FAST ADAPTIVE EIGENVALUE DECOMPOSITION : A MAXIMUM LIKELIHOOD APPROACH

Christian Riou *Thierry Chonavel*

ENSTB, S.C. Dpt., B.P. 832, 29285, Brest cedex, FRANCE
firstname.surname@enst-bretagne.fr

ABSTRACT

A new adaptive subspace estimation algorithm is presented, based on the maximisation of the likelihood functional. It requires little computational cost and the particular structure of the algorithm ensures the orthonormality of the estimated basis of eigenvectors. Application to moving sources localization shows the very good behavior of the algorithm when applied to problems of practical interest.

1. INTRODUCTION

Let $X = (X_n)_{n\in\mathbb{Z}}$ be a vector, complex valued, stochastic process ($X_n \in \mathbb{C}^N$). In signal processing, the computation (or estimation) of the EigenValue Decomposition (EVD) of the hermitian covariance matrix $R_{X,n} = E\left\{X_n^H X_n\right\}$ has been intensively used in different applications. In particular, Direction Of Arrival (DOA) estimation (MUSIC algorithm, e.g. [11]), frequencies estimation, blind channel impulse response estimation [8] or vector quantization [13], are classical applications in which this problem arises.

If the process X is stationary, $R_{X,n} = R_X$ can be estimated easily, and there exist in the literature several iterative methods to perform its EVD. Among these methods the orthogonal iteration, that requires a QR algorithm at each iteration is currently used [7]. Unfortunately, in many applications (moving sources, time varying transmission channels,...), the process X under study is not stationary, and the above approach cannot be applied directly.

To overcome this problem, several studies have been carried out during the last decade. First, straightforward extensions of the stationary case to the non stationary case have been proposed (e.g. [5]). They rely on adaptive estimation of the instantaneous covariance matrix $R_{X,n}$ by means of a smoothing technique, yielding estimates such as

$$\widehat{R}_{X,n} = (1-\alpha)\widehat{R}_{X,n-1} + \alpha x_n x_n^H \qquad (1)$$

where α is a forgetting factor, $0 < \alpha < 1$, and x_n is the observed data vector at time n. Then, one iteration of the classical orthogonal iteration EVD method is performed, that makes use of the EVD performed at the previous step, and of the covariance matrix $\widehat{R}_{X,n}$. The computational cost of such an approach is in general $O(N^2K)$, where K is the number of estimated eigen-vectors.

To reach lower computational cost, several authors proposed to replace in (1) the estimated covariance matrix by its eigenvalue decomposition (e.g. [4, 3]). In ([4]), some simplifying assumptions (the space is splitted into only 2 eigensubspaces, in view to obtain basis of the so-called noise and signal subspaces) permit to get an algorithm with computational cost in $O(NK)$. This algorithm makes use of Householder reflections matrices. By means of a Givens rotations-based algorithm, and other simplifying hypothesis, [3] presents an $O(NK)$ algorithm that ensures complete eigenvalue decomposition, and perfect orthonormality of the estimated eigenvectors.

Alternatively, another approach consists in using some constrained or unconstrained criterion, the optimum of which corresponds to the EVD of $R_{X,n}$, and optimize it by means of some stochastic gradient-like algorithm [9, 13, 10]. These stochastic algorithms have various computational cost : $O(N^2K)$ for Regalia's Givens rotations-based algorithm [9], with exact orthonormality of the estimated EVD, $O(NK^2)$ in [10], where only approximate orthonormality is ensured, as well as for Yang's $O(NK)$ RLS algorithm [13]. But for this last algorithm, an additional orthonormalization procedure is often necessary to get satisfactory results in practice (at the expense of $O(NK^2)$ additional operations).

In this paper, a fast adaptive eigenvalue decomposition is presented. It relies on the optimization of a log-likelihood functional by means of a stochastic gradient algorithm, that incorporates the orthonormality constraint upon the eigenvectors. Its computational cost is in the order of $O(NK)$.

The paper is organized as follows. In section two, the maximum likelihood approach is presented, and in section three, the algorithm is derived and its convergence properties and asymptotic variance are studied. The last section is devoted to simulations and comparisons with other methods.

