Riemannian Optimization and Approximate Joint Diagonalization for Blind Source Separation

Florent Bouchard, Jérôme Malick, Marco Congedo

Riemannian Optimization and Approximate Joint Diagonalization for Blind Source Separation

Florent Bouchard*, Jérôme Malick†, Marco Congedo*
*GIPSA-lab, CNRS, Univ. Grenoble Alpes, Grenoble Institute of Technology, Grenoble France
†LJK, CNRS, Univ. Grenoble Alpes, Grenoble France
Email: florent.bouchard@gipsa-lab.fr

***Abstract*—We consider the blind source separation (BSS) problem and the closely related approximate joint diagonalization (AJD) problem of symmetric positive difinite (SPD) matrices. These two problems can be reduced to an optimization problem with three key components: the criterion to minimize, the constraint on the solution, and the optimization algorithm to solve it. This article contains two contributions that allow to treat these issues independently. We build the first complete Riemannian optimization framework suited for BSS and AJD handling three classical constraints, and allowing to use a large panel of general optimization algorithms on manifolds. We also perform a thorough study of the AJD problem of SPD matrices from an information geometry point of view. We study AJD criteria based on several divergences of the set of SPD matrices, provide three optimization strategies to minimize them, and analyze their properties. Our numerical experiments on simulated and pseudo-real electroencephalographic data show the interest of the Riemannian optimization framework and of the different AJD criteria we consider.**

***Index Terms*—approximate joint diagonalization, Riemannian optimization, Riemannian geometry, symmetric positive definite matrices, blind source separation.**

I. Introduction

Blind source separation (BSS), a family of methods to which independent component analysis (ICA) belongs [1]–[3], has become a major tool for signal processing and data analysis in a wide variety of engineering fields such as communications, image processing, audio and biomedical signals analysis. See [3] for a full review of theory and applications. In this article we consider the linear instantaneous BSS problem, which is based on the following mixing model

$$x(t) = As(t), \tag{1}$$

where t is in $\{1, .., T\}$, $x(t)$ in $\mathbb{R}^n$ is the observation, $s(t)$ in $\mathbb{R}^p$ is the source process, and A in $\mathbb{R}^{n\times p}$ is a full rank mixing matrix. Here, we treat the determined case where $p = n$, however it is possible to handle the overdetermined case ($p < n$) by reducing the dimension of $x(t)$ with a *prewhitening* step. Knowing the T observations $x(t)$, the goal is to retrieve estimates $(\hat{A}, \hat{s})$ of (A, s) under some assumptions on the source process only, such as *statistical independence* [3]. It is well known that there is not a unique solution (A, s) to this recovery problem. For any permutation matrix $P \in \mathbb{R}^{n\times n}$ and non singular diagonal matrix $\Sigma \in \mathbb{R}^{n\times n}$, the couple $(AP^T\Sigma^{-1}, \Sigma P s)$ is a solution equivalent to (A, s). The BSS problem is usually reduced to seeking an invertible unmixing matrix B yielding estimates $\hat{A} = B^{-1}$ and $\hat{s}(t) = Bx(t)$, where the order and scaling of $\hat{s}(t)$ are arbitratry.

In the original formulation of BSS the independence of the sources is modeled in terms of *mutual information* (see chapter 2 in [3]). The solution is obtained when the mutual information of $Bx(t)$ is minimal. Another closely related approach is based on the *maximum likelihood* applied on $Bx(t)$ (see chapter 4 in [3]). It is also possible to solve the BSS by an *approximate joint diagonalization* (AJD) of matrices containing statistics of the observations $x(t)$. Introduced for BSS in [4], the concept of AJD was developped in the pioneering article [5]. Given a set $\{C_k\}$ of K symmetric matrices, we suppose that for all $1 \leq k \leq K$

$$C_k = A\Lambda_k A^T + N_k, \tag{2}$$

where matrices Λ_k are diagonal matrices containing statistics of the sources $s(t)$ and matrices N_k comprise estimation error and measurement noise. The solution of the AJD problem is a joint diagonalizer B of the matrices C_k, which is obtained by minimizing a criterion measuring the degree of diagonality of the set $\{BC_kB^T\}$. In this paper, we focus on the case where matrices C_k are *symmetric positive definite* (SPD), which has drawn much attention in previous research [5]–[8] and where a geometrical point of view can be exploited [8], [9]. Depending on the data $x(t)$ to analyze, one can build the set $\{C_k\}$ by exploiting second order statistics (see *e.g.* [3], [10], [11]), in which case the obtained matrices are SPD given enough samples. It is also possible to exploit higher-order statistics by diagonalizing slices of data cumulants [2]–[4]. Concerning the identifiability conditions of the AJD problem we refer the reader to [7]. So far, the overwhelming majority of proposed AJD methods have been using either the *Frobenius* distance (least squares) [4], [7], [12]–[14] or the *log-likelihood* [5]–[7] to define a diagonality measure. For the Frobenius distance, the resulting criterion is

$$f_{\mathrm{F}}(B) = \sum_k w_k \left\| BC_kB^T - \mathrm{ddiag}(BC_kB^T) \right\|_F^2, \tag{3}$$

where w_k are positive weights, $\|\cdot\|_F$ denotes the Frobenius norm and $\mathrm{ddiag}(\cdot)$ cancels the off-diagonal elements of its argument. Note that (3) is quite generic and only requires the matrices C_k to be symmetric. The log-likelihood [6], [10] (also referred to as the left Kullback-Leibler measure) reads

$$f_{\mathrm{lKL}}(B) = \sum_k w_k \log \frac{\det(\mathrm{ddiag}(BC_kB^T))}{\det(BC_kB^T)}, \tag{4}$$

where $\det(\cdot)$ denotes the determinant. In order to use (4) the matrices C_k must be SPD. There is no closed form solution for the minimizer of such criteria. Therefore, one has to approximate it with an iterative optimization process. Many optimization methods have been considered in previous research such as Jacobi-like algorithms [4]–[6], Lagrange multipliers [13], gradient descent [15], pseudo-Newton [16], Gauss iterations [14] and natural gradient [7], [17], [18].

In order to avoid degenerate solutions, for instance the trivial solution $B = 0$, constraints on B are needed. In early studies, the solution was supposed orthogonal after a prewhitening step meant to orthogonalize the data [2]–[4]. However, it is well known that this induces irreversible errors in the estimation of B, thus research has turned toward methods seeking a non-orthogonal unmixing matrix [6], [12], [17], [19], [20]. Various contraints have been introduced, such as the non-holonomic constraint [7], [15], [21], [22] (avoid diagonal scaling), fixing the determinant [22] or the norm of the lines of B [13], [14], [20], [23]. In this paper, we consider three different classic constraints. The first consists in fixing the determinant as

$$\det(BB^T) = 1. \tag{5}$$

The second is the unit lines norm constraint [14], [20], [23]

$$\mathrm{ddiag}(BB^T) = I_n, \tag{6}$$

where I_n is the identity matrix. Note that in [14], [23], the constraint is $\mathrm{ddiag}(BGB^T) = I_n$ for some SPD matrix G. It is equivalent to (6) when performing BSS on $G^{-1/2}x(t)$ instead that on $x(t)$. The third and last constraint we consider is specific to the AJD problem and also involves fixing the norm of the lines of B; called *intrinsic contraint*, it was introduced in [13], [16] and can be written as

$$\sum_{k=1}^{K} \mathrm{ddiag}(BC_kB^T)^2 = I_n. \tag{7}$$

Any AJD problem (and more generally linear BSS) boils down to solving an optimization problem over a subset $\mathcal{M}$ of GL_n (set of invertible matrices in $\mathbb{R}^{n\times n}$) endowing the constraints for a cost function $f : \mathrm{GL}_n \to \mathbb{R}$. Indeed, B is solution to

$$\underset{B\in\mathcal{M}\subset\mathrm{GL}_n}{\mathrm{argmin}} \quad f(B). \tag{8}$$

There are three key components when modeling and solving this problem: *(i)* the choice of the criterion f to optimize, *(ii)* the constraint on B (definition of $\mathcal{M}$) and how it is enforced, and *(iii)* the optimization algorithm employed to approximate (8). Here we propose an unifying framework to treat these three issues, subsuming and generalizing previous research. Our framework thus provides a synthetic, yet comprehensive view on the AJD of SPD matrices.

Our first contribution mainly concerns points *(ii)* and *(iii)*. As discussed before, to find the solution to (8) previous works are specific to a criterion, a constraint and an iterative process. Therefore, in most cases one cannot simply change one of these three components to obtain a new algorithm. Furthermore, constraints on B are often included as a penalty term in the cost function. A more natural way to handle constraints is to consider optimization on Riemannian manifolds [24]. This also allows to use generic optimization algorithms that are neither specific to a constraint nor to a cost function. Therefore, with Riemannian geometry it is simple to change *(i)*, *(ii)* or *(iii)* without affecting the others, increasing the modularity of developed methods. Ideas of Riemannian optimization have already been used in the context of BSS but only partly in [7], [17], [18], [22] and with a manifold that does not ensure the solution to be invertible in [25]. In section II, we define a new Riemannian manifold based on the *polar decomposition* (section II-A) and we derive appropriate submanifolds to handle constraints (5), (6) and (7) (section II-B). We develop all the tools needed to use any general optimization algorithms on manifolds [24] and to make use of the newly defined manifolds transparent to GL_n in practice. To the best of our knowledge, except from the parts anticipated in [26], this is the first paper to propose a general and complete Riemannian framework for BSS.

