SIMULTANEOUS EXTRACTION OF PRINCIPAL COMPONENTS USING GIVENS ROTATIONS AND OUTPUT VARIANCES

Deniz Erdogmus, Yadunandana N. Rao, Jose C. Principe, Jing Zhao, Kenneth E. Hild II

Computational NeuroEngineering Lab, University of Florida, Gainesville, FL 32611
[deniz,yadu,principe,jing,hildk]@cnel.ufl.edu

ABSTRACT

Principal Components Analysis (PCA) is an invaluable statistical tool in signal processing. In many cases, an on-line algorithm to adapt the PCA network to determine the principal projections in the input space is desired. Algorithms proposed until now use the traditional deflation or the inflation procedure to determine the intermediate components sequentially, after the convergence of the principal or minor component is achieved. In this paper, we propose a constrained linear network and a robust cost function to determine any number of principal components simultaneously. The topology exploits the fact that the eigenvector matrix sought is orthonormal. A gradient-based algorithm named SIPEX-G is also presented

1. INTRODUCTION

Principal component analysis (PCA) is a fundamental statistical technique that has proved its importance in numerous signal-processing applications including, but not limited to, feature extraction, signal estimation, detection, and speech separation [1-4]. Linear discriminant analysis (LDA) [5] is another example where the solution requires estimates of generalized eigenvalues. There are many algorithms that have been proposed for solving the PCA problem both off-line and on-line; Oja's rule [6] opened the door for many other useful on-line PCA algorithms including Sanger's rule [7], APEX [2] and Rubner and Tavan's method [8,9]. These topologies and their associated algorithms have been successfully applied to many problems of signal processing. However, they have shortcomings in speed of convergence mainly due to the fact that they are gradient algorithms and they depend heavily on the deflation procedure, which prevents the principal components from converging simultaneously. Although APEX and Rubner-Tavan networks achieve deflation using a lateral network of weights in the output layer, the convergence of the minor components is far from satisfactory. There are well-known fixed-point rules for PCA adaptation, which converge much faster than the slow gradient methods [5], [10]. However, they still use the deflation scheme to determine the subsequent principal components after the first principal component has converged. Xu's LMSER algorithm uses subspace techniques and a scalar amplification matrix to extract principal components simultaneously [11]. But, the convergence of LMSER is slow as it uses a simple gradient method to optimize an unconstrained network. In this paper, we present an on-line algorithm that converges to all eigenvectors simultaneously using a novel cost function by exploiting two key facts. The matrix that we seek, as the solution to the PCA problem, is an orthonormal matrix and the sum of the output variances are maximized for any number of primary components corresponding to the largest eigenvalues when their weight vectors are aligned with their corresponding eigenvectors. The performance of the proposed SIPEX-G algorithm is compared with that of Sanger's rule and Xu's LMSER algorithm.

2. COST FUNCTION

It is well known that the directions of the principal components are given by the eigenvectors of the covariance matrix of the input data, ordered according to their corresponding eigenvalues in descending order of magnitude [12]. Thus, PCA is nothing more than a coordinate transformation on the data, where in the new coordinate system the axes are aligned with the directions of maximal variation. This immediately points out that, the search for the weights of a principal component network can be restricted to the set of orthonormal matrices, since every orthonormal transformation corresponds to an axes-rotation on the input vector space. Consider the principal component network with $y=Rx$, where $x \in \Re^{nx1}$ and $y \in \Re^{nx1}$ are the input and output vectors respectively, and $R \in D \subset \Re^{nxn}$ is the weight matrix, which is restricted to the subset D of orthonormal matrices. The cost function in (1) could be maximized (or minimized) in order to determine the principal components of the input data, whose covariance matrix is given by Σ_x. The scalar gains g_o are chosen in descending order such that $g_1>g_2>\ldots>g_{n-1}>0$. Thus the cost function is just the weighted sum of first *(n-1)* output variances. In the subsequent discussions, we assume that the input data x is zero-mean, without loss of generality.

$$J = \sum_{o=1}^{n-1} g_o Var(y_o) \qquad (1)$$

Theorem 1: For the constrained network where the weight matrix R is an orthonormal matrix, the function J in (1) has a stationary point if and only if the rows of R consist of all the eigenvectors of Σ_x.
Proof: In the Appendix.
Lemma 1: There is a total of *n!* stationary points of *J* of which *(n-1)!* are local maxima, *(n-1)!* are local minima, and *(n-2)(n-1)!* are saddle points.
Proof: This follows easily from the ideas in the proof of Thm. 1.

