APPROXIMATE JOINT DIAGONALIZATION USING NON-ORTHOGONAL MATRICES

Arie Yeredor

Dept. of Elect. Eng. - Systems
Tel-Aviv University
Tel-Aviv, Israel

ABSTRACT

Several Blind Source Separation algorithms require the approximate joint diagonalization of a set of matrices. Most existing algorithms for approximate joint diagonalization are restricted to finding an *orthogonal* diagonalizing matrix. It has therefore become common practice to first find the non-orthogonal factor of the diagonalizing matrix by pre-whitening the data, and then proceed to find the orthogonal factor using existing algorithms. However, such an approach favors exact diagonalization of one matrix (usually the data correlation matrix), possibly at the expense of poor diagonalization of the other matrices. In this paper we propose a joint diagonalization algorithm which uses general (not necessarily orthogonal) matrices. The use of this algorithm enables to eliminate the traditional "hard whitening" phase, which is known to limit the performance, especially under noisy conditions. We demonstrate the improved performance via simulations results.

1. INTRODUCTION

The problem of approximate joint diagonalization is closely related to Blind Source Separation (BSS) and Independent Component Analysis (ICA). Several BSS algorithms, such as JADE [1], SOBI [2], Tong's [3], van der Veen's [4] and others (see also [5, 6, 7]), exploit the existence of some set $\mathcal{M}$ of K "true" matrices $\boldsymbol{M}_1, \boldsymbol{M}_2, \ldots \boldsymbol{M}_K \in \mathbb{C}^{N\times N}$ satisfying the following exact joint diagonalization property:

$$\boldsymbol{M}_k = \boldsymbol{A}\boldsymbol{\Lambda}_k\boldsymbol{A}^H \quad k = 1, 2, \ldots K \tag{1}$$

where the superscript ^{H} denotes the conjugate transpose, $\boldsymbol{A}$ denotes the unknown mixing matrix, and $\boldsymbol{\Lambda}_1, \boldsymbol{\Lambda}_2, \ldots \boldsymbol{\Lambda}_K$ denote some additional unknown diagonal matrices. Usually the set $\mathcal{M}$ is unavailable, but can be estimated (consistently) from the data. In that case, the estimated set $\hat{\mathcal{M}}$ of $\hat{\boldsymbol{M}}_1, \hat{\boldsymbol{M}}_2, \ldots \hat{\boldsymbol{M}}_K$ no longer admits exact joint diagonalization. However, it is then possible to find an estimate $\hat{\boldsymbol{A}}$ of $\boldsymbol{A}$, along with estimated diagonal matrices $\hat{\boldsymbol{\Lambda}}_1, \hat{\boldsymbol{\Lambda}}_2, \ldots \hat{\boldsymbol{\Lambda}}_K$, such that (1) is "as closely satisfied as possible" when the true matrices are replaced by their estimates. Consequently, $\hat{\boldsymbol{A}}$ can serve as the inferred estimate of the true diagonalizer of the "true" set (up to some irrelevant ambiguity issues, to be discussed later).

To quantify the desired fit, we may use the following weighted Least-Squares (LS) criterion:

$$C_{LS}(\hat{\boldsymbol{A}}, \hat{\boldsymbol{\Lambda}}_1, \hat{\boldsymbol{\Lambda}}_2, \ldots \hat{\boldsymbol{\Lambda}}_K) = \sum_{k=1}^{K} w_k \|\hat{\boldsymbol{M}}_k - \hat{\boldsymbol{A}}\hat{\boldsymbol{\Lambda}}_k\hat{\boldsymbol{A}}^H\|_F^2 \tag{2}$$

where $w_1, w_2, \ldots w_K \in \mathbb{R}$ are possible pre-specified positive weights, and where $\|\cdot\|_F^2$ denotes the squared Frobenius norm.

Most existing algorithms for approximate joint diagonalization seek an *orthogonal* diagonalizer $\hat{\boldsymbol{A}}$: Flury and Gautschi [8] propose a two-levels iterative algorithm termed *FG*; van der Veen and Paulraj [4] use a "super-generalized Schur Decomposition"; Cardoso [9] (see also Wax [6]) and Bunse-Gerstner et al. [10] use Jacobi-type rotations. The motivation for using an orthogonal diagonalizer is two-fold:

- The orthogonality constraint on $\hat{\boldsymbol{A}}$ simplifies the problem by commuting the transformations from $\hat{\boldsymbol{\Lambda}}_k$ to $\hat{\boldsymbol{M}}_k$ without changing the Frobenius norm; and
- The non-orthogonal factor of the general diagonalizer can often be approximated beforehand, using a pre-processing "whitening" phase (e.g. [5],[2]).

