RECURSIVE PRINCIPAL COMPONENTS ANALYSIS USING EIGENVECTOR MATRIX PERTURBATION

Deniz Erdogmus, Yadunandana N. Rao, Hemanth Peddaneni, Anant Hegde, Jose. C. Principe
CNEL, ECE Department, University of Florida, Gainesville, FL 32611, USA

Abstract. Principal components analysis is an important and well-studied subject in statistics and signal processing. The literature has an abundance of algorithms for solving this problem, where most of these algorithms could be grouped into one of the following three approaches: adaptation based on Hebbian updates and deflation, optimization of a second order statistical criterion (like reconstruction error or output variance), and fixed point update rules with deflation. In this paper, we take a completely different approach that avoids deflation and the optimization of a cost function using gradients. The proposed method updates the eigenvector and eigenvalue matrices simultaneously with every new sample such that the estimates approximately track their true values as would be calculated from the current sample estimate of the data covariance matrix. The performance of this algorithm is compared with that of traditional methods like Sanger's rule and APEX, as well as a structurally similar matrix perturbation-based method.

1. INTRODUCTION

Principal component analysis (PCA) is a well-known statistical technique that has been widely applied to solve important signal-processing problems like feature extraction, signal estimation, detection and speech separation [1-4]. Many analytical techniques exist, which can solve PCA once the entire input data is known [5]. However, most of the analytical methods require extensive matrix operations and hence they are unsuited for real-time applications. Further, in many applications such as direction of arrival (DOA) tracking, adaptive subspace estimation, etc., signal statistics change over time rendering the block methods virtually unacceptable. In such cases, fast, adaptive, on-line solutions are desirable. Majority of the existing algorithms for PCA are based on standard gradient procedures [2,3,6-9], which are extremely slow converging, and their performance heavily depends on step-sizes used. To alleviate this, subspace methods have been explored [10-12]. However, many of these subspace techniques are computationally intensive. The recently proposed fixed-point PCA algorithm [13] showed fast convergence with little or no change in complexity compared with gradient methods. However, this method and most of the existing methods in literature rely on using the standard deflation technique, which brings in sequential convergence of principal components that potentially reduces the overall speed of convergence. We recently explored a simultaneous principal component extraction algorithm called SIPEX [14] which reduced the gradient search only to the space of orthonormal matrices by using Givens rotations. Although SIPEX resulted in fast and simultaneous convergence of all principal components, the algorithm suffered from high computational complexity due to the involved trigonometric function evaluations. A recently proposed alternative approach suggested iterating the eigenvector estimates using a first order matrix perturbation formalism for the sample covariance estimate with every new sample obtained in real time [15]. However, the performance (speed and accuracy) of this algorithm is hindered by the general Toeplitz structure of the perturbed covariance matrix. In this paper, we will present an algorithm that undertakes a similar perturbation approach, but in contrast, the covariance matrix will be decomposed into its eigenvectors and eigenvalues at all times, which will reduce the perturbation step to be employed on the diagonal eigenvalue matrix. This further restriction of structure, as expected, alleviates the difficulties encountered in the operation of the previous first order perturbation algorithm, resulting in a fast converging and accurate subspace tracking algorithm.

This paper is organized as follows. First, we present a brief definition of the PCA problem to have a self-contained paper. Second, the proposed recursive PCA algorithm (RPCA) is motivated, derived, and extended to non-stationary and complex-valued signal situations. Next, a set of computer experiments is presented to demonstrate the convergence speed and accuracy characteristics of RPCA. Finally, we conclude the paper with remarks and observations about the algorithm.

2. PROBLEM DEFINITION

PCA is a well-known problem and it is extensively studied in the literature as we have pointed out in the introduction. However, for the sake of completeness, we will provide a brief definition of the problem in this section. For simplicity and without loss of generality, we will consider a real-valued zero-mean, n-dimensional random vector $\mathbf{x}$ and its n projections $y_1,\ldots,y_n$ such that $y_j = \mathbf{w}_j^T \mathbf{x}$, where $\mathbf{w}_j$'s are unit-norm

vectors defining the projection dimensions in the n-dimensional input space.

The first principal component direction is defined as the solution to the following constrained optimization problem, where $\mathbf{R}$ is the input covariance matrix:

$$\mathbf{w}_1 = \arg\max_{\mathbf{w}} \mathbf{w}^T \mathbf{R}\mathbf{w} \quad \text{subject to} \quad \mathbf{w}^T\mathbf{w} = 1 \qquad (1)$$

The subsequent principal components are defined by including additional constraints to the problem that enforce the orthogonality of the sought component to the previously discovered ones:

$$\mathbf{w}_j = \arg\max_{\mathbf{w}} \mathbf{w}^T\mathbf{R}\mathbf{w}, \text{s.t.}\, \mathbf{w}^T\mathbf{w} = 1, \mathbf{w}^T\mathbf{w}_l = 0, l < j \quad (2)$$

The overall solution to this problem turns out to be the eigenvector matrix of the input covariance $\mathbf{R}$. In particular, the principal component directions are given by the eigenvectors of $\mathbf{R}$ arranged according to their corresponding eigenvalues (largest to smallest) [5].

