Proceedings of the 28th IEEE
EMBS Annual International Conference
New York City, USA, Aug 30-Sept 3, 2006

Kernel Principal Component Analysis through Time for Voice Disorder Classification

Mauricio Alvarez, Ricardo Henao, Germán Castellanos, Juan I. Godino, Alvaro Orozco

***Abstract*— Kernel Principal Component analysis is a nonlinear generalization of the popular linear multivariate analysis method. However, this method assumes that the observed data is independent, a disadvantage for many practical applications. In order to overcome this difficulty, the authors propose a combination of Kernel Principal Component analysis and hidden Markov models. The novelty of the proposed method consists mainly in the way in which a static dimensionality reduction technique has been combined with a classic mixture model in time, to enhance the capabilities of transformation, reduction and classification of voice disorder data. Experimental results show improvements in classification accuracies even with highly reduced representations of the two databases used.**

I. INTRODUCTION

Principal Component Analysis (PCA) is a popular and powerful technique for feature extraction, dimensionality reduction and probably the most used of the techniques of multivariate analysis [1]. One of the most common definitions of PCA is that, for a set of observed d-dimensional data vectors $\mathbf{X} = \{\mathbf{x}_n\}$, $n = 1, \ldots, N$, the p principal axes $\mathbf{w}_i$, $i = 1, \ldots, p$, are those orthonormal axes onto which the retained variance under linear projection is maximal. Albeit PCA has many advantages, (i) it is assumed that the observed data is independent and usually multivariate normal and (ii) the subspace itself is restricted to a linear mapping, where high-order statistical information is discarded.

The first disadvantage is a non trivial problem because for time series, perhaps the most common type of non-independent data, even a very weak dependence relation between the data makes PCA unappropriate. Several techniques have been developed to exploit the temporal dependencies in order to optimize the representation of the data from a temporal context [2], [3], [4], [5].

Kernel Principal Component Analysis (KPCA) [6], overcomes the second disadvantage by using a "kernel trick" to avoid the direct evaluation of the required dot product in a high-dimensional feature space using a kernel function, in such way that the high-order statistical information is readily captured.

This work was co-financed under the grant TIC-2003-08956-C02-00 from the Ministry of Science and Technology of Spain and Colciencias under contract 1110-14-17904

M. Alvarez and A. Orozco are lecturers of Program of Electrical Engineering, Universidad Tecnológica de Pereira, Colombia. {malvarez,aaog}@utp.edu.co

R. Henao is lecturer of the School of Electrical Technology, Universidad Tecnológica de Pereira, Colombia. rhenao@utp.edu.co

G. Castellanos is with the Department of Electronic Engineering, Universidad Nacional de Colombia, Manizales, Colombia. gcastell@telesat.com.co

J.I. Godino is with EUIT de Telecomunicación, Universidad Politécnica de Madrid, Spain. igodino@ics.upm.es

In this paper a temporal version of KPCA is introduced by using a hidden Markov model (HMM) as a way to obtain an optimized representation of the observed data through time. With this scheme, every data point has an associated local representation corresponding to the most probable state produced by a trained HMM. The integration between HMM and KPCA was done by incorporating a factorization of the training data and the responsibility weights in the covariance of the observation model in the HMM in such way that KPCA can be readily calculated.

This paper is organized as follows: sections II and III contain reviews of KPCA and HMM respectively. In section IV an extension for KPCA through time is presented. In section V experimental results for voice disorder databases are reported and finally in section VI the conclusions of the entire work.

