A LINEAR LEAST-SQUARES ALGORITHM FOR JOINT DIAGONALIZATION

Andreas Ziehe[†], *Pavel Laskov*[†], *Klaus-Robert Müller*[†*], *Guido Nolte*[‡]

[†]Fraunhofer FIRST.IDA
Kekuléstr. 7
12489 Berlin, Germany

[*]University of Potsdam
Dept. of Computer Science
14482 Potsdam, Germany

[‡]National Institutes of Health
10 Center Drive MSC 1428
Bethesda, MD 20892, USA

ABSTRACT

We present a new approach to approximate joint diagonalization of a set of matrices. The main advantages of our method are computational efficiency and generality. We develop an iterative procedure, called LSDIAG, which is based on multiplicative updates and on linear least-squares optimization. The efficiency of our algorithm is achieved by the first-order approximation of the matrices being diagonalized. Numerical simulations demonstrate the usefulness of the method in general, and in particular, its capability to perform blind source separation without requiring the usual prewhitening of the data.

1. INTRODUCTION

Joint diagonalization of a set of matrices is an important optimization problem with various applications. In particular, this problem lies at the heart of many successful BSS algorithms, e.g. [1, 2, 3, 4, 5, 6, 7]. Although several diagonalization techniques have been proposed in the recent years [8, 9, 10, 11, 12], there is still the need for efficient and universal algorithms.

Mathematically, the joint diagonalization problem can be stated as follows: Given a set of matrices $\{C^k\}$ with $C \in I\!R^{N \times N}$ and $k = 1 \dots K$, find the matrix V such that the transformed matrices

$$F^k = VC^kV^T \tag{1}$$

are as diagonal as possible:

$$V = \operatorname{argmin} \sum_{k=1}^{K} \mathfrak{M}_D(F^k), \tag{2}$$

for a suitable diagonality measure $\mathfrak{M}_D$. In this paper we use the following measure:

$$\mathfrak{M}_D(F^k) = \sum_{i \neq j} (F^k_{ij})^2. \tag{3}$$

Alternative measures can be found e.g. in [10, 11].

The particular form of the matrices C and V depends on an application. For example, in the BSS problem, the matrices C^k result from the transformation of source correlation matrices by the mixing matrix A, e.g.

$$C^k = A\,E\{\mathbf{s}[t]\mathbf{s}[t+\tau_k]^T\}A^T.$$

In general, C^k can be arbitrary; however, it is common for BSS problems to consider symmetrized matrices C^k.

Some previous algorithms used additional assumptions, e.g. orthogonality of V [8], or positive-definiteness of C^k [11]. Such assumptions, however, might not hold in practice. In this contribution we propose a new algorithm, which is efficient and simple to implement, and at the same time does not make specific assumptions on the matrices. On the other hand, orthogonality can be utilized in our algorithm to further improve its performance.

Our approach is based on a perturbation expansion of the joint diagonalizer. We express the diagonality criterion (3) as a function of the perturbation. This function is still too complex to allow for an efficient optimization. To cope with this we linearize the transformation (1), which allows us to cast the optimization problem (2) as a *sparse linear least-squares problem.* A solution to the latter can be obtained in closed form with overall complexity[1] $\mathrm{O}(\mathcal{N})$ instead of $\mathrm{O}(\mathcal{N}^3)$. This solution is iteratively improved until it converges to the optimal diagonalizer of the given set of matrices.

A similar perturbation approach was considered in [13] with the assumption of orthogonality of the joint diagonalizer. In contrast, our Ansatz does not rely on orthogonality; however, it allows one to obtain the same result as in [13] if orthogonality is assumed. Another related algorithm, in which orthogonality is used for technical reasons, i.e. to circumvent trivial solutions, can be found in [12].

[1] Note that the size of the problem $\mathcal{N} = N^2 - N$ is inherently quadratic in the dimension N of the matrices.

