Probabilistic Kernel Principal Component Analysis Through Time

Mauricio Alvarez[1] and Ricardo Henao[2]

[1] Auxiliar Lecturer, Program of Electrical Engineering
[2] Auxiliar Lecturer, School of Electrical Technology
Universidad Tecnológica de Pereira, Colombia

Abstract. This paper introduces a temporal version of Probabilistic Kernel Principal Component Analysis by using a hidden Markov model in order to obtain optimized representations of observed data through time. Recently introduced, Probabilistic Kernel Principal Component Analysis overcomes the two main disadvantages of standard Principal Component Analysis, namely, absence of probability density model and lack of high-order statistical information due to its linear structure. We extend this probabilistic approach of KPCA to mixture models in time, to enhance the capabilities of transformation and reduction of time series vectors. Results over voice disorder databases show improvements in classification accuracies even with highly reduced representations.

1 Introduction

Principal Component Analysis (PCA) is a popular and powerful technique for feature extraction, dimensionality reduction and probably the most employed 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.

However, PCA has several disadvantages, among them, the absence of an associated probability density or generative model, the fact that the subspace itself is restricted to a linear mapping, where high-order statistical information is discarded and the assumption that observed data is independent, when modeling time series. The first two disadvantages were overcame by Probabilistic Principal Component Analysis (PPCA) [2] and Kernel Principal Component Analysis (KPCA) [3], respectively. Probabilistic Kernel Principal Component Analysis (PKPCA) has been proposed, as well, to deal with the first two disadvantages at the same time [4,5,6].

To overcome the third disadvantage, temporal versions for PKPCA and PPCA are introduced in this paper, 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 PCA variants was done by incorporating a factorization of the training

I. King et al. (Eds.): ICONIP 2006, Part I, LNCS 4232, pp. 747–754, 2006.

data and the responsibility weights in the covariance of the observation model in the HMM in such way that PPCA and PKPCA can be readily calculated.

The paper is organized as follows: sections 2 and 3 contains reviews of PKPCA and HMM respectively. In section 4 extensions for PPCA and PKPCA through time are presented. In section 5 experimental results for voice disorder databases are reported and finally in section 6 the conclusions are given.

2 Probabilistic Kernel Principal Component Analysis

PCA is intended to 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 f 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 $\mathbf{S}_F = \boldsymbol{\Phi}\mathbf{H}\mathbf{H}'\boldsymbol{\Phi}'$ where $\mathbf{H} = (I - 1/N\mathbf{1}\mathbf{1}')/N^{1/2}$. 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 [3]. 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 $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. Existence of such a kernel function is guaranteed by the Mercer's theorem [7]. One of the most popular kernels is the gaussian RBF kernel, defined as $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma_K^2\right)$ where σ_K^2 controls the RBF width.

In order to extend KPCA to its probabilistic version (PKPCA) following the Factor Analysis (FA) perspective [8], the feature dimension vector $\boldsymbol{\phi}(\mathbf{x}) \in \mathbb{R}^N$ can be expressed as a linear combination of base vectors and noise

$$\boldsymbol{\phi}(\mathbf{x}) = \mathbf{W}_F\mathbf{y} + \boldsymbol{\mu}_F + \boldsymbol{\epsilon}_F \tag{2}$$

where $\mathbf{y}$ is a p-dimensional data vector, $\mathbf{W}_F$ an $N \times p$ matrix relating the two set of variables $\boldsymbol{\phi}(\mathbf{x})$ and $\mathbf{y}$, and the $\boldsymbol{\mu}_F$ vector allows the model to have non-zero mean. The variance matrix $\boldsymbol{\Psi}$ of the noise process $\boldsymbol{\epsilon}_F$ should be isotropic $p(\boldsymbol{\epsilon}_F) \sim \mathcal{N}(0, N\sigma_F^2\mathbf{I}/f)$ and the latent variables $\mathbf{y}_1, ..., \mathbf{y}_p$ are independent gaussians of the form $p(\mathbf{y}) \sim \mathcal{N}(0, N\mathbf{I}/f)$. The difficulty is that f is usually unknown. However, it is possible to derive an estimation procedure for $\mathbf{W}_F$ and σ_F^2 that does not depend on f, using the kernel trick [6].

