



HAL
open science

Approximate Joint Diagonalization According to the Natural Riemannian Distance

Florent Bouchard, Jérôme Malick, Marco Congedo

► **To cite this version:**

Florent Bouchard, Jérôme Malick, Marco Congedo. Approximate Joint Diagonalization According to the Natural Riemannian Distance. LVA/ICA 2017 - 13th International Conference on Latent Variable Analysis and Signal Separation, Feb 2017, Grenoble, France. pp.290-299, 10.1007/978-3-319-53547-0_28 . hal-01551813

HAL Id: hal-01551813

<https://hal.science/hal-01551813v1>

Submitted on 30 Jun 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Approximate Joint Diagonalization according to the Natural Riemannian Distance

Florent Bouchard¹ *, Jérôme Malick², and Marco Congedo¹

¹ GIPSA-lab, CNRS, Univ. Grenoble Alpes, Grenoble Institute of Technology, Grenoble France

² LJK, CNRS, Univ. Grenoble Alpes, Grenoble France

Abstract. In this paper, we propose for the first time an approximate joint diagonalization (AJD) method based on the natural Riemannian distance of Hermitian positive definite matrices. We turn the AJD problem into an optimization problem with a Riemannian criterion and we develop a framework to optimize it. The originality of this criterion arises from the diagonal form it targets. We compare the performance of our Riemannian criterion to the classical ones based on the Frobenius norm and the log-det divergence, on both simulated data and real electroencephalographic (EEG) signals. Simulated data show that the Riemannian criterion is more accurate and allows faster convergence in terms of iterations. It also performs well on real data, suggesting that this new approach may be useful in other practical applications.

Keywords: approximate joint diagonalization; Riemannian geometry; Hermitian positive definite matrices; Riemannian optimization

1 Introduction

The approximate joint diagonalization (AJD) of a matrix set is instrumental to solve the blind source separation (BSS) problem. We refer to [1] for a complete review on theory and applications. The AJD of a set $\{C_k\}_{1 \leq k \leq K}$ of K Hermitian positive definite (HPD) matrices of size $n \times n$ consists in finding a full rank matrix B of size $n \times n$ such that the set $\{BC_k B^H\}_{1 \leq k \leq K}$ is composed of matrices as diagonal as possible according to some criterion, where superscript \cdot^H denotes the conjugate transpose.

Criteria of interest are diagonality measures that turn the AJD problem into an optimization problem. The three most popular cost functions are the following ones. A widely used cost function based on the Frobenius distance is given by

$$f_{\mathbb{F}}(B) = \sum_{k=1}^K \|BC_k B^H - \text{ddiag}(BC_k B^H)\|_{\mathbb{F}}^2, \quad (1)$$

where $\|\cdot\|_{\mathbb{F}}$ denotes the Frobenius norm and the $\text{ddiag}(\cdot)$ operator returns the matrix with the diagonal elements of its argument. Some variations of (1) have

* Corresponding author, florent.bouchard@gipsa-lab.fr

been proposed in [2, 3] in order to have better invariance properties of the cost function. In [4], (1) is minimized through an indirect strategy. The idea is that, given B , a matrix A is sought in order to solve the optimization subproblem with cost function

$$\tilde{f}_F(A) = \sum_{k=1}^K \|BC_k B^H - A \text{ddiag}(BC_k B^H) A^H\|_F^2. \quad (2)$$

The joint diagonalizer B is then updated as $B \leftarrow A^{-1}B$. Another popular cost function based on the log-det divergence and introduced in [5, 6] is given by

$$f_{\text{LD}}(B) = \sum_{k=1}^K \log \frac{\det(\text{ddiag}(BC_k B^H))}{\det(BC_k B^H)}, \quad (3)$$

where $\det(\cdot)$ denotes the determinant of its argument. Recently, a generalization of (3) based on the log-det α -divergence has been proposed in [3] showing promising results.