II. KERNEL PRINCIPAL COMPONENT ANALYSIS

PCA is intended to linearly work in the original observation data space $\mathbb{R}^d$. KPCA on the other hand, operates in a high-dimensional feature space F, which is related to the input space by a possible nonlinear map $\boldsymbol{\phi}(\cdot) : \mathbb{R}^d \rightarrow F$ where the dimension of F is greater than d and possibly infinite. The mapped set in F is denoted by $\boldsymbol{\Phi} = [\boldsymbol{\phi}(\mathbf{x}_1), \ldots, \boldsymbol{\phi}(\mathbf{x}_N)]$. In the feature space, PCA operates over the covariance matrix of the feature vectors defined as

$$\mathbf{S}_F = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{\phi}(\mathbf{x}_i) - \overline{\boldsymbol{\phi}})(\boldsymbol{\phi}(\mathbf{x}_i) - \overline{\boldsymbol{\phi}})' \tag{1}$$

where $\overline{\boldsymbol{\phi}} = \sum_{i=1}^{N} \boldsymbol{\phi}(\mathbf{x}_i)/N$ is the sample mean in feature space. Equivalently (1) can be expressed as

$$\mathbf{S}_F = \boldsymbol{\Phi}\mathbf{H}\mathbf{H}'\boldsymbol{\Phi}' \tag{2}$$

where $\mathbf{H}$ is a centering matrix of the form

$$\mathbf{H} = \frac{1}{N^{1/2}}\left(I - \frac{1}{N}\mathbf{1}\mathbf{1}'\right)$$

Since the matrix $\boldsymbol{\Phi}\mathbf{H}\mathbf{H}'\boldsymbol{\Phi}'$ has the same nonzero eigenvalues as $\mathbf{H}'\boldsymbol{\Phi}'\boldsymbol{\Phi}\mathbf{H} = \mathbf{H}'\mathbf{K}\mathbf{H}$, the kernel trick makes KPCA work over $\mathbf{H}'\mathbf{K}\mathbf{H}$ instead of $\mathbf{S}_F$, then the explicit knowledge of the mapping function $\boldsymbol{\phi}(\cdot)$ is not longer necessary [6].

A kernel representation $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\phi}(\mathbf{x}_i) \cdot \boldsymbol{\phi}(\mathbf{x}_j) \rangle$ can be used to calculate the dot matrix $\mathbf{K} = \boldsymbol{\Phi}'\boldsymbol{\Phi}$, with entries $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. The existence of such a kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ is guaranteed by the Mercer's theorem [7]. One of

the most used kernels in the literature is the gaussian kernel or RBF kernel, which is defined as

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\frac{1}{\sigma^2}\|\mathbf{x}_i - \mathbf{x}_j\|^2\right) \tag{3}$$

where σ^2 controls the RBF width. Following the definition of PCA, KPCA is equivalent to solve the dual eigenvalue problem

$$\mathbf{H}'\mathbf{K}\mathbf{H}\mathbf{v}_i = \lambda_i \mathbf{v}_i \tag{4}$$

where $\mathbf{v}_i$ denotes the column vector with entries $\mathrm{v}_{1i}, \ldots, \mathrm{v}_{Ni}$, $\mathbf{V} = [\mathbf{v_1}, \ldots, \mathbf{v}_N]$ the complete set of eigenvectors of $\mathbf{K}$ and $\mathbf{\Lambda}$ the corresponding eigenvalue diagonal matrix such that $\lambda_N \geq \ldots \geq \lambda_1$. For the purpose principal of component extraction, it is needed to calculate the projections on the eigenvectors $\mathbf{V}^l$ where $l = 1, \ldots, q, \ldots, p, \ldots, N$, q the index of the first nonzero eigenvalue of $\mathbf{\Lambda}$ and p the dimension of the principal subspace. Let $\mathbf{x}$ be a test point, with image $\phi(\mathbf{x})$ in F, then the projection $P^l\phi(\mathbf{x})$ onto the subspace spanned by the first p eigenvectors is then

$$\begin{aligned} P^l\phi(\mathbf{x}) &= \sum_{i=1}^{N} \mathrm{v}_i^l \langle \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}) \rangle \\ &= \sum_{i=1}^{N} \mathrm{v}_i^l \widetilde{k}(\mathbf{x}_i, \mathbf{x}) \end{aligned} \tag{5}$$

