Kernel PCA for Feature Extraction and De-Noising in Non-linear Regression

Roman Rosipal[1], Mark Girolami[1],
Leonard J. Trejo[2], Andrzej Cichocki[3]

[1]Computational Intelligence Research Unit
School of Information and Communication Technologies
University of Paisley
Paisley, PA1 2BE, Scotland
e-mail: {*rosi-ci0,giro-ci0*}*@paisley.ac.uk*

[2]Computational Sciences Division
NASA Ames Research Center
Moffett Field, CA
e-mail: *ltrejo@mail.arc.nasa.gov*

[3] Laboratory for Advanced Brain Signal Processing
Brain Science Institute RIKEN
2-1, Hirosawa, Wako-Shi, Japan
e-mail: *cia@brain.riken.go.jp*

Abstract

In this paper, we propose the application of the Kernel Principal Component Analysis (PCA) technique for feature selection in a high-dimensional feature space where input variables are mapped by a Gaussian kernel. The extracted features are employed in the regression problems of chaotic Mackey-Glass time-series prediction in a noisy environment and estimating human signal detection performance from brain event-related potentials elicited by task relevant signals. We compared results obtained using either Kernel PCA or linear PCA as data preprocessing steps. On the human signal detection task we report the superiority of Kernel PCA feature extraction over linear PCA. Similar to linear PCA we demonstrate de-noising of the original data by the appropriate selection of various non-linear principal components. The theoretical relation and experimental comparison of Kernel Principal Components Regression, Kernel Ridge Regression and ϵ-insensitive Support Vector Regression is also provided.

Key words: feature extraction; principal components; non-linear regression; kernel functions; de-noising; human performance monitoring.

1 Introduction

In many real world applications appropriate preprocessing transformations of high dimensional input data can increase overall performance of algorithms. In general, there exist some correlations among input variables; thus dimensionality reduction or so-called *feature extraction* allows us to restrict the entire input space to a sub-space of lower dimensionality.

In this study, we have used the recently proposed Kernel Principal Component Analysis (PCA) [1] method for feature selection in a high dimensional feature space $\mathcal{F}$ (with dimension $M \leq \infty$). This allows us to obtain features (nonlinear principal components) with higher-order correlations between input variables, and in addition, we can extract nonlinear components up to the number of data points n [1] (assuming $n \leq M$). Kernel PCA [1] is based on computation of the standard linear PCA [2] in a feature space, into which input data $\mathbf{x}$ are mapped via some nonlinear function $\Phi(\mathbf{x})$. To this end, we compute a canonical dot product in space $\mathcal{F}$ using a kernel function, i.e. $K(\mathbf{x}, \mathbf{y}) = (\Phi(\mathbf{x}).\Phi(\mathbf{y}))$. This 'kernel trick' allows us to carry out any algorithm, e.g. Support Vector Regression (SVR) [3, 4, 5], that can be expressed in the terms of dot products in space $\mathcal{F}$. Next, the selected features are used to train the ϵ-insensitive SVR (see reference for detailed description [3]) and Kernel Principal Components Regression (KPCR) [6] models to estimate the desired input-output mappings. Both techniques perform a linear regression in a feature space $\mathcal{F}$, however, different cost functions are used. Whilst the ϵ-insensitive cost function used in SVR is more robust for noise distributions close to uniform, in the case of Gaussian noise, the best approximation to the regression provides a quadratic cost function. Applying a quadratic cost function to SVR leads to Kernel Ridge Regression (KRR) [7, 5]. Both KRR and KPCR are the shrinkage estimators designed to deal with multicollinearity or near-linear dependence of regressors (see e.g. [8, 9, 2]). Multicollinearity results in large variances and covariances for the least-squares estimators of the regression coefficients and can dramatically influence the effectiveness of a regression model. We will give the theoretical basis of KPCR and will also highlight the relation to KRR.

In noisy environments linear PCA is a widely used de-noising technique. We can discard the finite variance due to the noise by projection of the data onto the main principal components. The same technique can be applied in feature space $\mathcal{F}$ by using the main nonlinear principal components computed by Kernel PCA. However, the number of nonlinear principal components extracted by Kernel PCA can be substantially higher (up to the number of

data points n). This can be nearly always advantageous, especially in the situation where the dimensionality N of the input data points is significantly smaller than the number of data points and a data structure is spread over all eigendirections. In this case decreasing the input dimensionality by projecting the input data to $l < N$ main linear principal components may lead to the loss of significant amounts of information. On the other hand, we can believe that "spreading" the information about the data structure into $k > N$ nonlinear principal components will give the potential of discarding some of the eigendirections where the noisy part of data is mainly contained.

On two data sets - the chaotic Mackey-Glass time series and human Evoked Related Potentials (ERPs) - we compared KPCR, KRR and SVR[1] techniques. We demonstrate that by selection of a subset of nonlinear principal components used in KPCR we can achieve superior or similar results compared to KRR, moreover, in the case of KPCR the final linear model in a feature space is significantly smaller. On the ERPs data set, the results suggest the superiority of Kernel PCA for feature extraction over linear PCA in some cases. In addition, the performance of KPCR and KRR models using the quadratic loss function is slightly superior to SVR. This suggests that on that particular data set a Gaussian type of noise is more likely; i.e. the regression models with a quadratic loss function are preferable.

The following section presents the Kernel PCA technique and linear regression models in a high dimensional kernel defined space. The problem of de-noising of the data set in the kernel space is also addressed. In Section 3 the construction of the data sets employed is described. Section 4 discusses the results. Section 5 concludes the paper.

2 Methods

2.1 Kernel PCA and Multi-Layer SVR

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

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

where $\Phi(\mathbf{x}_i)$ are centered nonlinear mappings of the input variables $\mathbf{x}_i \in \mathcal{R}^{\mathcal{N}}$ $i = 1, ..., n$ (the centralization of the mapped data in $\mathcal{F}$ is given in Appendix

[1] We are assuming an SVR model with the $\epsilon-$insensitive cost function.

A). The diagonalization represents a transformation of the original data to new coordinates defined by orthogonal eigenvectors $\mathbf{V}$. We have to find eigenvalues $\lambda \geq 0$ and non-zero eigenvectors $\mathbf{V} \in \mathcal{F}$ satisfying the eigenvalue equation

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

Realizing, that all solutions $\mathbf{V}$ with $\lambda \neq 0$ lie in the span of mappings $\Phi(\mathbf{x}_1), \ldots, \Phi(\mathbf{x}_n)$, Schölkopf et al. [1] derived the equivalent eigenvalue problem

$$n\lambda\boldsymbol{\alpha} = \mathbf{K}\boldsymbol{\alpha}, \tag{1}$$

where $\boldsymbol{\alpha}$ denotes the column vector with coefficients $\alpha_1, ..., \alpha_n$ such that

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

and $\mathbf{K}$ is a symmetric $(n \times n)$ *Gram* matrix with the elements

$$\mathrm{K}_{ij} = (\Phi(\mathbf{x}_i).\Phi(\mathbf{x}_j)) := K(\mathbf{x}_i, \mathbf{x}_j).$$

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

$$\beta(\mathbf{x})_k := (\mathbf{V}^k.\Phi(\mathbf{x})) = \sum_{i=1}^{n} \alpha_i^k K(\mathbf{x}_i, \mathbf{x}). \tag{2}$$

We then select the first $p < n$ nonlinear principal components, e.g. the directions which describe a desired percentage of data variance, and thus work in the p-dimensional sub-space of feature space $\mathcal{F}$. This allows us to construct multi-layer support vector machines [1], where a preprocessing layer extracts features for the next regression or classification task. In our study we focus on the regression problem.

Generally, the SVR problem (see e.g. [4]) can be defined as the determination of function $f(\mathbf{x}, \boldsymbol{\omega})$ which approximates an unknown desired function and has the form

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

where b is an unknown bias term and $\boldsymbol{\omega} \in \mathcal{F}$ is a vector of unknown coefficients. In [3] the following regularized risk functional has been used to compute the unknown coefficients b and $\boldsymbol{\omega}$:

$$R_{svr}(\boldsymbol{\omega}, b) = \frac{1}{n} \sum_{i=1}^{n} |Err|_\epsilon + \eta \|\boldsymbol{\omega}\|^2, \tag{3}$$

where $Err = y_i - f(\mathbf{x}_i, \boldsymbol{\omega})$, $\{y_i\}_{i=1}^n$ are the obtained outputs; $\eta \geq 0$ is a regularization constant to control the trade-off between complexity and accuracy of the regression model and $|Err|_\epsilon$ is Vapnik's ϵ-insensitive loss function [3].