However, a whitening phase practically distorts the weighted LS criterion. It attains *exact* diagonalization of one of the matrices in $\hat{\mathcal{M}}$ (the matrix by which the whitening transformation is determined, usually

the empirical data correlation matrix), at the cost of poor diagonalization of the others. Such an operation is equivalent to setting to infinity the corresponding weight in (2). Since the weights reflect the desired "sharing" of mis-diagonalization between all the matrices in $\hat{\mathbf{M}}$, pre-whitening followed by orthogonal diagonalization may deviate significantly from the desired solution. In fact, in [11] it is shown that the "hard whitening" operation imposes a limit on the attainable performance, especially in the presence of additive noise.

In this paper we propose an iterative algorithm for minimizing C_{LS}, where the diagonalizer $\hat{\boldsymbol{A}} \in \mathbb{C}^{N\times N}$ is a general (not necessarily orthogonal) matrix. We assume that although $\hat{\mathcal{M}}$ is a set of estimated matrices (not necessarily satisfying (1)), all matrices in the set are Hermitian. Note that any minimizer of C_{LS} is not unique, since columns scaling and permutations in $\hat{\boldsymbol{A}}$ can be absorbed into the $\hat{\boldsymbol{\Lambda}}_k$-s. To circumvent these ambiguities, arbitrary scaling and ordering of diagonal values may be imposed; However, since these ambiguities are immaterial to the solutions of the related statistical problems, we refrain from explicitly imposing such artificial constraints.

2. MINIMIZATION ALGORITHM

Our algorithm is given the acronym $AC-DC$, as it alternates between the two following minimization schemes:

- The AC ("Alternating Columns") phase minimizes $C_{LS}(\hat{\boldsymbol{A}},\hat{\boldsymbol{\Lambda}}_1,\hat{\boldsymbol{\Lambda}}_2,\dots\hat{\boldsymbol{\Lambda}}_K)$ with respect to a selected column of $\hat{\boldsymbol{A}}$, while keeping its other columns, as well as $\hat{\boldsymbol{\Lambda}}_1,\hat{\boldsymbol{\Lambda}}_2,\dots\hat{\boldsymbol{\Lambda}}_K$, fixed.

- The DC ("Diagonal Centers") phase minimizes $C_{LS}(\hat{\boldsymbol{A}},\hat{\boldsymbol{\Lambda}}_1,\hat{\boldsymbol{\Lambda}}_2,\dots\hat{\boldsymbol{\Lambda}}_K)$ with respect to the diagonal matrices $\hat{\boldsymbol{\Lambda}}_1,\hat{\boldsymbol{\Lambda}}_2,\dots\hat{\boldsymbol{\Lambda}}_K$ while keeping $\hat{\boldsymbol{A}}$ fixed.

2.1. The AC phase

In this phase we minimize C_{LS} with respect to the l-th column of $\hat{\boldsymbol{A}}$ ($1\le l\le N$). C_{LS} can be expressed by

$$C_{LS}(\hat{\boldsymbol{A}},\hat{\boldsymbol{\Lambda}}_1,\hat{\boldsymbol{\Lambda}}_2,\dots\hat{\boldsymbol{\Lambda}}_K) = \sum_{k=1}^{K} w_k \|\hat{\boldsymbol{M}}_k - \hat{\boldsymbol{A}}\hat{\boldsymbol{\Lambda}}_k\hat{\boldsymbol{A}}^H\|_F^2 = \sum_{k=1}^{K} w_k \left\|\hat{\boldsymbol{M}}_k - \sum_{n=1}^{N}\hat{\lambda}_n^{[k]}\hat{\boldsymbol{a}}_n\hat{\boldsymbol{a}}_n^H\right\|_F^2 \quad (3)$$

where $\hat{\boldsymbol{a}}_n$ is the n-th column of $\hat{\boldsymbol{A}}=[\hat{\boldsymbol{a}}_1\hat{\boldsymbol{a}}_2\cdots\hat{\boldsymbol{a}}_N]$, and where $\hat{\lambda}_n^{[k]}$ is the (n,n)-th entry of $\hat{\boldsymbol{\Lambda}}_k$. Defining, with respect to a specific choice of l,