2. PROBLEM STATEMENT AND ADAPTIVE MAXIMUM LIKELIHOOD

In many problems, the observed data model at time n is in the form

$$X_n = H_n S_n + B_n \qquad (2)$$

where X_n is a random $N-$vector, and S_n a random $p-$vector ($p < N$), with full rank covariance matrix. B_n is an additive noise $N-$vector uncorrelated with S_n, and with covariance matrix is $\sigma_n^2 I_N$ (I_K is the $K \times K$ identity matrix). H_n is a full rank, time-varying $N \times p$ matrix.

The problem is to estimate H_n, or some of its parameters of interest, out of the knowledge of the space spanned by its columns. Clearly, getting this vector space amounts to estimate the space spanned by the eigenvectors associated to the largest eigenvalues of $R_{X,n} = E[X_n X_n^H]$. Also, the complete knowledge of the EVD of $R_{X,n}$ is often interesting, for instance to estimate the possibly unknown range of H_n (see e.g. [6, 12]).

Let $R_{X,n} = U_n \Lambda_n U_n^H$ denote the EVD of $R_{X,n}$. Assuming that X_n is a zero mean complex valued gaussian random vector, the probabilty density function (p.d.f.) of X_n is parametrized by U and Λ, that represent the matrix of eigenvecors and the diagonal matrix of the eigenvalues, denoted $(\lambda_k)_{k=1,N}$, respectively. This p.d.f. is given by

$$f\left(x_n;\Lambda,U\right) = \frac{1}{\sqrt{2\pi}\left|U\Lambda U^H\right|} \exp\left(-\frac{1}{2} x_n^H U \Lambda^{-1} U^H x_n\right) \tag{3}$$

Up to additive and multiplicative constants that have been dropped, the opposite of the corresponding log-likelihood is given by

$$\Phi\left(x_n;\Lambda,U\right) = \sum_{k=1}^{N} \log\left(\lambda_k\right) + x_n^H U \Lambda^{-1} U^H x_n, \tag{4}$$

The minimization of the "distance" criterion $K_n\left(\Lambda,U\right) = E[\Phi\left(x_n;\Lambda,U\right)]$ can be searched adaptively by means of the following stochastic gradient algorithm :

$$\begin{aligned} y_n &= U_{n-1}^H x_n && \text{(5a)} \\ \Lambda_n &= \Lambda_{n-1} - \mu\left(\Lambda_{n-1}^{-1} - \Lambda_{n-1}^{-2}\operatorname{diag}(y_n y_n^H)\right) && \text{(5b)} \\ U_n &= U_{n-1} - \mu \Lambda_{n-1}^{-1} x_n y_n^H . && \text{(5c)} \end{aligned}$$

where $\operatorname{diag}(y_n y_n^H)$ is a diagonal matrix with diagonal terms matching those of $y_n y_n^H$. Unfortunately the convergence properties of this algorithm are not satisfactory. Indeed, let us consider for instance the equation of actualization of U_n. A stationary point (Λ, U) should satisfy $E[\Lambda^{-1} X X^H U] = \Lambda^{-1} R_X U = 0$. Obviously, the EVD of R_X does not match this constraint.

3. CONSTRAINED ADAPTIVE MAXIMUM LIKELIHOOD

To overcome the above mentionned problem, we will modify the previous algorithm by accounting for the orthonormality constraint upon the eigenvectors, in the computation of the gradient of $\Phi\left(x_n;\Lambda,U\right)$. We will show the good properties of the corresponding algorithm. In particular, we will show that it produces exactly unitary estimates at each iteration. Moreover, we prove its convergence towards the EVD of $R_{X,n}$ and study the asymptotic variance of its estimate.

