ADAPTIVE ALGORITHMS FOR EIGEN-DECOMPOSITION AND THEIR APPLICATIONS IN CDMA COMMUNICATION SYSTEMS

CHANCHAL CHATTERJEE
GDE SYSTEMS INC., 16250 TECHNOLOGY DRIVE, SAN DIEGO, CA 92127-1806.
AND
VWANI P. ROYCHOWDHURY
ELECTRICAL ENGINEERING DEPARTMENT, UCLA, LOS ANGELES, CA 90095

Abstract - We derive and discuss two new algorithms for principal component analysis (PCA) that are shown to converge faster than the traditional PCA algorithms due to Oja and Sanger. It is well known that the traditional PCA algorithms, which are derived by using the gradient ascent technique on an objective function, are slow to converge. Furthermore, the convergence of these algorithms depends on the appropriate selection of the gain sequences. Since online applications demand faster convergence and an adaptive choice of the gains, we present new algorithms to solve these problems. We first present a new *unconstrained* objective function which can be maximized to obtain the PCA components. Adaptive algorithms are derived from this objective function by the use of the (1) gradient ascent, (2) conjugate direction, and the (3) Newton-Rhapson methods of optimization. Although the gradient ascent technique results in the well-known Xu algorithm, the conjugate direction and Newton-Rhapson methods produce two new algorithms for PCA. Extensive experiments on synthetic Gaussian and real-world signal data show the faster convergence of the new algorithms over the traditional methods.

1. INTRODUCTION

Recent research in the area of adaptive principal component analysis (PCA) has produced a large number of algorithms that are derived from: (1) anti-Hebbian learning [1,5], (2) Hebbian learning [1,5,6], (3) lateral interaction algorithms [1,5], and (4) gradient-based learning [1,5]. It is well-known [1,5] that that the traditional PCA algorithms, which are derived by using the gradient ascent technique on an objective function, are slow to converge. Furthermore, both analytical and experimental studies show that the convergence of the algorithms depend on the appropriate selection of the gain sequence $\{\eta_k\}$ (see eqn. (1)). Moreover, it is proven [6] that if the gain sequence exceeds an upper bound, then the algorithms may diverge or converge to a false solution.

Since most of these algorithms are used for real-time (i.e., online) processing, it is especially difficult to determine an appropriate choice of the gain parameter at the start of the online process. Hence, it is important, for wider applicability of these algorithms, to: (1) speed up the convergence of the algorithms, and (2) automatically select the gain parameter based on the current data sample.

A traditional algorithm for PCA is due to Oja *et al* [6] and due to Sanger [8] as below

$$W_{k+1} = W_k + \eta_k \left(A_k W_k - W_k \, \mathrm{UT}\left[W_k^T A_k W_k \right] \right) \qquad (1)$$

where $\{\eta_k\}$ is a sequence of scalar gains, and UT[·] sets all elements below the diagonal of its matrix argument to zero, thereby making it upper triangular.

In this study, we investigate two different methods to speed up the traditional algorithm (1). In the process, we derive two new algorithms for PCA that are shown to converge faster than algorithm (1). We also select the gain sequence $\{\eta_k\}$ based on the current data sample.

Several researchers [1] have shown that algorithm (1) can be derived by maximizing an objective function under constraints. Hence, we can use the theory of nonlinear optimization to this maximization task. In order to achieve this goal, we first derive a **new unconstrained objective function** that can be maximized to obtain the PCA components.

We next apply the Conjugate Direction methods to this unconstrained objective function to derive a new algorithm for PCA. This algorithm is shown to converge faster than the traditional algorithm (1). Furthermore, this new algorithm adaptively computes the gain sequence $\{\eta_k\}$ based on the current data sample.

We next apply the Newton-Rhapson method to this unconstrained objective function to obtain a second new algorithm for PCA. This algorithm is also shown to converges faster than the traditional algorithm (1). This

1058-6393/98 $10.00 © 1998 IEEE

algorithm also adaptively computes the gain sequence $\{\eta_k\}$ based on the current data sample.

In Section 2, we present a new unconstrained objective function for PCA. We next apply the Conjugate Direction methods to this objective function to obtain a new algorithm for PCA. We further use the Newton-Rhapson method to this objective function to obtain a second new algorithm for PCA. In Section 3, we provide a proof of convergence of the new algorithms under some conditions. In Section 4, we present the experimental results with synthetic Gaussian and real-world signal data showing the faster convergence of the two new algorithms over the traditional algorithm (1).