In [3] it is shown that the regression estimate that minimizes the risk functional (3) has the form:

$$f(\mathbf{x}, \boldsymbol{\gamma}, \boldsymbol{\gamma}^*) = \sum_{i=1}^{n} (\gamma_i^* - \gamma_i) K_1(\mathbf{x}_i, \mathbf{x}) + b, \tag{4}$$

where $\{\gamma_i, \gamma_i^*\}_{i=1}^n$ are Lagrange multipliers.

Combining the Kernel PCA preprocessing step with SVR yields a multilayer SVR (MLSVR) in the following form [1]:

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

where components of vectors $\boldsymbol{\beta}$ are defined by (2). However, in practice the choice of appropriate kernel function K_1 can be difficult. In this study, a polynomial kernel of first order $K_1(\mathbf{x}, \mathbf{y}) = (\mathbf{x}.\mathbf{y})$ is employed. We are thus performing a linear SVR on the p-dimensional sub-space of $\mathcal{F}$. The advantage of linear SVR over ordinary linear regression is the possibility of using a large variety of loss functions to suit different noise models [4], e.g. Vapnik's proposed ϵ-insensitive function is more robust for noise distributions close to uniform and also provides a sparse solution to the regression problem. However, in the case of Gaussian noise the best approximation to the regression provides a least-squares method with the quadratic loss function of the form $L(y_i, f(\mathbf{x}_i, \mathbf{w})) = [y_i - f(\mathbf{x}_i, \mathbf{w})]^2$. We discuss methods using this loss function in the next section.

2.2 Feature Space Regularized Least-Squares Regression Models and Multicollinearity

The multicollinearity or near-linear dependence of regressors is a serious problem that can dramatically influence the usefulness of a regression model. Multicollinearity results in large variances and covariances for the least-squares estimators of the regression coefficients. Multicollinearity can also produce estimates of the regression coefficients that are too large in absolute value. Thus the values and signs of estimated regression coefficients may change considerably given different data samples. This effect can lead to a

regression model which fits the training data reasonably well, but in general bad generalization of the model can occur. This fact is in a very close relation to the argument stressed in [10], where the authors have shown that choosing the *flattest* function[2] in a feature space can, based on the smoothing properties of the selected kernel function, lead to a smooth function in the input space. There exist several methods to deal with multicollinearity; in our case we discuss the ridge regression (RR) and principal component regression (PCR) approaches. Using the theoretical basis of these techniques in input space, we will now discuss their parallel in a kernel defined feature space; i.e. KPCR and KRR.

2.2.1 Kernel Principal Component Regression

Consider the standard regression model in feature space $\mathcal{F}$

$$\mathbf{y} = \mathbf{\Phi}\boldsymbol{\xi} + \boldsymbol{\epsilon}, \tag{5}$$

where $\mathbf{y}$ is a vector of n observations of the dependent variable, $\mathbf{\Phi}$ is an $(n \times M)$ matrix of regressors whose i-th row is the vector $\Phi(\mathbf{x}_i)$ of the mapped $\mathbf{x}_i$ observation into $M \leq \infty$ dimensional feature space $\mathcal{F}$, $\boldsymbol{\xi}$ is a vector of regression coefficients and $\boldsymbol{\epsilon}$ is the vector of error terms whose elements have equal variance σ^2, and are independent of each other. We also assume that regressors $\{\Phi_j(\mathbf{x})\}_{j=1}^M$ are zero-mean. Thus $\mathbf{\Phi}^T\mathbf{\Phi}$ is proportional to the sample covariance matrix and Kernel PCA can be performed to extract M eigenvalues $\{\lambda_j\}_{j=1}^M$ and corresponding eigenvectors $\{\mathbf{V}^j\}_{j=1}^M$. The projection of the $\Phi(\mathbf{x})$ onto the k-th nonlinear principal component is given by (2). By projection of all original regressors onto the principal components we can rewrite (5) as

$$\mathbf{y} = \mathbf{B}\mathbf{w} + \boldsymbol{\epsilon}, \tag{6}$$

where $\mathbf{B} = \mathbf{\Phi}\mathbf{V}$ is now an $(n \times M)$ matrix of transformed regressors and $\mathbf{V}$ is an $(M \times M)$ matrix whose k-th column is the eigenvector $\mathbf{V}^k$. The columns of the matrix $\mathbf{B}$ are now orthogonal and the least squares estimate of the coefficients $\mathbf{w}$ becomes

$$\hat{\mathbf{w}} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{y} = \mathbf{\Lambda}^{-1}\mathbf{B}^T\mathbf{y}, \tag{7}$$

[2]The *flatness* is defined in the sense of penalizing high values of the regression coefficients estimate.

where $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_M)$. The results obtained using all principal components for the projection of the original regressor variables for (6) is equivalent to that obtained by least squares using the original regressors.

In fact we can express the estimate $\hat{\boldsymbol{\xi}}$ of the original model (5) as

$$\hat{\boldsymbol{\xi}} = \mathbf{V}\hat{\mathbf{w}} = \mathbf{V}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{y} = \sum_{i=1}^{M} \lambda_i^{-1}\mathbf{V}^i(\mathbf{V}^i)^T\mathbf{\Phi}^T\mathbf{y}$$

and its corresponding variance-covariance matrix [2] as

$$cov(\hat{\boldsymbol{\xi}}) = \sigma^2\mathbf{V}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{V}^T = \sigma^2\mathbf{V}\mathbf{\Lambda}^{-1}\mathbf{V}^T = \sigma^2\sum_{i=1}^{M} \lambda_i^{-1}\mathbf{V}^i(\mathbf{V}^i)^T. \qquad (8)$$

To avoid the problem of multicollinearity PCR uses only some of the principal components. It is clear from (8) that the influence of small eigenvalues can significantly increase the overall variance of the estimate. PCR simply deletes the principal components corresponding to small values of the eigenvalues λ_i, i.e. the principal components where multicollinearity may appear. The penalty we have to pay for the decrease in variance of the regression coefficient estimate is bias in the final estimate. However, if multicollinearity is a serious problem, the introduced bias can have a less significant effect in comparison to a high variance estimate. If the elements of $\mathbf{w}$ corresponding to deleted regressors are zero, an unbiased estimate is achieved [2].

Using the first p-nonlinear principal components (2) to create a linear model based on orthogonal regressors in feature space $\mathcal{F}$ we can formulate the KPCR model as

$$f(\mathbf{x}, \mathbf{c}) = \sum_{k=1}^{p} w_k\beta(\mathbf{x})_k + b = \sum_{k=1}^{p} w_k \sum_{i=1}^{n} \alpha_i^k K(\mathbf{x}_i, \mathbf{x}) + b = \sum_{i=1}^{n} c_i K(\mathbf{x}_i, \mathbf{x}) + b, \qquad (9)$$

where $\{c_i = \sum_{k=1}^{p} w_k\alpha_i^k\}_{i=1}^{n}$.

We have shown that by removing the principal components whose variances are very small we can eliminate large variances of the estimate due to multicollinearities. However, if the orthogonal regressors corresponding to those principal components have a large correlation with the dependent variable y such deletion is undesirable (experimentally demonstrated in [11]). There are several different strategies for selecting the appropriate orthogonal regressors for the final model (see [2, 12] and ref. therein). In [11] we considered the Covariance Inflation Criterion [13] for model selection in KPCR as a novel alternative to methods such as cross-validation.

2.2.2 Kernel Ridge Regression

KRR is another technique to deal with multicollinearity by assuming the linear regression model (5) whose solution is now achieved by minimizing

$$R_{rr}(\boldsymbol{\xi}, b) = \sum_{i=1}^{n} [y_i - f(\mathbf{x}_i, \boldsymbol{\xi})]^2 + \vartheta \|\boldsymbol{\xi}\|^2, \tag{10}$$

where $f(\mathbf{x}, \boldsymbol{\xi}) = \boldsymbol{\xi}^T \Phi(\mathbf{x}) + b$ and ϑ is a regularization term. The least-squares estimate of $\boldsymbol{\xi}$ is biased but the variance is decreased (see e.g.[9]). Similar to the KPCR case we can express the variance-covariance matrix of the $\boldsymbol{\xi}$ estimate [2] as

$$cov(\hat{\boldsymbol{\xi}}) = \sigma^2 \sum_{i=1}^{M} \lambda_i (\lambda_i + \vartheta)^{-2} \mathbf{V}^i (\mathbf{V}^i)^T.$$

We can see, that in contrast to KPCR, the variance reduction in KRR is achieved by giving less weight to small eigenvalue principal components via the factor ϑ.