3.1. Derivation of the MALASE algorithm

Let us consider the following differential of Φ :

$$d_U\Phi\left(x_n;\Lambda,U\right) = \Phi\left(x_n;\Lambda,U+dU\right) - \Phi\left(x_n;\Lambda,U\right) \tag{6}$$

Under the unitary constraint $U^H U = I_N$, we have $dU^H = -U^H dU U^H$, and first order differential yields :

$$\begin{aligned} d_U\Phi &= x_n^H \left(dU \Lambda^{-1} U^H + U \Lambda^{-1} dU^H\right) x_n \\ &= -y_n^H \left(\Lambda^{-1} U^H dU - U^H \Lambda^{-1} dU^H\right) y_n \end{aligned} \tag{7}$$

From this expression, the matrix $\frac{\partial\Phi(x_n;\Lambda,U)}{\partial U}$, with general term $\left[\frac{\partial\Phi(x_n;\Lambda,U)}{\partial U}\right]_{i,j} = \frac{\partial\Phi(x_n;\Lambda,U)}{\partial U_{ij}}$ is given by

$$\frac{\partial\Phi\left(x_n;\Lambda,U\right)}{\partial U} = -U\Lambda^{-1} y_n y_n^H + U y_n y_n^H \Lambda^{-1} \tag{8}$$

This yields the following stochastic algorithm :

$$\begin{aligned} y_n &= U_{n-1}^H x_n && \text{(9a)} \\ \Lambda_n &= \Lambda_{n-1} - \mu\left(\Lambda_{n-1}^{-1} - \Lambda_{n-1}^{-2}\operatorname{diag}(y_n y_n^H)\right) && \text{(9b)} \\ U_n &= U_{n-1}\left(I + \mu\left(\Lambda_{n-1}^{-1} y_n y_n^H - y_n y_n^H \Lambda_{n-1}^{-1}\right)\right) && \text{(9c)} \end{aligned}$$

where $\operatorname{diag}(y_n y_n^H)$ is a diagonal matrix with diagonal terms matching those of $y_n y_n^H$. This algorithm has been baptized MALASE (MAximum Likelihood Adaptive Subspace Estimation). The computational cost of this algorithm is $O(NK)$.

3.2. Convergence properties

To study the stationary points (Λ, U) of the algorithm, let us recall that they satisfy the equations :

$$\begin{aligned} h_1(\Lambda) &= \Lambda^{-1} - \Lambda^{-2} E[\operatorname{diag}(Y Y^H)] && \text{(10)} \\ &= \Lambda^{-1} - \Lambda^{-2}\operatorname{diag}(U^H R_X U) \\ &= 0 \\ h_2(U) &= E\left\{U\left(\Lambda^{-1} Y Y^H - Y Y^H \Lambda^{-1}\right)\right\} && \text{(11)} \\ &= U\left(\Lambda^{-1} U^H R_X U - U^H R_X U \Lambda^{-1}\right) \\ &= 0 \end{aligned}$$

One can easily check that a necessary and sufficient condition for Λ to be a stationary point is that $[\Lambda]_{i,i} = U_i^H R_X U_i$, and that the matrix $U^H R_X U$ is a block diagonal matrix where any block has all equal diagonal terms.

To study the stability, we consider perturbations of the stationary point (Λ, U). More particularly, we consider multiplicative perturbations in the form of Givens rotations ($U + dU = UG$, where G is a Given's rotation), and this permits to show simply that a necessary condition for stability is that $U^H R_X U$ is a diagonal matrix. This shows that only eigenvalue decompositions may represent stable points.

Furthermore, we account for the important fact, proved in the following, that if U_n is unitary, the matrix U_{n+1} is also unitary (up to a second order term in μ). Thus, since the stability of the the eigenvalue decomposition is ensured for Given's rotations multiplicative perturbations, and since such matrices parametrize all matrix transformations on the sphere of unitary matrices, the stability of eigenvalue decompositions is guaranteed. Thus, we have :

Proposition 1 *the point (Λ, U) is a stable point of the algorithm (9) iff it represents the EVD : $R_X = U\Lambda U^H$.*

