An Adaptive Stochastic Approximation Algorithm for Simultaneous Diagonalization of Matrix Sequences With Applications

Chanchal Chatterjee
and Vwani P. Roychowdhury

Abstract—We describe an adaptive algorithm based on stochastic approximation theory for the simultaneous diagonalization of the expectations of two random matrix sequences. Although there are several conventional approaches to solving this problem, there are many applications in pattern analysis and signal detection that require an online (i.e., real-time) procedure for this computation. In these applications, we are given two sequences of random matrices $\{A_k\}$ and $\{B_k\}$ as online observations, with $\lim_{k\to\infty}E[A_k] = A$ and $\lim_{k\to\infty}E[B_k] = B$, where A and B are real, symmetric and positive definite. For every sample (A_k, B_k), we need the current estimates Φ_k and Λ_k respectively of the eigenvectors Φ and eigenvalues Λ of $A^{-1}B$. We have described such an algorithm where Φ_k and Λ_k converge provably with probability one to Φ and Λ respectively. A novel computational procedure used in the algorithm is the adaptive computation of $A^{-1/2}$. Besides its use in the generalized eigen-decomposition problem, this procedure can be used on its own in several feature extraction problems. The performance of the algorithm is demonstrated with an example of detecting a high-dimensional signal in the presence of interference and noise, in a digital mobile communications problem. Experiments comparing computational complexity and performance demonstrate the effectiveness of the algorithm in this real-time application.

Index Terms—Adaptive generalized eigen-decomposition, interference cancellation.

1 Introduction

We describe an adaptive algorithm based on stochastic approximation theory for the simultaneous diagonalization of the expectations of two random matrix sequences. While there are several algorithms for the adaptive computation of eigenvectors of random symmetric matrices [1], adaptive algorithms for simultaneous diagonalization and generalized eigenvector computation are few. However, there are applications of pattern recognition and signal detection where the simultaneous diagonalization of data correlation or covariance matrices are needed. Examples in pattern recognition are dimensionality reduction for classification by several criteria [3] based upon linear discriminant analysis, Bhattacharyya distance measure, divergence, and Chernoff distance measure. Other examples include signal detection in the presence of interference and noise for digital mobile communications and broadcasting [8].

- *C. Chatterjee is with the Newport Corporation, 1791 Deere Ave., Irvine, CA 92606. E-mail: cchatterjee@newport.com.*
- *V.P. Roychowdhury is with the Electrical Engineering Department, University of California, Los Angeles, Los Angeles, CA 90095. E-mail: vwani@ee.ucla.edu*

Manuscript received Dec. 18, 1995; revised Nov. 12, 1996. Recommended for acceptance by K. Boyer.
For information on obtaining reprints of this article, please send e-mail to: transpami@computer.org, and reference IEEECS Log Number P96118.

1.1 Need for Adaptive Algorithms and Systems

The generalized eigen-decomposition problem $B\Phi = A\Phi\Lambda$ consists of evaluating the generalized eigenvector matrix Φ and the generalized eigenvalue matrix Λ. It involves the matrix pair (pencil) (A, B), where A and B are assumed to be real, symmetric and positive definite; commonly referred to as a *symmetric-definite pencil* [4]. In this situation, the generalized eigenvalue problem breaks down to a regular eigenvalue problem for the matrix $A^{-1}B$.

Although a solution to the problem may be obtained by a conventional (numerical analysis) method, there are several applications in pattern analysis, signal detection and automatic control where an online (i.e., in real-time) solution of generalized eigen-decomposition is desired. In these real-time situations, the matrices A and B are themselves unknown. Instead, there are available two sequences of random matrices $\{A_k\}$ and $\{B_k\}$ with $\lim_{k\to\infty}E[A_k] = A$ and $\lim_{k\to\infty}E[B_k] = B$, where A_k and B_k represent the online observations of the application. For every sample (A_k, B_k), we need to obtain the current estimates Φ_k and Λ_k of Φ and Λ respectively, such that Φ_k and Λ_k converge strongly to their true values. Thus, we require a computationally efficient adaptive algorithm involving simple matrix-vector multiplications, that keeps pace with the incoming data.