2. GENERAL STRUCTURE OF THE ALGORITHM

The difficulty of direct optimization of the cost function (3) is two-fold:

1. The trivial solution $V = 0$ must be avoided.
2. The matrix V in question occurs up to degree 4 in the cost function.

To tackle these problems we propose an iterative scheme, which employs some well-motivated approximations and heuristics. In the remainder of this section we present the general outline of the algorithm and the motivation behind it, while computational details are deferred to section 3.

The main ingredient of our approach is the *multiplicative update*:

$$V_{(n+1)} = (I + W_{(n)})\, V_{(n)}. \tag{4}$$

The update matrix $W_{(n)}$ is constrained to have zeros on the main diagonal in order to prevent convergence of $V_{(n)}$ to the trivial solution[2]. The off-diagonal elements of $W_{(n)}$ are computed so as to minimize the joint diagonality criterion for the matrices

$$C^k_{(n+1)} = V_{(n+1)} C^k_{(0)} V^T_{(n+1)}.$$

Let $D^k_{(n)}$ and $E^k_{(n)}$ denote the diagonal and off-diagonal components of $C^k_{(n)}$ respectively. In order to make optimization tractable we assume that the norms of $W_{(n)}$ and $E^k_{(n)}$ are small. Then the quadratic terms in the expression for the new set of matrices can be ignored:

$$\begin{aligned} C^k_{(n+1)} &= (I + W_{(n)})(D^k_{(n)} + E^k_{(n)})(I + W_{(n)})^T \\ &\approx D^k_{(n)} + W_{(n)} D^k_{(n)} + D^k_{(n)} W^T_{(n)} + E^k_{(n)}. \end{aligned} \tag{5}$$

With these simplifications[3], and ignoring already diagonal terms D^k, the diagonality measure (3) can be computed over the expressions linear in W:

$$F^k \approx \tilde{F}^k = W D^k + D^k W^T + E^k. \tag{6}$$

At the first iterations the approximation of the matrices $C^k_{(n+1)}$ may be inaccurate. Nevertheless, it makes sense to carry out the multiplicative update (4) with the following sanity check: if $||W_{(n+1)}|| > ||W_{(n)}||$,

$$W_{(n+1)} \leftarrow \alpha \frac{||W_{(n)}||}{||W_{(n+1)}||} W_{(n+1)}, \tag{7}$$

with α a small number less than but close to 1, say 0.95. The rationale behind the condition (7) is the following. If $||W_{(n+1)}||$ increases in comparison with $||W_{(n)}||$, the quality of approximation of F^k in (6) goes down. Therefore one should not trust the new W too much. On the other hand, artificially forcing $||W_{(n+1)}||$ to be small by scaling it with some small constant would not serve the purpose either: if the approximation quality was bad at the n-th iteration, a really small update would not improve this quality by much. The most sensible solution appears to be to scale the new W such that $||W_{(n+1)}||$ is just slightly less than $||W_{(n)}||$. This is precisely what is achieved by the rule (7).

Our global updating procedure is summarized as:

Algorithm 1 The LSDIAG algorithm

```
$C^k_{(0)} \leftarrow C^k_x$ , $W_{(0)} \leftarrow 0$ , $n \leftarrow 0$
repeat
    compute $W_{(n+1)}$ according to eq. (11) or (12)
    if $(n > 0)$ AND $||W_{(n+1)}|| > ||W_{(n)}||$ then
        $W_{(n+1)} \leftarrow \alpha \frac{||W_{(n)}||}{||W_{(n+1)}||} W_{(n+1)}$
    end if
    $V_{(n+1)} \leftarrow (I + W_{(n)})\, V_{(n)}$
    $C^k_{(n+1)} \leftarrow V_{(n+1)} C^k_{(n)} V^T_{(n+1)}$
    $n \leftarrow n + 1$
until convergence
```

While the above pseudo-code outlines the general framework, the details of computing the update matrix W are presented in the next section.

3. COMPUTATION OF THE UPDATE MATRIX

The key to computational efficiency of the proposed algorithm lies in utilizing the sparseness introduced by the approximation (6). The special structure of the problem can be best seen in the matrix-vector notation presented next.

