ON THE USE OF KPCA TO EXTRACT ARTIFACTS IN ONE-DIMENSIONAL BIOMEDICAL SIGNALS

A. R.Teixeira, A. M. Tomé *

DETI/IEETA
Universidade de Aveiro
3810-193 Aveiro, Portugal
(ana@ieeta.pt)

E. W. Lang, R. Schachtner, K. Stadlthanner †

Institute of Biophysics
University of Regensburg,
D-93040 Regensburg, Germany
(elmar.lang@biologie.uni-regensburg.de)

ABSTRACT

Kernel principal component analysis(KPCA) is a nonlinear projective technique that can be applied to decompose *multi-dimensional* signals and extract informative features as well as reduce any noise contributions. In this work we extend KPCA to extract and remove artifact-related contributions as well as noise from *one-dimensional* signal recordings. We introduce an embedding step which transforms the one-dimensional signal into a multi-dimensional vector. The latter is decomposed in feature space to extract artifact related contaminations. We further address the pre-image problem and propose an initialization procedure to the fixed-point algorithm which renders it more efficient. Finally we apply KPCA to extract dominant Electrooculogram (EOG) artifacts contaminating Electroencephalogram (EEG) recordings in a frontal channel.

1. INTRODUCTION

Kernel principal component analysis (KPCA) represents a nonlinear projective subspace technique which can be used to project *multi-dimensional* signals into mutually orthogonal subspaces in a meaningful way like it is done in subspace denoising applications [1]. But many signal processing applications are based on *one-dimensional* signals. Clearly projective subspace techniques are not available for such one dimensional signals. However time series analysis techniques rely on embedding one dimensional sensor signal into a space spanned by their time-delayed coordinates [2]. A similar transformation is also used in singular spectrum analysis (SSA) [3].

In this work we propose a similar strategy to remove large amplitude, artifact related contributions to recorded one-dimensional signals like single channel EEG recordings. The idea is to consider the large amplitude artifact as the "signal" and the remaining signal as "sort of a broadband noise" in signal recordings where artifact related contributions clearly dominate. With EEG signals, eye movement artifacts typically dominate in frontal channel EEG recordings just to mention an example that will be considered in this application. The methodology we present adapts KPCA to feature extraction and denoising applications of *one-dimensional* signals. It follows a matrix manipulation approach with special emphasis to signal reconstruction and pre-image estimation. Two methods of estimating pre-images, discussed in the literature, are considered and a particularly suitable way of finding the starting point of the fixed-point iterative method is suggested. Some toy data illustrate the application of KPCA to suppress broadband noise (white noise) contributions to signals. In particular, the influence of the pre-image computation on the performance of the method is considered. We also apply our method to extract prominent artifacts like EOG in EEG recordings.

2. KPCA FOR ONE-DIMENSIONAL SIGNAL

In practice, the goal of projective techniques is to represent the data with reduced dimensionality by extracting meaningful components while retaining the structure of the raw data. The application of KPCA to one-dimensional time series starts by embedding the signal in its time-delayed coordinates, obtaining that way a multidimensional signal. The transformation of multidimensional signals to a feature space is never explicitly computed as long as all manipulations can be based on dot products. This constitutes the so-called kernel trick. The projective techniques are applied in the feature space, but the reconstruction is computed in input space only. This is accomplished by joining the reconstruction in feature space and the pre-image calculation in an unique step. The one-dimensional signal is obtained finally by diagonal averaging of the data matrix entries which correspond to the denoised multidimensional signal.

*The authors are grateful to financial support by CRUP

†The authors gratefully acknowledge support by the DAAD and the DFG

2.1. Embedding

Embedding can be regarded as a mapping that transforms a one-dimensional time series $x = (x[0], x[1], ..., x[N-1])$ into a multidimensional sequence of $K = N-M+1$ lagged vectors

$$\mathbf{x}_k = [x[k-1+M-1], \ldots, x[k-1]]^T,\ k = 1, \ldots, K \quad (1)$$

with $M < N$ being the corresponding window length or the embedding dimension. The lagged vectors then constitute the columns of the trajectory matrix $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_K]$ which represents a Toeplitz matrix. The further processing of this data matrix $\mathbf{X}$ is performed by KPCA considering each column a point in a vector space of dimension M (input space).