and

$$\widetilde{k}(\mathbf{x}_i, \mathbf{x}) = \left(I - \frac{1}{N}\mathbf{1}\mathbf{1}'\right)\left(\mathbf{k}_\mathbf{x} - \frac{1}{N}\mathbf{K}\mathbf{1}\right)$$

where $\mathbf{k}_\mathbf{x} = [k(\mathbf{x}_1, \mathbf{x}), \ldots, k(\mathbf{x}_N, \mathbf{x})]'$. In summary, KPCA is equivalent to calculate PCA over the $\mathbf{H}'\mathbf{K}\mathbf{H}$ matrix with gaussian kernel matrix entries given by (3).

III. HIDDEN MARKOV MODELS

A hidden Markov model is basically a Markov chain where the output observation is a random variable generated according to an output probabilistic function associated with each state [8]. Formally, a hidden Markov model of N_s states is defined as

- $\mathbf{A} = \{a_{ij}\}$. A transition probability matrix where a_{ij} denotes the probability of taking a transition from state i to state j, i.e.

$$a_{ij} = P(s_n = j | s_{n-1} = i)$$

- $\mathbf{B} = \{b_i(k)\}$. An output probability matrix, where $b_i(k)$ is the probability of emitting symbol o_k when state i is entered. Let $\mathbf{X} = \{\mathbf{x}_n\}$ be the observed output of the HMM. The state sequence $S = s_1, s_2, \ldots, s_n, \ldots$ is not observed and $b_i(k)$ can be written as follows

$$b_i(k) = P(\mathbf{x}_n = o_k | s_n = i)$$

  If the observation does not come from a finite set, but from a continuous space, the discrete output distribution must be changed by a continuous output probability density function $b_i(\mathbf{x}_n)$. Among several alternatives, multivariate gaussian mixture density functions are employed.
- $\boldsymbol{\pi} = \{\pi_i\}$. A initial state distribution where

$$\pi_i = P(s_0 = i) \qquad 1 \leq i \leq Ns$$

For simplicity, the set of parameters is denoted as

$$\lambda = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$$

To train an HMM, Expectation-Maximization updates can be used in conjunction with the forward-backward algorithm [9]. In particular, when the observation model is a mixture of gaussians

$$p_\lambda(\mathbf{x}_n | s_n = j) = \sum_{m=1}^{M} c_{jm} \mathcal{N}(\mathbf{x}_n; \boldsymbol{\mu}_{jm}, \mathbf{S}_{jm}) \tag{6}$$

where M is the number of components in the mixture. It can be shown that the reestimation formulas are given as

$$\begin{aligned} \widehat{c}_{jk} &= \frac{\sum_{n=1}^{N} \gamma_n(j,k)}{\sum_{n=1}^{N}\sum_{k=1}^{M} \gamma_n(j,k)} \\ \widehat{\boldsymbol{\mu}}_{jk} &= \frac{\sum_{n=1}^{N} \gamma_n(j,k)\mathbf{x}_n}{\sum_{n=1}^{N} \gamma_n(j,k)} \\ \widehat{\mathbf{S}}_{jk} &= \frac{\sum_{n=1}^{N} \gamma_n(j,k)(\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_{jk})(\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_{jk})'}{\sum_{n=1}^{N} \gamma_n(j,k)} \end{aligned} \tag{7}$$

where $\gamma_n(j,k)$ is the probability of being in state j at instant n with the k-th mixture component accounting for $\mathbf{x}_n$. The term $\gamma_n(j,k)$ generalizes to $\gamma_n(j)$ in the case of a single mixture. To find the single best state sequence, we use the Viterbi algorithm [9].