The conventional approach for evaluating Φ and Λ requires the computation of (A, B) after collecting all of the samples, and then the application of a numerical procedure [4]; i.e., the approach works in a *batch* fashion. There are three problems with this approach. Firstly, the dimension of the samples may be large so that even if all of the samples are available, performing the generalized eigen-decomposition may be difficult or may take prohibitively large amount of computational time; e.g., $O(kd^3)$ computation, where k = number of iterations required by the algorithm, and d = dimension of the samples. Secondly, the conventional schemes can not adapt to slow or small changes in the data (e.g., a few incoming samples). When a new sample (A_k, B_k) is added, it is quite simple to get the corresponding new correlation matrices (A_{new}, B_{new}) as $A_{new} = (nA + A_k)/(n + 1)$ and $B_{new} = (nB + B_k)/(n + 1)$, where n is the total number of samples used to compute (A, B). However, all the computations for solving $B_{new}\Phi = A_{new}\Phi\Lambda$ need to be repeated to obtain Φ and Λ. Thirdly, for the adaptive approach, we may be able to make the computation concurrent to the data acquisition process, and, at every instant, the current estimates Φ_k and Λ_k of Φ and Λ respectively are available. The conventional approach, on the other hand, will not only involve the time delay needed to collect all of the samples to compute A and B, but also involve the subsequent time required to compute $B\Phi = A\Phi\Lambda$. In addition, the conventional approach will not, in general, exploit the fact that, in several applications, the time variation of the optimal Φ and Λ are gradual [8]. So the approach is not suitable for real-time applications where the samples come incrementally or in an *online* fashion.

1.2 Examples of Classification With Adaptive Generalized Eigen-Decomposition

In pattern recognition, there are problems where we are given samples $\mathbf{x}$ and $\mathbf{y} \in \Re^d$ from different populations, and we seek the optimum linear transform $W \in \Re^{dXp}$ $(p \le d)$, such that the scatter of $\mathbf{x}$ in the transformed space is maximized with respect to the corresponding scatter of $\mathbf{y}$. The scatters are usually measured by using the correlation matrices of $\mathbf{x}$ and $\mathbf{y}$ in the transformed space as W^TAW and W^TBW respectively, where A and B are the correlation matrices of $\mathbf{x}$ and $\mathbf{y}$ respectively.

The well-known problem of linear discriminant analysis (LDA) [3] seeks a transform W for samples from a finite set of pattern classes, such that the interclass distance (measured by the scatter

of the patterns around their mixture mean) is maximized, while at the same time the intraclass distance (measured by the scatter of the patterns around their respective class means) is as small as possible. The objective of this transform is to group the classes into well separated clusters. The former scatter matrix, known as the mixture scatter matrix is denoted by B, and the latter matrix, known as the within-class scatter matrix, is denoted by A [3]. When the first column $\mathbf{w}$ of W is needed (i.e., $p = 1$), the problem can be formulated in the constrained optimization framework as

$$\text{Maximize } \mathbf{w}^T B\mathbf{w} \text{ subject to } \mathbf{w}^T A\mathbf{w} = 1. \tag{1}$$

A solution to (1) leads to the generalized eigen-decomposition problem $B\mathbf{w} = \lambda A\mathbf{w}$, where λ is the largest generalized eigenvalue of B with respect to A.

Applications of real-time training and recognition, such as the training and detection of machine printed (e.g., inkjet, hot-stamped and laser-marked) characters in manufacturing lines [2], require an online feature extraction and classification method. Since LDA is a powerful feature extraction tool, our adaptive algorithm is suited to this problem. Off line computation of the LDA transform requires large amounts of storage, and can only consider a finite number of samples. Computing the LDA transform online by the conventional method is slow, and may reduce the manufacturing speed. Adaptive algorithms offer an efficient means of estimating the LDA transform for every sample.

While the LDA solution is used for the separation of a given class in the presence of a set of pattern classes, we next discuss an analogous problem of detecting a desired signal in the presence of interference. Here, we seek the optimum linear transform W for weighting the signal plus interference such that the desired signal is detected with maximum power, and minimum interference. Given the matrix pair (A, B), where B is the correlation matrix of the signal plus interference (plus noise), and A is the correlation matrix of interference (plus noise), we can formulate the signal detection problem as the constrained maximization problem [8] in (1). Here, we maximize the signal power, and minimize the power of the interference. The solution for W consists of the $p \le d$ largest generalized eigenvectors of the matrix pencil (A, B) [8].

As an example of a system where online signal detection using adaptive generalized eigen-decomposition scheme is necessary, we study the problem of signal detection in digital mobile communications. The problem occurs when the desired user transmits a signal from a far distance to the receiver (base), while another user simultaneously transmits very near to the base. For common receivers, the quality of the received signal from the desired user is dominated by interference from the user close to the base [8]. Due to the high rate and large dimension of the transmitted data, the system demands an accurate detection method over a few data samples.

If we use the conventional method, signal detection will require a significant part of the time slot allotted to a receiver, accordingly reducing the effective communication rate. Adaptive generalized eigen-decomposition algorithms, on the other hand, allow the tracking of slow changes in the incoming data [8], and directly performs signal detection, thereby overcoming the conventional method of stringent power control, and improving the capacity of the transmission method. Experiments with high dimension practical data for signal and interference (see Section 4) show the effectiveness of our algorithm.

In summary, we require the following from our adaptive generalized eigen-decomposition algorithm:

1) the algorithm should adapt to small changes in the data;
2) the computation involved in the algorithm is inexpensive such that the statistical procedure can keep pace with the incoming data stream;
3) the estimates obtained from the algorithm should converge strongly (with probability one) to their asymptotic values; and
4) numerical performance with high dimension data for finite number of samples should be comparable to the closed form solution.