In practice we usually do not know the explicit mapping $\Phi(.)$ or its computation in the high-dimensional feature space $\mathcal{F}$ may be numerically intractable. In [7], using the dual representation of the linear RR model the authors derived the formula for estimation of the weights $\boldsymbol{\xi}$ for the linear RR model $y = \boldsymbol{\xi}^T \Phi(\mathbf{x})$ in feature space $\mathcal{F}$; i.e. (non-linear) KRR. Again, using the fact that $K(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x})^T \Phi(\mathbf{y})$ we can express the final KRR model in the dot product form [7, 5]

$$f(\mathbf{x}) = \mathbf{c}^T \mathbf{k} = \mathbf{y}^T (\mathbf{K} + \vartheta \mathbf{I})^{-1} \mathbf{k}\,, \tag{11}$$

where $\mathbf{K}$ is again an $(n \times n)$ Gram matrix consisting of dot products $\mathrm{K}_{ij} = (\Phi(\mathbf{x}_i).\Phi(\mathbf{x}_j))$ $i, j = 1, \ldots, n$; $\mathbf{k}$ is the vector of dot products of a new mapped input example $\Phi(\mathbf{x})$ and the vectors of the training set; i.e. $\mathrm{k}_i = (\Phi(\mathbf{x}_i).\Phi(\mathbf{x}))$ and $\mathbf{I}$ is an $(n \times n)$ identity matrix,. It is worth noting that the same solution to the RR problem in the feature space $\mathcal{F}$ can also be derived based on the dual representation of Regularization Networks (see e.g. [14]) or through the techniques derived from Gaussian processes [15, 5].

We can see that including a possible bias term into the model leads to its penalization through the ϑ term. However, in the case of regression or classification tasks there is no reason to penalize the shift of $f(.)$ by a constant. To overcome this we can add an extra unpenalized bias term to

our linear regression model in $\mathcal{F}$. Effectively, it means using a new kernel of the form

$$\hat{K}(\mathbf{x}, \mathbf{y}) = K(\mathbf{x}, \mathbf{y}) + \vartheta_0, \quad \vartheta_0 \in R.$$

Now, the solution will take the form [14, 16, 17]

$$f(\mathbf{x}) = \sum_{i=1}^{n} c_i \hat{K}(\mathbf{x}, \mathbf{x}_i) + \hat{b} = \sum_{i=1}^{n} c_i (K(\mathbf{x}, \mathbf{x}_i) + \vartheta_0) + \hat{b} = \sum_{i=1}^{n} c_i K(\mathbf{x}, \mathbf{x}_i) + b \quad (12)$$

and the unknown coefficients $\{c_i\}_{i=1}^{n}, b = \sum_{i=1}^{n} c_i \vartheta_0 + \hat{b}$ can be found by solving the following system of linear equations [14, 17]

$$(\hat{\mathbf{K}} + \vartheta \mathbf{I})\mathbf{c} + \mathbf{1}b = (\mathbf{K} + (\vartheta + \vartheta_0)\mathbf{I})\mathbf{c} + \mathbf{1}b = (\mathbf{K} + \vartheta_{new}\mathbf{I})\mathbf{c} + \mathbf{1}b = \mathbf{y},$$

$$\sum_{i=1}^{n} c_i = 0. \quad (13)$$

where $\mathbf{1}$ is an $(n \times 1)$ vector of ones. Thus we still can use a positive definite kernel K as the only change is to estimate new b and ϑ_{new} terms. Recall that the solution of the SVR, i.e. assuming the linear regression model $y = \boldsymbol{\xi}^T \Phi(\mathbf{x}) + b$ in the feature space $\mathcal{F}$, leads to the non-linear regression model (12). In fact, in [18] the authors have shown that using the quadratic loss function in the case of SVR transforms the general quadratic optimization problem [4] for finding the estimate of the weights $\boldsymbol{\xi} = \sum_{i=1}^{n} c_i \Phi(\mathbf{x}_i)$ and b to the solution of the linear equations (13).

Another technique in removing a "bias" term is to "centralize" the regression problem in feature space; i.e. assume the sample mean of the mapped data $\tilde{\Phi}(\mathbf{x}_i)$ and targets $\tilde{y}$ to be zero. This will lead to the regression problem $\tilde{y} = \tilde{\boldsymbol{\xi}}^T \tilde{\Phi}(\mathbf{x})$ without the bias term. The centralization of the individual mapped data points $\Phi(\mathbf{x})$ can be done by the same "centralization" of the Gram matrix $\mathbf{K}$ and vector $\mathbf{k}$ as described in Appendix A. The solution is then given by modification of (11) to the form

$$\tilde{f}(\mathbf{x}) = \tilde{\mathbf{y}}^T (\tilde{\mathbf{K}} + \vartheta \mathbf{I})^{-1} \tilde{\mathbf{k}} \, . \quad (14)$$

In [19] we observed that both approaches provide the same results.

2.2.3 Summing Up

Using the analogy with PCR and RR in input data space, a connection between regularized linear regression models in feature space $\mathcal{F}$ corresponding

to KPCR and KRR has been established. Both methods belong to the class of shrinkage estimators; i.e. they shrink the ordinary least squares solution from the directions of low data spread to directions of larger data spread. This effectively means that we can achieve the desired lower variance of the estimated regression coefficients at the cost of a biased estimate. Whilst with KPCR we project the data mainly to the principal components corresponding to larger eigenvalues, with KRR we are giving less weight to the smaller eigenvalues. Thus, in both cases we are faced with a model selection problem; i.e. selection of non-linear principal components in KPCR and setting the regularization term ϑ in KRR, respectively. In KPCR one of the straightforward model selection criteria is based on choosing the first p principal components describing the predefined amount of overall variance.

Both methods can also be advantageous in noisy environments where the noise is spread in the eigendirections corresponding to small eigenvalues. We hypothesize, that in situations where these eigendirections represent mainly the noisy part of the signal, KPCR can be profitable due to the data not being projected onto these eigendirections. We discuss the topic of de-noising by PCA in the next section.

2.3 PCA De-Noising

White additive noise will change the covariance matrix of the investigated signal by adding a diagonal matrix, with corresponding variances of individual noise components on the diagonal. In the case of isotropic noise this will lead to the same increase of all eigenvalues computed from the clear signal. If the signal to noise ratio is sufficiently high we can assume that the noise will mainly affect the directions of the principal components corresponding to smaller eigenvalues. This allows us to discard the finite variance due to the noise by projection of the data onto the principal components corresponding to higher eigenvalues. However, a nonlinear transformation of the measured signal consisting of a signal and additive noise can smear the noise into certain directions. Thus, discarding the finite variance due to the noise can lead to a higher loss of the signal information; i.e. we have to deal with the balance between noise reduction and information loss. We have investigated this situation in the case of the noisy Mackey-Glass time series and the nonlinearity $\Phi(.)$ induced by using the Gaussian kernel. From Figure 1 (left) we can see that the noise increases the variance in the directions with smaller eigenvalues but decreases the variance in the main signal components. We can infer from this that a more uniform smearing of the

investigated signal into all directions was induced. Cutting the directions with the smaller eigenvalues will provide a level of noise reduction, however loss of information in the main signal direction will also appear.

3 Data Sample Construction

3.1 Chaotic Mackey-Glass Time-Series

The chaotic Mackey-Glass time-series is defined by the differential equation

$$\frac{ds(t)}{dt} = -bs(t) + a\frac{s(t-\tau)}{1+s(t-\tau)^{10}}$$

with $a = 0.2$, $b = 0.1$. The data were generated with $\tau = 17$ and using a second-order Runge-Kutta method with a step size 0.1. Training data is from t=200 to t=3200 while test data is in the range t= 5000 to 5500. To this generated time-series we added noise with normal distribution and with different levels corresponding to ratios of the standard deviation of the noise and the "clean" Mackey-Glass time-series.

3.2 Human Signal Detection Performance Monitoring

We have used Event Related Potentials (ERPs) and performance data from an earlier study [20, 21, 22]. Eight (A, B, ..., H) male Navy technicians experienced in the operation of display systems performed a signal detection task. Each technician was trained to a stable level of performance and tested in multiple blocks of 50–72 trials each on two separate days. Blocks were separated by 1-minute rest intervals. A set of 1000 trials were performed by each subject. Inter-trial intervals were of random duration with a mean of 3s and a range of 2.5–3.5s. The entire experiment was computer-controlled and performed with a 19-inch color CRT display (Figure 2). Triangular symbols subtending 42 minutes of arc and of three different luminance contrasts (0.17, 0.43, or 0.53) were presented parafoveally at a constant eccentricity of 2 degrees visual angle. One symbol was designated as the target, the other as the non-target. On some blocks, targets contained a central dot whereas the non-targets did not. However, the association of symbols to targets was alternated between blocks to prevent the development of automatic processing. A single symbol was presented per trial, at a randomly selected position on a 2-degree annulus. Fixation was monitored with an infrared eye tracking device. Subjects were required to classify the symbols as targets

or non-targets using button presses and then to indicate their subjective confidence on a 3-point scale using a 3-button mouse. Performance was measured as a linear composite of speed, accuracy, and confidence. A single measure, PF1, was derived using factor analysis of the performance data for all subjects, and validated within subjects. The computational formula for PF1 was

$$\text{PF1} = 0.33*\text{Accuracy} + 0.53*\text{Confidence} - 0.51*\text{Reaction Time}$$

using standard scores for accuracy, confidence, and reaction time based on the mean and variance of their distributions across all subjects. PF1 varied continuously, being high for fast, accurate, and confident responses and low for slow, inaccurate, and unconfident responses.

