FAST PRINCIPAL COMPONENT EXTRACTION BY A HOMOGENEOUS NEURAL NETWORK

Shan Ouyang
Department of Communications and Information
Guilin University of Electronic Technology
Guilin 541004, Guangxi, P.R.China

Zheng Bao
Key Laboratory for Radar Signal Processing
Xidian University
Xi'an 710071, Shaanxi, P.R.China

ABSTRACT

On the basis of the concepts of both weighted subspace criterion and information maximization, this paper proposes a weighted information criterion (WINC) for searching for the optimal solution of a homogeneous neural network. We develop two adaptive algorithms based on the WINC for extracting in parallel multiple principal components. The both algorithms are be able to provide an adaptive step size which leads to a significant improvement in the learning performance. Furthermore, the recursive least squares (RLS) version of WINC algorithms has a low computational complexity $O(Np)$, where N is the input vector dimension and p is the number of desired principal components. Since the weighting matrix does not require an accurate value, it facilitates the system design of the WINC algorithm for real applications. Simulation results are provided to illustrate the effectiveness of WINC algorithms for PCA.

1. INTRODUCTION

Since the introduction of a simplified linear neuron with constrained Hebbian learning rule by Oja [1] which extracts a single principal component from stationary input data, neural approaches to perform Principal Component Analysis (PCA) have received a great deal of attention, and a variety of learning algorithms for PCA have been proposed. For a good reference, see [2]. The learning algorithms for multiple component extraction can be divided into sequential and parallel versions. The sequential version [3,4,5] can be implemented by making explicit use of the "deflation" procedure which is a computationally inexpensive orthonormalization method for extracting lower order components. A main advantage of the sequential version is that one can adaptively increase the number of neurons needed for PCA. The drawbacks of this version are that 1) one needs more memory to store the input samples to be repeatedly used; 2) the version produces a long processing delay due to the different component extraction one after another. In the parallel versions, the principal components of interest are extracted simultaneously. The parallel version [6,7,8,11] has been studied extensively since the parallel version can overcome the drawbacks of the sequential version and one can conveniently derive the sequential version from the parallel version.

It is well known that there have been two significant ways capable of performing in parallel the true PCA. The first way is based on the hierarchically structured lateral inhibition network [2]. While the resultant algorithm has the characteristic similar to the lattice filter, the speeds of convergence for different neurons are necessarily sorted according to their orders [6] . The other way is based on the weighted subspace criterion, but the network architecture is still symmetrical and the learning algorithm for different component extraction is local and homogeneous [7,8,9]. The properties of locality and homogeneity imply modularity and regularity in implementing the algorithm in parallel hardware [8]. But, the weighted subspace algorithm (WSA) [8] as well as Xu's Least Mean Square Error Reconstruction (LMSER) algorithm [7] is the fixed step size algorithm. In practice, the proper choice of the step size is often a difficult task. This often implies that a trade-off between the convergence speed and the steady-state error occurs easily [13].

Recently, Miao and Hua [10] proposed a novel information criterion (NIC) for searching for the optimum weights of a two-layer linear neural network. They have shown in [10] that unlike the mean square error (MSE), the NIC is nonquadratic and has a steep landscape along the trajectory from a small weight matrix to the optimum one, so that the NIC yields faster gradient-based algorithms. However, it is worth noting that the NIC is a subspace tracking criterion which produces just the principal subspace analysis (PSA) instead of true PCA. To perform true PCA, reorthormalization is needed [5], [10]. The reason is that the PSA algorithm is based on both the symmetrical neural network architectures and the symmetrical optimization criterion.

In this paper, we present a new approach for extracting in parallel multiple principal components. It is based on an idea of adding a weight to NIC [10] so that the optimum weights at equilibrium points will be exactly the desired eigenvectors of a covariance matrix instead of an arbitrary orthonormal basis in the principal subspace. The reformulated information maximization criterion, called the Weighted INformation Criterion (WINC) herein, yields some interesting results such as the improvement convergence when compared to WSA.[8] for PCA.

