

Approximate Joint Diagonalization Using a Natural Gradient Approach

Arie Yeredor¹, Andreas Ziehe², and Klaus-Robert Müller²

¹ School of Electrical Engineering, Tel-Aviv University, Israel
arie@eng.tau.ac.il

² Fraunhofer FIRSt, Germany
{ziehe,klaus}@first.fhg.de

Abstract. We present a new algorithm for non-unitary approximate joint diagonalization (AJD), based on a “natural gradient”-type multiplicative update of the diagonalizing matrix, complemented by step-size optimization at each iteration. The advantages of the new algorithm over existing non-unitary AJD algorithms are in the ability to accommodate non-positive-definite matrices (compared to Pham’s algorithm), in the low computational load per iteration (compared to Yeredor’s AC-DC algorithm), and in the theoretically guaranteed convergence to a true (possibly local) minimum (compared to Ziehe *et al.*’s FFdiag algorithm).

1 Introduction

The approximate joint diagonalization (AJD) of a set of matrices constitutes a fundamental stage in many batch-type algorithms for Independent Components Analysis (ICA) or Blind Source Separation (BSS). Usually, in this context, a set of unknown “target matrices” exists, which, assuming a linear static noiseless BSS model, admits exact joint diagonalization. The diagonalizing matrix (or the mixing matrix), can thus be theoretically extracted by jointly diagonalizing these matrices, which usually amounts to applying a generalized eigenvalue decomposition to any couple of matrices from the set. However, in practice the “target set” is unknown, and has to be estimated from the available data. In the presence of estimation errors, the estimated set usually no longer admits exact joint diagonalization. In such cases, one must resort to *approximate* joint diagonalization of the entire set in order to estimate the mixing matrix (or its inverse), as the matrix which diagonalizes the estimated set “as closely as possible”.

To formulate the problem, let $\check{\mathbf{M}}_1, \check{\mathbf{M}}_2, \dots, \check{\mathbf{M}}_K \in \mathbb{C}^{N \times N}$ denote the set of K true (usually unavailable) “target matrices” satisfying the exact joint diagonalization model

$$\check{\mathbf{M}}_k = \check{\mathbf{A}}\check{\mathbf{A}}_k\check{\mathbf{A}}^T \quad \text{or} \quad \check{\mathbf{A}}_k = \check{\mathbf{B}}\check{\mathbf{M}}_k\check{\mathbf{B}}^T, \quad k = 1, 2, \dots, K \quad (1)$$

where $\check{\mathbf{A}}$ is the true mixing matrix (assumed non-singular), $\check{\mathbf{B}}$ is its inverse (the true “demixing” matrix) and $\{\check{\mathbf{A}}_k\}_{k=1}^K$ is a set of diagonal matrices, usually associated with the sources’ statistical or structural properties, so that their diagonality dwells on the statistical independence of the sources. Some examples of such sets as used in BSS algorithms are:

- Cumulant matrices (in JADE, [1]);
- Correlation matrices of differently time-lagged or filtered signals (in SOBI [2] or OFI [3]);
- Joint time-frequency distributions at selected times and frequencies (in [4]);
- Hessians of the joint characteristic function (in CHESS, [5]);

and many more, extending also to the context of convolutive BSS, e.g., when working on separate frequency bins, such as in [6, 7].

As mentioned earlier, only estimates $\{\mathbf{M}_k\}_{k=1}^K$ (of $\{\check{\mathbf{M}}_k\}_{k=1}^K$) are available in practice, and the AJD problem consists of seeking the implied estimate \mathbf{A} of $\check{\mathbf{A}}$ (or \mathbf{B} of $\check{\mathbf{B}}$), along with “nuisance estimates” $\{\mathbf{A}_k\}_{k=1}^K$ of $\{\check{\mathbf{A}}_k\}_{k=1}^K$, such that the respective relation in (1) is most closely satisfied.

