# EXPLOITING LOW-RANK APPROXIMATIONS OF KERNEL MATRICES IN DENOISING APPLICATIONS

*A. R.Teixeira, A. M. Tomé* *

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

*E. W. Lang,* †

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

**ABSTRACT**

The eigendecomposition of a kernel matrix can present a computational burden in many kernel methods. Nevertheless only the largest eigenvalues and corresponding eigenvectors need to be computed. In this work we discuss the Nyström low-rank approximations of the kernel matrix and its applications in KPCA denoising tasks. Furthermore, the low-rank approximations have the advantage of being related with a smaller subset of the training data which constitute then a basis of a subspace. In a common algebraic framework we discuss the different approaches to compute the basis. Numerical simulations concerning the denoising are presented to compare the discussed approaches.

## 1. INTRODUCTION

Kernel Principal Component Analysis (KPCA) relies on a non-linear mapping of given data to a higher dimensional space, called feature space. Then KPCA can simultaneously retain the non-linear structure of the data while denoising is achieved with better performance because the projections are accomplished in the higher-dimensional feature space. The KPCA method represents a projective subspace technique applied in feature space and created by a non-linear transformation of the original data. In the feature space a linear principal component analysis is performed. The denoising is achieved by considering the projections related to the largest eigenvalues of the covariance/scatter matrix. The mapping in the feature space is avoided by using kernel functions which implicitly define a dot product in the feature space computed using the data in input space [1]. The kernel matrix (a dot product matrix) of the mapped data is easily achieved and naturally its dimension depends on the size of the data set. The entries $(i, j)$ of the matrix depend on the corresponding data points and are computed according to the defined kernel function. The kernel matrix dimension represents a computational burden once its eigendecomposition must be achieved. In practice, the goal of projective subspace techniques is to describe the data with reduced dimensionality by extracting meaningful components while still retaining the structure of the raw data. Then only the projections on the directions corresponding to the most significant eigenvalues of the kernel (or covariance matrix) need to be computed. The exploitation of methods like Nyström to achieve the low rank eigendecomposition is a strategy that has been considered [2],[3]. Furthermore those techniques can also achieve a solution without the manipulation of the full matrix. We show how Nyström's method can be applied to KPCA leading to what is usually known as greedy KPCA. In this work we compare the different Nyström approaches to greedy KPCA under the same algebraic formulation. The main differences are the complexity of the different approaches and the properties of the computed projections. An experimental study will show the performance of the methods in what concerns denoising applications.

*The authors are grateful to financial support by CRUP. A.R. Teixeira received a PhD Scholarship (SFRH/BD/28404/2006) supported by the Portuguese Foundation for Science and Technology (FCT).

†The authors gratefully acknowledge support by the DAAD.

## 2. DENOISING USING GREEDY APPROACH

Kernel Principal Component Analysis (KPCA) relies on a non-linear mapping of given data to a higher dimensional space, called feature space. Without loosing generality, let's assume that the data set is centered and split into two parts yielding the mapped data set

$$\begin{aligned} \mathbf{\Phi} &= [\phi(\mathbf{x}_1)\phi(\mathbf{x}_2)\dots\phi(\mathbf{x}_r), \phi(\mathbf{x}_{r+1})\dots\phi(\mathbf{x}_K)] \\ &= [\ \mathbf{\Phi}_R \quad \mathbf{\Phi}_S] \end{aligned} \tag{1}$$

In denoising applications, the first step of KPCA is to compute the projections of a mapped data set onto a feature subspace. Considering $L$ eigenvectors (columns of $\mathbf{U}$) of a covariance matrix (a correlation matrix if the data is centered) corresponding to the $L$ largest eigenvalues, the pro-

jections of the $K$ vectors of the mapped data set $\mathbf{\Phi}$ are

$$\mathbf{Z} = \mathbf{U}^T \mathbf{\Phi} \tag{2}$$

The columns of the matrix $\mathbf{U}$ form the basis in feature space onto which to project the data set. This basis can be written as a linear combination of the mapped data