In signal processing applications, the needs are different. The input samples are usually acquired one at a time (i.e., sequentially as opposed to in batches), which necessitates sample-by-sample update rules for the covariance and its eigenvector estimates. In this setting, this analytical solution is of little use, since it is not practical to update the input covariance estimate and solve a full eigendecomposition problem per sample. However, utilizing the recursive structure of the covariance estimate, it is possible to come up with a recursive formula for the eigenvectors of the covariance as well. This will be described in the next section.

3. RECURSIVE PCA DESCRIPTION

Suppose a sequence of n-dimensional zero-mean wide-sense stationary input vectors $\mathbf{x}_k$ are arriving, where k is the sample (time) index. The sample covariance estimate at time k for the input vector is[1]

$$\mathbf{R}_k = \frac{1}{k}\sum_{i=1}^{k} \mathbf{x}_i\mathbf{x}_i^T = \frac{(k-1)}{k}\mathbf{R}_{k-1} + \frac{1}{k}\mathbf{x}_k\mathbf{x}_k^T \qquad (3)$$

Let $\mathbf{R}_k = \mathbf{Q}_k\mathbf{\Lambda}_k\mathbf{Q}_k^T$ and $\mathbf{R}_{k-1} = \mathbf{Q}_{k-1}\mathbf{\Lambda}_{k-1}\mathbf{Q}_{k-1}^T$, where $\mathbf{Q}$ and $\mathbf{\Lambda}$ denote the orthonormal eigenvector and diagonal eigenvalue matrices respectively. Also define $\boldsymbol{\alpha}_k = \mathbf{Q}_{k-1}^T\mathbf{x}_k$. Substituting these definitions in (3), we obtain the following recursive formula for the eigenvectors and eigenvalues:

$$\mathbf{Q}_k(k\mathbf{\Lambda}_k)\mathbf{Q}_k^T = \mathbf{Q}_{k-1}[(k-1)\mathbf{\Lambda}_{k-1} + \boldsymbol{\alpha}_k\boldsymbol{\alpha}_k^T]\mathbf{Q}_{k-1}^T \quad (4)$$

[1] In practice, if the samples are not generated by a zero-mean process, a running sample mean estimator could be employed to compensate for this fact. Then this biased estimator can be replaced by the unbiased version and the following derivations can be modified accordingly.

Clearly, if we can determine the eigendecomposition of the matrix $[(k-1)\mathbf{\Lambda}_{k-1} + \boldsymbol{\alpha}_k\boldsymbol{\alpha}_k^T]$, which is denoted by $\mathbf{V}_k\mathbf{D}_k\mathbf{V}_k^T$, where $\mathbf{V}$ is orthonormal and $\mathbf{D}$ is diagonal, then (4) becomes

$$\mathbf{Q}_k(k\mathbf{\Lambda}_k)\mathbf{Q}_k^T = \mathbf{Q}_{k-1}\mathbf{V}_k\mathbf{D}_k\mathbf{V}_k^T\mathbf{Q}_{k-1}^T \qquad (5)$$

By direct comparison, the recursive update rules for the eigenvectors and the eigenvalues are determined to be:

$$\begin{aligned}\mathbf{Q}_k &= \mathbf{Q}_{k-1}\mathbf{V}_k \\ \mathbf{\Lambda}_k &= \mathbf{D}_k / k\end{aligned} \qquad (6)$$

In spite of the fact that the matrix $[(k-1)\mathbf{\Lambda}_{k-1} + \boldsymbol{\alpha}_k\boldsymbol{\alpha}_k^T]$ has a special structure much simpler than that of a general covariance matrix, determining the eigendecomposition $\mathbf{V}_k\mathbf{D}_k\mathbf{V}_k^T$ analytically is difficult. However, especially if k is large, the problem can be solved in a simpler way using a matrix perturbation analysis approach. This will be described next.

3.1. Perturbation Analysis for Rank-One Update

When k is large, the matrix $[(k-1)\mathbf{\Lambda}_{k-1} + \boldsymbol{\alpha}_k\boldsymbol{\alpha}_k^T]$ is strongly diagonally dominant; hence (due to the Gershgorin theorem) its eigenvalues will be close to those of the diagonal portion $(k\text{-}1)\mathbf{\Lambda}_{k-1}$. In addition, its eigenvectors will also be close to identity (i.e., the eigenvectors of the diagonal portion of the sum).