2. TWO NEW ALGORITHMS FOR PCA

It is well-known [1] that the standard quadratic problem leading to a PCA solution is one of how to maximize the objective function

$\mathbf{w}_i^T A\mathbf{w}_i$ under constraint $\mathbf{w}_i^T\mathbf{w}_j = \delta_{ij}$ for $j = 1,...,i-1$,

where $\mathbf{w}_i \in \Re^n$ is the i^{th} principal eigenvector of a positive definite matrix $A \in \Re^{nXn}$, and δ_{ij} is the Kronecker's delta. This is equivalent to the unconstrained maximization problem

$$\tilde{J}(\mathbf{w}_i) = \mathbf{w}_i^T A\mathbf{w}_i - \alpha\left(\mathbf{w}_i^T\mathbf{w}_i - 1\right) - 2\sum_{j=1}^{i-1}\beta_j\mathbf{w}_i^T\mathbf{w}_j, \quad (2)$$

where α and β_j (j=1,...,i–1) are the Lagrange multipliers. By equating the gradient of $\tilde{J}$ with respect to $\mathbf{w}_i$ to zero, we obtain

$$(1/2)\nabla_{\mathbf{w}_i}\tilde{J}(\mathbf{w}_i) = A\mathbf{w}_i - \alpha\mathbf{w}_i - \sum_{j=1}^{i-1}\beta_j\mathbf{w}_j = 0.$$

By multiplying $\nabla_{\mathbf{w}_i}\tilde{J}(\mathbf{w}_i)$ to the left by $\mathbf{w}_i^T$ and applying the constraints, we obtain

$$\alpha = \mathbf{w}_i^T A\mathbf{w}_i.$$

Similarly, by multiplying $\nabla_{\mathbf{w}_i}\tilde{J}(\mathbf{w}_i)$ to the left by $\mathbf{w}_j^T$ and applying the constraints, we obtain

$$\beta_j = \mathbf{w}_j^T A\mathbf{w}_i \text{ for } j = 1,...,i-1.$$

Replacing these values of the Lagrange multipliers into (2), we obtain a new unconstrained objective function

$$J(\mathbf{w}_i) = \mathbf{w}_i^T A\mathbf{w}_i - \mathbf{w}_i^T A\mathbf{w}_i\left(\mathbf{w}_i^T\mathbf{w}_i - 1\right) - 2\sum_{j=1}^{i-1}\mathbf{w}_j^T A\mathbf{w}_i\mathbf{w}_i^T\mathbf{w}_j. \quad (3)$$

We can now apply different methods for maximizing this objective function (3) to obtain different algorithms for PCA.

2.1 Gradient Ascent Method

We first apply the gradient ascent technique on this objective function (3) to obtain

$$(1/2)\nabla_{\mathbf{w}_i}J(\mathbf{w}_i) = A\mathbf{w}_i - \mathbf{w}_i\mathbf{w}_i^T A\mathbf{w}_i - \sum_{j=1}^{i-1}\mathbf{w}_j\mathbf{w}_j^T A\mathbf{w}_i - A\mathbf{w}_i\mathbf{w}_i^T\mathbf{w}_i - \sum_{j=1}^{i-1}A\mathbf{w}_j\mathbf{w}_j^T\mathbf{w}_i. \quad (4)$$

This gives the following adaptive algorithm for PCA

$$\mathbf{w}_{k+1}^i = \mathbf{w}_k^i + \eta_k\Big(2A_k\mathbf{w}_k^i - \mathbf{w}_k^i\mathbf{w}_k^{iT}A_k\mathbf{w}_k^i - \sum_{j=1}^{i-1}\mathbf{w}_k^j\mathbf{w}_k^{jT}A_k\mathbf{w}_k^i - A_k\mathbf{w}_k^i\mathbf{w}_k^{iT}\mathbf{w}_k^i - \sum_{j=1}^{i-1}A_k\mathbf{w}_k^j\mathbf{w}_k^{jT}\mathbf{w}_k^i\Big) \quad (5)$$

where A_k is the online observation for A. It is convenient to define a matrix W=[$\mathbf{w}_1$,...,$\mathbf{w}_p$] ($p \le n$), for which the columns are the p weight vectors that converge to the p principal eigenvectors of A respectively. Then, (5) can be represented as

$$W_{k+1} = W_k + \eta_k\left(2A_kW_k - W_k\mathrm{UT}\left[W_k^TA_kW_k\right] - A_kW_k\mathrm{UT}\left[W_k^TW_k\right]\right) \quad (6)$$

Note that (5) is an algorithm due to Xu [10] that can also be derived from a least mean squared error criterion of a feedforward neural network.