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

The matrix $\mathbf{A}$ is a matrix of coefficients and either $\mathbf{\Phi}_B = \mathbf{\Phi}$ (KPCA) or $\mathbf{\Phi}_B = \mathbf{\Phi}_R$ (greedy KPCA), representing a subset of the data set only. Note that the column $j$ of $\mathbf{Z}$ depends on the dot products $\mathbf{\Phi}_B^T \phi(\mathbf{x}_j)$. However to avoid an explicit mapping into feature space, all data manipulations are achieved by dot products [1] and the kernel trick is applied. For instance, using RBF kernel, the dot product between a vector $i$, belonging to $B$ subset, and $\phi(\mathbf{x}_j)$ is computed with a kernel function that only depends on the input data

$$k(\mathbf{x}_i, \mathbf{x}_j) = exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}) \tag{4}$$

Finally, to recover the noise-reduced signal after denoising in feature space, the non-linear mapping must be reverted, i.e. the pre-image in input space of every signal, denoised and reconstructed in feature space, must be estimated. Denoising using KPCA thus comprises two steps after the computation of the projections in the feature space: a) the reconstruction in feature space and b) the estimation of the pre-image of the reconstructed point $\hat{\phi}(\mathbf{x}_j) = \mathbf{U}\mathbf{z}_j$, where $\mathbf{z}_j$ represents the projections of a noisy point $\phi(\mathbf{x}_j)$. These two steps can be joined together by minimizing the Euclidian distance of the image $\phi(\mathbf{p})$ of a yet unknown point $\mathbf{p}$ from $\hat{\phi}(\mathbf{x}_j)$

$$\begin{aligned} \tilde{d}^{(2)} &= \|\phi(\mathbf{p}) - \hat{\phi}(\mathbf{x}_j)\|^2 \\ &= (\phi(\mathbf{p}) - \hat{\phi}(\mathbf{x}_j))^T (\phi(\mathbf{p}) - \hat{\phi}(\mathbf{x}_j)) \end{aligned} \tag{5}$$

The central idea of the fixed-point method [1] consists in computing the unknown pre-image of a reconstructed point in the projected feature subspace by finding a $\mathbf{p}$ which minimizes the distance (see eqn. 5). If an RBF kernel is considered, the iterative procedure is described by the following equation

$$\mathbf{p}_{t+1} = \frac{\mathbf{X}_B(\mathbf{g} \diamondsuit \mathbf{k}_{p_t})}{\mathbf{g}^T \mathbf{k}_{\mathbf{p_t}}} \tag{6}$$

where $\diamondsuit$ represents a Hadamard product, $\mathbf{g} = \mathbf{A}\mathbf{z}_j$. 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 subset $\mathbf{X}_B$. The algorithm must be initialized and $\mathbf{p}_0 \equiv \mathbf{x}_i$ is a valid choice [10]. The points $\mathbf{p}_k$ then form the columns of $\hat{\mathbf{X}}$, the noise-reduced multidimensional signal.

### 2.1. Computing the Basis

The projections $\mathbf{Z}$ of the training set are also related with the eigenvectors $\mathbf{V}$ of a matrix computed using only dot products ($\mathbf{K}$), the kernel matrix. Naturally the entries of the matrix can be easily achieved using the kernel tricky. Considering the singular value decomposition of the training data set using $R$ non-zero singular values we can write

$$\mathbf{\Phi} = \mathbf{U}\mathbf{D}^{1/2}\mathbf{V}^T \tag{7}$$

where $\mathbf{D}$ is a diagonal matrix with ordered eigenvalues ($\lambda_1 > \lambda_2 > ... > \lambda_L ... > \lambda_R$) of kernel matrix (or of the scatter matrix); and $\mathbf{V}$ and $\mathbf{U}$ are the $R$ eigenvectors of the kernel and scatter matrices, respectively. Considering an SVD approximation with $L$ most significant singular values and substituting it in equation (2), the $L$ projections are