2.2. Kernel-PCA

In kernel-PCA (KPCA) a multidimensional signal $\mathbf{x}_k,\ k = 1...K$, is considered to be mapped through a non-linear function $\phi(\mathbf{x}_k)$ into a feature space yielding the mapped data set $\mathbf{\Phi} = [\phi(\mathbf{x}_1)\phi(\mathbf{x}_2)...\phi(\mathbf{x}_K)]$. In feature space then a linear PCA is performed estimating the eigenvectors and eigenvalues of a matrix of *outer* products, called a *covariance matrix* which for zero mean data reads $K\mathbf{C} = \mathbf{\Phi}\mathbf{\Phi}^T$. It can be shown that these eigenvectors and eigenvalues are related to those of a matrix of *inner* products, called a *kernel matrix* $\mathbf{K} = \mathbf{\Phi}^T\mathbf{\Phi}$. Using the kernel trick [4], this kernel matrix can be computed without explicitly mapping the data. The latter are considered centered in feature space according to

$$\mathbf{\Phi}_c = \mathbf{\Phi}\left(\mathbf{I} - \frac{1}{K}\mathbf{j}_K\mathbf{j}_K^T\right) \quad (2)$$

where $\mathbf{j}_K = [1, 1, \ldots, 1]^T$ is a vector with dimension $K \times 1$, and $\mathbf{I}$ is a $K \times K$ identity matrix. The centered kernel matrix $\mathbf{K}_c$ can be computed as $\mathbf{K}_c = \mathbf{\Phi}_c^T\mathbf{\Phi}_c$.

Notice that each element $k(i,j) \equiv k(\mathbf{x}_i, \mathbf{x}_j)$ of the kernel matrices depends on the inner product $\phi^T(\mathbf{x}_i)\phi(\mathbf{x}_j)$ which can be computed using only the data $\mathbf{x}_k$ in input space. For instance if an RBF kernel is used, $k(i,j)$ is calculated according to

$$k(i,j) \equiv k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) \quad (3)$$

where σ^2 is a free parameter that can be chosen according to some data spread criterium.

Because the eigenvalues of the non-normalized covariance matrix coincide with the eigenvalues of the kernel matrix, their eigenvectors are related by

$$\mathbf{U} = \mathbf{\Phi}_c\mathbf{V}\mathbf{D}^{-1/2} \quad (4)$$

where $\mathbf{U}$ is the matrix with L eigenvectors of the covariance matrix, $\mathbf{V}$ is the matrix with L eigenvectors of kernel matrix and $\mathbf{D}$ is a diagonal matrix with the corresponding $L \leq K$ largest eigenvalues of both matrices. Once a data point of the input space $\mathbf{y}_j$ is mapped to the feature space to obtain its image $\phi(\mathbf{y}_j)$, the latter can be projected onto the L eigenvectors spanning the feature space to obtain

$$\mathbf{z}_j = \mathbf{D}^{-1/2}\mathbf{V}^T\left(\mathbf{I} - \frac{1}{K}\mathbf{j}_K\mathbf{j}_K^T\right)\mathbf{k}_{y_j} \quad (5)$$

where

$$\mathbf{k}_{y_j} = [k(\mathbf{x}_1, \mathbf{y}_j), k(\mathbf{x}_2, \mathbf{y}_j), \ldots, k(\mathbf{x}_K, \mathbf{y}_j)]^T \quad (6)$$

represents the vector of inner products between the training data $\mathbf{\Phi}$ and $\phi(\mathbf{y}_j)$, naturally computed using the kernel trick. There are many applications like classification where the projections contain necessary and sufficient information to characterize the problem. However, in feature extraction and denoising applications it is needed to reconstruct any point in feature space using the L eigenvectors and then estimate the position of the corresponding point in the input space, i.e. compute its pre-image [4, 5].