IV. BUILDING KPCA MODELS THROUGH TIME

A natural link between HMM and KPCA arises from the fact that both, the observation model for HMM in (6) and the eigenvalue problem in KPCA requires the computation of a sample covariance matrix given by (1), but should be noted that the expression $\widehat{\mathbf{S}}$ in (7) represents a weighted version of $\mathbf{S}_F$, even though is preferable to obtain an expression where kernel and the weights $\gamma_n(j,k)$ belongs to different matrix representations in a similar way to (2). To do this, from (7) it can be shown that

$$\begin{aligned} \widehat{\mathbf{S}}_{jk} &= \sum_{n=1}^{N} r_n(j,k)(\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_{jk})(\mathbf{x}_n - \widehat{\boldsymbol{\mu}}_{jk})' \\ &= \mathbf{X}\mathbf{R}_{jk}\mathbf{R}_{jk}'\mathbf{X}' \\ &= \mathbf{R}_{jk}'\mathbf{X}'\mathbf{X}\mathbf{R}_{jk} \end{aligned} \tag{8}$$

where $\widehat{\mathbf{S}}_{jk}$ is the local weighted sample covariance matrix for the state j and the k-th mixture component. The terms in the right side are defined as

$$r_n(j,k) = \frac{\gamma_n(j,k)}{\sum_{n=1}^{N}\gamma_n(j,k)}$$
$$\widehat{\boldsymbol{\mu}}_{jk} = \sum_{n=1}^{N} r_n(j,k)\mathbf{x}_n$$
$$\mathbf{R}_{jk} = (\mathbf{I} - \mathbf{r}(j,k)\mathbf{1}')\mathbf{D}_{jk}$$

where $\mathbf{r}(j,k) = [r_1(j,k),\ldots,r_N(j,k)]'$ and $\mathbf{D}_{jk}$ is a diagonal matrix with entries $r_1(j,k)^{1/2},\ldots,r_N(j,k)^{1/2}$. The matrix $\mathbf{R}_{jk}$ can be seen as the responsibility matrix for the k-th mixture component and the state j.

Since $\mathbf{X}'\mathbf{X}$ is a centered sample covariance matrix, it is easy to see from here that the model for KPCA can be builded for each mixture component of each corresponding state in the HMM model, in such way that the structure of the KPCA models contains the information in time provided by the resulting $\mathbf{R}_{jk}$ matrix after the HMM training process.

Exploiting the similar structure of (2) and (8), to build KPCA model through time, the eigenvalue problem in (4) can still be solved but replacing $\mathbf{K}$ by $\mathbf{X}'\mathbf{X}$ from (8) as

$$\mathbf{R}'_{jk}\mathbf{K}\mathbf{R}_{jk}\mathbf{v}_i = \lambda_i\mathbf{v}_i$$

with solution $\mathbf{V}^l_{jk}$ and $\mathbf{\Lambda}_{jk}$. The corresponding projection $P^l\phi(\mathbf{x})$ from (5) onto the subspace spanned by the first p principal components for the mixture k-th in the state j is

$$P^l_{jk}\phi(\mathbf{x}) = \sum_{i=1}^{N} \mathrm{v}^l_i \widetilde{k}(\mathbf{x}_i,\mathbf{x})$$

where the p-dimensional reduced representation of $\mathbf{x}$ is $P_{jk}\phi(\mathbf{x}) = [P^1_{jk}\phi(\mathbf{x}),\ldots,P^p_{jk}\phi(\mathbf{x})]'$.

A. Application to Time Series Classification

We employ the M KPCA models $\mathbf{V}^l_{jk}$ calculated for each state, for the transformation of the time sequences that are supposed to be classified. To accomplish this task, we first use the Viterbi algorithm to find the most likely sequence of states accounting for the particular vector series. Application of the Viterbi algorithm is made using the simple hidden Markov model with gaussian observation density functions that was already used in the process of estimation of the KPCAs models. Once it has been identified which state explains better a particular observation, corresponding in this case to a set of features of a segmented signal window, we use the model of KPCA belonging to that state to transform this specific observation, assuming that the information contained in such segment is relevant to explain the signal behavior. This process is applied to all sequences involved in the recognition problem using the models of the class to which those time series belong to. In this way, transformation of time series is made in a supervised fashion and in a dynamic mode, as well.