$$\mathbf{Z} = \mathbf{D}^{1/2}\mathbf{V}^T \tag{8}$$

where each column $j$ of $\mathbf{Z}$, an $L \times K$ matrix, is related with a corresponding row of $\mathbf{V}$ and correspond to the projections of $\phi(\mathbf{x}_j)$. The two approaches, KPCA and greedy KPCA, respectively, arise from two distinct strategies to deal with the eigendecomposition of the kernel matrix ($\mathbf{K}$) of the data set. In KPCA the whole data set is used to compute the kernel matrix, then $\mathbf{A}$ is computed using the largest eigenvalues ($\mathbf{D}$) and corresponding eigenvectors. The combination of equations (2) and (8) leads to $\mathbf{U}^T\mathbf{\Phi} = \mathbf{D}^{1/2}\mathbf{V}^T$. Multiplying both sides of the previous equation by $\mathbf{V}\mathbf{D}^{-1/2}$, and considering the columns of eigenvector matrices orthogonal, the basis vector matrix is

$$\mathbf{U} = \mathbf{\Phi}\mathbf{V}\mathbf{D}^{-1/2} \tag{9}$$

In greedy KPCA a low-rank approximation of the kernel matrix is considered. This leads to the eigendecomposition of matrices with reduced size. Considering that the training set is divided into two subsets, the $K \times K$ kernel matrix can be written in block notation [3],[2]

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_r & \mathbf{K}_{rs} \\ \mathbf{K}_{rs}^T & \mathbf{K}_s \end{bmatrix} \tag{10}$$

where the $\mathbf{K}_r$ is the kernel matrix within subset $\mathbf{\Phi}_R$, $\mathbf{K}_s$ is the kernel matrix within the subset $\mathbf{\Phi}_S$ and $\mathbf{K}_{rs}$ is the kernel matrix between subset $\mathbf{\Phi}_R$ and $\mathbf{\Phi}_S$. The approximation is written using the upper blocks of the original matrix [3], [2]

$$\tilde{\mathbf{K}} = \begin{bmatrix} \mathbf{K}_r \\ \mathbf{K}_{rs}^T \end{bmatrix} \mathbf{K}_r^{-1} \begin{bmatrix} \mathbf{K}_r & \mathbf{K}_{rs} \end{bmatrix} \tag{11}$$

It can be verified that the lower block is approximated by $\mathbf{K}_s \approx \mathbf{K}_{rs}^T \mathbf{K}_r^{-1} \mathbf{K}_{rs}$. The $R$ eigenvectors $\mathbf{V}$ corresponding to the $R$ largest eigenvalues are then computed as

$$\mathbf{V}^T = \mathbf{H}^T \begin{bmatrix} \mathbf{K}_r & \mathbf{K}_{rs} \end{bmatrix} = \mathbf{H}^T \mathbf{\Phi}_R^T \begin{bmatrix} \mathbf{\Phi}_R & \mathbf{\Phi}_S \end{bmatrix} \tag{12}$$

There are different approaches to compute H conducting to an orthogonal or to a non-ortoghonal solution to V. Then the projections in the feature space of the data set, Z, can be non-correlated or correlated as can be easily verified by the manipulation of eqn. (8).

### 2.1.1. *Non-orthogonal Approach*

In this case a non-orthogonal matrix $\mathbf{V}$ is computed using the eigendecomposition of $\mathbf{K}_p = \mathbf{V}_p \mathbf{D}_p \mathbf{V}_p^T$ [3]. Considering that the eigenvalues of K and $\mathbf{K}_p$ are related by a common scale factor $(K/R)$, the matrix H is

$$\mathbf{H} = \mathbf{V}_p \mathbf{D}^{-1} \tag{13}$$

Then manipulating the equations (12), (8) and (2) the basis vector matrix is

$$\mathbf{U} = \mathbf{\Phi}_R \mathbf{V}_p \mathbf{D}^{-1/2} \tag{14}$$

The number of columns of U is $L$ by considering the $L$ largest eigenvalues and corresponding eigenvectors of the matrix $\mathbf{K}_p$.