In summary, the problem reduces to finding the eigendecomposition of a matrix in the form $(\mathbf{\Lambda}+\boldsymbol{\alpha}\boldsymbol{\alpha}^T)$, i.e. a rank-one update on a diagonal matrix $\mathbf{\Lambda}$, using the following approximations: $\mathbf{D}=\mathbf{\Lambda}+\mathbf{P}_{\mathbf{\Lambda}}$ and $\mathbf{V}=\mathbf{I}+\mathbf{P}_{\mathbf{V}}$, where $\mathbf{P}_{\mathbf{\Lambda}}$ and $\mathbf{P}_{\mathbf{V}}$ are small perturbation matrices. The eigenvalue perturbation matrix $\mathbf{P}_{\mathbf{\Lambda}}$ is naturally diagonal. With these definitions, when $\mathbf{V}\mathbf{D}\mathbf{V}^T$ is expanded, we get

$$\begin{aligned}\mathbf{V}\mathbf{D}\mathbf{V}^T &= (\mathbf{I}+\mathbf{P}_{\mathbf{V}})(\mathbf{\Lambda}+\mathbf{P}_{\mathbf{\Lambda}})(\mathbf{I}+\mathbf{P}_{\mathbf{V}})^T \\ &= \mathbf{\Lambda} + \mathbf{\Lambda}\mathbf{P}_{\mathbf{V}}^T + \mathbf{P}_{\mathbf{\Lambda}} + \mathbf{P}_{\mathbf{\Lambda}}\mathbf{P}_{\mathbf{V}}^T + \mathbf{P}_{\mathbf{V}}\mathbf{\Lambda} \\ &\quad + \mathbf{P}_{\mathbf{V}}\mathbf{\Lambda}\mathbf{P}_{\mathbf{V}}^T + \mathbf{P}_{\mathbf{V}}\mathbf{P}_{\mathbf{\Lambda}} + \mathbf{P}_{\mathbf{V}}\mathbf{P}_{\mathbf{\Lambda}}\mathbf{P}_{\mathbf{V}}^T \\ &= \mathbf{\Lambda} + \mathbf{P}_{\mathbf{\Lambda}} + \mathbf{D}\mathbf{P}_{\mathbf{V}}^T + \mathbf{P}_{\mathbf{V}}\mathbf{D} \\ &\quad + \mathbf{P}_{\mathbf{V}}\mathbf{\Lambda}\mathbf{P}_{\mathbf{V}}^T + \mathbf{P}_{\mathbf{V}}\mathbf{P}_{\mathbf{\Lambda}}\mathbf{P}_{\mathbf{V}}^T\end{aligned} \qquad (7)$$

Equating (7) to $\mathbf{\Lambda} + \boldsymbol{\alpha}\boldsymbol{\alpha}^T$, and assuming that the terms $\mathbf{P}_{\mathbf{V}}\mathbf{\Lambda}\mathbf{P}_{\mathbf{V}}^T$ and $\mathbf{P}_{\mathbf{V}}\mathbf{P}_{\mathbf{\Lambda}}\mathbf{P}_{\mathbf{V}}^T$ are negligible, we get

$$\boldsymbol{\alpha}\boldsymbol{\alpha}^T = \mathbf{P}_{\mathbf{\Lambda}} + \mathbf{D}\mathbf{P}_{\mathbf{V}}^T + \mathbf{P}_{\mathbf{V}}\mathbf{D} \qquad (8)$$

The orthonormality of $\mathbf{V}$ brings an additional equation that characterizes $\mathbf{P}_{\mathbf{V}}$. Substituting $\mathbf{V}=\mathbf{I}+\mathbf{P}_{\mathbf{V}}$ in $\mathbf{V}\mathbf{V}^T=\mathbf{I}$, and assuming that $\mathbf{P}_{\mathbf{V}}\mathbf{P}_{\mathbf{V}}^T \approx \mathbf{0}$, we have $\mathbf{P}_{\mathbf{V}} = -\mathbf{P}_{\mathbf{V}}^T$.

Combining the fact that the eigenvector perturbation matrix $\mathbf{P}_{\mathbf{V}}$ is anti-symmetric with the fact that $\mathbf{P}_{\mathbf{\Lambda}}$ and $\mathbf{D}$ are diagonal, the solution for the

perturbation matrices are found from (8) as follows: the i^{th} diagonal entry of $\mathbf{P}_{\mathbf{\Lambda}}$ is α_i^2 and the $(i,j)^{\text{th}}$ entry of $\mathbf{P}_{\mathbf{V}}$ is $\alpha_i\alpha_j/(\lambda_j+\alpha_j^2-\lambda_i-\alpha_i^2)$ if $j\neq i$, and 0 if $j=i$.

3.2. The Recursive PCA Algorithm

The RPCA algorithm is summarized in Table 1. There are a few of practical issues regarding the operation of the algorithm, which will be addressed in this subsection.