ERPs were recorded from midline frontal, central, and parietal electrodes (Fz, Cz, and Pz), referred to average mastoids, filtered digitally to a band-pass of 0.1 to 25 Hz, and decimated to a final sampling rate of 50 Hz. The prestimulus baseline (200 ms) was adjusted to zero to remove any DC off-set. Vertical and horizontal electrooculograms (EOG) were also recorded. Epochs containing artifacts were rejected and EOG-contaminated epochs were corrected. Furthermore, any trial in which no detection response or confidence rating was made by a subject was excluded along with the corresponding ERP.

Within each block of trials, a running-mean ERP was computed for each trial (Figure 3). Each running-mean ERP was the average of the ERPs over a window that included the current trial plus the 9 preceding trials for a maximum of 10 trials per average. Within this 10-trial window, a minimum of 7 artifact-free ERPs were required to compute the running-mean ERP. If fewer than 7 were available, the running mean for that trial was excluded. Thus each running mean was based on at least 7 but no more than 10 artifact-free ERPs. This 10-trial window corresponds to about 30s of task time. The PF1 scores for each trial were also averaged using the same running-mean window applied to the ERPs, excluding PF1 scores for trials in which ERPs were rejected. Prior to analysis, the running-mean ERPs were clipped to extend from time zero (stimulus onset time) to 1500 ms post-stimulus, for a total of 75 time points.

4 Results

The present work was carried out with Gaussian kernels; $K(\mathbf{x},\mathbf{y}) = e^{-(\frac{\|\mathbf{x}-\mathbf{y}\|^2}{L})}$, where L determines the width of the Gaussian function. The Gaussian kernel possesses a good smoothness properties (suppression of the higher frequency components) and in the case we do not have a priori knowledge about the regression problem we would prefer a smooth estimate [14, 10].

4.1 Chaotic Mackey-Glass Time-Series

On the (noisy) chaotic Mackey-Glass time-series we compared KPCR using the regressors extracted by Kernel PCA preprocessing with KRR. Both regression models were trained to predict the value at time $t+85$ from inputs at time $t, t-6, t-12, t-18$. The training data partitions were constructed by moving a "sliding window" over the 3000 training samples in steps of 500 samples. This window had two sizes - 500 samples and 1000 samples, respectively. This created six partitions of size 500 samples and five partitions of size 1000 samples. We estimated the variance of the overall clean training set and, based on this estimate $\hat{\sigma}^2 \doteq 0.05$, we repeated our simulations for the width L from the range $\langle 0.2\hat{\sigma}^2, 20\hat{\sigma}^2 \rangle$ using the step size 0.01. A fixed test set of size 500 data points (see Section 3.1) was used in all experiments. The regularization parameter ϑ in KRR was estimated by cross-validation using 20% of training data partitions for the validation set. In fact, to find the value of ϑ, we did the cross-validation in two steps. First the order of ϑ was estimated and then the finer structure of the values in the range ± 1 order was taken to estimate a "optimal" value of ϑ.

The performance of the regression models to predict a "clean" Mackey-Glass time series was evaluated in terms of the normalized root mean squared error (NRMSE). The best results on the test set averaged over all individual runs are summarized in Table 1. In Figure 4 we also compared the results on the noisy time series and their dependence on the width L of the Gaussian kernel. Although from Table 1 no significant differences can be noted between the KPCR and KRR methods, results in Figure 4 suggest that especially for a lower level of the noise the KPCR method provides slightly better results with smaller variance over different training data partitions.

A relatively small width L of the Gaussian kernel for which we observed the best performance of KPCR on test set suggests that for our Mackey-Glass time-series prediction problem with the Kernel PCA preprocessing step mainly local correlations of the data points on the attractor are taken

into account. Increasing the value of L leads to a faster decay of the eigenvalues (see e.g. [23]) and to the potential loss of the "finer" data structure due to a smaller number of the nonlinear principal components describing the same percentage of all the data variance. Increasing levels of the noise has the tendency to increase the optimal value for the L parameter which coincides with the intuitive assumption about smearing out the local structure.

The significant difference between the prediction accuracy on the clean and on the noisy Mackey-Glass time series gives rise to the question whether it is at all possible to sufficiently reduce the level of the noise in kernel space due to the violation of the additive and uncorrelated essence of the noise introduced by the nonlinear transformation. This may potentially have a stronger effect on the main principal components (see Figure 1 (left)). Therefore, we have to deal with the trade off between noise reduction and the associated signal information loss.

Method	*n/s=0.0%*		*n/s=11%*		*n/s=22%*	
	500	1000	500	1000	500	1000
KPCR	0.038	0.008	0.307	0.280	0.443	0.414
	(0.025)	(0.004)	(0.030)	(0.003)	(0.033)	(0.010)
KRR	0.038	0.007	0.312	0.279	0.446	0.404
	(0.024)	(0.003)	(0.032)	(0.010)	(0.036)	(0.006)

Table 1: The comparison of the approximation errors (NRMSE) of prediction for 2 different sizes of Mackey-Glass training set. The values represent an average of 6 simulations in the case of 500 training points and 5 simulations in the case of 1000 training points, respectively. Corresponding standard deviation is presented in parentheses. *n/s* represents the ratio between the standard deviation of the added Gaussian noise and the underlying time-series. For KPCR computed on 500 training points we used the first 495, 100 and 50 nonlinear principal components corresponding to the case of *n/s=0.0%, n/s=11%* and *n/s=22%*, respectively. For KPCR computed on 1000 training points we used the first 750, 125 and 75 nonlinear principal components.

The solution of the eigenvalue problem (1) can be numerically unstable when we are dealing with matrix $\mathbf{K}$ of higher dimensionality (in our case 1000×1000). However, on the noisy Mackey-Glass time series we observed that the best performance of KPCR was achieved using less than 150 main nonlinear principal components. This simply gives rise to the possibility

to use the reduced training data set to compute the main eigenvalues and eigenvectors and simply project the remaining training data points onto the extracted nonlinear principal components. In following experiments we compared the performance of KPCR when the whole training data set of size 1000 was used to estimate the main nonlinear principal components with the approach when the principal components were estimated from the first half of training data set. First, in Figure 1 (right) we compare the main 150 eigenvalues estimated from the first 500 data points with these computed from the 1000 data points. The small difference between both eigenspectra suggest that the first half of the training data set can sufficiently describe the sub-space of the feature space $\mathcal{F}$ which is generated by the nonlinear transformation of the time series. In Table 2 we compare the performance of both approaches. We cannot observe any significant degradation in performance when the reduced training data set is used to estimate the main principal components. However, from Table 2 we can also see that reducing the number of eigenvectors used to 495 in the case of the clean Mackey-Glass leads to a significant decrease of the overall performance (NRMSE 0.014) compared to the results in Table 1 where the best performance was achieved using 750 eigenvectors (NRMSE 0.008). We can conjecture that, although in the case of clean Mackey-Glass using some of the principal components corresponding to small eigenvalues may improve the overall performance, by adding noise to a time series these principal components are negatively affected and we can achieve better results by their removal. However, similar to the previous discussion this leads to signal information loss.

When extraction of a smaller subspace of the nonlinear principal components is desired we can also avoid the problem of direct diagonalization of the high dimensional Gram matrix $\mathbf{K}$ by using the approaches for iterative estimation of the principal components. In [19] we have successfully used the expectation maximization approach to Kernel PCA (EMKPCA) [24] which iteratively estimates only a subspace of the main principal components.

4.2 Human Signal Detection Performance Monitoring

The desired output PF1 was linearly normalized to have a range of 0 to 1. We trained the models on 50% of the ERPs and tested on the remaining data. The described results, for each setting of the parameters, are an average of 10 runs each on a different partition of training and testing data. To be consistent with the previous results reported in [20, 22] the validity of the models was measured in terms of normalized mean squared error (NMSE)

Method	*n/s=0.0%*	*n/s=11%*	*n/s=22%*
KPCR_{1000}	0.014 (0.005)	0.280 (0.003)	0.414 (0.010)
KPCR_{500}	0.017 (0.009)	0.282 (0.005)	0.414 (0.008)

Table 2: The comparison of the approximation errors (NRMSE) of the KPCR method using all 1000 training data points (KPCR_{1000}) to estimate eigenvectors and eigenvalues with the KPCR method where the first half (500) of the training points was used KPCR_{500}. In the later case, the rest of the training points was projected onto the estimated eigenvectors. The values represent an average of 5 simulations. Corresponding standard deviation is presented in parentheses. *n/s* represents the ratio between the standard deviation of the added Gaussian noise and the underlying time-series. We used the first 495, 125 and 75 nonlinear principal components corresponding to the case of *n/s=0.0%, n/s=11%* and *n/s=22%*, respectively.

and in terms of the proportion of data for which PF1 was correctly predicted with 10% tolerance (test proportion correct (TPC)); i.e ± 0.1 in our case.

