In the case of functions of moderate smoothness, comparable results with periodic cubic splines were achieved.

We also proposed a locally-based kernel PLS regression model. This method was designed to incorporate prior knowledge (or knowledge which may be derived from the experimental data) about the approximate location of changes in signal curvature, discontinuities or other spatial inhomogeneities occurring in the signal. The motivation to incorporate the existing knowledge about the signal of interest came from the problem of smoothing ERP signals corrupted by high levels of noise. Knowledge about the approximate shape of ERP is known from many psychological observations, however, the exact shape of ERP varies from trial to trial. The good results on the ERP data set justify usefulness of the approach. However, where there is large variation in the latency of ERP components, additional operations may be needed to determine the proper intervals and weighting functions for locally-based kernel PLS. Our results also suggest that this method tends to be superior to the global kernel PLS approach on this data set. The results also encourage us to considering the (locally-based) kernel PLS methodology on other problems of de-noising biological data, e.g., removal of eye-blinks from EEG data. On the heavisine function corrupted with high levels of white Gaussian noise we observed that, by including the knowledge about approximate location of a local inhomogeneity, locally-based KPLS provided better results in comparison to hybrid adaptive splines and wavelet shrinkage methods—the methods which are also design to deal with local inhomogeneities in the signal. This superiority was observed in terms of the visual detection of local inhomogeneity and also in the terms of averaged NRMSE and CC over individual trials. Finally, we have to note that the concept of localization may be also potentially implemented into the other kernel-based regression models, e.g., support vector regression, kernel principal components regression or kernel ridge regression (Vapnik, 1998; Cristianini & Shawe-Taylor, 2000; Schölkopf & Smola, 2002; Rosipal, Girolami, & Trejo, 2000; Rosipal et al., 2001; Saunders, Gammerman, & Vovk, 1998).

One of the main issues of existing nonparametric regression techniques is appropriate selection of regularization (smoothing parameters). On data sets with uncorrelated Gaussian noise added to clean signals, we observed a good performance of smoothing splines using GCV criterion to set a level of regularization. However, in the case of colored noise this criterion without a priori knowledge or an appropriate estimate of the variance-covariance matrix generally tends to underestimate smoothing parameters (Diggle & Hutchinson, 1989; Wang, 1998; Opsomer, Wang, & Yang, 2001). From this point it seems to be interesting to stress the good behavior of our locally-based and global kernel PLS regression methods on a problem with colored noise added to generated ERP. Using the clean desired signals we have tried to tune the level of smoothness in the case of smoothing splines to its optimal or near optimal values. However, this approach did not outperform results achieved with the (locally-based) kernel PLS regression because comparable values of NRMSE and CC were observed. The kernel PLS based approaches also resulted in better performance in comparison to the wavelet de-noising using the shrinkage criterion design to deal with colored noise (Johnstone & Silverman, 1997). This success may be explained by the different approach to the constructing the final regression model in kernel PLS. We construct a set of orthogonal components that increasingly describe the variance of the noisy observed signals. Thus, with the aim to recover a smooth signal of interest we may limit the number of components to a predefined maximum value and to use a model selection criterion to set a final number of them. In this paper we have used the VC-based model selection criterion that provided satisfactory results. However, we have to note, that on the investigated data sets we have observed a failure of this model selection criterion when the maximum number of components allowed to enter the model was higher (generally, more than 8-10 in the case of locally-based kernel PLS and more than 14-16 in the case of kernel PLS). We may hypothesize that this is due to the rapid increase of complexity (as defined in Section 2.3) of higher components which is not appropriately reflected in the setting of experimentally designed VC-based criterion used. It also remains an open task to compare different existing model selection criteria in this scenario.

On the artificially generated ERP data set we have observed that by using different spatio-temporal arrangements of the signals from different electrodes in our multivariate (locally-based) kernel PLS models we achieved very comparable results. This may be consider as both negative and positive. On the negative side, our belief that this arrangement may provide us some additional common information from measurements on different electrodes and trials in comparison to single electrode and single trial was not confirmed. More detailed inspection suggests that the differences between these two strategies of setting the outputs start to be more evident when higher numbers of PLS components (describing also noisy part of the signal) are included into the final regression model. However, this is contradictory to our goal of smoothing the ERP signals where a lower number of components is needed. Our observations indicate that these very first components are generally very similar in the case of univariate (single trial single electrode) and also in the case of multivariate outputs based on a different spatio-temporal setting. Thus, it is not surprising that using these components in the final regression model results in comparable performance. On the positive side, the possibility to extract desired components using all electrodes from one or several trials occurred in our case to be computationally easier in comparison to the extraction of components from each electrode and trial independently.

Acknowledgement

The authors would like to thank Dr. Peter Tiňo for helpful discussions during the study.

References

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

Chen, S., Cowan, C.F.N., & Grant, P.M. (1991). Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks. *IEEE Transactions on Neural Networks, 2*, 302–309.

Cherkassky, V., & Shao, X. (2001). Signal estimation and denoising using VC-theory. *Neural Networks, 14*, 37–52.

Cherkassky, V., Shao, X., Mulier, F.M., & Vapnik, V.N. (1999). Model Complexity Control for Regression Using VC Generalization Bounds. *IEEE Transactions on Neural Networks, 10*, 1075–1089.