At this stage, all time sequences involved in the recognition problem have been transformed using a KPCA model. Now on, these new time vector series are classified using any time series classifier. Again, a hidden Markov model could be employed.

V. Experimental Results

A. Databases

For all experiments two different databases are used. Database DB1 belongs to Universidad Nacional de Colombia, Manizales, Colombia and contains 80 cases of sustained vowel /a/, pronounced by 40 normal speech patients and 40 dysphonic speech patients. Sampling frequency for this database is 24 KHz. Database DB2 belongs to Universidad Politécnica de Madrid, Spain. It contains 160 samples of sustained vowel /a/ pronounced by 80 normal speech patients and 80 different pathological speech patients (nodules, polypus, oedemas, cysts, sulcus, carcinomas). For this database sampling frequency is 50 KHz.

B. Feature Extraction

Speech samples are windowed using frames of 30 milliseconds (ms) length with an overlapping of 20 ms. For each frame, 12 Mel-Frequency Cepstrum Coefficients (MFCC) and energy coefficient were extracted. First and second order deltas are also included so we get a final observation vector of 39 variables for each frame.

C. Classification

For the classification of the different sets of time series vectors (Original MFCC, MFCC transformed with KPCA and MFCC reduced with KPCA), we use hidden Markov models with gaussian observation densities (see equation (6)). HMM topologies are ergodic and different number of states for the Markov chain are examined. Experiments are done using databases DB1 and DB2. The parameters of the model: number of states N_s, number of mixtures M and width of the gaussian kernel σ^2 were chosen for the less complex model[1] and with the best resulting accuracy, under a 5-fold crossvalidation scheme. More detailed classification results may be found in [10].

D. Experiment 1. Accuracies using the complete space

Two different vector time series are used as features for training, namely, the original MFCC vectors and the transformed MFCC vectors using KPCA model. All transformations are made using the method explained before (see section IV-A). For this case, the dimension of all time series is 39, the original dimensionality of the multivariate MFCC coefficients. Accuracies for both databases are shown in table I and II.

TABLE I
ACCURACY RESULTS FOR DATABASE DB1

Dataset	N_s	M	σ^2	Accuracy %
Original	3	5	-	90.00 ± 9.88
KPCA	3	1	10	$\mathbf{100.00 \pm 0.00}$

[1]In this context less complex means, the model with less states and mixtures without losing accuracy.

Results show that the HMM classifier trained with the transformed MFCC vectors using KPCA outperforms the one trained with the original raw dataset even in terms of model complexity.

TABLE II
ACCURACY RESULTS FOR DATABASE DB2

Dataset	N_s	M	σ^2	Accuracy %
Original	3	5	-	70.63 ± 4.74
KPCA	3	1	5	$\mathbf{100.00 \pm 0.00}$

E. Experiment 2. Performance for Reduced Dimensionality

We evaluate classification performance of time series vectors with reduced dimension using the transformed and reduced MFCC vectors using KPCA model. Different values for p in equation (5) are used, even for $p > 39$, i.e. taking dimensionality values greater than the original. Obtained results for both databases are shown in tables III and IV. For this experiment, the number of mixtures was manually fixed to 1.

TABLE III
RESULTS FOR DATABASE DB1 USING REDUCED TIME SERIES VECTORS

p	N_s	σ^2	Accuracy %
5	5	15	96.00 ± 3.42
15	3	10	$\mathbf{100.00 \pm 0.00}$
30	3	10	100.00 ± 0.00
33	3	5	100.00 ± 0.00
36	3	5	100.00 ± 0.00
42	3	5	100.00 ± 0.00
45	3	5	100.00 ± 0.00
48	3	10	100.00 ± 0.00

