Chapter 3. Subspace Techniques and Biomedical Time Series Analysis

A. M. Tomé[a,*], A. R. Teixeira[a], E. W. Lang[b]

[a]IEETA/DETI, Universidade de Aveiro, P-3810-193 Aveiro, Portugal
[b]CIML Group, Biophysics, University of Regensburg, D-93040 Regensburg, Germany
*E-mail: ana@ieeta.pt

Abstract: The application of subspace techniques to univariate (single-sensor) biomedical time series is presented. Both linear and non-linear methods are described using algebraic models, and the dot product is the most important operation concerning data manipulations. The covariance/correlation matrices, computed in the space of time-delayed coordinates or in a feature space created by a non-linear mapping, are employed to deduce orthogonal models. Linear methods encompass singular spectrum analysis (SSA), singular value decomposition (SVD) or principal component analysis (PCA). Local SSA is a variant of SSA which can approximate non-linear trajectories of the embedded signal by introducing a clustering step. Generically non-linear methods encompass kernel principal component analysis (KPCA) and greedy KPCA. The latter is a variant where the subspace model is based on a selected subset of data only.

Key words: Kernel methods, projective subspace techniques, time series analysis

1. Projective Techniques

Subspace techniques are widely used in many fields of research like face recognition and related computer vision tasks, denoising applications like speech enhancement and so on. Projective techniques generate alternative data representations that can be interpreted more easily or which facilitate further processing of the data. The application of these techniques to univariate, single sensor time series is presented. The linear and non-linear subspace models are integrated into a common algebraic framework, where the model is described by a matrix and data manipulations are mostly achieved using dot products and matrix/vector additions.

The linear projective models are described by a matrix, or a couple of matrices, and generally comprise three steps:

- the projection of the multidimensional data $\mathbf{x} \in \Re^M$ onto the subspace model ($\mathbf{U}$)
- the selection of the relevant components
- the eventual reconstruction of the, possibly modified, signal in the original domain.

The projecting matrix $\mathbf{U}$, whose columns represent basis vectors $\mathbf{u}_m, m = 1 \ldots M$ in an M dimensional space, will transform any input data vector $\mathbf{x}$ according to

$$\mathbf{y} = \mathbf{U}^T\mathbf{x} \tag{1}$$

where the entries of $\mathbf{y}$ result from the dot product between the basis vectors and data vector $\mathbf{x}$. The projections $\mathbf{y}$ constitute a new representation of the data, and often only a subset of the entries constitute interesting components. The remaining components often are related with noise. Moreover, there are applications where the projective technique is applied to extract informative components and simultaneously perform a dimension reduction by neglecting the remaining components. However, in other applications a reconstruction step is required by manipulating the selected components in order to switch to the input space of the measured data. The selection and reconstruction steps can be written as

$$\hat{\mathbf{x}} = \mathbf{BPy} = \mathbf{BPU}^T\mathbf{x} \tag{2}$$

Here $\mathbf{P}$ is a diagonal matrix with the *mth* diagonal entry equal to 1 if the *mth* entry of $\mathbf{y}$ is to be selected or equal to 0 if it is to be removed. Note that after the product $\mathbf{Py}$, a new vector is obtained where the entries of $\mathbf{y}$ to be removed are substituted by 0 while the other entries are copied to the new vector. Then it is possible to consider that $\hat{\mathbf{x}}$ is a linear combination of the columns of matrix $\mathbf{B}$, where the weights form the entries of the new vector. Furthermore, the reconstruction matrix $\mathbf{B}$ is the inverse, or pseudo-inverse, of $\mathbf{U}$. Hence, if all components of $\mathbf{y}$ are selected, all diagonal elements of the selection matrix $\mathbf{P}$ are one and thus the identity matrix $\mathbf{I}$ results

$$\mathbf{BPU}^T = \mathbf{BU}^T = \mathbf{I}$$

The most widely used techniques to compute $\mathbf{U}$ are either principal component analysis (PCA), blind source separation (BSS) or independent component analysis (ICA). In the first case the projection matrix has orthogonal columns, i.e. $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ and $\mathbf{B} = \mathbf{U}$. It is also possible to identify the components which keep the largest variance of the raw data. BSS separation methods also achieve an non-orthogonal matrix but often only use second-order statistics to estimate the components of the

All rights reserved - © 2011 Bentham Science Publishers Ltd.

data [1], [2], [3]. The goal of ICA is a decomposition into statistically independent components and the projecting matrix is usually a non-orthogonal matrix. All these methods are considered linear methods as the original data can be approximated by a linear combination of basis vectors, the columns of $\mathbf{U}$, or each entry of $\hat{\mathbf{x}}$ a linear combination the extracted components (sources).

Projective Techniques based on kernel functions rely on a nonlinear mapping of the data to extract non-linear components. However, several adaptations need to be accomplished in order to perform data manipulations without an explicit mapping of the data. The key idea is to substitute the dot product in feature space, created by the nonlinear mapping of the data, by the evaluation of a related kernel function. Therefore, considering that the mapped training data set is given by $\Phi = \Phi(\mathbf{X}) = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2) \dots \phi(\mathbf{x}_K)]$ and the dual form of the subspace model, obtained as a linear combination of the data set, is represented by

$$\mathbf{U} = \mathbf{\Phi A}, \tag{3}$$

the projections in the feature space read

$$\mathbf{y} = \mathbf{A}^T \mathbf{\Phi}^T \phi(\mathbf{x}) \tag{4}$$

Note that the subspace model $\mathbf{U}$ is not explicitly computed, but the coefficients $\mathbf{A}$ and the training data set $\mathbf{X}$ need to be available to describe the model. Besides that, the dot products of eqn 4 are evaluated by the application of the kernel function. The most widely used technique that follows this approach is kernel principal component analysis (KPCA) where the subspace model $\mathbf{U}$ is also an orthogonal matrix as in the linear case. The projections $\mathbf{y}$ are values in feature space which represent non-linear components of the data. The reconstruction in the feature space is also formulated as in PCA by

$$\hat{\phi}(\mathbf{x}) = \mathbf{UPy} \tag{5}$$

where $\mathbf{P}$ is defined as in the linear case. However, the reconstructed point is never computed explicitly. Rather it is integrated into a common step, called pre-image problem, whose goal is to achieve the image of the reconstructed point in the original space. Therefore, reconstruction and pre-image are formalized into one step via the optimization of a criterium which again exploits dot product operations. For instance, using the Euclidian distance

$$\begin{aligned} \tilde{d}^{(2)} &= \|\phi(\mathbf{p}) - \hat{\phi}(\mathbf{x})\|^2 \\ &= (\phi(\mathbf{p}) - \hat{\phi}(\mathbf{x}))^T (\phi(\mathbf{p}) - \hat{\phi}(\mathbf{x})) \end{aligned} \tag{6}$$

where $\phi(\mathbf{p})$ represents a yet unknown point $\mathbf{p}$ which must be close to the reconstructed point $\hat{\phi}(\mathbf{x})$. The manipulation of eqn (6) leads to a sum of dot products which then are evaluated by using the kernel function. The most widely used techniques consist in iteratively finding $\mathbf{p}$ that minimizes eqn. (6), but algebraic solutions are also proposed [4].