In this paper, we propose for the first time an AJD method based on the natural Riemannian distance on the cone of HPD matrices [7, 8], which has recently attracted much interest in the signal processing and machine learning communities. This new approach exploits the geometrical properties of HPD matrices. Our solution is original compared to those obtained with previous criteria since the diagonal form it targets is profoundly different as we will show.

This paper is divided into four sections including this introduction. In section 2 the Riemannian cost function (section 2.3) is defined along with a framework to optimize it (sections 2.3 and 2.4). This new criterion stems from the Riemannian diagonality measure [3] (section 2.2) on the cone of HPD matrices [7, 8] (section 2.1). In section 3 we compare the performance of our Riemannian criterion to three state of the art ones on both simulated data and on a real electroencephalographic (EEG) recording. On simulated data (section 3.1) our Riemannian criterion proves more accurate and allows faster convergence in terms of iterations. It also performs well on the real EEG recording (section 3.2). Finally, in section 4, some conclusions are drawn.

2 Method: Riemannian Distance and Optimization

2.1 Cone of Hermitian Positive Definite Matrices

Let $\mathcal{M}_n(\mathbb{C})$ be the set of $n \times n$ complex matrices and $\mathcal{H}_n = \{C \in \mathcal{M}_n(\mathbb{C}) : C^H = C\}$ be the set of Hermitian matrices. The cone of HPD matrices \mathcal{H}_n^{++} is defined as the set $\{C \in \mathcal{H}_n : C \succ 0\}$. We give a succinct introduction of \mathcal{H}_n^{++} , the reader is referred to [7] for a complete presentation.

\mathcal{H}_n^{++} is an open subspace of \mathcal{H}_n , thus the tangent space $T_C \mathcal{H}_n^{++}$ at $C \in \mathcal{H}_n^{++}$ can be identified as \mathcal{H}_n . To turn \mathcal{H}_n^{++} into a Riemannian manifold, we need to endow every tangent space with a Riemannian metric, that is a smooth inner

product. At $C \in \mathcal{H}_n^{++}$, the *natural Riemannian metric* on $T_C \mathcal{H}_n^{++}$ is defined for all ξ and η in \mathcal{H}_n as [7]

$$\langle \xi, \eta \rangle_C = \text{tr}(C^{-1} \xi C^{-1} \eta), \quad (4)$$

where $\text{tr}(\cdot)$ denotes the trace operator. Note that this metric is positive definite and yields a norm on $T_C \mathcal{H}_n^{++}$ defined as $\|\xi\|_C = \langle \xi, \xi \rangle_C^{1/2}$. It leads to a *natural Riemannian distance* on \mathcal{H}_n^{++} defined for C_1 and C_2 as [7, 8]

$$d_{\mathcal{H}_n^{++}}(C_1, C_2) = \left\| \log(C_1^{-1/2} C_2 C_1^{-1/2}) \right\|_F, \quad (5)$$

where $\log(\cdot)$ denotes the matrix logarithm. The function $C_1 \mapsto d_{\mathcal{H}_n^{++}}(C_1, C_2)$ is convex. Having defined the Riemannian distance $d_{\mathcal{H}_n^{++}}$ on \mathcal{H}_n^{++} , we can define a proper Riemannian diagonality measure.

2.2 Riemannian Diagonality Measure

The subset \mathcal{D}_n^{++} of diagonal positive definite matrices is a closed submanifold of \mathcal{H}_n^{++} with respect to the Riemannian metric (4). The closest diagonal matrix A in \mathcal{D}_n^{++} to a matrix C in \mathcal{H}_n^{++} according to a distance or a divergence d is [3]

$$\underset{A \in \mathcal{D}_n^{++}}{\text{argmin}} \quad d(C, A). \quad (6)$$

The diagonality measure of C relative to d is therefore the distance or divergence to its closest diagonal matrix A . For the distance based on the Frobenius norm (corresponding to functional (1)) and for the log-det divergence (corresponding to (3)), the closest diagonal matrix to a matrix C is simply its diagonal part $\text{ddiag}(C)$ [3]. Using the Riemannian distance (5), the closest diagonal matrix A is the unique solution to equation [3]