Thus, AJD is essentially a non-convex (possibly constrained) optimization problem, whose solution depends on the precise formulation of the target criterion (which has to reflect the proximity of the solution to the state of exact diagonalization). Numerous approaches have been proposed in recent years both for the formulation of the diagonalization criterion and for the iterative solution taken in its minimization:

- One of the most popular and computationally appealing approaches is the unitary AJD (Cardoso and Souloumiac, [8]), which minimizes the criterion

$$C_1(\mathbf{B}) = \sum_{k=1}^K \text{off}_1(\mathbf{B}\mathbf{M}_k\mathbf{B}^T) \quad (2)$$

with respect to (w.r.t.) \mathbf{B} , subject to the unitarity constraint $\mathbf{B}^T\mathbf{B} = \mathbf{I}$, where

$$\text{off}_1(\mathbf{P}) \triangleq \sum_{i \neq j} |P_{ij}|^2. \quad (3)$$

The unitarity constraint avoids the trivial minimizer $\mathbf{B} = \mathbf{0}$, but implies the assumption of a unitary mixing matrix. Hence, in the general case a pre-processing “spatial hard-whitening” stage is required, in which the non-unitary factor of the overall demixing matrix is found and applied to the data. In turn, this “hard whitening” stage implies exact joint diagonalization of the (spatial) correlation matrix, possibly at the expense of poor diagonalization of other matrices in the set. This implied unbalanced weighting has been observed [9] to limit the performance in the context of a general BSS problem.

- In order to avoid the unitarity constraint, an approach for non-unitary AJD has been proposed (the “AC-DC” algorithm, Yeredor [10]), which minimizes

$$C_2(\mathbf{A}) = \sum_{k=1}^K \|\mathbf{M}_k - \mathbf{A}\mathbf{A}_k\mathbf{A}^T\|_F^2 \quad (4)$$

(where $\|\bullet\|_F$ denotes the Frobenius norm) w.r.t. \mathbf{A} and $\{\mathbf{A}_k\}_{k=1}^K$, without constraining \mathbf{A} . While computationally efficient in small-scale problems, this algorithm has been observed [11] to exhibit extremely slow convergence in large-scale problems.

- A computationally efficient unconstrained minimization algorithm w.r.t. \mathbf{B} was proposed as well (Pham, [12]), whose target criterion is given by

$$C_3(\mathbf{B}) = \sum_{k=1}^K \text{off}_3(\mathbf{B}\mathbf{M}_k\mathbf{B}^T), \quad (5)$$

where in this case $\text{off}_3(\bullet)$ measures the Kullback-Leibler divergence between the $N \times N$ operand and the diagonal matrix with the same diagonal as the operand. This approach requires all the target matrices to be positive-definite, which poses a limit on its applicability as a generic BSS tool.

- A recently proposed approach (Ziehe et al., [11]) offers another computationally efficient algorithm, which avoids both the unitarity constraint and the positive-definiteness requirement. It aims at minimizing $C_1(\mathbf{B})$ with a different constraint on \mathbf{B} : Rather than impose unitarity, it inherently requires (by construction) that \mathbf{B} be representable as a product of matrices of the following form:

$$\mathbf{B} = \left[\prod_{m=1}^M (\mathbf{I} + \mathbf{W}^{(m)}) \right] \mathbf{B}^{(0)}, \quad (6)$$

where $\mathbf{B}^{(0)}$ is some initial guess, M is the number of iterations, and $\mathbf{W}^{(m)}$ are small “update matrices” with imposed zero diagonals, calculated along the iterations. Thus, if $\mathbf{B}^{(0)}$ is nonsingular and the norms of all $\mathbf{W}^{(m)}$ are maintained sufficiently small, it can be shown that the resulting \mathbf{B} must be invertible, hence the trivial minimizer is avoided. Moreover, this constraint does not limit the generality of the solution, since any two nonsingular matrices, say \mathbf{B}_1 and \mathbf{B}_2 , maintain the relationship $\mathbf{B}_2 = \mathbf{D}(\mathbf{I} + \mathbf{W})\mathbf{B}_1$, where \mathbf{D} is some nonsingular diagonal matrix and \mathbf{W} has a null diagonal. Thus, considering the inherent scale-ambiguity in BSS, the structural constraint (6) does not pose any practical restriction on the attainable solutions.

While computationally attractive, this algorithm has a few weak points from a theoretical point of view. It dwells on an approximation that may not always be valid in the presence of large errors in estimating the target matrices, and it involves some heuristics which are justified more on the practical-empirical side than on the theoretical side. Consequently, although its fast convergence has been verified empirically, it is not theoretically guaranteed to converge, and even upon convergence, \mathbf{B} is not always guaranteed to be a true (even local) minimizer of $C_1(\mathbf{B})$.