Note that all stationary points of *J(R)* are valid PCA solutions and if necessary, ordering can be done easily by observing the output variances estimated in *J*. This theorem practically states that, we can adapt a rotation matrix (in batch mode or on a sample-by-sample basis) in order to obtain all the principal components of the input data at the output of this linear network. It is also possible to include in the cost function given in (1), only the variances of the first *m* outputs, which will result in convergence of the first *m* rows of the rotation matrix to the first *m* principal components. This case however, requires careful choice of the gains. Although the proof of this fact follows the same principals as the proof of Thm. 1, it is more involved and therefore we omit it in this paper to save space.

3. GIVENS ROTATIONS

Every orthonormal matrix can be considered a rotation matrix, thus they can be parameterized in terms of Givens rotation angles, each of which define a rotation in a single plane of the high-dimensional vector space. Then, these individual rotations can be cascaded to span the whole set of rotation matrices. Every rotation matrix has a unique set of Gives rotation angles that characterize it. In *n*-dimensions, a Givens rotation matrix in the plane formed by the i^{th} and j^{th} axes is denoted by R^{ij}, and is given by an identity matrix whose four entries at the intersection of i^{th} and j^{th} rows with i^{th} and j^{th} columns are modified as follows: The $(i,i)^{th}$ and $(j,j)^{th}$ entries are $cosq_{ij}$, and the $(i,j)^{th}$ and $(j,i)^{th}$ entries are $-sinq_{ij}$ and $sinq_{ij}$, respectively [13]. A rotation matrix is then formed from these sparse matrices according to

$$R = \prod_{p=1}^{n-1} \prod_{q=p+1}^{n} R^{pq} \tag{2}$$

The multiplication order can be always from the left or always from the right. It is not crucial to the generality of this formula as long as we maintain the same order when taking the derivative of the matrix with respect to a rotation angle.

4. ADAPTATION ALGORITHM: SIPEX-G

Our aim is to solve the following constrained optimization problem that becomes unconstrained if Givens angles are used.
<u>*Problem:*</u> Let q_{kl}, $k = 1,\ldots,n-1$, $l = k+1,\ldots,n$ be the Givens rotation angles that form up our parameter vector Θ. The cost function is explicitly given by

$$J = \sum_{o=1}^{n-1} g_o Var(y_o) = \sum_{o=1}^{n-1} g_o \sum_{i=1}^{n}\sum_{j=1}^{n} R_{oi} R_{oj} \Sigma_{x,ij} \tag{3}$$

where, R_{ij} is the (i,j)th entry of the rotation matrix R, which is constructed using the Givens angles as shown in (2).

The variances of each output component are evaluated using the covariance matrix of the input data and the entries of the rotation matrix, as clearly seen from (3). The reason for this is to obtain robust performance when this approach is used on-line, where the rotation matrix is updated after every new sample. If we were to estimate the output variances directly from the output samples then, due to the variation of the rotation matrix from sample to sample, it would be impossible to obtain an accurate estimate of the current value of the cost function. On the other hand, formulating the output variance in terms of the input covariance matrix and the current values of the rotation matrix allows us to use robust sample-by-sample updates to our estimation of the input covariance matrix, both in stationary or non-stationary environments using a suitable forgetting factor. Thus, in order to solve this optimization problem in an on-line fashion, we present the algorithm outlined below.
<u>*Algorithm:*</u> Simultaneous principal component extraction using the gradient approach (SIPEX-G)
Step 1. Initialize Givens angles (randomly or to all zeros so that the initial rotation matrix is the identity matrix).
Step 2. Use the first *N>n* samples of the input data to obtain an unbiased estimate to the covariance matrix Σ_x.

$$R_x = \frac{1}{N-n}\sum_{k=1}^{N} x_k x_k^T \tag{4}$$

Step 3. In non-WSS environments, update the covariance estimate with the following recursive formula.

