Proc. IJCNN 2003, Portland, USA, pp - .

An Accurate and Fast Neural Method for PCA Extraction

J. B. O. Souza Filho, L. P. Calôba, J. M. Seixas

COPPE & EP - LPS, Signal Processing Lab
UFRJ, Federal University of Rio de Janeiro
CP 68504, Rio de Janeiro 21945-970, Brazil
e-mail: [nash,caloba,seixas]@lps.ufrj.br

Abstract - **Principal Component Analysis (PCA) is a characteristic extraction method, whose main objective function is the reconstruction of the original data space. PCA is a linear optimal method, in the sense of mean squared error, and is applied in a wide variety of knowledge areas. In this paper, a new neural method for PCA extraction is proposed and compared, in terms of accuracy and computational costs, to other well accepted neural extraction methods, such as GHA and APEX. The performance comparison was evaluated using preprocessed spectra from passive sonar signals. It was verified that the proposed method performed better than all other methods, exhibiting easier implementation, lower computational costs and higher accuracy.**

I. INTRODUCTION

In a considerable number of knowledge areas, the manipulation of large dates sets is necessary. So, relevant characteristic extraction is mandatory, and it is dependent on the type of information searched for. Basically, characteristic extraction may have the objective of efficient reconstruction of data or pattern classification. For each objective, different methods and heuristics are necessary [1].

Principal Component Analysis is a characteristic extraction method that is typically used for achieving the reduction of the dimensionality of the input data space. PCA is an optimum linear method, under the criterion of minimization of the mean quadratic error in data reconstruction [2]. This methodology is applied in various fields of the knowledge, among them: data analysis, image processing and codification, the solution of great equations systems, patterns detection and recognition, spectral analysis, signal processing of antennas arrays, etc. [3].

In this work, a new neural method for PCA extraction is presented and its performance is evaluated using a set of spectra of real passive sonar signals. The method is also compared to others PCA neural extraction methods, considering accuracy and computational costs.

The structure of this paper is the following: PCA and neural methods for component extraction are briefly presented. In this context we introduce the new proposed method, and deduce the refered training equations. In the sequence, the data used to evaluate the algorithm performance is described and tests are conducted. Conclusions are then derived in the last section.

II. PRINCIPAL COMPONENTS ANALYSIS

The principal component analysis is based on the *Karhunen-Loëve series expansion* [4]. In a N-dimensional data space, PCA aims at finding a M-orthogonal vector set (principal directions) whose data projection variance is maximized. For zero mean processes, variance is equal to energy, so PCA vectors are directions for which energy concentration is maximum. Typically, $M << N$, so that PCA can ve considered as an optimal linear method for data reconstruction, providing M directions for which the lost of information is minimal, when mean-square error is considered as a figure of merit.

The most used PCA extraction methods may be grouped in two categories: classic and neural. In the first group, it is usually necessary to compute the covariance matrix and to extract its eigenvalues and eigenvectors [5]. Neural methods make use of linear neural networks whose training method might be supervised or not, the latter being more frequently discussed in the literature [2].

Generally, classical methods are computing demanding, the algorithms are complex and make use of large amount of memory. They form an algorithm set which use is more convenient when most of components of a stochastic process should be extracted, especially in offline data analysis. For online systems, or when only a few components are required, neural based methods tend to be the best solution.

Neural methods are adaptative algorithms, so a better trade-off between accuracy and computational cost can be established. Considering these methods, the computational cost for methods having similar complexity is usually measured by the number of training steps. Another important advantage of neural methods is the algorithm's simplicity. Since they use few additions and multiplications, their implementation on multiple platforms can be easily realized, especially when memory restrictions apply, because the correlation matrix needs to be computed or stored during the training process.

III. NEURAL PCA METHODS

The most popular unsupervised neural methods for PCA extraction are GHA [6], APEX [7] and PAST [8]. Among supervised methods, there is the constructive auto-associative neural network, here referred to as MLP-I architecture, which was used in [9] for dimensionality reduction of passive sonar signals spectra.