2.3. Reconstruction and Pre-Image

Considering the reconstructed point

$$\hat{\phi}(\mathbf{y}_j) = \mathbf{U}\mathbf{z}_j = \mathbf{\Phi}\left(\mathbf{I} - \frac{1}{K}\mathbf{j}_K\mathbf{j}_K^T\right)\mathbf{V}\mathbf{D}^{-1/2}\mathbf{z}_j = \mathbf{\Phi}\mathbf{g} \quad (7)$$

where the entries of $\mathbf{z}_j$ are the projections onto the L eigenvectors of the covariance matrix, i.e. the columns of $\mathbf{U}$. In order to avoid to work with the possibly high-dimensional feature set $\mathbf{\Phi}$, the methods described in literature combine the reconstruction and the pre-image computations in one step. This goal is achieved by using the square Euclidian distance definition which can be written using dot products then substituted by kernel evaluations. Considering next an arbitrary point in input space $\mathbf{p}$. Its distance in feature space to the reconstructed point $\hat{\phi}(\mathbf{y}_j)$ is defined by

$$\tilde{d}^{(2)} = \|\phi(\mathbf{p}) - \hat{\phi}(\mathbf{y}_j)\|^2 = k(\mathbf{p},\mathbf{p}) - 2\mathbf{g}^T\mathbf{k}_p + \mathbf{g}^T\mathbf{K}\mathbf{g} \quad (8)$$

where the reconstructed point $\hat{\phi}(\mathbf{y}_j)$ is obtained by eq. 7. The entries of the vector $\mathbf{k}_p$ form dot products of $\phi(\mathbf{p})$ with training set images $\mathbf{\Phi}$ which can be computed as described by equation 6. The two methods use the euclidian distance definition within different strategies, and consequently the input space point $\mathbf{p}$ has to be chosen accordingly.

2.3.1. Distance method

Recent work [5] to find the pre-image of a feature space point 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. Hence, the distances of the reconstructed point, $\hat{\phi}(\mathbf{y}_j)$, to every point in the training set $\mathbf{\Phi}$ are computed. So in equation (8), the point $\mathbf{p} = \mathbf{x}_k$ is chosen as an element of the training set. Then computing the distances to all points of the training set, $k = 1, \ldots, K$ the following vector is obtained

$$\tilde{\mathbf{d}}^{(2)} = diag(\mathbf{K}) - 2\mathbf{g}^T\mathbf{K} + \mathbf{g}^T\mathbf{K}\mathbf{g} \tag{9}$$

With certain kernels it is possible to evaluate the distance in input space knowing the corresponding distances in feature space. If an RBF kernel is considered, there is a simple relation between input space distances $\mathbf{d}^{(2)}$ and corresponding feature space distances. Once a distance in feature space can be computed as

$$\tilde{\mathbf{d}}^{(2)} = 2\left(\mathbf{j}_K - \exp\left(-\frac{\mathbf{d}^{(2)}}{2\sigma^2}\right)\right) \tag{10}$$

the corresponding vector of distances in input space can be obtained from

$$\mathbf{d}^{(2)} = -2\sigma^2 \ln(\mathbf{j}_K - 0.5\tilde{\mathbf{d}}^{(2)}) \tag{11}$$

Considering a subset of neighbors of the reconstructed point $\hat{\phi}(\mathbf{y}_j)$, i.e. selecting from the training set the S points with smallest distance $\tilde{\mathbf{d}}^{(2)}$, the corresponding points in input space are designated $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_S]$. The coordinate system for the subset may be chosen from the columns of the eigenvector matrix $\mathbf{E}$ of the data covariance matrix. After centering the set of neighbors

$$\mathbf{Q}_c = \mathbf{Q}\left(\mathbf{I} - \frac{1}{S}\mathbf{j}_S^T\mathbf{j}_S\right) \tag{12}$$

the columns of the matrix $\mathbf{W} = \mathbf{E}^T\mathbf{Q}_c$ represent the coordinates of the points $\mathbf{Q}_c$. Their distance to the origin is obtained as $\mathbf{d}_0^{(2)} = [\| \mathbf{w}_1 \|^2, \| \mathbf{w}_2 \|^2, \ldots, \| \mathbf{w}_S \|^2]$. Then the coordinates $\tilde{\mathbf{p}}$ of the point are given by

$$\mathbf{W}^T\tilde{\mathbf{p}} = -\frac{1}{2}(\mathbf{d}^{(2)} - \mathbf{d}_0^{(2)}) \tag{13}$$