$$\tilde{\boldsymbol{M}}_k \triangleq \hat{\boldsymbol{M}}_k - \sum_{\substack{n=1\\ n\neq l}}^{N}\hat{\lambda}_n^{[k]}\hat{\boldsymbol{a}}_n\hat{\boldsymbol{a}}_n^H \quad (4)$$

we have (using the fact that all $\hat{\lambda}_n^{[k]}$ are real-valued, as evident in the DC phase below)

$$C_{LS}(\hat{\boldsymbol{A}},\hat{\boldsymbol{\Lambda}}_1,\hat{\boldsymbol{\Lambda}}_2,\dots\hat{\boldsymbol{\Lambda}}_K) = \sum_{k=1}^{K} w_k\|\tilde{\boldsymbol{M}}_k - \hat{\lambda}_l^{[k]}\hat{\boldsymbol{a}}_l\hat{\boldsymbol{a}}_l^H\|_F^2 =$$
$$\sum_{k=1}^{K} w_k\, Tr\left\{[\tilde{\boldsymbol{M}}_k - \hat{\lambda}_l^{[k]}\hat{\boldsymbol{a}}_l\hat{\boldsymbol{a}}_l^H]^H[\tilde{\boldsymbol{M}}_k - \hat{\lambda}_l^{[k]}\hat{\boldsymbol{a}}_l\hat{\boldsymbol{a}}_l^H]\right\} =$$
$$\tilde{C} - Tr\left\{\sum_{k=1}^{K} w_k\hat{\lambda}_l^{[k]}[\tilde{\boldsymbol{M}}_k^H\hat{\boldsymbol{a}}_l\hat{\boldsymbol{a}}_l^H + \hat{\boldsymbol{a}}_l\hat{\boldsymbol{a}}_l^H\tilde{\boldsymbol{M}}_k]\right\} +$$
$$Tr\left\{\sum_{k=1}^{K} w_k(\hat{\lambda}_l^{[k]})^2\hat{\boldsymbol{a}}_l\hat{\boldsymbol{a}}_l^H\hat{\boldsymbol{a}}_l\hat{\boldsymbol{a}}_l^H\right\} =$$
$$\tilde{C} - \hat{\boldsymbol{a}}_l^H\left[\sum_{k=1}^{K} w_k\hat{\lambda}_l^{[k]}[\tilde{\boldsymbol{M}}_k^H + \tilde{\boldsymbol{M}}_k]\right]\hat{\boldsymbol{a}}_l +$$
$$(\hat{\boldsymbol{a}}_l^H\hat{\boldsymbol{a}}_l)^2\sum_{k=1}^{K} w_k(\hat{\lambda}_l^{[k]})^2 \quad (5)$$

where $\tilde{C}$ is an independent constant, and where $Tr\{\cdot\}$ denotes the trace.

Decomposing $\hat{\boldsymbol{a}}_l$ into a real-valued scale a times a unit-norm vector $\boldsymbol{\alpha}$ ($\hat{\boldsymbol{a}}_l = a\boldsymbol{\alpha}$, such that $\boldsymbol{\alpha}^H\boldsymbol{\alpha}=1$), (5) can be reduced into

$$C_{LS}(a,\boldsymbol{\alpha}) = \tilde{C} - 2a^2\boldsymbol{\alpha}^H\boldsymbol{P}\boldsymbol{\alpha} + a^4 p \quad (6)$$

where $\boldsymbol{P}$ is the Hermitian matrix

$$\boldsymbol{P} \triangleq \sum_{k=1}^{K} w_k\hat{\lambda}_l^{[k]}\tilde{\boldsymbol{M}}_k \quad (7)$$

and

$$p \triangleq \sum_{k=1}^{K} w_k(\hat{\lambda}_l^{[k]})^2. \quad (8)$$

Differentiating with respect to a and equating zero yields the solutions $a=0$ or

$$a^2 = \boldsymbol{\alpha}^H\boldsymbol{P}\boldsymbol{\alpha}/p. \quad (9)$$

Since $\boldsymbol{P}$ is Hermitian, (9) is real-valued. Thus, if (9) is positive, then the minimizing a equals its square root,

otherwise it equals zero (because then $a = 0$ minimizes C_{LS} (6), since p is positive[1]). Consequently, if $\boldsymbol{P}$ is negative-definite then minimization of C_{LS} with respect to $\hat{\boldsymbol{a}}_l$ is attained by $\hat{\boldsymbol{a}}_l = \boldsymbol{0}$. However, normally this is not the case, and substituting a^2 back into (6) reduces the problem into minimization with respect to $\boldsymbol{\alpha}$ of

$$C_{LS}(\boldsymbol{\alpha}) = \tilde{C} - (\boldsymbol{\alpha}^H \boldsymbol{P} \boldsymbol{\alpha})^2 / p \tag{10}$$

subject to $\boldsymbol{\alpha}^H \boldsymbol{\alpha} = 1$. The desired solution is attained as the eigenvector of $\boldsymbol{P}$ associated with the largest (*positive*) eigenvalue. Note, however, that the constrained maximization is insensitive to multiplication of $\boldsymbol{\alpha}$ by any unit-magnitude (arbitrary phase) complex scalar.