### 2.1.2. *Orthogonal Approach*

The alternative approaches consider the kernel matrix decomposed as $\bar{\mathbf{K}} = \mathbf{C}^T\mathbf{C}$, where C has dimension $R \times K$ and is computed as follows

$$\mathbf{C} = \begin{bmatrix} \mathbf{L} & \mathbf{L}^{-T}\mathbf{K}_{ps} \end{bmatrix} \tag{15}$$

where L can be computed using the Cholesky decomposition [4], [5] or the square root [2] of $\mathbf{K}_p$

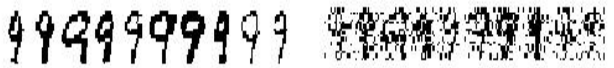
- $\mathbf{K}_p = \mathbf{L}^T\mathbf{L}$, where L is a triangular matrix. Note that if the matrix is symmetric positive definite there exists an unique $R \times R$ triangular matrix that accomplishes the decomposition without any pivoting scheme. Alternatively, an incomplete Cholesky decomposition of the full matrix K can be performed [4]. In this case the matrix C is the output of the algorithm and the indices of the pivoting can identify the subset $R$.
- $\mathbf{L} = \mathbf{K}_p^{1/2} = \mathbf{V}_p \mathbf{D}_p^{1/2} \mathbf{V}_p^T$, which is a symmetric matrix.

The low rank approximation of $\bar{\mathbf{K}} = \mathbf{V}\mathbf{D}\mathbf{V}^T$ is based on the eigendecomposition of an $R \times R$ matrix defined by

$$\mathbf{Q} = \mathbf{C}\mathbf{C}^T = \mathbf{V}_q \mathbf{D} \mathbf{V}_q^T \tag{16}$$

The result of this eigendecomposition as well as the decomposition of $\mathbf{K}_p$ leads to

$$\mathbf{H} = \mathbf{L}^{-1}\mathbf{V}_q \mathbf{D}^{-1/2} \tag{17}$$



(a) (b)

**Fig. 1**. Set of digits (a) Original, (b) with Gaussian noise ($\sigma^2 = 0.25$)

The matrix of eigenvectors V is orthogonal as can easily be verified. By the manipulation of the equations (12), (8),(2) the basis vector matrix is

$$\mathbf{U} = \mathbf{\Phi}_R \mathbf{L}^{-1} \mathbf{V}_q \tag{18}$$

The number of columns of U is $L$ by forming the matrix $\mathbf{V}_q$ with the eigenvectors corresponding to the $L$ largest eigenvalues.

### 2.2. Splitting the data set

In the last section the training set is considered split into two groups. In what concerns the Nyström approach it is said that the first $R$ rows should represent the linear independent rows of the kernel matrix. Usually, $R$ rows randomly chosen are used to organize the upper block of the kernel matrix. This strategy is also suggested by most of published works [3],[6], [2] for huge data sets considering that there is an high probability of the random chosen subset still represent the training set distribution. However, the quality of the approximation is ruled by the norm of Schur's complement. And some works consider practical criteria derived from the Schur's complement to iteratively update the subset $R$. The methodology is based on the minimization of the trace $tr(\mathbf{K}_s - \mathbf{K}_{ps}^T \mathbf{K}_p^{-1} \mathbf{K}_{ps})$. By identifying the maximal value of the trace operator (the pivot), an element of subset $S$ is moved to the subset $R$ and the matrix $\mathbf{K}_p$ increases its size while the others decrease. The process stops when the trace of the matrix corresponding to the actual approximation is less than a threshold or even when the matrix $\mathbf{K}_p$ is not well conditioned. In [7] the criterion is defined as the minimization of a square error, in [8] and in [9] as relative error. The stop conditions are thresholds [7] or the rank of matrix $\mathbf{K}_p$ [9].

## 3. NUMERICAL SIMULATIONS

The goal of the numerical simulations is to study the impact of the projection method and its relation with the choice of subset $\mathbf{\Phi}_R$. For convenience of the exposition we point out the following schemes to deal with the computation of the parameters of the model:

- **Chol**- Incomplete Cholesky decomposition using symmetric pivoting. The subset $\mathbf{\Phi}_R$ is chosen according

the set of pivoting indices and the matrix of basis vectors is computed using eqn. (18).

- **Cholr**- Random selection of the subset $\mathbf{\Phi}_R$ followed by the Cholesky decomposition of $\mathbf{K}_r$. The matrix C and the matrix of basis vectors are computed using eqn. (15) and eqn. (18), respectively.
- **Nort**- random selection of $\mathbf{\Phi}_R$ using the eigendecomposition of $\mathbf{K}_r$ to compute the matrix of basis vector as described by eqn. (14).