2.2 Conjugate Direction Method

We define the online gradient $\mathbf{g}_k^i$ of the objective function $J(\mathbf{w}_i)$ in (3) as follows:

$$\mathbf{g}_k^i = 2A_k\mathbf{w}_k^i - \mathbf{w}_k^i\mathbf{w}_k^{iT}A_k\mathbf{w}_k^i - \sum_{j=1}^{i-1}\mathbf{w}_k^j\mathbf{w}_k^{jT}A_k\mathbf{w}_k^i - A_k\mathbf{w}_k^i\mathbf{w}_k^{iT}\mathbf{w}_k^i - \sum_{j=1}^{i-1}A_k\mathbf{w}_k^j\mathbf{w}_k^{jT}\mathbf{w}_k^i. \quad (7)$$

The adaptive conjugate direction algorithm for PCA can be obtained as follows:

$$\mathbf{w}_{k+1}^i = \mathbf{w}_k^i + \alpha_k^i\mathbf{d}_k^i$$
$$\mathbf{d}_{k+1}^i = \mathbf{g}_k^i + \beta_k^i\mathbf{d}_k^i. \quad (8)$$

We next discuss the choices of α_k^i and β_k^i from the current data sample A_k. The gain constant α_k^i is chosen as the α that maximizes $J\left(\mathbf{w}_k^i + \alpha\mathbf{g}_k^i\right)$. Since we have an expression for $J(\mathbf{w}_i)$ in (3), we maximize the function

$J\left(\mathbf{w}_k^i + \alpha \mathbf{g}_k^i\right)$ with respect to α and obtain the following cubic equation:

$$c_3\alpha^3 + c_2\alpha^2 + c_1\alpha + c_0 = 0. \quad (9)$$

For the principal eigenvector case, the coefficients in this equation (9) are

$$c_3 = 2\mathbf{g}_k^{1\,T} A_k \mathbf{g}_k^1 \mathbf{g}_k^{1\,T} \mathbf{g}_k^1$$

$$c_2 = 3\left(\mathbf{g}_k^{1\,T} A_k \mathbf{g}_k^1 \mathbf{w}_k^{1\,T} \mathbf{g}_k^i + \mathbf{g}_k^{1\,T} \mathbf{g}_k^1 \mathbf{w}_k^{1\,T} A_k \mathbf{g}_k^1\right)$$

$$c_1 = \mathbf{g}_k^{1\,T}\left(\mathbf{w}_k^{1\,T} A_k \mathbf{w}_k^1 I + 4A_k \mathbf{w}_k^1 \mathbf{w}_k^{1\,T} + \mathbf{w}_k^{1\,T} \mathbf{w}_k^1 A_k - 2A_k\right)\mathbf{g}_k^1$$

$$c_0 = \mathbf{g}_k^{1\,T} \mathbf{g}_k^1.$$

With known values of $\mathbf{w}_k^i$ and $\mathbf{g}_k^i$, the above equation can be solved to obtain an α that maximizes $J\left(\mathbf{w}_k^i + \alpha \mathbf{g}_k^i\right)$.

For the choice of β_k^i, we can use a number of methods as described below:

Hestenes-Stiefel:

$$\beta_k^i = \mathbf{g}_{k+1}^{i\,T}\left(\mathbf{g}_{k+1}^i - \mathbf{g}_k^i\right) \Big/ \mathbf{d}_k^{i\,T}\left(\mathbf{g}_{k+1}^i - \mathbf{g}_k^i\right),$$

Polak-Ribiere: $\beta_k^i = \mathbf{g}_{k+1}^{i\,T}\left(\mathbf{g}_{k+1}^i - \mathbf{g}_k^i\right) \Big/ \mathbf{g}_k^{i\,T} \mathbf{g}_k^i$,

Fletcher-Reeves: $\beta_k^i = \mathbf{g}_{k+1}^{i\,T} \mathbf{g}_{k+1}^i \Big/ \mathbf{g}_k^{i\,T} \mathbf{g}_k^i$, and

Powell: $\beta_k^i = \max\left[0, \mathbf{g}_{k+1}^{i\,T}\left(\mathbf{g}_{k+1}^i - \mathbf{g}_k^i\right) \Big/ \mathbf{g}_k^{i\,T} \mathbf{g}_k^i\right]$.