Following is a summary of the AC phase:

AC phase:
Minimization of C_{LS} with respect to the l-th column of $\hat{\boldsymbol{A}}$:

Inputs:

- Target matrices $\hat{\boldsymbol{M}}_1, \hat{\boldsymbol{M}}_2, \dots \hat{\boldsymbol{M}}_K$;
- Diagonal matrices $\hat{\boldsymbol{\Lambda}}_1, \hat{\boldsymbol{\Lambda}}_2, \dots \hat{\boldsymbol{\Lambda}}_K$ (denoting the diagonal elements of $\hat{\boldsymbol{\Lambda}}_k$ as $\hat{\lambda}_1^{[k]}, \hat{\lambda}_2^{[k]}, \dots \hat{\lambda}_N^{[k]}$);
- Weights $w_1, w_2, \dots w_K$;
- Diagonalizing matrix $\hat{\boldsymbol{A}} = [\hat{\boldsymbol{a}}_1 \hat{\boldsymbol{a}}_2 \cdots \hat{\boldsymbol{a}}_N]$;
- Selected column index l.

Algorithm:

1. Calculate

$$\boldsymbol{P} = \sum_{k=1}^{K} w_k \hat{\lambda}_l^{[k]} \left[\hat{\boldsymbol{M}}_k - \sum_{\substack{n=1 \\ n \neq l}}^{N} \hat{\lambda}_n^{[k]} \hat{\boldsymbol{a}}_n \hat{\boldsymbol{a}}_n^H \right];$$

2. Find the largest eigenvalue μ and the associated unit-norm eigenvector $\boldsymbol{\alpha}$ of $\boldsymbol{P}$;
3. If $\mu < 0$, set $\hat{\boldsymbol{a}}_l = \boldsymbol{0}$, otherwise set

$$\hat{\boldsymbol{a}}_l = \frac{\boldsymbol{\alpha}\sqrt{\mu}}{\sqrt{\sum_{k=1}^{K} w_k (\hat{\lambda}_l^{[k]})^2}}.$$

---

[1] p can be zero only if all $\hat{\lambda}_l^{[k]}$ are zero, in which case C_{LS} is independent of $\hat{\boldsymbol{a}}_l$, so that any $\hat{\boldsymbol{a}}_l$ is a "minimizer".

If the largest eigenvalue occurs with multiplicity of more than one, then any of the associated eigenvectors (or any unit-norm linear combination thereof) yields a possible solution.

2.2. The DC phase

In this phase we minimize C_{LS} with respect to the diagonal matrices $\hat{\boldsymbol{\Lambda}}_1, \hat{\boldsymbol{\Lambda}}_2, \dots \hat{\boldsymbol{\Lambda}}_K$. Obviously, the minimization can be separated into K distinct minimization problems (for $k = 1, 2, \dots K$), each minimizing

$$C_k(\hat{\boldsymbol{\Lambda}}_k) \stackrel{\triangle}{=} \| \hat{\boldsymbol{M}}_k - \hat{\boldsymbol{A}} \hat{\boldsymbol{\Lambda}}_k \hat{\boldsymbol{A}}^H \|_F^2, \tag{11}$$

which is a linear LS problem in the parameters vector $\hat{\boldsymbol{\lambda}}_k = \mathrm{diag}\{\hat{\boldsymbol{\Lambda}}_k\}$, with $\hat{\boldsymbol{A}}$ fixed. To formulate as such, we define $\boldsymbol{m} = \mathrm{vec}\{\hat{\boldsymbol{M}}_k\}$ ($\mathrm{vec}\{\cdot\}$ denoting the matrix-to-vector conversion by concatenation of columns) and rewrite (11) as

$$C_k(\hat{\boldsymbol{\lambda}}_k) = [\boldsymbol{m} - \boldsymbol{H}\hat{\boldsymbol{\lambda}}_k]^H [\boldsymbol{m} - \boldsymbol{H}\hat{\boldsymbol{\lambda}}_k] \tag{12}$$