A. Generalized Hebbian Algorithm (GHA)

Oja has shown that a single linear neuron submitted to a Hebbian training rule finds the first principal component [10]. Sanger generalized this model, developing a linear network for the extraction of an arbitrary number of components (p) [6], whose training equations are:

$$\Delta w_{ij}(n) = \eta . y_i(n) . \left[x_j(n) - \sum_{k=1}^{i} w_{kj}(n) . y_k(n) \right], \quad \begin{cases} i = 1 \ldots p, \\ j = 1 \ldots N, \end{cases} \tag{1}$$

where:

$$y_j(n) = \sum_{k=1}^{N} w_{jk}(n) . x_k(n), \tag{2}$$

for a N-dimensional input data space, being $\Delta w_{ij}(n)$ the weights-update at n-th iteration and η the learning rate.

B.Adaptive Principal Components Extraction (APEX)

APEX method was proposed by Kung and Diamantaras [7] and, for the extraction of p components, makes use of p linear neurons that, in addition to feed-forward weights ($\mathbf{w}_m$), have lateral connections established by inhibition weights ($\mathbf{c}_m$). The training equations are:

$$\begin{aligned} \Delta \mathbf{w}_m(n) &= \eta(n) . y_m(n) . [\mathbf{x}(n) - y_m(n) . \mathbf{w}_m(n)], \\ \Delta \mathbf{c}_m(n) &= \eta(n) . y_m(n) . [\mathbf{y}(n) - y_m(n) . \mathbf{c}_m(n)], \qquad m = 1..p, \end{aligned} \tag{3-4}$$

where:

$$\mathbf{y}(n) = [\mathbf{w}_1 \quad \cdots \quad \mathbf{w}_{m-1}]^T . \mathbf{x}(n) \tag{5}$$

$$y_m(n) = \mathbf{w}_m^T . \mathbf{x}(n) - \mathbf{c_m}(n)^T . \mathbf{y}(n) \tag{6}$$

The learning rate factor (η) can be determined, iteratively, by the following equation:

$$\eta(n) = \frac{\eta(n-1)}{\beta + y_m(n)^2 . \eta(n-1)}, \qquad 0 < \beta \le 1, \tag{7}$$

where β is the forgetting factor. According to [7], the optimal choice of step size parameter has a profound impact on the convergence speed of the algorithm.

C. Projection Approximation Subspace Tracking - PAST

Proposed, independently, by [8] and [11], this algorithm produces weight update equations identical to (1), except for the learning rate factor which is determined iteratively by an equation equivalent to (7). Since η is optimally determined for each training step, the convergence behavior of PAST, in sense of speed and accuracy, is much better than GHA.

D. Auto-associative Multi-Layer Perceptron I – MLP-I

According to this method, in order to extract p components from a N-dimensional data space, the extraction process starts with a linear neural network with N input nodes, one neuron at the hidden layer and N neurons at the output layer, which are trained to reproduce the input patterns (auto-association) using the back-propagation algorithm. After convergence, one more hidden neuron is introduced at the network, and the training process is restarted, keeping the weights refered to the previous training cycle unchanged. This process is repeated until the hidden layer has *p* trained neurons.

Some important characteristics of this network are [12]:

1) The principal components are sequentially extracted.
2) Two estimates of each component are available: one by the weights that connect the input nodes to the hidden layer ($\underline{\mathbf{w}}$), here referred as projection weights, and other by the weights that connect the hidden layer to output layer ($\overline{\mathbf{w}}$), here referred to as reconstruction weights.

The convergence behavior of this network was analyzed in [12]. It was experimentally shown that the convergence of the reconstruction weights is faster than the projection weights. Another relevant feature is that the estimates provided by the reconstruction weights are more accurate.

An important fact emerges after some manipulations on the error equations for this network. For an input pattern **x**, the error equation related to *k*-th (*k*=1..*p*) neuron is given by:

$$\mathbf{e}_k(n) = \left[\mathbf{x}(n) - \sum_{i=1}^{k} \overline{\mathbf{w}}_{\mathbf{i}}(n).\underline{\mathbf{w}}_{\mathbf{i}}^{\mathbf{T}}(n).\mathbf{x}(n)\right] \tag{8}$$

Another form to express (8) is:

$$\mathbf{e}_k(n) = \left[\mathbf{t}_{\mathbf{k}}(n) - \overline{\mathbf{w}}_{\mathbf{k}}(n).\underline{\mathbf{w}}_{\mathbf{k}}^{\mathbf{T}}(n).\mathbf{x}(n)\right], \tag{9}$$

where:

$$\mathbf{t}_k(n) = \mathbf{x}(n) - \sum_{i=1}^{k-1} \overline{\mathbf{w}}_i(n).\underline{\mathbf{w}}_{\mathbf{i}}^{\mathbf{T}}(n).\mathbf{x}(n) \tag{10}$$

Observing (9), we conclude that MLP-I network is similar to *p* linear networks with *N* input nodes, one neuron at hidden layer and *N* neurons in the output layer, having a common input vector **x** and different target vectors given by (10). This architecture will be referred to here as MULTI-NET.

