FAST ALGORITHMS FOR MINOR COMPONENT ANALYSIS

S. Bartelmaos, K. Abed-Meraim* and S. Attallah***

*ENST-Paris, TSI Department , 37/39 rue Dareau 75014, Paris Cedex 14, France
**NUS university, Dept. of Elec. & Comp. Eng, 4 Engineering Drive 3 Singapore 117576

ABSTRACT

In this paper, we propose new adaptive algorithms for the extraction and tracking of the least (minor) eigenvectors of a positive Hermitian covariance matrix. The proposed algorithms are said fast in the sense that their computational cost is of order $O(np)$ flops per iteration where n is the size of the observation vector and $p < n$ is the number of minor eigenvectors we need to estimate.

Two classes of algorithms are considered : namely the PASTd (Projection Approximation Subspace Tracking with deflation) that is derived using projection approximation in conjunction with power iteration and the Oja that uses stochastic gradient technique. Using appropriate fast orthogonalization techniques we introduce for each class new fast algorithms that extract the minor eigenvectors and guarantee the orthogonality of the weight matrix at each iteration.

1. INTRODUCTION

Principal and minor component analysis (PCA and MCA), which are part of the more general principal and minor subspace (PSA and MSA) analysis, are two important problems that are frequently encountered in many information processing fields. In our case we are interested in the minor component analysis.

From the computational point of view, we may distinguish between methods requiring $O(n^2p)$, $O(n^2)$, $O(np^2)$, or $O(np)$ operations per update. The wide range of the computational complexity is due to the fact that some algorithms update the complete eigenstructure, with or without the explicit computation of the sample correlation matrix, whereas other ones track only the desired principal or minor subspace. For example, the parallel method proposed by Moonen [1], which updates the SVD by interlaced QR triangularization and Jacobi rotations, requires $O(n^2)$ operations. The gradient type algorithms track either the principal or the minor subspace. They demand $O(np)$ operations for the gradient-ascent or gradient-descent step and additional $O(np^2)$ operations for the orthogonalization of the eigenvector estimates.

In this paper, we apply a fast orthogonalization technique to the (MCA-Oja) by conserving the same order $O(np)$ of the computational complexity. This orthogonalization results in a better numerical stability and estimation accuracy. The rest of the paper is about the modification of the orthonormal PASTd algorithm [2] with the aim of reducing its computational complexity from $O(np^2)$ to $O(np)$ flops per iteration.

2. ORTHOGONAL MCA-OJA

Let $\mathbf{x}(k)$ be a sequence of $n \times 1$ random vectors with covariance matrix $\mathbf{C} = E[\mathbf{x}(k)\mathbf{x}^H(k)]$. Consider the problem of extracting $p < n$ minor eigenvectors of the covariance matrix. To solve this problem, several subspace extraction algorithms have so far been proposed [3] including the Oja method. The minor component analysis algorithm of Oja (MCA-OJA) existing in the literature can be formulated as ([4],[5]).

$$\mathbf{W}(i) = \mathbf{W}(i-1) - \beta(\mathbf{x}(i)\mathbf{z}^H(i) - \mathbf{z}_p(i)\mathbf{y}^H(i)). \quad (1)$$

Where $\mathbf{W}(i) \in \mathbf{C}^{n\times p}$ is the minor subspace estimate, $\mathbf{y}(i) \triangleq \mathbf{W}^H(i-1)\mathbf{x}(i)$, $\mathbf{z}(i) \triangleq \mathbf{L}\mathbf{y}(i)$ and $\mathbf{z}_p(i) \triangleq \mathbf{W}(i-1)\mathbf{z}(i)$, where $\mathbf{L}$ is a pxp positive diagonal matrix containing elements with different weight for extracting the p minor eigenvectors *i.e.* $L = diag(l_1, \dots, l_p)$ with $l_1 > l_2 \cdots > l_p > 0$. $\beta > 0$ is the step size of (MCA-Oja) method. Equation (1) represents the updating of the weight matrix $\mathbf{W}(i)$ at the i-th iteration. The MCA-Oja algorithm is numerically unstable (see Figure 1). We propose here to stabilize it using orthogonalization of the weight matrix at each step. Moreover, orthogonality is an important property that is desired in many subspace based estimation methods [6]. To this end, we set (using informal notation):

$$\mathbf{W}(i) := \mathbf{W}(i)(\mathbf{W}^H(i)\mathbf{W}(i))^{-1/2}. \quad (2)$$