where it is straightforward to show that the matrix $\boldsymbol{H}$ is given by

$$\boldsymbol{H} = (\hat{\boldsymbol{A}}^* \otimes \boldsymbol{1}) \odot (\boldsymbol{1} \otimes \hat{\boldsymbol{A}}) \tag{13}$$

where $\boldsymbol{1}$ denotes an $N \times 1$ vector of 1-s, $\otimes$ denotes Kronecker's product, $\odot$ denotes Hadamard's (element-wise) product, and the superscript * denotes conjugation. The well-known minimizer of the linear LS problem is

$$\hat{\boldsymbol{\lambda}}_k = [\boldsymbol{H}^H \boldsymbol{H}]^{-1} \boldsymbol{H}^H \boldsymbol{m} \tag{14}$$

where it is again straightforward to show that

$$\boldsymbol{H}^H \boldsymbol{H} = (\hat{\boldsymbol{A}}^H \hat{\boldsymbol{A}})^* \odot (\hat{\boldsymbol{A}}^H \hat{\boldsymbol{A}}) \tag{15}$$

and

$$\boldsymbol{H}^H \boldsymbol{a} = \mathrm{diag}\{\hat{\boldsymbol{A}}^H \hat{\boldsymbol{M}}_k \hat{\boldsymbol{A}}\}. \tag{16}$$

If $\boldsymbol{H}^H \boldsymbol{H}$ is not invertible, then the LS minimizer is not unique. In that case any $\hat{\boldsymbol{\lambda}}_k$ that satisfies $\boldsymbol{H}^H \boldsymbol{H} \hat{\boldsymbol{\lambda}}_k = \boldsymbol{H}^H \boldsymbol{m}$ is a minimizing solution. To eliminate the associated ambiguity, the minimum-norm solution may be chosen.

Note that $\boldsymbol{H}^H \boldsymbol{H}$ is always real-valued, as it contains the element-wise squared magnitude of $\hat{\boldsymbol{A}}^H \hat{\boldsymbol{A}}$; In addition, $\mathrm{diag}\{\hat{\boldsymbol{A}}^H \hat{\boldsymbol{M}}_k \hat{\boldsymbol{A}}\}$ is also real-valued, since $\hat{\boldsymbol{M}}_k$ is Hermitian. Consequently, the resulting $\hat{\boldsymbol{\lambda}}_k$ is always real-valued. The DC phase is summarized below.

DC phase (Hermitian version):
Minimization of C_{LS} with respect to $\hat{\mathbf{\Lambda}}_1, \hat{\mathbf{\Lambda}}_2, \ldots \hat{\mathbf{\Lambda}}_K$:

Inputs:

- Target matrices $\hat{\boldsymbol{M}}_1, \hat{\boldsymbol{M}}_2, \ldots \hat{\boldsymbol{M}}_K$;
- Diagonalization matrix $\hat{\boldsymbol{A}}$;

Algorithm:

1. Prepare
   $\boldsymbol{G} = \left[(\hat{\boldsymbol{A}}^H \hat{\boldsymbol{A}})^* \odot (\hat{\boldsymbol{A}}^H \hat{\boldsymbol{A}}) \right]^{-1}$
2. For $k = 1, 2, \ldots K$,
   Set $\hat{\mathbf{\Lambda}}_k = \mathrm{diag}\{\boldsymbol{G}\,\mathrm{diag}\{\hat{\boldsymbol{A}}^H \hat{\boldsymbol{M}}_k \hat{\boldsymbol{A}}\}\}$.

3. INTERLACING THE *AC* AND *DC* PHASES, CONVERGENCE ISSUES

The "natural" objective of the $AC - DC$ algorithm is to alternate between minimization with respect to $\hat{\mathbf{\Lambda}}_1, \hat{\mathbf{\Lambda}}_2, \ldots \hat{\mathbf{\Lambda}}_K$ and minimization with respect to $\hat{\boldsymbol{A}}$. While the former is attained via a single DC run, the latter requires consecutive AC "sweeps" over all N columns of $\hat{\boldsymbol{A}}$ (i.e. for $l = 1, 2, \ldots N$, but not necessarily in that order). Thus, in principle, each DC phase is to be followed by an "infinite" number of AC sweeps before the next DC phase is run.