The kernel matrix was computed using the RBF kernel with $\sigma = max_i(\|\mathbf{x}_i - \mathbf{x}_{mean}\|), i = 1, \ldots, K$, where $\mathbf{x}_{mean}$ is the mean of the data set. The matrix of basis vector $\mathbf{U}$ with $L$ columns is computed according to the described methods and the data is projected. Finally, to yield a denoised version $\hat{\mathbf{x}}_k$ of the noisy $\mathbf{x}_k$, the pre-image $\hat{\mathbf{x}}_k$ of the reconstructed $\hat{\Phi}(\mathbf{x}_k) = \mathbf{U}\mathbf{z}_k$ was estimated applying the fixed point iteration as described by eqn. (6).

### *3.0.1. USPS data set*

The data set consists of $16 \times 16$ handwritten digits. Then the input data vector, $\mathbf{x}_k$ has dimension 256 and is formed by row concatenation of the original image after adding white Gaussian noise (zero mean and variance of 0.25). Figure 1 shows a set of digits and its noisy versions. The kernel matrix for each type of digit, computed with the total number of elements, is a full rank matrix (the smallest eigenvalues are $\simeq 0.17$). The data set for each digit type has a different number of elements (in the range $568 - 1005$) so we consider to constitute the subset $\mathbf{\Phi}_R$ with a fixed percentage of the available data and present results for 5% and 30%. Notice that adding to each digit a noise with fixed variance the signal-to-noise ratio (SNR) is different (see second column of table 1). The denoising was achieved by projecting the data onto the leading $L < R$ eigenvectors founded according to leveling off of the eigenspectrum of the respective kernel matrix (in the range of $5 - 15$).

The orthogonal approaches (**Chol** and **Cholr**) have better performance than the nonorthogonal approach (**Nort**). Fig. 2 illustrates the performance of the methods for the two subsets and we can verify that the differences between the orthogonal approaches might not be visually detected. The table 1 presents the mean values of SNR of the denoised images for all the digits of the data set. And can also be verified that all methods perform better if the subset $\mathbf{\Phi}_R$ is larger. However the differences in performance for the two subsets are less accentuated with **Chol**, it does not exceed the $0.8dB$ for all the digits. It has to be noticed that **Cholr** presents a similar level of performance for the larger subset, the difference with **Chol** is less than $0.4dB$. This difference in performance might not justify an increase in the complexity of the algorithm mainly because is not easy to find

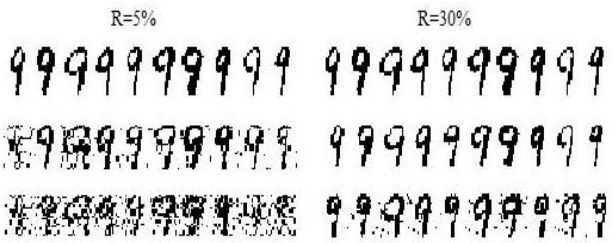



**Fig. 2**. Set of denoised digits: first line -**Chol**, second line-**Cholr**,third line -**Nort**

a threshold to stop the decomposition and we have to deal with the whole data set to implement the pivoting scheme [4].