Selecting the memory depth parameter: In a stationary situation, where we would like to weight each individual sample equally, this parameter must be set to $\lambda_k=1/k$. In this case, the recursive update for the covariance matrix is as shown in (3). In a non-stationary environment, a first-order dynamical forgetting strategy could be employed by selecting a fixed decay rate. Setting $\lambda_k=\lambda$ corresponds to the following recursive covariance update equation:

$$\mathbf{R}_k=(1-\lambda)\mathbf{R}_k+\lambda\mathbf{x}_k\mathbf{x}_k^T \tag{9}$$

Typically, in this forgetting scheme, $\lambda\in(0,1)$ is selected to be very small. Considering that the average memory depth of this recursion is $1/\lambda$ samples, the selection of this parameter presents a trade-off between tracking capability and estimation variance.

Initializing the eigenvectors and the eigenvalues: The natural way to initialize the eigenvector matrix $\mathbf{Q}_0$ and the eigenvalue matrix $\mathbf{\Lambda}_0$ is to use the first N_0 samples to obtain an unbiased estimate of the covariance matrix and determine its eigendecomposition ($N_0>n$). The iterations in step 2 can then be applied to the following samples. This means in step 2, $k=N_0+1,\ldots,N$. In the stationary case ($\lambda_k=1/k$), this means in the first few iterations of step 2, the perturbation approximations will be least accurate (compared to the subsequent iterations). This is simply due to $(1-\lambda_k)\mathbf{\Lambda}_{k-1}+\lambda_k\boldsymbol{\alpha}_k\boldsymbol{\alpha}_k^T$ not being strongly diagonally dominant for small values of k. Compensating the errors induced in the estimations at this stage might require a large number of samples later on.

This problem could be avoided if in the iteration stage (step 2), the index k could be started from a large initial value. In order to achieve this without introducing any bias to the estimates, one needs to use a large number of samples in the initialization (i.e., choose a large N_0). In practice, however, this is undesirable. The alternative is to perform the initialization still using a small number of samples (i.e., a small N_0), but setting the memory depth parameter to $\lambda_k=1/(k+(\tau-1)N_0)$. This way, when the iterations start at sample $k=N_0+1$, the algorithm *thinks* that the initialization is actually performed using $\gamma=\tau N_0$ samples. Therefore, from the point-of-view of the algorithm, the data set looks like

$$\underbrace{\{\mathbf{x}_1,\ldots,\mathbf{x}_{N_0}\},\ldots,\{\mathbf{x}_1,\ldots,\mathbf{x}_{N_0}\}}_{\text{repeated }\tau\text{ times}},\{\mathbf{x}_{N_0+1},\ldots,\mathbf{x}_N\} \tag{10}$$

The corresponding covariance estimator is then naturally biased. At the end of the iterations, the estimated covariance matrix is

$$\mathbf{R}_{N,\,biased}=\frac{N}{N+(\tau-1)N_0}\mathbf{R}_N+\frac{(\tau-1)N_0}{N+(\tau-1)N_0}\mathbf{R}_{N_0} \tag{11}$$

where $\mathbf{R}_M=(1/M)\sum_{j=1}^{M}\mathbf{x}_j\mathbf{x}_j^T$. Consequently, we conclude that, the bias introduced to the estimation by tricking the algorithm can be asymptotically diminished (as $N\to\infty$).

In practice, we actually do not want to solve for an eigendecomposition problem at all. Therefore, one could simply initialize the estimated eigenvector to identity ($\mathbf{Q}_0=\mathbf{I}$) and the eigenvalues to the sample variances of each input entry over N_0 samples ($\mathbf{\Lambda}_0=\operatorname{diag}\mathbf{R}_{N_0}$). We then start the iterations over the samples $k=1,\ldots,N$ and set the memory depth parameter to $\lambda_k=1/(k-1+\gamma)$. Effectively this corresponds to the following biased (but asymptotically unbiased as $N\to\infty$) covariance estimate:

$$\mathbf{R}_{N,\,biased}=\frac{N}{N+\gamma}\mathbf{R}_N+\frac{\gamma}{N+\gamma}\mathbf{\Lambda}_0 \tag{12}$$

This latter initialization strategy is utilized in all the computer experiments that are presented in the following sections.[2]

In the case of a forgetting covariance estimator (i.e., $\lambda_k=\lambda$), the initialization bias is not a problem, since its effect will diminish in accordance with the forgetting time constant any way. Therefore, in the non-stationary case, once again, we suggest using the latter initialization strategy: $\mathbf{Q}_0=\mathbf{I}$ and $\mathbf{\Lambda}_0=\operatorname{diag}\mathbf{R}_{N_0}$. In this case, in order to guarantee the accuracy of the first order perturbation approximation, we need to choose the forgetting factor λ such that the ratio $(1-\lambda)/\lambda$ is large. Typically, a forgetting factor $\lambda<10^{-2}$ will yield accurate results, although if necessary values up to $\lambda=10^{-1}$ could be utilized.