The $N(N-1)$ entries of the update matrix W are arranged as

$$w = (W_{12}, W_{21}, ..., W_{ij}, W_{ji}, ...)^T. \tag{8}$$

Notice that this is *not* the usual vectorization operation $\operatorname{vec} W$, as the order of elements in w reflects the pairwise relationship of the elements in W. In a similar way the $KN(N-1)$ entries of the off-diagonal matrices E^k are arranged as

$$e = (E^1_{12}, E^1_{21}, ..., E^1_{ij}, E^1_{ji}, ..., E^k_{ij}, E^k_{ji}, ...)^T. \tag{9}$$

[2] Clearly, all diagonal entries of the matrix $(I + W)^T(I + W)$ are ≥ 1. Since the trace of a positive-definite matrix is equal to the sum of its eigenvalues, it follows that the largest eigenvalue is ≥ 1. Therefore $||I + W||_2 \geq 1$.

[3] From now on the iteration indices will be dropped in the expressions if all quantities refer to the same iteration.

Finally, a large but sparse, $KN(N-1) \times N(N-1)$ matrix J is built, in the form:

$$J = \begin{pmatrix} J_1 \\ \vdots \\ J_K \end{pmatrix} \text{ with } J_k = \begin{pmatrix} \mathcal{D}_{12}^k & & & \\ & \ddots & & \\ & & \mathcal{D}_{ij}^k & \\ & & & \ddots \end{pmatrix}$$

where each J_k is block-diagonal, containing $N(N-1)/2$ matrices of dimension 2×2

$$\mathcal{D}_{ij}^k = \begin{pmatrix} D_j^k & D_i^k \\ D_j^k & D_i^k \end{pmatrix}.$$

Now the approximate cost function can be re-written as the familiar linear least-squares problem

$$\mathcal{L}(w) = \sum_k \sum_{i \neq j} (\tilde{F}_{ij}^k)^2 = (Jw + e)^T (Jw + e).$$

The well-known solution of this problem [14] reads

$$w = -(J^T J)^{-1} J^T e. \tag{10}$$

One can now reap the fruit of the problem's sparseness. Writing out the matrix product $J^T J$ yields a block-diagonal matrix

$$\begin{pmatrix} \sum_k (\mathcal{D}_{12}^k)^T \mathcal{D}_{12}^k & & & \\ & \ddots & & \\ & & \sum_k (\mathcal{D}_{ij}^k)^T \mathcal{D}_{ij}^k & \\ & & & \ddots \end{pmatrix}$$

whose blocks are 2×2 matrices. Thus the solution (10) consists of decoupled equations

$$\begin{pmatrix} W_{ij} \\ W_{ji} \end{pmatrix} = \begin{pmatrix} z_{jj} & z_{ij} \\ z_{ij} & z_{ii} \end{pmatrix}^{-1} \begin{pmatrix} y_{ij} \\ y_{ji} \end{pmatrix},$$

where

$$z_{ij} = \sum_k D_i^k D_j^k$$

$$y_{ij} = \sum_k D_j^k \frac{E_{ij}^k + E_{ji}^k}{2}.$$

The matrix inverse can be computed in closed form, leading to the following expressions for the update of the entries[4] of W:

$$\begin{aligned} W_{ij} &= \frac{z_{ji} y_{ji} - z_{ii} y_{ij}}{z_{jj} z_{ii} - z_{ij}^2} \\ W_{ji} &= \frac{z_{ij} y_{ij} - z_{jj} y_{ji}}{z_{jj} z_{ii} - z_{ij}^2}. \end{aligned} \tag{11}$$

An even simpler solution can be obtained if from the very beginning the diagonalization matrix V is assumed to be orthogonal. This assumption is frequently made in the literature on joint diagonalization [15, 8, 16]. Orthogonality of V implies that W is skew-symmetric, i.e., $W = -W^T$, and hence only one in each pair of its entries needs to be computed. In this case the structure of the problem is already apparent in the scalar notation, and one can easily obtain the partial derivatives of the cost function. Equating the latter to zero yields the following expression for the update of W:

$$W_{ij} = \frac{\sum_k E_{ji}^k (D_i^k - D_j^k)}{\sum_k (D_i^k - D_j^k)^2}, \tag{12}$$

which agrees with the result of Cardoso [13]. By using formula (12) the computational cost per iteration is reduced by half compared to (11).