1.3 State of the Art Methods for Generalized Eigen-Decomposition

The *conventional method* to solve the generalized eigen-decomposition problem consists of two steps, each involving a symmetric eigenvalue computation. First, we compute the eigenvectors E and eigenvalues Θ of A, from which we obtain $A^{-1/2}$ as $E\Theta^{-1/2}$. Next, we solve a symmetric eigenvalue problem $A^{-1/2}B(A^{-1/2})^T\Psi = \Psi\Lambda$, where $\Psi = (A^{1/2})^T\Phi$. Note that $A^{-1/2}B(A^{-1/2})^T$ is real and symmetric, and Ψ is real. For an orthonormal Ψ, we have $\Psi^T\Psi = \Phi^T A\Phi = I$. Therefore, Φ is real and orthonormal with respect to A. Furthermore, $\Phi^T B\Phi = \Lambda$ which is diagonal, real and positive definite. This is also known as the *simultaneous diagonalization* [4] of matrices A and B. The method requires an iterative procedure to obtain a solution.

A second approach is a recursive method due to Mao and Jain [6]. If we are given two sets of *pooled* or *stored* matrices $\{A_i, i = 1, ..., n_1\}$ and $\{B_i, i = 1, ..., n_2\}$, then the following procedure can be used to compute the eigenvectors of $A^{-1}B$. Using the set $\{A_i, i = 1, ..., n_1\}$, we compute the eigenvectors and eigenvalues of A from which we obtain $A^{-1/2}$. Next, we consider a set $\{A^{-1/2}B_i(A^{-1/2})^T, i = 1, ..., n_2\}$, which is used to compute the eigenvectors of $A^{-1/2}B(A^{-1/2})^T$. Eigenvectors of $A^{-1}B$ are obtained from the two results. In its current formulation, this method can not be used for a sequence or *flow* of data since complete convergence of the eigenvectors and eigenvalues of A are required before the second step can be used. Thus, although useful for high dimension data, the method requires a pooled data for training, which may be unrealistic in many real-time environments. We have given a more direct solution to this problem.

Since the above algorithms can not compute the eigenvectors and eigenvalues of $A^{-1}B$ for a sequence of inputs, we need an adaptive algorithm. In our method, we directly (adaptively) compute a matrix W_k for each sample A_k, where W_k tends to a symmetric positive definite matrix $A^{-1/2}$ with probability one (w.p.1) as $k \to \infty$. Next, we consider a sequence $\{C_k = W_{k-1}B_kW_{k-1}\}$, which is used to adaptively compute a matrix V_k, where V_k tends to the eigenvector matrix of $\lim_{k\to\infty}E[C_k]$ w.p.1 as $k \to \infty$. In this study, we use the Sanger's adaptive eigen-decomposition algorithm [7] for this step, although any other adaptive eigen-decomposition algorithm can be used. In conjunction, the two steps yield W_kV_k, which is proven to converge w.p.1 to Φ as $k \to \infty$. Thus, the two steps can proceed *simultaneously* and converge strongly to the eigenvector matrix Φ.

In applications where the estimates are required after every sample, we demonstrate the computational advantages of adaptive algorithms. In evaluating time complexity, we consider the commonly used form of $A_k = \mathbf{x}_k\mathbf{x}_k^T$ and $B_k = \mathbf{y}_k\mathbf{y}_k^T$ where $\{\mathbf{x}_k\}$ and $\{\mathbf{y}_k\}$ are d-dimensional vector sequences. The conventional method performs a Cholesky decomposition followed by eigen-decomposition, and hence requires $O(kd^3)$ computation, where k is the number of iterations required by the algorithm to converge [4]. In comparison, the adaptive method (see Section 3) uses matrix-vector multiplications, and requires $O(\max(dp^2, d^2))$ computation to evaluate p generalized eigenvectors. It is well-known that matrix-vector multiplication is much faster than eigen-decomposition used in the conventional method. Furthermore, the adaptive algorithms can be easily implemented in commonly available hardware.

The error analyses for the conventional (numerical analysis) methods are well-studied [4]. However, the proof of convergence of our algorithm (see Section 3) demonstrate that the estimates

asymptotically converge to their actual values. The performance of this class of adaptive algorithms for finite samples is a subject of ongoing research [1], [2] and an analysis is beyond the scope of this paper. However, our experiments (see Section 4) indicate that even for a small number of samples in the signal detection example, our algorithm converges to within 1% of the principal generalized eigenvector computed by the conventional method, while maintaining its computational advantages.

1.4 The $A^{-½}$ Algorithm

A key procedure in the generalized eigen decomposition algorithm is the adaptive computation of $A^{-½}$. Besides its application in the current problem, this step can be used, on its own, in several feature extraction problems such as the evaluation of the quadratic discriminant function [2] for Gaussian data. We, therefore, describe a novel algorithm for this computational step, and offer a rigorous proof of convergence. Our method of proof relies on the results given by Ljung [5] concerning with probability one convergence of stochastic approximation algorithms.