3.3. Extension to Complex-Valued PCA

The extension of RPCA to complex-valued signals is trivial. Basically, all matrix transpose operations need to be replaced by Hermitian (conjugate-transpose) operators. Below, we briefly discuss the derivation of the complex-valued RPCA algorithm following the steps of the real-valued version.

---

[2] A further modification that might be installed is to use a time-varying γ value. In the experiments, we used an exponentially decaying profile for γ, $\gamma=\gamma_0\exp(-k/\tau)$. This forces the covariance estimation bias to diminish even faster.

Table 1. The Recursive PCA Algorithm Outline

1. Initialize $\mathbf{Q}_0$ and $\mathbf{\Lambda}_0$.
2. At each time instant k do the following:
   a. Get input sample $\mathbf{x}_k$.
   b. Set memory depth parameter λ_k.
   c. Calculate $\boldsymbol{\alpha}_k = \mathbf{Q}_{k-1}^T \mathbf{x}_k$.
   d. Find perturbations $\mathbf{P_V}$ and $\mathbf{P_\Lambda}$ corresponding to $(1-\lambda_k)\mathbf{\Lambda}_{k-1} + \lambda_k \boldsymbol{\alpha}_k \boldsymbol{\alpha}_k^T$
   e. Update eigenvector and eigenvalue matrices:
      $\widetilde{\mathbf{Q}}_k = \mathbf{Q}_{k-1}(\mathbf{I} + \mathbf{P_V})$
      $\widetilde{\mathbf{\Lambda}}_k = (1-\lambda_k)\mathbf{\Lambda}_{k-1} + \mathbf{P_\Lambda}$
   f. Normalize the norms of eigenvector estimates by $\mathbf{Q}_k = \widetilde{\mathbf{Q}}_k \mathbf{T}_k$, where $\mathbf{T}_k$ is a diagonal matrix containing the inverses of the norms of each column of $\widetilde{\mathbf{Q}}_k$.
   g. Correct eigenvalue estimates by $\mathbf{\Lambda}_k = \widetilde{\mathbf{\Lambda}}_k \mathbf{T}_k^{-2}$, where $\mathbf{T}_k^{-2}$ is a diagonal matrix containing the squared norms of the columns of $\widetilde{\mathbf{Q}}_k$.

The sample covariance estimate for zero-mean complex data is given by

$$\mathbf{R}_k = \frac{1}{k}\sum_{i=1}^{k} \mathbf{x}_i \mathbf{x}_i^H = \frac{(k-1)}{k}\mathbf{R}_{k-1} + \frac{1}{k}\mathbf{x}_k \mathbf{x}_k^H \qquad (13)$$

where the eigendecomposition is $\mathbf{R}_k = \mathbf{Q}_k \mathbf{\Lambda}_k \mathbf{Q}_k^H$. Note that the eigenvalues are still real-valued in this case, but the eigenvectors are complex vectors. Defining $\boldsymbol{\alpha}_k = \mathbf{Q}_{k-1}^H \mathbf{x}_k$ and following the same steps as in (4) to (8), we determine that $\mathbf{P_V} = -\mathbf{P_V}^H$. Therefore, as opposed to the expressions derived in section 3.1, here the complex conjugation * and magnitude |.| operations are utilized. The i^{th} diagonal entry of $\mathbf{P_\Lambda}$ is found to be $|\alpha_i|^2$ and the $(i,j)^{\text{th}}$ entry of $\mathbf{P_V}$ is $\alpha_i \alpha_j^* / (\lambda_j + |\alpha_j|^2 - \lambda_i - |\alpha_i|^2)$ if $j \neq i$, and 0 if $j = i$. The algorithm in Table 1 is utilized as it is except for the modifications mentioned in this section.

4. NUMERICAL EXPERIMENTS

The PCA problem is extensively studied in the literature and there exist an excessive variety of algorithms to solve this problem. Therefore, an exhaustive comparison of the proposed method with existing algorithms is not practical. Instead, a comparison with a structurally similar algorithm (which is also based on first order matrix perturbations) will be presented [15]. We will also comment on the performances of traditional benchmark algorithms like Sanger's rule and APEX in similar setups, although no explicit detailed numerical results will be provided.

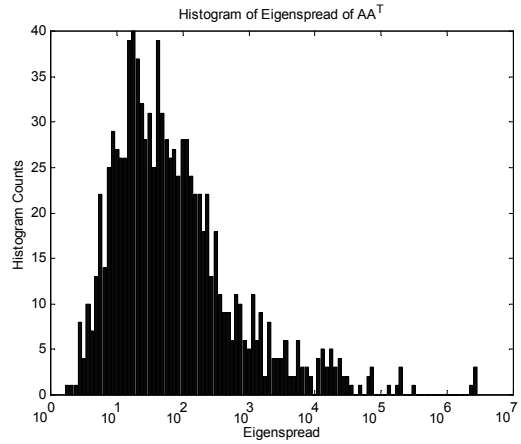

Figure 1. Distribution of eigenspread values for $\mathbf{AA}^T$, where $\mathbf{A}_{3x3}$ is generated to have Gaussian distributed random entries.