3.3. Orthonormality properties

To show the unitarity conservation property of MALASE algorithm, let us recall first a straightforward property of skew-symmetric matrices (i.e. matrices A such that $A^H = -A$) : for any such matrix A,

$$\exp(A) \times \exp(A^H) = I_N, \tag{12}$$

i.e. $\exp(A)$ is unitary. Now, let us remark that the actualisation of U in (9) can be rewritten as

$$U_n = U_{n-1} \times \exp(\mu[\Lambda_{n-1}^{-1} y_n y_n^H - y_n y_n^H \Lambda_{n-1}^{-1}]) + O(\mu^2) \tag{13}$$

and that the matrix

$$\Lambda_{n-1}^{-1} y_n y_n^H - y_n y_n^H \Lambda_{n-1}^{-1} \tag{14}$$

is skew-symetric. Thus, if U_0 is a unitary matrix, up to the second order in μ, the sequence of estimated matrices U_n produced by the algorithm $MALASE$ will be unitary.

Let us point out that the previous remark suggests a new algorithm for which exact ortonormality will occur at each step, provided U_0 is unitary. This algorithm is obtained by replacing equation (9c) in MALASE by :

$$U_n = U_{n-1} \exp\left(\mu[\Lambda_{n-1}^{-1} y_n y_n^H - y_n y_n^H \Lambda_{n-1}^{-1}]\right). \tag{15}$$

Although it might seem to be quite difficult to compute the exponential matrix that appears in this algorithm, the following straightforward result

$$e^{uv^H} = I + \frac{uv^H}{v^H u}\left(e^{(v^H u)} - 1\right) \quad \forall u, v \in \mathbb{C}^N, \tag{16}$$

yields in (15):

$$\begin{aligned} U_n \quad = \quad & U_{n-1} \times \qquad (17) \\ & \left[I + \left(\exp\left(\mu Y_n^H \Lambda_n^{-1} y_n\right) - 1\right) \frac{\Lambda_n^{-1} y_n y_n^H}{y_n^H \Lambda_n^{-1} y_n}\right] \\ & \times \left[I + \left(\exp\left(-\mu y_n^H \Lambda_n^{-1} y_n\right) - 1\right) \frac{y_n y_n^H \Lambda_n^{-1}}{y_n^H \Lambda_n^{-1} y_n}\right] \end{aligned}$$

where only a scalar exponential computation is involved. This shows that this algorithm has a computational cost of $O(NK)$. Since this algorithm only differs from MALASE by a factor $O(\mu^2)$, they both yield similar results.

3.4. Asymptotic performances

From ([2], p. 103), we know that the asymptotic covariance matrix of the vector parameters estimated by a stochastic gradient algorithm can be estimated by solving a Lyapounov equation. Let us consider the $(N^2 + N)$-vector $H = [U_1^T \dots U_N^T \, \lambda_1 \dots \lambda_N]^T$ where U_k denotes the kth column of U. Then, for MALASE algorithm, the asymptotic covariance matrix of the error of estimation of H is the $(N^2 + N) \times (N^2 + N)$ matrix given by

$$P = \mu \frac{1}{2} I_{N^2+N} \tag{18}$$

This is a very interesting result since the asymptotic variance depends only on μ, and not on the eigenvalues of R_X, as in other algorithms (e.g. [10]). Furthermore, let us remark that any of the N^2+N partameters are asymptotically uncorrelated, and have the same variance$\mu \frac{1}{2}$.

4. APPLICATIONS AND COMPARISONS

Let us remark that the algorithm MALASE has a structure close to that of the algorithm PROTEUS-1 [3]. The latter differs from MALASE by the fact that Λ_n is obtained from a minimum mean square error criterion ($\| \Lambda - U^H R_{X,n} U \|$), and that $U_n = U_{n-1} \times \exp(\Theta_n)$, where the skew matrix Θ_n satisfies $[\Theta_n]_{i,j} = \mu y_{n,i} y_{n,j}^* / (\lambda_{n-1,i} - \lambda_{n-1,j})$, instead of $[\Theta_n]_{i,j} = \mu y_{n,i} y_{n,j}^* (1/\lambda_{n-1,i} - 1/\lambda_{n-1,j})$ for MALASE.

