YAST ALGORITHM FOR MINOR SUBSPACE TRACKING

Roland BADEAU, Bertrand DAVID, Gaël RICHARD

Télécom Paris - Département TSI
46 rue Barrault - 75634 PARIS cedex 13

ABSTRACT

This paper introduces a new algorithm for tracking the minor subspace of the correlation matrix associated with time series. This algorithm is shown to have a better convergence rate than existing methods. Moreover, it guarantees the orthonormality of the subspace weighting matrix at each iteration, and reaches a linear complexity.

1. INTRODUCTION

Fast estimation and tracking of the principal or minor subspace of a sequence of random vectors is a major problem in many applications. We can cite, for example, code division multiple access (CDMA) communications, where many multiuser detection algorithms are actually subspace-based [1]. Recently, we presented in [2] a new principal subspace tracker dedicated to time series analysis, which is derived from the SP algorithm by C.E. Davila [3]. This new algorithm, referred to as YAST, reaches the lowest complexity found in the literature, and outperforms classical methods in terms of subspace estimation. Moreover, it guarantees the orthonormality of the subspace weighting matrix at each time step. In this paper, we focus on minor subspace analysis (MSA). In the literature, it is commonly admitted that MSA is a more difficult problem than principal subspace analysis (PCA). In particular, the classical Oja algorithm [4] is known to diverge. Some more robust MSA algorithms have been presented in [5–9]. However the convergence rate of these algorithms is much lower than that of the classical PCA techniques. Here we propose a version of the YAST algorithm dedicated to MSA, which is shown to have better convergence properties.

The paper is organized as follows. In section 2, the basic principle of the YAST algorithm is presented. Then a fast implementation of YAST is proposed in section 3. The performance of this minor subspace tracker is illustrated in section 4. Finally, the main conclusions of this paper are summarized in section 5.

2. PRINCIPLE

Let $\{\boldsymbol{x}(t)\}_{t\in\mathbb{Z}}$ be a sequence of n-dimensional data vectors. We are interested in tracking the minor subspace spanned by its correlation matrix $\boldsymbol{C}_{xx}(t)$. This matrix can be recursively updated according to

$$\boldsymbol{C}_{xx}(t) = \beta\,\boldsymbol{C}_{xx}(t-1) + \boldsymbol{x}(t)\,\boldsymbol{x}(t)^H \tag{1}$$

where $0 < \beta < 1$ is the forgetting factor.

The YAST algorithm for tracking the minor subspace relies on the following principle: an $n \times r$ orthonormal matrix $\boldsymbol{W}(t)$ spans the r-dimensional minor subspace of $\boldsymbol{C}_{xx}(t)$ if and only if it minimizes the criterion

$$\mathcal{J}\left(\boldsymbol{W}(t)\right) = \mathrm{trace}\left(\boldsymbol{W}(t)^H \boldsymbol{C}_{xx}(t)\boldsymbol{W}(t)\right).$$

In particular, the minimum of this criterion is equal to the sum of the r lowest eigenvalues of $\boldsymbol{C}_{xx}(t)$. However, implementing this minimization over all orthonormal matrices is computationally demanding (the complexity is $O(n^2 r)$), and does not lead to a simple recursion between $\boldsymbol{W}(t)$ and $\boldsymbol{W}(t-1)$).

In order to reduce the computational cost, this search is limited to the range space of $\boldsymbol{W}(t-1)$ plus one or two additional search directions. In other words, the r-dimensional range space of $\boldsymbol{W}(t)$ is to be found as a subspace of the $(r+p)$-dimensional space spanned by the $n \times (r+p)$ matrix

$$\underline{\boldsymbol{V}}(t) = [\boldsymbol{W}(t-1),\, \underline{\boldsymbol{x}}(t)] \tag{2}$$

where $\underline{\boldsymbol{x}}(t)$ contains $p = 1$ or 2 columns. In practice, it is proposed in [3] to choose $\underline{\boldsymbol{x}}(t) = \boldsymbol{x}(t)$ or $\underline{\boldsymbol{x}}(t) = [\boldsymbol{x}(t),\, \boldsymbol{C}_{xx}(t-1)\,\boldsymbol{x}(t)]$.