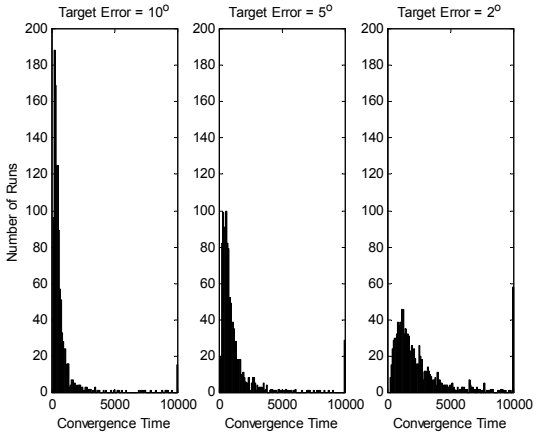

Figure 2. The convergence time histograms for RPCA in the 3-dimensional case for three different target accuracy levels.

4.1. Convergence Speed Analysis

In the first experimental setup, the goal is to investigate the convergence speed and accuracy of the RPCA algorithm. For this, *n*-dimensional random vectors are drawn from a normal distribution with an arbitrary covariance matrix. In particular, the theoretical covariance matrix of the data is given by $\mathbf{AA}^T$, where $\mathbf{A}$ is an *nxn* real-valued matrix whose entries are drawn from a zero-mean unit-variance Gaussian distribution. This process results in a wide range of eigenspreads (as shown in Fig. 1), therefore the convergence results shown here encompass such effects.

Specifically, the results of the 3-dimensional case study are presented here, where the data is generated by 3-dimensional normal distributions with randomly selected covariance matrices. A total of 1000 simulations (Monte Carlo runs) are carried out for each of the three target eigenvector estimation accuracies

(measured in terms of degrees between the estimated and actual eigenvectors): 10^o, 5^o, and 2^o. The convergence time is measured in terms of number of iterations it takes the algorithm to converge to the target eigenvector accuracy in all eigenvectors (not just the principal component). The histograms of convergence times (up to 10000 samples) for these three target accuracies are shown in Fig. 2, where everything above 10000 is also lumped into the last bin. In these Monte Carlo runs, the initial eigenvector estimates were set to the identity matrix and the randomly selected data covariance matrices were forced to have eigenvectors such that all the initial eigenvector estimation errors were at least 25^o. The initial γ value was set to 400 and the decay time constant was selected to be 50 samples. Values in this range were found to work best in terms of final accuracy and convergence speed in extensive Monte Carlo runs.

It is expected that there are some cases, especially those with high eigenspreads, which require a very large number of samples to achieve very accurate eigenvector estimations, especially for the minor components. The number of iterations required for convergence to a certain accuracy level is also expected to increase with the dimensionality of the problem. For example, in the 3 dimensional case, about 2% of the simulations failed to converge within 10^o in 10000 on-line iterations, whereas this ratio is about 17% for 5 dimensions. The failure to converge within the given number of iterations is observed for eigenspreads over $5x10^4$.

In a similar setup, Sanger's rule achieves a mean convergence speed of 8400 iterations with a standard deviation of 2600 iterations. This results in an average eigenvector direction error of about 9^o with a standard deviation of 8^o. APEX on the other hand converges rarely to within 10^o. Its average eigenvector direction error is about 30^o with a standard deviation of 15^o.

4.2. Comparison with First Order Perturbation PCA

The first order perturbation PCA algorithm [15] is structurally similar to the RPCA algorithm presented here. The main difference is the nature of the perturbed matrix: the former works on a perturbation approximation for the complete covariance matrix, whereas the latter considers the perturbation of a diagonal matrix. We expect this structural restriction to improve performance in terms of overall algorithm performance. To test this hypothesis, an experimental setup similar to the one in section 4.1 is utilized. This time, however, the data is generated by a colored time-series using a time-delay line (making the procedure a temporal PCA case study). Gaussian white noise is colored using a two-pole filter whose poles are selected from a random uniform distribution on the interval (0,1). A set of 15 Monte Carlo simulations was run on 3 dimensional data generated according to this procedure.

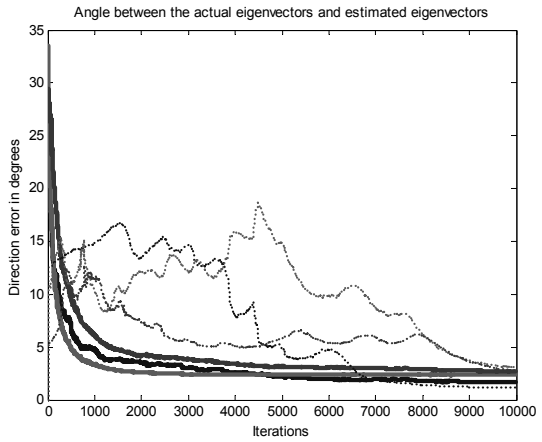

Figure 3. The average eigenvector direction estimation errors versus iterations are shown for the first order perturbation method (thin dotted lines) and for RPCA (thick solid lines).