Cristianini, N., & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines.* Cambridge University Press.

~~Diggle, P.J., & Hutchinson, M.F. (1989). On spline smoothing with autocorrelated errors.~~ *The Australian Journal of Statistics, 31*, 161–182.

Donoho, D.L., & Johnstone, I.M. (1995). Adapting to Unknown Smoothness via Wavelet Shrinkage. *Journal of the American Statistical Association, 90*, 1200–1224.

Evgeniou, T., Pontil, M., & Poggio, T. (2000). Regularization Networks and Support Vector Machines. *Advances in Computational Mathematics, 13*, 1–50.

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

Girosi, F. (1998). An equivalence between sparse approximation and Support Vector Machines. *Neural Computation, 10*, 1455–1480.

Girosi, F., Jones, M., & Poggio, T. (1995). Regularization Theory and Neural Network Architectures. *Neural Computation, 7*, 219–269.

Green, P.J., & Silverman, B.W. (1994). *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach.* London: Chapman & Hall.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The Elements of Statistical Learning.* Springer.

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

Hillyard, S.A., & Kutas, M. (1983). Electrophysiology of cognitive processing. *Annual review of Psychology, 34*, 33–61.

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

Jasper, H.H. (1958). The ten-twenty electrode system of the International Federation. *Electroencephalography and Clinical Neurophysiology, 10*, 371–375.

Johnstone, I.M., & Silverman, B. (1997). Wavelet Threshold Estimators for Data with Correlated noise. *Journal of the Royal Statistical Society, series B, 59*, 319–351.

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

Lin, Y., & Brown, L.D. (2002). Statistical Properties of the Method of Regularization with Periodic Gaussian Reproducing Kernel (Tech. Rep. No. 1062). Madison, Wisconsin: Department of Statistics, University of Wisconsin.

Luo, Z., & Wahba, G. (1997). Hybrid adaptive splines. *Journal of the American Statistical Association, 92*, 107–116.

Manne, R. (1997). Analysis of Two Partial-Least-Squares Algorithms for Multivariate Calibration. *Chemometrics and Intelligent Laboratory Systems, 2*, 187–197.

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

Naatanen, R., & Picton, T.W. (1986). N2 and automatic versus controlled processes. *Electroencephalography and Clinical Neurophysiology Supplement, 38*, 169–186.

Naatanen, R., & Picton, T.W. (1987). The N1 wave of the human electric and magnetic response to sound: a review and an analysis of the component structure. *Psychophysiology, 24*, 375–425.

Opsomer, J.D., Wang, Y., & Yang, Y. (2001). Nonparametric regression with correlated errors. *Statistical Science, 16*, 134–153.

Orr, M.J.L. (1995). Regularisation in the selection of Radial Basis Function centers. *Neural Computation, 7*, 606–623.

Parasuraman, R., & Beatty, J. (1980). Brain events underlying detection and recognition of weak sensory signals. *Science, 210*, 80–83.

Picton, T.W., Hillyard, S.A., Kraus, H.I., & Galambos, R. (1974). Human auditory evoked potentials. I. Evaluation of components. *Electroencephalography and Clinical Neurophysiology, 36*, 179–190.

Platt, C.J. (1991). A resource-allocating network for function interpolation. *Neural Computation, 3*, 213–225.

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

Rosipal, R., & Girolami, M., & Trejo, L.J. (2000). Kernel PCA for Feature Extraction of Event-Related Potentials for Human Signal Detection Performance. In *Proc. ANNIMAB-1 Conference, Götegorg, Sweden*, (pp. 321–326).

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

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

Saitoh, S. (1988). *Theory of Reproducing Kernels and its Applications.* Harlow, England: Longman Scientific & Technical.

Saunders, C., Gammerman, A., & Vovk, V. (1998). Ridge Regression Learning Algorithm in Dual Variables. In *Proc.of the 15th International Conference on Machine Learning,Madison, Wisconsin* (pp. 515–521).

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

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

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

Tipping, M. (2001). The relevance vector machine. *Journal of Machine Learning Research, 1*, 211–244.

Tipping, M.E., & Bishop, C.M. (1999). Mixtures of probabilistic principal component analysers. *Neural Computation, 11*, 443–482.

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

Wahba, G. (1990). *Splines Models of Observational Data*, volume 59 of *Series in Applied Mathematics.* Philadelphia, SIAM.

Wahba, G. (1999). Support Vector Machines, Reproducing Kernel Hilbert Spaces, and Randomized GACV. In B. Schölkopf, C.J.C. Burges, & A.J. Smola (Eds.), *Advances in Kernel Methods - Support Vector Learning* (pp. 69–88). Cambridge, MA: The MIT Press.

Wu, W., Massarat, D.L., & de Jong, S. (1997). The kernel PCA algorithms for wide data. Part II: Fast cross-validation and application in classification of NIR data. *Chemometrics and Intelligent Laboratory Systems, 37*, 271–280.

Wang, Y. (1998). Smoothing Spline Models With Correlated Random Errors. *Journal of the American Statistical Association, 93*, 341–348.