We now represent the adaptive PCA algorithm (8) in the matrix form. We define the following matrices:

$W_k = [\mathbf{w}_k^1, ..., \mathbf{w}_k^p]$, $G_k = [\mathbf{g}_k^1, ..., \mathbf{g}_k^p]$, $D_k = [\mathbf{d}_k^1, ..., \mathbf{d}_k^p]$,

$\Gamma_k = \text{diag}(\alpha_k^1, ..., \alpha_k^p)$, and $\Pi_k = \text{diag}(\beta_k^1, ..., \beta_k^p)$.

Then, the new adaptive PCA algorithm is:

$$G_k = 2A_k W_k - W_k \text{UT}\left[W_k^T A_k W_k\right] - A_k W_k \text{UT}\left[W_k^T W_k\right]$$

$$W_{k+1} = W_k + D_k \Gamma_k,$$

$$D_{k+1} = G_k + D_k \Pi_k. \quad (10)$$

Here UT[·] is same as in (1).

2.3 Newton-Rhapson Method

The Newton-Rhapson algorithm is also based on the objective function $J(\mathbf{w}_i)$ in (3). Since Newton-Rhapson is a function minimization algorithm, we consider the minimization of $-J(\mathbf{w}_i)$ to compute the i^{th} principal eigenvector of A. The online gradient $\mathbf{g}_k^i$ of $-J(\mathbf{w}_i)$ is:

$$\mathbf{g}_k^i = -2A_k \mathbf{w}_k^i + \mathbf{w}_k^i \mathbf{w}_k^{i\,T} A_k \mathbf{w}_k^i + \sum_{j=1}^{i-1} \mathbf{w}_k^j \mathbf{w}_k^{j\,T} A_k \mathbf{w}_k^i + A_k \mathbf{w}_k^i \mathbf{w}_k^{i\,T} \mathbf{w}_k^i + \sum_{j=1}^{i-1} A_k \mathbf{w}_k^j \mathbf{w}_k^{j\,T} \mathbf{w}_k^i. \quad (11)$$

The online Hessian F_k^i of $-J(\mathbf{w}_i)$ is given by:

$$F_k^i = -2A_k + \left(\mathbf{w}_k^{i\,T} A_k \mathbf{w}_k^i\right) I + \left(\mathbf{w}_k^{i\,T} \mathbf{w}_k^i\right) A_k + 2\mathbf{w}_k^i \mathbf{w}_k^{i\,T} A_k + 2A_k \mathbf{w}_k^i \mathbf{w}_k^{i\,T} + \sum_{j=1}^{i-1} \mathbf{w}_k^j \mathbf{w}_k^{j\,T} A_k + \sum_{j=1}^{i-1} A_k \mathbf{w}_k^j \mathbf{w}_k^{j\,T}. \quad (12)$$

The adaptive Newton-Rhapson algorithm for PCA is:

$$\mathbf{w}_{k+1}^i = \mathbf{w}_k^i - \left(F_k^i\right)^{-1} \mathbf{g}_k^i. \quad (13)$$

The main concerns in this algorithm are that F_k^i should be positive definite, and that we should adaptively obtain an estimate of F_k^{i-1} in order to make the algorithm computationally efficient. These two concerns are met if we approximate the Hessian by dropping the terms

$$-A_k + \mathbf{w}_k^{i\,T} \mathbf{w}_k^i A_k.$$

The new Hessian is

$$F_k^i \approx \mathbf{w}_k^{i\,T} A_k \mathbf{w}_k^i I - \tilde{A}_k + 2A_k \mathbf{w}_k^i \mathbf{w}_k^{i\,T} + 2\mathbf{w}_k^i \mathbf{w}_k^{i\,T} A_k, \quad (14)$$

where

$$\tilde{A}_k = A_k - \sum_{j=1}^{i-1} \mathbf{w}_k^j \mathbf{w}_k^{j\,T} A_k - \sum_{j=1}^{i-1} A_k \mathbf{w}_k^j \mathbf{w}_k^{j\,T}.$$