2. NEW LEARNING ALGORITHMS FOR PCA

Suppose the N-dimensional input vector $\mathbf{x}(k), k = 1,2,\cdots,$ be a zero-mean stationary stochastic process whose covariance matrix $\mathbf{R} = E\{\mathbf{x}(k)\mathbf{x}^T(k)\}$ with N positive eigenvalues. We arrange the orthonormal eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_p, \cdots, \mathbf{v}_N$ so that the corre-sponding eigenvalue sequence is in descent order: $\lambda_1 > \lambda_2 > \cdots > \lambda_p > \lambda_{p+1} \geq \cdots \geq \lambda_N > 0$ .For some applications, such as feature extraction and data compression, it is usually expected to obtain the first p dominant eigen-components spanning a principal subspace. In this case, the p-dimensional output vector $\mathbf{y}(k)$ in time k of a linear neural network is a linear

This work is supported in part by NNSF,Guangxi NSF, Foundation of Electronic Academic Institute, and Guangxi Education Bureau, China.

function of its inputs, namely

$$\mathbf{y}(k) = \mathbf{W}^T(k-1)\mathbf{x}(k) \tag{1}$$

where $\mathbf{W}(k-1)$ is an $N \times p$ connection weight matrix. To find the optimal weight matrix, Let's define the following objective function

$$J_{WINC}(\mathbf{W}) = \frac{1}{2}\{tr[\log(\mathbf{W}^T\mathbf{R}\mathbf{W}\mathbf{A})] - tr(\mathbf{W}^T\mathbf{W})\} \tag{2}$$

where $tr(\cdot)$ denotes a matrix trace and $\mathbf{A} = diag(a_1, a_2, \cdots, a_p)$ is a weighting matrix with $a_1 > a_2 > \cdots > a_p > 0$. It is a novel criterion for extracting in parallel multiple principal components, referred to as Weighted INformation Criterion (WINC) herein. Note that if let $\mathbf{A} = \mathbf{I}_p$ in (2) where $\mathbf{I}_p$ is an identical matrix, then WINC reduces to NIC [10] which performs PSA. So, this is a universal criterion for PCA and PSA.

By applying the gradient ascent searching to $J_{WINC}(\mathbf{W})$, we can obtain the corresponding gradient equation with respect to $\mathbf{W}$ in the following

$$\nabla J_{WINC}(\mathbf{W}) = \mathbf{R}\mathbf{W}\mathbf{A}(\mathbf{A}\mathbf{W}^T\mathbf{R}\mathbf{W})^{-1} - \mathbf{W} \tag{3}$$

The batch implementation for updating $\mathbf{W}(k)$ based on the above gradient equation is straightforward:

$$\mathbf{W}(k) = \mathbf{W}(k-1) + \eta[\hat{\mathbf{R}}(k)\mathbf{W}(k-1) - \mathbf{W}(k-1)\mathbf{A}\mathbf{W}^T(k-1)$$
$$\hat{\mathbf{R}}(k)\mathbf{W}(k-1)\mathbf{A}^{-1}][\mathbf{A}\mathbf{W}^T(k-1)\hat{\mathbf{R}}(k)\mathbf{W}(k-1)\mathbf{A}^{-1}]^{-1} \tag{4}$$

where $0 < \eta \le 1$ denotes a fixed step size and $\hat{\mathbf{R}}(k)$ is the estimate of $\mathbf{R}$ given by

$$\hat{\mathbf{R}}(k) = \frac{(k-1)\gamma}{k}\hat{\mathbf{R}}(k-1) + \frac{1}{k}\mathbf{x}(k)\mathbf{x}^T(k) \tag{5}$$

where $0 < \gamma \le 1$ denotes the forgetting factor which is used to track the nonstationary environment. It has been shown in [12] that $\mathbf{W}(k)$ will asymptotically converge to true principal eigen-vectors of $\mathbf{R}$ as $\hat{\mathbf{R}}(k)$ converges to $\mathbf{R}$.