E. Auto-associative Multi-Layer Perceptron II – MLP-II

Motivated by the drawbacks of the MLP-I method, and considering the equivalence shown by (8), we propose a new training method where the projection and reconstruction weights are made equal for each training step.

Considering the *i*-th (*i*=1..*p*) linear neural network of MULTI-NET, the *j*-th element of output layer vector is given by:

$$z_j^i = \overline{w}_{ji}.\sum_{k=1}^{N} \underline{w}_{ik}.x_k \tag{11}$$

Making $w_{ij} = \overline{w}_{ji} = \underline{w}_{ij}$, the sensitivity of z_j^i to w_{ik} is:

$$\frac{\partial z_j^i}{\partial w_{ik}} = \delta(j-k).\sum_{l=1}^{N} x_l.w_{il} + w_{ij}.x_k, \tag{12}$$

where: $\delta(j-k) = 1$ if $j = k$ else 0.

The square-error for the input pattern **x** is given by:

$$eq^i = \sum_{l=1}^{N}\left(t_l^i - x_l\right)^2 = \sum_{l=1}^{N} \varepsilon_l^2 \tag{13}$$

The gradient for the squared-error function is:

$$\frac{\partial eq^i}{\partial w_{ik}} = -2.\sum_{l=1}^{N} \varepsilon_l.\frac{\partial z_j^i}{\partial w_{ik}} \tag{14}$$

Using the stochastic gradient for weight updates results:

$$\Delta w_{ij} = \eta.x_j.\sum_{l=1}^{N} \varepsilon_l.w_{ij} + \eta.\varepsilon_j.\sum_{l=1}^{N} x_l.w_{il} \tag{15}$$

According to what was observed by [13], the first term in (15) can be considered less important than the second one. Thus, (15) can be approximated by

$$\Delta w_{ij} = \eta.\left(t_j^i - x_j\right).y_i \tag{16}$$

If we apply (10) in (16) results:

$$\Delta w_{ij}(n) = \eta.y_i(n).\left[x_j(n) - \sum_{k=1}^{i} w_{kj}(n).y_k(n)\right], \quad \begin{cases} i = 1...p, \\ j = 1...N, \end{cases} \tag{17}$$

which is the same as (1). We conclude that GHA is an approximation of the proposed method. Therefore, a convenient choice for η in (15) is given by (7). This choice improves the convergence speed of MLP-II algorithm, as a sub-optimal learning factor is used at each training step.

IV. RESULTS

The dataset used to evaluate the proposed extraction method consists of pre-processed passive sonar spectra. These signals were obtained by the acquisition of radiated noise from ships belonging to four classes, considering different runs and machinery conditions, at an acoustic path with, approximately, 45 meters deep. Each run consisted in making a ship to pass over a hydrophone with constant speed and with a specific machinery condition. The sensor used was an omnidirection hydrophone near the ocean bottom. The signals were digitized by an 8-bit ADC (analog-to-digital converter) at a sampling rate of 22.05 kHz.

The digitalized signals were applied to a pre-processing system developed to implement an efficient and robust neural classifier based on the spectra of these signals. The pre-processing system adopted here is shown in Figure 1. Basically, this system obtains the signal spectra in the frequency-range of interest (0 - 2871 Hz), improving signal characteristics relevant to ships classification [14,15,16]. A Two-Pass Split Window (TPSW) algorithm [17] is used for background noise reduction and spectra normalization, improving classification efficiency significantly.

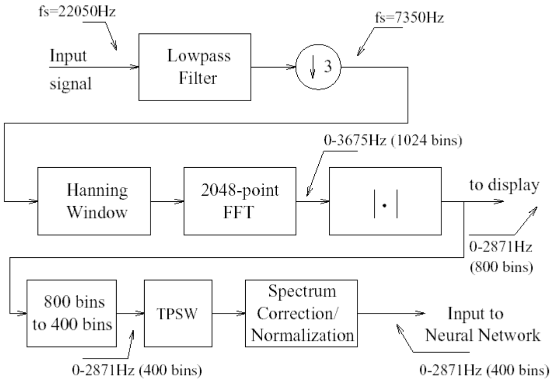

Fig 1. Block diagram of the signal preprocessing method performed on the incoming signal.