The fast computation of (2) is obtained thanks to the following result. (the proof can be find in [7])

Lemma 1*: 1- Let $\mathbf{R}$ be a d-rank hermitian matrix spanned by the column vectors $\mathbf{p}_1, \dots, \mathbf{p}_d$, then the eigendecomposition of $\mathbf{R}$ is given by $\mathbf{R} = \mathbf{E}\mathbf{D}\mathbf{E}^H$, where $\mathbf{D} = diag(\lambda_1 \dots \lambda_d)$*

and $\mathbf{E} = [\mathbf{e}_1 \dots \mathbf{e}_d]$ *are computed by*

$$\begin{array}{rcl} \mathbf{E} & = & \mathbf{PT} \\ \mathbf{P} & \triangleq & [\mathbf{p}_1 \dots \mathbf{p}_d] \\ \mathbf{M} & = & (\mathbf{P}^H\mathbf{P})^{-1}\mathbf{P}^H\mathbf{RP} \quad = \quad \mathbf{TDT}^{-1} \end{array}$$

2- Let $\mathbf{N} = \mathbf{I} + \mathbf{EDE}^H$ *where* $\mathbf{E}$ *is orthonormal. Then, an inverse square root of* $\mathbf{N}$ *is given by*

$$\mathbf{N}^{-\frac{1}{2}} = \mathbf{I} + \mathbf{ED}^{'}\mathbf{E}^H$$

where

$$\mathbf{D}^{'} = diag(\frac{1}{\sqrt{1+\lambda_1}} - 1 \dots \frac{1}{\sqrt{1+\lambda_d}} - 1)$$

By applying *Lemma 1* to $\mathbf{W^H}(i)$ (see appendix for details), we get the Orthogonal version of MCA-Oja summarized in Table 1.

$$\begin{array}{rcl} \mathbf{y}(i) & = & \mathbf{W}^H(i-1)\mathbf{x}(i) \\ \mathbf{z}(i) & = & \mathbf{Ly}(i) \\ \mathbf{z_p}(i) & = & \mathbf{W}(i-1)\mathbf{y}(i) \\ \mathbf{R} & = & \beta^2(\|\mathbf{x}(i)\|^2\mathbf{z}(i)\mathbf{z}^H(i) + \|\mathbf{z}_p(i)\|^2\mathbf{y}(i)\mathbf{y}^H(i) - \\ & & \mathbf{x}^H(i)\mathbf{z}_p(i)\mathbf{z}(i)\mathbf{y}^H(i) - \mathbf{z}_p^H(i)\mathbf{x}(i)\mathbf{y}(i)\mathbf{z}^H(i)) \\ \mathbf{P} & = & [\mathbf{y}(i) \;\; \mathbf{z}(i)] \\ \mathbf{M} & = & (\mathbf{P}^H\mathbf{P})^{-1}\mathbf{P}^H\mathbf{RP} \\ \mathbf{eig}(\mathbf{M}) & = & \mathbf{T}diag(\lambda_1, \lambda_2)\mathbf{T}^{-1} \\ \mathbf{E} & = & [\mathbf{e}_1 \; \mathbf{e}_2] = \mathbf{PT} \\ \mathbf{\Sigma} & = & diag(\frac{1}{\|\mathbf{e}_1\|}, \frac{1}{\|\mathbf{e}_2\|}) \\ \mathbf{E}^{'} & = & \mathbf{E\Sigma} \triangleq \left[\mathbf{e}_1^{'} \; \mathbf{e}_2^{'}\right] \qquad \mathbf{T}^{'} = \mathbf{T\Sigma} \\ \tau_1 & = & \frac{1}{\sqrt{1+\lambda_1}} - 1 \qquad \tau_2 = \frac{1}{\sqrt{1+\lambda_2}} - 1 \\ \mathbf{T}^{'-1} & = & \left[\begin{array}{cc} t_{11} & t_{12} \\ t_{21} & t_{22} \end{array}\right] \\ \mathbf{E_p} & = & \mathbf{W}(i-1)\mathbf{E}^{'} \triangleq [\mathbf{e}_{p_1} \; \mathbf{e}_{p_2}] \\ \mathbf{p} & = & (\tau_1 + \beta(1+\tau_1)t_{11}t_{12})\mathbf{e}_{p_1} + \\ & & \beta(1+\tau_1)t_{11}t_{22}\mathbf{e}_{p_2} - \beta(1+\tau_1)t_{12}\mathbf{x}(i) \\ \mathbf{q} & = & (\tau_2 + \beta(1+\tau_2)t_{21}t_{22})\mathbf{e}_{p_2} + \\ & & \beta(1+\tau_2)t_{21}t_{12}\mathbf{e}_{p_1} - \beta(1+\tau_2)t_{22}\mathbf{x}(i) \\ \mathbf{W}(i) & = & \mathbf{W}(i-1) + \mathbf{pe}_1^{'H} + \mathbf{qe}_2^{'H}. \end{array}$$

Table 1. The MCA-OOja.