$$R_x(k) = (1-a)R_x(k-1) + a x_k x_k^T \tag{5}$$

The memory depth of this recursion is $1/a$. If the input data is WSS, the following unbiased recursion may be used.

$$R_x(k) = \frac{k-n-1}{k-n} R_x(k-1) + \frac{1}{k-n} x_k x_k^T \tag{6}$$

Step 4. Calculate the gradient of the cost function with respect to the Givens angles using the covariance estimate in (5) or (6) in place of the actual covariance matrix in (3). This gradient is

$$\frac{\partial J}{\partial q_{kl}} = \sum_{o=1}^{n-1}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(R_{oi}\frac{\partial R_{oj}}{\partial q_{kl}} + \frac{\partial R_{oi}}{\partial q_{kl}} R_{oj}\right) R_{x,ij} \tag{7}$$

Step 5. Update the Givens angles using gradient ascent.

$$\Theta(k+1) = \Theta(k) + h\,\frac{\partial J}{\partial \Theta} \tag{8}$$

Step 6. Go back to step 3 and continue until convergence.

Using this algorithm, it is also possible to extract any desired number of principal components simultaneously. To determine the largest $m < n$ principal components, the upper limit for the summation index o in (3) and all the succeeding equations must be replaced by m. If $m = n-1$, the algorithm finds all of the n principal components, as once the first $n-1$ components are determined, the last one is automatically set.

Alternatively, one can minimize the cost function using the same algorithm except for a negative sign on the gradient to obtain m minor components of the input data.

A key concern in many adaptive algorithms is the computational complexity. It is clear that if the multiplications in (2) are performed from the left, the first output is only affected from the Givens angles with indices q_{1q}, $q=2,\ldots,n$, the second is affected by all the angles q_{1q}, $q=1,\ldots,n$ and q_{2q}, $q=3,\ldots,n$, and so on. Thus, if we wish to extract the first m principal components, we only need to adapt the angles q_{ij}, $i=1,\ldots,m$, $j=i+1,\ldots,n$, which makes a total of $mn-m(m+1)/2$ parameters, which is less than the mn parameters required in many PCA algorithms. But then, we will have to evaluate either the *sin* or *cos* of all these parameters once. In addition, the necessary matrix and vector multiplications in the algorithm will be performed at each iteration, which amount to $O(n^2)$.

5. CASE STUDIES

Consider the determination of the principal components of a three-dimensional Gaussian distribution with a randomly selected covariance matrix. The eigenspread of the chosen input covariance matrix is quite high. Specifically, the eigenvalues are 8.42, 0.45, and 0.02. We compare the performance of SIPEX-G with that of Sanger's rule. Both algorithms are initialized to the identity matrix. The step sizes of both algorithms are experimentally set such that the convergence of the first principal component is achieved in approximately 500 samples (iterations). With SIPEX-G, all three eigenvectors converged in less than 2000 samples almost simultaneously, with the designated step size. Sanger's rule took 3500 samples for the convergence of the second eigenvector and more than 10000 samples for the third. These results are summarized in Fig. 1, where the convergence of the direction cosines between the estimated and actual eigenvectors for both algorithms is presented. Recall that the value of ± 1 for the direction cosine means the two vectors are perfectly aligned.

As a second example, we consider a real world time series collected from a violin playing a single note. The 1000-sample time series is stationary. Using a 4-delay-line 5-D input samples for each time step are obtained. Both SIPEX-G algorithm and Xu's LMSER algorithm are applied to the same data set to extract the five principal components. The step sizes of both algorithms are set to yield convergence of the first component in approximately 500 iterations. Fig. 2 shows the convergence results for these algorithms in terms of the direction cosines between the estimated and the true eigenvectors (determined using the complete data set off-line). SIPEX-G converges to all eigenvectors in less than 1000 iterations whereas LMSER gives only the first two components for the same number of iterations. The LMSER uses a scalar amplification matrix similar to the scalar gains we use for weighting the output variances. It does not, however, explicitly constrain the norms of the weight vectors and uses a slow gradient subspace algorithm for updating the weights. In contrast, SIPEX-G always uses a rotation matrix, which reduces the search to the set of orthonormal matrices. This vastly improves the speed of convergence over LMSER or any other gradient method.