The pre-image $\mathbf{p}$ of $\hat{\phi}(\mathbf{y}_j)$ is finally obtained as

$$\mathbf{p} = \mathbf{E}\tilde{\mathbf{p}} + \frac{1}{S}\mathbf{Q}\mathbf{j}_S = \mathbf{E}\tilde{\mathbf{p}} + \mathbf{p}_0 \tag{14}$$

where $\mathbf{p}_0$ is the mean of the selected neighbors. This method is usually applied considering that the number S of neighbors is less than the dimension M of the points [5]. In that case there are $M - S$ null coordinates in the point $\tilde{\mathbf{p}}$, hence the covariance matrix of the set of points $\mathbf{Q}$ can have at most S non-zero eigenvalues. Noted that if $\mathbf{W}$ represents a rectangular matrix, the SVD solution of eqn. 13 is the minimum norm one. Then the second term of equation (14), the mean of the set of neighbors, might constitute the dominant term in the solution of the pre-image of $\hat{\phi}(\mathbf{y}_j)$.

2.3.2. Fixed-point method

The central idea of this method consists in computing the unknown $\mathbf{p}$ that minimizes the euclidian distance in feature space by setting to zero the gradient of eqn. 8

$$\frac{\partial \tilde{d}^{(2)}}{\partial \mathbf{p}} = \frac{\partial k(\mathbf{p},\mathbf{p})}{\partial \mathbf{p}} - 2\frac{\partial \mathbf{g}^T\mathbf{k}_p}{\partial \mathbf{p}} \tag{15}$$

Substituting the RBF kernel, the first term of the previous equation is zero as $k(\mathbf{p},\mathbf{p}) = 1$. Hence the zeros of the gradient are obtained by

$$\sum_{i=1}^{K} g_i(\mathbf{x}_i - \mathbf{p}) \exp\left(\frac{\|\mathbf{x}_i - \mathbf{p}\|^2}{\sigma^2}\right) = \mathbf{X}(\mathbf{g} \Diamond \mathbf{k}_p) - \mathbf{p}\mathbf{g}^T\mathbf{k}_p = 0 \tag{16}$$

where $\Diamond$ represents the Hadamard product. The zeroes can be computed iteratively by the fixed point algorithm

$$\mathbf{p}_{t+1} = \frac{\mathbf{X}(\mathbf{g} \Diamond \mathbf{k}_{p_t})}{\mathbf{g}^T\mathbf{k}_{\mathbf{p}_t}} \tag{17}$$

The starting point $\mathbf{p}_0$ is chosen randomly and the iterative procedure stops when $|\mathbf{p}_{t+1} - \mathbf{p}_t|$ is less than a threshold and/or a maximum number of iterations t is achieved. The fixed-point iteration is very easy to implement. However, in practice it is often slow or prone to numerical instabilities whenever its denominator becomes too small [5] or even turns negative causing a divergence of the algorithm. To rectify these drawbacks we propose to choose the starting point within the training set as considered in the distance method. Hence $\mathbf{p}_0$ correspond to the second term of equation 14. If an RBF kernel is considered, the dot product of $\hat{\phi}(\mathbf{y}_j)$ with the training set $\mathbf{\Phi}$ can be used to find the neighbors, avoiding the computation of euclidian distances in feature space. As the dot product of every mapped data point in feature space is normalized to one, the closest neighbors are obtained by maximizing the expression $\max_i(\hat{\phi}^T(\mathbf{y}_j)\phi(\mathbf{x}_i)), i = 1, \ldots, K$. Computing the vector of dot products of $\hat{\phi}(\mathbf{y}_j)$ with the training set $\mathbf{\Phi}$ yields