3. MCA-OJA USING HOUSEHOLDER TRANSFORM

In terms of orthogonality errors, the MCA-OOja algorithm guarantees the orthogonality of the minor subspace at each iteration. This insures much more stability to the algorithm compared to its original version given by (1). However this improvement doesn't mean absolute stability and the algorithm remains sensitive to numerical rounding errors. This can be shown by theoretical derivation using a similar analysis to that in [8] and is illustrated here by the simulation example of Figure 2 To mitigate the effect of rounding errors, we propose to use the numerically well-behaved Householder orthogonal matrices [9]. To this end, we proved the following result:

Lemma 2*: The updating equation of the weight matrix given by Table 1, can be reformulated as:*

$$\mathbf{W}(i) = \mathbf{H}_1(i)\mathbf{H}_2(i)\mathbf{W}(i-1) \tag{3}$$

Where $\mathbf{H}_1(i)$ *and* $\mathbf{H}_2(i)$ *are the Householder transforms given by*

$$\begin{array}{c} \mathbf{H}_1(i) \triangleq \mathbf{I} - 2\mathbf{u}(i)\mathbf{u}^H(i). \\ \mathbf{H}_2(i) \triangleq \mathbf{I} - 2\mathbf{v}(i)\mathbf{v}^H(i). \end{array}$$

Where $\mathbf{u}(i)$ *(resp.* $\mathbf{v}(i)$*) is the principal left singular eigenvector of* $\mathbf{R} \triangleq \mathbf{W}(i) - \mathbf{W}(i-1)$ *(resp. of* $\mathbf{H}_1(i)\mathbf{W}(i) - \mathbf{W}(i-1)$*).*

The fast computation of $\mathbf{u}(i)$ and $\mathbf{v}(i)$ of order $O(n)$ is given in Table 2 (see appendix for details).
Hence, MCA-OOjaH (H stands for Householder) algorithm can be obtained by adding the equations of Table 2 to the previous table excluding the updating equation of the minor subspace weight matrix. In table 2, we used the notation $\mathbf{Z}(:,1)$ to denote the first column vector of $\mathbf{Z}$, $\mathbf{w}_1(i-1)$ to denote the first column vector of $\mathbf{W}(i-1)$ and $\mathbf{e}_1^{'*}$ to denote the complex conjugate of $\mathbf{e}_1^{'}$.

$$\begin{array}{rcl} \mathbf{Q} & = & [\mathbf{p} \; \mathbf{q}] \\ \Lambda & = & \left[\begin{array}{cc} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{11} \end{array}\right] \triangleq \\ & & \left[\begin{array}{cc} \|\mathbf{e}_1^{'}\|^2\|\mathbf{p}\|^2 + \mathbf{e}_1^{'H}\mathbf{e}_2^{'}\mathbf{q}^H\mathbf{p} & \mathbf{e}_2^{'H}\mathbf{e}_1^{'}\|\mathbf{p}\|^2 + \|\mathbf{e}_2^{'}\|^2\mathbf{q}^H\mathbf{p} \\ \|\mathbf{e}_1^{'}\|^2\mathbf{p}^H\mathbf{q} + \mathbf{e}_1^{'H}\mathbf{e}_2^{'}\|\mathbf{q}\|^2 & \mathbf{e}_2^{'H}\mathbf{e}_1^{'}\mathbf{p}^H\mathbf{q} + \|\mathbf{e}_2^{'}\|^2\|\mathbf{q}\|^2 \end{array}\right] \\ \mathbf{Q}_2 & \triangleq & [\lambda_{11}\mathbf{p} + \lambda_{12}\mathbf{q} \quad \lambda_{21}\mathbf{p} + \lambda_{22}\mathbf{q}] \\ \mathbf{R}_q & = & (\mathbf{Q}^H\mathbf{Q})^{-1}\mathbf{Q}^H\mathbf{Q}_2 \\ eig(\mathbf{R}_q) & = & \mathbf{T}_1 diag(\lambda_{q_1}, \lambda_{q_2})\mathbf{T}_1^{-1} \\ \mathbf{H} & = & [\mathbf{h}_1 \;\; \mathbf{h}_2] = \mathbf{QT}_1 \\ \mathbf{u} & = & \frac{\mathbf{h}_1}{\|\mathbf{h}_1\|} \\ \mathbf{Z}(:,1) & = & 2\mathbf{u}(\mathbf{u}^H\mathbf{pe}_1^{'*}(1) + \mathbf{u}^H\mathbf{pe}_2^{'*}(1) + \mathbf{u}^H\mathbf{w}_1(i-1)) \\ & & - \mathbf{pe}_1^{'*}(1) - \mathbf{qe}_2^{'*}(1) \\ \mathbf{v} & = & \frac{\mathbf{Z(:,1)}}{\|\mathbf{Z(:,1)}\|} \\ \mathbf{W}(i) & = & (\mathbf{I} - 2\mathbf{uu}^H)(\mathbf{I} - 2\mathbf{vv}^H)\mathbf{W}(i-1) \end{array}$$