Alternatively, any fixed number of AC sweeps (or even incomplete sweeps) may interlace DC runs, not attaining "true" minimization with respect to $\hat{\boldsymbol{A}}$ between true minimizations with respect to $\hat{\mathbf{\Lambda}}_1, \hat{\mathbf{\Lambda}}_2, \ldots \hat{\mathbf{\Lambda}}_K$.

Nevertheless, with either strategy, C_{LS} is guaranteed not to increase (usually to decrease) with each AC/DC iteration, thus (being bounded below) convergence of C_{LS} is guaranteed. Assuming momentarily that in each AC iteration the largest eigenvalue is unique, and that in each DC iteration $\boldsymbol{H}^H \boldsymbol{H}$ is invertible, the solution of each AC/DC iteration is the unique global minimizer with respect to the respective parameters subset. Consequently, convergence of C_{LS} implies convergence of all the parameters. In turn, convergence of the parameters implies by construction that the derivatives of C_{LS} with respect to all the parameters vanish simultaneously, and hence a true stationary point is attained.

This claim is somewhat weakened by the fact that possibly not all the AC/DC iterations have unique solutions. However, since in such cases additional artificial constraints may be used, uniqueness can still be imposed, which still implies convergence of the parameters due to convergence of C_{LS}.

To attain convergence in practice, the magnitude of change either in all the parameters or in C_{LS} can be monitored and compared to a sufficiently small threshold for a stopping criterion.

A possible intelligent initialization would be to set $\hat{\boldsymbol{A}}$ to the (exact) joint diagonalizer of any two matrices in $\hat{\mathcal{M}}$, say $\hat{\boldsymbol{M}}_1$ and $\hat{\boldsymbol{M}}_2$, e.g. via diagonalization of $\hat{\boldsymbol{M}}_1 \hat{\boldsymbol{M}}_2^{-1}$ (assuming $\hat{\boldsymbol{M}}_2$ is invertible). The $AC - DC$ algorithm will then start with a DC phase. If an intelligent guess as to the anticipated diagonal values is available, then $\hat{\mathbf{\Lambda}}_1, \hat{\mathbf{\Lambda}}_2, \ldots \hat{\mathbf{\Lambda}}_K$ can be initialized accordingly, and the $AC - DC$ algorithm will start with an AC phase.

In figure 1 we demonstrate typical convergence patterns of C_{LS} for one and for two AC sweeps interlaced between each DC phase. The "true" set $\mathcal{M}$ of 10 real-valued 4×4 matrices was generated by drawing at random the elements of the true $\boldsymbol{A}$ (independent, Normal Standard random variables) and of the diagonals of the true $\mathbf{\Lambda}_k$ (independent, uniformly distributed in $(0, 1]$).

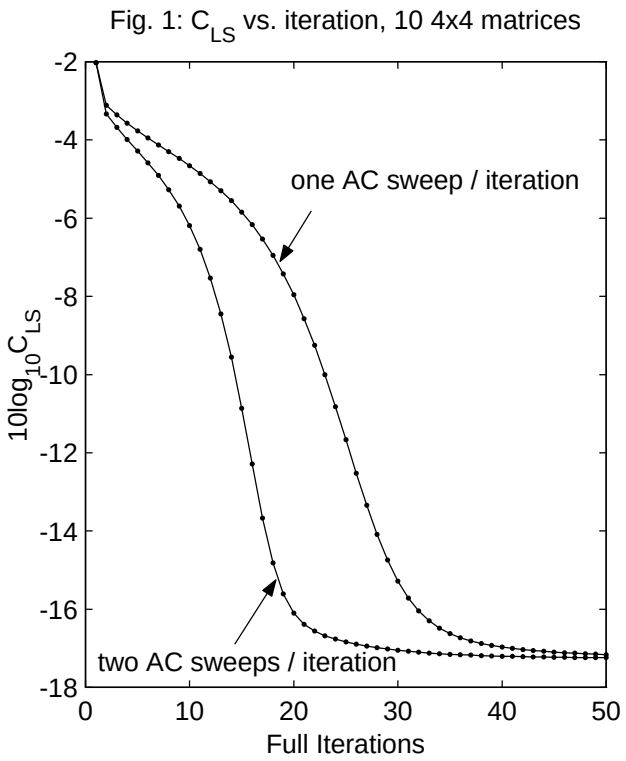

Figure 1: Typical convergence patterns of C_{LS} for randomly generated matrices with 1 and 2 full AC sweeps per iteration.