$$\text{ddiag}(\log(C^{-1}A)) = 0. \quad (7)$$

To our knowledge, there is no closed form solution to this equation, however A can be numerically estimated by solving the optimization problem (6) with $d = d_{\mathcal{H}_n^{++}}$. This can be done as it follows: the directional derivative of $g : A \mapsto d_{\mathcal{H}_n^{++}}(C, A)$ in the direction $\xi \in \mathcal{D}_n$ (set of diagonal matrices) is given by

$$Dg(A)[\xi] = 2 \text{tr}(A^{-1} A \text{ddiag}(\log(C^{-1}A)) A^{-1} \xi). \quad (8)$$

This is a consequence of proposition 2.1 in [8], basic calculations, and of the fact that \mathcal{D}_n^{++} is a closed submanifold of \mathcal{H}_n^{++} . From (8), one can obtain the Riemannian gradient of g at A with the identification $\langle \text{grad } g(A), \xi \rangle_A = Dg(A)[\xi]$ [9]. This yields

$$\text{grad } g(A) = 2A \text{ddiag}(\log(C^{-1}A)). \quad (9)$$

Starting from an initial guess A_0 (for example $\text{ddiag}(C)$), one can obtain a sequence of iterates $\{A_i\}$. Given iterate A_i , the Riemannian gradient (9) of g at

A_i leads to a descent direction ξ_i in \mathcal{D}_n using for example a steepest-gradient or a conjugate gradient scheme (stepsize included in ξ_i if any). Finally, the exponential map of \mathcal{D}_n^{++} yields the iterate A_{i+1} as

$$A_{i+1} = A_i \exp(A_i^{-1} \xi_i). \quad (10)$$

2.3 AJD based on the Riemannian Distance of \mathcal{H}_n^{++}

Similarly to (1), the cost function based on the natural Riemannian distance (5) is defined by

$$\begin{aligned} f_{\text{R}}(B) &= \sum_{k=1}^K d_{\mathcal{H}_n^{++}}(BC_k B^H, A_k) \\ &= \sum_{k=1}^K \left\| \log((BC_k B^H)^{-1/2} A_k (BC_k B^H)^{-1/2}) \right\|_F^2, \end{aligned} \quad (11)$$

where A_k is the closest diagonal matrix to $BC_k B^H$ as per section 2.2. We can minimize (11) by taking an approach similar to the one introduced in [4]. Given the sets $\{BC_k B^H\}$ and $\{A_k\}$, the idea is to find a matrix A such that the set $\{AA_k A^H\}$ gets closer (according to (5)) to the set $\{BC_k B^H\}$. This way, matrices $A^{-1}BC_k B^H A^{-H}$ are closer to diagonal form. Further, note that when the best possible matrix A is the identity matrix I_n , $\{BC_k B^H\}$ contains matrices as diagonal as possible according to (5). To find A , we define the optimization subproblem with cost function \tilde{f}_{R} as

$$\tilde{f}_{\text{R}}(A) = \sum_{k=1}^K \left\| \log((BC_k B^H)^{-1/2} A A_k A^H (BC_k B^H)^{-1/2}) \right\|_F^2. \quad (12)$$

This function has a simpler expression and we can minimize it with a Riemannian gradient based method that we describe in section 2.4. We then update the matrix B according to

$$B \leftarrow A^{-1} B. \quad (13)$$

This leads to Algorithm 1.

Algorithm 1: RD-AJD (Riemannian Distance AJD)

Input: set of K matrices $\{C_k\}$ in \mathcal{H}_n^{++} , initial guess B_0 for B , maximum number of iteration i_{max} .

Output: estimated joint diagonalizer B .