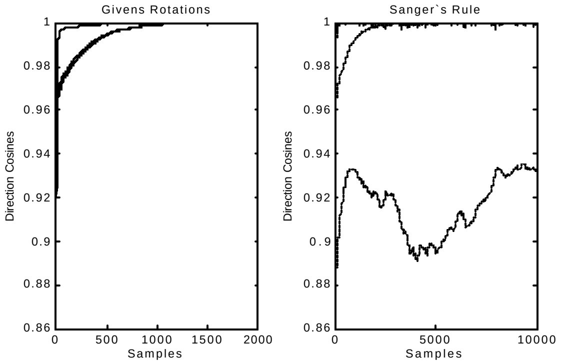

Figure 1. Multivariate Gaussian data: Direction cosines between the estimated and actual eigenvectors.

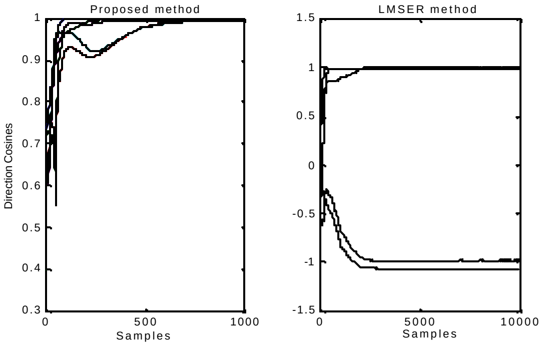

Figure 2. Violin time-series: Direction cosines between the estimated and actual eigenvectors.

The scalar gains of SIPEX-G were set to {3,2} in the first case and to {5,4,3,2} in the second example. For LMSER, in the second example, the gains were selected to be {5,4,3,2,1}. These gains help remove the plateau that might exist near the optima when the eigenspread of the data is high which can otherwise slow down the convergence of the weights to optimal values.

6. CONCLUSIONS

PCA is a crucial part of statistical signal processing and there are many on-line algorithms in the literature that determine the eigenvectors of the input covariance matrix. These algorithms, however, do not exploit the fact that the solution we seek for the weight matrix lies in the subset of orthonormal matrices. Due to this, many algorithms rely on deflation to obtain each eigenvector in a descending order sequentially. In this paper, we have addressed this question about the existence of on-line PCA algorithms that avoid the process of deflation and converge to all the desired principal components of the data simultaneously.

Exploiting the fact that the solution lies within the set of orthonormal matrices, we have parameterized the weight matrix of the linear PCA network using the Givens rotation angles. Furthermore, we have suggested a cost function, of which all stationary points are valid PCA solutions, to optimize these parameters and proved that its global maximum occurs at the desired eigenvector matrix. Additional advantages of the proposed approach, which we named SIPEX-G, are that the orthonormality of the estimated eigenvectors is guaranteed at every iteration and that the cost function can alternatively be minimized to obtain the minor components of the data. On the other hand, since the cost function explicitly depends on the output variances, and we still use a gradient approach, the convergence is still dependent on the step size. Future work would be directed towards the development of a fixed-point algorithm to further increase the efficiency of the algorithm and also to reduce the computational burden.

Acknowledgments: This work is partially supported by NSF grant ECS-9900394.

APPENDIX

Proof of Theorem 1: Due to limited space, we will not give the complete details of the proof, but only sketch the methodology. The covariance matrix of the output vector is given by $\Sigma_y = R\Sigma_x R^T$, whose diagonal entries correspond to the variances of the corresponding outputs, i.e. $Var(y_o) = \Sigma_{y,oo}$. An arbitrary rotation matrix can be decomposed into two orthonormal matrices $R = \overline{R}Q_x^T$, where Q_x is the ordered eigenvector matrix for the input covariance matrix. Thus we can consider $\overline{R}$ as the optimization variable (parameterized in terms of Givens angles).

$$\begin{aligned} J(R) = J(\overline{R}) &= \sum_{o=1}^{n-1} g_o Var(y_o) = \sum_{o=1}^{n-1} g_o \Sigma_{y,oo} = \sum_{o=1}^{n-1} g_o \left(R\Sigma_x R^T\right)_{oo} \\ &= \sum_{o=1}^{n-1} g_o \left(\overline{R}Q_x^T \Sigma_x Q_x \overline{R}^T\right)_{oo} = \sum_{o=1}^{n-1} g_o \left(\overline{R}\Lambda_x \overline{R}^T\right)_{oo} \end{aligned} \qquad \text{(A.1)}$$