First, the performance of SVR and KRR methods trained on data preprocessed by linear PCA (LPCA) in the input space was compared with the results achieved by using MLSVR and KPCR on features extracted by Kernel PCA[3]. In the next step we compared the MLSVR technique trained on selected nonlinear principal components with the SVR technique trained on all data points without PCA preprocessing.

We used $\epsilon = 0.01, \eta = 0.01$ parameters values for SVR models. In the case of KRR the regularization term ϑ was estimated by cross-validation using 20% of training data set as validation set. The same cross-validation strategy as applied on Mackey-Glass time series was used. The results achieved on subject A(891 ERPs), C(417 ERPs), D(702 ERPs), F(614 ERPs) and H(776 ERPs) are depicted in Figures 5–7. From Figures 5 and 6 we can see consistently better results on features extracted by Kernel PCA on subjects D and F. These superior results achieved using the Kernel PCA

[3]Although there exist several approaches for selection of the "best"subset of principal components [2], we used the criterion based on the amount of variance described by the selected principal components. In the case of linear PCA we used the sample covariance matrix to estimate principal components.

representation were also observed on the remaining 5 subjects. However, on subject C the performance with the features selected by linear PCA was slightly better. In the next step, for individual subjects, we selected the results for a Gaussian kernel width L on which KRR (with linear PCA preprocessed data) and KPCR (with Kernel PCA preprocessing) achieved the minimal NMSE on the test set. In Figure 8 a boxplot with lines at the lower quartile, median, and upper quartile values and a whisker plot for individual subjects is depicted. The boxplots suggest the differences between the results on subjects D to H. Using the sign test and the Wilcoxon matched-pairs signed-ranks test we tested the hypotheses about the *direction* and *size* of the differences within pairs. On subjects D to H the p-values < 0.03 indicate the statistically significant difference between the results achieved using linear PCA and Kernel PCA preprocessing steps. The alternative hypothesis regarding the superiority of LPCA leads to p-values < 0.02. Although both tests on subjects A, B and C did not show a statistically significant difference between the results (p-values between 0.11 and 0.75), the alternative Wilcoxon test about the superiority of LPCA leads to a higher p-value only on subject C (A - 0.12, B - 0.25, C - 0.88). Note that on subject C the smallest number of ERPs is available (417). Figure 8 also indicates the weakest results with the highest variance over individual runs. This result suggests that the number of ERPs from this subject were insufficient to model the desired dependencies between ERPs and the subject performance. Moreover, in this case the dimension of matrix $\mathbf{K}$ in the feature space $\mathcal{F}$ is lower (209) than the input dimensionality (225) and we so cannot exploit the advantage of Kernel PCA to improve overall performance by using more components in the feature space than the number available in the input space.

In Figure 7 we demonstrate that without the Kernel PCA preprocessing step in the feature space $\mathcal{F}$ we did not increase the overall performance. On the contrary, on subjects A, B and H the performance using the MLSVR method was slightly superior. On the remaining subjects the difference was insignificant. In the case of subject C, where the number of data points is less than the input dimensionality, SVR provides superior results over any of the methods considered which utilize Kernel PCA preprocessing.

In the next experiments we compared the SVR, KRR and KPCR methods on a data set using all eight subjects. We split the overall data set (5594 ERPs) into three different training (2765 ERPs) and testing (2829 ERPs) data pairs. 20% of the training data set was used for cross-validation to estimate ϵ, η and ϑ parameters in SVR and KRR, respectively. In the case of SVR the direct solution of the quadratic optimization problem to find

the $\boldsymbol{\gamma}, \boldsymbol{\gamma}^*$ and b coefficients (4) was replaced by using *SVMTorch* [25] algorithm designed to deal with a large-scale regression problems. In the case of KPCR the eigenvectors and eigenvalues were estimated using the EMKPCA approach with 30 EM steps. Based on the results reported in [19] we have used the 2600 main nonlinear principal components. A Gaussian kernel of width $L = 6000$ was used.

Table 3 summarizes the performance of the individual methods. We can see a slightly better performance achieved with the KPCR and KRR models in comparison to SVR. Together with the results achieved on individual subjects, results in Table 3 suggest that on this data set a Gaussian type of noise is more likely; i.e. the regression models with a quadratic cost function are preferable.

Method	NMSE	TPC
KPCR (*with EMKPCA*)	**0.1543**	**83.28**
KRR	**0.1546**	**83.50**
SVR (*with SVMTorch*)	**0.1611**	**82.76**

Table 3: The comparison of the NMSE and TPC prediction errors on the test set for the model based on all subjects ERPs. The values represent an average of 3 different simulations.

5 Conclusions

The Kernel PCA method for feature extraction has been investigated and the selected features were used in a regression problem. On the performance monitoring data set, in more than half of the cases, we demonstrated that the kernel regression methods with a (nonlinear) Kernel PCA preprocessing step provide significantly superior results over those with data preprocessed by linear PCA. Only in one case was an indication of the superiority of linear PCA observed, however, the sufficiency of the data representation in this case is questionable.

In contrast to [20] where one training (odd-numbered blocks of trials)-testing (even-numbered blocks of trials) data pair was used, in our study we created the different training-testing data partitions by random sampling from all blocks of trials. By using the kernel regression models on these data partitions we achieved approximately twice the level of improvement

in terms of TPC. This is a quite significant improvement on this biomedical application. However, in our future work the same data setting and representation (discrete wavelet transforms of ERPs) as reported in [20] will be used to make more objective conclusions.

Moreover, we have shown that reduction of the overall number of nonlinear principal components can reduce the noise present. Similar to the investigated Mackey-Glass time series prediction task, this can be exploited especially in the situation where the low-dimensional input data are spread in all directions and the noise reduction by projection to a lower number of linear principal components leads to information loss.

The solution of the eigenvalue problem (1) can be numerically difficult in the case of a high number of data samples. On the noisy Mackey-Glass time series we demonstrated that estimation of the main eigenvalues and eigenvectors can be sufficient from a smaller data representation. This implies a possibility to significantly reduce the computation and memory requirements and to deal with large-scale regression problems. Moreover, in such situations methods for the iterative estimation of the eigenvalues can also be efficiently used [24, 26].

On both data sets, by employing KPCR on the selected nonlinear principal components we demonstrated the comparable performance with KRR and SVR techniques. The computational cost of this approach is comparable with Kernel PCA as the estimation of the regression coefficients requires a diagonal matrix inversion of the order p. Moreover, the extracted regressors are linearly independent which is advantageous for subset selection techniques used in linear regression. Using various strategies (see e.g. [2, 11] and references therein) for deciding which nonlinear principal components to delete from the regression model can only improve the performance of the proposed KPCR model in the feature space $\mathcal{F}$.

Acknowledgments

The authors thank Professor Colin Fyfe for helpful discussions and comments. The first author is funded by a research grant for the project "Objective Measures of Depth of Anaesthesia"; University of Paisley and Glasgow Western Infirmary NHS trust, and is partially supported by Slovak Grant Agency for Science (grants No. 2/5088/00 and No. 00/5305/468). Data were obtained under a grant from the US Navy Office of Naval Research (PE60115N), monitored by Joel Davis and Harold Hawkins. Dr. Trejo was

supported by the NASA Aerospace Operations Systems Program and by the NASA Intelligent Systems Program.

References

[1] Schölkopf B, Smola AJ, Müller KR. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. Neural Computation 1998; 10:1299–1319

[2] Jolliffe IT. Principal Component Analysis. Springer-Verlag, New York, 1986

[3] Vapnik V. The Nature of Statistical Learning Theory. Springer, New York, 1998

[4] Smola AJ, Schölkopf B. A Tutorial on Support Vector Regression. Technical Report NC2-TR-1998-030, NeuroColt2 Technical Report Series, 1998.

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

[6] Rosipal R, Girolami M, Trejo LJ. Kernel PCA for Feature Extraction of Event-Related Potentials for Human Signal Detection Performance. In: Proceedings of ANNIMAB-1 Conference, Götegorg, Sweden, Springer, 2000, pp.321–326.