Furthermore, the PROTEUS-1 algorithm requires $O(N^2 K)$ multiplications. It can be approximated to get the $O(NK)$ PROTEUS-2 algorithm. Together with MALASE these are the only algorithms that ensure $O(NK)$ computational cost and perfect orthonormality of the eigenvectors. However, since PROTEUS-2 is obtained thanks to simplifying asumptions, we checked that this results in worse results than for MALASE in practical non stationary situations, as shown figure 1.

We checked the efficiency of the proposed algorithm to locate and track moving sources, and compared it to Proteus-1 and S. Affes' method [1]. At each instant n, we update the sources positions $\varphi_{i,n}$ as in [10] and deal with crossing sources by means of a kinematic model (see [1]). The step is $\mu_1 = 0.01$ for our algorithm and Proteus-1, and $\mu_2 = 0.005$ for S. Affes method. These values have been chosen in order to guarantee optimum tracking capability in the context of our simulations. Two moving crossing sources $s1$ and $s2$ are tracked over 10,000 iterations using an array of $N = 16$ sensors. We assume the presence of a white noise with $SNR = 5dB$. The time averaged bias $b = E\{\varepsilon\}$ and standard deviation $\sigma = \left[E\left\{|\varepsilon - b|^2\right\}\right]^{1/2}$ of the estimated location $\varepsilon = \varphi - \widehat{\varphi}$ are computed. We check in table 1 that our algorithm stands the comparison to PROTEUS-1. Affes' method appears to show better results. But it requires the use of a kinematic model, the adjustment of which is not straightforward. Furthermore, for problems where non Toeplitz hermitian matrices are considered (e.g. blind deconvolution of mutli-channel digital communications signals), it cannot be used any longer.

Finally, a zoom over 1000 iterations is displayed on figures 2 and 3, showing the similarity of performances between MALASE ($O(NK)$) and PROTEUS-1 ($O(NK^2)$) methods.

	b		σ	
	$s1$	$s2$	$s1$	$s2$
Proteus-1	0.24	0.30	0.36	0.41
MALASE	0.22	0.27	0.33	0.38
S. Affes	0.08	0.10	0.18	0.19

Table 1: Bias and Standard Deviation of the error of estimation, in degree, over 10000 iterations.

5. REFERENCES

[1] S. Affes, S. Gazor, and Y. Grenier. An algorithm for multi-source beamforming and multi-target tracking.

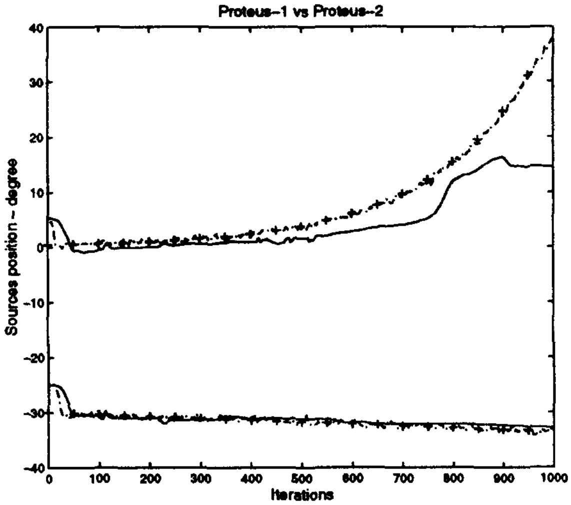

Figure 1: Proteus-1 versus Proteus-2 : tracking capability; + : true position φ - Dashed line : estimated position $\hat{\varphi}$, Proteus-1 - Solid line : estimated position $\hat{\varphi}$, Proteus-2 - $SNR = 10dB$.

IEEE Trans. on Signal Processing, 44(6):1512–1522, June 1996.

[2] A. Benveniste, M. Metivier, and P. Priouret. *Algorithmes Adaptatifs et Approximations Stochastiques, Techniques stochastiques.* Masson, 1987.

[3] B. Champagne and Q.G. Liu. A new family of EVD tracking algorithms using Givens rotations. In *ICASSP'96*, pages 1–4, 1996.

[4] R.D. DeGroat. Noniterative subspace tracking. *IEEE Trans. on Signal Processing*, 40(3):571–577, March 1992.