- 1 Initialization: set $i = 0$ and compute matrices $\{B_0 C_k B_0^H\}$.
 - 2 **while** *not convergence* **and** $i < i_{\text{max}}$ **do**
 - 3 Find diagonal matrices $\{A_{k,i}\}$ as described in section 2.2.
 - 4 Set $A_0 = I_n$ and find A_i by performing one step in a descent direction of (12) (see section 2.4).
 - 5 $B_{i+1} \leftarrow A_i^{-1} B_i$.
 - 6 $i = i + 1$.
-

2.4 Riemannian Optimization for the Subproblem

It remains to define a method to optimize the cost function \tilde{f}_R defined in (12). In order to do so, we perform a Riemannian optimization on the *special polar manifold* [10] defined as

$$\mathcal{SP}_n = \{(U, P) \in \mathcal{H}_n^{++} \times \mathcal{O}_n : \det(P) = 1\}. \quad (14)$$

This manifold is a consequence of the following observation: every full rank matrix A admits the *unique polar decomposition* $A = UP$, where $U \in \mathcal{O}_n$ (group of orthogonal matrices) and $P \in \mathcal{H}_n^{++}$, thus the manifold of full rank matrices is equivalent to the product manifold $\mathcal{H}_n^{++} \times \mathcal{O}_n$. To avoid degenerated solutions, we impose $\det(AA^H) = 1$, which is equivalent to $\det(P) = 1$.

In the following, notations \mathcal{A} is used to denote the point (U, P) in \mathcal{SP}_n and A to denote its corresponding full rank matrix UP . Given the initialization $\mathcal{A}_0 = (I_n, I_n)$ (corresponding to $A_0 = I_n$), one can obtain a descent direction ξ_i from the Riemannian gradient of (12) by a steepest-descent or conjugate gradient algorithm (stepsize included in ξ_i if any). The iterate \mathcal{A}_i (corresponding to the matrix A_i) is then obtained through the retraction R of \mathcal{SP}_n as

$$\mathcal{A}_i = R_{\mathcal{A}_0}(\xi_i). \quad (15)$$

From proposition 2.1 in [8] and basic calculations, the directional derivative of \tilde{f}_R at \mathcal{A} in the direction \mathcal{Z} is

$$D\tilde{f}_R(\mathcal{A})[\mathcal{Z}] = 4 \sum_{k=1}^K \text{tr}(\log((BC_k B^H)^{-1} A \Lambda_k A^H) U Z_U^H) + \text{tr}(P^{-1} \log(A^H (BC_k B^H)^{-1} A \Lambda_k) Z_P^H). \quad (16)$$

Thus, the Euclidean gradient of \tilde{f}_R on the ambient space of \mathcal{SP}_n is

$$\text{grad}_{\mathcal{E}} \tilde{f}_R(\mathcal{A}) = 4 \sum_{k=1}^K \begin{pmatrix} \log((BC_k B^H)^{-1} A \Lambda_k A^H) U, \\ P^{-1} \log(A^H (BC_k B^H)^{-1} A \Lambda_k) \end{pmatrix} \quad (17)$$

and the Riemannian gradient of (12) is

$$\text{grad}_{\tilde{f}_R}(U, P) = (\Pi_U(\text{grad}_{\mathcal{E}} \tilde{f}_R(U)), \Pi_P(P \text{Herm}(\text{grad}_{\mathcal{E}} \tilde{f}_R(P))P)), \quad (18)$$

where $\text{grad}_{\mathcal{E}} \tilde{f}_R(U)$ and $\text{grad}_{\mathcal{E}} \tilde{f}_R(P)$ denote the first and second component of $\text{grad}_{\mathcal{E}} \tilde{f}_R(\mathcal{A})$, respectively. $\Pi_{\mathcal{A}} = (\Pi_U, \Pi_P)$ is the projection map from the ambient space onto the tangent space $T_{\mathcal{A}}\mathcal{SP}_n$ at \mathcal{A} . It is given, for $\mathcal{Z} = (Z_U, Z_P) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times n}$, by