A total of 16 different runs, resulting in 6192 acquisition windows with 400 spectral samples were used to validate the proposed method. In terms of practical application, passive sonar signals provide complex high-dimensional data, so that such dataset may be considered as an appropriate set to develop the analysis of the convergence behavior and robustness of the proposed method for multiple components extraction in real world problems.

A. Experimental Tests

In order to evaluate the proposed method, we considered the computational cost and accuracy achieved for each extracted component. A reference for the analysis was established by means of an implementation of the package *eispack* [18], widely used in the scientific community. For this evaluation, the proposed method was compared to well accepted neural extraction methods: Sanger (GHA), APEX and MLP-I, all of them extracting serially the target components.

Along the networks training procedure, a component was considered extracted when, after a defined number of training steps, the changes on the value of the associated eigenvalue estimate were smaller than a specified value (μ). After some trials, appropriate values for learning rate (η), number of training steps for the evaluation of eigenvalue estimate (α), forgetting factor (β), and convergence value (μ) were chosen, as seen in Table I.

In accuracy test, the components extracted by each method were compared to the reference dataset by measuring the module of the angle, in degrees, between both of them. Despite the extraction of 400 components, just the first 31 components were accurately extracted by all methods and thus used for analysis. In Figure 2 and 3, the angles obtained for the first 31 components are presented. Table II summarizes the mean angle for each method considering the first 20 and 31 components.

TABLE I

EXTRACTION PARAMETERS FOR THE TRAINING SET

	GHA and MLP-I	APEX	MLP-II
η	1.0E-6	-	-
α	10	20	20
β	-	0.9999	0.99999
μ	5E-4	5E-4	2.5E-4

TABLE II

MEAN ERROR ANGLES

Method	Mean Angle (degrees)	
Components	1-20	1-31
MLP-I (Projection)	18.4	26.9
MLP-I (Reconstruction)	7.7	19.3
Sanger	4.6	20.1
APEX	8.2	20.5
Proposed Method	1.8	3.7

Observing Figures 2 and 3, it can be verified that the estimates provided by the proposed method are more accurate than the others, especially for components greater than 20th. According to these results, the methods could be ranked in terms of accuracy performance in the following order: Proposed Method, GHA, MLP-I (reconstruction weights), APEX and MLP-II (projection weights).

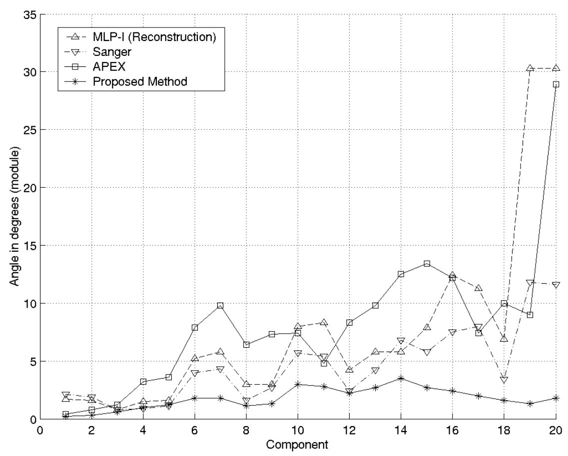

Fig 2. Error angles for each method (components 1 to 20) .

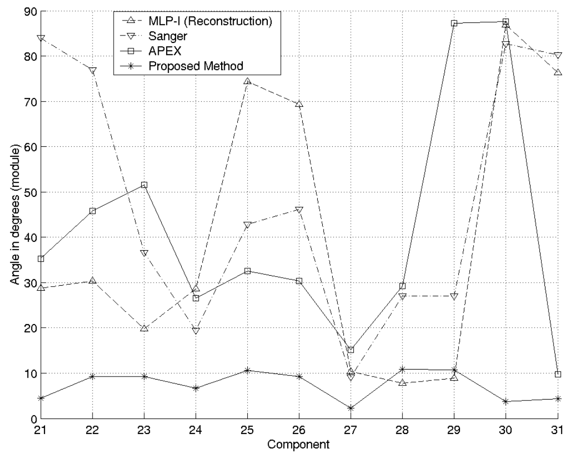

Fig 3. Error angles for each method (components 21 to 31) .