$$\mathbf{r} = \mathbf{g}^T\mathbf{K} \tag{18}$$

The S closest neighbors, i.e. the ones that exhibit the highest dot product, are chosen. Selecting the corresponding points $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \ldots \mathbf{q}_S]$ in input space, the fixed point iteration should start with $\mathbf{p}_0 = (1/S)\mathbf{Q}\mathbf{j}_S$. Notice that the denominator of equation 17 is also a dot product between the reconstructed point and $\mathbf{p}_t$. That way the numerical instability on the first iteration can be avoided once $\mathbf{p}_0$ was chosen according to the maximal dot product criterion.

2.4. Diagonal Averaging

The last step is to transform the denoised trajectory matrix into a one-dimensional signal with N samples. After applying the described steps to every column of the trajectory

matrix $\mathbf{X}$, a new matrix $\hat{\mathbf{X}}$ of the denoised data is obtained. Notice that in general elements along each descending diagonal of $\hat{\mathbf{X}}$ will not be identical like in case of the original trajectory matrix $\mathbf{X}$. This can be cured by replacing the entries in each diagonal by their average, obtaining a new matrix $\mathbf{X}_r$. This procedure to get a matrix with all diagonals equal assures that the Frobenius norm of the difference $(\mathbf{X}_r - \hat{\mathbf{X}})$ has minimum value among all possible solutions. The noise-reduced one-dimensional signal, $\hat{x}[n]$, is then obtained by reverting the embedding of matrix $\mathbf{X}_r$, i.e. by forming the signal with an entry of each descendent diagonal.

3. NUMERICAL SIMULATIONS

The algorithms were implemented in MATLAB using the toolbox provided by Franc [6], where basic pattern recognition tools and kernel methods can be found. In the following the methods discussed above will be applied to toy examples as well as EEG signals. The toy examples are mainly used to illustrate the performance of the pre-image methods of the KPCA. Later, the EOG is extracted from a frontal EEG following a similar methodology to the one used to the toy example.

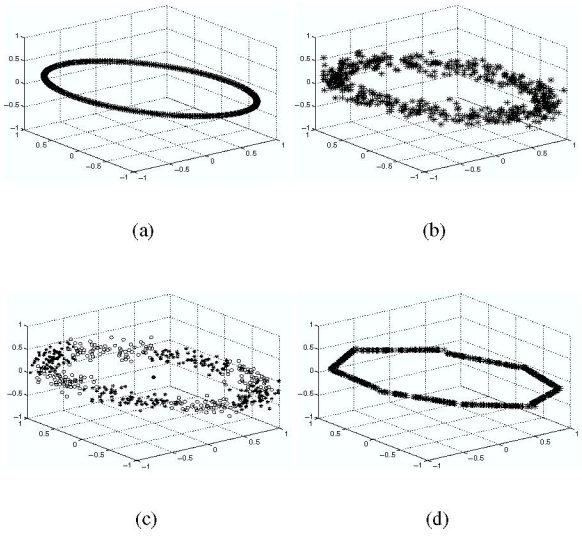


Fig. 1. Toy example: Embedding with $M = 3$: sinusoid (a);sinusoid with noise (b); local SSA: 8 clusters marked alternatingly with stars or circles (c), reconstruction with L=1 in each cluster (d)

3.1. Toy example: sinusoidal time-series

The toy example to be discussed considers an artificially generated sinusoid. The time series $\tilde{x}[n]$ with $N = 500$ samples was contaminated with Gaussian white noise $x[n] = \tilde{x}[n] + r[n]$ to result in a signal-to-noise ratio of $SNR = 20dB$ and was embedded in $M = 3$ dimensions (see Figure (1 a and b)). Using plain SSA the data can be projected into $L = 1$ or $L = 2$ principal directions to achieve the denoising. With $L = 1$ a straight line will result and with $L = 2$ the reconstructed version is similar to the noisy version. But using local SSA [7] which incorporates a clustering step in SSA an improved solution is achieved. Then, with the maximal principal component ($L = 1$) in each cluster, the underlying trajectory in phase space could be reconstructed satisfactorily in a piecewise linear way (see figure 1 c) and d)). An alternative approach to this nonlinear problem is to apply to KPCA to the trajectory matrix. KPCA was implemented using an RBF kernel with $\sigma =$