Since there is not closed-form analytic solution for the PKPCA model, the model should be obtained by iterative maximization of $\mathcal{L}_F$,

$$\mathcal{L}_F = \log(p(\boldsymbol{\Phi}|\mathbf{W}_F, \boldsymbol{\mu}_F, \sigma_F^2)) = -\frac{f}{2}\{\log|\mathbf{C}| + \frac{1}{N}\mathrm{tr}(\mathbf{C}_F^{-1}\mathbf{S}_F)\}$$

with some constant terms omitted, and where $\mathbf{C}_F = \mathbf{W}_F\mathbf{W}'_F + N\sigma_F^2\mathbf{I}/f$ and $\mathbf{S}_F$ from (1) is the covariance matrix of the feature vectors. The ML estimator for $\boldsymbol{\mu}_F$ is the sample mean in feature space, so $\boldsymbol{\mu}_{F(ML)} = \overline{\phi}$. The estimations for $\mathbf{W}_F$ and σ_F^2 can be obtained by iterative maximization of $\mathcal{L}_F$ in terms only of the kernel matrix $\mathbf{K}$ and not of the feature space dimension f, as in [6]

$$\widehat{\mathbf{W}}_\mathbf{F} = \mathbf{H}'\mathbf{K}\mathbf{H}\mathbf{W}_F(N\sigma_F^2\mathbf{I} + \mathbf{M}_F^{-1}\mathbf{W}'_F\mathbf{H}'\mathbf{K}\mathbf{H}\mathbf{W}_F)^{-1} \tag{3}$$

$$\widehat{\sigma}_F^2 = \frac{1}{N^2}\mathrm{tr}(\mathbf{H}'\mathbf{K}\mathbf{H} - \mathbf{H}'\mathbf{K}\mathbf{H}\mathbf{M}_F^{-1}\widehat{\mathbf{W}}'_F) \tag{4}$$

where $\widehat{\mathbf{W}}_F$ means "new" and $\mathbf{M}_F = \mathbf{W}'_F\mathbf{W}_F + \sigma_F^2\mathbf{I}$. Then, the equations in (3) and (4) are iterated in sequence until the algorithm is judge to have converged.

In order to obtain standard PPCA [2] from the above formulation, the model in (2) should be modified making $\boldsymbol{\Phi} = \mathbf{x}$. Besides, $\mathbf{H}'\mathbf{K}\mathbf{H}$ should be replaced by $\mathbf{S} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})'$, the covariance matrix of the original observation data space, where $\overline{\mathbf{x}} = \sum_n \mathbf{x}_n/N$. It can be shown [2] that for standard PPCA, the estimations for $\mathbf{W}$ and σ^2 can be obtained similar to (3) and (4) as

$$\widehat{\mathbf{W}} = \mathbf{S}\mathbf{W}(\sigma^2\mathbf{I} + \mathbf{M}^{-1}\mathbf{W}'\mathbf{S}\mathbf{W})^{-1} \qquad \widehat{\sigma}^2 = \frac{1}{d}\mathrm{tr}(\mathbf{S} - \mathbf{S}\mathbf{M}^{-1}\widehat{\mathbf{W}}') \tag{5}$$

Notice that we have removed the subindex of the feature space for clarity.

3 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 [9]. A hidden Markov model of N_s states is defined by a set of parameters $\lambda = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$, where $\boldsymbol{\pi}$ is the initial state distribution, $\mathbf{A}$ is the transition probability matrix and $\mathbf{B}$ is the output probability matrix. Output probabilities can be modeled through continuous probability density functions.

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

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

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

$$\widehat{c}_{jk} = \frac{\sum_{n=1}^{N}\gamma_n(j,k)}{\sum_{n=1}^{T}\sum_{k=1}^{M}\gamma_n(j,k)} \qquad \widehat{\boldsymbol{\mu}}_{jk} = \frac{\sum_{n=1}^{N}\gamma_n(j,k)\mathbf{x}_n}{\sum_{n=1}^{N}\gamma_n(j,k)}$$

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

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

4 Building PCA Models Through Time

A natural link between HMM and PCA arises from the fact that both, the observation model for HMM in (6) and the formulation of PCA requires the computation of a sample covariance matrix given by (1). It 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 the observed data (or kernel) and the weights $\gamma_n(j,k)$ belongs to different matrix representations in a similar way to $\mathbf{H}'\mathbf{K}\mathbf{H}$. To do this, from (7) it can be shown that

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

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 left side are defined as

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

where the vector $\mathbf{r}(j,k) = [r_1(j,k), \ldots, r_N(j,k)]'$ 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 state j.

Since $\mathbf{X}\mathbf{R}_{jk}\mathbf{R}'_{jk}\mathbf{X}' = \mathbf{R}'_{jk}\mathbf{X}'\mathbf{X}\mathbf{R}_{jk}$ in (8) and $\mathbf{X}'\mathbf{X}$ is a centered sample covariance matrix, it is easy to see from here that models for PPCA and PKPCA can be builded for each mixture component of each corresponding state in the HMM model. Then the structure of the PCA models contain the information in time provided by the resulting $\mathbf{R}_{jk}$ matrix after the HMM training process.