Let $\underline{\boldsymbol{W}}(t)$ be an $n \times (r+p)$ orthonormal matrix spanning the range space of $\underline{\boldsymbol{V}}(t)$. Then $\boldsymbol{W}(t)$ will be written in the form

$$\boldsymbol{W}(t) = \underline{\boldsymbol{W}}(t)\,\boldsymbol{U}(t). \tag{3}$$

where $\boldsymbol{U}(t)$ is an $(r+p) \times r$ orthonormal matrix. In this case

$$\mathcal{J}\left(\boldsymbol{W}(t)\right) = \mathrm{trace}\left(\boldsymbol{U}(t)^H \underline{\boldsymbol{C}}_{yy}(t)\boldsymbol{U}(t)\right) \tag{4}$$

where $\underline{\boldsymbol{C}}_{yy}(t)$ is the $(r+p) \times (r+p)$ matrix

$$\underline{\boldsymbol{C}}_{yy}(t) = \underline{\boldsymbol{W}}(t)^H\,\boldsymbol{C}_{xx}(t)\,\underline{\boldsymbol{W}}(t). \tag{5}$$

The exhaustive search among all $n \times r$ orthonormal matrices $\boldsymbol{W}(t)$ is replaced by the minimization of (4) over all $(r+p) \times r$ orthonormal matrices $\boldsymbol{U}(t)$. The result of this minimization is well-known: $\boldsymbol{U}(t)$ must span the r-dimensional minor subspace of $\underline{\boldsymbol{C}}_{yy}(t)$. Thus the subspace weighting matrix $\boldsymbol{W}(t)$ can be tracked by computing

- an orthonormal basis $\underline{\boldsymbol{W}}(t)$ of the range space of $\underline{\boldsymbol{V}}(t)$,
- the matrix $\underline{\boldsymbol{C}}_{yy}(t) = \underline{\boldsymbol{W}}(t)^H\,\boldsymbol{C}_{xx}(t)\,\underline{\boldsymbol{W}}(t)$,
- an $(r+p) \times r$ orthonormal matrix $\boldsymbol{U}(t)$ spanning the r-dimensional minor subspace of $\underline{\boldsymbol{C}}_{yy}(t)$,
- the matrix $\boldsymbol{W}(t) = \underline{\boldsymbol{W}}(t)\,\boldsymbol{U}(t)$.

In a particular implementation, $\boldsymbol{U}(t)$ can be obtained via the eigenvalue decomposition of $\underline{\boldsymbol{C}}_{yy}(t)$. As a consequence, the columns of the resulting matrix $\boldsymbol{W}(t)$ defined in equation (3) correspond to the r minor eigenvectors of $\boldsymbol{C}_{xx}(t)$. However, this calculation leads to an overall complexity of $O(nr^2)$.

In order to reduce the global complexity to $O(nr)$, we choose a different strategy which avoids the eigenvalue decomposition. As mentioned above, $\boldsymbol{U}(t)$ must be an orthonormal matrix spanning the r-dimensional minor subspace of $\underline{\boldsymbol{C}}_{yy}(t)$. Therefore $\boldsymbol{U}(t)$ can be

The authors would like to acknowledge the support of the IST-Kspace network of excellence and the ACI *Masse de données* Music Discover.

obtained as an orthogonal complement of the p-dimensional dominant subspace of $\underline{\boldsymbol{C}}_{yy}(t)$. Thus the YAST algorithm consists in first computing $\underline{\boldsymbol{C}}_{yy}(t)$ and its p-dimensional dominant subspace, then $\boldsymbol{U}(t)$ as an orthogonal complement of this subspace. As shown in section 3, this algorithm can be efficiently implemented by updating the $r \times r$ compressed correlation matrix $\boldsymbol{C}_{yy}(t)$, defined as

$$\boldsymbol{C}_{yy}(t) = \boldsymbol{W}(t)^H \boldsymbol{C}_{xx}(t) \boldsymbol{W}(t). \tag{6}$$

3. FAST IMPLEMENTATION OF YAST

Below, a fast implementation of the YAST algorithm is proposed, whose global cost is only $(5p+1)nr$ flops[1]. It can be decomposed into four steps: computation of $\underline{\boldsymbol{W}}(t)$ (section 3.1), computation of $\underline{\boldsymbol{C}}_{yy}(t)$ (section 3.2), update of $\boldsymbol{W}(t)$ (section 3.3), update of $\boldsymbol{C}_{yy}(t)$ (section 3.4). This implementation is summarized in section 3.5.

3.1. Computation of $\underline{\boldsymbol{W}}(t)$

Define the $r \times p$ matrix $\underline{\boldsymbol{y}}(t) = \boldsymbol{W}(t-1)^H \underline{\boldsymbol{x}}(t)$ and let

$$\underline{\boldsymbol{e}}(t) = \underline{\boldsymbol{x}}(t) - \boldsymbol{W}(t-1)\, \underline{\boldsymbol{y}}(t). \tag{7}$$