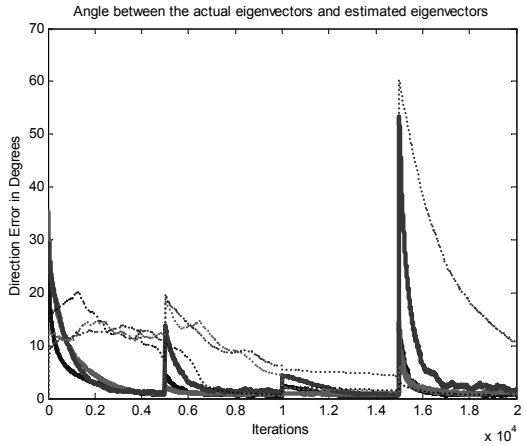

Figure 4. The average eigenvector direction estimation errors versus iterations for the first order perturbation method (thin dotted lines) and for RPCA (thick solid lines) in a piecewise stationary situation are shown. The eigenstructure of the input abruptly changes every 5000 samples.

The two parameters of the first order perturbation method were set to $\varepsilon=10^{-3}/6.5$ and $\delta=10^{-2}$. The parameters of RPCA were set to $\gamma_0=300$ and $\tau=100$. The average eigenvector direction estimation convergence curves are shown in Fig. 3.

Often, signal subspace tracking is necessary in signal processing applications dealing with nonstationary signals. To illustrate the performance of RPCA for such cases, a piecewise stationary colored noise sequence is generated by filtering white Gaussian noise with single-pole filters with the following poles: 0.5, 0.7, 0.3, 0.9 (in order of appearance). The forgetting factor is set to a constant $\lambda=10^{-3}$. The two parameters of the first order perturbation method were again set to $\varepsilon=10^{-3}/6.5$ and $\delta=10^{-2}$. The results of 30 Monte Carlo runs were averaged to obtain Fig. 4.

4.3. Direction of Arrival Estimation

The use of subspace methods for direction of arrival estimation in sensor arrays has been extensively studied (see [14] and the references therein). In Fig. 5, a sample run from a computer simulation of DOA according to the experimental setup described in [14] is presented to illustrate the performance of the complex-valued RPCA algorithm. To provide a benchmark (and an upper limit in convergence speed, we also performed this simulation using Matlab's *eig* function several times on the sample covariance estimate. The latter typically converged to the final accuracy demonstrated here within 10-20 samples. The RPCA estimates on the other hand take a few hundred samples due to the transient in the γ value. The main difference in the application of RPCA is that typical DOA algorithm will convert the complex PCA problem into a structured PCA problem with double the number of dimensions, whereas the RPCA algorithm works directly with the complex-valued input vectors to solve the original complex PCA problem.

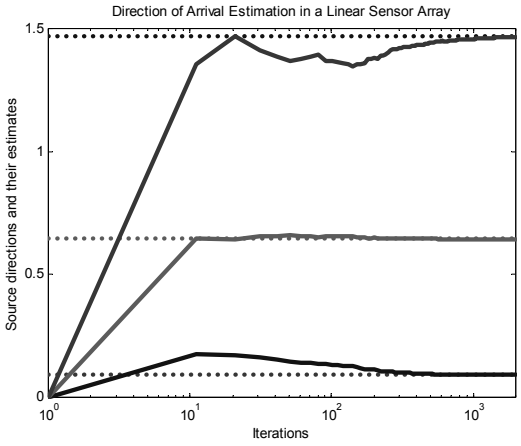

Figure 5. Direction of arrival estimation using complex-valued RPCA in a 3-source 6-sensor case.

4.4. An Example with 20 Dimensions

The numerical examples considered in the previous examples were 3-dimensional and 12-dimensional (6 dimensions in complex variables). The latter did not require all the eigenvectors to converge since only the 6-dimensional signal subspace was necessary to estimate the source directions; hence the problem was actually easier than 12 dimensions. To demonstrate the applicability to higher dimensional situations, an example with 20 dimensions is presented here. The PCA algorithms generally cannot cope well with higher dimensional problems because the interplay between two competing structural properties of the eigenspace becomes increasingly difficult to make a compromise from one or the other. Specifically, these two characteristics are the eigenspread ($\max\lambda_i/\min\lambda_i$) and the distribution of ratios of consecutive eigenvalues ($\lambda_n/\lambda_{n-1},\ldots,\lambda_2/\lambda_1$) when they are ordered from largest to smallest (where $\lambda_n>\ldots>\lambda_1$ are the ordered eigenvalues). Large eigenspreads lead to slow convergence due to the scarcity of samples representing the minor components. In small dimensional problems, this is typically the dominant issue that controls the convergence speeds of PCA algorithms. On the other hand, as the dimensionality increases, while very large eigenspreads are still undesirable due to the same reason, smaller and previously acceptable eigenspread values too become undesirable because consecutive eigenvalues approach each other. This causes the discriminability of the eigenvectors corresponding to these eigenvalues diminish as their ratio approaches unity. Therefore, the trade-off between small and large eigenspreads becomes significantly difficult. Ideally, the ratios between consecutive eigenvalues must be identical for equal discriminability of all subspace components. Variations from this uniformity will result in faster convergence in some eigenvectors, while others will suffer from almost spherical subspaces indiscriminability.