Table 2. The MCA-OOjaH.

The Householder MCA-OOja becomes numerically very stable, as illustrated by the simulation results of Figure 2.

4. MCA-PASTD

Oja-type algorithms are relatively slow in terms of convergence rate. PASTd algorithm is (usually a faster) alternative approach based on power iteration in conjunction with projection approximation. In [2], a PASTd algorithm with orthogonalization has been proposed for the MCA. Its complexity is of order $O(np^2)$ due to the Gramm Schmidt orthogonalization step (see Table 3), we refer to this algorithm by MCA-PASTd-np^2. We propose here a modified version of the algorithm in [2] that insures the orthogonality of the weight matrix with a complexity order of $O(np)$. The main idea is expressed by the following result:

Lemma 3*: Let consider the algorithm in Table 3 (without the Gramm Schmidt orthogonalization) where, at each iteration i, we modify the weight matrix*

$$\mathbf{W}(i) \triangleq [\mathbf{w}_1(i) \dots \mathbf{w}_p(i)]$$

in such a way to satisfy:

$\mathbf{w}_1(i) \perp \mathbf{w}_k(i)$ *for* $k = 2 \dots p$ ($\perp$ *stands for orthogonality*)

$\|\mathbf{w}_k(i)\| = 1$ *for* $k = 1 \dots p$
The weight matrix $\mathbf{W}(i) \triangleq \left[\begin{array}{ccc} \mathbf{w}_1(i) & \dots & \mathbf{w}_p(i) \end{array}\right]$ *is then orthogonal i.e.* $\mathbf{W}^H(i)\mathbf{W}(i) = \mathbf{I} \quad \forall\, i$.

The modified algorithm referred to as MCA-PASTD-np is summarized in Table 4

$For\ k$	$=$	$1 \dots p$
ϵ_k	$=$	$\mathbf{w}_k^H(i-1)\mathbf{x}(i)$
$q_k(i)$	$=$	$\alpha q_k(i-1) + \epsilon_k^2$
$\mathbf{\Psi}_k$	$=$	$\mathbf{x}(i) - \mathbf{w}_k(i-1)\epsilon_k$
$\mathbf{w}_k(i)$	$=$	$\mathbf{w}_k(i-1) - \frac{\mathbf{\Psi}_k \epsilon_k}{q_k(i)}$
If $k > 1$ $\mathbf{w}_k(i)$	$:=$	$\mathbf{w}_k(i) - \mathbf{W}_{1:k-1}(i)\mathbf{W}_{1:k-1}^H(i)\mathbf{w}_k(i)$
$\mathbf{w}_k(i)$	$:=$	$\frac{\mathbf{w}_k(i)}{\|\mathbf{w}_k(i)\|}$
$\mathbf{x}(i)$	$:=$	$\mathbf{x}(i) - \mathbf{w}_k(i)\epsilon_k$

Table 3. The MCA-PASTd-np^2.