The $n \times p$ matrix $\underline{\boldsymbol{e}}(t)$ is orthogonal to $\boldsymbol{W}(t-1)$. Let

$$\underline{\boldsymbol{u}}(t)\, \underline{\sigma}(t) = \underline{\boldsymbol{e}}(t) \tag{8}$$

be a factorization of the matrix $\underline{\boldsymbol{e}}(t)$, where $\underline{\boldsymbol{u}}(t)$ is a $n \times p$ orthonormal matrix, and $\underline{\sigma}(t)$ is a $p \times p$ matrix. Below, we suppose that $\underline{\sigma}(t)$ is non-singular. Note that the matrix

$$\underline{\boldsymbol{W}}(t) = [\boldsymbol{W}(t-1),\, \underline{\boldsymbol{u}}(t)] \tag{9}$$

is orthonormal. In particular, $\underline{\boldsymbol{W}}(t)$ can be written in the form

$$\underline{\boldsymbol{W}}(t) = \underline{\boldsymbol{V}}(t)\, \underline{\boldsymbol{T}}(t) \tag{10}$$

where $\underline{\boldsymbol{T}}(t)$ is the $(r+p) \times (r+p)$ non-singular matrix

$$\underline{\boldsymbol{T}}(t) = \left[\begin{array}{c|c} \boldsymbol{I}_r & -\underline{\boldsymbol{y}}(t)\, \underline{\sigma}(t)^{-1} \\ \hline \boldsymbol{0}_{p \times r} & \underline{\sigma}(t)^{-1} \end{array}\right]. \tag{11}$$

3.2. Computation of $\underline{\boldsymbol{C}}_{yy}(t)$

Substituting equation (10) into equation (5) yields

$$\underline{\boldsymbol{C}}_{yy}(t) = \underline{\boldsymbol{T}}(t)^H \underline{\boldsymbol{C}}'_{yy}(t)\, \underline{\boldsymbol{T}}(t) \tag{12}$$

where $\underline{\boldsymbol{C}}'_{yy}(t)$ is the $(r+p) \times (r+p)$ matrix

$$\underline{\boldsymbol{C}}'_{yy}(t) = \underline{\boldsymbol{V}}(t)^H \boldsymbol{C}_{xx}(t)\, \underline{\boldsymbol{V}}(t). \tag{13}$$

Then let

$$\begin{aligned} \underline{\boldsymbol{\alpha}}(t) &= \boldsymbol{x}(t)^H \underline{\boldsymbol{x}}(t) && (14)\\ \boldsymbol{y}(t) &= \boldsymbol{W}(t-1)^H \boldsymbol{x}(t) && (15)\\ \underline{\boldsymbol{x}}'(t) &= \boldsymbol{C}_{xx}(t-1)\, \underline{\boldsymbol{x}}(t) && (16)\\ \underline{\boldsymbol{y}}'(t) &= \boldsymbol{W}(t-1)^H \underline{\boldsymbol{x}}'(t). && (17) \end{aligned}$$

[1] In this paper, a flop is a multiply / accumulate (MAC) operation.

Substituting equations (1) and (2) into equation (13) yields

$$\underline{\boldsymbol{C}}'_{yy}(t) = \left[\begin{array}{c|c} \boldsymbol{C}'_{yy}(t) & \underline{\boldsymbol{y}}''(t) \\ \hline \underline{\boldsymbol{y}}''(t)^H & \underline{c}_{yy}(t) \end{array}\right] \tag{18}$$

where

$$\begin{aligned} \boldsymbol{C}'_{yy}(t) &= \beta\, \boldsymbol{C}_{yy}(t-1) + \boldsymbol{y}(t)\, \boldsymbol{y}(t)^H && (19)\\ \underline{\boldsymbol{y}}''(t) &= \beta\, \underline{\boldsymbol{y}}'(t) + \boldsymbol{y}(t)\, \underline{\boldsymbol{\alpha}}(t) && (20)\\ \underline{c}_{yy}(t) &= \beta\, \underline{\boldsymbol{x}}(t)^H \underline{\boldsymbol{x}}'(t) + \underline{\boldsymbol{\alpha}}(t)^H \underline{\boldsymbol{\alpha}}(t). && (21) \end{aligned}$$

Finally, substituting equations (11) and (18) into equation (12) shows that

$$\underline{\boldsymbol{C}}_{yy}(t) = \left[\begin{array}{c|c} \boldsymbol{C}'_{yy}(t) & -\underline{\boldsymbol{g}}(t) \\ \hline -\underline{\boldsymbol{g}}(t)^H & \underline{\gamma}'(t) \end{array}\right] \tag{22}$$

where

$$\begin{aligned} \underline{\boldsymbol{g}}(t) &= -\underline{\boldsymbol{h}}(t)\, \underline{\sigma}(t)^{-1} && (23)\\ \underline{\gamma}'(t) &= \underline{\sigma}(t)^{-H} \underline{\gamma}(t)\, \underline{\sigma}(t)^{-1} && (24)\\ \underline{\boldsymbol{h}}(t) &= \underline{\boldsymbol{y}}''(t) - \boldsymbol{C}'_{yy}(t)\, \underline{\boldsymbol{y}}(t) && (25)\\ \underline{\gamma}(t) &= \underline{c}_{yy}(t) - \underline{\boldsymbol{y}}(t)^H \underline{\boldsymbol{h}}(t) - \underline{\boldsymbol{y}}''(t)^H \underline{\boldsymbol{y}}(t) && (26) \end{aligned}$$