4. NUMERICAL SIMULATIONS

4.1. Experiment 1

First, we demonstrate the performance of our joint diagonalization method on artificially generated test matrices. We use $K = 15$ diagonal matrices D^k of size 5×5, where the elements on the diagonal have been drawn from a uniform distribution in the range $[-1 \dots 1]$ (cf. [12]). These are 'mixed' by a random, orthogonal matrix A according to AD^kA^T to obtain the set of matrices $\{C^k\}$ to be diagonalized. This problem serves as a standard "sanity check" of most joint diagonalization algorithms, since such set of matrices is by construction perfectly diagonalizable.

We use the skew-symmetric update rule (12) and a separate re-normalization of C^k with their respective norms. The algorithm is initialized with the identity matrix $V_{(0)} = I$ and 20 iterations have been performed. To monitor the convergence of the algorithm we plot the diagonalization error per iteration for 10 independent trials as shown in Fig. 1. One can see that the algorithm has converged to the correct solution after at most 4 iterations in all trials. The actual running time was less than 0.5 seconds per trial.

4.2. Experiment 2

In a second example we investigate how our algorithm performs for a larger number of matrices and larger dimensions.

Similar to the first experiment, we generate a diagonalizable set of $K = 30$ matrices of dimension 25×25. The mixture is carried out by non-orthogonal random matrices A, which requires the use of the update rule (11). The results of this simulation are shown in Fig. 2.

[4] Note that only the off-diagonal elements ($i \neq j$) need to be computed and the diagonal terms of W are set to zero.

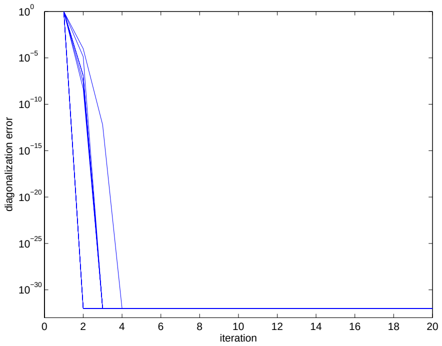

Figure 1: Diagonalization errors of the LSDIAG algorithm for a perfectly diagonalizable problem.

One can see that this problem is much more difficult, and in some trials (2 out of 10) the algorithm did not sufficiently converge after 200 iterations. These trials are indicated by dashed lines. However it can be seen that even in these hard cases the algorithm makes persistent improvements at each step. Furthermore the norm of W, as shown in the middle panel, monotonically decreases, which implies that the approximation quality steadily improves, and the algorithm will eventually converge. It is also interesting to observe (Fig. 2, lower panel) the operation of the heuristic (7) and the impact it has on the convergence of the algorithm. In these plots the value of the ratio $\frac{||W_{(n)}||}{||W_{(n+1)}||}$ is displayed. When this value is less than one, the heuristic (7) is used in the update; otherwise the shown value is 1 and the heuristic is not activated. One can see that in all trials this heuristic has been activated for some time intervals. Moreover, these intervals seem to be exactly the regions where the algorithm “takes a decisive turn” towards convergence. In the two most difficult trials, in which full convergence is not observed during the first 200 iterations, the heuristic can be seen to be still active. These observations underline the importance of the heuristic in the convergence behavior of the LS-DIAG algorithm.