Then the "estimated" set $\hat{\mathcal{M}}$ was generated by noising (symmetrically) all elements of the matrices in $\mathcal{M}$ with independent Normal random variables with zero mean and standard deviation 0.01. The patterns show that with two AC sweeps in each iteration, the convergence rate is nearly two times faster. Since the associated computational load per iteration is nearly two times higher, there's no distinct preference for either strategy in this example.

4. SIMULATION RESULTS WITH JADE

To demonstrate the potential improvement in performance, we present in Figure 2 some simulations results by applying the JADE algorithm [1] to a 3×3 noisy BSS problem. We shall not go into detail in describing the JADE algorithm in here. We only mention that it is based on the joint diagonalization of empirical fourth cumulant matrices of the observed (mixed) data. The standard application of JADE requires a pre-whitening phase, in which the estimated data correlation matrix $\hat{\boldsymbol{R}}$ is calculated, and a whitening matrix $\hat{\boldsymbol{W}}$ is found, such that $\hat{\boldsymbol{W}}\hat{\boldsymbol{R}}\hat{\boldsymbol{W}}^H$ equals the Identity matrix. $\hat{\boldsymbol{W}}$ is in turn applied to the data (or directly to the estimated cumulants matrices), and then the orthogonal joint diagonalizer $\hat{\boldsymbol{U}}$ of the estimated cumulants matrices is found using successive Jacobi rotations [9]. The joint diagonalizer of the entire set (including the correlation matrix) is then given by $\hat{\boldsymbol{A}} = \hat{\boldsymbol{U}}\hat{\boldsymbol{W}}$.

As an alternative, we applied the AC-DC algorithm to the entire set. All the weights were set to unity. We used the output of the JADE algorithm as an initial guess.

The simulations setup was the following: The three source signals were generated as independent samples, uniformly distributed between $-\sqrt{3}$ and $\sqrt{3}$ (thus having zero mean and unit variance). The mixing matrix was

$$\boldsymbol{A} = \begin{bmatrix} 3 & -1 & 1 \\ 2 & 1 & 2 \\ 0 & -1 & 4 \end{bmatrix}.$$

The observed signals were contaminated by additive independent Normal Standard noise (hence a relatively low SNR of $0dB$ was used). T samples from each observed signals were used for estimating the correlation and cumulants matrices. The results are measured in terms of the average Interference to Signal Ratio (ISR) implied by the estimate of the mixing matrix. The ISR per trial is calculated as follows: first, the composite separation matrix $\hat{\boldsymbol{A}}^{-1}\boldsymbol{A}$ is calculated; then all its elements are squared; the ISR in each row is the sum of all (squared) elements in the row except for the largest, divided by the largest; the average ISR is the average of all the row ISRs. Note that in the low SNR scenario used here, the ISR is much lower than the Noise level, so in practice these results are only significant in as far as estimates of the true mixing matrix under noisy conditions are desired.

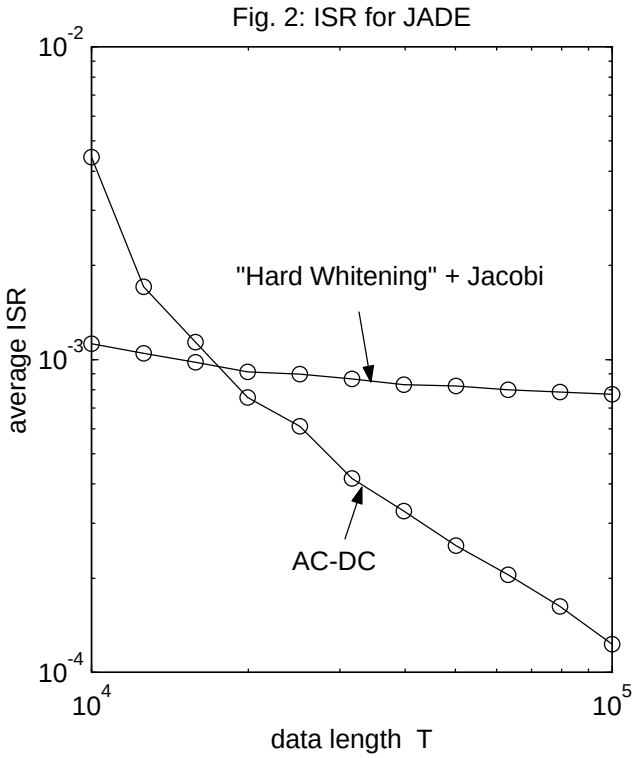

Figure 2: Average ISR for a 3×3 BSS problem at $0dB$ SNR vs. the observation length T, using the JADE algorithm. Simulations results are presented for conventional JADE, as well as for AC-DC with uniform weights. Each point reflects the average of 500 trials. Both algorithms were run with the same data.