In this paper we propose a novel AJD algorithm, also aimed at the minimization of $C_1(\mathbf{B})$ subject to the same non-restrictive structural constraint (6) as in [11]. Similarly, our algorithm is computationally attractive, and does not require positive-definiteness of the set. Moreover, $C_1(\mathbf{B})$ is guaranteed to decrease in each iteration, so that its convergence is guaranteed. Also, since no approximations or heuristics are involved, upon convergence \mathbf{B} is guaranteed to be a true (possibly local) minimizer of $C_1(\mathbf{B})$.

The algorithm is based on the notion of a multiplicative “natural-gradient” (e.g., [13]), as opposed to the “standard” gradient (used, e.g., in [14]). The “natural gradient” is often applied in the context of “on-line” BSS algorithms, but also suits the AJD problem with the structural constraint (6). Our algorithm was named DOMUNG¹ (Diagonalization Of Matrices Using Natural Gradient).

2 Algorithm Derivation

Throughout the derivation we shall frequently use the operation of nullifying the diagonal of a matrix. We shall denote this operation by using an upper bar. More specifically, for any square matrix \mathbf{P} we define the notation $\overline{\mathbf{P}}$ as

$$\overline{\mathbf{P}} \triangleq \mathbf{P} - \tilde{\mathbf{P}} = \mathbf{P} - \mathbf{P} \odot \mathbf{I}. \quad (7)$$

The $\text{off}_1(\bullet)$ operator (3) can then be expressed based on the trace of a matrix:

$$\text{off}_1(\mathbf{P}) = \|\overline{\mathbf{P}}\|_F^2 = \text{tr}\{\overline{\mathbf{P}}^T \overline{\mathbf{P}}\} = \text{tr}\{\mathbf{P}^T \overline{\mathbf{P}}\}. \quad (8)$$

For simplicity of the derivations we shall assume that the target matrices are all real-valued and symmetric, which is often (but not always) the case in BSS applications. Extension to the more general case along similar guidelines is possible, but would extend beyond the scope of this limited-length paper.

We propose the following iterative process. Denote $\mathbf{B}^{(m)}$ the estimated diagonalizing (demixing) matrix after the m -th iteration, updated using $\mathbf{B}^{(m)} = (\mathbf{I} + \mathbf{W}^{(m)})\mathbf{B}^{(m-1)}$ $m = 1, 2, \dots$, where $\mathbf{B}^{(0)}$ is some initial guess and $\mathbf{W}^{(m)}$ is a matrix with a null main diagonal, which we shall eventually specify. Denoting

$$\mathbf{M}_k^{(m)} = \mathbf{B}^{(m-1)} \mathbf{M}_k \mathbf{B}^{(m-1)T} \quad k = 1, 2, \dots, K \quad m = 1, 2, \dots \quad (9)$$

as the “transformed” target set after the $(m-1)$ -th iteration, it is readily seen that at the m -th iteration the criterion function is given by

$$C_1(\mathbf{B}^{(m)}) = \sum_{k=1}^K \text{off}_1(\mathbf{B}^{(m)} \mathbf{M}_k \mathbf{B}^{(m)T}) = \sum_{k=1}^K \text{off}_1((\mathbf{I} + \mathbf{W}^{(m)}) \mathbf{M}_k^{(m)} (\mathbf{I} + \mathbf{W}^{(m)})^T). \quad (10)$$

We may therefore define, for each iteration m ,

$$C_1^{(m)}(\mathbf{W}) \triangleq \sum_{k=1}^K \text{off}_1((\mathbf{I} + \mathbf{W}) \mathbf{M}_k^{(m)} (\mathbf{I} + \mathbf{W})^T), \quad (11)$$

as a criterion function which we seek to minimize (w.r.t. \mathbf{W}) at that iteration, subject to the constraint on the structure of \mathbf{W} , namely that \mathbf{W} should have a null main diagonal. To this end, we now seek the gradient $\partial C_1^{(m)}(\mathbf{W}) / \partial \mathbf{W}$, which is a matrix whose (i, j) -th element is the derivative of $C_1^{(m)}(\mathbf{W})$ w.r.t.

¹ DOMUNG is a language spoken in Papua New Guinea.