TABLE IV
RESULTS FOR DATABASE DB2 USING REDUCED TIME SERIES VECTORS

p	N_s	σ^2	Accuracy %
5	3	15	95.83 ± 4.77
15	3	15	$\mathbf{100.00 \pm 0.00}$
30	3	5	100.00 ± 0.00
33	3	5	100.00 ± 0.00
36	3	15	100.00 ± 0.00
42	3	5	100.00 ± 0.00
45	3	15	100.00 ± 0.00
48	3	5	100.00 ± 0.00

The results in both table III and IV shows that for just 15 of the 39 dimensions in the original dataset, the trained classifier has accuracies of 100% using the transformed and reduced MFCC vectors using KPCA model. Besides, this is not in any way a coincidence since for selected dimensionality between 15 and 48, the accuracy rates remains the same even with the same number of states. From table III is noticeable the fact that for $p = 5$, the classifier has required more states $N_s = 5$ to maintain the accuracy percentage high enough.

VI. CONCLUSIONS

Classification results show that the proposed methodology greatly improves the performance of the hidden Markov model classifier. In particular, transforming time series vectors using a kernel principal component analyzer dependent of time, allows to obtain extremely high accuracies with low variance. Even with few components in the transformed multivariate sequences, it is still posible to discriminate between both classes in each database. Novelty of the method consists mainly in the way in which a static dimensionality reduction technique has been combined with a classic mixture model in time, namely, the hidden Markov model, to enhance capabilities of transformation, reduction and classification of time voice disorder data.

A difficult question that must be solved remains in the model selection problem. First, the number of states in the Markov chain that better explains dynamic behavior must be carefully chosen. Second, a common problem with kernel methods is related with the election of a suitable kernel. Even when a kernel is chosen it still remains the issue of parameter selection, which it is well known to be a difficult matter. We have examined some alternatives for this parameter and validated them using computational expensive cross-validation techniques. It is not clear how the parameters of the kernel could be obtained in a simple way. For the choice of the reduced dimensionality of time series vectors, perhaps applying sparse-oriented principal component analyzers should be a direction.

VII. ACKNOWLEDGEMENTS

Authors would like to thank to CIE (Centro de Investigaciones y Extensión, Universidad Tecnológica de Pereira) for partially support this project and reviewer's comments.

REFERENCES

[1] Jolliffe I. T. , *Principal Component Analysis*, Springer Verlag, Second Edition, ISBN 0-387-98950-1, 2002.
[2] Voegtlin T., Recursive Principal Components Analysis, *Neural Networks*, vol. 18(8), 2005, pp. 1040-1050.
[3] Ku W. and Storer R.H. and Georgakis C., Disturbance Detection and Isolation by Dynamic Principal Component Analysis, *Chemometrics and Intelligent Laboratory Systems*, vol. 30(1), 1995, pp. 179-196.
[4] Jolliffe I. T. , "Principal Component Analysis for Time Series and other non Independent Data", *in Principal Component Analysis*, Springer Verlag, Second Edition, ISBN 0-387-95442-2, 2002, pp. 299-337.
[5] Shumway R.H. and Stoffer D.S., "Statistical Methods in Frequency Domain", *in Time Series Analysis and Its Applications*, Springer Verlag, Second Edition, ISBN 0-387-98950-1, 2005, pp. 465-483.
[6] Schölkopf B. and Smola A. and Müller K-R., Nonlinear Component Analysis as a Kernel Eigenvalue Problem, *Neural Computation*, vol. 10(5), 1998, pp. 1299-1319.
[7] Schölkopf B. and Smola A., *Learning with Kernels Support Vector Machines, Regularization, Optimization and Beyond*, MIT Press, ISBN 0-387-94559-8, 2002.
[8] Huang X. and Acero A. and Hon H-W, *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*, Prentice Hall, ISBN 0-130-22616-5, 2001.
[9] Rabiner L. R. , A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *Proceedings of The IEEE*, vol. 77(2), 1989.
[10] Alvarez M. and Henao R., PCA for Time Series Classification - Supplementary Material, UTP, 2006. Available at http://ohm.utp.edu.co/ rhenao/adminsite/elements/files/kpcatt_sm.pdf.