Compared with the WSA proposed by Oja *et al* in [8]

$$\mathbf{W}(k) = \mathbf{W}(k-1) + \eta[\hat{\mathbf{R}}(k)\mathbf{W}(k-1) -$$
$$\mathbf{W}(k-1)\mathbf{A}\mathbf{W}^T(k-1)\hat{\mathbf{R}}(k)\mathbf{W}(k-1)\mathbf{A}^{-1}] \tag{6}$$

it is clear that WINC algorithm in (4) has an adaptive step size $[\mathbf{A}\mathbf{W}^T(k-1)\hat{\mathbf{R}}(k)\mathbf{W}(k-1)\mathbf{A}^{-1}]^{-1}$. This property brings about a significantly improved learning performance compared with WSA, as will be seen in Section III.

Although the batch WINC algorithm (4) is able to provide good learning performance, it requires, like WSA, a high computational complexity of $O(N^2p)$ every update. Furthermore, the matrix inverse, in despite of reduced dimension, is explicitly involved in computations. This is inconvenient for real applications and modular design of hardware. In the following, we develop a computationally efficient algorithm with $O(Np)$ operations every update based directly on the input vector sequence $\{\mathbf{x}(k)\}$.

Note that since

$$\hat{\mathbf{R}}(k)\mathbf{W}(k-1) = \frac{1}{k}\sum_{i=1}^{k}\gamma^{k-i}\mathbf{x}(i)\mathbf{y}^T(i) \tag{7}$$

where $\mathbf{y}(i) = \mathbf{W}^T(k-1)\mathbf{x}(i)$, considering the projection approximation $\mathbf{y}(i) = \mathbf{W}^T(i-1)\mathbf{x}(i)$ similar to [5],[10], (4) can be rewritten as

$$\mathbf{W}(k) = (1-\eta)\mathbf{W}(k-1) + \eta\tilde{\mathbf{W}}(k) \tag{8}$$

where $\tilde{\mathbf{W}}(k) = \left[\sum_{i=1}^{k}\gamma^{k-i}\mathbf{x}(i)\mathbf{y}^T(i)\right]\mathbf{A}\left[\sum_{i=1}^{k}\gamma^{k-i}\mathbf{y}(i)\mathbf{y}^T(i)\right]^{-1}\mathbf{A}^{-1}$ which can be calculated recursively [12]. Due to the limitation of the space herein, the RLS implementation of WINC algorithm is summarized in the following without derivations.

Initialization: Choose $\mathbf{P}(0)$, $\mathbf{W}(0)$, and $\tilde{\mathbf{W}}(0)$ properly.

Update equations (for $k \ge 1$):

$$\mathbf{y}(k) = \mathbf{W}^T(k-1)\mathbf{x}(k) \tag{9}$$

$$\mathbf{g}(k) = \frac{\mathbf{P}(k-1)\mathbf{y}(k)}{\gamma + \mathbf{y}^T(k)\mathbf{P}(k-1)\mathbf{y}(k)} \tag{10}$$

$$\mathbf{P}(k) = \gamma^{-1}(\mathbf{P}(k-1) - \mathbf{g}(k)\mathbf{y}^T(k)\mathbf{P}(k-1)) \tag{11}$$

$$\tilde{\mathbf{W}}(k) = \tilde{\mathbf{W}}(k-1) + \mathbf{x}(k)\tilde{\mathbf{g}}^T(k) - \tilde{\mathbf{x}}(k)\mathbf{g}^T(k)\mathbf{A}^{-1} \tag{12}$$

where $\tilde{\mathbf{g}}(k) = \mathbf{A}^{-1}\mathbf{P}(k)\mathbf{A}\mathbf{y}(k)$, $\tilde{\mathbf{x}}(k) = \tilde{\mathbf{W}}(k-1)\mathbf{A}\mathbf{y}(k)$.