3.3. Update of $\boldsymbol{W}(t)$

Let $\underline{\boldsymbol{\phi}}(t)$ be an $(r+p) \times p$ orthonormal matrix whose columns span the p-dimensional dominant subspace of the positive definite matrix $\underline{\boldsymbol{C}}_{yy}(t)$ of dimension $(r+p) \times (r+p)$. In particular, there is a $p \times p$ positive definite matrix $\underline{\lambda}(t)$ such that

$$\underline{\boldsymbol{C}}_{yy}(t)\, \underline{\boldsymbol{\phi}}(t) = \underline{\boldsymbol{\phi}}(t)\, \underline{\lambda}(t). \tag{27}$$

Let $\underline{\boldsymbol{\varphi}}(t)$ be the $r \times p$ matrix containing the r first rows of $\underline{\boldsymbol{\phi}}(t)$, and $\underline{z}(t)$ be the $p \times p$ matrix containing its p last rows:

$$\underline{\boldsymbol{\phi}}(t)^T = \left[\underline{\boldsymbol{\varphi}}(t)^T,\, \underline{z}(t)^T\right]. \tag{28}$$

Let

$$\underline{z}(t) = \underline{\rho}(t)\, \underline{\theta}(t) \tag{29}$$

be the polar decomposition of $\underline{z}(t)$, where $\underline{\rho}(t)$ is positive definite and $\underline{\theta}(t)$ is orthonormal. Let[2]

$$\begin{aligned} \underline{\boldsymbol{f}}(t) &= \underline{\boldsymbol{\varphi}}(t)\, \underline{\theta}(t)^H && (30)\\ \underline{\boldsymbol{f}}'(t) &= \underline{\boldsymbol{f}}(t) \left(\underline{I} + \underline{\rho}(t)\right)^{-1}. && (31) \end{aligned}$$

Then it can be readily verified that the $(r+p) \times r$ matrix

$$\boldsymbol{U}(t) = \left[\begin{array}{c} \boldsymbol{I}_r - \underline{\boldsymbol{f}}'(t)\, \underline{\boldsymbol{f}}(t)^H \\ \hline -\underline{\boldsymbol{f}}(t)^H \end{array}\right] \tag{32}$$

is orthonormal and satisfies $\boldsymbol{U}(t)^H \underline{\boldsymbol{\phi}}(t) = \boldsymbol{0}$. Therefore $\boldsymbol{U}(t)$ is an orthonormal basis of the r-dimensional minor subspace of $\underline{\boldsymbol{C}}_{yy}(t)$. Substituting equations (9) and (32) into equation (3) shows a recursion for the subspace weighting matrix:

$$\boldsymbol{W}(t) = \boldsymbol{W}(t-1) - \underline{\boldsymbol{e}}'(t)\, \underline{\boldsymbol{f}}(t)^H \tag{33}$$

where

$$\underline{\boldsymbol{e}}'(t) = \underline{\boldsymbol{u}}(t) + \boldsymbol{W}(t-1)\, \underline{\boldsymbol{f}}'(t). \tag{34}$$

[2] Since $\underline{\rho}(t)$ is positive definite, the $p \times p$ matrix $\underline{I} + \underline{\rho}(t)$ is also positive definite. In particular, $\underline{I} + \underline{\rho}(t)$ is non-singular.

3.4. Update of $\boldsymbol{C}_{yy}(t)$

The auxiliary matrix $\boldsymbol{C}_{yy}(t)$ can also be efficiently updated. Indeed, substituting equations (3) and (5) into equation (6) yields

$$\boldsymbol{C}_{yy}(t) = \boldsymbol{U}(t)^H \underline{\boldsymbol{C}}_{yy}(t) \boldsymbol{U}(t). \tag{35}$$

Substituting equations (31), (30), and (28) into (32) shows that

$$\boldsymbol{U}(t) = \left[\begin{array}{c} -\underline{\boldsymbol{I}}_r \\ \hline -\underline{\boldsymbol{f}}'(t)^H \end{array}\right] - \underline{\boldsymbol{\phi}}(t)\, \underline{\theta}(t)^H \underline{\boldsymbol{f}}'(t)^H. \tag{36}$$