$$\Pi_{\mathcal{A}}(\mathcal{Z}) = (Z_U - U \text{Herm}(U^H Z_U), \text{Herm}(Z_P) - \frac{1}{n} \text{tr}(P^{-1} \text{Herm}(Z_P))P), \quad (19)$$

where $\text{Herm}(\cdot)$ returns the Hermitian part of its argument. Finally, the retraction $R_{\mathcal{A}}$ mapping a tangent vector back onto the manifold is, for $\xi_{\mathcal{A}} = (\xi_U, \xi_P)$,

$$R_{\mathcal{A}}(\xi_{\mathcal{A}}) = \left(\text{uf}(U + \xi_U), P^{1/2} \exp(P^{-1/2} \xi_P P^{-1/2}) P^{1/2} \right) \quad (20)$$

where $\text{uf}(\cdot)$ extracts the orthogonal factor of its argument by Lödwin's orthogonalization [11].

3 Numerical Experiments

We now have all the ingredients to perform RD-AJD. We estimate its performance on simulated data and we compare our approach with those using cost functions (1), (2) and (3). For RD-AJD, we compute the closest diagonal matrices using a Riemannian conjugate gradient on \mathcal{D}_n^{++} . Classical algorithms minimizing those cost functions (1), (2) and (3) generally use specific optimization schemes and constraints [4, 6]. In order to compare the performance of the criteria, we perform all optimization on \mathcal{SP}_n with the same Riemannian conjugate gradient method. This way, differences in performance are not due to the optimization scheme but solely to the criterion employed. This Riemannian optimization scheme has been shown to perform well in general, see for example [10].

We refer to the resulting algorithms by acronyms FD-AJD (Frobenius distance) for (1), mFD-AJD (modified Frobenius distance) for (2) and LD-AJD (log-det) for (3). We initialize all algorithms with the inverse of the square root of the arithmetic mean of the target matrices. For all of them, the stopping criterion for iterate i is defined as $\|B_{i-1}B_i^{-1} - I_n\|_F^2/n$ and is set to 10^{-6} . Note that when comparing the performance of different algorithms, it is very important to use the same stopping criterion. The Riemannian conjugate gradient method is performed using manopt toolbox [12].

3.1 Simulated Data

We simulate sets of K real valued $n \times n$ matrices $\{C_k\}$ according to the model [13]

$$C_k = AA_kA^T + \frac{1}{\sigma}E_kA_k^N E_k^T + \alpha I_n, \quad (21)$$

where matrices A and E_k are random matrices with i.i.d. elements drawn from the normal distribution, σ is a free parameter defining the expected signal to noise ratio, $\alpha = 10^{-3}$ is a free parameter representing uncorrelated noise and matrices A_k and A_k^N are diagonal matrices with i.i.d. elements respectively corresponding to signal matrices and noise. The p^{th} element $\lambda_{p,k}$ is drawn from a chi-squared distribution with expectation $n/p^{1.5}$.

To estimate how the methods behave, we use two criteria. The first one is the Moreau-Amari index $I_{\text{M-A}}$ [14], which is a measure of accuracy, *i.e.*, of how close to the true one is the estimated solution. It is defined as

$$I_{\text{M-A}}(M) = \frac{1}{2n(n-1)} \sum_{p=1}^n \left[\frac{\sum_{q=1}^n |M_{pq}|}{\max_{1 \leq q \leq n} |M_{pq}|} + \frac{\sum_{q=1}^n |M_{qp}|}{\max_{1 \leq q \leq n} |M_{qp}|} \right] - \frac{1}{(n-1)}, \quad (22)$$

where $M = BA$, with B the estimated joint diagonalizer and A the true mixing matrix of the signal part in (21). The second criterion I_C concerns the convergence speed of the algorithms. It measures the distance between the iterate B_i