W_{ij} (W_{ij} denoting the (i, j) -th element of \mathbf{W}). To find this gradient matrix, let us first find the gradient of each summand in (11). We do so by expressing the $\text{off}_1(\bullet)$ function in (11) in the vicinity of $\mathbf{W} = \mathbf{0}$ up to first-order terms in $\mathbf{W} = \mathcal{E}$, where \mathcal{E} is a sufficiently small matrix (for shorthand we shall use, in the following expressions, \mathbf{M} instead of $\mathbf{M}_k^{(m)}$):

$$\begin{aligned}
 \text{off}_1((\mathbf{I} + \mathcal{E})\mathbf{M}(\mathbf{I} + \mathcal{E})^T) &= \text{tr}\{[(\mathbf{I} + \mathcal{E})\mathbf{M}(\mathbf{I} + \mathcal{E})^T]^T \overline{(\mathbf{I} + \mathcal{E})\mathbf{M}(\mathbf{I} + \mathcal{E})^T}\} \\
 &= \text{tr}\{(\mathbf{I} + \mathcal{E})\mathbf{M}(\mathbf{I} + \mathcal{E})^T \overline{(\mathbf{I} + \mathcal{E})\mathbf{M}(\mathbf{I} + \mathcal{E})^T}\} \\
 &\approx \text{tr}\{(\mathbf{M} + \mathcal{E}\mathbf{M} + \mathbf{M}\mathcal{E}^T) \overline{(\mathbf{M} + \mathcal{E}\mathbf{M} + \mathbf{M}\mathcal{E}^T)}\} \\
 &\approx \text{tr}\{\mathbf{M}\overline{\mathbf{M}} + \mathbf{M}\overline{\mathcal{E}\mathbf{M}} + \overline{\mathbf{M}\mathbf{M}\mathcal{E}^T} + \mathcal{E}\overline{\mathbf{M}\mathbf{M}} + \mathbf{M}\overline{\mathcal{E}^T\mathbf{M}}\} \\
 &= \text{tr}\{\mathbf{M}\overline{\mathbf{M}} + \mathbf{M}\overline{\mathbf{M}}\mathcal{E} + \overline{\mathbf{M}\mathbf{M}}\mathcal{E} + \mathbf{M}\overline{\mathbf{M}}\mathcal{E} + \mathbf{M}\overline{\mathbf{M}}\mathcal{E}\} \\
 &= \text{tr}\{\mathbf{M}\overline{\mathbf{M}}\} + 2 \text{tr}\{(\mathbf{M}\overline{\mathbf{M}} + \overline{\mathbf{M}\mathbf{M}})\mathcal{E}\}. \quad (12)
 \end{aligned}$$

We used (8) in the first line, and the identities $\text{tr}\{\mathbf{P}\} = \text{tr}\{\mathbf{P}^T\}$, $\text{tr}\{\mathbf{P}\mathbf{Q}\} = \text{tr}\{\mathbf{Q}\mathbf{P}\}$ and $\text{tr}\{\mathbf{P}\overline{\mathbf{Q}}\} = \text{tr}\{\overline{\mathbf{P}}\mathbf{Q}\}$ in the transition from the fourth line to the fifth. The \approx symbol on the third and fourth lines indicates the elimination of terms of second or higher order in \mathcal{E} in the respective transitions.

Noting that $\partial \text{tr}\{\mathbf{P}\mathcal{E}\}/\partial \mathcal{E} = \mathbf{P}^T$, we obtain that the gradient of the $\text{off}_1(\bullet)$ function w.r.t. \mathbf{W} is $4(\overline{\mathbf{M}\mathbf{M}})$. Reinstating the full notation we obtain the gradient of $C_1^{(m)}$ w.r.t. \mathbf{W} at the m -th iteration:

$$\mathbf{G}^{(m)} \triangleq \frac{\partial C_1^{(m)}(\mathbf{W})}{\partial \mathbf{W}} = 4 \sum_{k=1}^K \overline{\mathbf{M}_k^{(m)}} \mathbf{M}_k^{(m)}. \quad (13)$$

Since we wish to decrease $C_1^{(m)}$ in each iteration, we shall apply a ‘‘steepest descent’’ step, by setting \mathbf{W} to $\mu \mathbf{D}^{(m)}$, where μ is some positive constant (whose optimal value will be discussed shortly), and $\mathbf{D}^{(m)} \triangleq -\overline{\mathbf{G}^{(m)}}$ is an ‘‘anti-gradient’’ matrix. The use of $\overline{\mathbf{G}^{(m)}}$ (rather than $\mathbf{G}^{(m)}$) as the gradient direction is due to the null-diagonal constraint on \mathbf{W} , which implies that its diagonal elements must remain zero, so that the only elements participating in the descent are the off-diagonal ones.