Thus substituting equations (36), (22), (27) into (35) yields

$$\boldsymbol{C}_{yy}(t) = \boldsymbol{C}'_{yy}(t) + \underline{\boldsymbol{g}}'(t)\, \underline{\boldsymbol{f}}'(t)^H + \underline{\boldsymbol{f}}'(t)\, \underline{\boldsymbol{g}}(t)^H \tag{37}$$

where

$$\underline{\boldsymbol{g}}'(t) = \underline{\boldsymbol{g}}(t) + \underline{\boldsymbol{f}}'(t) \left(\underline{\gamma}'(t) - \underline{\theta}(t) \underline{\lambda}(t) \underline{\theta}(t)^H\right). \tag{38}$$

Table 1. Pseudo-code of the YAST algorithm

	eq.:	flops:
$\boldsymbol{y}(t) = \boldsymbol{W}(t-1)^H \boldsymbol{x}(t)$		nr
$\boldsymbol{x}'(t) = \boldsymbol{C}_{xx}(t-1)\, \boldsymbol{x}(t)$		$9n$
$\boldsymbol{y}'(t) = \boldsymbol{W}(t-1)^H \boldsymbol{x}'(t)$		nr
switch p {		
case 1: $\underline{\boldsymbol{x}}(t) = \boldsymbol{x}(t),\ \underline{\boldsymbol{x}}'(t) = \boldsymbol{x}'(t),$		
$\underline{\boldsymbol{y}}(t) = \boldsymbol{y}(t),\ \underline{\boldsymbol{y}}'(t) = \boldsymbol{y}'(t).$		
case 2: $\boldsymbol{x}''(t) = \boldsymbol{C}_{xx}(t-1)^2\, \boldsymbol{x}(t),$		$32n$
$\boldsymbol{y}''(t) = \boldsymbol{W}(t-1)^H \boldsymbol{x}''(t),$		nr
$\underline{\boldsymbol{x}}(t) = \left[\boldsymbol{x}(t), \boldsymbol{x}'(t)\right], \underline{\boldsymbol{x}}'(t) = \left[\boldsymbol{x}'(t), \boldsymbol{x}''(t)\right],$		
$\underline{\boldsymbol{y}}(t) = \left[\boldsymbol{y}(t), \boldsymbol{y}'(t)\right], \underline{\boldsymbol{y}}'(t) = \left[\boldsymbol{y}'(t), \boldsymbol{y}''(t)\right].$}		
$\underline{\boldsymbol{e}}(t) = \underline{\boldsymbol{x}}(t) - \boldsymbol{W}(t-1)\, \underline{\boldsymbol{y}}(t)$	(7)	pnr
$\underline{\boldsymbol{u}}(t)\, \underline{\sigma}(t) = \underline{\boldsymbol{e}}(t)$	(8)	$2p^2 n$
$\underline{\boldsymbol{\alpha}}(t) = \boldsymbol{x}(t)^H \underline{\boldsymbol{x}}(t)$	(14)	pn
$\underline{\boldsymbol{y}}''(t) = \beta \underline{\boldsymbol{y}}'(t) + \boldsymbol{y}(t)\, \underline{\boldsymbol{\alpha}}(t)$	(20)	pr
$\underline{c}_{yy}(t) = \beta \underline{\boldsymbol{x}}(t)^H \underline{\boldsymbol{x}}'(t) + \underline{\boldsymbol{\alpha}}(t)^H \underline{\boldsymbol{\alpha}}(t)$	(21)	$p^2 n$
$\boldsymbol{C}'_{yy}(t) = \beta \boldsymbol{C}_{yy}(t-1) + \boldsymbol{y}(t)\, \boldsymbol{y}(t)^H$	(18)	r^2
$\underline{\boldsymbol{h}}(t) = \underline{\boldsymbol{y}}''(t) - \boldsymbol{C}'_{yy}(t)\, \underline{\boldsymbol{y}}(t)$	(25)	pr^2
$\underline{\gamma}(t) = \underline{c}_{yy}(t) - \underline{\boldsymbol{y}}(t)^H \underline{\boldsymbol{h}}(t) - \underline{\boldsymbol{y}}''(t)^H \underline{\boldsymbol{y}}(t)$	(26)	$2p^2 r$
$\underline{\boldsymbol{g}}(t) = -\underline{\boldsymbol{h}}(t)\, \underline{\sigma}(t)^{-1}$	(23)	$p^2 r$
$\underline{\gamma}'(t) = \underline{\sigma}(t)^{-H} \underline{\gamma}(t)\, \underline{\sigma}(t)^{-1}$	(24)	$2p^3$
$\underline{\boldsymbol{C}}_{yy}(t) = \left[\boldsymbol{C}'_{yy}(t), -\underline{\boldsymbol{g}}(t); -\underline{\boldsymbol{g}}(t)^H, \underline{\gamma}'(t)\right]$	(22)	
$\left(\underline{\boldsymbol{\phi}}(t), \underline{\lambda}(t)\right) = \mathrm{eigs}\left(\underline{\boldsymbol{C}}_{yy}(t), p\right)$	(27)	$O(pr^2)$
$\left[\underline{\boldsymbol{\varphi}}(t)^T, \underline{z}(t)^T\right] = \underline{\boldsymbol{\phi}}(t)^T$	(28)	
$\left(\underline{\rho}(t), \underline{\theta}(t)\right) = \mathrm{polar}\left(\underline{z}(t)\right)$	(29)	$O(p^3)$
$\underline{\boldsymbol{f}}(t) = \underline{\boldsymbol{\varphi}}(t)\, \underline{\theta}(t)^H$	(30)	$p^2 r$
$\underline{\boldsymbol{f}}'(t) = \underline{\boldsymbol{f}}(t) \left(\underline{I} + \underline{\rho}(t)\right)^{-1}$	(31)	$p^2 r$
$\underline{\boldsymbol{e}}'(t) = \underline{\boldsymbol{u}}(t) + \boldsymbol{W}(t-1) \left(\underline{\boldsymbol{f}}'(t) - \boldsymbol{W}(t-1)^H \underline{\boldsymbol{u}}(t)\right)$	(39)	$2pnr$
$\boldsymbol{W}(t) = \boldsymbol{W}(t-1) - \underline{\boldsymbol{e}}'(t)\, \underline{\boldsymbol{f}}(t)^H$	(33)	pnr
$\underline{\boldsymbol{g}}'(t) = \underline{\boldsymbol{g}}(t) + \underline{\boldsymbol{f}}'(t) \left(\underline{\gamma}'(t) - \underline{\theta}(t) \underline{\lambda}(t) \underline{\theta}(t)^H\right)$	(38)	$p^2 r$
$\boldsymbol{C}_{yy}(t) = \boldsymbol{C}'_{yy}(t) + \underline{\boldsymbol{g}}'(t)\, \underline{\boldsymbol{f}}'(t)^H + \underline{\boldsymbol{f}}'(t)\, \underline{\boldsymbol{g}}(t)^H$	(37)	$2pr^2$
Total:		$(5p+1)nr + O(n+r^2)$