$$\mathbf{W}(k) = (1-\eta)\mathbf{W}(k-1) + \eta\tilde{\mathbf{W}}(k) \tag{13}$$

From (9)-(13), it is easy to find that WINC algorithm requires $6Np + 3p^2 + 4p$ operations every update. If $\eta = 1$, it requires only $4Np + 3p^2 + 4p$ operations every update while WSA with the data driving requires $5Np + 2p$ operations every update. In general, the input dimension N is much larger than the output dimension p for many real applications. Thus the WINC algorithm is cheaper than the WSA if $p < N/3 - 1$. Notice that some subspace tracking algorithms [5],[10] demand $O(Np)$ operations for obtaining an arbitrary orthonormal basis and additional $O(N^2p)$ operations for postprocessing of the eigen-vector estimates. In comparison with NIC algorithm [10], the WINC algorithm demands an increase in the computational complexity of $Np + p^2 + 2p$ operations every update in order to perform directly the true eigenvector estimates. However, the WINC algorithm uses an incorporate network to implement extracting in parallel multiple principal eigenvectors without the need of extra design of the postprocessing network. In fact, the WINC algorithm corresponds to a three-layer linear neural network model (as shown in Fig.1). The model features a novel structure of fewer hidden units than input and output units fully connected by fixed weights, referred to as a weighting matrix.

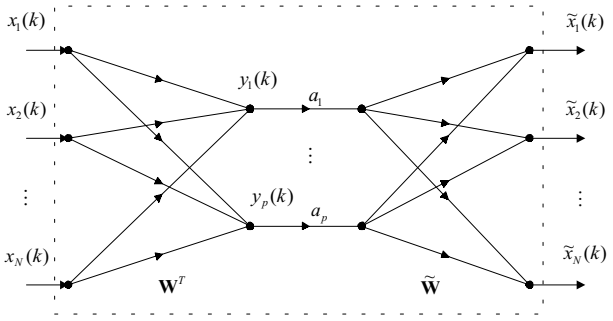

Fig.1 A three-layer linear PCA neural network model.

3. SIMULATION RESULTS

In this section, we present two simulation results to demonstrate the behavior and the applicability of the WINC algorithm. The first simulation serves to show the transient behavior of the learning in the principal eigenvectors. The results are compared with those obtained using the WSA. In the second simulation, we apply the WINC algorithm to compress the image data. The results are compared with those obtained using both the standard Karhunen-Loeve transformation (KLT) and the WSA. After the sample covariance matrix $\mathbf{R}$ has been updated via (5), its complete eigenvalue decomposition (EVD) is computed by a standard batch method in Matlab for obtaining the real principal eigenvectors.

3.1. Transient Behavior

In this simulation, the data sequence was generated by the first-order autoregressive process [4]

$$x(k) = 0.9x(k-1) + e(k)$$

where $e(k)$ is a zero-mean uncorrelated Gaussian driving sequence with unit variance. The data points are arranged in blocks of size six (N=6). In order to estimate the error for each eigenvector for each iteration, we calculate the direction cosine given by

$$\text{Direction Cosine}(k) = \left|\mathbf{w}_i^T(k)\mathbf{v}_i\right| \Big/ \left(\left\|\mathbf{w}_i(k)\right\|\left\|\mathbf{v}_i\right\|\right)$$

where $\mathbf{w}_i(k)$ is the estimated ith principal eigenvector at time k, and $\mathbf{v}_i$ denotes the actual ith principal eigenvector [13]. Clearly, if $\mathbf{w}_i(k)$ is exactly same as $\mathbf{v}_i$, then the maximum value of the direction cosine should be unity. Hence, for a good algorithm, the direction cosine should converge fast to unity as iterations.

Let us consider the first three (p=3) principal component extraction. Two algorithms—the WINC algorithm (4) and the WSA (6) — are run for the same random $\mathbf{W}(0)$ and $\mathbf{A}$=diag[1,0.9,0.8] with η=0.5 and η=0.01, respectively. Note that the η determined by trial and error is almost optimal for the WSA convergence. Fig.2 shows the transient behavior of the two algorithms for extracting the first three principal components in parallel.