In order to prove that only permutation matrices for $\overline{R}$ are stationary points, we consider $J(\overline{R})$ and $J(\overline{R} + d\ \overline{R})$, where $d\ \overline{R}$ is a perturbation matrix that satisfies the orthonormality constraint $(\overline{R} + d\ \overline{R})^T(\overline{R} + d\ \overline{R}) = I$. Both the original and perturbed matrices must also satisfy the magnitude constraint on their individual entries, i.e. $|\overline{r}_{ij}| \le 1$. Now, considering two perturbations $\pm d\ \overline{R}$ to $\overline{R} \ne P$, where P is a permutation matrix, we observe that the cost function both increases and decreases in one of these perturbation directions. Thus, we conclude that any rotation matrix *R*, which is not a permutation of the eigenvectors, is not a stationary point. The magnitude constraint on the entries prevents this conclusion from applying to permutation matrices.

In order to prove that all permutations of the eigenvector matrix, i.e. all cases where $\overline{R} = P$ are stationary points, we express $\overline{R}$ parametrically in terms of Givens rotations as

$$\overline{R} = \prod_{p=1}^{n-1} \prod_{q=p+1}^{n} \overline{R}^{pq} \qquad \text{(A.2)}$$

where $\overline{q}_{pq}$ are the Givens angles. Noticing that each and every permutation matrix corresponds to the case where these Givens angles are integer multiples of $p/2$, one can show with brute force method that the gradient in (7) vanishes because each term of the summation becomes zero. Thus we conclude that every permutation of the eigenvector matrix is a stationary point of *J.*

Combining these two results, we conclude that the function *J(R)* has a stationary point when and only when the rotation matrix is a permutation of the eigenvector matrix.

REFERENCES

[1] R. O. Duda, P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.

[2] S. Y. Kung, K. I. Diamantaras, J.S. Taur, *"Adaptive Principal Component Extraction (APEX) and Applications," IEEE Tran. Sig. Proc., vol. 42, May 1994.*

[3] J. Mao, A. K. Jain, "Artificial Neural Networks for Feature Extraction and Multivariate Data Projection," *IEEE Tran. Neural Networks*, vol. 6, no. 2, March 1995.

[4] Y. Cao, S. Sridharan, M. Moody, "Multichannel Speech Separation by Eigendecomposition and its Application to Co-Talker Interference Removal," *IEEE Tran. Speech and Audio Proc.*, vol. 5, no. 3, May 1997.

[5] G. Golub, C.V. Loan, *Matrix Computation*, John Hopkins University Press, Baltimore, MD, 1993.

[6] E. Oja, *Subspace Methods for Pattern Recognition*, Wiley, New York, 1983.

[7] T. D. Sanger, "Optimal Unsupevised Learning in a Single Layer Linear Feedforward Neural Network," *Neural Networks*, vol. 2, no. 6, pp. 459-473, 1989.

[8] J. Rubner, K. Schulten, " Development of Feature Detectors by Self Organization," *Biol. Cybern.*, vol. 62, pp. 193-199, 1990.

[9] J. Rubner, P. Tavan, "A Self Organizing Network for Principal Component Analysis," *Europhysics Letters*, vol. 10, pp. 693-698, 1989.

[10] Y. N. Rao, J. C. Principe, "A Fast, On-Line Algorithm for PCA and its Convergence Characteristics," *Proc. NNSP X*, vol. 1, pp. 299-307, 2000.

[11] L. Xu, "Least Mean Square Error Reconstruction Principle for Self-Organizing Neural-Nets," *Neural Networks*, vol. 6, pp. 627-648, 1993.

[12] I. T. Jolliffe, *Principal Component Analysis*, Springer-Verlag, New York, 1986.

[13] K. E. Hild II, D. Erdogmus, J. C. Principe, "Blind Source Separation Using Renyi's Mutual Information", *IEEE Signal Proc. Letters*, vol. 8, no. 6, pp. 174-176, June 2001.