2. Singular Value Decomposition and Orthonormal Models

The data manipulations follow very similar principles either using linear or nonlinear models. As mentioned before, the estimation of the subspace model $\mathbf{U}$ can be based on different assumptions. However, in this work, the focus will be exclusively on methods that rely on the eigendecomposition of matrices which exploit second order correlations or dots products within the data set. Without loss of generality we can assume that the $M \times K$ - dimensional data set $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_K]$ is centered. Therefore each row of the data matrix has zero mean and the non-normalized correlation matrix is given by

$$\mathbf{S} = \mathbf{X}\mathbf{X}^T \tag{7}$$

Its eigendecomposition $\mathbf{S} = \mathbf{UDU}^T$ is used to compute the subspace model $\mathbf{U}$. In that case the dimension, $M \times M$, of the subspace model corresponds to the number of rows of the data matrix $\mathbf{X}$. And the columns of the subspace model $\mathbf{U}$ form an *orthonormal basis* in the space of the data, i. e. $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. This model is also called Principal Component Analysis (PCA) and by arranging in descending order the eigenvalues, i.e. the diagonal entries of matrix $\mathbf{D}$, the corresponding columns of $\mathbf{U}$ have the eigenvectors aligned with the directions of decreasing variability of the data. The first eigenvector corresponds to the largest variance, called the principal direction of the data, the second column to the second largest and so on. The important issue is that the signals do not spread equally along all directions of the $M-dimensional$ space. Hence, by selecting only the principal directions, noise reduction can be achieved.

An alternative to the correlation matrix, which is formed by an outer product of the data vectors, is obtained by computing the matrix of dots products, called kernel matrix, which is formed by an inner product of the data vectors

$$\mathbf{K} = \mathbf{X}^T\mathbf{X} \tag{8}$$

whose eigendecomposition is given by $\mathbf{K} = \mathbf{VDV}^T$, where the *orthonormal* eigenvectors form the columns of $\mathbf{V}$. The related eigenvalues are again the diagonal entries of matrix $\mathbf{D}$, and their non-zero values coincide with the eigenvalues of the correlation matrix $\mathbf{S}$. Therefore the Singular Value Decomposition (SVD) of the data matrix allows to relate the eigenvectors of the correlation matrix with the ones of the kernel matrix. The SVD of the data matrix is obtained via

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{9}$$

where the diagonal matrix $\mathbf{\Sigma} = \mathbf{D}^{1/2}$ is the matrix of singular values which correspond to the square root of the eigenvalues of both the correlation matrix and the kernel matrix. Then by multiplying both sides of the previous equation by $\mathbf{V}\mathbf{D}^{-1/2}$, the *dual form of the subspace model* is obtained

$$\mathbf{U} = \mathbf{X}\mathbf{V}\mathbf{D}^{-1/2} \tag{10}$$

The dual form is used advantageously when the number M of rows of the data matrix $\mathbf{X}$ is larger than the number K of columns, i.e. $(M > K)$, or if it is intended to substitute the dot product in input space by the dot product in a space created by the non-linear mapping of the data. The dimension of the subspace model $\mathbf{U}$ described by the dual form is determined by the number of non-zero eigenvalues of the kernel matrix. The eigendecomposition of the kernel matrix will have at most $min(M, K)$ non-zero eigenvalues. Consequently, the subspace model will be represented by an $M \times min(M, K)$ matrix $\mathbf{U}$. Note, that if $M > K$, the number of non-zero eigenvalues of the correlation matrix $\mathbf{S}$ is also at most K.

3. Linear Models

Given a data set $\mathbf{X}$, the orthogonal subspace model $\mathbf{U}$ is computed as explained in the previous section. The data matrix $\mathbf{X}$ can be linearly transformed by

$$\mathbf{Y} = \mathbf{U}^T\mathbf{X} \tag{11}$$

where the m-th row of $\mathbf{Y}$ represents the dot product between the m-th column $\mathbf{u}_m^T$ of $\mathbf{U}$ with each vector $\mathbf{x}_k, k = 1, \ldots K$ of the data set $\mathbf{X}$. Having the eigenvalues organized into descending order, the corresponding columns of the subspace model are aligned according to the variance of the raw data. The first eigenvector, representing the principal direction, corresponds to the largest variance, the second eigenvector to the second largest variance and so on. Substituting eqn. (9) into the eqn.(11) and computing the m-th row of $\mathbf{Y}$, one obtaines

$$\mathbf{Y}_m = \mathbf{u}_m^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \lambda_m^{1/2}\mathbf{v}_m^T \tag{12}$$

where λ_m is the eigenvalue that corresponds to the *mth* eigenvector $\mathbf{u}_m$. The energy of the data projections is then equal to the corresponding eigenvalues. Further notice that the new representation of the data, $\mathbf{Y}\mathbf{Y}^T = \mathbf{\Sigma}^2 = \mathbf{D}$ is uncorrelated, i.e. the rows of $\mathbf{Y}$ are *orthogonal*.

The data manipulations, leading to a reconstructed version of the data set, can be written as

$$\hat{\mathbf{X}} = \mathbf{U}\mathbf{P}\mathbf{Y} = \mathbf{U}\mathbf{P}\mathbf{U}^T\mathbf{X} \tag{13}$$

where the selection matrix $\mathbf{P}$ is a diagonal matrix which represents the selection process: with the m-th diagonal entry equal to $p_{mm} = 1$ if the m-th row of $\mathbf{Y}$ is to be selected, and equal to $p_{mm} = 0$ if it is to be neglected.

Assuming $M < N$, there are at most M eigenvalues/singular-values different of zero [5]. Then eqn. (13) can be expressed as a sum of components corresponding to the vectors of the subspace model related with the non-zero eigenvalues. Substituting the subspace model by $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M]$, and using block matrix operations, those components are $M \times K$ matrices of rank one. Therefore, the eqn. (13) can be re-written as

$$\hat{\mathbf{X}} = \mathbf{u}_1 p_{11}\mathbf{u}_1^T\mathbf{X} + \mathbf{u}_2 p_{22}\mathbf{u}_2^T\mathbf{X} + \ldots + \mathbf{u}_M p_{MM}\mathbf{u}_M^T\mathbf{X} \tag{14}$$

If all components are selected, i.e, $p_{mm} = 1 \ \forall \ m$, the matrix $\hat{\mathbf{X}} = \mathbf{X}$. Otherwise the reconstruction error, computed as the mean square error (MSE), can be related with the discarded eigenvalues [5].

3.1. Eigenspectrum Scaling

In denoising applications often the selected components correspond to the $m = 1, \ldots, L$ largest eigenvalues of the correlation matrix $\mathbf{S}$. Assuming the eigenvalues are arranged in descending order, i.e. $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_M$, one possible strategy to estimate L is to define a threshold *th* in the range of $80 - 95$ and compute the following inequality

$$\frac{\lambda_1 + \lambda_2 + \ldots + \lambda_L}{\lambda_1 + \lambda_2 + \ldots + \lambda_M} * 100 > th \tag{15}$$

Therefore L is computed by defining the percentage of variance of the data that should be preserved. Other alternatives to estimate L are based on model order selection strategies based on the minimum description length (MDL) criterion, Akaike's information criterion (AIC) or the Bayes information criterion (BIC) [6], [7].