3.5. Implementation

The YAST algorithm as presented above shares one drawback with some other minor subspace trackers : a numerical instability makes the subspace weighting matrix slowly deviate from orthonormality. In the literature, this deviation is often corrected by using Householder transformations [7–9]. Here, we observed that this deviation is due to rounding errors in the normalization of $\underline{\boldsymbol{e}}(t)$ in equation (8). Because of the divisions involved in this operation, $\underline{\boldsymbol{u}}(t)$ is not exactly orthogonal to $\boldsymbol{W}(t-1)$ in practice. Nevertheless, the orthogonality can be restored by projecting $\underline{\boldsymbol{u}}(t)$ onto the orthogonal complement of $\mathrm{span}\left(\boldsymbol{W}(t)\right)$: $\underline{\boldsymbol{u}}(t) = \left(\boldsymbol{I}_n - \boldsymbol{W}(t)\, \boldsymbol{W}(t)^H\right) \underline{\boldsymbol{u}}(t)$. Substituting this last equation into equation (34) yields

$$\underline{\boldsymbol{e}}'(t) = \underline{\boldsymbol{u}}(t) + \boldsymbol{W}(t-1) \left(\underline{\boldsymbol{f}}'(t) - \boldsymbol{W}(t-1)^H \underline{\boldsymbol{u}}(t)\right). \tag{39}$$

Thus replacing equation (34) by equation (39) makes the YAST algorithm numerically very stable. Regarding the computational complexity, the calculation cost of the vectors $\boldsymbol{x}'(t) = \boldsymbol{C}_{xx}(t-1)\, \boldsymbol{x}(t)$ and $\boldsymbol{x}''(t) = \boldsymbol{C}_{xx}(t-1)^2 \boldsymbol{x}(t)$ is normally $O(n^2)$. In the case of time series analysis, this cost can be reduced to $O(n)$ by means of the technique described in [3], which exploits the shift invariance property of the correlation matrix. Thus the global cost of the YAST algorithm for MSA of time series is $6nr$ flops in the case $p = 1$, whereas that of NOOja [8] and NFRANS [9] is $4nr$, that of CAL [5] is $6nr$, and that of ODKA [7] is $7nr$. In the case $p = 2$, the overall cost of YAST is $11nr$, which remains one order of magnitude lower than that of the $O(n^2 r)$ QRI algorithm [6].