Our second contribution, reported in section III, is a thorough theoretical study of the AJD problem with respect to the points *(i)* and *(iii)*. Previous research on AJD criteria have been mainly empirical: criterion (3) arises from the practical reasonning that one needs to cancel the off-diagonal elements of the matrices in order to diagonalize them and criterion (4) was derived from the maximum likelihood approach [10]. Recent advances in the study of the geometry of SPD matrices [8], [27]–[32] have yielded a new perspective: all AJD criteria can be generalized as

$$f(B) = \sum_k w_k d(BC_kB^T, \Lambda_k(B)), \tag{9}$$

where $d(\cdot,\cdot)$ is a divergence function and $\Lambda_k(B)$ are diagonal matrices. Such criterion f measures the degree of diagonality of the set $\{BC_kB^T\}$ according to the divergence $d(\cdot,\cdot)$ and the target diagonal matrices $\Lambda_k(B)$. So far, all research has considered $\Lambda_k(B) = \mathrm{ddiag}(BC_kB^T)$. However, [8] showed that this is not always the closest diagonal matrix to BC_kB^T according to $d(\cdot,\cdot)$. Thus, it might be advantageous to choose $\Lambda_k(B)$ otherwise. This shows that for AJD, issue *(i)* is actually divided into two subissues: the choice of the divergence and the choice of the target diagonal matrices. To solve (8) for cost functions with form (9), different optimization strategies can be employed and we give three possible Riemannian-oriented choices in section III-A. Then in section III-B we consider criteria with form (9) obtained from the Frobenius distance, Kullback-Leibler divergence (which yields a left, right and symmetric measure), log-det α-divergence, log-Euclidean, natural Riemannian and Wasserstein distances. In previous research, only the Frobenius distance, left Kullback-Leibler measure and log-det α-divergence [8] have been considered. The natural Riemannian distance was partly anticipated in [33]. Different choices of divergence and target diagonal matrices yield different AJD criteria, which possess different properties. In section III-C, we give four properties shown to be of interest [7]–[9] and we establish whether the criteria considered in this article possess them.

Section IV contains numerical experiments on the AJD problem where we compare the performance of the algorithms resulting from sections II and III with state of the art competitors [6], [14], [34] both on simulated SPD matrices (section IV-A) and pseudo-real electroencephalographic (EEG) data (section IV-B). We perform a systematic evaluation of all possible combinations of constraints, optimization strategies and criteria. Therefore, we can estimate the influence of those constituents of the AJD problem.

Finally, section V draws several conclusions and perspectives. We discuss possible extensions of the Riemannian optimization framework and perspectives for future AJD research, especially on the need to better understand the links between AJD and the concept of center of mass of SPD matrices We conclude on the possible generalizations of this work to other models beyond linear BSS. For better readibility, the proofs of the propositions and other results are given in supplementary materials.

II. The Riemannian optimization framework

In this section we present a Riemannian framework to solve optimization problems of the form (8) with contraints (5), (6) or (7). Riemannian optimization [24] requires three ingredients: a Riemannian matrix manifold $\mathcal{M}$, a function f defined from $\mathcal{M}$ onto $\mathbb{R}$ (along with its Riemannian gradient and Hessian), and a retraction R on $\mathcal{M}$ (mapping from the tangent space back onto the manifold). As shown in figure 1, given iterate B_i, a descent direction ξ_i in the tangent space $T_{B_i}\mathcal{M}$ of $\mathcal{M}$ at B_i is obtained from the Riemannian gradient (and Hessian) and the next iterate $B_{i+1} = R_{B_i}(\xi_i)$ results from the retraction of ξ_i at B_i. The descent direction ξ_i is obtained using Riemannian versions of generic methods such as steepest-descent, conjugate gradient, Newton's method or trust-region method [24]. Therefore, we first need to define the Riemannian manifolds of interest, which, in addition to the definition of the set, requires to describe its tangent space and a Riemannian metric (inner product on the tangent space). Then, in order to manipulate any cost function we need the relation between Riemannian and Euclidean (classical definitions) gradient and Hessian. Finally, we need a retraction on the manifolds of interest in order to update the solution. There is an implicit retraction defined on a manifold $\mathcal{M}$ that arises from its Riemannian geometry: the exponential map defined from the shortest path connecting two points (the geodesic). Other retractions can be considered and are sometimes preferable [24], [35].

In our context, a Riemannian optimization framework can be developed on the polar manifold $\mathcal{P}_n$, a Riemannian manifold equivalent to GL_n that we study in section II-A. We then define new submanifolds of $\mathcal{P}_n$ embedding constraints (5),

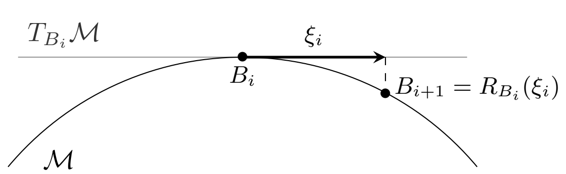

Fig. 1: Schematic illustration of Riemannian optimization. Given the iterate B_i in $\mathcal{M}$ and the descent direction ξ_i in $T_{B_i}\mathcal{M}$, the next iterate B_{i+1} is obtained through the retraction of ξ_i at B_i.

(6) and (7) in section II-B. For better readibility, the proofs of the following lemmas and propositions are reported in supplementary materials.

A. Polar manifold $\mathcal{P}_n$

A way to avoid GL_n is to consider the *polar decomposition*: given B in GL_n, there exist unique matrices $S \in \mathcal{S}_n^{++}$ (set of SPD matrices) and $U \in \mathcal{O}_n$ (set of orthogonal matrices) such that $B = SU$. It follows that GL_n is "equivalent" to the product manifold $\mathcal{P}_n = \mathcal{S}_n^{++} \times \mathcal{O}_n$, which we call the *polar manifold*. More precisely, $\mathcal{P}_n$ is diffeomorphic to GL_n with the mapping $\pi\colon \mathcal{P}_n \to \mathrm{GL}_n$ such that $\pi(S,U) = SU$. In the following, $\mathcal{B}$ denotes a couple (S,U) and B denotes the corresponding matrix $\pi(\mathcal{B})$. By slight abuse of notation, f denotes a function either defined on GL_n or $\mathcal{P}_n$ (the corresponding function is obtained by taking $f\circ\pi$ or $f\circ\pi^{-1}$).

The geometry of $\mathcal{P}_n$ follows from those of $\mathcal{S}_n^{++}$ and $\mathcal{O}_n$, which are well known [24], [27]. We recall here the main ingredients that are used in this article. First note that $\mathcal{P}_n$ lies into the Euclidean space $\mathcal{E}_n = \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times n}$. The tangent space $T_{\mathcal{B}}\mathcal{P}_n$ of $\mathcal{P}_n$ at $\mathcal{B} = (S,U)$ is $T_S\mathcal{S}_n^{++} \times T_U\mathcal{O}_n$, *i.e.*,

$$T_{\mathcal{B}}\mathcal{P}_n = \{(\xi_S, \xi_U) \in \mathcal{E}_n \colon \xi_S^T = \xi_S,\ U\xi_U^T + \xi_U U^T = 0\}. \quad (10)$$

We endow $\mathcal{P}_n$ with metric $\langle\cdot,\cdot\rangle_{\cdot}$, which corresponds to the sum of the classical metrics of $\mathcal{S}_n^{++}$ and $\mathcal{O}_n$. It is defined for $\mathcal{B} = (S,U)$ in $\mathcal{P}_n$, $\xi = (\xi_S, \xi_U)$ and $\eta = (\eta_S, \eta_U)$ in $T_{\mathcal{B}}\mathcal{P}_n$ as

$$\langle \xi, \eta \rangle_{\mathcal{B}} = \mathrm{tr}(S^{-1}\xi_S S^{-1}\eta_S) + \mathrm{tr}(\xi_U^T \eta_U), \quad (11)$$

where $\mathrm{tr}(\cdot)$ denotes the trace operator.

In order to obtain the Riemannian gradient and Hessian of a function f on $\mathcal{P}_n$ from the classical ones in $\mathcal{E}_n$, we first need to define the orthogonal projection map $P_{\mathcal{B}}$ to the metric (11) defined from $\mathcal{E}_n$ onto $T_{\mathcal{B}}\mathcal{P}_n$ such that, for $\mathcal{Z} = (Z_S, Z_U)$,

$$P_{\mathcal{B}}(\mathcal{Z}) = \left(\mathrm{sym}(Z_S), Z_U - U\,\mathrm{sym}(U^T Z_U)\right), \quad (12)$$

where $\mathrm{sym}(\cdot)$ returns the symmetrical part of its argument. In the following, $P_S(Z_S)$ and $P_U(Z_U)$ denote the first and second components of $P_{\mathcal{B}}(\mathcal{Z})$ respectively (the same type of notations are used for gradients, Hessians and retractions). One then obtain the Riemannian gradient and Hessian in $T_{\mathcal{B}}\mathcal{P}_n$ of f at $\mathcal{B}$ in $\mathcal{P}_n$ from the Euclidean ones in $\mathcal{E}_n$ with expressions

$$\mathrm{grad}_{\mathcal{P}_n} f(\mathcal{B}) = \left(SP_S(\mathrm{grad}_{\mathcal{E}_n} f(S))S, P_U(\mathrm{grad}_{\mathcal{E}_n} f(U))\right) \quad (13)$$

and

$$\begin{aligned} \mathrm{Hess}_{\mathcal{P}_n} f(S)[\xi] &= SP_S(\mathrm{Hess}_{\mathcal{E}_n} f(S)[\xi])S \\ &\quad + P_S(\xi_S P_S(\mathrm{grad}_{\mathcal{E}_n} f(S))S), \\ \mathrm{Hess}_{\mathcal{P}_n} f(U)[\xi] &= P_U(\mathrm{Hess}_{\mathcal{E}_n} f(U)[\xi]) \\ &\quad - P_U(\xi_U\,\mathrm{sym}(U^T \mathrm{grad}_{\mathcal{E}_n} f(U))). \end{aligned} \quad (14)$$