$For\ k$	$=$	$1 \dots p$
ϵ_k	$=$	$\mathbf{w}_k^H(i-1)\mathbf{x}(i)$
$q_k(i)$	$=$	$\alpha q_k(i-1) + \epsilon_k^2$
$\mathbf{\Psi}_k$	$=$	$\mathbf{x}(i) - \mathbf{w}_k(i-1)\epsilon_k$
$\mathbf{w}_k(i)$	$=$	$\mathbf{w}_k(i-1) - \frac{\mathbf{\Psi}_k \epsilon_k}{q_k(i)}$
If $k = 1$ $\mathbf{w}_1(i)$	$:=$	$\mathbf{w}_1(i) - \mathbf{W}_{2:p}(i-1)\mathbf{W}_{2:p}^H(i-1)\mathbf{w}_1(i)$
$\mathbf{w}_k(i)$	$:=$	$\frac{\mathbf{w}_k(i)}{\|\mathbf{w}_k(i)\|}$
χ_k	$=$	$\mathbf{w}_k^H(i)\mathbf{x}(i)$
$\mathbf{x}(i)$	$:=$	$\mathbf{x}(i) - \mathbf{w}_k(i)\chi_k$

Table 4. The MCA-PASTd-np

5. SIMULATIONS

To assess the performance of our algorithms, we calculate the ensemble average of the performance factors

$$\rho(i) = \frac{1}{p.r_0} \sum_{r=1}^{r_0} \|\mathbf{W}_r(i) - \mathbf{E}_2\|^2 \tag{4}$$

$$\eta(i) = \frac{1}{r_0} \sum_{r=1}^{r_0} \|\mathbf{W}_r^H(i)\mathbf{W}_r(i) - \mathbf{I}\|_F^2 \tag{5}$$

where the number of algorithm runs is $r_0 = 100$, r indicates that the associated variable depends on the particular run. $\|.\|_F$ denotes the Frobenius norm, and $\mathbf{E_2}$ is the nxp matrix of the p minor eigenvectors. The first performance index ρ measures the averaged estimation accuracy of the eigenvectors while the second performance index η measures the orthogonality of the weight matrix. Note that, as the eigenvectors are estimated up to a phase indeterminacy, we remove this ambiguity (by forcing the first entry of each eigenvector to be positive) before the comparison in (4).
In the simulation experiment, we have considered an iid sequence of n-dimensional (with $n = 4$) random vectors $\mathbf{x}(i)$. The random sequence is generated using a zero mean Gaussian-distribution with positive definite covariance matrix $\mathbf{C}$ that is generated randomly at each run.
We extract here the $p = 2$ least eigenvectors of $\mathbf{C}$ using the proposed Oja-type methods with a step size $\beta = 0.01$ and the PASTd method with a forgetting factor $\alpha = 0.99$.

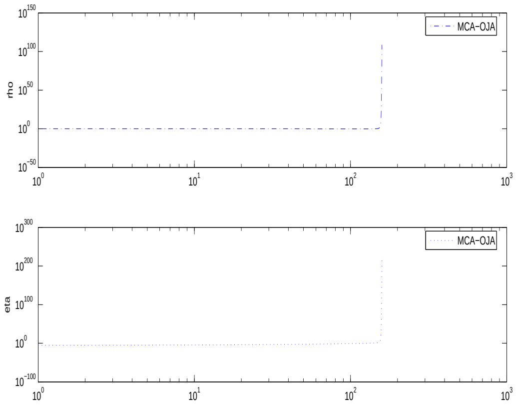

Fig. 1. Performance of MCA-Oja

Fig.1 shows the instability of MC-Oja. The algorithm diverges very quickly if no orthogonalization is performed. The orthogonalization step is necessary when we analyse the minor component.
In Fig.2, we compare the performance of MC-OOja with MC-OOjaH. We can see clearly the improvement of the numerical stability when we use Householder transform.

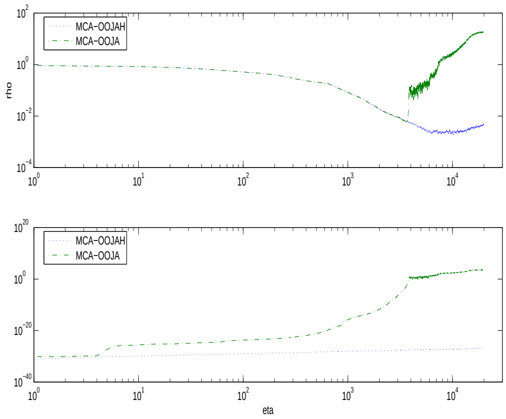

Fig. 2. Performance of MCA-OOja and MCA-OOjaH

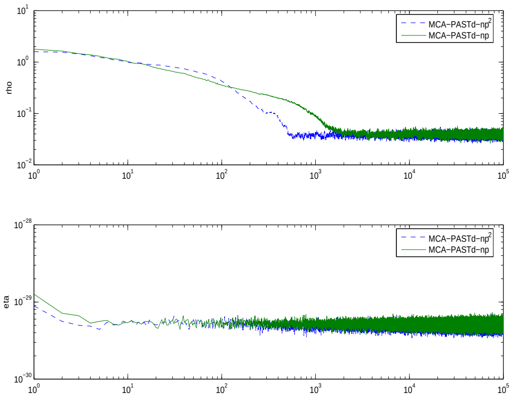

Fig. 3. Performance of MCA-PASTd(np-np^2)