The eigenspectrum, i.e. the distribution of eigenvalues, of $\hat{\mathbf{X}}$ is a truncated version of the eigenspectrum of the original data matrix $\mathbf{X}$ whenever $p_{mm} = 1$, if $m \leq L$, and $p_{mm} = 0$ if $m > L$. But the values $p_{mm}, m \leq L$ can also be assigned such that there is a compensation to the possible noise contributions to the related directions. In that case, p_{mm} depends on the corresponding eigenvalue (λ_m) and the variance (η) of the noise. For instance, with $p_{mm} = \sqrt{1 - \eta/\lambda_m}$, with $\lambda_m > \eta$, the projection of the data set onto $\mathbf{u}_m$ reads

$$\mathbf{Y}_m = p_{mm}\mathbf{u}_m^T\mathbf{X} = p_{mm}\lambda_m^{1/2}\mathbf{v}_m = (\lambda_m - \eta)^{\frac{1}{2}}\mathbf{v}_m^T \tag{16}$$

where $\mathbf{Y}_m$ is the m-th row of $\mathbf{Y}$ whose energy is affected by the noise variance. Other possibilities to compute the values of p_{mm} can be found, also assuming that the variance of the noise should be smaller than the selected eigenvalues [8], [9]. The table 1 shows alternatives to compute $p_{mm}, m \leq L$ applied in speech enhancement applications, and also in the principal component analysis stages of ICA algorithms [1].

Table 1: Different alternatives to compute $p_{mm},\ m \leq L$ and eventually change the eigenspectrum of the reconstructed data

Least squares	$p_{mm} = 1$
Modified Least Squares	$p_{mm} = \sqrt{1 - \eta/\lambda_m}$
Minimum Variance	$p_{mm} = (1 - \eta/\lambda_m)$

3.2. *Time series and subspace models*

Subspace models rely on a multidimensional representation of the data. In multi-sensor, often also called multi-channel, signal processing, the data vector $\mathbf{x}$ is naturally formed with samples of the different sensors. However, projective subspace techniques can also be applied to single-sensor, hence single-channel, signals by forming vectors with sliding windows of the signal. These embedding methods can be found in literature under distinct names depending on the domain of application: Singular Spectrum Analysis (SSA), as for instance in climate time series analysis [10], [11], and SVD, as for instance in speech enhancement [12], [8], [9]. The aim of SSA is to achieve a decomposition of the original time series or signal into a sum of a small number of interpretable components such as a slowly varying trend, oscillatory components and noise. While the aim of Speech Enhancement is simply to eliminate noise which is related with the components related with the smallest eigenvalues.

3.2.1. *Embedding*

Time series analysis techniques often rely on embedding a one-dimensional sensor signal in the space of its time-delayed coordinates. Embedding can be regarded as a mapping that transforms the one-dimensional time series to a multi-dimensional sequence of lagged vectors. Considering a segment of a signal $(x[0], x[1], \ldots, x[N-1])$, the multidimensional signal is obtained by $\mathbf{x}_k = (x[k-1+M-1], \ldots, x[k-1])^T, k = 1, \ldots, K$. The lagged vectors lie in a space of dimension M and constitute the columns of the related *trajectory matrix* $\mathbf{X}$ [13]:

$$\mathbf{X} = \begin{bmatrix} x[M-1] & x[M] & \ldots & x[N-1] \\ x[M-2] & x[M-1] & \ldots & x[N-2] \\ x[M-3] & x[M-2] & \ldots & x[N-3] \\ \vdots & \vdots & \ddots & \vdots \\ x[0] & x[1] & \ldots & x[N-M] \end{bmatrix} \quad (17)$$

Note that the trajectory matrix $\mathbf{X}$ has identical entries along its diagonals, thus forming a *Toeplitz matrix*. There are other alternatives to build the data matrix via embedding the signal in an $M-dimensional$ space such that the resulting data matrix has identical elements along its anti-diagonals, thus forming a *Hankel matrix* [8], [10]. However, the processing steps are identical and only need to be adapted to cope with the differences in data organization.

3.2.2. *Reverse Embedding*

After applying the subspace model to the trajectory matrix, the reconstructed version $\hat{\mathbf{X}}$ does in general not exhibit identical elements along each of its descending diagonals, like in case of the original trajectory matrix $\mathbf{X}$. In singular spectrum analysis (SSA) these differing entries in each diagonal (or anti-diagonal) are replaced by their average recovering again a Toeplitz (or an Hankel) matrix $\mathbf{X}_r$. An univariate signal $\hat{x}[n]$ is then obtained by reverting the embedding, i.e. by forming the signal with the mean of the values along each descendent diagonal of $\hat{\mathbf{X}}$. Also it is possible to show that the projection and the reconstruction steps, including the diagonal averaging, form a zero-phase filter bank. This filter bank analysis/synthesis interpretation, where the projection step represents the analysis block and the reconstruction step with diagonal averaging is the synthesis block, will be discussed in the following from a linear systems perspective.

3.3. *Linear Subspace Models and Filter Banks*

As mentioned before, each row $\mathbf{Y}_m, m = 1, 2, \ldots M$, of the projected data $\mathbf{Y}$ is obtained by projecting the set of data vectors onto the vectors spanning the subspace. Thus each element of the $1 \times K$ matrix $\mathbf{Y}_m$ is the dot product of the m-th eigenvector $\mathbf{u}_m$ and a column $\mathbf{x}_k$ of the data matrix $\mathbf{X}$. This operation can be formulated as the weighted sum of a sequence of samples of the original time series,

$$y_m[n] = \sum_{i=1}^{M} u_{im} x[n-i+1] \quad (18)$$

where $(M-1) \leq n < N$, and $y_m[n]$ are the rows of matrix $\mathbf{Y}_m$ taken in their natural order. Therefore, the row vector $\mathbf{Y}_m$ has K samples starting with time index $(M-1)$, similar to the first row of the trajectory matrix $\mathbf{X}$.

The entries of the vector $\mathbf{u}_m$, representing the $m-th$ column of the subspace model, correspond to the coefficients of a finite impulse response (FIR) filter. The transfer function of the analysis step of the filter can be computed by substituting every delay operation in eqn 18 by the corresponding z transform. For instance, $x[n \pm d]$ by $z^{\pm d} X(z)$ and $y_m[n]$ by $Y_m(z)$ [14], [15], yielding:

$$H_m(z) = \frac{Y_m(z)}{X(z)} = (u_{1m} + u_{2m} z^{-1} + \ldots u_{Mm} z^{-(M-1)}) \quad (19)$$

The transfer function $H_m(z), m = 1, \ldots, M$ is an output-input ratio and constitutes the analysis block of the filter as it decomposes the input into several components $y_m[n], m = 1, \ldots, M$.

Using the terminology of filter banks, the analysis filters are followed by synthesis filters which recombine the components. To facilitate the exposition of the filter operation, the reconstruction of one of the terms of eqn. 14 with $p_{mm} = 1$, will be given

$$\mathbf{A}_m = \mathbf{u}_m \mathbf{u}_m^T \mathbf{X} = \mathbf{u}_m \mathbf{Y}_m \tag{20}$$

Therefore each row of the rank-one matrix is a scaled version of the rank-one matrix $\mathbf{Y}_m$

$$\begin{bmatrix} u_{1m}y_m[M-1] & u_{1m}y_m[M] & \dots & u_{1m}y_m[N-1] \\ u_{2m}y_m[M-1] & u_{2m}y_m[M] & \dots & u_{2m}y_m[N-1] \\ u_{3m}y_m[M-1] & u_{3m}y_m[M] & \dots & u_{3m}y_m[N-1] \\ \vdots & \vdots & & \vdots \\ u_{Mm}y_m[M-1] & u_{Mm}y_m[M] & \dots & u_{Mm}y_m[N-1] \end{bmatrix} \tag{21}$$

As can be seen, each row is a scaled version of $y_m[n]$. Obviously, the resulting matrix does not have the Toeplitz structure of the original trajectory matrix. But by replacing the entries in each diagonal of $\mathbf{A}_m$ by their average, a Toeplitz matrix $\hat{\mathbf{A}}_m$ is obtained again. Interestingly, the diagonal averaging can equally well be formulated as a linear filtering operation according to

$$a_m[n] = \frac{1}{M_d} \sum_{i=l}^{s} u_{im} y_m[n+i-1] \tag{22}$$

where the factors M_d, l and s have values according to the number of elements in the diagonals of the matrix defined in eqn 21. The following situations have to be considered:

- With M elements, eqn. (22) represents the *steady state response* of the filter which corresponds to $(M-1) \leq n \leq (N-M)$ and $M_d = M$, $l = 1$ $s = M$.
- With less than M elements, eqn. (22) represents the transitory response of the filter and,
  - if $0 \leq n \leq (M-2)$, corresponding to the lower left corner of the matrix, then $M_d = n+1$, $l = M - M_d + 1$ and $s = M$.
  - if $(N-M+1) \leq n \leq (N-1)$, corresponding to the upper right corner of the matrix, then $M_d = N - n$, $l = 1$ and $s = M_d$.