This simplifies the manipulations of a cost function f defined on $\mathcal{P}_n$, however to solve problem (8) one may wish to manipulate only f defined on GL_n. This is the purpose of the following proposition, where the Euclidean gradient and Hessian of f on $\mathcal{P}_n$ are obtained from the Euclidean gradient and Hessian of f on GL_n denoted $\mathrm{grad}_{\mathrm{GL}} f$ and $\mathrm{Hess}_{\mathrm{GL}} f$:

Proposition 1. *Let $\mathcal{B} = (S, U)$ and $\xi = (\xi_S, \xi_U)$. Then,*

$$\operatorname{grad}_{\mathcal{E}_n} f(\mathcal{B}) = (\operatorname{grad}_{\mathrm{GL}} f(\pi(\mathcal{B}))U^T, S \operatorname{grad}_{\mathrm{GL}} f(\pi(\mathcal{B}))),$$

$$\begin{aligned}\operatorname{Hess}_{\mathcal{E}_n} f(S)[\xi] = {} & \operatorname{Hess}_{\mathrm{GL}} f(\pi(\mathcal{B}))[\xi_S U + S\xi_U]U^T \\ & + \operatorname{grad}_{\mathrm{GL}} f(\pi(\mathcal{B}))\xi_U^T,\end{aligned}$$

$$\begin{aligned}\operatorname{Hess}_{\mathcal{E}_n} f(U)[\xi] = {} & S \operatorname{Hess}_{\mathrm{GL}} f(\pi(\mathcal{B}))[\xi_S U + S\xi_U] \\ & + \xi_S \operatorname{grad}_{\mathrm{GL}} f(\pi(\mathcal{B})).\end{aligned}$$

Finally a retraction $R_{\mathcal{B}}$ on $\mathcal{P}_n$ at $\mathcal{B}$ is properly defined by the exponential map on $\mathcal{P}_n$, which is given, for all $\xi = (\xi_S, \xi_U)$ in $T_{\mathcal{B}}\mathcal{P}_n$, by

$$R_{\mathcal{B}}(\xi) = \exp_{\mathcal{B}}^{\mathcal{P}_n}(\xi) = \left(\exp_S^{\mathcal{S}_n^{++}}(\xi_S), \exp_U^{\mathcal{O}_n}(\xi_U)\right), \quad (15)$$

where $\exp_S^{\mathcal{S}_n^{++}}(\xi_S) = S^{1/2}\exp(S^{-1/2}\xi_S S^{-1/2})S^{1/2}$ is the exponential map on $\mathcal{S}_n^{++}$ ($\exp(\cdot)$ is the standard matrix exponential) and $\exp_U^{\mathcal{O}_n}(\xi_U)$ is the exponential map on $\mathcal{O}_n$, which is given in equation (5.26) of [24]. All the tools needed for solving (8) through Riemannian optimization on $\mathcal{P}_n$ have been introduced. We can now define appropriate submanifolds embedding the constraints.

B. Submanifolds of $\mathcal{P}_n$

Our objective is to embed constraints (5), (6) and (7) by defining the corresponding subsets of $\mathcal{P}_n$. Constraint (5) is expressed in $\mathcal{P}_n$ as $\det(\pi(\mathcal{B})\pi(\mathcal{B})^T) = 1$. It leads to the subspace of $\mathcal{P}_n$

$$\mathcal{SP}_n = \{(S, U) \in \mathcal{P}_n : \det(S) = 1\}, \quad (16)$$

which we call the *special polar manifold*. Constraint (6) becomes $\operatorname{ddiag}(\pi(\mathcal{B})\pi(\mathcal{B})^T) = I_n$ in $\mathcal{P}_n$. It leads to the subspace of $\mathcal{P}_n$

$$\mathcal{OP}_n = \{(S, U) \in \mathcal{P}_n : \operatorname{ddiag}(S^2) = I_n\}, \quad (17)$$

which we name the *oblique polar manifold*. Finally, constraint (7) leads to the subspace

$$\mathcal{IP}_n = \left\{(S, U) \in \mathcal{P}_n : \sum_k \operatorname{ddiag}(BC_kB^T)^2 = I_n\right\}, \quad (18)$$

which we refer to as the *intrinsic polar manifold*. In proposition 2, we give the first ingredient needed for Riemannian optimization, *i.e.* we define the Riemannian geometry of these three subspaces.

Proposition 2. *Subspaces $\mathcal{SP}_n$, $\mathcal{OP}_n$ and $\mathcal{IP}_n$ are connected submanifolds of $\mathcal{P}_n$. The tangent spaces of $\mathcal{SP}_n$, $\mathcal{OP}_n$ and $\mathcal{IP}_n$ at $\mathcal{B}$ are subspaces of $T_{\mathcal{B}}\mathcal{P}_n$ characterized for $\xi = (\xi_S, \xi_U)$ by equations*

$\mathcal{SP}_n$	$\operatorname{tr}(S^{-1}\xi_S) = 0$
$\mathcal{OP}_n$	$\operatorname{ddiag}(S\xi_S) = 0$
$\mathcal{IP}_n$	$\operatorname{ddiag}(\dot{B}Q) = 0$

where $\dot{B} = \xi_S U + S\xi_U$ is the derivative of $B = \pi(\mathcal{B})$ in direction ξ and $Q = \sum_k C_k B^T \operatorname{ddiag}(BC_kB^T)$. Finally, manifolds $\mathcal{SP}_n$, $\mathcal{OP}_n$ and $\mathcal{IP}_n$ inherit their metrics from the one of $\mathcal{P}_n$ defined in (11).

As for the polar manifold, we need to define the orthogonal projection maps on the tangent spaces of $\mathcal{SP}_n$, $\mathcal{OP}_n$ and $\mathcal{IP}_n$ in order to obtain the Riemannian gradients and Hessians. In the following lemma we give the orthogonal projection maps from the tangent space of $\mathcal{P}_n$. They can be obtained from the ambient space $\mathcal{E}_n$ with (12).

Lemma 1. *The projection maps $P_{\mathcal{B}}^{\mathcal{S}}$, $P_{\mathcal{B}}^{\mathcal{O}}$ and $P_{\mathcal{B}}^{\mathcal{I}}$ on $T_{\mathcal{B}}\mathcal{SP}_n$, $T_{\mathcal{B}}\mathcal{OP}_n$ and $T_{\mathcal{B}}\mathcal{IP}_n$ are respectively given, for $\mathcal{Z} = (Z_S, Z_U)$ in $T_{\mathcal{B}}\mathcal{P}_n$, by*

$\mathcal{SP}_n$	$(Z_S - \frac{1}{n}\operatorname{tr}(S^{-1}Z_S)S, Z_U)$
$\mathcal{OP}_n$	$(Z_S - P_S(S^2\Delta_{\mathcal{O}}S), Z_U)$
$\mathcal{IP}_n$	$(Z_S - SP_S(UQ\Delta_{\mathcal{I}})S, Z_U - P_U(S\Delta_{\mathcal{I}}Q^T))$

where $\Delta_{\mathcal{O}}$ is the unique diagonal matrix solution to

$$\operatorname{ddiag}(P_S(S^2\Delta_{\mathcal{O}}S)S) = \operatorname{ddiag}(Z_S S) \quad (19)$$

and $\Delta_{\mathcal{I}}$ is the unique diagonal matrix solution to

$$\begin{aligned}&\operatorname{ddiag}(S(P_S(UQ\Delta_{\mathcal{I}})B + P_U(S\Delta_{\mathcal{I}}Q^T))Q) \\ &\qquad = \operatorname{ddiag}((Z_S U + SZ_U)Q). \quad (20)\end{aligned}$$

We also need the derivatives of Q (proposition 2), $\Delta_{\mathcal{O}}$ and $\Delta_{\mathcal{I}}$ (lemma 1) at $\mathcal{B}$ in the direction ξ to obtain the Hessians. We easily obtain the derivative $\dot{Q}$ of Q as

$$\dot{Q} = \sum_k C_k(\dot{B}^T \operatorname{ddiag}(BC_kB^T) + 2B^T \operatorname{ddiag}(\dot{B}C_kB^T)). \quad (21)$$

For $\Delta_{\mathcal{O}}$ and $\Delta_{\mathcal{I}}$, it is more complicated since they are defined implicitly. We thus need the following lemmas.

Lemma 2. *The derivative $\dot{\Delta}_{\mathcal{O}}$ of $\Delta_{\mathcal{O}}$ defined in* (19) *at S in the direction ξ_S is the unique diagonal matrix solution to*

$$\operatorname{ddiag}(P_S(S^2\dot{\Delta}_{\mathcal{O}}S)S) = \operatorname{ddiag}(Z_S\xi_S - W_{\mathcal{O}})$$

where

$$W_{\mathcal{O}} = P_S(S^2\Delta_{\mathcal{O}}\xi_S + (\xi_S S + S\xi_S)\Delta_{\mathcal{O}}S)S + P_S(S^2\Delta_{\mathcal{O}}S)\xi_S.$$

Lemma 3. *The derivative $\dot{\Delta}_{\mathcal{I}}$ of $\Delta_{\mathcal{I}}$ defined in* (20) *at $\mathcal{B}$ in the direction $\xi = (\xi_S, \xi_U)$ is the unique diagonal matrix solution to*

$$\begin{aligned}&\operatorname{ddiag}(S(P_S(UQ\dot{\Delta}_{\mathcal{I}})B + P_U(S\dot{\Delta}_{\mathcal{I}}Q^T))Q) = \\ &\quad \operatorname{ddiag}((SZ_U + Z_S U)\dot{Q} + (\xi_S Z_U + Z_S\xi_U)Q - W_{\mathcal{I}}),\end{aligned}$$

where

$$\begin{aligned}W_{\mathcal{I}} &= \xi_S EQ + SHQ + SE\dot{Q}, \\ E &= P_S(UQ\Delta_{\mathcal{I}})B + P_U(S\Delta_{\mathcal{I}}Q^T), \\ H &= \frac{1}{2}(\xi_S\Delta_{\mathcal{I}}Q^T + S\Delta_{\mathcal{I}}\dot{Q}^T) \\ &\quad + \frac{\Delta_{\mathcal{I}}}{2}(\dot{Q}^T U^T B + Q^T\xi_U^T B + Q^T U^T\dot{B}).\end{aligned}$$

This leads to proposition 3 where the Riemannian gradients and Hessians of f in $\mathcal{SP}_n$, $\mathcal{OP}_n$ and $\mathcal{IP}_n$ are obtained from the Riemannian gradient and Hessian of f in $\mathcal{P}_n$. One can obtain them from the gradient and Hessian of f in GL_n by applying proposition 1, (13) and (14) first.