and the final estimated joint diagonalizer B . It is defined as [13]

$$I_C(B_i) = \frac{\|B_i - B\|_F^2}{\|B\|_F^2}. \quad (23)$$

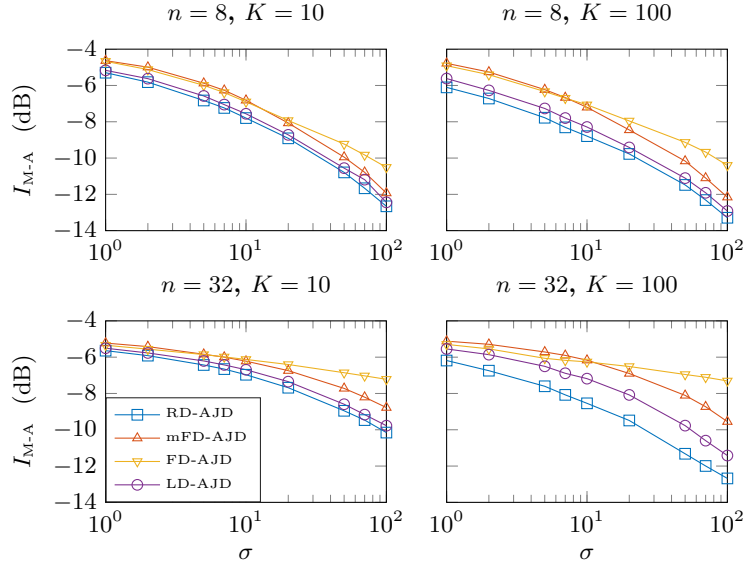


Fig. 1: mean of the performance of the considered algorithms as a function of the noise parameter σ over 50 trials for different values of n (input matrix dimension) and K (number of matrices). RD-AJD outperforms the other approaches in every cases. The difference with the other methods increases with n and/or K . See text for details.

First, we analyze the quality of the results as a function of the noise parameter σ . One can see in figure 1 that RD-AJD outperforms the other methods in all cases. As expected, the performance of the algorithms increases with K and decreases with n . The difference between RD-AJD and the others increases with K and n . This shows that this criterion is more robust with respect to these parameters. This property may be important in practical applications where the size and number of matrices are large.

Concerning the convergence of the algorithms (figure 2), RD-AJD generally reaches its final solution faster than the other algorithms in terms of iterations. When it does not ($n = 32$, FD-AJD), it can be explained by the fact that faster methods converge to a less satisfying solution, which is closer to the initial guess B_0 . However, in terms computational time, RD-AJD, as performed here, is slower than the other methods since it needs to find the closest diagonal matrices at each iteration. This could be corrected by taking approximations of matrices A_k . We will further investigate this in future work.

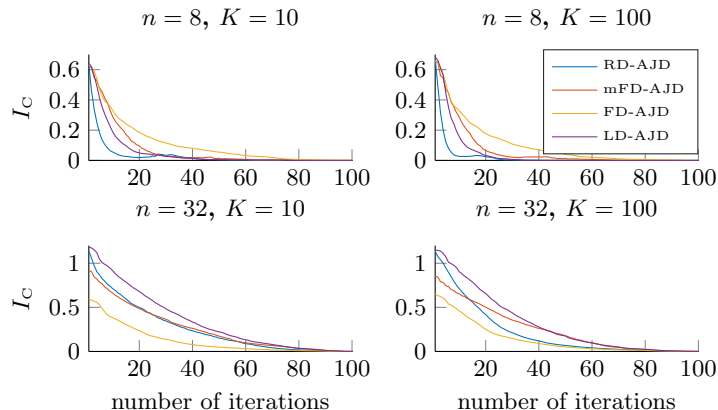


Fig. 2: mean of the convergence of the considered algorithms as a function of the number of iterations for $\sigma = 50$ over 20 trials. RD-AJD generally converges faster in terms of iterations. When it does not, faster methods converge to a less satisfying solution closer to the initial guess B_0 . See text for details.