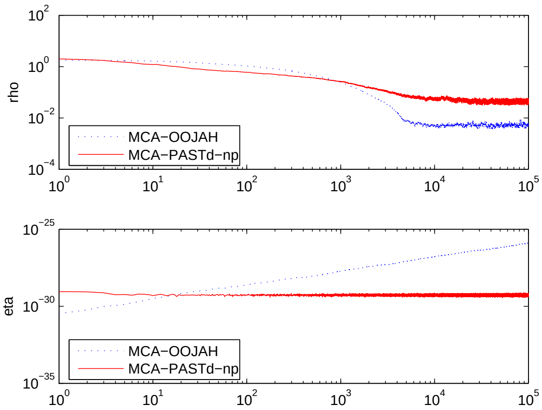

Fig. 4. Performance of MCA-PASTd-np and MCA-OOjaH

Fig.3 shows the effect of the modification of MC-PASTd. The reduction of computational complexity from $O(np^2)$ to $O(np)$ slows down slightly the convergence rate.
In Fig.4, we compare the performance of MC-OOjaH with MC-PASTd-np. As we can see, MC-OOjaH behaves better than MC-PASTd-np.

6. APPENDIX

6.1. Orthogonalization

To compute (2), we use the updating equation of $\mathbf{W}(i)$. Keeping in mind that $\mathbf{W}(i-1)$ is now an orthogonal matrix, we have (we omit the index i for simplicity)

$$\begin{aligned}\mathbf{N} &\triangleq \mathbf{W}^H(i)\mathbf{W}(i)\\ &= \mathbf{I}+\beta^2(\|\mathbf{x}\|^2\mathbf{z}\mathbf{z}^H+\|\mathbf{z}_p\|^2\mathbf{y}\mathbf{y}^H-(\mathbf{x}^H\mathbf{z}_p)\mathbf{z}\mathbf{y}^H\\ &\quad -(\mathbf{z}_p^H\mathbf{x})\mathbf{y}\mathbf{z}^H)\\ &= \mathbf{I}+\mathbf{R}\end{aligned}$$

$\mathbf{R}$ is a rank-2 Hermitian matrix and hence one can apply *Lemma 1* result to obtain its fast eigendecomposition and to compute the inverse square root of $\mathbf{N}$ according to the following steps:

$$\begin{aligned}\mathbf{P} &= [\mathbf{y}\ \mathbf{z}]\\ \mathbf{M} &= (\mathbf{P}^H\mathbf{P})^{-1}\mathbf{P}^H\mathbf{R}\mathbf{P}\\ \mathbf{M} &= \mathbf{T}diag(\lambda_1,\lambda_2)\mathbf{T}^{-1}\\ \mathbf{E} &= \mathbf{P}\mathbf{T}=[\mathbf{e}_1\ \mathbf{e}_2]\end{aligned}$$

As the eigenvectors $\mathbf{T}$ are computed up to scalar factors, one needs to normalize the columns of $\mathbf{E}$ to force it to be unitary, *i.e.*

$$\begin{aligned}\mathbf{\Sigma} &= diag(\frac{1}{\|\mathbf{e}_1\|},\frac{1}{\|\mathbf{e}_2\|})\\ \mathbf{E}' &= \mathbf{E}\mathbf{\Sigma}\ \text{and}\ \mathbf{T}'=\mathbf{T}\mathbf{\Sigma}\end{aligned}$$

Now, according to *Lemma 1*, we have

$$\mathbf{N}^{\frac{-1}{2}}=\mathbf{I}+\mathbf{E}'\mathbf{D}'\mathbf{E}'^H \tag{6}$$

where

$$\begin{aligned}\mathbf{D}' &= diag(\frac{1}{\sqrt{1+\lambda_1}}-1,\frac{1}{\sqrt{1+\lambda_2}}-1)\\ &= diag(\tau_1,\tau_2)\end{aligned} \tag{7}$$