Whatever is the case, the filter is an *anti-causal* FIR filter, and we can consider that $y_m[n] = 0$ for the time indexes $0 \leq n \leq (M-2)$ and $n \geq N$ and always compute eqn. (22) as in the steady-state case. Therefore, the transfer function of the synthesis step reads

$$F_m(z) = \frac{1}{M}(u_{1m} + u_{2m}z^1 + \ldots + u_{Mm}z^{(M-1)}) \tag{23}$$

Notice that the analysis and synthesis transfer functions differ by a scale factor $(1/M)$ and by the sign of the exponent of z. Therefore, the magnitudes of the frequency responses of both filters are related by this scale factor $(1/M)$, and their phases are symmetric.

The transfer function of the global system is a cascaded formed of the analysis filter (projection step) and the synthesis filter (reconstruction step with diagonal averaging) according to

$$T_m(z) = \frac{A_m(z)}{X(z)} = F_m(z)H_m(z) = \sum_{k=-(M-1)}^{M-1} t_{km} z^k \tag{24}$$

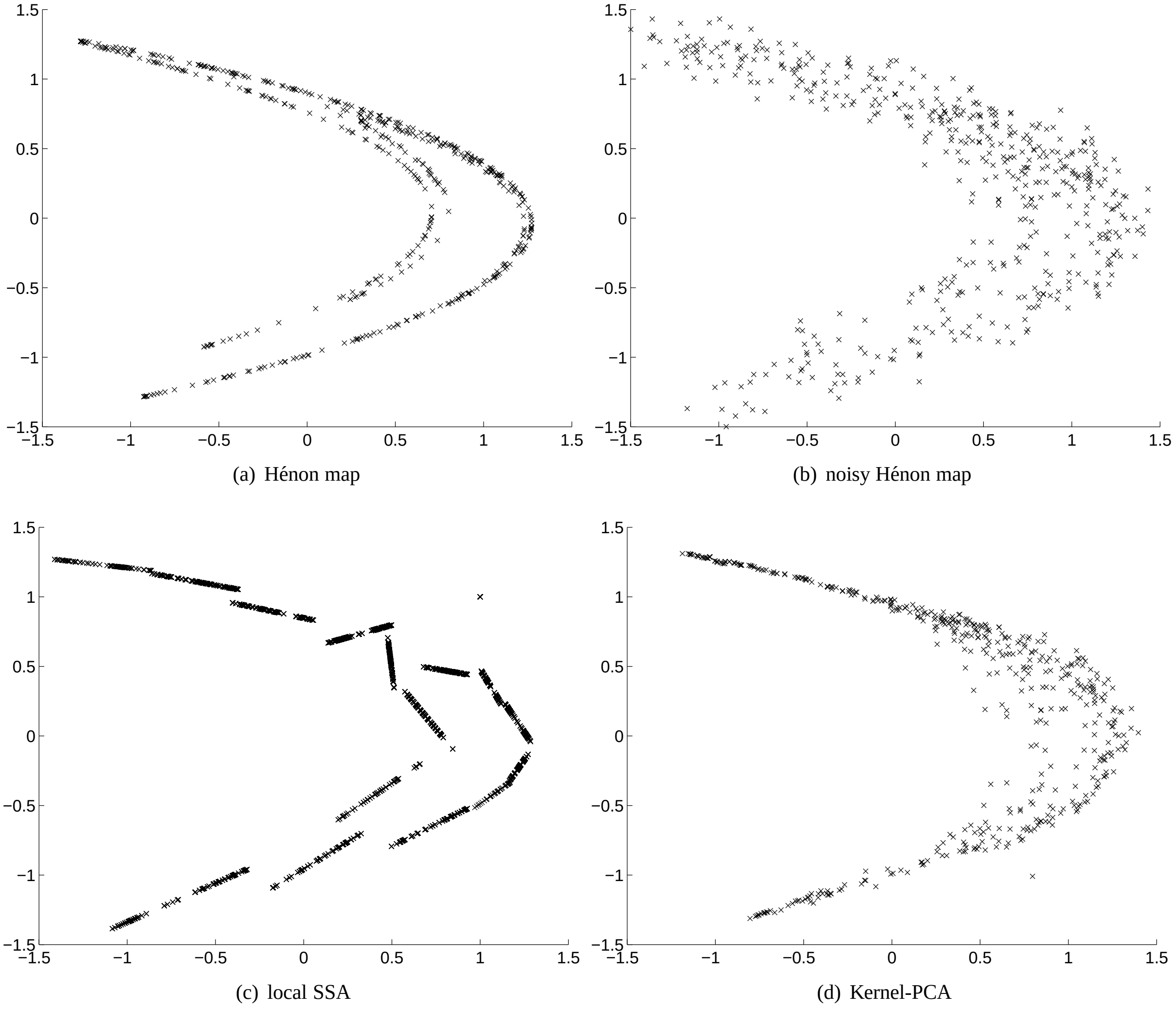
The coefficients t_{km} result from the product of two polynomials with identical coefficients but symmetric exponents. It can be shown easily that $t_{km} = t_{-km}$, $k = 1, \ldots, (M-1)$ [15]. Therefore, the frequency response $T_m(\omega) \equiv T_m(e^{j\omega})$ of the global filter has the following expression

$$T_m(e^{j\omega}) = t_{0m} + \sum_{k=1}^{M-1} 2t_{km} \cos(k\omega) \tag{25}$$

where $j = \sqrt{-1}$ represents the imaginary unit. The frequency response $T_m(\omega)$ is a real function with a period equal to $T = 2\pi = \omega$ yielding a normalized sampling rate $\omega = 2\pi$. Hence the global filter corresponds to a zero-phase filter meaning that each extracted component $a_m[n]$ is always in-phase with its related original signal $x[n]$. It has to be noticed that the global system maintains this characteristic even when the trajectory matrix is organized as an Hankel matrix, but the characteristics of the analysis and synthesis filters then interchange. Furthermore, the sequences $y_m[n], m = 1 \ldots M$, corresponding to the outputs of $\mathbf{u}_m, m = 1 \ldots M$ and related inputs $x[n]$, are *orthogonal* sequences which represent filtered versions of the input. Furthermore, the amplitudes of these components can be scaled using the coefficients p_{mm} as discussed above.