3.2 Electroencephalographic (EEG) Data

We tested our AJD optimization using the four criteria on an EEG recording acquired on an epileptic patient with 19 electrodes placed according to the international 10-20 system. The sampling rate was 128Hz and the band-pass was 1-32Hz. The data comprised 30 seconds. The BSS of the data was performed using the procedure detailed in [15]. In summary, after a whitening step retaining at least 99.9% of the total variance of the data (reduction to dimension 17 with this data), AJD was performed on the set of Fourier cospectra estimated by 75% overlapping sliding windows of 1 second (Welch method) for frequencies 1 to 32Hz with 1Hz resolution.

Figure 3 shows the last 5 seconds of the original data and the corresponding 17 sources estimated using AJDC [15] performed by the classical AJD algorithm [4], mFD-AJD, mFD-AJD with trace-normalized cospectra, LD-AJD and RD-AJD respectively. Note that FD-AJD gives similar results as compared to mFD-AJD. We are interested here in the three peak-slow wave complexes, often seen in epileptic patients and visible in the original data at frontal locations (figure 3, electrode labels starting with F). For all criteria with the exception of mFD-AJD and LD-AJD, the three peak-slow wave complexes are well separated in an unique source (s3 in figure 3). This shows that RD-AJD does not need any ad-hoc normalization of the input matrices in order to give satisfying results, whereas mFD-AJD and FD-AJD do. Further investigations should be done to compare the sources found by the different methods because the obtained joint diagonalizer are not equivalent. These results demonstrate the accuracy of our optimization procedure and the feasibility of using the natural Riemannian distance criterion.

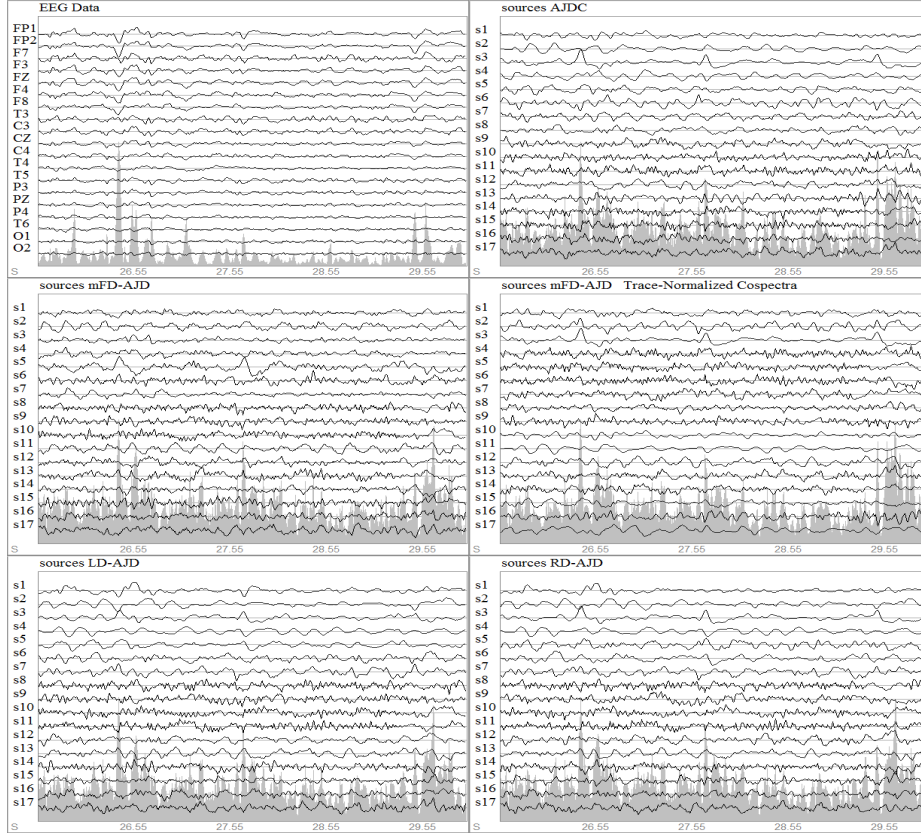


Fig. 3: 5 last seconds of the original EEG data and the 17 sources obtained with classical algorithm AJDC [15], mFD-AJD, mFD-AJD with trace-normalized cospectra, LD-AJD and RD-AJD. See text for details.