By substituting (6) into (2) we obtain

$$\begin{aligned}\mathbf{W}(i) := \ &\mathbf{W}(i-1)+\tau_1\mathbf{W}(i-1)\mathbf{e}_1'-\beta\tau_1\mathbf{z}^H\mathbf{e}_1'\mathbf{x}+\\ &\beta\tau_1\mathbf{y}^H\mathbf{e}_1'\mathbf{W}(i-1)\mathbf{z})\mathbf{e}_1'^H+\tau_2\mathbf{W}(i-1)\mathbf{e}_2'\\ &-\beta\tau_2\mathbf{z}^H\mathbf{e}_2'\mathbf{x}+\beta\tau_2\mathbf{y}^H\mathbf{e}_2'\mathbf{W}(i-1)\mathbf{z}\mathbf{e}_2'^H\\ &-\beta\mathbf{x}\mathbf{z}^H+\beta\mathbf{W}(i-1)\mathbf{z}\mathbf{y}^H\end{aligned} \tag{8}$$

Using the orthonormality of $\mathbf{E}^{'}$ (i.e., $\mathbf{E}^{'H}\mathbf{E}^{'} = \mathbf{I}$) we obtain ($\mathbf{T}^{'-1} = \mathbf{E}^{'H}\mathbf{P}$). Hence, we can write

$$\begin{aligned}\mathbf{T}^{'-H} &= \left[\begin{array}{cc} t_{11} & t_{12} \\ t_{21} & t_{22} \end{array}\right]^{H} \\ &= \left[\begin{array}{cc} \mathbf{y}^{H}\mathbf{e}_1^{'} & \mathbf{y}^{H}\mathbf{e}_2^{'} \\ \mathbf{z}^{H}\mathbf{e}_1^{'} & \mathbf{z}^{H}\mathbf{e}_2^{'} \end{array}\right] \end{aligned} \tag{9}$$

Also, by developing $\mathbf{P} = \mathbf{E}^{'}\mathbf{T}^{'-1}$ we get

$$\mathbf{y} = t_{11}\mathbf{e}_1^{'} + t_{21}\mathbf{e}_2^{'} \tag{10}$$

$$\mathbf{z} = t_{12}\mathbf{e}_1^{'} + t_{22}\mathbf{e}_2^{'} \tag{11}$$

Finally, if we replace the results obtained in (9), (10) and (11) into (8) we obtain

$$\mathbf{W}(i) = \mathbf{W}(i-1) + \mathbf{p}\mathbf{e}_1^{'H} + \mathbf{q}\mathbf{e}_2^{'H} \tag{12}$$

Where

$$\begin{aligned}\mathbf{p} &= (\tau_1 + \beta(1+\tau_1)t_{11}t_{12})\mathbf{e}_{\mathbf{p_1}} + \beta(1+\tau_1)t_{11}t_{22}\mathbf{e}_{\mathbf{p_2}} \\ &\quad -\beta(1+\tau_1)t_{12}\mathbf{x} \\ \mathbf{q} &= (\tau_2 + \beta(1+\tau_2)t_{21}t_{22})\mathbf{e}_{\mathbf{p_2}} + \beta(1+\tau_2)t_{21}t_{12}\mathbf{e}_{\mathbf{p_1}} \\ &\quad -\beta(1+\tau_2)t_{22}\mathbf{x} \\ \mathbf{E}_{\mathbf{p}} &= \mathbf{W}\mathbf{E}^{'} \triangleq [\mathbf{e}_{\mathbf{p_1}}\ \mathbf{e}_{\mathbf{p_2}}]\end{aligned}$$

6.2. Householder

As stated by *Lemma 2*, $\mathbf{u}$ is calculated as the principal left singular eigenvector of

$$\mathbf{Q}^{'} = \mathbf{W}(i) - \mathbf{W}(i-1) \tag{13}$$

where $\mathbf{Q}^{'} = \mathbf{Q}\mathbf{E}^{H} = \mathbf{p}\mathbf{e}_1^{'H} + \mathbf{q}\mathbf{e}_2^{'H}$.
Equivalently, $\mathbf{u}$ can be seen as the principal eigenvector of the rank 2 Hermitian matrix

$$\mathbf{R} = \mathbf{Q}^{'}\mathbf{Q}^{'H}$$

This can be done by using *Lemma 1* results as follows:

$$\begin{aligned}\mathbf{Q} &= [\mathbf{p}\ \mathbf{q}] \\ \mathbf{R}_2 &= (\mathbf{Q}^{H}\mathbf{Q})^{-1}\mathbf{Q}^{H}\mathbf{R}\mathbf{Q} = (\mathbf{Q}^{H}\mathbf{Q})^{-1}\mathbf{Q}^{H}\mathbf{Q}_2\end{aligned}$$