In order to evaluate the computational costs, we considered the mean value of epochs for each method. Note that both the proposed method and GHA achieves approximately the same computational cost per epoch, and this cost is much lower than for APEX and MLP-I. The results are summarized in Table III. As we can see, the computational cost of the proposed method is significantly smaller.

TABLE III

NUMBER OF MEAN EPOCHES FOR EACH METHOD

Method	Number of Epochs	
Components #	20	31
MLP-I	2762	1786
GHA	1087	1950
APEX	3343	3369
Proposed Method	764	922

V. SUMMARY

A new supervised neural method for principal components extraction was proposed and compared to other well-accepted neural methods concerning accuracy and computational cost. The evaluation was based on a real dataset from a passive sonar system application, which provided a large number of realizations and high dimensionality. It was shown that the proposed method has a better convergence behavior and lower computational cost than the other methods, becoming an interesting option among the multiple PCA extraction techniques.

VI. REFERENCES

[1] Lerner B. , Guterman H., Aladjem M, *et all*. "Feature extraction by neural network nonlinear mapping for pattern classification". *International Conference on Pattern Recognition*, 1996.

[2] Haykin S. *Neural Networks: a Comprehensive Foundation*, Prentice-Hall, 1999.

[3] Comon P, Golub G. "Tracking a few extreme singular values and vectors in signal processing". Proceedings of the IEEE, v. 78, n. 8, pp 1327-2343, August 1990.

[4] Papoulis A. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, 1991.

[5] Golub G., Loan C. *Matrix Computations*. The Johns Hopkins University Press, 1989.

[6] Sanger T. "Optimal unsupervised learning in a single linear feedforward network". *Neural Networks*, v. 5, n. 5, pp. 684-697, Setember 1994.

[7] Kung S., Diamantaras K. and Taur J. "Adaptative principal component extraction (APEX) and applications". *IEEE Transactions on Signal Processing*, v. 142, n.5, pp. 12502-1207, May 1994.

[8] Yang, B. "Projection Approximation Subspace Track ing". IEEE Transactions on Signal Processing, v. 43, n. 1, January 1995.

[9] W.Soares Filho, J.M Seixas and L.P Calôba. "Principal components analysis for classifying passive sonar signals". *International Symposium on Circuit and Systems - ISCAS*, Sidney, Australia, 2001.

[10] Oja E. "A simplified neuron model as a principal component analyser". *Journal of Mathematical Biology*, n. 15, pp 267-273, 1982.

[11] Bannour, S. and Azimi-Sadjadi, M. R. "Principal Components Extraction Using Recursive Least Square Learning". IEEE Transactions on Neural Networks, v. 6, n. 2, March 1995.

[12] Souza Filho J.B.O., Souza M., Calôba L.P., Seixas J.M. "Principal Component Extraction on High Dimesionality Applications". *V Braziliam Symposium on Smart Automation*, November 2000, Canela – RS, Brazil [In Portuguese].

[13] Karhunen, J. and Joutsensalo, J. "Representation and Separation of Signals Using Nonlinear PCA Type Learning". Neural Networks, v. 7, n. 1, pp 113-127, 1994.

[14] M Seixas, W.Soares Filho, J.B.O Souza Filho, D.O Damazio and N.N Moura. "A compact online neural system for classifying passive sonar signals". International Conference on Signal Processing, Applications & Technology - ICSPAT, Orlando, 1999.

[15] W.Soares Filho, J.M Seixas and L.P Calôba. "Averaging spectra to improve the classification of the noise radiated by ships using neural networks". *VIth Brazilian Symposium on Neural Networks –VI SBRN*, Rio de Janeiro, Brazil, 2000.

[16] J.B.O Souza Filho, J.M Seixas and W.Soares Filho. "Online Neural Classifier for passive sonar using a high performance digital signal processor". V Braziliam Congress on Neural Networks – V CBRN, Rio de Janeiro, Brazil, 2001 [In Portuguese].

[17] Nielsen, R.O. "Sonar Signal Processing". Artech house, 1991.

[18] Press W., Vatterling W., Teukolsy S., *et al*. *Numerical Recipies in C – The Art of Scientific Computing*. Cambrige University Press, 1995.