4. Local SSA

The reconstruction step in the multi-dimensional space aligns the data with the directions of the eigenvectors as can be deduced from eqn. 21. Each column of $\mathbf{A}_m$ has the eigenvector multiplied by a scalar corresponding to the projection of the data onto the eigenvector. However, signals can have trajectories in the multi-dimensional space that cannot be represented by this model. Fig. **1**-a) shows the phase space trajectories generated by a dynamical model known as the Hénon map. Fig. **1**-b) shows the corresponding data set with added Gaussian noise. By using one eigenvector to denoise the signal, its trajectory cannot be recovered. However, denoising and the recovery of the trajectory can



(a) Hénon map

(b) noisy Hénon map

(c) local SSA

(d) Kernel-PCA

Figure 1: A nonlinear time series resulting from the following dynamics $x[n + 1] = 1 - ax^2[n] + bx[n - 1]$ where $a = 1.4, b = 0.3$. The signal was embedded in time-delayed coordinates $M = 2$ (Henon map)

be achieved locally using *Local SSA* which will be discussed next.

Local SSA [16] introduces a clustering step in the SSA algorithm in order to have eigenvectors computed for each cluster. The main steps of the algorithm are:

- Clustering the columns of the trajectory matrix into q clusters [17]. The algorithm *k-means*[17] can be applied to achieve the clustering.
- Applying SSA to data within each cluster and then reconstructing the data using the local subspace model. The MDL criterion in conjunction with a couple of empirical rules has been used to find the dimension L of the subspace model in each cluster[16].
- Reverting the clustering, i.e., forming a reconstructed trajectory matrix by ordering the data points according to their position in the original trajectory matrix.
- Forming the time series using diagonal averaging.

The application of SSA to data in each of $q = 15$ clusters and using only the eigenvector to form the model of each cluster results in the ”denoised” trajectories shown in Fig. **1**-c). It can be seen that the local approximations to the underlying dynamics reflect the general trend of the data very well. So by discovering directions of high variance only locally, i.e. within clusters of points taken from the trajectory matrix of the signal in $2{-}D$ space, can be provide an acceptable solution. But it is also obvious that the mapping is not always smooth. This results from the structure of the local clusters which possess principal directions deviating from the underlying dynamics due to strong fluctuations in the noise. Also it can be seen that the fine structure of the Hénon map cannot be captured where the spacing of different segments of the trajectory are too closely spaced compared to the spread of the noise. This illustrates that in such cases the local linear approximations need to be replaced by generically non-linear data analysis methods. One such method will be discussed next.

5. Kernel subspace models

The subspace model is now described in a feature space created by a non-linear mapping of the original data. Comparing eqns. 3 and 10, the matrix of coefficients **A** be computed using the eigendecomposition of the kernel matrix $\mathbf{K} = \mathbf{\Phi}^T\mathbf{\Phi}$ according to

$$\mathbf{U} = \mathbf{\Phi}\mathbf{V}\mathbf{D}^{-1/2} \tag{26}$$

The dimension of the subspace model **U** is $\Im \times R$, where $\Im$ is the dimension of the feature space, determined by the non-linear mapping, and the number of columns is $(R \leq K)$. As discussed above, the subspace model, i.e. the non-linear mapping of the data, is not computed explicitly. Rather, all data manipulations should be achieved by using dot products. For instance, the projections of a point **x** in feature space read

$$\mathbf{y} = \mathbf{D}^{-1/2}\mathbf{V}^T\mathbf{\Phi}^T\phi(\mathbf{x}) \tag{27}$$

where $\mathbf{\Phi}^T\phi(\mathbf{x})$ denotes a vector of dot products between $\phi(\mathbf{x})$ and each element j of the training set. The points are not explicitly mapped and dot products are evaluated by replacing them by suitable kernel functions. For instance, using an RBF kernel, the dot product between one pair of points is given by

$$\phi^T(\mathbf{x}_j)\phi(\mathbf{x}) = k(\mathbf{x}_j, \mathbf{x}) = \exp\left(-\frac{\left\|\mathbf{x}_j - \mathbf{x}\right\|^2}{2\sigma^2}\right) \tag{28}$$

where σ is a value to be assigned according to the range of values of the data set. The entries k_{ij} of the kernel matrix **K** replace dot products between pairs of elements of the training set, and the eigendecomposition of the kernel matrix is employed to build the subspace model (eqn. 26). Therefore the number of columns of the subspace model **U** is also determined by the number of non-zero eigenvalues. With the RBF kernel, the sum of eigenvalues is always K. However, its decay is controlled by σ: if it is too small, then $\mathbf{K} \simeq \mathbf{I}$ and if it is too large, the matrix has all entries close to 1. The first case leads to K eigenvalues of value 1, the second case leaves only one non-zero eigenvalue. Therefore note that unlike linear PCA, kernel-PCA allows to extract a number of principal components which exceeds the dimensionality of the input data. Notice that having $K \geq M$ examples of data with dimension M, working in input space, the maximum number of nonzero eigenvalues will also be M as can be seen by either computing the covariance/correlation matrix or the matrix of dot products. In kernel-PCA, instead, the kernel matrix in feature space will have size $K \times K$, and the number of nonzero eigenvalues can often be higher than M. The results in Fig. **1**-d) were obtained with an RBF kernel with a width parameter $\sigma = 1$ and 4 principal components in feature space.

In practice, the projections **y** often might represent the final goal of applying KPCA. But having to come back to the domain of the original data, the related pre-image problem must also be considered. The pre-images of the data, which were reconstructed in feature space, have been achieved by using one of the techniques to solve eqn. 6 which will be presented latter. Fig. **1**-d) shows that the resulting trajectories are much smoother than in case of local SSA, but they also are much more noisy still.

In large training data sets, the corresponding kernel matrix **K** becomes prohibitively large. Consequently, its eigendecomposition is often not feasible and impractical to achieve in real data applications. Therefore, the drawbacks of this methods are:

- The size of **K**: The eigendecomposition of large matrices can be unfeasible in practical applications requiring the manipulation of large data sets.
- The dual form of the model (eqn. 26): This form in kernel methods necessitates that the complete training set **X** must be available to compute the projections of any new point $\phi(\mathbf{x})$ into the model subspace.

Considering that the training set is projected into the model subspace, the projections read

$$\mathbf{Y} = \mathbf{D}^{1/2}\mathbf{V}^T \tag{29}$$

However, in most of the cases, the projections onto the eigenvectors related to the most significant eigenvalues are required only. Assigning the diagonal entries of the selection matrix **P** such that the R largest eigenvalues are selected, a low rank approximation of the kernel matrix is then achieved by $\mathbf{K} \simeq \mathbf{V}\mathbf{D}^{1/2}\mathbf{P}\mathbf{D}^{1/2}\mathbf{V}^T$. Several works suggested this principle as strategy to reduce the complexity of kernel subspace models.

5.1. Reducing complexity of kernel subspace methods

In [18], a low-rank approximation is proposed based on an incomplete Cholesky decomposition of the kernel matrix $\mathbf{K} \simeq \mathbf{C}^T\mathbf{C}$. The related pivoting scheme of the decomposition determines the division of the data into two blocks according to $\mathbf{\Phi} = [\ \mathbf{\Phi}_r \quad \mathbf{\Phi}_s\ ]$.