**Table 1**. SNR of the original and denoised images

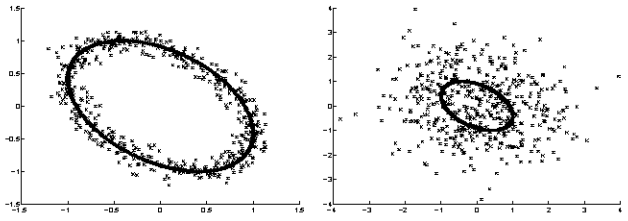
| | | | SNR | | |
|---|---|---|---|---|---|
| Digit | Image | R | **Chol** | **Cholr** | **Nort** |
| 1 | $\bar{x} = 0.162$ | 5 % | 2.879 | 2.298 | 1.580 |
| | $\sigma^2 = 2.177$ | 30 % | 3.471 | 3.084 | 2.016 |
| 2 | $\bar{x} = 2.729$ | 5 % | 4.196 | 2.547 | 2.346 |
| | $\sigma^2 = 2.834$ | 30 % | 4.927 | 4.897 | 4.06 |
| 3 | $\bar{x} = 2.890$ | 5 % | 4.843 | 3.031 | 2.8928 |
| | $\sigma^2 = 2.077$ | 30 % | 5.235 | 5.108 | 4.372 |
| 4 | $\bar{x} = 1.532$ | 5 % | 3.788 | 1.865 | 1.678 |
| | $\sigma^2 = 2.780$ | 30 % | 4.085 | 3.985 | 3.450 |
| 5 | $\bar{x} = 2.967$ | 5 % | 4.498 | 3.086 | 3.018 |
| | $\sigma^2 = 2.202$ | 30 % | 5.269 | 5.118 | 4.859 |
| 6 | $\bar{x} = 2.317$ | 5 % | 4.343 | 3.016 | 2.897 |
| | $\sigma^2 = 2.35$ | 30 % | 5.030 | 5.149 | 4.247 |
| 7 | $\bar{x} = 1.436$ | 5 % | 4.126 | 2.081 | 1.999 |
| | $\sigma^2 = 2.774$ | 30 % | 4.671 | 4.453 | 3.836 |
| 8 | $\bar{x} = 2.771$ | 5 % | 3.891 | 2.255 | 2.615 |
| | $\sigma^2 = 2.235$ | 30 % | 4.698 | 4.613 | 4.431 |
| 9 | $\bar{x} = 1.753$ | 5 % | 4.425 | 3.012 | 2.767 |
| | $\sigma^2 = 2.591$ | 30 % | 4.877 | 4.722 | 4.215 |

### *3.0.2. Time series Denoising*

Considering a signal embedded in its time-delayed coordinates. Embedding can be regarded as a mapping that transforms a one-dimensional time series $x[n], n = 0...N-1$, to a multidimensional sequence of $K = N - M + 1$ lagged vectors

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

The lagged vectors form a point in a space with dimension $M$.The multidimensional signal can be denoised using KPCA. The points $\mathbf{p}_k$ then form the columns of $\hat{\mathbf{X}}$,



(a) SNR=20dB (b) SNR=0dB

**Fig. 3**. Embedded signals in 2D space. Sinusoid (+) and sinusoid+gaussian noise(*)

the noise-reduced multidimensional signal matrix in input space. The one-dimensional 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 $\hat{\mathbf{X}}$ [10].

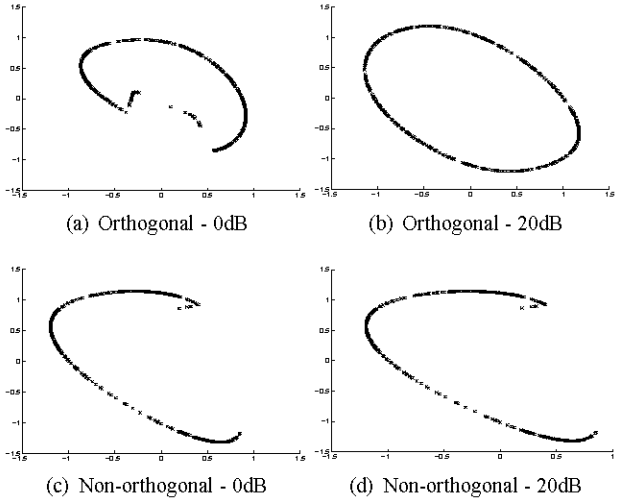
Denoising a sinusoid

Fig. 3 shows the original sinusoid and noisy sinusoid embedded in $2D$ space. The kernel matrix of the noisy 2D signal has a dimension of $K = 498$ but the rank is 141 and 327 for $SNR = 20dB$ and $SNR = 0dB$, respectively. In the feature space, the subspace dimension to recover the embedded sinusoid was $L = 2$. The three strategies to compute the basis vector $\mathbf{U}$ in the feature space were implemented varying the size of subset $\mathbf{\Phi}_R$ between 10 and the rank of the kernel matrix. Table 2 shows the mean square errors between the original sinusoid and denoised versions for two of the total set of experiments. Fig.4 illustrates in 2D input space the results when subset $\mathbf{\Phi}_R$ has $R = 10$ elements. The figure shows that the ellipse trajectory of the embedded sinusoid is recovered with mean-square error of $MSE \simeq 0.16$. The table also shows that the orthogonal approaches (non-correlated projections) are always better than the corresponding non-orthogonal approach. The difference is lower when the size of subset $\mathbf{\Phi}_R$ increases. However,