Proposition 3. *The Riemannian gradients of f at $\mathcal{B}$ in $\mathcal{SP}_n$, $\mathcal{OP}_n$ and $\mathcal{IP}_n$ are*

$\mathcal{SP}_n$	$P^{\mathcal{S}}_{\mathcal{B}}(\mathrm{grad}_{\mathcal{P}}\, f(\mathcal{B}))$
$\mathcal{OP}_n$	$P^{\mathcal{O}}_{\mathcal{B}}(\mathrm{grad}_{\mathcal{P}}\, f(\mathcal{B}))$
$\mathcal{IP}_n$	$P^{\mathcal{I}}_{\mathcal{B}}(\mathrm{grad}_{\mathcal{P}}\, f(\mathcal{B}))$

.

The Hessians are given, for a tangent vector $\xi = (\xi_S, \xi_U)$, by

$\mathcal{SP}_n$	$P^{\mathcal{S}}_{\mathcal{B}}(\mathrm{Hess}_{\mathcal{P}}\, f(\mathcal{B})[\xi])$
$\mathcal{OP}_n$	$P^{\mathcal{O}}_{\mathcal{B}}(\mathrm{Hess}_{\mathcal{P}}\, f(\mathcal{B})[\xi]) - P^{\mathcal{O}}_{\mathcal{B}}(SP_S(\dot{M}_{\mathcal{O}})S, 0)$
$\mathcal{IP}_n$	$P^{\mathcal{I}}_{\mathcal{B}}(\mathrm{Hess}_{\mathcal{P}}\, f(\mathcal{B})[\xi])$ $-P^{\mathcal{I}}_{\mathcal{B}}(P_{\mathcal{B}}(\xi_S P_S(M_{\mathcal{I}})S + S\dot{M}_{\mathcal{I}}S,$ $-\xi_U\, \mathrm{sym}(U^T N_{\mathcal{I}}) + \dot{N}_{\mathcal{I}}))$

,

where $\dot{M}_{\mathcal{O}} = \dot{\Delta}_{\mathcal{O}}S + \Delta_{\mathcal{O}}\xi_S$, $M_{\mathcal{I}} = UQ\Delta_{\mathcal{I}}$, $\dot{M}_{\mathcal{I}} = \xi_U Q\Delta_{\mathcal{I}} + U\dot{Q}\Delta_{\mathcal{I}} + UQ\dot{\Delta}_{\mathcal{I}}$, $N_{\mathcal{I}} = S\Delta_{\mathcal{I}}Q^T$ and $\dot{N}_{\mathcal{I}} = \xi_S\Delta_{\mathcal{I}}Q^T + S\dot{\Delta}_{\mathcal{I}}Q^T + S\Delta_{\mathcal{I}}\dot{Q}^T$. Note that $\Delta_{\mathcal{O}}$, $\Delta_{\mathcal{I}}$, $\dot{\Delta}_{\mathcal{O}}$, $\dot{\Delta}_{\mathcal{I}}$ are obtained with (19), (20), *lemmas 2 and 3 with $\mathcal{Z} = \mathrm{grad}_{\mathcal{P}}\, f(\mathcal{B})$.*

It remains to define proper retractions on manifolds $\mathcal{SP}_n$, $\mathcal{OP}_n$ and $\mathcal{IP}_n$. This is achieved by the following

Proposition 4. *The following functions $R^{\mathcal{S}}_{\mathcal{B}}$, $R^{\mathcal{O}}_{\mathcal{B}}$ and $R^{\mathcal{I}}_{\mathcal{B}}$ are retractions at $\mathcal{B}$ in $\mathcal{SP}_n$, $\mathcal{OP}_n$ and $\mathcal{IP}_n$ respectively. They are defined for $\xi = (\xi_S, \xi_U)$ in the tangent space as*

$\mathcal{SP}_n$	$R_{\mathcal{B}}(\xi)$
$\mathcal{OP}_n$	$(F(R_S(\xi_S)), R_U(\xi_U))$
$\mathcal{IP}_n$	$\pi^{-1}(\Upsilon\pi(R_{\mathcal{B}}(\xi)))$

where the function F is defined for R in $\mathcal{S}^{++}_n$ as

$$F(R) = \left(\mathrm{ddiag}(R^2)^{-1/2} R^2\, \mathrm{ddiag}(R^2)^{-1/2}\right)^{1/2} \tag{22}$$

and

$$\Upsilon = \left(\sum_k \mathrm{ddiag}(\pi(R_{\mathcal{B}}(\xi))C_k\pi(R_{\mathcal{B}}(\xi))^T)^2\right)^{-1/4}. \tag{23}$$

This completes our Riemannian optimization framework and we can now use it to develop new AJD methods for SPD matrices, given classical (Euclidean) gradients and Hessians of the criteria in GL_n.

III. Application to the AJD of SPD matrices

The goal of AJD is: given a set $\{C_k\}$ of K SPD matrices, find a joint diagonalizer B in GL_n such that the set $\{BC_kB^T\}$ contains matrices as much diagonal as possible according to a criterion with form (9). Such cost function only depends on the choice of the divergence $d(\cdot,\cdot)$ and the set $\{\Lambda_k(B)\}$ of target diagonal matrices. As illustrated in figure 2, these two factors mainly determine how matrices BC_kB^T converge toward $\mathcal{D}^{++}_n$ (set of diagonal matrices with strictly positive elements) as (9) is optimized. Therefore, one can expect that modifying at least one of them yields a different solution B. For a given divergence $d(\cdot,\cdot)$, a natural choice for the target matrices $\Lambda_k(B)$ arises from geometry [8]. These are the closest diagonal matrices to matrices BC_kB^T according to $d(\cdot,\cdot)$, which are defined as

$$\Lambda_k(B) = \underset{\Lambda\in\mathcal{D}^{++}_n}{\mathrm{argmin}}\; d(BC_kB^T, \Lambda). \tag{24}$$

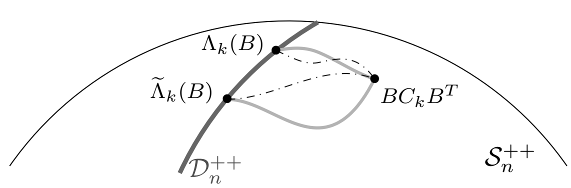

Fig. 2: Schematic illustration of the influence of the divergence and target diagonal matrices on the AJD criterion. The target diagonal matrix ($\Lambda_k(B)$ or $\widetilde{\Lambda}_k(B)$ here) determines where on $\mathcal{D}^{++}_n$ one wants to go. The divergence, which corresponds to a path on $\mathcal{S}^{++}_n$ (curves joining BC_kB^T and $\Lambda_k(B)$ or $\widetilde{\Lambda}_k(B)$ here), determines how to get closer to the target diagonal matrix.

Regarding the minimization of (9), the intuitive approach is to optimize it directly. However, [14] has shown that it can be advantageous to use an indirect optimization scheme in terms of accuracy and numerical efficiency. In such approach an optimization subproblem with a cost function closely related to (9) is considered at each iteration. In section III-A, we present three different optimization strategies: the direct one, a new indirect approach and a Riemannian-oriented version of [14]. A major advantage of the two latter strategies is that they are not dependent on the choice of the target matrices $\Lambda_k(B)$.

Section III-B contains the different divergences that we consider along with the criteria they lead to. In this paper, we are interested in the *Frobenius* distance, *Kullback-Leibler* divergence, *log-det* α-divergence, *natural Riemannian* distance, *log-Euclidean* distance and *Wasserstein* distance. Whenever possible, we use all the optimization strategies of section III-A. For the direct strategy we always choose the closest diagonal matrices according to the divergence as target matrices. For the two others, any diagonal matrices can be used even though we only use the closest ones in our numerical experiments (section IV).

In section III-C, we study four desirable properties of AJD criteria [7]–[9]. For criteria with form (9), these properties depends on the chosen divergence and diagonal target matrices. The first one [7]–[9] concerns the diagonal scaling ambiguity, which is intrinsic to the BSS and AJD problems as explained in section I. The three others [9] are related to the effects of some manipulations of the input matrix set $\{C_k\}$. Here, we are interested in the rescaling of the matrices C_k by strictly positive scalars a_k, their inversion and their congruence with a matrix W in GL_n. Note that the two latter properties are particularly interesting not only in practice, but also since they are linked to well known properties of centers of mass of SPD matrices (self-duality and congruence invariance) [9]. Finally, we determine if the divergences of section III-B yield criteria possessing those properties.

A. Optimization algorithms

All three strategies to minimize (9) share the steps presented at the begining of algorithm 1. The differences between them resides in the update rule on the estimated solution B and in the way they treat the dependence of the target matrices Λ_k

relatively to B. The convergence can be defined in various ways and the choice we make for our numerical experiments is given in section IV.

Algorithm 1: Riemannian AJD

Input: matrices $\{C_k\}$ in $\mathcal{S}_n^{++}$, initial guess B_0 for B
Output: iterates B_i of the estimated joint diagonalizer

Compute matrices $\{B_0 C_k B_0^H\}$ and set $i = 0$.
while *not convergence* **do**
  Obtain B_{i+1} through update rule (a), (b) or (c) on B_i.

Update rule (a): direct strategy

Obtain a descent direction ξ_i from the gradient and Hessian of (9) at B_i.
Compute B_{i+1} through the retraction of ξ_i at B_i.