It can be proven that the incomplete Cholesky decomposition then can be written as

$$\mathbf{C} = \begin{bmatrix} \mathbf{L} & \mathbf{L}^{-T}\mathbf{K}_{rs} \end{bmatrix} \tag{30}$$

where the matrix $\mathbf{L}$ represents a triangular matrix corresponding to $\mathbf{K}_r = \mathbf{\Phi}_r^T \mathbf{\Phi}_r = \mathbf{L}^T \mathbf{L}$ and $\mathbf{K}_{rs} = \mathbf{\Phi}_r^T \mathbf{\Phi}_s$. The eigendecomposition of the following $R \times R$ matrix

$$\mathbf{Q} = \mathbf{C}\mathbf{C}^T = \mathbf{V}_q \mathbf{D} \mathbf{V}_q^T \tag{31}$$

allows to compute uncorrelated projections of the training data set in the feature space as follows

$$\mathbf{Y} = \mathbf{V}_q^T \mathbf{C} = \mathbf{V}_q^T \mathbf{L}^{-T} \mathbf{\Phi}_r^T \begin{bmatrix} \mathbf{\Phi}_r & \mathbf{\Phi}_s \end{bmatrix} \tag{32}$$

Consequently, the basis vector matrix of the subspace model can be expressed as

$$\mathbf{U} = \mathbf{\Phi}_r \mathbf{L}^{-1} \mathbf{V}_q \tag{33}$$

This dual form of the basis vector matrix is thus obtained by using only a subset of the training data set, thereby reducing the storage and computational requirements substantially. Note that the R vectors form an *orthogonal basis* in feature space, i.e. $\mathbf{U}^T \mathbf{U} = \mathbf{I}$. Several algorithms [19], [20], [21] lead to similar solutions by exploiting the idea that there are samples in the training set that can be expressed as a linear combination of others. As described in [22], [23], [24], it is possible to obtain an incomplete Cholesky decomposition of the kernel matrix having as input the complete training set but not having to compute explicitly the kernel matrix.

5.2. Pre-image problem

In some application, the reconstruction of the point in the feature space needs to be combined with the estimate of the pre-image in the input space. Combining these two steps, the kernel trick can be used and the reconstruction never needs to be computed explicitly [4], thus avoiding to work in feature space. As not every point in feature space may have a corresponding pre-image in input space [25], two main strategies can be found to solve eqn. 6.

- If an RBF kernel is considered, the iterative procedure is described by the following equation [26]

  $$\mathbf{p}_{t+1} = \frac{\mathbf{X}_B(\mathbf{g} \diamond \mathbf{k}_{p_t})}{\mathbf{g}^T \mathbf{k}_{\mathbf{p_t}}} \tag{34}$$

  where $\diamond$ represents the Hadamard product. Further, $\mathbf{g} = \mathbf{A}\mathbf{y}$, where $\mathbf{A}$ is the matrix that linearly combines the training set (or subset)(see eqn. 4) and $\mathbf{y}_j$ are the feature space components of the point $\mathbf{x}$. The components of the vector $\mathbf{k}_{\mathbf{p_t}} = \mathbf{k}(\mathbf{X}_B, \mathbf{p}_t)$ are given by the dot products between $\phi(\mathbf{p}_t)$ and the images $\mathbf{\Phi}_B$ of the training set (or subset) $\mathbf{X}_B$.

- In [4], the pre-image of a point of the feature space is based on the fact that it is possible to compute the coordinates of a new point if we know its distance to a set of known points [27], resulting in an algebraic approach to the problem.

These two methods have been addressed in [28] and in case of an RBF kernel an hybrid solution has been developed based on an iterative procedure described by eqn. 34. Thereby an initialization resulting from the algebraic solution [4] has been employed.

6. Centering the Data

All previous deductions were conducted assuming that the data is centered. In input space, centering can be considered a pre-processing step which must be accomplished before computing the scatter or kernel matrix, and before projecting any new data vector. But in kernel methods, this step has to be integrated into the projection step. To facilitate the exposition of the centering in feature space, we will consider a vector $\mathbf{m}$ with K elements, all of which equal to $1/K$, and a matrix $\mathbf{M}$ filled with K column vectors $\mathbf{m}$. Then to project a new data point $\phi(\mathbf{a})$ into the feature subspace and to center the training data set, the following transformation of the mapped data must be considered

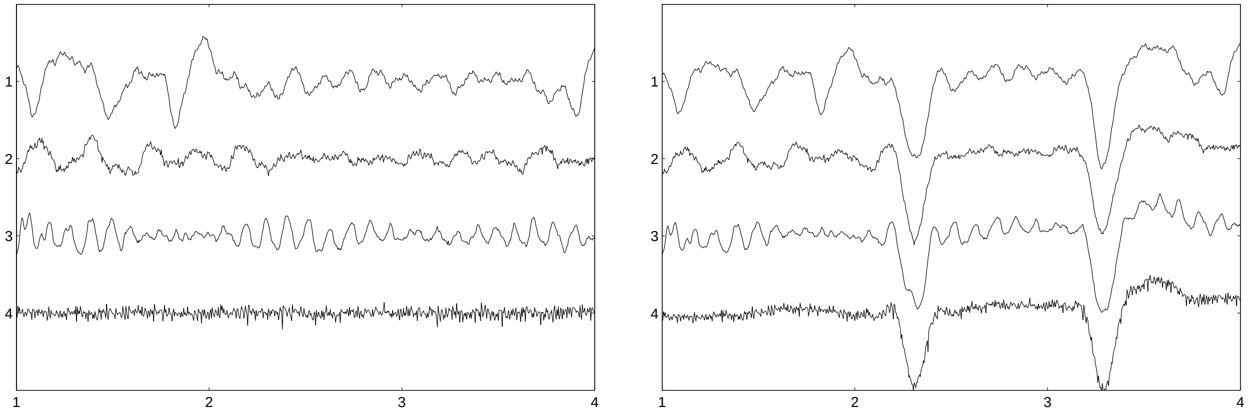
$$(\mathbf{\Phi} - \mathbf{\Phi}\mathbf{M})^T (\phi(\mathbf{a}) - \mathbf{\Phi}\mathbf{m}) \tag{35}$$

To compute the subspace model, these steps can also be applied after the kernel matrix $\mathbf{K}$ is computed, or even after the incomplete Cholesky decomposition of the kernel matrix is performed. For instance, by applying the centering step to matrix $\mathbf{C}$, a low rank approximation of the centered data can be computed

$$\tilde{\mathbf{K}}_c = (\mathbf{I} - \mathbf{M})\mathbf{C}^T \mathbf{C}(\mathbf{I} - \mathbf{M}) \tag{36}$$

7. Artefact Elimination in Electroencephalogram

Ocular activity represents the major source of artifacts in electroencephalogram (EEG) signals, especially when recorded from frontal channels. Such ocular artifacts are labeled electro-oculograms (EOG). Especially when measured at frontal locations of the scalp close to the eyes, the EOG signal amplitude can be several times larger than the brain generated scalp potentials. Eye movements and blinks are very frequent and inevitably will occur during EEG recordings while objects perform various tasks. To reduce the presence of such disturbing eye movement activity in EEG recordings, the subject is often asked to suppress eye blinking or to fixate the eyes onto a given target. However, this goal is never fully accomplished either because of the nature of the task to be examined or because the subject is not willing or able to cooperate. Several works can be found that address this problem using subspace based models to remove the artifacts from the multi-channel recording. Most of the more recent works use independent component analysis to compute the subspace model: [29] used the *INFOMAX* algorithm, [30] and [31] applied the



(a) Original

(b) Artificial mixtures

Figure 2: Sub-segments EEG signals with different activities *Top to Down*: 1st - Delta (type A), 2nd - Theta (type B), 3rd- Alpha (type C) and 4th-Beta (type D).

joint approximate diagonalization of eigenmatrices algorithm (*JADE*), in [32] an approximate joint diagonalization of time-delayed correlation matrices (*SOBI*) was used while in [33] the fast fixed point algorithm (FASTICA) was applied. In [34] the *Local SSA* was proposed and applied to reduce high-amplitude artifacts like EOG. Since then the application of subspace models to single-channels EEG signals have been case studies for various algorithms [22]. But usually artifactual contributions to the recorded EEG signals are considered ”the signal” and the actual EEG signals as ”sort of a broadband noise”. Consequently the projective subspace techniques referred to above are applied to separate the artifacts from the ”pure” EEG signals. The following sections illustrate the application of the discussed methods either using artificially mixed EEG signals or real data.