**Table 2**. Mean square error (MSE) between original and denoised versions. Note that **Cholr** and **Nort** the entries are mean of the result of 1000 random subset selections

| SNR | | R=10 | R=50 |
|---|---|---|---|
| 0dB | **Chol** | 0.152 | 0.141 |
| | **Cholr** | 0.368 | 0.168 |
| | **Nort** | 0.671 | 0.386 |
| 20dB | **Chol** | 0.004 | 0.004 |
| | **Cholr** | 0.162 | 0.004 |
| | **Nort** | 0.415 | 0.006 |

this toy example shows that if the SNR decreases the subset size (in the random strategies like **Cholr** and **Nort**) should increase to assure that the subset covers the distribution of the input data set. And in fact, the pivoting scheme of Cholesky assures the coverage of the input data distribution in a systematic way.



(a) Orthogonal - 0dB (b) Orthogonal - 20dB

(c) Non-orthogonal - 0dB (d) Non-orthogonal - 20dB

**Fig. 4**. Denoising the embedded sinusoid considering different levels of noise, R=10 with **Cholr** and **Nort**.

Removing high-amplitude artifact

We apply the method to extract prominent artifacts like electro-oculograms (EOG) in electro-encephalograms (EEG). Note that in this example, the artifact-related contributions to the recorded EEG signals are considered "the signal" and the actual EEG signal is considered a "sort of a broadband noise". Consequently, we can use the projective subspace techniques referred to above to separate the dominating artifacts from the "pure" EEG signals. Then if $\hat{x}[n]$ corresponds to the high amplitude artifact, then the corrected signal is computed as $y[n] = x[n] - \hat{x}[n]$. A segment of a frontal EEG channel with $6s$ is shown in Fig. 5 (first plot). The signal was embedded with $M = 11$ and the matrix $\mathbf{K}$, corresponding to the multidimensional signal, is full rank. The three proposed variants of the greedy approach were applied considering the subset $R$ with 5% and 30% of whole multidimensional data set. The matrix of vector basis $\mathbf{U}$ has $L = 6$ columns.

The results with the EEG signal confirm the results obtained with other data sets. The orthogonal approaches to compute the basis have always the best performance for the smallest subset $R$. But the difference between **Chol** and **Cholr** was not visually detected, in fact the correlation coefficients between the corrected signals by the two methods is always $> 0.91$. While with **Nort** are very clear, mainly if the subset $R$ is small (see Fig. 6 ). The correlation coefficient between the corrected signals shown in the figure is around 0.70. As before if the training subset $R$ increases the differences are less visible but even in that case the corre-

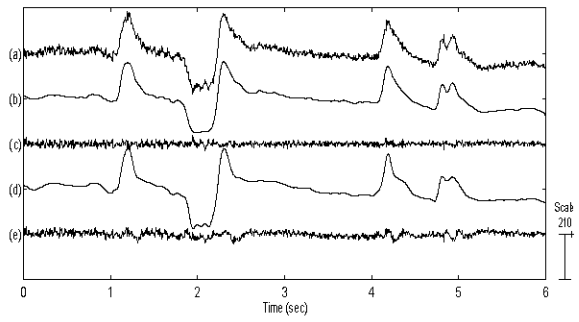



**Fig. 5**. Using 30 % of the data. (a)- Original EEG; (b) - Extracted EEG by **Chol**; (c) - Corrected EEG by **Chol**; (d) - Extracted EEG by **Nort**; (e) - Corrected EEG by **Nort**