4. SIMULATION RESULTS

4.1. A classical example

In the following, $\boldsymbol{x}(t)$ is a sequence of $n = 4$ dimensional independent jointly-Gaussian random vectors, with zero mean and covariance matrix

$$\boldsymbol{C} = \begin{bmatrix} 0.9 & 0.4 & 0.7 & 0.3 \\ 0.4 & 0.3 & 0.5 & 0.4 \\ 0.7 & 0.5 & 1.0 & 0.6 \\ 0.3 & 0.4 & 0.6 & 0.9 \end{bmatrix}.$$

We choose $r = 2$ and $\boldsymbol{W}(0) = \begin{bmatrix} \boldsymbol{I_r} \\ \boldsymbol{0}_{(n-r,r)} \end{bmatrix}$, as recommended in [5]. As in [6–9], we calculate the ensemble averages of the performance factors

$$\rho(t) = \frac{1}{K} \sum_{k=1}^{K} \frac{\mathrm{tr}\left(\boldsymbol{W}_k(t)^H \boldsymbol{E}_1\, \boldsymbol{E}_1^H \boldsymbol{W}_k(t)\right)}{\mathrm{tr}\left(\boldsymbol{W}_k(t)^H \boldsymbol{E}_2\, \boldsymbol{E}_2^H \boldsymbol{W}_k(t)\right)},$$

$$\eta(t) = \frac{1}{K} \sum_{k=1}^{K} \left\| \boldsymbol{W}_k(t)^H \boldsymbol{W}_k(t) - \boldsymbol{I}_r \right\|_F^2,$$

where the number of algorithm runs is $K = 50$, k indicates that the associated variable depends on the particular run, $\|.\|_F$ denotes the Frobenius norm, $\boldsymbol{E}_1$ is the exact $(n-r)$-dimensional principal subspace, and $\boldsymbol{E}_2$ is the exact r-dimensional minor subspace of $\boldsymbol{C}$. The functions $\rho(t)$ and $\eta(t)$ measure the estimation error and the departure from orthogonality of the subspace weighting matrix.

Figure 1-a shows the tracking results obtained with four minor subspace trackers: QRI [6], ODKA [7], NOOja [8], and YAST[3]. It can be noticed that YAST converges faster than the three other algorithms. Besides, the estimation error reached by YAST and QRI at convergence is lower than that of ODKA and NOOja. Figure 1-b shows the departures from orthogonality of the four abovementioned subspace trackers. It can be noticed that QRI and ODKA are very stable, whereas the orthonormality error of YAST and NOOja slowly increases. Nevertheless, this error remains of the order of the machine precision after 10000 iterations.

[3]The QRI algorithm was implemented with parameter $\alpha = 0.99$, ODKA with $\mu = 0.01$, NOOja with $\beta = 0.05$ and $\gamma = 0.4$, and YAST with $\beta = 0.99$ and $p = 1$.