Inverting this Hessian consists of inverting the matrix $B_k = \mathbf{w}_k^{i\,T} A_k \mathbf{w}_k^i I - \tilde{A}_k$, and two rank-one updates. An approximate inverse of this matrix B_k is given by

$$B_k^{-1} = \left(\mathbf{w}_k^{i\,T} A_k \mathbf{w}_k^i I - \tilde{A}_k\right)^{-1} \approx \frac{I + \tilde{A}_k \Big/ \mathbf{w}_k^{i\,T} A_k \mathbf{w}_k^i}{\mathbf{w}_k^{i\,T} A_k \mathbf{w}_k^i}. \quad (15)$$

An adaptive algorithm for inverting the Hessian F_k^i in (14) can be obtained by two rank-one updates. Let us define

$$C_k^i = B_k + 2A_k \mathbf{w}_k^i \mathbf{w}_k^{i\,T}. \quad (16)$$

Then from (14), an update formula for $(F_k^i)^{-1}$ is:

$$\left(F_k^i\right)^{-1} = \left(C_k^i\right)^{-1} - \frac{2\left(C_k^i\right)^{-1} \mathbf{w}_k^i \mathbf{w}_k^{i\,T} A_k \left(C_k^i\right)^{-1}}{1 - 2\mathbf{w}_k^{i\,T} A_k \left(C_k^i\right)^{-1} \mathbf{w}_k^i}, \quad (17)$$

where

$$\left(C_k^i\right)^{-1} = B_k^{-1} - \frac{2B_k^{-1} A_k \mathbf{w}_k^i \mathbf{w}_k^{i\,T} B_k^{-1}}{1 + 2\mathbf{w}_k^{i\,T} B_k^{-1} A_k \mathbf{w}_k^i}. \quad (18)$$

3. PROOFS OF CONVERGENCE

Although the new PCA algorithms are derived by following standard optimization techniques on an objective function (3), their derivations do not constitute a proof of convergence. Hence, it is important to provide a proof of convergence of the algorithms described above.

Algorithm (6) is derived by following a gradient ascent technique on the objective function $J(\mathbf{w}_i)$ in (3). This algorithm has been previously derived by Xu [10] from a least mean squared error criterion of a feedforward neural network. The convergence of the algorithm is also proven by Xu [10]. Note that this algorithm is a stochastic approximation procedure [2,3]. A proof of convergence of this algorithm based on the stochastic approximation theory for the generalized eigen-decomposition case is given in [3].

Algorithm (10) is obtained by following the conjugate direction method on the objective function $J(\mathbf{w}_i)$ in (3). For its proof of convergence, we assume the following:

Assumption (A1). The random observations $\{A_k\}$ have the dominated convergence property, i.e., there exists a positive definite matrix A such that

$$A_k \rightarrow A. \text{ w.p.1.}$$

Assumption (A2). The p largest eigenvalues of A are each of unit multiplicity.

There are several models for generating the random observation sequence $\{A_k\}$. One useful model that is commonly used in signal and image processing applications [3,5] is where $\{A_k\}$ is generated from a sequence of random vectors $\{\mathbf{x}_k\}$, whereby $\lim_{k\rightarrow\infty} E[\mathbf{x}_k \mathbf{x}_k^T] = A$. In order to generate $\{A_k\}$ that satisfies $A_k \rightarrow A$ w.p.1 as required by Assumption A1, we obtain $\{A_k\}$ as the output of a dynamical system given by the following algorithm

$$A_k = A_{k-1} + \gamma_k \left(\mathbf{x}_k \mathbf{x}_k^T - A_k\right), \qquad (19)$$

where A_0 is symmetric, and $\{\gamma_k\}$ is a positive scalar gain sequence that satisfies:

$$\gamma_k \rightarrow 0, \sum_{k=0}^{\infty} \gamma_k = \infty, \text{ and } \sum_{k=0}^{\infty} \gamma_k^2 < \infty. \qquad (20)$$

In the following discussions, we denote $\lambda_1 > \lambda_2 > \ldots > \lambda_p \geq \ldots \geq \lambda_n > 0$ as the eigenvalues of A, and ϕ_i as the eigenvector corresponding to λ_i such that $\phi_1, \ldots, \phi_n$ are orthonormal. Let $\Phi=[\phi_1 \ \ldots \ \phi_n]$ and $\Lambda=\text{diag}(\lambda_1,\ldots,\lambda_n)$ denote the matrix of eigenvectors and eigenvalues of A. Notice that if ϕ_i is an eigenvector of A, then $d_i\phi_i$ ($d_i=\pm 1$) is also an eigenvector of A.