7.1. Local SSA and SSA

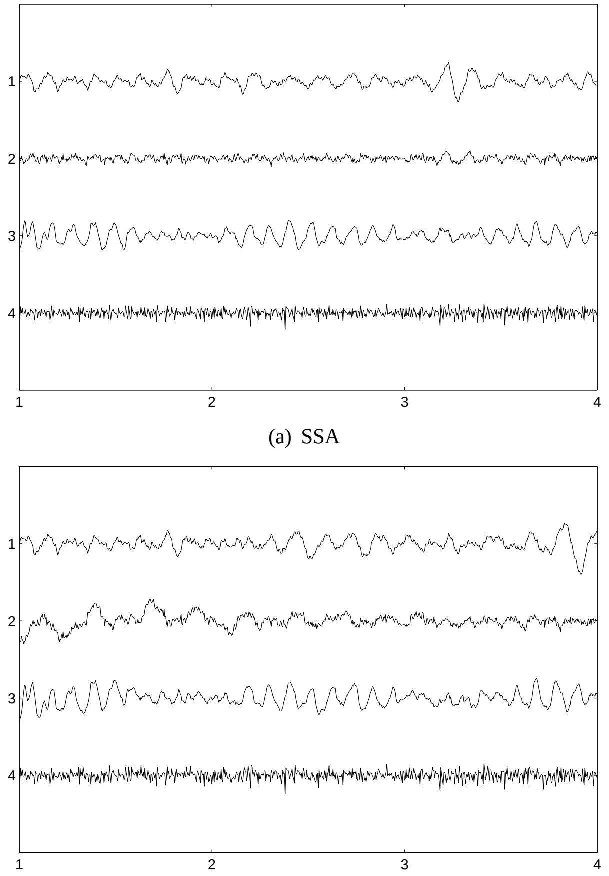
EEG segments of 10*s* were chosen with a predominance of specific signal components in one of the characteristic bands but clean of any artifacts (EOG, EMG or movements of the patient). In addition, EOG artifacts were selected with different amplitudes as well as a number of eye blink artifacts, ranging from 1-7, in a segment of 10s. Fig. **2** shows an example of the different types of EEG segments and the corresponding artificially contaminated signal segments.

The implementation of the algorithm requires the assignment of the following parameters: the embedding dimension M and the number q of clusters. The heuristic rules proposed in the literature to choose the embedding dimension M point towards a minimum value that is related with the period of the signal to be extracted. An heuristic was also proposed to find the number of clusters [34]. Experimentally we conclude that the heuristics work quite consistently if *Local SSA* is applied to all this signals, but not for SSA. In particular, the embedding dimension M can be chosen according to that heuristic ($M = 75$, $250Hz$ is the sampling rate), while with SSA, M has also to be chosen according to the frequency contents of the EEG. In the latter case we show results for varying embedding dimensions M corresponding to the outcome with the highest correlation coefficient between the original and corrected EEG. *Local SSA* has an high correlation coefficient ($cc > 0.8$) for the segments where beta and alpha waves dominate. When the dominant activity is a theta wave, *local SSA* proofs better than the other variant. Finally, when delta waves dominate, both variants have a correlation coefficient of similar magnitude. Fig. **3** illustrates the output of the algorithm for sub-segment signals where differences in performance are clearly visible.

7.2. KPCA Algorithm

The KPCA algorithm is applied in parallel to recordings from 4 different EEG electrodes, all of which contain a high amplitude EOG interference. The segments have a duration of 10s each and Fig. **4** (on the left) shows a subsegment of 3*s*.

The multi-dimensional signal is constructed employing an embedding dimension $M = 11$ and analyzed applying an RBF kernel with width parameter $\sigma = max_i(\|\mathbf{x}_i - \mathbf{x}_{mean}\|), i = 1, \ldots, J$, where $\mathbf{x}_{mean}$ is the mean of the data set. Beforehand, the multi-dimensional data set was divide into training and test sets. Then, an incomplete Cholesky decomposition was performed while fixing the number of elements in the subset ($R = 20$) of the training set. In feature space, the mapped multi-dimensional signal is projected onto L subspace directions chosen according to the eigenspectrum of matrix $\mathbf{Q}$. Fig. **4** corroborates that artifacts were significantly reduced this way in the processed channels.



(a) SSA

(b) Local SSA

Figure 3: Corrected EEG subsegments using SSA and Local SSA algorithms using the artificially mixed signals of Fig. **2**

8. Conclusions

The application of projective subspace models to biomedical time series was discussed. Starting by SSA/SVD methods, the main steps were presented and explained using a linear invariant systems perspective. These methods are clearly linear approaches both in what concerns the multidimensional representation of the data and in what concerns the time series components as they are filtered versions of the original time series. After considering multi-dimensional representations of such univariate data, employing their trajectory matrix and its characteristics, *local SSA* was introduced. Following a similar strategy, *kernel PCA* and its greedy variant was considered to extract non-linear components of the multi-dimensional data. The methods were illustrated on a high-amplitude ocular artifact removal from recorded EEG data.

ACKNOWLEDGMENT

A.R. Teixeira received a PhD Scholarship (SFRH/BD/28404/2006) supported by the Portuguese Foundation for Science and Technology (FCT).

References

[1] A. Cichocki, S.-I. Amari, Adaptive Blind Signal and Image Processing, Wiley, 2002.

[2] L. Parra, P. Sajda, Blind source separation via generalized eigenvalue decomposition, Journal of Machine Learning Research 4 (2003) 1261–1269.

[3] A. M. Tomé, The generalized eigendecomposition approach to the blind source separation problem, Digital Signal Processing 16 (3) (2006) 288–302.
URL `www.sciencedirect.com`

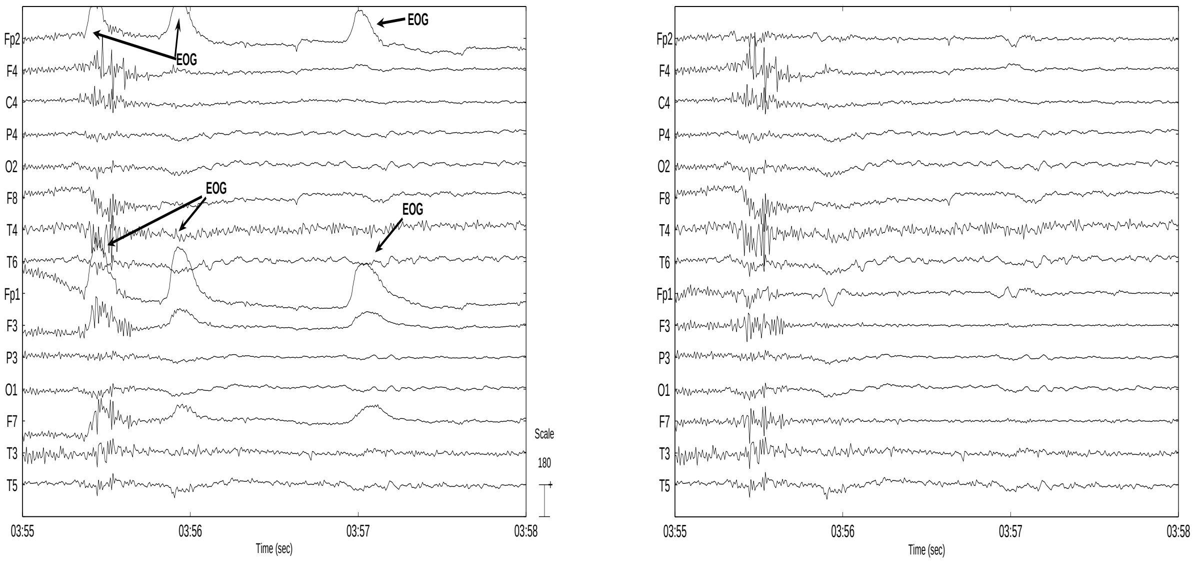
[4] J. T. Kwok, I. W. Tsang, The pre-image problem in kernel methods, IEEE Transactions on Neural Networks 15 (6) (2004) 1517–1525.