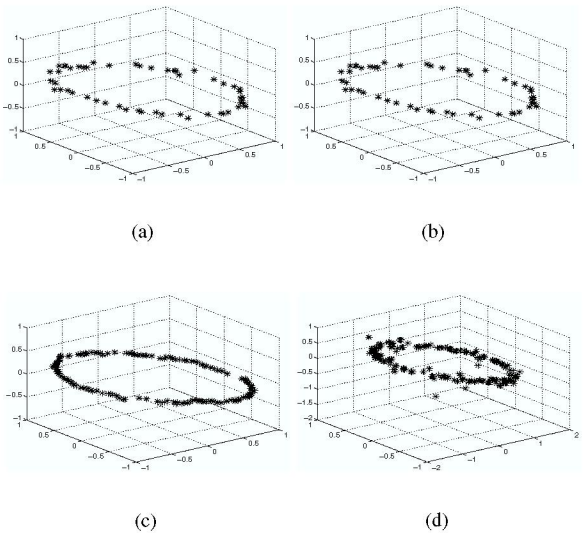


Fig. 2. Distance Method using: $S = 1$ neighbors: the closest point (a); distance method result (b);$S = 5$ the mean of neighbors (c); distance method (d).

$max_{i,j}(||\mathbf{x}_i - \mathbf{x}_j||), i = 1, \ldots, K$ and $j = i, \ldots, K$. The reconstruction in feature space was achieved using $L = 2$ principal components. To estimate the pre-image of every denoised point in feature space, both the distance and the fixed point method need a set of neighbors of a denoised point to be selected from the training set. Pre-images were estimated varying the number of neighbors S and using: 1) only the mean $\mathbf{p_0}$ of the selected neighbors (see 14); 2) the distance method (14); 3) fixed point method using $\mathbf{p_0}$ as starting point. The impact of the pre-image calculation can

be evaluated for the denoised one-dimensional signal $\hat{x}[n]$. The mean-square error was computed between the denoised sequence $\hat{x}[n]$ and the original sequence $\tilde{x}[n]$. The table 3.1 shows the mean-square error between the reconstructed and the original signal for all three methods of estimating the pre-image. We can see that, given an embedding dimension

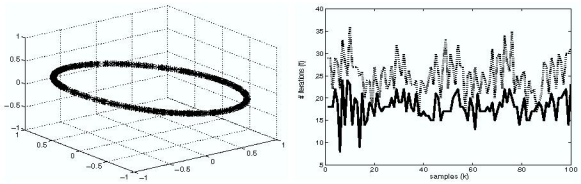
Table 1. Mean square error: original versus the denoised signal with KPCA using different algorithms to compute pre-image

S	**Distance**	**Mean**	**Fixed Point**
1	0.0843	0.0843	0.0843
2	0.0813	0.0800	0.0843
3	0.0813	0.0789	0.0843
4	0.1138	0.0771	0.0843
5	0.5381	0.0753	0.0843