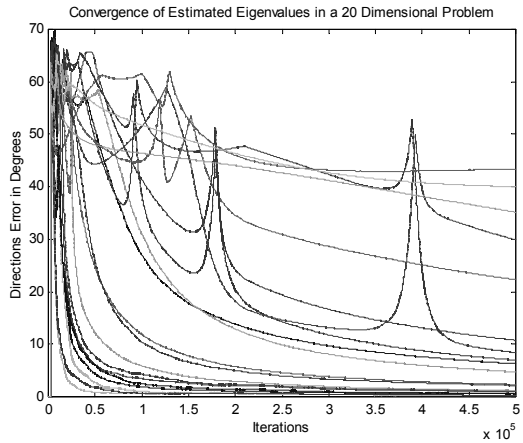

Figure 6. The angle error between the estimated eigenvectors (using RPCA) and their corresponding true eigenvectors in a 20 dimensional PCA problem is shown versus on-line iterations.

In Fig. 6, the convergence of the 20 estimated eigenvectors to their corresponding true values is illustrated in terms of the angle between them (in degrees) versus the number of on-line iterations. The data is generated by a 20-dimensional jointly Gaussian distribution with zero-mean, and a covariance matrix with eigenvalues equal to the powers (from 0 to 19) of 1.5 and eigenvectors selected randomly.[3] This result is typical of higher dimensional cases where major components converge relatively fast and minor components take much longer (in terms of samples and iterations) to reach the same level of accuracy.

[3] This corresponds to an eigenspread of $1.5^{19} \approx 2217$.

5. CONCLUSIONS

In this paper, a novel approximate fixed-point algorithm for subspace tracking is presented. The fast tracking capability is enabled by the recursive nature of the complete eigenvector matrix updates. The proposed algorithm is feasible for real-time implementation since the recursions are based on well-structured matrix multiplications that are the consequences of the rank-one perturbation updates exploited in the derivation of the algorithm. Performance comparisons with traditional algorithms, as well as a structurally similar perturbation-based approach demonstrated the advantages of the recursive PCA algorithm in terms of convergence speed and accuracy.

Acknowledgment. This work is supported by NSF grant ECS-0300340.

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 Transactions on Signal Processing, vol. 42, no. 5, pp. 1202-1217, 1994.

[3] J. Mao, A.K. Jain, "Artificial Neural Networks for Feature Extraction and Multivariate Data Projection," IEEE Transactions on Neural Networks, vol. 6., no. 2, pp. 296-317, 1995.

[4] Y. Cao, S. Sridharan, M. Moody, "Multichannel Speech Separation by Eigendecomposition and its Application to Co-Talker Interference Removal," IEEE Transactions on Speech and Audio Processing, vol. 5, no. 3, pp. 209-219, 1997.

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

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

[7] T.D. Sanger, "Optimal Unsupervised 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," Biological Cybernetics, 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] L. Xu, "Least Mean Square Error Reconstruction Principle for Self-Organizing Neural-Nets", Neural Networks, vol. 6, pp. 627-648, 1993.

[11] B. Yang, "Projection Approximation Subspace Tracking", IEEE Transactions on Signal Processing, vol. 43, no. 1, pp. 95-107, 1995.

[12] Y. Hua, Y. Xiang, T. Chen, K. Abed-Meriam, Y. Miao, "Natural Power Method for Fast Subspace Tracking", Proceedings of NNSP'99, pp. 176-185, 1999.

[13] Y.N. Rao, J.C. Principe, "Robust On-line Principal Component Analysis Based on a Fixed-Point Approach". Proceedings of ICASSP'02, vol. 1, pp. 981-984, 2002.

[14] D. Erdogmus, Y.N. Rao, K.E. Hild II, J.C. Principe, "Simultaneous Principal Component Extraction with Application to Adaptive Blind Multiuser Detection," EURASIP Journal on Applied Signal Processing, vol. 2002, no. 12, pp. 1473-1484, 2002.

[15] B. Champagne, "Adaptive Eigendecomposition of Data Covariance Matrices Based on First-Order Perturbations," IEEE Transactions on Signal Processing, vol. 42, no. 10, 1994.