In Section 2 we describe the new $A^{-½}$ algorithm with a proof of convergence. Section 3 describes the adaptive algorithm for the computation of eigenvectors and eigenvalues of $A^{-1}B$ from two random matrix sequences. Section 4 has simulation results for a signal detection problem in digital mobile communications. Section 5 has the concluding remarks.

2 Adaptive Computation of $A^{-½}$ and a Stochastic Approximation Proof

Given a sequence of symmetric random matrices $A_k \in \Re^{dXd}$ for k = 1, 2, ... with $\lim_{k\to\infty}E[A_k] = A$, the algorithm for the adaptive computation of $A^{-½}$ is

$$W_k = W_{k-1} + \eta_k\left(I - W_{k-1}A_kW_{k-1}\right), \tag{2}$$

for W_0 symmetric and nonnegative definite. The algorithm for W_0 nonpositive definite is discussed later. In (2), $\{\eta_k\}$ is a scalar gain sequence.

Note that there is no unique solution for $A^{-½}$. Let $A = \Phi\Lambda\Phi^T$ be the eigen decomposition of A, with $\Lambda = \mathrm{diag}(\lambda_1, ..., \lambda_d)$. A solution for $A^{-½}$ is ΦD, where $D = diag\left(\pm\lambda_1^{-1/2}, \ldots, \pm\lambda_d^{-1/2}\right)$. However, in general, this is not a symmetric solution, and for any orthonormal matrix R, ΦDR is also a solution. It can be shown that there are 2^d symmetric solutions of the form $\Phi D\Phi^T$. Defining $\Lambda^{-1/2} = diag\left(\pm\lambda_1^{-1/2}, \ldots, \lambda_d^{-1/2}\right)$, we obtain the unique symmetric and positive definite solution for $A^{-½}$ as $\Phi\Lambda^{-½}\Phi^T$. We shall prove that $W_k \to A^{-½}$ with probability one (w.p.1) as $k \to \infty$, where $A^{-½} = \Phi\Lambda^{-½}\Phi^T$.

In stochastic approximation theory, we study the asymptotic properties of (2) in terms of the ordinary differential equation (ODE) $dW/dt = \lim_{k\to\infty}E[I - WA_kW]$. In particular, we observe that:

1) W_k can converge only to stable stationary points of the ODE;
2) if W_k belongs to a domain of attraction of a stable stationary point W^* infinitely often with probability one (w.p.1), then W_k converges w.p.1 to W^* as $k \to \infty$; and
3) the trajectories of the ODE are the "asymptotic paths" of W_k generated by (2).

The method of proof requires the following steps:

1) establish a set of conditions to be imposed on A, A_k, and η_k,
2) find the stable stationary points of the ODE, and
3) demonstrate that W_k belongs to a compact subset of the domain of attraction of a stable stationary point infinitely often.

Note that for the following lemmas and theorems, all proofs are presented in [2].

DEFINITIONS. *A sequence $\{A_k\}$ is said to be in* general position *if every matrix $\tilde{\mathbf{A}} = [A_k \ldots A_{k+d-1}]$ with d consecutive matrices has rank d. The sequence is in* uniform general position *if the smallest singular value of $\tilde{\mathbf{A}}$ is uniformly bounded away from zero.*

We need the following assumptions:

ASSUMPTION (A1). *Each A_k is bounded w.p.1, symmetric, real and nonnegative definite, with $\lim_{k\to\infty}E[A_k] = A$, where A is positive definite.*

ASSUMPTION (A2). *$\{\eta_k \in \Re^+\}$ is a decreasing sequence such that $\sum_{k=0}^{\infty}\eta_k = \infty$, $\sum_{k=0}^{\infty}\eta_k^r < \infty$, for some $r > 1$, and $\lim_{k\to\infty}\sup\left(\eta_k^{-1} - \eta_{k-1}^{-1}\right) < \infty$.*

ASSUMPTION (A3). *The sequence of matrices A_k for k = 1, 2, ... is in uniform general position.*

We shall use Theorem 1 of Ljung [5] for the convergence proof. The theorem deals with nonlinear stochastic algorithms of the form $W_k = W_{k-1}+\eta_k h(W_{k-1}, A_k)$, which include (2). The assumptions of the Theorem on $h(\cdot)$ are [5].

L1. The function $h(W, A_k)$ is continuously differentiable with respect to W and A_k. The derivatives are, for fixed W and A_k, bounded in k.

L2. The so called *mean vector field* $\bar{h}(W) = \lim_{k\to\infty}E\left[h(W, A_k)\right]$ exists and is regular; i.e., locally Lipschitz. The expectation is with respect to the distribution of A_k for a fixed value of W.