[7] Saunders C, Gammerman A, Vovk V. Ridge Regression Learning Algorithm in Dual Variables. In: Proceedings of the 15th International Conference on Machine Learning, 1998

[8] Frank IE, Friedman JH. A Statistical View of Some Chemometrics Regression Tools. Technometrics 1993; 35:109–147

[9] Montgomery DC, Peck EA. Introduction to Linear Regression Analysis, 2nd edn. John Wiley & Sons, 1992

[10] Smola AJ, Schölkopf B, Müller KR. The connection between regularization operators and support vector kernels. Neural Networks 1998; 11:637–649

[11] Rosipal R, Girolami M, Trejo LJ. On Kernel Principal Component Regression with Covariance Inflation Criterion for Model Selection. Technical report, CIS, University of Paisley, 2000.

[12] Jolliffe IT. A Note on the Use of Principal Components in Regression. Applied Statistics 1982; 31:300–302

[13] Tibshirani R, Knight K. The covariance inflation criterion for adaptive model selection. J R Statist Soc B 1999; 61:529–546

[14] Girosi F, Jones M, Poggio T. Priors, stabilizers and basis functions: From regularization to radial, tensor and additive splines. Technical Report A.I. Memo No. 1430, MIT, 1993

[15] Williams CKI. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In: Jordan MI (eds.). Learning and Inference in Graphical Models. Kluwer, 1998.

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

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

[18] Suykens JAK, Lukas L, Vandewalle J. Sparse approximation using least squares support vector machines. In: IEEE International Symposium on Circuits and Systems ISCAS'2000, 2000.

[19] Rosipal R, Trejo LJ, Cichocki A. Kernel Principal Component Regression with EM Approach to Nonlinear Principal Components Extraction. Technical report, CIS, University of Paisley, 2000.

[20] Trejo LJ, Shensa MJ. Feature Extraction of ERPs Using Wavelets: An Application to Human Performance Monitoring. Brain and Language 1999; 66:89-107

[21] Trejo LJ, Kramer AF, Arnold JA. Event-related Potentials as Indices of Display-monitoring Performance. Biological Psychology 1995; 40:33–71

[22] Koska M, Rosipal R, König A, Trejo LJ. Estimation of human signal detection performance from ERPs using feed-forward network model. In: Computer Intensive Methods in Control and Signal Processing, The Curse of Dimensionality. Birkhauser, Boston, 1997.

[23] Williamson RC, Smola AJ, Schölkopf B. Generalization performance of regularization networks and support vector machines via entropy numbers of compact operators. Technical Report NC-TR-98-019, NeuroCOLT, Royal Holloway College, 1998.

[24] Rosipal R, Girolami M. An Expectation-Maximization Approach to Nonlinear Component Analysis. to appear *Neural Computation*, 2000.

[25] Collobert R, Bengio S. Support Vector Machines for Large-Scale Regression Problems. Technical Report 00-17, IDIAP, 2000

[26] Golub GH, van Loan ChF. Matrix Computations. The John Hopkins University Press, London, 1996

Appendix A

In Section 2.1 we assumed that we are dealing with centralized data $\Phi(\mathbf{x})$ in a feature space. In practical computation, the centralization of the data leads to the modification of (1) to the form [1]

$$n\tilde{\lambda}\tilde{\boldsymbol{\alpha}} = \tilde{\mathbf{K}}\tilde{\boldsymbol{\alpha}}, \tag{15}$$

where the requirement of centralized data $\Phi(\mathbf{x})$ was transformed to the change of $\mathbf{K}$ matrix to $\tilde{\mathbf{K}} = \mathbf{K} - \mathbf{1}_n\mathbf{K} - \mathbf{K}\mathbf{1}_n + \mathbf{1}_n\mathbf{K}\mathbf{1}_n$ where $\mathbf{1}_n$ is an $(n \times n)$ matrix of $1/n$ elements. Similarly we have to change the $(n_t \times n)$ "test" matrix $\mathbf{K}^{test}$ whose elements are $\mathrm{K}_{ij}^{test} := K(\mathbf{x}_i, \mathbf{x}_j)$ where $\{\mathbf{x}_i\}_{i=1}^{n_t}$ and $\{\mathbf{x}_j\}_{j=1}^{n}$ are testing and training points, respectively. The centralization of the matrix $\mathbf{K}^{test}$ is given by $\tilde{\mathbf{K}}^{test} = \mathbf{K}^{test} - \mathbf{1}_{n_t}\mathbf{K} - \mathbf{K}^{test}\mathbf{1}_n + \mathbf{1}_{n_t}\mathbf{K}\mathbf{1}_n$ where $\mathbf{1}_{n_t}$ is now an $(n_t \times n)$ matrix with the same entries $1/n$.

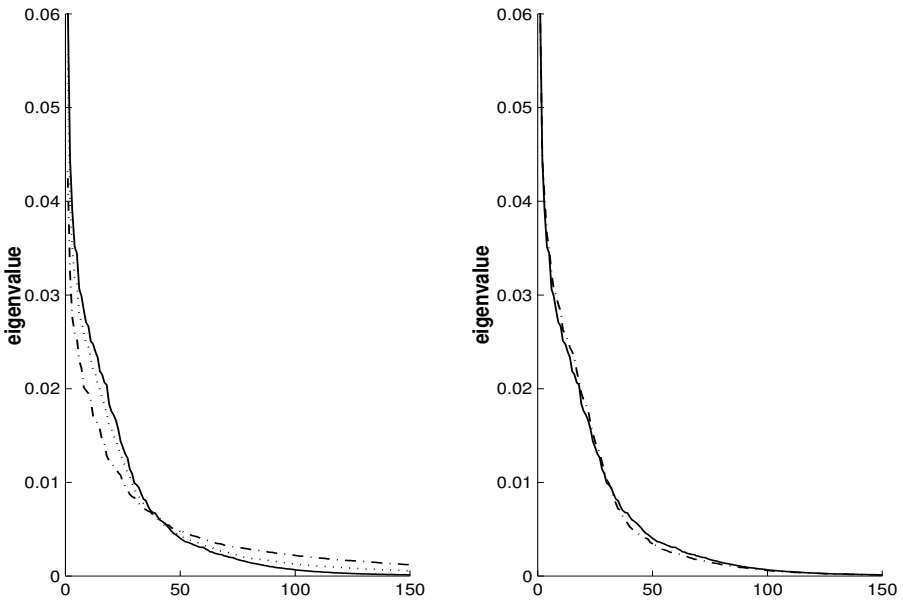

Figure 1: *Left:* Eigenvalues computed from embedded Mackey-Glass time series transformed to kernel space. Different levels of noise were added (*n/s* represents the ratio between standard deviation of the noise and signal, respectively); *n/s=0%* (solid line), *n/s=11%* (dots), *n/s=22%* (dash dotted line). *Right:* Comparison of the eigenvalues computed from 500 (solid line) and 1000 (dash dotted line) data samples.

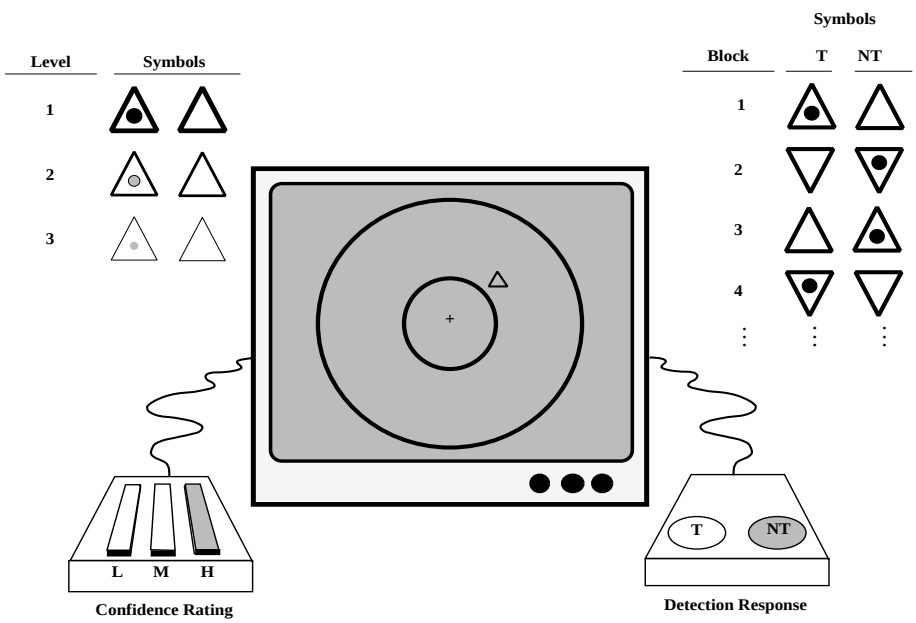

Figure 2: Display, input device configuration and symbols for task-relevant stimuli for the signal detection task.