4.3. Experiment 3

Finally, we apply our method to the blind source separation problem. For this, eight independent audio signals (shown in Fig. 4) have been mixed by an invertible non-orthogonal random 8×8 matrix A according to the model

$$\mathbf{x}[t] = A\,\mathbf{s}[t].$$

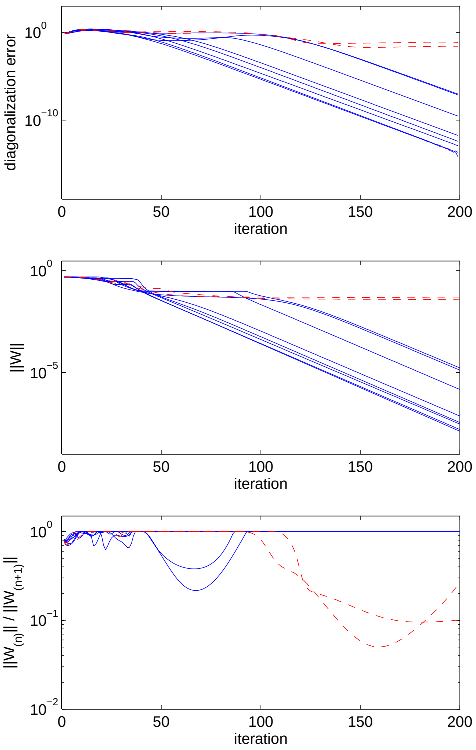

Figure 2: Diagonalization performance of the LSDIAG algorithm on a large ”non-orthogonal” problem.

The problem of identifying A can be solved by joint diagonalization if one is able to define certain matrices that are diagonal for the source signals $\mathbf{s}[t]$ and ”similar to diagonal” for the observed mixtures $\mathbf{x}[t]$. For example one could use correlation matrices of different segments of the data [5, 6], or matrices obtained from spatial time-frequency distributions [17, 4, 18] or slices of the cumulant tensor [19] and of course any combinations thereof [20]. In this experiment we compute 21 symmetrized, time-delayed correlation matrices (cf. [2, 3]) of the form

$$C^k = \frac{1}{2(T-1)} \sum_{t=1}^{T-\tau_k} \left(\mathbf{x}[t]\mathbf{x}[t+\tau_k]^T + \mathbf{x}[t+\tau_k]\mathbf{x}[t]^T \right)$$

with $\tau = [0 \ldots 20]$ and apply our LSDIAG algorithm with an initial $V_{(0)} = I$. Figure 3 shows the evolution of the diagonalization error over 50 iterations.

One can see in Fig. 3 that the diagonalization error is not as small as in the first experiments, but still reaches a sufficiently low level. Even if the error is not zero, because the C^k matrices are no longer perfectly diagonalizable, it indicates the good performance of our method.

This can also be verified, if we compare the joint diagonalizer $V_{(50)}$—which is supposed to be the estimated inverse of our mixing matrix A—with the true A by showing the entries of the product VA normalized by the maximal element in each column in Fig. 5.

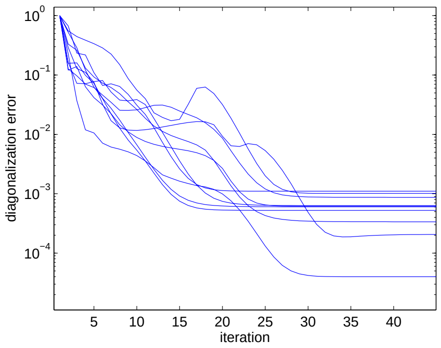

Figure 3: Diagonalization error per iteration in the BSS experiment.

5. CONCLUSIONS

In this paper we have presented a simple and efficient algorithm LSDIAG for joint diagonalization of a set of matrices. The algorithm is based on the least-squares optimization of the linearized diagonalization transformation. Although this transformation can only approximately represent the diagonalization error, it contains enough information to guide our iterative procedure towards convergence. If needed, an additional heuristic provides further impetus in this process (cf. eq. (7)).