4.1 PKPCA Model Through Time

The model for PKPCA can be builded performing iterative evaluations of $\widehat{\mathbf{W}}_{Fjk}$ and $\widehat{\sigma}^2_{Fjk}$ using (3) and (4). Besides, the term $\mathbf{H}'\mathbf{K}\mathbf{H}$ should be replaced by $\mathbf{R}'_{jk}\mathbf{K}\mathbf{R}_{jk}$ in order to conform the mixture model. The p-dimensional reduced representation of $\mathbf{x}$ corresponding to the k-th mixture in the state j can be obtained as

$$\mathbf{y}_{jk} = \mathbf{M}^{-1}_{Fjk}\mathbf{W}'_{Fjk}\widetilde{k}(\mathbf{x}_i, \mathbf{x}) \tag{9}$$

where $\widetilde{k}(\mathbf{x}_i,\mathbf{x}) = (\mathbf{I} - \mathbf{r}(j,k)\mathbf{1}')\,(\mathbf{k}_\mathbf{x} - \mathbf{K}\mathbf{r}(j,k))$ and $\mathbf{k}_\mathbf{x} = [k(\mathbf{x}_1,\mathbf{x}), \ldots, k(\mathbf{x}_N,\mathbf{x})]'$.

Additionally, to build the PPCA model through time, iterative evaluations of $\widehat{\mathbf{W}}_{jk}$ and $\widehat{\sigma}^2_{jk}$ should be calculated by using (5), but replacing $\mathbf{S}$ by $\widehat{\mathbf{S}}_{jk}$ from (8). The p-dimensional reduced representation of the observed vector $\mathbf{x}$ is calculated using $\mathbf{y}_{jk} = \mathbf{M}^{-1}_{jk}\mathbf{W}'_{jk}(\mathbf{x} - \widehat{\mu}_{jk})$.

4.2 Time Series Classification

Application of the described methodology for PKPCA can be summarized in algorithm 1. A similar algorithm for PPCA can be applied to time series classification using the corresponding equations.

Algorithm 1. PKPCA through time Algorithm

Require: Training observation $\mathbf{X}$, N_s, M and σ_K
1: Calculate $\mathbf{K}$
2: Initialize PKPCA models ($\widehat{\mathbf{W}}_{Fjk}$, $\widehat{\sigma}^2_{Fjk}$) using PCA
3: Initialize one HMM per class with gaussian observation models
4: % Train HMM models
5: **repeat**
6: Reestimate HMM model parameters λ using EM updates
7: Reestimate $\widehat{\mathbf{W}}_{Fjk}$ and $\widehat{\sigma}^2_{Fjk}$ using $\widehat{\mathbf{S}}_{jk}$ (equations (3) and (4))
8: **until** convergence
9: % Obtain the transformed and/or reduced representation
10: path=VITERBI($\mathbf{X}$,λ) % path is the best state sequence
11: Project every $\mathbf{x}_i \in \mathbf{X}$ using the PKPCA model given by path_i (equation (9))
12: Train HMM based classifier λ_C using EM algorithm and transformed observations
13: **return** $\widehat{\mathbf{W}}_{Fjk}$, $\widehat{\sigma}^2_{Fjk}$ and λ % PKPCA through time parameters
14: **return** λ_C % HMM classifier

In order to test unseen observations, $\widehat{\mathbf{W}}_{Fjk}$, $\widehat{\sigma}^2_{Fjk}$ and λ can be employed to obtain the transformed and/or reduced representation and λ_C to classify them.

5 Experimental Results

5.1 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. 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).

5.2 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.

5.3 Classification

For the classification of the different sets of vector time series (Original MFCC, MFCC transformed with PKPCA and MFCC transformed with PPCA), we use hidden Markov models with gaussian observation densities (see equation (6)). HMM topologies are ergodic [9] 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_K^2 are shown for the less complex model[1] with the best resulting accuracy, under a 5-fold crossvalidation scheme. More detailed classification results may be found in [10].

5.4 Experiment 1. Accuracies Using the Complete Space

Three different time series vectors are used as features for training, namely, the original MFCC vectors, the transformed MFCC vectors using PKPCA models and the transformed MFCC vectors using PPCA model. All transformations are made using the method explained before (see section 4.2). 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 1.

For database DB1, results show that the HMM classifier trained with the transformed MFCC vectors using PKPCA and PPCA outperforms the one trained with the original raw dataset even in terms of model complexity. For models with three states, PKPCA and PPCA performances do not seem to be different, although for more states PPCA degrades its performance while PKPCA keeps its high recognition rate. On the other hand, for database DB2, it can be seen that again transformed MFCC vectors using PKPCA and PPCA outperforms the one trained with the original raw dataset even in terms of model complexity. PKPCA shows better results over PPCA, for which a high standard deviation in accuracy is obtained.