$M = 3$ the number of neighbors S influences differently the estimation methods. The strongest impact is seen in the distance method. Fig. 2 -a) and -c) illustrate results, in the multidimensional space, using the mean only while Fig. 2 -b) and -d) show results obtained using the distance method (eqn. 14). Obviously the latter yields no significant improvement. Note that in [5] S was chosen always less than M. Further, considering the situation $S = 5 > M = 3$, the results obtained using the mean $\mathbf{p_0}$ are now significantly better than results using $\mathbf{p}$ from eqn. 14. The latter result is still noisy and shows too many outliers. In this simple example, the pre-image mean method is a strategy to find the best match of $\phi(\hat{\mathbf{y}}_j)$ in the training set based on single neighbor ($S = 1$) or by doing a kind of adaptive clustering ($S > 1$) in feature space. The denoised point in input space is obtained as the mean of points in input space which correspond to the nearest neighbors in feature space. Note that MSE decreases with increasing S (see second column of table). The iterative fixed point algorithm exhibits a more robust performance as it is not dependent on the starting point (see last column of table). Also notice that the underlying trajectory of the signal is smoother than what is obtained by the mean method (compare 3-a) and 2-a)-c)). The initialization of fixed point with the mean of the neighbors, turns algorithm faster (figure 3- b)) and avoids very low values in the denominator of the equation 17.

3.2. EEG data

Biomedical signals are often contaminated with artifact related signals which severely distort the signals to be investigated. As an example we will study the removal of the prominent EOG artefacts from EEG recordings. A frontal (Fp1-Cz) EEG channel sampled at $128Hz$ was used. A segment of the signal of $12s$ of duration containing high-amplitude EOG artifacts was considered and divided into 4



(a) (b)

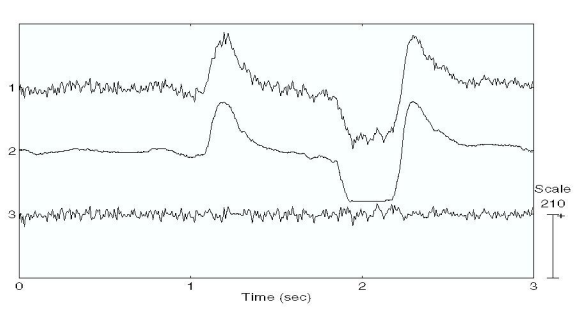
Fig. 3. Fixed-point method: using nearest neighbor $S = 1$ as starting (a); number of iterations using random (dotted) or mean of neighbors initialization (b).

sub-segments with $N = 384$ samples. KPCA was applied separately to each sub-segment. The one-dimensional signal was embedded in $M = 11$ dimensions, but only $L = 4$ principal components were used for reconstruction in the feature space. From the estimated pre-images the EOG was obtained after reverting the embedding. Then the extracted EOG signal was subtracted from the original EEG to obtain a corrected EEG.

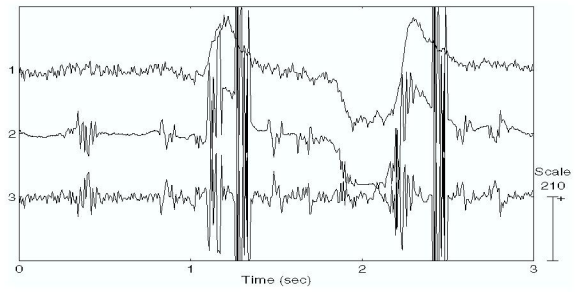
Visual inspection of the extracted signals confirmed that the results strongly depend on the method to estimate the pre-image corroborating results obtained with the toy example. Further experiments showed that the performance of the distance method is strongly dependent on the number S of neighbors yielding in some cases unreliable solutions (see fig. 4-b)). The fixed point algorithm, on the contrary, leads to more stable solutions (see figure 4-a) for an example) whatever is the number of neighbors used as starting point. To provide a global overview of the performance of the methods correlation coefficients, between a reference signal and signals resulting from changing either the method of estimating the pre-image or varying the number of neighbors S, are calculated. Figure 5 shows the results considering as reference the signal obtained with the fixed point method initialized with the best match in the training set ($S = 1$). Note that with $S = \{3, 6, 7, 12\}$ neighbors, the distance method does not yield a reliable solution (see also figure 4-b)) but also if $S > M$ the correlation coefficients are low. If $S < 20$, the result of the mean method is very close to the fixed point algorithm. Fig. 6 shows that the power spectral density is identical in the corrected EEG signals while the extracted EOG shows no visible difference between the two methods.