We now wish to ensure that the step-size in the anti-gradient direction yields the largest decrease in the criterion $C_1^{(m)}(\mathbf{W})$. Since this step-size is controlled by the parameter μ , we may now minimize $C_1^{(m)}(\mathbf{W}) = C_1^{(m)}(\mu \mathbf{D}^{(m)})$ w.r.t. μ . More specifically, substituting into (11) we obtain

$$\begin{aligned}
 C_1^{(m)}(\mu \mathbf{D}^{(m)}) &= \sum_{k=1}^K \text{off}_1((\mathbf{I} + \mu \mathbf{D}^{(m)})\mathbf{M}_k^{(m)}(\mathbf{I} + \mu \mathbf{D}^{(m)})^T) \\
 &= \sum_{k=1}^K \text{tr}\{(\mathbf{I} + \mu \mathbf{D}^{(m)})\mathbf{M}_k^{(m)}(\mathbf{I} + \mu \mathbf{D}^{(m)})^T \overline{(\mathbf{I} + \mu \mathbf{D}^{(m)})\mathbf{M}_k^{(m)}(\mathbf{I} + \mu \mathbf{D}^{(m)})^T}\} \\
 &\triangleq a_0^{(m)} + a_1^{(m)} \mu + a_2^{(m)} \mu^2 + a_3^{(m)} \mu^3 + a_4^{(m)} \mu^4 \quad (14)
 \end{aligned}$$

where the coefficients $\{a_l^{(m)}\}_{l=0}^2$ are given⁴ by $a_l^{(m)} = \sum_{k=1}^K \text{tr}\{\mathbf{F}_{l,k}^{(m)}\}$, with $\mathbf{F}_{l,k}^{(m)}$ summarized in Table 1:

Table 1.

$\mathbf{F}_{0,k}^{(m)}$	$\mathbf{M}_k^{(m)} \overline{\mathbf{M}_k^{(m)}}$
$\mathbf{F}_{1,k}^{(m)}$	$4\mathbf{M}_k^{(m)} \overline{\mathbf{M}_k^{(m)}} \mathbf{D}^{(m)}$
$\mathbf{F}_{2,k}^{(m)}$	$2 \left[(\mathbf{D}^{(m)} \mathbf{M}_k^{(m)} + \mathbf{M}_k^{(m)} \mathbf{D}^{(m)T}) \overline{\mathbf{D}^{(m)} \mathbf{M}_k^{(m)}} + \mathbf{D}^{(m)} \mathbf{M}_k^{(m)} \mathbf{D}^{(m)T} \overline{\mathbf{M}_k^{(m)}} \right]$
$\mathbf{F}_{3,k}^{(m)}$	$4\mathbf{D}^{(m)} \mathbf{M}_k^{(m)} \mathbf{D}^{(m)T} \overline{\mathbf{D}^{(m)} \mathbf{M}_k^{(m)}}$
$\mathbf{F}_{4,k}^{(m)}$	$4\mathbf{D}^{(m)} \mathbf{M}_k^{(m)} \mathbf{D}^{(m)T} \overline{\mathbf{D}^{(m)} \mathbf{M}_k^{(m)} \mathbf{D}^{(m)T}}$

Thus, since $C_1^{(m)}(\mu \mathbf{D}^{(m)})$ is evidently a fourth-order polynomial in μ , the optimal μ for the m -th iteration can be found by polynomial rooting of the derivative third-order polynomial, namely by solving (w.r.t. μ)

$$4a_4^{(m)} \mu^3 + 3a_3^{(m)} \mu^2 + 2a_2^{(m)} \mu + a_1^{(m)} = 0, \quad (15)$$

To which there is at least one real-valued solution. In the case of three real-valued solutions, the true minimum can be found by substituting each solution back into the polynomial (14) and selecting the solution that yields the smallest value. The algorithm is summarized below.