We modify the result of Ljung (Theorem 1) [5] to suit the present algorithm in the following lemma.

LEMMA 1. *Let A1-A3 hold. Let W^* be a locally asymptotically stable (in the sense of Lyapunov) solution to the ordinary differential equation (ODE)*

$$\frac{dW}{dt} = I - WAW \tag{3}$$

with domain of attraction $D(W^*)$. Then if there is a compact subset S of $D(W^*)$ such that $W_k \in S$ infinitely often, then we have $W_k \to W^*$ with probability one as $k \to \infty$.

Let $\lambda_1(W)$ denote the largest, and $\lambda_d(W)$ denote the smallest eigenvalue of W. Let $(\lambda_1(A), ..., \lambda_d(A))$ denote the eigenvalues of A in decreasing order. In the following lemma, we determine the domain of attraction $D(W^*)$ of an asymptotically stable solution W^*, and also show that $\{W_k\}$ visits a compact subset S of $D(W^*)$ infinitely often.

LEMMA 2. *Let Φ and Λ respectively denote the eigenvector and eigenvalue matrices of A. Let A1-A3 hold. Then for (2), the following hold:*

1) *The point $W^* = \Phi\Lambda^{-½}\Phi^T$ is (uniformly) asymptotically stable.*
2) *The domain of attraction of W^* includes*

$$D(W^*) = \{W \in \Re^{dXd}: W = W^T \text{ and } \lambda_d(W) > -\lambda_1(A)^{-½}\}. \tag{4}$$

3) *There exists a uniform upper bound for $\lambda_1(W_k)$ for all k.*
4) *There exists a uniform upper bound for η_k such that $\lambda_d(W_k) \geq 0$ uniformly for all k.*

The lemma also gives an upper bound for $\lambda_1(W_0)$ that we need to satisfy at the start of the algorithm.

The convergence of algorithm (2) can now be stated as a direct corollary of the above lemmas.

THEOREM 1. *Let A1-A3 hold. Let W_0 and η_0 be within the uniform upper bounds stated in Lemma 2. If W_0 is assigned random weights such that W_0 is symmetric and nonnegative definite, then with probability one, algorithm (2) will converge, and $W_k \to W^*$ as $k \to \infty$, where $W^* = \Phi\Lambda^{-½}\Phi^T$ is the unique symmetric positive definite solution for $A^{-½}$.*

Unlike (2), if W_0 is nonpositive definite, then we change (2) as follows

$$W_k = W_{k-1} + \eta_k\left(W_{k-1}A_kW_{k-1} - I\right), \tag{5}$$

for W_0 symmetric and nonpositive definite. This algorithm is discussed in [2].

3 ADAPTIVE ALGORITHM FOR EIGENVECTORS AND EIGENVALUES OF $A^{-1}B$

In this section, we shall describe an adaptive algorithm for the computation of the eigenvector matrix Φ and eigenvalue matrix Λ for $A^{-1}B$, from two random matrix sequences $\{A_k\}$ and $\{B_k\}$. We assume that $\{A_k\}$ satisfies assumptions A1-A3 described in Section 2. The assumptions on $\{B_k\}$ are

ASSUMPTION (A4). *Each B_k is real, symmetric, and bounded w.p.1, such that $\lim_{k\to\infty} E\left[B_k\right] = B$, where B is positive definite.*

Note that if B is assumed nonpositive definite, then a simple transform $B + cA$ where c is a positive scalar constant will transform B to a positive definite matrix. It is easy to show that both B and $B + cA$ have the same generalized eigenvectors with respect to A, with eigenvalues λ_i and $\lambda_i + c$ respectively.

As described before, the algorithm consists of two steps. The first step estimates $A^{-½}$ with the new algorithm from an input sequence $\{A_k\}$. The second step estimates the eigenvector matrix $\Psi = A^{½}\Phi$ of the symmetric matrix $A^{-½}BA^{-½}$ from a new sequence obtained from $\{B_k\}$ and the *current estimate of* $A^{-½}$ from the first step. Combining the two steps, we obtain an estimate of $A^{-½}A^{½}\Phi = \Phi$. Thus, the two steps can proceed simultaneously, and together they converge strongly to Φ.

In the second step, we use the Sanger's algorithm [7] modified to the current context. Note, however, that we can use any other algorithm for eigenvector computation. The reasons for using Sanger's algorithm are:

1) it converges for random starting values and for a wide choice of gains $\{\gamma_k\}$ (see (6)),
2) it computes the eigenvectors ordered by decreasing eigenvalue, and
3) it can be relatively easily implemented with only local operations [7].