It is obviously observed from Fig.2 that the WINC algorithm outperforms the WSA for extracting all the principal components. It converges fast to the true principal components. The WSA is obviously hierarchical algorithm for extracting multiple principal components, whereas the WINC algorithm can almost provide the consistent convergence speeds for extracting different principal component. This property is very significant for real time applications where the multiple principal component needs fast parallel extraction. The results show that the WINC algorithm provide the improved convergence speed for the parallel extraction of the multiple principal component via the adaptable step-size.

3.2 Image Data Compression

It is well known that the practical value of PCA is that it provides an effective technique for dimensionality reduction. To illustrate the applicability of WINC algorithm, let us consider the example of a Lena image . It has a resolution of 512×512 pixels with 256 gray levels. To train the algorithm, 8×8 nonoverlapping blocks of the image are used and then arranged into series of a 64-dimension input vector, with the image scanned from left to right and top to bottom to give a total of training samples K=4096. Once the training process is completed, the $\mathbf{W}(K)$ is used to reconstruct the image data. The reconstructed input vector is given by $\hat{\mathbf{x}}(k) = \mathbf{W}(K)\mathbf{W}^T(K)\mathbf{x}(k)$. The quality of the reconstructed image is measured by [5]

$$\text{SNR} = 10\log\frac{\sum_{k=1}^{K}\left\|\mathbf{x}(k)\right\|^2}{\sum_{k=1}^{K}\left\|\mathbf{x}(k)-\hat{\mathbf{x}}(k)\right\|^2}$$

In this simulation, two algorithms—the WINC algorithm (11)-(13) and the WSA (6) with the data driving—are run for the same initial $\mathbf{W}(0) = [\mathbf{I}_p, \mathbf{0}]^T$ and $\mathbf{A}$ with η=0.5 and η = 5×10^{-6}, respectively. The elements of the diagonal matrix $\mathbf{A}$ are set in exponential decrease. We take $\mathbf{P}(0) = 0.05\mathbf{I}_p$ and $\gamma = 1$ in the WINC algorithm. We use the result of the standard KLT as the benchmark to examine the performance of algorithms.

Fig.3 shows the SNR vs. different dimension p for two algorithms. We see that the difference between the WINC algorithm and the KLT in SNR is less than 0.5dB for $p \le 25$. In particular, the performance of the WINC algorithm is nearly identical to the KLT for $p \le 16$. Again, the performance of WSA is not better than that of the WINC algorithm. In particular, for

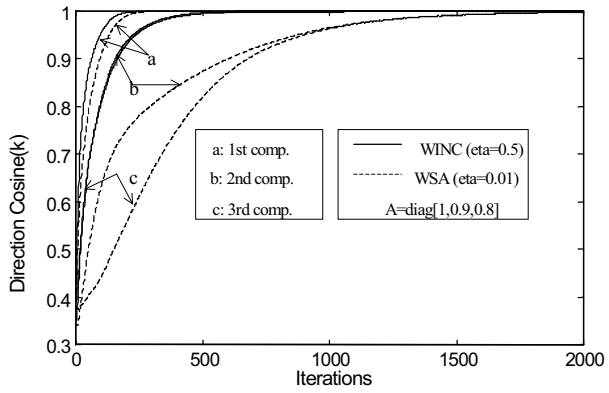

Fig.2 The direction cosine curves of WINC algorithm and WSA for extracting the first three principal eigenvectors.

$p \geq 13$ the SNR of WSA does not increase in direct ratio with the dimension p. This implies that the corresponding minor components do not converge to correct components, which results in the fact that the part of the signal energy is distributed to the lower order components which are rejected in the reduction process. So, the quality of the reconstructed image is not improved by the increased dimension. Obviously, this is not true for the WINC algorithm. Although the difference between the WINC algorithm and the KLT in SNR becomes increasing with p, the performance of the WINC algorithm is not degraded severely. For example, the difference between the WINC algorithm and the KLT in SNR is less than 1dB at $p = 30$.