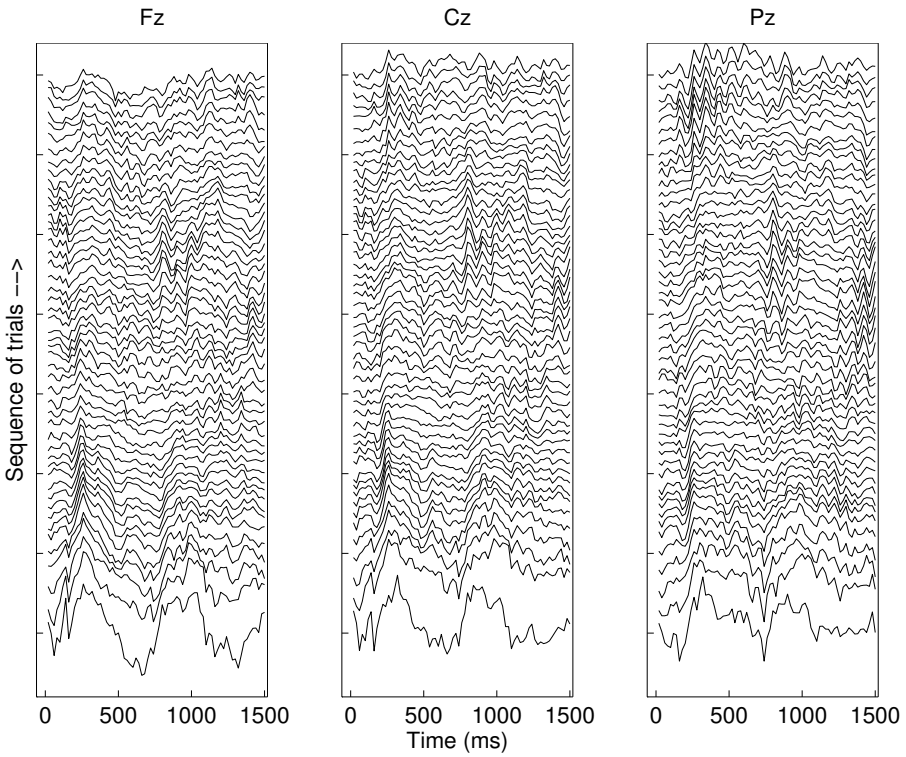

Figure 3: Running-mean ERPs at sites Fz, Cz and Pz for subject B in the first 50 running-mean ERPs.

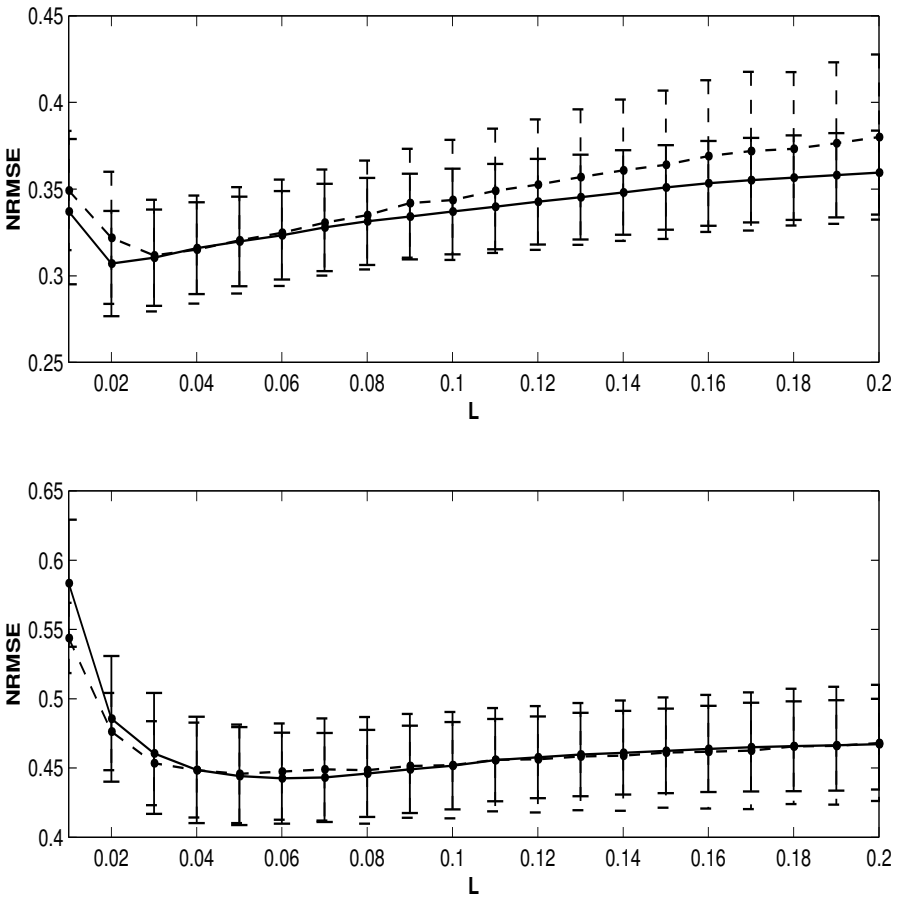

Figure 4: Comparison of the results achieved on the noisy Mackey-Glass time series with the KPCR (solid) and KRR (dashed) methods. Six different training sets of size 500 data points were used. The performance for different widths (L) of the Gaussian kernel is compared in normalized root mean squared error (NRMSE) terms. *Top*: *n/s=11%*. *Bottom*: *n/s=22%*. *n/s* represents the ratio between the standard deviation of the added Gaussian noise and the underlying time-series.

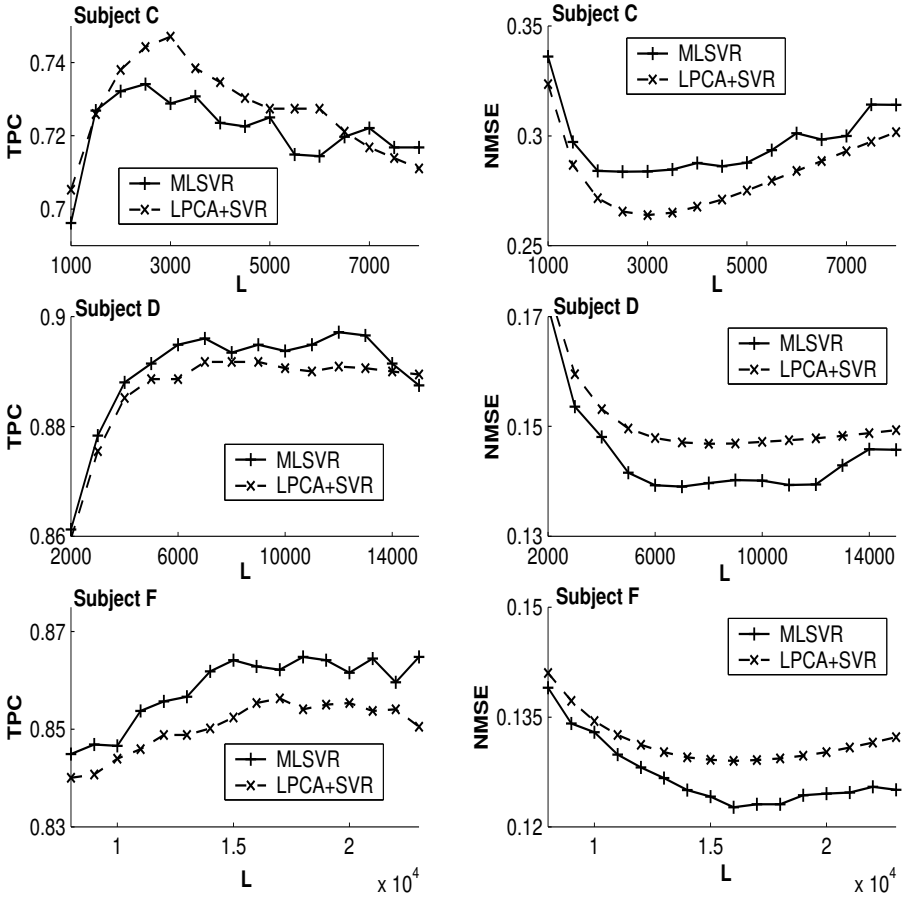

Figure 5: Comparison of the results achieved on subjects C, D and F with MLSVR and SVR on data preprocessed by linear PCA (LPCA + SVR), respectively. In both cases the principal components describing 99% of variance were used. The performance for the different widths (L) of the Gaussian kernel is compared in terms of test proportion correct (TPC) and normalized mean squared error (NMSE).