Lemma 1. *Let A2 hold. Then, all the equilibrium points of the objective function $J(\mathbf{w}_i)$ in (3) are up to an arbitrary permutation of the p eigenvectors of A, i.e., any point $d_i\phi_{(i)}$, where d_i = 0, +1 or –1, is an equilibrium point of $J(\mathbf{w}_i)$.*

■

Lemma 2. *Let A2 hold. Then, the points $d_i\phi_i$, where d_i = +1 or –1, are the strict global maximum points of the objective function $J(\mathbf{w}_i)$ in (3).*

■

The convergence of algorithm (10) can now be proven by an application of the dominated convergence theorem [9], and by using Lemmas 1 and 2.

Theorem 1. *Let A1 and A2 hold. If W_0 is sufficiently close to the strict global maximum solution W^* in Lemma 2, then the sequence $\{W_k\}$ generated by algorithm (10) converges to W^* with probability one.*

■

Proofs of Lemmas 1, 2 and of Theorem 1 are omitted due to lack of space.

4. EXPERIMENTAL RESULTS

4.1 Experiments with Synthetic Data

Here we generated 500 samples of 5-dimensional Gaussian data with mean zero and covariance given below. Note that this covariance matrix is obtained from the first covariance matrix in [7] multiplied by 2. The covariance matrix is

0.1820	0.0760	-0.1060	-0.0100	0.0200
0.0760	0.7460	0.0360	-0.0560	-0.0220
-0.1060	0.0360	2.8600	0.0340	0.1100
-0.0100	-0.0560	0.0340	0.1680	-0.0100
0.0200	-0.0220	0.1100	-0.0100	0.1420

The eigenvalues of the covariance matrix are:

2.8694, 0.7619, 0.1860, 0.1622, 0.1184.

The eigenvector matrix Φ is

-0.0387	0.1312	-0.7779	-0.4134	-0.4531
0.0148	0.9863	0.1185	-0.0314	0.1094
0.9983	-0.0072	-0.0150	-0.0061	-0.0560
0.0123	-0.0951	0.3256	-0.9090	0.2418
0.0398	-0.0305	-0.5240	0.0419	0.8492

In order to compute the online observations $\{A_k\}$, we generated random data vectors $\{\mathbf{x}_k\}$ from the above covariance matrix. We next generated $\{A_k\}$ from $\{\mathbf{x}_k\}$ by

using algorithm (19) with $\gamma_k=1/k$. We compute the covariance matrix A by collecting all 500 samples $\mathbf{x}_k$. We refer to the eigenvectors and eigenvalues computed from this A by a standard numerical analysis method [4] as the *actual values*.

We used the Sanger's algorithm (1), the gradient ascent algorithm (6) and the new conjugate direction algorithm (10) on the random observation sequence $\{A_k\}$. In order to measure the convergence and accuracy of the algorithms, we computed the direction cosine at k^{th} update of each adaptive algorithm as

$$\text{Direction Cosine } (k) = \mathbf{w}_k^T \phi / \| \mathbf{w}_k \| \| \phi \|, \qquad (21)$$

where $\mathbf{w}_k$ is the estimated eigenvector of A at k^{th} update, and ϕ is the actual eigenvector computed from all collected samples by a conventional numerical analysis method.

Figure 1 below shows the iterates of the three algorithms (1), (6) and (10) to compute the first principal eigenvector of A. For the conjugate direction method, the parameters α_k^i and β_k^i are computed from the online data sample A_k by the algorithm described in Section 2.2. It is clear from this figure that the conjugate direction method converges faster than the traditional methods, and also does not require an apriori selection of the gain sequence $\{\eta_k\}$.

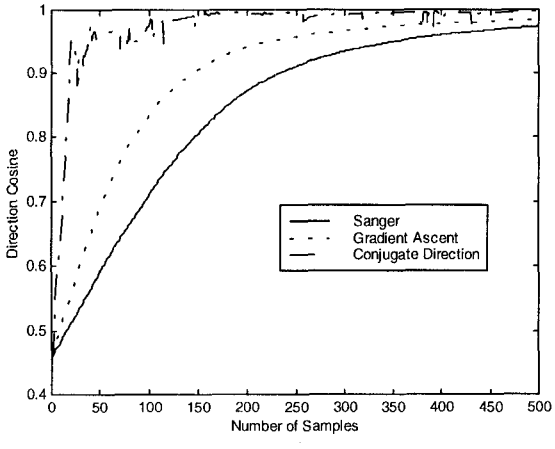

Figure 1. Convergence of the First Principal Eigenvector of A by the Sanger's Algorithm (1), Gradient Ascent Algorithm (6) and the New Conjugate Direction Algorithm (10) with $\eta_k=1/(200+k)$ for the Sanger's and the Gradient Ascent Algorithms.