Update rule (b): indirect strategy

Compute matrices Λ_k.
Obtain a descent direction ξ_i from the gradient and Hessian of (25) at B_i.
Compute B_{i+1} through the retraction of ξ_i at B_i.

Update rule (c): inverse indirect strategy

Compute matrices Λ_k and set $A_0 = I_n$.
Obtain a descent direction ξ_i from the gradient and Hessian of (26) at A_0.
Compute A_i through the retraction of ξ_i at A_0.
$B_{i+1} \leftarrow A_i^{-1} B_i$.

The most natural and intuitive way to minimize (9) is to optimize it directly as described in update rule (a) of algorithm 1. As illustrated in figure 3, the update of B both depends on the divergence and on the resulting modifications of the target diagonal matrices. Indeed, in order to optimize (9) with $\Lambda_k(B)$ as a function of B we need to differentiate it to obtain the gradient and Hessian of (9). However, it can be very complicated to differentiate some functions $B \mapsto \Lambda_k(B)$ and, when testing different possibilities, we need to obtain the gradients and Hessians of (9) for each case, which can be quite annoying.

A way to overcome these limitations is to develop methods where the matrices $\Lambda_k(B)$ are considered fixed at each iteration, *i.e.*, though updated at each iteration, they are treated as a constant when deriving the gradient and Hessian. When applying this strategy on (9) directly, we fix $\Lambda_k = \Lambda_k(B)$ and we consider another optimization subproblem with criterion

$$\widehat{f}(B) = \sum_k w_k d\big(BC_kB^T, \Lambda_k\big) \,. \tag{25}$$

This leads to update rule (b) of algorithm 1. As illustrated in figure 3, at each iteration matrices BC_kB^T get closer to $\Lambda_k(B)$ according to the chosen divergence without taking into account how target matrices $\Lambda_k(B)$ are modified.

Another solution is to use the approach introduced in [14] and adapt it for Riemannian optimization as described in update rule (c) of algorithm 1. At each iteration, given B and targets $\Lambda_k(B)$ we consider the optimization subproblem with criterion

$$\widetilde{f}(A) = \sum_k w_k d\big(BC_kB^T, A\Lambda_kA^T\big) \,. \tag{26}$$

Starting from $A_0 = I_n$, we perform one step in a descent direction of $\widetilde{f}$ to obtain A and update B such as $B \leftarrow A^{-1}B$. Again, matrices BC_kB^T get closer to $\Lambda_k(B)$ without taking into account how target matrices are modified. The difference is that we first move matrices $\Lambda_k(B)$ in $\mathcal{S}_n^{++}$ so that they get closer to BC_kB^T and we then inverse A to move BC_kB^T closer to $\Lambda_k(B)$ as shown in figure 3. With this procedure, optimizing (26) on $\mathcal{OP}_n$ or $\mathcal{IP}_n$ does not ensure constraints (6) and (7). Thus, we only consider this approach on $\mathcal{SP}_n$ here. To use the other constraints, one can optimize (26) on $\mathcal{P}_n$ and apply them on $A^{-1}B$ or develop other suited manifolds.

It would be desirable to prove convergence of algorithm 1, to consider its rate, and to study if the reached solutions are global (or only local) minimizers of the problem. These questions are linked to the property of geodesical convexity of the criteria in the used manifold and would require further studies of the geodesics and of the divergences, which arer beyond the scope of this paper. Let us mention that we have observed numerical convergence for the criteria that we consider in the next section. Another point related to convergence would be a theoretical comparison of the attained solutions. Note that the three update rules generate different iterates and, assuming convergence, they can reach in principle

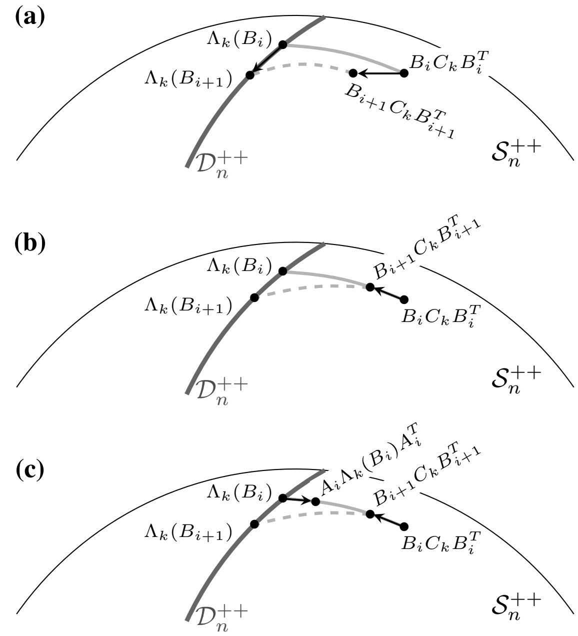

Fig. 3: Schematic illustration of the different optimization strategies in algorithm 1. (a) corresponds to update rule (a) where both the chosen divergence and how the target diagonal matrices are modified are taken into account to obtain B_{i+1}. (b) represents update rule (b) where B_i is modified so that matrices $B_iC_kB_i^T$ get closer to targets $\Lambda_k(B_i)$ according to the divergence, without taking into account the resulting changes in the target diagonal matrices. (c) illustrates update rule (c) where we first find A_i so that matrices $A_i\Lambda_k(B_i)A_i^T$ get closer to $B_iC_kB_i^T$ and then take $B_{i+1} = A_i^{-1}B_i$ to get closer to targets $\Lambda_k(B_i)$. Again, the modifications in the target diagonal matrices are not taken into account.

different (local) minima. However, by looking at the gradients of the criteria of (9) and (25) with $\Lambda_k(B)$ defined as in (24) (see supplementary materials), we see that a local minimum of (9) is also a local minimum of (25). It is more complicated to establish such a result for (26); the reader is referred to [14] for further discussions on this subject.

B. Studied criteria

We consider here different divergences (possibly squared distances) to obtain AJD criteria. For each divergence we give its definition, explain its interest, define its closest diagonal matrix and finally give the different optimization strategies of algorithm 1 that we can consider. For better readibility, Euclidean gradients and Hessians of considered criteria are given in supplementary materials.

1) Frobenius distance: This distance is defined as

$$\delta_{\mathrm{F}}^2(M,\Lambda) = \|M-\Lambda\|_F^2. \tag{27}$$

This is the distance on $\mathcal{S}_n^{++}$ equipped with the Euclidean metric. As explained in section I, it yields the most studied AJD criterion. It arises from the practical reasoning that one wishes to minimize the off-diagonal elements of the matrices to diagonalize. The closest diagonal matrix to M according to this distance is $\Lambda = \mathrm{ddiag}(M)$. With this distance, we can consider all three strategies of algorithm 1. To use update rule (a), one needs to consider the criterion corresponding to (9), which is denoted f_{F} and defined for $\Lambda_k(B) = \mathrm{ddiag}(BC_kB^T)$. For update rule (b), the criterion corresponding to (25) is denoted $\widehat{f}_{\mathrm{F}}$. Finally, the criterion corresponding to (26) is denoted $\widetilde{f}_{\mathrm{F}}$ and allows to use update rule (c). Note that it corresponds to the method proposed in [14].

2) Kullback-Leibler divergence: The Kullback-Leibler divergence between two Gaussian distributions with covariance P and S is defined as

$$d_{\mathrm{KL}}(P,S) = \mathrm{tr}(PS^{-1}-I_n) - \log\det(PS^{-1}). \tag{28}$$

The interest of this divergence arises from the statistical information it gives and its link with the likelihood and the mutual information [10]. As this divergence is not symmetric with respect to its arguments, it gives birth to different diagonality measures to be used in (9).

The first one, referred as the left Kullback-Leibler measure, is $d_{\mathrm{lKL}}(M,\Lambda) = d_{\mathrm{KL}}(M,\Lambda)$. The closest diagonal matrix to M according to this measure is, again, $\Lambda = \mathrm{ddiag}(M)$ [8]. Choosing $\Lambda_k(B) = \mathrm{ddiag}(BC_kB^T)$ yields the historical log likelihood criterion given in (4) and denoted f_{lKL}. Due to the properties of the determinant, it only makes sense to use update rule (a) with this measure. Indeed, if we fix $\Lambda_k(B)$ the gradient and Hessian of the resulting criteria will no longer depend on the data but only on the variable B or A.

The right Kullback-Leibler measure is $d_{\mathrm{rKL}}(M,\Lambda) = d_{\mathrm{KL}}(\Lambda,M)$. The closest diagonal matrix to M for d_{rKL} is $\Lambda = \mathrm{ddiag}(M^{-1})^{-1}$ [8]. Again, the only suited update rule for this measure is (a) and we consider f_{rKL} corresponding to (9) and defined for $\Lambda_k(B) = \mathrm{ddiag}((BC_kB^T)^{-1})^{-1}$.

It is also possible to consider a symmetrized version of the Kullback-Leibler divergence, simply defined as

$$d_{\mathrm{sKL}}(M,\Lambda) = \frac{1}{2}(d_{\mathrm{lKL}}(M,\Lambda) + d_{\mathrm{rKL}}(M,\Lambda)). \tag{29}$$

In this case, the closest diagonal matrix to M is $\Lambda = \mathrm{ddiag}(M)^{1/2}\,\mathrm{ddiag}(M^{-1})^{-1/2}$ [8]. Since the terms involving the determinant vanish, we can use all three update rules of algorithm 1 in this case. The criterion corresponding to (9) is denoted f_{sKL} and defined for

$$\Lambda_k(B) = \mathrm{ddiag}(BC_kB^T)^{1/2}\,\mathrm{ddiag}((BC_kB^T)^{-1})^{-1/2}.$$

The criteria corresponding to (25) and (26) are denoted $\widehat{f}_{\mathrm{sKL}}$ and $\widetilde{f}_{\mathrm{sKL}}$ respectively.