[5] S. Erlich and R. Yao. Convergences of adaptative block simultaneous iteration method for eigenstructure decomposition. *Signal Processing*, 37:1–13, January 1994.

[6] J.J. Fuchs. Estimation of the number of signals in the presence of unknow correlated sensor noise. *IEEE Trans. on Signal Processing*, 40(5):1053–1061, May 1992.

[7] G. H. Golub and C. F. van Loan. *Matrix Computation - Second Edition.* John Hopkins Series in Mathematical Sciences. John Hopkins University Press, 1990.

[8] E. Moulines, P. Duhamel, J.F. Cardoso, and S. Mayrargue. Subspace methods for the blind identification of multichannel FIR filters. *IEEE Trans. on Signal Processing.*, 43(2):516–525, February 1995.

[9] P.A. Regalia. An adaptative unit norm filter with application to signal analysis and Karhunen–Loeve transformations. *IEEE Trans. on CAS*, 37(5):646–649, May 1990.

[10] C. Riou, T. Chonavel, and P.Y. Cochet. Adaptive subspace estimation : Aplication to moving sources localization and blind channel identification. In *ICASSP'96*, pages 1649–1652, 1996.

[11] R.O. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Trans. on Antenna Propagation*, AP-34:276–280, March 1986.

[12] Q. Wu and D.R. Fuhrmann. A parametic method for determining the number of signals in narrow-band direction finding. *IEEE Trans. on Signal Processing*, 39(8):1848–1857, August 1991.

[13] B. Yang. Projection approximation subspace tracking. *IEEE Trans. on Signal Processing*, 43(1):95–107, January 1995.

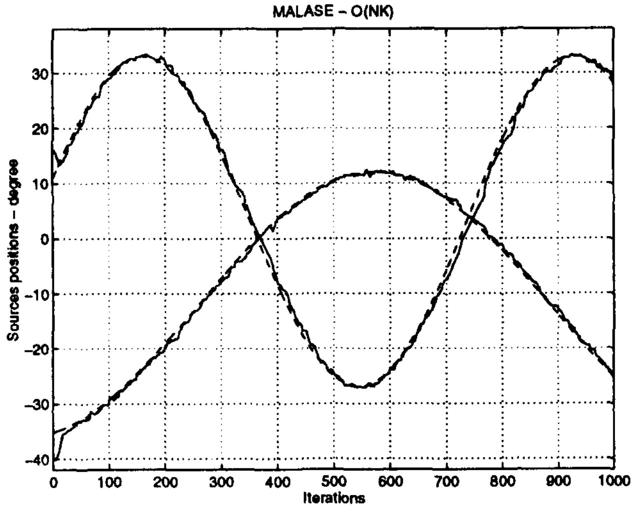

Figure 2: Tracking of two moving sources - Dashed line : true position φ - Solid line : estimated position $\hat{\varphi}$ - $SNR = 5dB$.

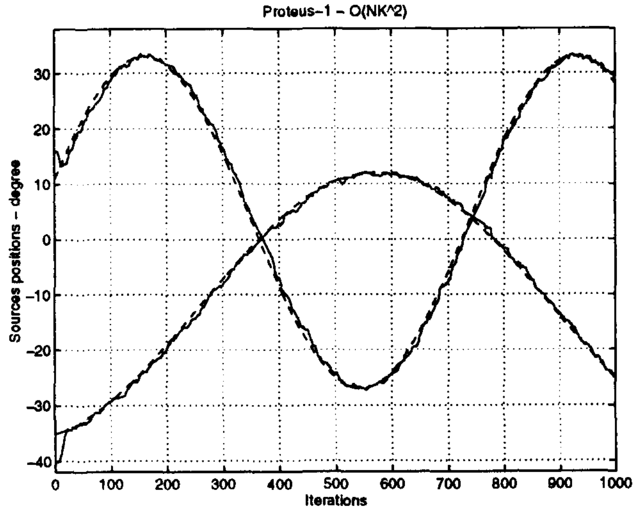

Figure 3: Tracking of two moving sources - Dashed line : true position φ - Solid line : estimated position $\hat{\varphi}$ - $SNR = 5dB$.