4. CONCLUSION

This paper proposes an unconstrained optimization criterion—the WINC for parallel multiple principal component extraction on the basis of the concepts of the weighted subspace and the information maximization. Based on the gradient-ascent method, we derive two WINC algorithms for performing the true PCA recursively. The gradient-ascent version of the WINC algorithm turns out to be an extended WSA with the adaptive learning rate which leads to a significant improvement in the convergence speed. More importantly, Its RLS version not only provides the fast convergence and the high accuracy but also has the low computational complexity. The simulation results sufficiently show the high efficiency of the WINC algorithm for the parallel multiple principal component extraction. Furthermore, the WINC also generalizes some well-known PCA/PSA algorithms by introducing two adjustable parameters η and $\mathbf{A}$. Thus, the WINC provides a fast, flexible adaptive method for many potential real applications.

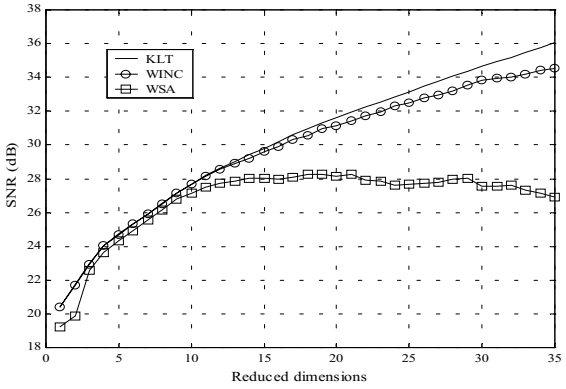

Fig.3 SNR of KL image compression vs. the reduced dimensions

.

REFERENCES

[1] E. Oja, "A simplified neuron model as a principal component analyzer," *J Math. Biology* ,vol.15,pp.267-273,1982

[2] K.I.Diamantaras and S.Y.Kung, *Principal Component Neural Networks-Theory and Applications* . Wiley, 1996.

[3] T.D.Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network," *Neural Networks*, vol.2, pp.459-473, 1989.

[4] S. Bannour and M. R. Azimi-Sadjadi, "Principal component extraction using recursive least squares learning," *IEEE Trans. Neural Networks*, vol.6, no.2, pp.457-469,1995.

[5] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Processing*, vol.43, no.1, pp.95-107,1995.

[6] S.Y.Kung, K.I.Diamantaras, and J.S.Taur, "Adaptive principal component extraction (APEX) and applications," *IEEE Trans. Signal Processing*, vol.42, no.5, pp.1202-1217, May 1994.

[7] L.Xu, "Least mean square error reconstruction principal for self-organizing neural-nets," *Neural Networks*, vol.6, pp.627-648, 1993.

[8] E.Oja, H.Ogawa, and J.Wangviwattana, "Principal component analysis by homogeneous neural networks, part I: weighted subspace criterion," *IEICE Trans. Information and System*, E75-D, no.3, pp.366-375, 1992.

[9] --, "Principal component analysis by homogeneous neural networks, part II: Analysis and extensions of the learning algorithms," *IEICE Trans. Information and System*, E75-D, no.3, pp.376-382, 1992.

[10] Y.F.Miao and Y.B. Hua, "Fast subspace tracking and neural network learning by a novel information criterion," *IEEE Trans. Signal Processing*, vol.46, pp.1967-1979, 1998.

[11] E.Oja, "Principal components, minor components, and linear neural networks," *Neural Networks*, vol.5, pp.927-935, 1992.

[12] S.Ouyang, *Neural learning algorithms for principal and minor components analysis and applications*. Ph.D. dissertation, Xidian University, Xi'an, June 2000.

[13] C.Chatterjee, V.P.Roychowdhury, and E.K.P.Chong, "On relative convergence properties of principal component analysis algorithms," *IEEE Trans. Neural Networks*, vol.9, no.2, pp.319-329, 1998.