[5] K. I. Diamantaras, S. Kung, Principal Component Neural Networks, Theory and Applications, Wiley, 1996.

[6] A. P. Liavas, P. A. Regalia, On the behavior of information theoretic criteria for model order selection, IEEE Transactions on Signal Processing 49 (8) (2001) 1689–1695.

[7] Z. Leonowicz, J. Karvanen, T. Tanaka, J. Rezmer, Model order selection criteria: comparative study and applications, in: International Workshop Computational Problems of Electrical Engineering, 2004.

[8] P. C. Hansen, S. H. Jensen, Subspace-based noise reduction for speech signals via diagonal and triangular matrix decomposi-



(a) Original EEG

(b) Corrected EEG

Figure 4: EEG signals placed according $10 - 20$ system with reference to Cz. Only $3s$ of the $10s$ are plotted. Channels processed: Fp2, Fp1, F3 and F7

tions: Survey and analysis, Eurasip Journal on Advances in Signal Processing Vol 2007.

[9] K. Hermus, P. Wambacq, H. V. hamme, A review of signal subspace speech enhancement and its application to noise robust speech recognition, Eurasip Journal on Advances in Signal Processing.

[10] N. Golyandina, V. Nekrutkin, A. Zhigljavsky, Analysis of Time Series Structure: SSA and Related Techniques, Chapman & HALL/CRC, 2001.

[11] M. Ghil, M. Allen, M. D. Dettinger, K. Ide, e. al, Advanced spectral methods for climatic time series, Reviews of Geophysics 40 (1) (2002) 3.1–3.41.

[12] Y. Ephraim, H. L. V. Trees, A signal subspace approach for speech enhancement, IEEE Transactions on Acoustic, Speech and Signal Processing, 3 (4) (1995) 251–266.

[13] A. R. Teixeira, A. M. Tomé, M. Böhm, C. G. Puntonet, E. W. Lang, How to apply non-linear subspace techniques to univariate biomedical time series, IEEE Transactions on Instrumentation & Measurement 58 (8) (2009) 2433–2443.

[14] L. B. Jackson, Signals, Systems and Transforms, Addison-Wesley, 1991.

[15] L. B. Jackson, Digital Filters and Signal Processing, Kluwer Academic Publishers, 1996.

[16] A. R. Teixeira, A.M.Tomé, E.W.Lang, R. Schachtner, K.Stadlthanner, On the use of KPCA to extract artifacts in one-dimensional biomedical signals, in: S. McLoone, J. Larsen, M. V. Hulle, A. Rogers, S. C. Douglas (Eds.), Machine Learning for Signal Porcessing, MLSP 2006, IEEE, Dublin, 2006, pp. 385–390.

[17] R. Duda, P. Hart, D. G. Stork, Pattern Classification, John Wiley & Sons, 2001.

[18] A. R. Teixeira, A. M. Tomé, E. W. Lang, Feature extraction using low-rank approximations of the kernel matrix, in: A. Campilho, M. Kamel (Eds.), LNCS 5112- ICIAR 2008, Porto, 2008, pp. 404–412.

[19] V. Franc, V. Hlaváč, Greedy algorithm for a training set reduction in the kernel methods, in: 10th International Conference on Computer Analysis of Images and Patterns, Springer, Groningen, Holland, 2003, pp. 426–433.

[20] G. C. Cawley, N. L. C. Talbot, Efficient formation of a basis in a kernel induced feature space, in: M. Verleysen (Ed.), European Symposium on Artificial Neural Networks, d-side, Bruges, Belgium, 2002, pp. 1–6.

[21] G. Baudat, F. Anouar, Feature vector selection and projection using kernels, Neurocomputing 55 (2003) 21–38.

[22] A. R. Teixeira, A. M. Tomé, E. W. Lang, Feature extraction using linear and non-linear subspace techniques, in: C. Alippi, M. Polycarpou (Eds.), Artificial Neural Networks- ICANN 2009, Vol. II, Springer-Verlag, Cyprus, 2009, pp. 115–124.

[23] F. R. Bach, M. I. Jordan, Kernel independent component analysis, Journal of Machine Learning Research 3 (2002) 1–48.

[24] S. Fine, K. Scheinberg, Efficient SVM training using low-rank kernel representations, Journal of Machine Learning Research 2 (2001) 243–264.

[25] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, B. Schölkopf, An introduction to kernel-based algorithms, IEEE Transactions on Neural Networks 12 (2) (2001) 181–202.

[26] B. Schölkopf, S. Mika, C. J. Barges, P. Knirsch, K.-R. Müller, G. Ratsch, A. J.Smola, Input space versus feature space in kernel-based methods, IEEE Transactions on Neural Networks 10 (5) (1999) 1000–1016.

[27] J. C. Gower, Adding a point to vector diagram in multivariate analysis, Biometrika 55 (1968) 582–585.

[28] A. R. Teixeira, A. M. Tomé, K. Stadlthanner, E. W. Lang, KPCA denoising and the pre-image problem revisited, Digital Signal Processing 18 (2008) 568–590.

[29] T.-P. Jung, S. Makeig, M. Westerfield, J. Townsend, E. Courchesne, T. J. Sejnowski, Removal of eye activity artifacts from visual event-related potentials in normal and clinical subjects, Clinical Neurophysiology 111 (2000) 1745–1758.

[30] E. Urrestarazu, J. Iriarte, M. Alegre, M. Valencia, C. Viteri, J. Artieda, Independent component analysis removing artifacts

in ictal recordings, Epilepsia 45 (9) (2004) 1071–1078.

[31] W. Zhou, J. Gotman, Removing eye-movement artifacts from the EEG during the intracarotid amobarbital procedure, Epilepsia 46 (3) (2005) 409–414.

[32] C. A. Joyce, I. F. Gorodniysky, M. Kutas, Automatic removal of eye movement and blink artifacts from EEG data using blind component separation, Psychophysiology 41 (2004) 313–325.

[33] R. N. Vigário, Extraction of ocular artefacts from EEG using independent component analysis, Electroencephalography and Clinical Neurophysiology 103 (1997) 395–404.

[34] A. R. Teixeira, A. M. Tomé, E. Lang, P. Gruber, A. M. d. Silva, Automatic removal of high-amplitude artifacts from single-channnel electroencephalograms, Computer Methods and Programs in Biomedicine 83 (2) (2006) 125–138.