DOMUNG - Diagonalization Of Matrices Using Natural Gradient
– Denote the original “target set” as $\mathbf{M}_1^{(0)}, \mathbf{M}_2^{(0)}, \dots, \mathbf{M}_K^{(0)}$, and let $\mathbf{W}^{(0)} = \mathbf{0}$ and $\mathbf{B}^{(0)} = \mathbf{I}$.
– For $m = 1, 2, \dots$ until convergence
• Compute the updated target set $\mathbf{M}_k^{(m)} = (\mathbf{I} + \mathbf{W}^{(m-1)}) \mathbf{M}_k^{(m-1)} (\mathbf{I} + \mathbf{W}^{(m-1)})^T$ for $k = 1, 2, \dots, K$;
• Compute $\mathbf{G}^{(m)} = 4 \sum_{k=1}^K \overline{\mathbf{M}_k^{(m)}} \mathbf{M}_k^{(m)}$ and set $\mathbf{D}^{(m)} = -\overline{\mathbf{G}^{(m)}}$;
• Compute the coefficients $a_0^{(m)}, a_1^{(m)}, \dots, a_4^{(m)}$ using Table 1, and compute the real-valued root / three roots of the polynomial (15);
• Set μ to the root that yields the smallest value in (14);
• Set $\mathbf{W}^{(m)} = \mu \mathbf{D}^{(m)}$, $\mathbf{B}^{(m)} = (\mathbf{I} + \mathbf{W}^{(m)}) \mathbf{B}^{(m-1)}$.
– Upon convergence ($m = M$), the unmixing matrix is given by $\mathbf{B}^{(M)}$.

We did not specify a convergence criterion - but since the target criterion $C_1(\mathbf{B}^{(m)})$ is guaranteed to decrease (or at least not to increase) in each iteration, and it is bounded below, the sequence of its values over iterations must converge. Thus a stopping criterion that halts when the decrease in $C_1(\mathbf{B}^{(m)})$ falls below any (arbitrarily small) specified positive value, is guaranteed to be met.

² After using similar algebraic manipulations as in (12).

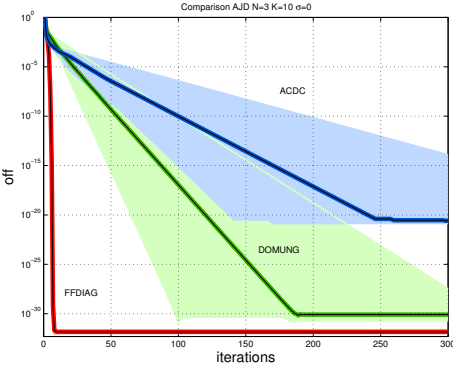


Fig. 1. Diagonalization errors on perfectly diagonalizable matrices

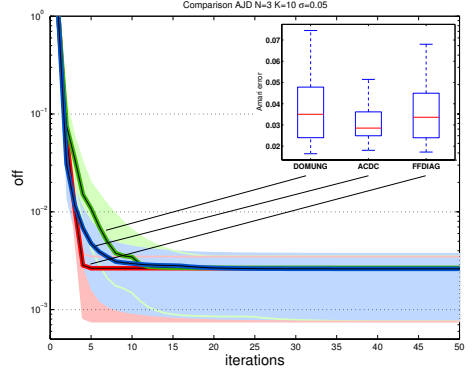


Fig. 2. Diagonalization errors and Amari errors [15] on non-diagonalizable matrices

3 Simulations

Here we provide a comparison of DOMUNG with two previously established algorithms: ACDC [10] and FFDIAG [11].

Noise free case The test data for the experiments is generated as follows. We use $K = 10$ diagonal matrices \mathbf{A}_k of size 3×3 where the elements on the diagonal are drawn from a uniform distribution in the range $[-1 \dots 1]$. These matrices are ‘mixed’ using the fixed matrix $\mathbf{A} = \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}$ according to the model $\mathbf{A}\mathbf{A}_k\mathbf{A}^T$ to obtain the set of matrices $\{\mathbf{M}_k\}$ to be diagonalized.

The convergence behavior of the 3 algorithms in 10 runs is shown in Fig. 1. The diagonalization error is measured by the $\text{off}_1(\cdot)$ function. The shaded area denotes the minima and maxima, while the bold line indicates the median over the 10 runs. In all cases the algorithms converged to the correct solution within the numerical computing precision. The differences in the final levels are only due to the use of slightly different stopping criteria.