3) Log-det α-divergence: This divergence [29] is defined as

$$d_{\alpha\mathrm{LD}}(M,\Lambda) = \frac{4}{1-\alpha^2}\log\frac{\det(\frac{1-\alpha}{2}M + \frac{1+\alpha}{2}\Lambda)}{\det(M)^{\frac{1-\alpha}{2}}\det(\Lambda)^{\frac{1+\alpha}{2}}}, \tag{30}$$

for $\alpha \in]-1,1[$. It was recently used in [8] for AJD. The interesting properties of this divergence is that we have a continuum in α and when $\alpha \to -1$ and $\alpha \to 1$, it coincides with the right and left Kullback-Leibler measures presented previously. Furthermore, α=0 yields the Bhattacharyya distance [29], also named S-divergence [30]. This distance is of particular importance since it is closely related to the natural Riemannian distance on $\mathcal{S}_n^{++}$ while being numerically cheaper [30]. As shown in [8], the closest diagonal matrix to M according to this measure is the unique Λ solution to

$$\mathrm{ddiag}\left(\left(\frac{1-\alpha}{2}M + \frac{1+\alpha}{2}\Lambda\right)^{-1}\right) = \Lambda^{-1}. \tag{31}$$

A method to solve this equation can be found in supplementary materials. Since this equation is difficult to differentiate, we do not consider the direct optimization strategy of update rule (a) with targets $\Lambda_k(B)$ defined as in (31). However, we can use the schemes of update rules (b) and (c) with criteria $\widehat{f}_{\alpha\mathrm{LD}}$ and $\widetilde{f}_{\alpha\mathrm{LD}}$ corresponding to (25) and (26) respectively.

4) Natural Riemannian distance: As shown in [27], the natural Riemannian distance on $\mathcal{S}_n^{++}$ is

$$\delta_{\mathrm{R}}^2(M,\Lambda) = \left\|\log(\Lambda^{-1/2}M\Lambda^{-1/2})\right\|_F^2. \tag{32}$$

This distance has been obtained both from a pure *differential geometric* point of view [27], [28], [36], where it corresponds to the length of the geodesic (shortest path between two points) and from an *information geometric* point of view, assuming the multivariate Normal distribution of the data and adopting the *Fisher information metric* [37], [38], dating back to the seminal works of Rao [39] and Amari [40]. This Riemannian distance therefore has a major interest when dealing with SPD matrices. As shown in [8], the closest matrix to M is the unique Λ solution to

$$\mathrm{ddiag}(\log(M^{-1}\Lambda)) = 0. \tag{33}$$

A method to solve this equation can be found in supplementary materials. It is cumbersome to consider the direct optimization strategy (update rule (a)) with $\Lambda_k(B)$ defined as in (33). Therefore, we only consider update rules (b) and (c) with criteria $\widehat{f}_{\mathrm{R}}$ and $\widetilde{f}_{\mathrm{R}}$ corresponding to (25) and (26), respectively.

5) Log-Euclidean distance: This distance [31], [36] is

$$\delta^2_{\mathrm{LE}}(M,\Lambda) = \|\log(M) - \log(\Lambda)\|_F^2. \tag{34}$$

If M and Λ commute, the log-Euclidean distance is equivalent to the Riemannian distance. It can be seen as a linearization of the natural Riemannian distance around the identity matrix. Indeed, it corresponds to the Frobenius distance of the projections of M and Λ on the tangent space at the identity matrix. Since this tangent space is a linearization of the manifold of SPD matrices around the identity, its natural distance is the Frobenius one. The closest diagonal matrix to M is $\Lambda = \exp(\mathrm{ddiag}(\log(M)))$. It yields the criterion f_{LE} corresponding to (9) defined for $\Lambda_k(B) = \exp(\mathrm{ddiag}(\log(BC_kB^T)))$. Interestingly, f_{LE} can also be obtained by projecting the matrix BC_kB^T onto the tangent space of I_n (equivalent to projecting C_k on the tangent space of $(B^TB)^{-1}$) and then using the Frobenius distance. We can use the three update rules of algorithm 1 with this distance. The criteria corresponding to (25) and (26) are denoted $\widehat{f}_{\mathrm{LE}}$ and $\widetilde{f}_{\mathrm{LE}}$, respectively.

6) Wasserstein distance: This distance is given by

$$\delta^2_{\mathrm{W}}(M,\Lambda) = \mathrm{tr}\left(\frac{1}{2}(M+\Lambda) - (\Lambda^{1/2}M\Lambda^{1/2})^{1/2}\right). \tag{35}$$

This distance has the major advantage of being defined when matrices M and Λ are positive semidefinite. It plays an important role for optimal transport [41]–[43] and has found interest in the study of covariance matrices [44], [45] and spectral analysis of time series [46]. This distance yields a Riemannian geometry on $\mathcal{S}_n^{++}$, which is fully described in [32]. The closest diagonal matrix to M is Λ solution to

$$\mathrm{ddiag}((\Lambda^{1/2}M\Lambda^{1/2})^{1/2}) = \Lambda. \tag{36}$$

The proof of this result along with a method to find Λ are in supplementary materials. Again, we only consider update rules (b) and (c) and criteria corresponding to (25) and (26) are denoted $\widehat{f}_{\mathrm{W}}$ and $\widetilde{f}_{\mathrm{W}}$.

C. AJD properties

By studying the AJD problem, we can infer several desirable properties for the AJD criteria. In this paper, we are interested in four of them, which have been noticed in [7]–[9]. The first property of interest arises from the fact that B and ΣB are equivalent solutions for any non singular diagonal matrix Σ as explained in section I. Therefore, it is desirable that the criterion f takes the same values at B and ΣB, *i.e.*, that it satisfies the following

Property 1. *The criterion f is said to be invariant by diagonal scaling if, for B in GL_n and any non singular diagonal matrix Σ, we have*

$$f(B) = f(\Sigma B). \tag{37}$$

Assuming that $\Lambda_k(\Sigma B) = \Sigma\Lambda_k(B)\Sigma$, criteria with form (9) possess this property if the divergence satisfies

$$d(\Sigma M\Sigma, \Lambda\Sigma^2) = d(M,\Lambda), \tag{38}$$

for M in $\mathcal{S}_n^{++}$, Λ in $\mathcal{D}_n^{++}$ and a non singular diagonal matrix Σ.

The three other properties concern the effects of some transformations of the matrices C_k to diagonalize. In the following, we denote an AJD criterion of the set $\{C_k\}$ by $f_{\{C_k\}}$ and the corresponding target diagonal matrices are written $\Lambda_{C_k}(B)$ when it is needed. A diagonality measure of a matrix C_k should not depend on its scalingy [9], let us thus consider the following

Property 2. *The criterion f is said to be invariant by rescaling of the input matrices C_k if, given B in GL_n and any strictly positive scalars a_k, we have*

$$f_{\{a_kC_k\}}(B) = f_{\{C_k\}}(B). \tag{39}$$

Assuming that $\Lambda_{a_kC_k}(B) = a_k\Lambda_{C_k}(B)$, (9) possesses this property if we have

$$d(aM, a\Lambda) = d(M,\Lambda), \tag{40}$$

for M in $\mathcal{S}_n^{++}$, Λ in $\mathcal{D}_n^{++}$ and a in $\mathbb{R}_*^+$. Thus, a criterion satisfying property 1 also satisfies property 2. As shown in [11] for the Frobenius distance, criteria without this property might induce the need of an ad-hoc normalization of the matrices C_k in practical BSS applications.

For the third property, observe that if B is the joint diagonalizer of the set $\{C_k\}$, then the joint diagonalizer of the set $\{C_k^{-1}\}$ should be B^{-T}, up to permutation and diagonal scaling ambiguities [9]. Therefore, we consider the following

Property 3. *The criterion f is said to be invariant by inversion of the input matrices $\{C_k\}$ if, for all B in GL_n, we have*

$$f_{\{C_k\}}(B) = f_{\{C_k^{-1}\}}(B^{-T}). \tag{41}$$

For (9) to satisfy this property, we need to have

$$d(M^{-1},\Lambda^{-1}) = d(M,\Lambda), \tag{42}$$

$$\text{and} \quad \Lambda_{C_k}(B) = \Lambda_{C_k^{-1}}(B^{-T})^{-1}. \tag{43}$$

Note that if a divergence satisfies (42) then the corresponding closest diagonal matrix resulting from (24) satisfies (43).

Finally, the last property is obtained by noticing that if B is the joint diagonalizer of the set $\{WC_kW^T\}$ for W in GL_n, then the joint diagonalizer of the set $\{C_k\}$ should be BW, up to permutation and diagonal scaling ambiguities [9], *i.e.*,

Property 4. *The criterion f is said to be invariant by congruence of the input matrices $\{C_k\}$ if, given W and B in GL_n, we have*

$$f_{\{WC_kW^T\}}(B) = f_{\{C_k\}}(BW). \tag{44}$$

Assuming that $\Lambda_{WC_kW^T}(B) = \Lambda_{C_k}(BW)$, this property is always true for criteria with form (9).

The following table indicates which of the properties the criteria of section III-B satisfy.

	F	lKL	rKL	sKL	αLD		R	LE	W
					$\alpha \neq 0$	$\alpha = 0$			
prop. 1		✓	✓	✓	✓	✓	✓		
prop. 2		✓	✓	✓	✓	✓	✓	✓	
prop. 3				✓		✓	✓	✓	
prop. 4	✓	✓	✓	✓	✓	✓	✓	✓	✓

We see that only the symmetric Kullback-Leibler measure, the log-det α-divergence for $\alpha = 0$ and the natural Riemannian distance satisfy all properties. Thus, we hold here that these three criteria are to be preferred, at least from a theoretical point of view. This is in contrast with all previous AJD literature, which has considered only the Frobenius distance, the left Kullback-Leibler measure and the log-det α-divergence, the latter without considering the appropriate target diagonal matrices.