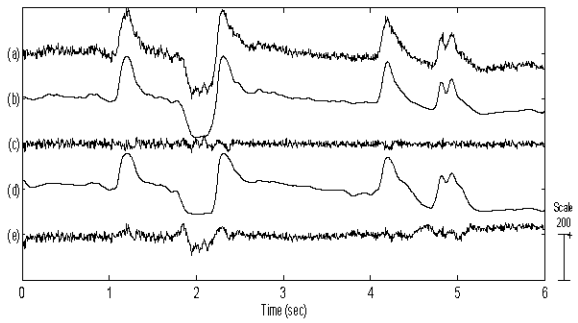



**Fig. 6**. Using 5 % of the data. (a)- Original EEG; (b) - Extracted EEG by **Chol**; (c) - Corrected EEG by **Chol**; (d) - Extracted EEG by **Nort**; (e) - Corrected EEG by **Nort**

lation coefficient between the signals corrected EEG of the figure (see Fig. 5) is 0.79. KPCA was also applied in previous work [10] computing the kernel matrix in segments of 3s, i.e, dividing the segment of figure into two subsegments and compute the basis vector in each. Comparing the corrected EEGs obtained with KPCA and this greedy approach no visual difference can be found and the correlation coefficient is around 0.94. Our goal is to develop the technique using segments of 10$s$ (typical window size on displays) without having to divide the signal into subsegments.

## 4. CONCLUDING REMARKS

These simulations discussed show that greedy KPCA performs better with orthogonal approaches both for rank or non-rank defficient kernel matrices. The best results (in what concerns the size of subset $R$) were always achieved with incomplete Cholesky with symmetric pivoting (**Chol**). But Cholesky decomposition (**Cholr**) after a random choice can achieve very similar results at expenses of increasing the size of subset $R$. The tradeoff between increasing the complexity of the algorithm by adding the pivoting scheme versus increasing the size of the subset should be further studied. Mainly because the optimum thresholds (of the value of the pivots or the value of the norm of Schur) to stop the algorithm are dependent on the problem and/or the level of noise. The artifact extraction in EEG recording, with the orthogonal approach schemes will be further studied in order to provide a tool to remove artifacts of critical segments like the onset of epileptic seizures in long term recording sessions.

## 5. REFERENCES

[1] Klaus-Robert Müller, Sebastian Mika, Gunnar Rätsch, Koji Tsuda, and Bernhard Schölkopf, "An introduction to kernel-based algorithms," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 181–202, 2001.

[2] Charless Fowlkes, Sergie Belongie, Fan Chung, and Jitendra Malik, "Spectral grouping using the nyström method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 214–225, 2004.

[3] Christopher K.I. Williams and Mathias Seeger, "Using the Nyström method to speed up kernel machines," in *Advances in Neural Information Processing Systems*. 2000, pp. 682–688, MIT Press.

[4] Francis R. Bach and Michael I. Jordan, "Kernel independent component analysis," *Journal of Machine Learning Research*, vol. 3, pp. 1–48, 2002.

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

[6] Rong Liu, Varun Jain, and Hao Zhang, "Sub-sampling for effcient spectral mesh processing," in *Computer Graphics International*, Seidel et all, Ed., 2006, pp. 172–184.

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

[8] G. Baudat and F. Anouar, "Kernel-based methods and function approximation," in *International Joint Conference on Neural Networks*, Washington, USA, 2001, vol. 2, pp. 1244–1249, IEEE.

[9] Gavin C. Cawley and Nicola L. C. Talbot, "Efficient formation of a basis in a kernel induced feature space," in *European Symposium on Artificial Neural Networks*, Michel Verleysen, Ed., Bruges, Belgium, 2002, pp. 1–6, d-side.

[10] A. R. Teixeira, A.M.Tomé, E.W.Lang, R. Schachtner, and K.Stadlthanner, "On the use of KPCA to extract artifacts in one-dimensional biomedical signals," in *Machine Learning for Signal Porcessing, MLSP 2006*, Seán McLoone, Jan Larsen, Marc Van Hulle, Alan Rogers, and Scott C. Douglas, Eds., Dublin, 2006, pp. 385–390, IEEE.