We observed that increasing the values of $\{\eta_k\}$ leads to a faster convergence of the Sanger's (1) and gradient ascent algorithms (6). In light of this observation, we increased the values of $\{\eta_k\}$ in the Sanger's algorithm (1) and gradient ascent algorithm (6) till $\{\eta_k\}$ reached its upper bound. Figures 2 and 3 shows the convergence plots of all three algorithms for $\eta_k=1/(100+k)$ and $\eta_k=1/(50+k)$.

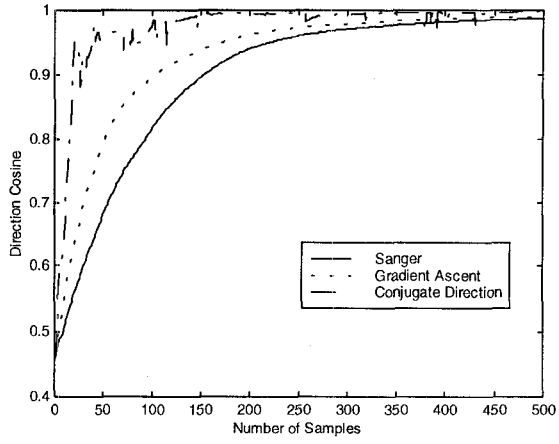

Figure 2. Convergence of the First Principal Eigenvector of A by the Sanger's Algorithm (1), Gradient Ascent Algorithm (6) and the New Conjugate Direction Algorithm (10) with $\eta_k=1/(100+k)$ for the Sanger's and the Gradient Ascent Algorithms.

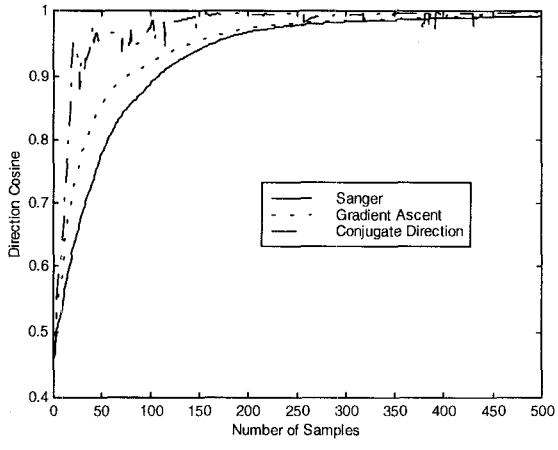

Figure 3. Convergence of the First Principal Eigenvector of A by the Sanger's Algorithm (1), Gradient Ascent Algorithm (6) and the New Conjugate Direction Algorithm (10) with $\eta_k=1/(50+k)$ for the Sanger's and the Gradient Ascent Algorithms.

It is clear from Figures 1-3, that the new PCA algorithm (10) based on the conjugate direction method converges faster than the Sanger's algorithm (1) in spite of a careful selection of η_k for the Sanger's algorithm.

We next compare the performance of the Newton-Rhapson algorithm (13) against the Sanger's algorithm (1) and the gradient ascent algorithm (6) in Figure 4. Clearly, the new Newton-Rhapson algorithm (13)

converges much faster than the Sanger's algorithm (1) and the gradient ascent algorithm (6).

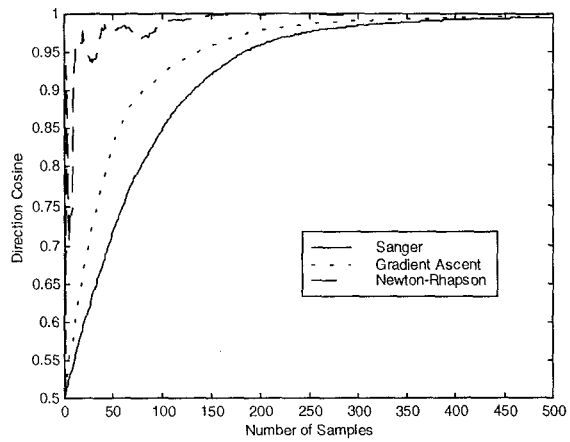

Figure 4. Convergence of the First Principal Eigenvector of A by the Sanger's Algorithm (1), Gradient Ascent Algorithm (6) and the New Newton Rhapson Algorithm (13) with η_k=1/(100+k) for the Sanger's and the Gradient Ascent Algorithms.