IV. Numerical experiments

In this section, we study the performance of AJD methods on simulated covariance matrices (SPD) first and then on pseudo-real data where real EEG is mixed with a synthetic source signal. Our goal is to compare our Riemannian algorithms with classical ones and to compare the constraints, optimization strategies and cost functions.

AJD methods are obtained by optimizing the criteria of section III-B with the strategies of section III-A. Optimization is performed within the Riemannian framework of section II for constraints (5), (6) and (7). We use a hybrid method that consists in using a Riemannian conjugate gradient (RCG) algorithm first and then a Riemannian trust region (RTR) method initialized with the output of the RCG. We refer the reader to [24] for details on those algorithms. For methods where Λ_k are fixed, we choose the closest diagonal matrices to BC_kB^T according to the distance or divergence the cost function corresponds to. When we do not have explicit solutions for those closest diagonal matrices we compute them by solving a minimum distance optimization problem by the same above-described algorithm combining conjugate gradient and trust region (see supplementary materials).

We denote our algorithms as follows: we first indicate the divergence it corresponds to, then the manifold used for optimization and finally whether we use update rule (a), (b) or (c) with letters a, b or c, respectively. For example, the algorithm resulting from the Frobenius distance δ_{F} on the special polar manifold $\mathcal{SP}_n$ with update rule (a) is denoted F-SP-a. For the log-det α-divergence, we consider values $\alpha \in \{-0.5, 0, 0.5\}$. When needed, we indicate the value of α with a subscript, *e.g.* αLD_0.

The stopping criterion for the iterate B_i is defined as $\|B_{i-1}B_i^{-1} - I_n\|_F^2/n$. Its tolerance is set to $\varepsilon_{\text{RCG}} = 10^{-5}$ for RCG and $\varepsilon_{\text{RTR}} = 10^{-6}$ for RTR. The optimization is performed with the manopt toolbox [47] in Matlab. We compare our algorithms with previously published *NOJoB* [34], *uwedge* [14] and *jadiag* [6], which correspond to F-IP-a, F-SP-c and lKL-SP-a respectively. Here, we compare the performance of the algorithms in term of accuracy and not in term of computing time. In fact, the computing times of the Riemannian algorithms are larger than the ones of classical algorithms, which is partly explained by the fact that we use a generic toolbox [47].

To analyze the results, we perform one-way repeated measures ANOVA (analysis of variance) to test the null hypothesis that the mean performance of the methods within a given comparison block is equivalent. If the ANOVA is significant, which means that at least one of the means is statistically dominant, we perform all pairwise post-hoc comparisons of the mean performance for all methods within the blocks using a repeated measure t-test. In order to account for the number of statistical tests performed we apply a Bonferroni correction: for the ANOVAs, the type one error rate was set to 5.10^{-4}, which approximately corresponds to 0.05 divided by the total number of ANOVAs performed, while the threshold for the post-hoc comparisons was set to 3.10^{-5}, which corresponds to 5.10^{-4} divided by the maximum number of pairwise comparisons within each block.

A. Simulated covariance matrices

We simulate sets of K real valued $n \times n$ SPD matrices C_k according to model [48]

$$C_k = A\Lambda_k A^T + \frac{1}{\sigma} E_k \Delta_k E_k^T + \beta I_n, \tag{45}$$

where matrices A and E_k are random matrices with i.i.d. elements drawn from the standard normal distribution. We further ensure that the condition number with respect to inversion of A is below 20. Free parameter σ and $\beta = 10^{-3}$ define the expected signal to noise ratio and uncorrelated noise, respectively. Diagonal matrices Λ_k and Δ_k hold signal and noise source energies and have i.i.d. elements with the p^{th} element drawn from a chi-squared distribution with expectation $n/p^{1.5}$. Here, we perform the AJD with unit weights w_k for all matrices, we do a pre-whitening of the matrices C_k with the inverse square root of their arithmetic mean and we initialize all the algorithms with the identity matrix.

To estimate the performances of the algorithms, we measure the accuracy by means of the Moreau-Amari index [49]

$$I_{\text{M-A}}(M) = \frac{1}{2n(n-1)} \sum_{p=1}^{n} \left(\frac{\sum_{q=1}^{n} |M_{pq}|}{\max\limits_{1\leq q\leq n} |M_{pq}|} - 1 \right) + \frac{1}{2n(n-1)} \sum_{p=1}^{n} \left(\frac{\sum\limits_{q=1}^{n} |M_{qp}|}{\max\limits_{1\leq q\leq n} |M_{qp}|} - 1 \right), \tag{46}$$

where $M = BA$, with B the estimated joint diagonalizer and A the true mixing matrix of the signal part in (45). Thus, $I_{\text{M-A}}$ is a measure in $[0,1]$ with zero indicating a perfect recovering of signal sources.

1) Riemannian optimization: We first check that Riemannian optimization is appropriate for AJD by comparing the performance of the classical algorithms *NOJoB*, *uwedge* and *jadiag* with the corresponding Riemannian versions F-IP-a, F-SP-c and lKL-SP-a. The median of the performance obtained for these methods are plotted in figure 4 for $n = 32$. The ANOVAs indicate that no difference in the performance of Riemannian and classical algorithms is significant. The same result is obtained with $n = 8$ (data not shown). We conclude that Riemannian optimization is a viable solution to treat the AJD problem.

2) Constraints and AJD optimization algorithms: We then study the influence of the constraints and optimization strategies corresponding to update rules (a), (b) and (c) in algorithm 1. We obtain significant differences in the performance only for the Frobenius distance (figure 5), the log-det α-divergence (figure 6), the Riemannian distance (not

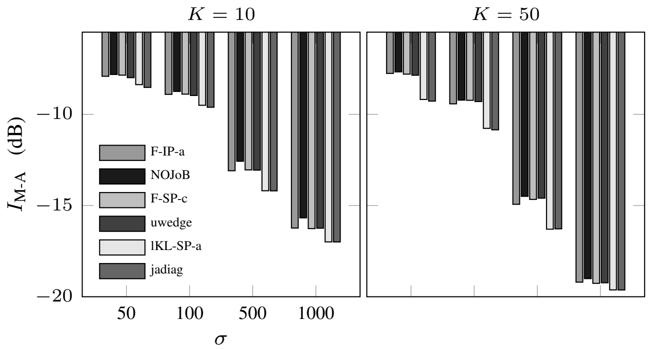

Fig. 4: Median of the performance of the classical algorithms and their corresponding Riemannian versions on simulated data with respect to the noise parameter σ over 100 trials for $n = 32$ and $K = \{10, 50\}$.

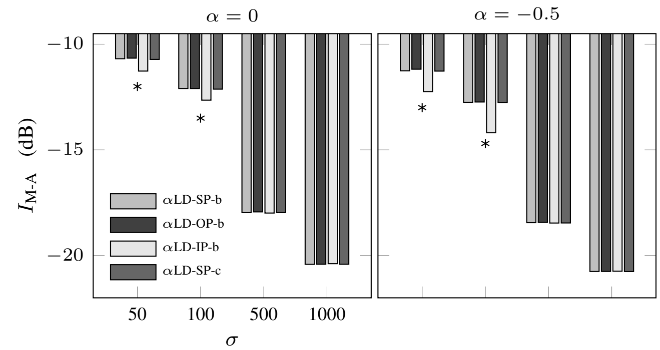

Fig. 6: Median of the performance of algorithms based on the log-det α-divergence for $\alpha = \{0, -0.5\}$ on simulated data with respect to the noise parameter σ over 100 trials for $n = 32$ and $K = 50$. For $\alpha = 0.5$, the results of all algortihms are similar. $*$: significative differences when performing a one-way anova ($p < 5.10^{-4}$) on the performance of all algorithms.

represented since very similar to the log-det α-divergence for $\alpha = 0$) and the Wasserstein distance (figure 7). For the other divergences, no significant result was found, thus we do not show them here. Note that the optimization on $\mathcal{SP}_n$ is not always appropriate when associated with update rule (a) because it often yields degenerate solutions (bad scaling) for the symmetric Kullback-Leibler measure and the log-Euclidean distance (data not shown).

In figure 5, the ANOVAs reveal that the differences between algorithms based on the Frobenius distance are significant for $n = 32$. Post-hoc comparisons reveal that F-SP-a and F-SP-b perform poorly as compared to the others for $\sigma = \{100, 500, 1000\}$. They further show that F-OP-a and F-OP-b exhibit significantly lower performance (higher Moreau-Amari index) as compared to F-IP-a, F-IP-b and F-SP-c for $\sigma = \{50, 100\}$.

In figure 6, the ANOVAs show that there are significant differences between methods based on the log-det α-divergence with $\alpha = \{0, -0.5\}$ for $n = 32$, $K = 50$ and $\sigma = \{50, 100\}$. There are also significant differences for $n = 32$, $K = 10$ and $\sigma = 50$ (data not shown). The post-hoc comparisons show that the performance of αLD-IP-b is higher as compared to the other algorithms for $K = 50$ and $\sigma = \{50, 100\}$.

In figure 7, the ANOVAs reveal significant differences for the methods based on the Wasserstein distance for $n = 32$ and all values of K and σ. Post-hoc comparisons show that W-SP-b features lower performance for $n = 32$ and all values of K and σ as compared to all other algorithms. Furthermore, the performance of W-OP-b is significantly worse as compared to W-IP-b and W-SP-c for all values of K and $\sigma = \{50, 100\}$. Finally, W-SP-c yields significantly better results for $K = 50$ and $\sigma = \{50, 100\}$. Thus, W-SP-c appears to be the most appropriate alogrithm with our simulated data.