4 Conclusions

In this paper, we have provided for the first time an optimization framework for the AJD cost function based on the natural Riemannian distance on the cone of HPD matrices. This problem could not find a solution so far, despite it has been recognized as important [3]. The results obtained on simulated data are promising since our method outperforms the others in terms of accuracy in all cases investigated. Results obtained on real EEG data show that the Riemannian criterion allows to retrieve sources of interest without having to scale the target matrices. Actually, it is invariant by any diagonal scaling which is theoretically advantageous as pointed out in [2, 6].

Here, we limited ourselves to a first order optimization method for simplicity. We will investigate second order methods in future works. We will also study if the differences in the performances arise from the properties of the Riemannian distance, the choice of the closest diagonal matrices, or both. Indeed, we can

replace the diagonal matrices in (11) by those in (2) and reciprocally. Note that this substitution cannot be operated using (3) due to the properties of the determinant.

Acknowledgment

This work has been partially supported by the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01) funded by the French program “Investissement d’avenir” and the European Research Council, project CHESSE 2012-ERC-AdG-320684.

References

1. P. Comon and C. Jutten. *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. Academic Press, 1st edition, 2010.
2. B. Afsari. Sensitivity analysis for the problem of matrix joint diagonalization. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1148–1171, 2008.
3. K. Alyani, M. Congedo, and M. Moakher. Diagonality measures of Hermitian positive-definite matrices with application to the approximate joint diagonalization problem. *Linear Algebra and its Applications*, to be published.
4. P. Tichavský and A. Yeredor. Fast approximate joint diagonalization incorporating weight matrices. *Signal Processing, IEEE Transactions on*, 57(3):878–891, 2009.
5. 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*, 7(1):169–184, 1986.
6. D.-T. Pham. Joint approximate diagonalization of positive definite Hermitian matrices. *SIAM J. Matrix Anal. Appl.*, 22(4):1136–1152, jul 2000.
7. R. Bhatia. *Positive definite matrices*. Princeton University Press, 2009.
8. Maher Moakher. A differential geometric approach to the geometric mean of symmetric positive-definite matrices. *SIAM Journal on Matrix Analysis and Applications*, 26(3):735–747, 2005.
9. P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, USA, 2008.
10. F. Bouchard, L. Korczowski, J. Malick, and M. Congedo. Approximate joint diagonalization within the Riemannian geometry framework. In *24th European Signal Processing Conference (EUSIPCO-2016)*, pages 210–214, 2016.
11. B. C. Carlson and J. M. Keller. Orthogonalization procedures and the localization of wannier functions. *Phys. Rev.*, 105:102–103, Jan 1957.
12. N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15:1455–1459, 2014.
13. M. Congedo, R. Phlypo, and A. Barachant. A fixed-point algorithm for estimating power means of positive definite matrices. In *24th European Signal Processing Conference (EUSIPCO-2016)*, pages 2106–2110, 2016.
14. E. Moreau and O. Macchi. New self-adaptative algorithms for source separation based on contrast functions. In *Higher-Order Statistics, 1993., IEEE Signal Processing Workshop on*, pages 215–219. IEEE, 1993.
15. M. Congedo, C. Gouy-Pailler, and C. Jutten. On the blind source separation of human electroencephalogram by approximate joint diagonalization of second order statistics. *Clinical Neurophysiology*, 119(12):2677–2686, 2008.