4.2 Experiments with Signal Data

We selected an example of digital mobile communications to enhance a signal with respect to noise. The signal from the user is received by 8 antennas and sampled at 0.5 microsecond interval. This produces a data vector $\mathbf{x}_k$ of dimension 8. The solution consists of adaptively computing of the principal eigenvector of the signal correlation matrix A. This vector forms the weights of a transversal filter that enhances the signal with respect to noise.

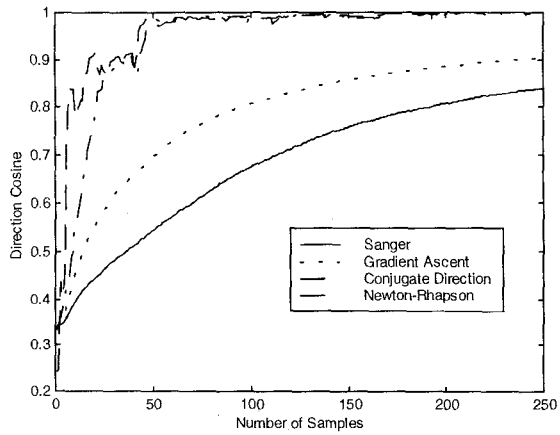

Figure 5. Convergence of the First Principal Eigenvector of the Signal Correlation Matrix A by the Sanger's Alg. (1), Gradient Ascent Algorithm (6), the New Conjugate Direction Algorithm (10), and the New Newton-Rhapson Algorithm (13) with η_k=1/(100+k) for the Sanger's and the Gradient Ascent Algorithms.

The signal consists of 250 sample vectors $\mathbf{x}_k$. The correlation matrix A is computed by averaging all 250 $\mathbf{x}_k\mathbf{x}_k^T$. The eigenvalues of A are

2.5320 2.1226 1.2384 0.9574 0.6088 0.3625 0.1493.

The observation matrices A_k are generated by algorithm (19) with γ_k=1/k. We use the four algorithms (1), (6), (10) and (13) to adaptively compute the principal eigenvector of A. The results of this experiment are shown in Figure 5.

Once again, the new conjugate direction algorithm (10) and the new Newton-Rhapson algorithm (13) converges faster than the Sanger's algorithm (1) and the gradient ascent algorithm (6).

References

[1] P.Baldi and K.Hornik, "Learning in Linear Neural Networks: A Survey", *IEEE Transactions on Neural Networks*, Vol. 6, No. 4, pp. 837-857, 1995.

[2] A.Benveniste, A.Metivier and P.Priouret, *Adaptive Algorithms and Stochastic Approximations*, New York: Springer-Verlag, 1990.

[3] C.Chatterjee, V.P.Roychowdhury, J.Ramos, M.D. Zoltowski, "Self-Organizing Algorithms for Generalized Eigen-Decomposition", to appear in *IEEE Transactions on Neural Networks*.

[4] G.H.Golub and C.F.VanLoan, *Matrix Computations*, Baltimore, MD: Johns Hopkins University Press, 1983.

[5] S.Haykin, *Neural Networks - A Comprehensive Foundation*, Maxwell Macmillan International, New York, 1994.

[6] E.Oja and J.Karhunen, "On Stochastic Approximation of the Eigenvectors and Eigenvalues of the Expectation of a Random Matrix", *Journ. of Math. Anal. Appl.*, Vol. 106, pp. 69-84, 1985.

[7] T.Okada and S.Tomita, "An Optimal Orthonormal System for Discriminant Analysis", *Pattern Recognition*, Vol. 18, No. 2, pp. 139-144, 1985.

[8] T.D.Sanger, "Optimal Unsupervised Learning in a Single-Layer Linear Feedforward Neural Network", *Neural Networks*, Vol. 2, pp. 459-473, 1989.

[9] R.L.Wheeden and A.Zygmund, *Measure and Integral - An Introduction to Real Analysis*, New York: Marcel Dekker, Inc., 1977.

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