In Figure 2 we demonstrate the resulting average ISR vs. the number of samples used T. As T increases, the advantage of the AC-DC algorithm over the conventional JADE algorithm becomes evident. Note that the superiority of AC-DC is, as expected, more pronounced as the correlation matrix becomes less accurate with respect to the cumulant matrices. Since the correlation matrix estimate is in general more accurate than the cumulants matrices estimates, the superior weighting attributed by the hard-whitening in conventional JADE is naturally justified. However, with additive Gaussian noise and with an increased number of samples, the estimated cumulant matrices remain unbiased (and their variance decreases), whereas the correlation matrix is biased. Under such circum-

stances, elimination of the "hard whitening" is justly warranted. The poor performance of AC-DC with the shorter T can be mitigated by proper choice of the weights $w_1, w_2, \ldots w_K$. These weights can be determined by considering statistics of the estimation errors, such as by the approach taken in [12].

5. CONCLUSION

We proposed an iterative algorithm termed $AC - DC$ for approximate joint diagonalization of a given set of matrices in the weighted LS sense. The algorithm can serve as a substitute to the conventional approach of "hard whitening" followed by orthogonal approximate joint diagonalization.

Performance improvement is attained when the estimate of the correlation matrix is of comparable accuracy to the estimates of the other matrices in the set. For example, with the JADE algorithm, this situation only occurs under low-SNR (with Gaussian noise) and long observation time conditions.

Finally, note that the proposed algorithm may be easily adapted to find the 'reduced' joint diagonalization with smaller diagonal matrices, $\hat{\mathbf{\Lambda}}_1, \hat{\mathbf{\Lambda}}_2, \ldots \hat{\mathbf{\Lambda}}_K \in \mathbb{C}^{M \times M}$ where $M < N$, so that the diagonalizing matrix $\hat{\boldsymbol{A}} \in \mathbb{C}^{N \times M}$ is not square. Such an application occurs, e.g., in BSS with more sensors than sources.

6. REFERENCES

[1] Jean-François Cardoso and Antoine Souloumiac, "Blind beamforming for non gaussian signals," *IEE - Proceedings -F*, vol. 140, no. 6, pp. 362–370, 1993.

[2] Adel Belouchrani, Karim Abed-Meraim, Jean-François Cardoso, and Eric Moulines, "A blind source separation technique using second-order statistics," *IEEE Trans. Signal Processing*, vol. 45, no. 2, pp. 434–444, 1997.

[3] Lang Tong, Yujiro Inouye, and Liu Ruei-wen, "Waveform-preserving blind estimation of multiple independent sources," *IEEE Trans. Signal Processing*, vol. 41, no. 7, pp. 2462–2470, 1993.

[4] Alle-Jan van der Veen and Arogyaswami Paulraj, "An analytical constant modulus algorithm," *IEEE Trans. Signal Processing*, vol. 44, no. 5, pp. 1136–1155, 1996.

[5] Jean-François Cardoso, "Source separation using higher order moments," *Proceedings ICASSP'89*, pp. 2109–2112, 1989.

[6] Mati Wax and Jacob Sheinvald, "A least-squares approach to joint diagonalization," *IEEE Signal Processing Letters*, vol. 4, no. 2, pp. 52–53, 1997.

[7] Arie Yeredor, "Blind source separation via the second characteristic function," To appear in Signal Processing.

[8] Bernhard N. Flury and Walter Gautschi, "An algorithm for simultaneous orthogonal transformation of several positive definite symmetric matrices to nearly diagonal form," *SIAM Journal on Sceintific and Statistical Computing*, vol. 7, no. 1, pp. 169–184, 1986.

[9] Jean-François Cardoso and Antoine Souloumiac, "Jacobi angles for simultaneous diagonalization," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 1, pp. 161–164, 1996.

[10] Angelika Bunse-Gerstner, Ralph Byers, and Volker Mehrmann, "Numerical methods for simultaneous diagonalization," *SIAM Journal on Matrix Analysis and Applications*, vol. 14, no. 4, pp. 927–949, 1993.

[11] Jean-François Cardoso, "On the performance of orthogonal source separation algorithms," *Proceedings of EUSIPCO'94*, pp. 776–779, 1994.

[12] Arie Yeredor, "Blind separation of gaussian sources via second-order statistics with asymptotically optimal weighting," Submitted to IEEE Signal Processing Letters.