Table 1. Accuracy results for Databases DB1 and DB2

	DB1				**DB2**			
Dataset	N_s	M	σ_K^2	Accuracy %	N_s	M	σ_K^2	Accuracy %
Original	3	5	-	90.00 ± 9.88	3	5	-	70.63 ± 4.74
PPCA	3	1	-	$\mathbf{100.00 \pm 0.00}$	10	1	-	91.25 ± 7.46
PKPCA	3	1	5	$\mathbf{100.00 \pm 0.00}$	3	1	5	$\mathbf{100.00 \pm 0.00}$

5.5 Experiment 2. Performance for Reduced Dimensionality

We evaluate classification performance of time series vectors using the transformed and reduced MFCC vectors with PKPCA and PPCA models. Different

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

values for p are used, even for $p > 39$, i.e. taking dimensionality values greater than the original. Obtained results for both databases are shown in table 2. In this experiment, the number of mixtures was manually fixed to 1.

Table 2. Results for Databases DB1 and DB2 using reduced time series vectors

		DB1			DB2		
p	Dataset	N_s	σ_K^2	Accuracy %	N_s	σ_K^2	Accuracy %
5	PKPCA	3	15	96.25 ± 5.59	3	15	82.50 ± 3.56
15	PKPCA	3	15	$\mathbf{100.00 \pm 0.00}$	3	15	98.75 ± 1.71
30	PPCA	5	-	96.25 ± 3.42	5	-	87.50 ± 3.12
	PKPCA	3	5	100.00 ± 0.00	3	5	$\mathbf{100.00 \pm 0.00}$
33	PPCA	3	-	100.00 ± 0.00	5	-	93.12 ± 9.21
	PKPCA	3	5	100.00 ± 0.00	3	5	100.00 ± 0.00
36	PPCA	3	-	98.75 ± 2.80	5	-	90.62 ± 6.62
	PKPCA	3	5	100.00 ± 0.00	3	15	100.00 ± 0.00
42	PKPCA	3	5	100.00 ± 0.00	3	5	100.00 ± 0.00
45	PKPCA	3	5	100.00 ± 0.00	3	15	100.00 ± 0.00
48	PKPCA	3	5	100.00 ± 0.00	3	5	100.00 ± 0.00

The results in table 2 show that for just 15 of the 39 dimensions in DB1 and 30 of the 39 in DB2, the trained classifier has accuracies of 100% using the transformed and reduced MFCC vectors with the PKPCA model. Besides, this is not in any way a coincidence since for selected dimensionalities between 15 and 48, the accuracy rates remains the same even with the same number of states. Also from table 2 is noticeable the fact that for PPCA model more states are needed to achieve satisfactory results, while in PKPCA, with just 3 states, better accuracies are obtained. Intuitively, PKPCA model requires a less complex structure than PPCA model to explain the current data. For database DB2, accuracies using PPCA have large standard deviations mainly due to the nature of samples.

6 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 probabilistic 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.

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.

Finally, straightforward extensions of the proposed methodology could be formulated for latent variable models relying on covariance like matrices.

Acknowledgements

Authors would like to thank to CIE (Centro de Investigaciones y Extensión) and Faculty of Technology of Universidad Tecnológica de Pereira for partially support this research project.

References

1. Jolliffe, I.: Principal Component Analysis. Second edn. Springer Verlag (2002)
2. Tipping, M., Bishop, C.: Probabilistic principal component analysis. Journal of the Royal Statistical Society, Series B **21**(3) (1999) 611–622
3. Schölkopf, B., Smola, A., Müller, K.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation **10**(5) (1998) 1299–1319
4. Tipping, M.: Sparse kernel principal component analysis. In Press, M., ed.: Neural Information Processing Systems, NIPS'00. (2000) 633–639
5. Zhou, C.: Probabilistic analysis of kernel principal components: mixture modeling and classification. Cfar technical report, car-tr-993, University of Maryland, Department of Electrical and Computer Engineering, Maryland (2003)
6. Zhang, Z., Wang, G., Yeung, D., Kwok, J.: Probabilistic kernel principal component analysis. Technical report hkust-cs04-03, The Hong Kong University of Science and Technology, Department of Computer Science, Hong Kong (2004)
7. Schölkopf, B., Smola, A.: Learning with Kernels Support Vector Machines, Regularization, Optimization and Beyond. MIT Press, Cambridge, MA (2002)
8. Härdle, W., Simar, L.: Applied Multivariate Statistical Analysis. Springer, N.Y. (2003)
9. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of The IEEE **77**(2) (1989)
10. Alvarez, M., Henao, R.: PCA for time series classification - supplementary material. Technical report, Universidad Tecnológica de Pereira, Pereira, Colombia, Available at `http://ohm.utp.edu.co/~rhenao/adminsite/elements/files/kpcatt_sm.pdf`. (2006)