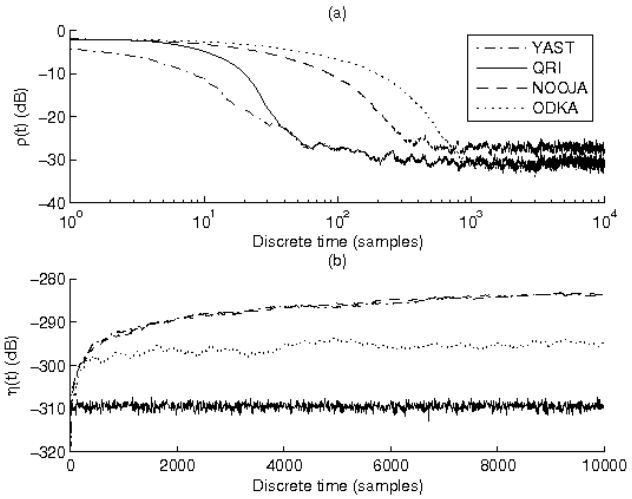


Fig. 1. A classical example

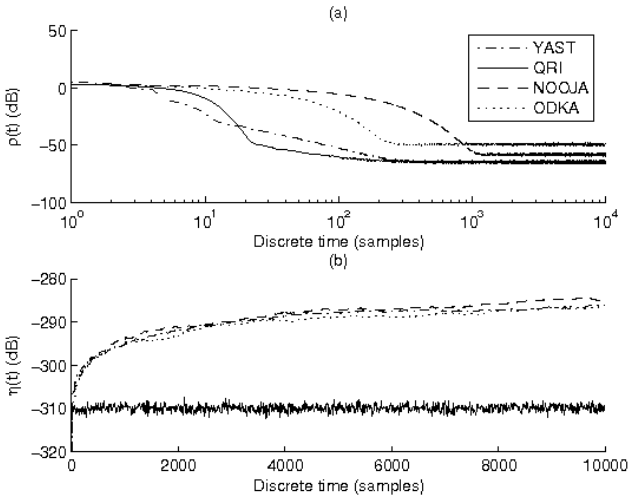


Fig. 2. Time series analysis

Note that this example, usually encountered in the literature, does not belong to the field of times series analysis. Therefore the optimization mentioned in section 3.5 cannot be applied, and the global complexity of YAST becomes pn^2. Nevertheless, this cost remains lower than that of the QRI algorithm, which is $O(n^2r)$. Below, the YAST algorithm is applied to frequency estimation, so that its complexity is only $(5p+1)nr$.

4.2. Time series analysis

In this section, the test signal is a sum of 4 complex sinusoidal sources, of same unitary amplitude, random phases, and normalized frequencies equal to 0.2, 0.4, 0.5 and 0.8. This signal is perturbed by an additive white gaussian noise, so that the SNR is 30 dB. The data vectors are composed of $n = 6$ successive samples of the noisy signal, so that the dimension of the noise subspace is $r = 2$. However, the $n \times n$ matrix $\boldsymbol{C}$ is defined as the covariance matrix of the noiseless data vectors. The performance of the various subspace trackers is measured by the functions $\rho(t)$ and $\eta(t)$ defined above.

Figure 2-a shows the tracking results obtained with the four minor subspace trackers[4]. It can be noticed that the convergence properties of YAST and QRI are similar. Again, the estimation error reached by YAST and QRI at convergence is lower than that of ODKA and NOOja. Figure 2-b shows the departures from orthogonality of the four above-mentioned subspace trackers. It can be noticed that QRI is more stable than the other algorithms. Nevertheless, the orthonormality errors of all these algorithms remain of the order of the machine precision after 10000 iterations.

5. CONCLUSIONS

In this paper, a new algorithm for minor subspace tracking was presented, which is derived from the YAST principal subspace tracker. In the case of time series analysis, this algorithm reaches the linear complexity $O(nr)$. Moreover, it greatly outperforms classical minor subspace trackers of the same complexity, and guarantees the orthonormality of the subspace weighting matrix at each time step. It can also be adapted to the minor subspace tracking of a sliding-window correlation matrix[5].

[4]The QRI algorithm was implemented with parameter $\alpha = 0.99$, ODKA with $\mu = 0.01$, NOOja with $\beta = 0.05$ and $\gamma = 0.4$, and YAST with $\beta = 0.99$ and $p = 2$.

[5]In this case, the column vector $\boldsymbol{x}(t-l)$ must be appended to the directional search matrix $\underline{\boldsymbol{x}}(t)$, where l is the window length.

6. REFERENCES

[1] X. Wang and V. H. Poor, "Blind multiuser detection: A subspace approach," *IEEE Trans. Inform. Theory*, vol. 44, pp. 677–689, Mar. 1998.

[2] R. Badeau, B. David, and G. Richard, "Yet Another Subspace Tracker," in *Proc. of ICASSP'05*, Philadelphia, PA, USA, Mar. 2005, vol. 4, pp. 329–332, IEEE.

[3] C. E. Davila, "Efficient, high performance, subspace tracking for time-domain data," *IEEE Trans. Signal Processing*, vol. 48, no. 12, pp. 3307–3315, Dec. 2000.

[4] E. Oja, "Principal components, minor components, and linear neural networks," *Neural Networks*, vol. 5, pp. 927–935, Nov./Dec. 1992.

[5] T.P. Chen, S-I. Amari, and Q. Lin, "A unified algorithm for principal and minor components extraction," *Neural Networks*, vol. 11, no. 3, pp. 385–390, 1998.

[6] P. Strobach, "Square root QR inverse iteration for tracking the minor subspace," *IEEE Trans. Signal Processing*, vol. 48, no. 11, pp. 2994–2999, 2000.

[7] S. C. Douglas, "Numerically-robust adaptive subspace tracking using Householder transformations," in *Proc. of Sensor Array and Multichannel Signal Proc. Workshop*, 2000, pp. 499–503.

[8] S. Attallah and K. Abed-Meraim, "Fast algorithms for subspace tracking," *IEEE Signal Proc. Letters*, vol. 8, no. 7, 2001.

[9] S. Attallah, "Adaptive algorithms for noise subspace estimation with application to MC-CDMA," in *Proc. of the Seventh International Symposium on Signal Processing and its Applications ISSPA'03*, July 2003, vol. 1, pp. 501–504.