where

$$\mathbf{Q}_2 = [\lambda_{11}\mathbf{p} + \lambda_{12}\mathbf{q} \quad \lambda_{21}\mathbf{p} + \lambda_{22}\mathbf{q}]$$

$$\Lambda = \left[\begin{array}{cc} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{11} \end{array}\right] =$$

$$\left[\begin{array}{cc} \|\mathbf{e}_1^{'}\|^2\|\mathbf{p}\|^2 + \mathbf{e}_1^{'H}\mathbf{e}_2^{'}\mathbf{q}^{H}\mathbf{p} & \mathbf{e}_2^{'H}\mathbf{e}_1^{'}\|\mathbf{p}\|^2 + \|\mathbf{e}_2^{'}\|^2\mathbf{q}^{H}\mathbf{p} \\ \|\mathbf{e}_1^{'}\|^2\mathbf{p}^{H}\mathbf{q} + \mathbf{e}_1^{'H}\mathbf{e}_2^{'}\|\mathbf{q}\|^2 & \mathbf{e}_2^{'H}\mathbf{e}_1^{'}\mathbf{p}^{H}\mathbf{q} + \|\mathbf{e}_2^{'}\|^2\|\mathbf{q}\|^2 \end{array}\right]$$

The principal eigenvector of $\mathbf{R}$ are given by $\mathbf{H} = \mathbf{Q}\mathbf{T}_1 = [\mathbf{h}_1\ \mathbf{h}_2]$, $\mathbf{T}_1$ being the eigenvector matrix of $\mathbf{R}_2$, *i.e.*

$$eig(\mathbf{R}_2) = \mathbf{T}_1 diag(\lambda_{q_1}, \lambda_{q_2})\mathbf{T}_1^{-1}.$$

Note that both principal eigenvectors of $\mathbf{R}$ can do the job, *i.e.* one can choose randomly $\mathbf{u} = \frac{\mathbf{h}_1}{\|\mathbf{h}_1\|}$ or $\mathbf{u} = \frac{\mathbf{h}_2}{\|\mathbf{h}_2\|}$.
Now, once $\mathbf{u}$ is computed, we can observe that

$$\mathbf{Z} = \mathbf{H}_1\mathbf{W}(i) - \mathbf{W}(i-1) = -2\mathbf{v}\mathbf{v}^{H}\mathbf{W}(i-1)$$

is a rank-1 matrix. All column vectors of the previous matrix are equal to $\mathbf{v}$ (up to scalar constant). Hence, it is sufficient to compute only its first column vector $\mathbf{Z}(:,1)$ and take $\mathbf{v}$ as its normalized version, *i.e.* $\mathbf{v} = \frac{\mathbf{Z}(:,\mathbf{1})}{\|\mathbf{Z}(:,\mathbf{1})\|}$.
This leads to the updating equation in Table 2.

7. REFERENCES

[1] P. van Dooren M. Moonen and J. vandewalle, "Updating singular value decompositions: A parallel implementations," *in Proc. SPIE Adv. Algorithms Archtectures Signal Processing*, pp. 80–91, 1989.

[2] K. Shmizu H. Sakai, "A new adaptative algorithm for minor component analysis," *Signal Processing 71*, pp. 301–308, 1998.

[3] S.-I Amari T. Chen and Q. Lin, "A unified algorithm for principal and minor components extraction," *Neural Networks*, vol. 11, pp. 385–390, Aug. 1998.

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

[5] H. Ogawa E.Oja and J. Wangviwattan, "Principal components analysis by homogeneous neural networks– parti: Weighted subspace criterion," *IEICE trans. Inform. Syst.*, vol. E75-D, no. 3, pp. 366–375, 1992.

[6] A. Marsal S. Marcos and M. Benidir, "The propagator method for source bearing estimation," *Signal Processing*, vol. 42, pp. 121–138, Aug. 1989.

[7] Y. Hua K. Abed-Meraim, A. Chkeif and S. Attallah, "On a class of orthonormal algorithms for principal and minor subspace tracking," *Journal of VLSI Signal Processing Systems (invited paper)*, 2001.

[8] X. Doukopoulos, "Power techniques for blind channel estimation in wirless communication systems," *Phd thesis*, pp. 125–130, October 2004.

[9] G.H. Golub and C.F. Van loan, "Matrix computations," *Baltimore, MD: Johns Hopkings Univ. Press*, pp. 205–213, 1996.