Let $\{C_k\}$ denote a sequence of symmetric real random matrices with $\lim_{k\to\infty}E[C_k] = C$. The modified Sanger's algorithm is

$$V_k = V_{k-1} + \gamma_k\left(C_kV_{k-1} - V_{k-1}\mathrm{UT}\left[V_{k-1}^TC_kV_{k-1}\right]\right), \tag{6}$$

where UT[·] sets all elements below the diagonal of its matrix argument to zero, thereby making the matrix upper triangular, and $\{\gamma_k\}$ satisfies assumption A2. Due to the convergence of the Sanger's algorithm [7], if V_0 is assigned random weights, V_k will converge w.p.1 to a matrix whose columns are the eigenvectors of C, ordered by decreasing eigenvalue.

The stochastic approximation algorithm for the computation of the eigenvector matrix Φ of $A^{-1}B$, from random matrix sequences $\{A_k\}$ and $\{B_k\}$ is as follows:

Step 1. Use algorithm (2) or (5) with input $\{A_k\}$, where $W_k \to A^{-½}$ w.p.1 as $k \to \infty$.
Step 2. For each k, compute $C_k = W_{k-1}B_kW_{k-1}$.
Step 3. Use algorithm (6) with input $\{C_k\}$, where V_k tends to the eigenvectors of $\lim_{k\to\infty}E[C_k] = A^{-½}BA^{-½}$ w.p.1 as $k \to \infty$.
Step 4. For each k, compute the eigenvector matrix $\Phi_k = W_kV_k$.

Notice that the eigenvectors of $A^{-1}B$ are ordered by decreasing eigenvalue.

The following theorem and discussion presents the proof of convergence of the entire algorithm.

THEOREM 2. *Let A1-A4 hold, and $\{\gamma_k\}$ satisfy A2. If V_0 is assigned random weights, then for the input sequence $\{C_k\}$ algorithm (6) will converge, and $V_k \to \Psi$ w.p.1 as $k \to \infty$, where $\Psi = A^{½}\Phi$.*

From the convergence of W_k and V_k, we obtain $\lim_{k\to\infty}\Phi_k = \lim_{k\to\infty}W_k \lim_{k\to\infty}V_k = A^{-½}A^{½}\Phi = \Phi$ w.p.1.

Instead of the eigenvector matrix Φ of $A^{-1}B$, if the eigenvalues λ_i for $i = 1, ..., d$, of $A^{-1}B$ are required, we use the following stochastic approximation algorithm

$$\theta_i(k) = \theta_i(k-1) + \delta_k\left(\phi_i(k-1)^TB_k\phi_i(k-1) - \theta_i(k-1)\right)$$
$$\text{for } i = 1, \ldots, d. \tag{7}$$

We can prove (see [2]) that $\theta_i(k) \to \lambda_i$ w.p.1 as $k \to \infty$ for $i = 1, ..., d$.

4 EXPERIMENTAL RESULTS

In our experiments, we use a signal detection problem in digital mobile communications. The problem occurs when the desired user transmits a signal from a far distance to the receiver (base), while another user simultaneously transmits very near to the base providing significant interference to the desired signal. The solution involves the design of a system that efficiently and accurately detects the desired signal in the presence of interference and receiver noise. An adaptive signal detection solution overcomes the conventional method of stringent power control, and improves the capacity of the transmission method. A particular method that is proven effective [8] is adopted here.

4.1 Data Model

In this application the duration of each transmitted code from the desired user is t_d, and the interference from other users is of duration t_i. The total duration of each transmitted code that is received at the base is $t_d + t_i = t$, which is known as a *bit period*. In most applications, t_i is much larger than t_d. In order to detect the signal accurately, a common method is to receive the signal with m antennas [8]. Hence, at any instant, we have m signals, and we can define two correlation matrices—signal correlation matrix B over duration t_d, and interference correlation matrix A over duration t_i. It can be shown [8] that the optimum weight matrix W that maximizes the signal power (leading to accurate signal detection) consists of the $p < m$ generalized eigenvectors of B with respect to A corresponding to the p largest generalized eigenvalues. For a practical system, our goal is to detect the signal (i.e., determine W) in as few bits as possible. For the state of the art, accurate detection in 5-15 bits (depending on the dimensionality of the signal) is considered acceptable [8]. We use our adaptive procedure to estimate W from a sequence of samples by adopting two separate schemes for obtaining the data sequence.

SCHEME 1. Here we time sample the signal from each antenna over the entire bit period t. For each antenna, we obtain n_d samples of the desired signal plus interference, and n_i samples of the interference from other users for the bit period t. Due to m antennas, and due to the real and imaginary components of the transmitted code, for each time sample, we ob-

tain a data vector of dimension $d = 2m$. Thus, for each bit period, we obtain n_d signal vectors $\mathbf{y}_k$ (from the desired user) giving us n_d signal correlation matrices $B_k = \mathbf{y}_k\mathbf{y}_k^T$, and n_i interference vectors $\mathbf{x}_k$ (from other users) giving us n_i interference correlation matrices $A_k = \mathbf{x}_k\mathbf{x}_k^T$. We start the algorithm after the first signal and interference vectors are received.