4. CONCLUDING REMARKS

In this work we show that KPCA can be applied to extract high-amplitude components of a signal. In this case the pre-image step influences the outcome but we show that



(a) Fix-Point method ($S = 12$)



(b) Distance method ($S = 12$)

Fig. 4. Segment of signal processed by KPCA using either a) the fixed point or b) the distance method to estimate the pre-image (*top*: the original EEG , *middle*: the extracted EOG signal, *bottom*: the corrected EEG).

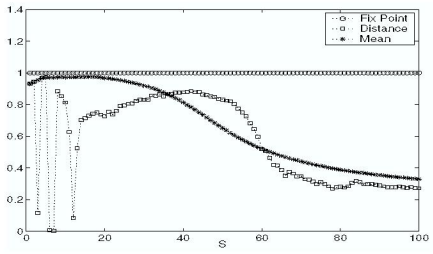


Fig. 5. Correlation coefficient between a reference signal and all the signals resulting from changing the pre-image method and/or varying S.

the fixed-point initialized with the mean of neighbors is the most robust. In a previous work [7] local SSA was also applied and similar results were obtained: the correlation coefficient between the extracted EOGs resulting from both methods is 0.999 and the one between the corrected EEG is 0.833. The latter value results from the fact that local SSA

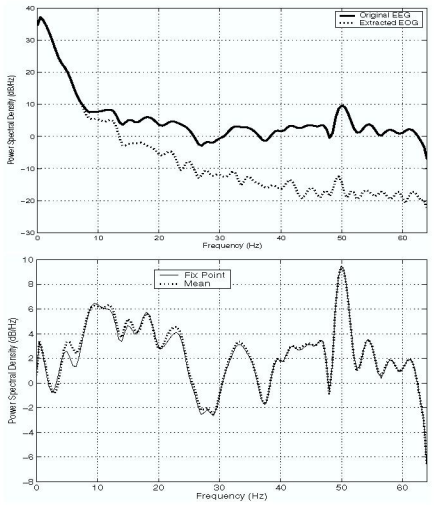


Fig. 6. Power Spectral density of *left*: the original EEG and extracted EOG, *right*: the corrected EEG comparing the Fixed Point vs Mean algorithm ($S = 12$).

also extracts the 50Hz interference. The low frequency contents of the corrected EEG is also affected differently by both methods however KPCA seems to preserve more information on the low frequency contents.

5. REFERENCES

[1] P. Gruber, K. Stadlthanner, M. Böhm, F.J. Theis, E.W. Lang, A.M. Tomé, A.R. Teixeira, C.G. Puntonet, and J.M. Gorriz Saéz, "Denoising using local projective subspace methods," *Neurocomputing*, vol. in print.

[2] Chang Huai You, Soo Ngee Koh, and Susanto Rahardja, "Signal subspace speech enhancement for audible noise reduction," in *ICASPP 2005*, Philadelphia, USA, 2005, vol. I, pp. 145–148, IEEE.

[3] M. Ghil, M.R. Allen, M. D. Dettinger, K. Ide, and et al, "Advanced spectral methods for climatic time series," *Reviews of Geophysics*, vol. 40, no. 1, pp. 3.1–3.41, 2002.

[4] Berhard Schölkopf, Sebastian Mika, Chris J. Barges, Philip Knirsch, Klaus-Robert Müller, Grunnar Ratsch, and Alexander J.Smola, "Input space versus feature space in kernel-based methods," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1000–1016, 1999.

[5] James T. Kwok and Ivor W. Tsang, "The pre-image problem in kernel methods," *IEEE Transactions on Neural Networks*, vol. 15, no. 6, pp. 1517–1525, 2004.

[6] Vojtěch Franc and Václav Hlaváč, "Stastical pattern recognition toolbox for matlab," 2004.

[7] A. R. Teixeira, A. M. Tomé, E.W.Lang, P. Gruber, and A. Martins da Silva, "On the use of clustering and local singular spectrum analysis to remove ocular artifacts from electroencephalograms," in *IJCNN2005*, Montréal, Canada, 2005, pp. 2514–2519, IEEE.