The key to the algorithm's efficiency is a linear least-squares formulation which has a high degree of sparsity and allows for an analytic solution. As a result, each of the $N(N-1)$ parameter updates per iteration can be computed in closed form. To the best of our knowledge, such a favorable property was so far only enjoyed by Jacobi-type methods in the orthogonal case [8].

Moreover, when additional information is available on the structure of the matrices involved in joint diagonalization, some of this information can be used to further improve the performance of our algorithm. For instance, it can be easily adapted to the orthogonality of the diagonalizer.

It remains an open issue, if convergence of the LSDIAG algorithm, consistently observed in our experiments, can be formally proved. A comprehensive experimental comparison of the new algorithm with other existing joint diagonalization techniques is part of our ongoing work.

The new algorithm is particularly suited to BSS applications. First, it is not limited to orthogonal diagonalizers and hence does not require pre-whitening of the data. The benefit of this is that one can avoid errors introduced by pre-whitening that cannot be compensated in the later rotation step [21, 20]. The second attractive feature of the algorithm is the use of multiplicative updates [22, 23, 24], which ensure that the resulting diagonalizer always belongs to the group of invertible matrices. Finally, the algorithm scales well with both the dimensionality and the number of matrices to be diagonalized.

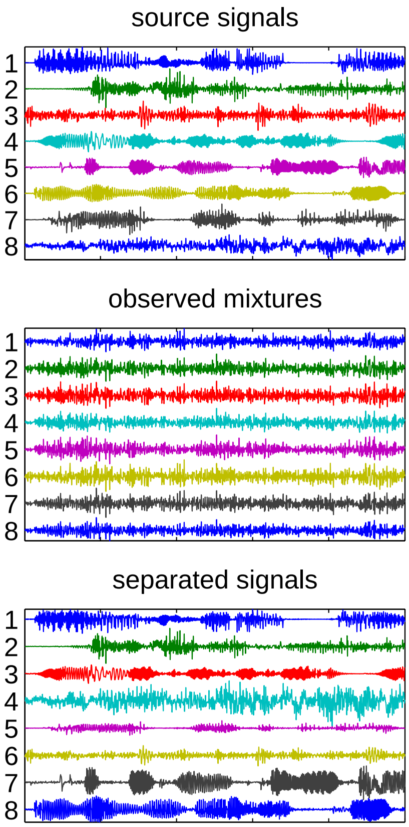

Figure 4: Waveforms of the eight audio signals.

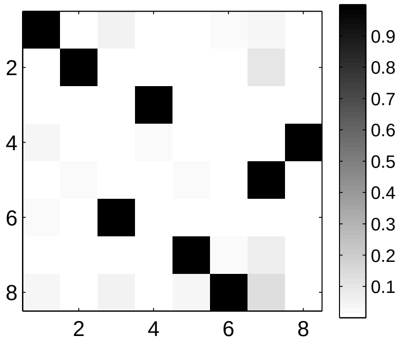

Figure 5: The normalized product of the estimated demixing matrix $V_{(50)}$ and the true mixing matrix A. The closeness of this product matrix to a permutation matrix indicates a good performance.

Acknowledgments The authors wish to thank Benjamin Blankertz, Guido Dornhege, Stefan Harmeling, Motoaki Kawanabe, Steven Lemm, Frank Meinecke and Sebastian Mika for fruitful discussions and helpful suggestions. A.Z. and K.-R.M. acknowledge partial funding in the EU project BLISS (IST-1999-14190) and the support from the BMBF under contract FKZ 01IBB02A. G.N. has been supported by a grant from the National Foundation for Functional Brain Imaging.

6. REFERENCES

[1] L. Molgedey and H. G. Schuster, "Separation of a mixture of independent signals using time delayed correlations," *Physical Review Letters*, vol. 72, no. 23, pp. 3634–3637, 1994.

[2] A. Belouchrani, K. Abed Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique based on second order statistics," *IEEE Trans. on Signal Processing*, vol. 45, no. 2, pp. 434–444, 1997.