SCHEME 2. Here we take f frequency samples for the duration t_d of the desired signal. The frequency samples from all m antennas are concatenated to obtain one desired signal vector $\mathbf{y}_k$ with the corresponding frequency domain correlation matrix $B_k = \mathbf{y}_k\mathbf{y}_k^T$. Due to the real and imaginary components of the signal, the dimension of the signal vector is $d = 2fm$. We repeat this process over the duration t_i of the interference by taking f frequency samples over a sliding interval of duration t_d. Since t_i is usually much larger than t_d, we obtain several interference samples for each bit period, each of dimension $d = 2fm$. Thus, for each signal vector $\mathbf{y}_k$ (here, $n_d = 1$ for each bit period), we obtain n_i interference vectors $\mathbf{x}_k$ giving us n_i interference correlation matrices $A_k = \mathbf{x}_k\mathbf{x}_k^T$.

4.2 Numerical Results for Scheme 1

We obtained numerical data from a practical mobile communications setup described in [8]. In this example, the bit period $t = 127\mu s$ with $t_d = 10\mu s$ and $t_i = 117\mu s$. The number of antennas $m = 8$ giving us data vectors of dimension $d = 16$. Here, we sampled the signal at $0.5\mu s$ interval. We obtained $n_d = 20$ signal vectors $\mathbf{y}_k$, and $n_i = 234$ interference vectors $\mathbf{x}_k$.

We first computed the signal and interference correlation matrices B and A respectively, by averaging all B_ks and A_ks collected over five bit periods. We shall refer to the generalized eigenvectors and eigenvalues computed from this B and A matrices as the *actual values*. The four largest generalized eigenvalues of B with respect to A are 25.88, 14.76, 1.80, and 1.60. Clearly, the first two generalized eigenvalues and the corresponding generalized eigenvectors are important. We used the adaptive algorithm to compute the first and second generalized eigenvectors and eigenvalues of B with respect to A. The results are given in Fig. 1. In order to measure the accuracy of the estimated generalized eigenvectors, we computed the direction cosine given by

$$\text{Direction Cosine} = \left|\mathbf{w}_k^T\phi\right| / \left\|\mathbf{w}_k\right\| \left\|\phi\right\|,$$

where $\mathbf{w}_k$ is the estimated generalized eigenvector at kth recursion of the adaptive algorithm, and ϕ is the actual generalized eigenvector computed from all collected samples by the conventional method.

Fig. 1 shows that the estimated first generalized eigenvector converged to 95.4% of its actual value in two bit periods, and to 99.0% in five bit periods. The first generalized eigenvalue converged to 25.278 (i.e., 97.7% of its actual value) in just two bit periods, and to 25.708 (i.e., 99.3% of its actual value) in five bit periods. The second generalized eigenvector converged to 98.6% of its actual value in two bit periods, and to 99.8% of its actual value in five bit periods. The second generalized eigenvalue converged to 14.181 (i.e., 96.1% of its actual value) in two bit periods, and to 14.732 (i.e., 99.8% of its actual value) in five bit periods.

We also compared computational time for the conventional method with each recursion of the adaptive method by counting the MATLAB *flops* (floating point operations). The MATLAB algorithm required 80,882 flops for generalized eigen-decomposition and sorting by decreasing eigenvalue. In comparison, by using $B_k = \mathbf{y}_k\mathbf{y}_k^T$ and $A_k = \mathbf{x}_k\mathbf{x}_k^T$, one recursion of the adaptive algorithm needed 6,432 flops for the first and second generalized eigenvector computation.

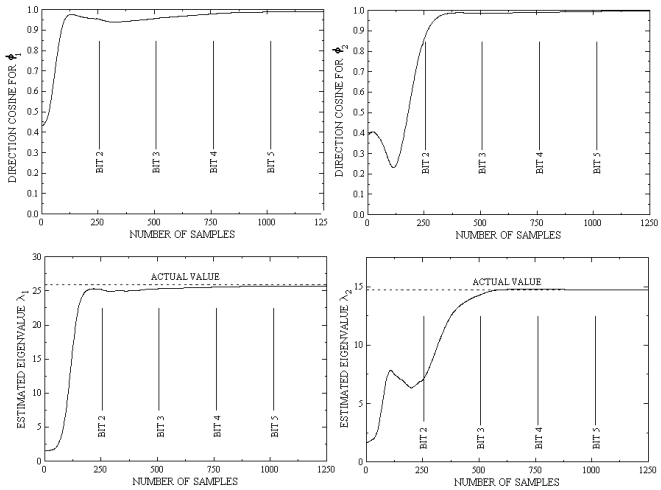

Fig. 1. Convergence of first and second estimated generalized eigenvectors and eigenvalues for 16-dimensional signal data.

4.3 Numerical Results for Scheme 2

Here we take $f = 9$ frequency samples equally spaced between –0.4MHz to +0.4MHz for the $t_d = 10\mu s$ microseconds of the signal duration. For each bit period, we obtain $n_d = 1$ signal vector $\mathbf{y}_k$ of dimension $d = 144$, and $n_i = 72$ interference vectors $\mathbf{x}_k$ also of dimension 144.