Noisy case of non-diagonalizable matrices We also investigated robustness of the three algorithms against non-diagonalizability of the set of matrices.

Non-diagonalizability is modeled by adding random ‘noise’ matrices to the input matrices:

$$\mathbf{M}_k = \mathbf{A}\mathbf{A}_k\mathbf{A}^T + \sigma\mathbf{R}_k,$$

where \mathbf{R}_k are symmetric matrices, whose free elements are independently drawn from a standard normal distribution. The parameter σ determines the noise level, i.e. impact of the non-diagonalizable component.

Fig. 2 shows the error curves of 10 trials for a noise level of $\sigma = 0.05$, as well as distances from the true solution as measured by the Amari error [15] for 10 trials. One can see that all algorithms converge to the same level of the (normalized) cost function.

4 Conclusions

We proposed a new algorithm for simultaneous diagonalization of a set of symmetric matrices, where we combined: (i) a structural constraint to prevent the trivial solutions (ii) optimal (exact) line search procedure (iii) multiplicative updates based on natural gradient.

Extensions for further research would be to develop other “direction set methods”, e.g. conjugate gradient, using the new optimal line search procedure. Additionally, a scale-invariant target criterion would better reflect the BSS-related optimization requirement. Such a modification to the criterion, along with the implied adaptation of the algorithm, are also subject of our future research.

Acknowledgement AZ and KRM acknowledge partly funding by the EU PASCAL network (IST-2002506778).

References

1. Cardoso, J.F., Souloumiac, A.: Blind beamforming for non Gaussian signals. *IEE - Proceedings -F* **140** (1993) 362–370
2. Belouchrani, A., Abed-Meraim, K., Cardoso, J.F., Moulines, E.: A blind source separation technique using second-order statistics. *IEEE Trans. Signal Processing* **45** (1997) 434–444
3. Ziehe, A., Nolte, G., Curio, G., Müller, K.R.: OFI: Optimal filtering algorithms for source separation. In: *Proc. ICA2000, Helsinki, Finland (2000)* 127–132
4. Belouchrani, A., Amin, M.G.: Blind source separation based on time-frequency signal representations. *IEEE Trans. Signal Processing* **46** (1998) 2888–2897
5. Yeredor, A.: Blind source separation via the second characteristic function. *Signal Processing* **80** (2000) 897–902
6. Murata, N., Ikeda, S., Ziehe, A.: An approach to blind source separation based on temporal structure of speech signals. *Neurocomputing* **41** (2001) 1–24
7. Rahbar, K., Reilly, J.P., Manton, J.H.: Blind identification of MIMO FIR systems driven by quasistationary sources using second-order statistics: A frequency domain approach. *IEEE Trans. Signal Processing* **52** (2004) 406–417
8. Cardoso, J.F., Souloumiac, A.: Jacobi angles for simultaneous diagonalization. *SIAM Journal on Matrix Analysis and Applications* **17** (1996) 161–164
9. Cardoso, J.F.: On the performance of orthogonal source separation algorithms. *Proceedings of EUSIPCO’94 (1994)* 776–779
10. Yeredor, A.: Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation. *IEEE Trans. Signal Processing* **50** (2002) 1545–1553
11. Ziehe, A., Laskov, P., Müller, K.R., Nolte, G.: A linear least-squares algorithm for joint diagonalization. *Proceedings ICA2003 (2003)* 469–474
12. Pham, D.T.: Joint approximate diagonalization of positive definite matrices. *SIAM J. on Matrix Anal. and Appl.* **22** (2001) 1136–1152
13. Amari, S.I., Douglas, S.: Why natural gradient. *ICASSP’98* **2** (1998) 1213–1216
14. Joho, M., Mathis, H.: Joint diagonalization of correlation matrices by using gradient methods with application to blind signal separation. In: *Proc. of IEEE Sensor Array and Multichannel Signal Processing Workshop SAM. (2002)* 273–277
15. Amari, S., Cichocki, A., Yang, H.H.: A new learning algorithm for blind source separation. In *Touretzky, D.S., Mozer, M.C., Hasselmo, M.E., eds.: Advances in Neural Information Processing Systems. Volume 8. MIT Press (1996)* 757–763