[3] A. Ziehe and K.-R. Müller, "TDSEP – an efficient algorithm for blind separation using time structure," in *Proc. Int. Conf. on Artificial Neural Networks (ICANN'98)*, L. Niklasson, M. Bodén, and T. Ziemke, Eds., Skövde, Sweden, 1998, pp. 675 – 680, Springer Verlag.

[4] H.-C. Wu and J.C. Principe, "Simultaneous diagonalization in the frequency domain (SDIF) for source separation," in *Proc. Int. Workshop on Independent Component Analysis and Blind Source Separation (ICA'99)*, Aussois, France, January 11–15, 1999, pp. 245–250.

[5] D.-T. Pham and J.-F. Cardoso, "Blind separation of instantaneous mixtures of non-stationary sources," in *Proc. Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2000)*, Helsinki, Finland, 2000, pp. 187–193.

[6] S. Choi, A. Cichocki, and A. Belouchrani, "Blind separation of second-order nonstationary and temporally colored sources," in *Proc. IEEE Workshop on Statistical Signal Processing (IEEE SSP 2001)*, Singapore, 2001, pp. 444–447.

[7] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, John Wiley & Sons, New York, 2001.

[8] J.-F. Cardoso and A. Souloumiac, "Jacobi angles for simultaneous diagonalization," *SIAM J. Mat. Anal. Appl.*, vol. 17, no. 1, pp. 161–164, Jan. 1996.

[9] G. Hori, "Joint diagonalization and matrix differential equations," in *Proc. of NOLTA'99*. 1999, pp. 675–678, IEICE.

[10] A. Yeredor, "Blind separation of gaussian sources via second-order statistics with asymptotically optimal weighting," *IEEE Signal Processing Letters*, vol. 7, no. 7, pp. 197–200, 2000.

[11] D.T. Pham, "Joint approximate diagonalization of positive definite matrices," *SIAM J. on Matrix Anal. and Appl.*, vol. 22, no. 4, pp. 1136–1152, 2001.

[12] M. Joho and K. Rahbar, "Joint diagonalization of correlation matrices by using newton methods with application to blind signal separation," in *Proc. of IEEE Sensor Array and Multichannel Signal Processing Workshop SAM*, 2002.

[13] J.-F. Cardoso, "Perturbation of joint diagonalizers. ref# 94d027," Tech. Rep., Télécom Paris, 1994.

[14] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipies in C*, Cambridge University Press., Cambridge, 1992.

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

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

[17] D.-T. Pham and P. Garrat, "Blind separation of mixture of independent sources through a quasi-maximum likelihood approach," *IEEE Trans. on Signal Processing*, vol. 45, no. 7, pp. 1712–1725, 1997.

[18] A. Ziehe, G. Nolte, G. Curio, and K.-R. Müller, "OFI: Optimal filtering algorithms for source separation," in *Proc. Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2000)*, Helsinki, Finland, 2000, pp. 127–132.

[19] J.-F. Cardoso, "High-order contrasts for independent component analysis," *Neural Computation*, vol. 11, no. 1, pp. 157–192, Jan. 1999.

[20] K.-R. Müller, P. Philips, and A. Ziehe, "$JADE_{TD}$: Combining higher-order statistics and temporal information for blind source separation (with noise)," in *Proc. Int. Workshop on Independent Component Analysis and Blind Source Separation (ICA'99)*, Aussois, 1999, pp. 87–92.

[21] J.-F. Cardoso, "Asymptotic performance of second order blind separation," in *Proceedings ICASSP*, 1994, vol. 4, pp. 277–280.

[22] B. Laheld and J.-F. Cardoso, "Equivariant adaptive source separation," *IEEE Trans. on Signal Processing*, vol. 44, no. 12, pp. 3017–3030, 1996.

[23] T. Akuzawa and N. Murata, "Multiplicative nonholonomic newton -like algorithm," *Chaos, Solitons & Fractals*, vol. 12, pp. 785f, 2001.

[24] M. Zibulevsky, "Relative newton method for quasi-ml blind source separation," *Journal of Machine Learning Research*, to appear.