We collected all signal and interference samples for 16 bit periods, and computed the signal and interference correlation matrices B and A respectively. The four largest generalized eigenvalues of B with respect to A are 3.3927, 0.2003, 0.1035, and 0.0981. Although the first generalized eigenvalue and the corresponding generalized eigenvector are important, we computed the first two generalized eigenvectors and eigenvalues. The results are shown in Fig. 2.

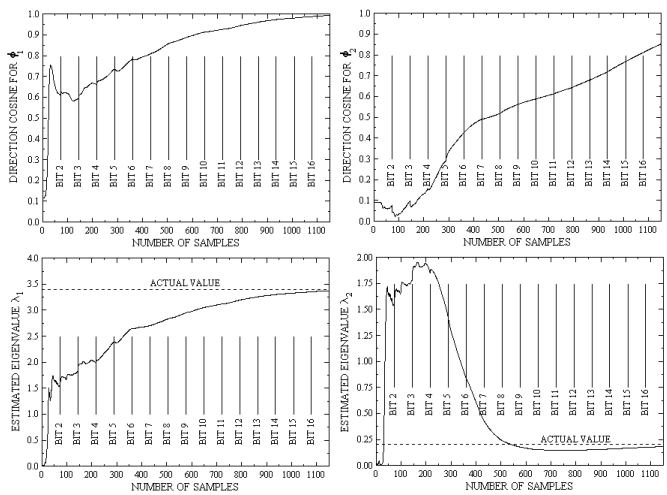

Fig. 2. Convergence of first and second estimated generalized eigenvectors and eigenvalues for 144-dimensional signal data.

We see (from Fig. 2) that the first generalized eigenvector converged to 92.6% of its actual value in 10 bit periods, to 98.0% in 14 bit periods, and to 99.3% in 16 bit periods. The first generalized eigenvalue converged to 3.1174 (91.9% of its actual value) in 10 bit

periods, to 3.3257 (98% of its actual value) in 14 bit periods, and to 3.3739 (99.4% of its actual value) in 16 bit periods. The second generalized eigenvector converged to 85.3% of its actual value in 16 bit periods. The second generalized eigenvalue converged to 0.1800 (90% of its actual value) in 16 bit periods.

The MATLAB algorithm required 48,388,500 flops for generalized eigen-decomposition and sorting by decreasing eigenvalue. By using $B_k = \mathbf{y}_k \mathbf{y}_k^T$ and $A_k = \mathbf{x}_k \mathbf{x}_k^T$, one recursion of the adaptive algorithm required 343,873 flops for the first (principal) generalized eigenvector estimation, and 472,609 flops for the first two generalized eigenvectors estimation.

5 Concluding Remarks

We described an adaptive algorithm for the estimation of the generalized eigenvectors and eigenvalues of the symmetric-definite matrix pencil (A, B) from a sequence of samples $\{A_k\}$ and $\{B_k\}$. The method is useful in applications that require an efficient generalized eigenvector evaluation for every sample (A_k, B_k). A proof of convergence of the algorithm is given by using stochastic approximation theory. The usefulness of the algorithm is demonstrated by detecting a high-dimensional signal in the presence of interference and noise, in a digital mobile communications problem. Experiments comparing computational complexity and performance show the effectiveness of the algorithm over conventional methods in this real-time application.

Acknowledgments

The authors wish to thank Prof. M.D. Zoltowski and Dr. J. Ramos for providing the experimental data. This work was supported in part by National Science Foundation grants no. ECS-9308814 and no. ECS-9523423.

References

[1] P.F. Baldi and K. Hornik, "Learning in Linear Neural Networks: A Survey," *IEEE Trans. Neural Networks*, vol. 6, no. 4, pp. 837-858, 1995.

[2] C. Chatterjee, "Adaptive Self-Organizing Neural Networks for Matrix Eigen-Decomposition Problems and their Applications to Feature Extraction," PhD dissertation, Purdue Univ., School of Electrical Engineering, West Lafayette, Ind., May 1996.

[3] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Second ed. New York: Academic Press, 1990.

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

[5] L. Ljung, "Analysis of Recursive Stochastic Algorithms," *IEEE Trans. Automatic Control*, vol. 22, no. 4, pp. 551-575, Aug. 1977.

[6] J. Mao and A.K. Jain, "Artificial Neural Networks for Feature Extraction and Multivariate Data Projection," *IEEE Trans. Neural Networks*, vol. 6, no. 2, pp. 296-316, 1995.

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

[8] M.D. Zoltowski, J. Ramos, C. Chatterjee, and V.P. Roychowdhury, "Blind Adaptive 2D RAKE Receiver for DS-CDMA Based on Space-Frequency MVDR Processing," unpublished paper.