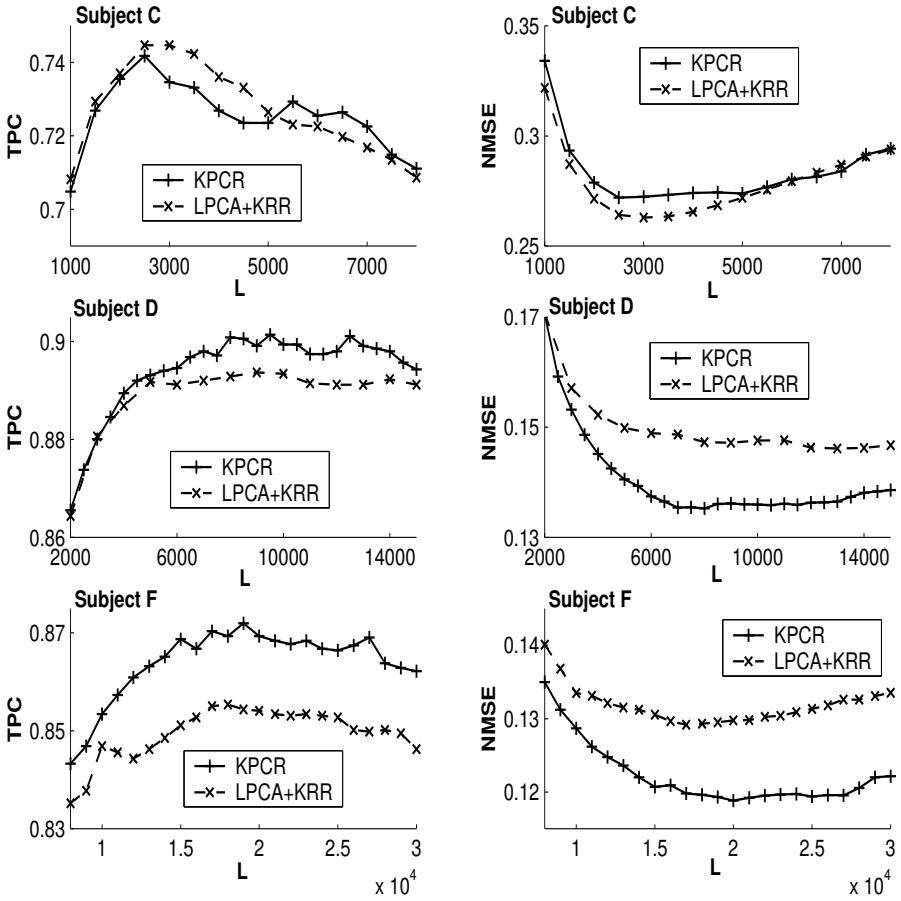

Figure 6: Comparison of the results achieved on subjects C, D and F with KPCR and KRR on data preprocessed by linear PCA (LPCA + KRR), respectively. In both cases the principal components describing 99% of variance were used. The performance for the different widths (L) of the Gaussian kernel is compared in terms of test proportion correct (TPC) and normalized mean squared error (NMSE).

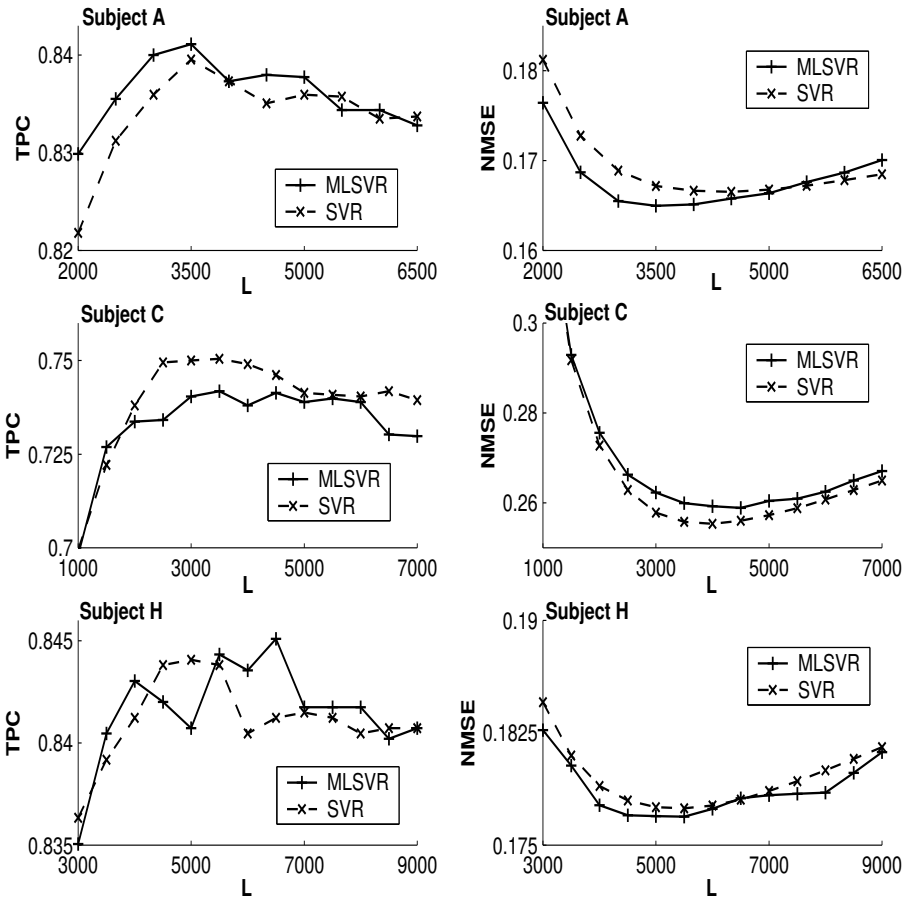

Figure 7: Comparison of the MLSVR and SVR on subjects A, C and H. 90% of all nonlinear principal components were used in the case of MLSVR. The performance for the different widths (L) of the Gaussian kernel is compared in terms of test proportion correct (TPC) and normalized mean squared error (NMSE).

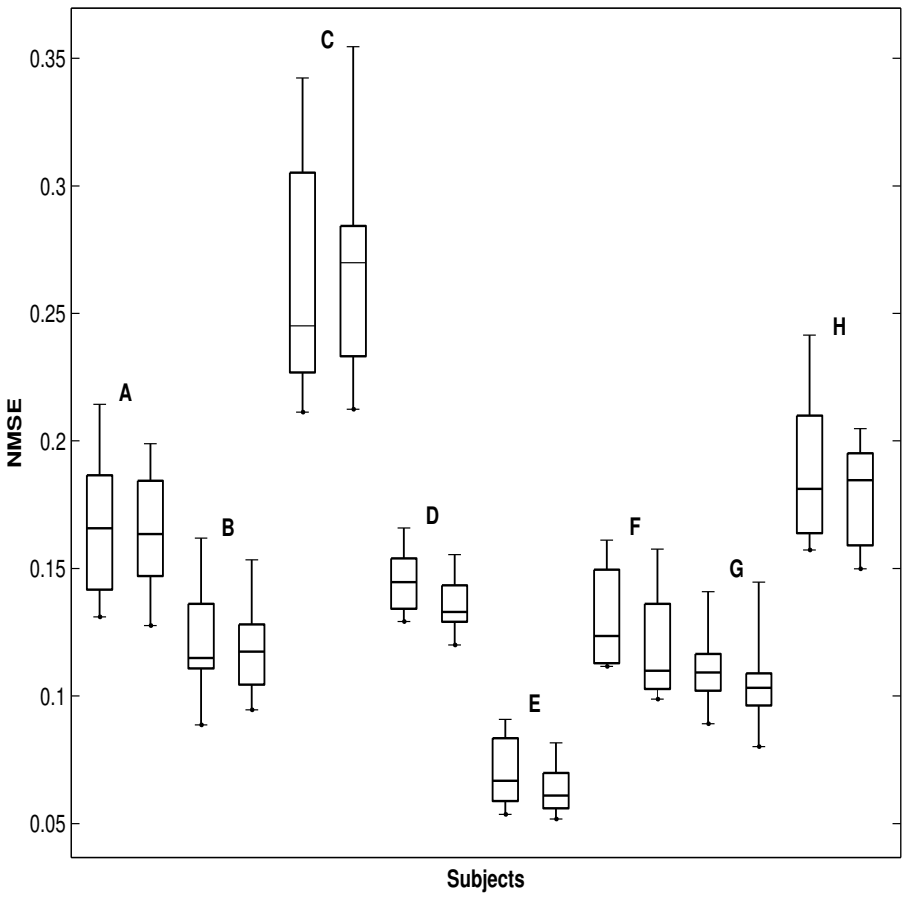

Figure 8: Boxplots with lines at the lower quartile, median, and upper quartile values and whisker plot for subjects A to H. The performance of KRR with LPCA preprocessing step (left-hand boxplots) is compared with KPCR on data preprocessed by KPCA (right-hand boxplots) in terms of normalized mean squared error (NMSE). The boxplots are computed on results from 10 different runs using the widths of the Gaussian kernel on which the methods achieved minimal NMSE on test set.