3) Criteria: Finally, we look at the influence of the choice of the divergence and target matrices. In figure 8, the performance of algorithms associated with the three Kullback-Leibler measures (left, right, symmetric) and with the log-det α-divergence for $\alpha = \{-0.5, 0, 0.5\}$ are plotted. The ANOVAs show significant differences for $n = 32$ and all values of K and σ. Except for $\alpha\text{LD}_{\text{-0.5}}$-IP-b, a linear trend is clearly visible with the performance increasing in the order of the methods as displayed in figure 8. Post-hoc comparisons reveal that this trend is exacerbated as K increases and σ decreases. They also show that the only significant differences in the performance of $\alpha\text{LD}_{\text{-0.5}}$-IP-b and rKL-IP-a are for $K = 50$ and $\sigma = \{50, 100\}$. Note that these differences are not observed with the other algorithms based on the log-det α-divergence for $\alpha = -0.5$. Therefore, $\alpha\text{LD}_{\text{-0.5}}$-IP-b appears to be the most accurate as compared to the algorithms based on the three Kullback-Leibler measures and the log-det α-divergence for

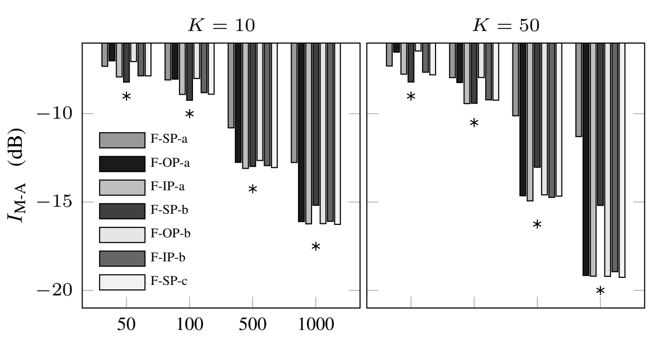

Fig. 5: Median of the performance of algorithms based on the Frobenius distance on simulated data with respect to the noise parameter σ over 100 trials for $n = 32$ and $K = \{10, 50\}$. $*$: significative differences when performing a one-way anova ($p < 5.10^{-4}$) on the performance of all algorithms.

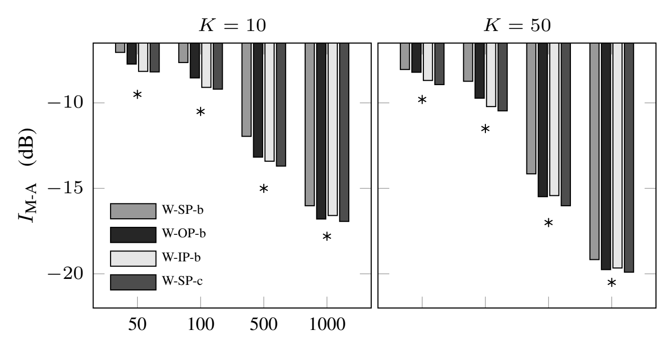

Fig. 7: Median of the performance of algorithms based on the Wasserstein distance on simulated data with respect to the noise parameter σ over 100 trials for $n = 32$ and $K = \{10, 50\}$. $*$: significative differences when performing a one-way anova ($p < 5.10^{-4}$) on the performance of all algorithms.

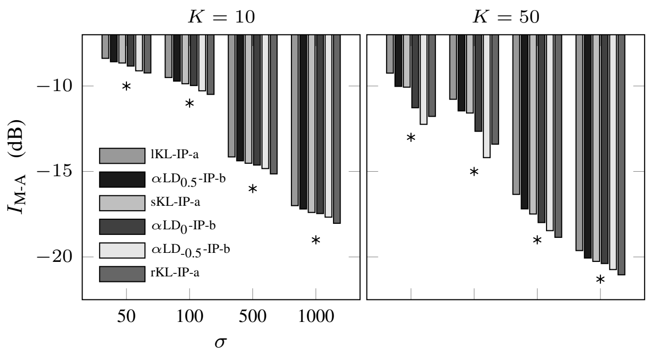

Fig. 8: Median of the performance of algorithms based on the Kullback-Leibler and log-det α divergences on simulated data with respect to the noise parameter σ over 100 trials for $n = 32$ and $K = \{10, 50\}$. For criteria based on the log-det α-divergence ($\alpha = 0$ and -0.5), algorithms with the best performance are plotted (see figure 6). For the other criteria, all algorithms give similar performance. $*$: significative differences when performing a one-way anova ($p < 5.10^{-4}$) on the performance of all algorithms.

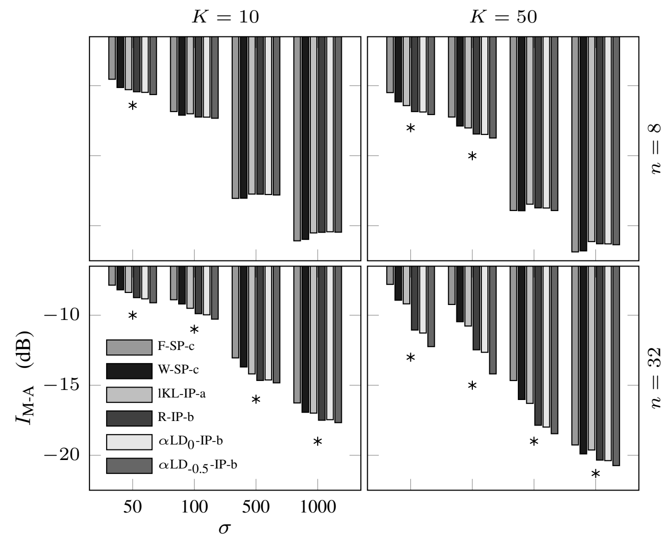

Fig. 9: Median of the performance of selected algorithms for most criteria on simulated data with respect to the noise parameter σ over 100 trials for $n = \{8, 32\}$ and $K = \{10, 50\}$. For the Kullback-Leibler and log-det α-divergences, we only plotted the left Kullback-Leibler and the log-det α-divergence for $\alpha = \{0, -0.5\}$ (see figure 8 for the others). The log-Euclidean distance is not represented here since its performance are close to those of the Riemannian distance. $*$: significative differences when performing a one-way anova ($p < 5.10^{-4}$) on the performance of all algorithms.

$\alpha = \{0, 0.5\}$.

In figure 9, we give the performance of algorithms corresponding to the Frobenius distance, the left and right Kullback-Leibler measures, the log-det α-divergence for $\alpha = 0$, and the natural Riemannian and Wasserstein distances. The log-Euclidean distance is not represented since it yields results very close to those of the natural Riemannian distance. Concerning the significance, we observe the same trend as in figure 8. When there are significant differences, the Frobenius distance always gives the worst results and the log-det α-divergence with $\alpha = -0.5$ always gives the best ones.

In summary, on data simulated according to model (45), the general Riemannian optimization framework we propose here gives results that are equivalent to those obtained by highly specific algorithms. Optimization on $\mathcal{SP}_n$ appears advantageous only when associated with update rule (c), while with other update rules, $\mathcal{IP}_n$ seems more robust. Finally, the performance of the different divergences is ordered, and it appears that it is more appropriate to use the log-det α-divergence with $\alpha = -0.5$. Note that these results are specific to model (45) and we cannot expect them to be transposable in general. However, the differences in the performance show that all methods are not equivalent and that one of them might be more appropriate depending on the data at hand.

B. Pseudo-real EEG data

Next, we study the performance of the AJD methods when performing the BSS of pseudo-real EEG data. These data are obtained by adding one mixed realistic simulated source signal to real EEG recordings. The major interest of using such data is that it allows to define objective and quantitative performance criteria for the recovery while remaining in a realistic framework. Indeed, one can measure how well the added synthetic source signal is separated from the real EEG by comparing the waveforms and the spatial mixing vectors of the estimated and true simulated source signals.

Concerning the real data, we use the continuously recording eyes-closed resting state EEG database from the Nova Tech EEG (NTE) database (see Chapter VII in [50] for a full description of these data). This database is composed of 84 subjects recorded with $n = 19$ electrodes (10-20 international system) at a sample rate of $f_s = 128$Hz. The signals are filtered between $2-32$Hz and 16 seconds ($T = 2048$ samples) of each recording are used here. The real data are denoted X_{real} (in $\mathbb{R}^{n\times T}$).

To build the synthetic database, we simulate signals using autoregressive model

$$\boldsymbol{s}(t) = 2r\cos\left(\frac{2\pi f}{f_s}\right)\boldsymbol{s}(t-1) - r^2\boldsymbol{s}(t-2) + m(t), \quad (47)$$

where f is the frequency, $r = 0.95$ defines the bandwidth and m is a gaussian noise with zero mean and standard deviation of 0.3. The synthetic source is defined as $\boldsymbol{s} = \widetilde{\boldsymbol{s}} + \widehat{\boldsymbol{s}}$ in $\mathbb{R}^{1\times T}$, where $\widetilde{\boldsymbol{s}}$ and $\widehat{\boldsymbol{s}}$ are simulated using (47) with $f = 13$Hz and $f = 21$Hz respectively. This source is projected on the eletrodes with a random mixing vector $\boldsymbol{a}$ in $\mathbb{R}^{n\times 1}$ to create the synthetic EEG $X_{\text{sim}} = \boldsymbol{a}\boldsymbol{s}$.

The BSS is performed on

$$X = \sigma\frac{X_{\text{sim}}}{\|X_{\text{sim}}\|_F} + \frac{X_{\text{real}}}{\|X_{\text{real}}\|_F}, \quad (48)$$

where σ defines the signal to noise ratio. The analysis is achieved by the AJD of Fourier cospectra estimated by 75% overlapping sliding windows of 1 second (Welch method) for frequencies 2-24Hz with 1Hz resolution. As explained in supplementary materials, an ad-hoc normalization of the input matrices is needed for methods based on criteria that are not invariant with respect to their rescaling. Here, we choose to normalize the trace of each cospectrum [11]. The weights w_k are defined by the non-diagonality measure of the matrices C_k proposed in [11]. As for the simulated data, we do a pre-whitening of the matrices C_k with the inverse square root of

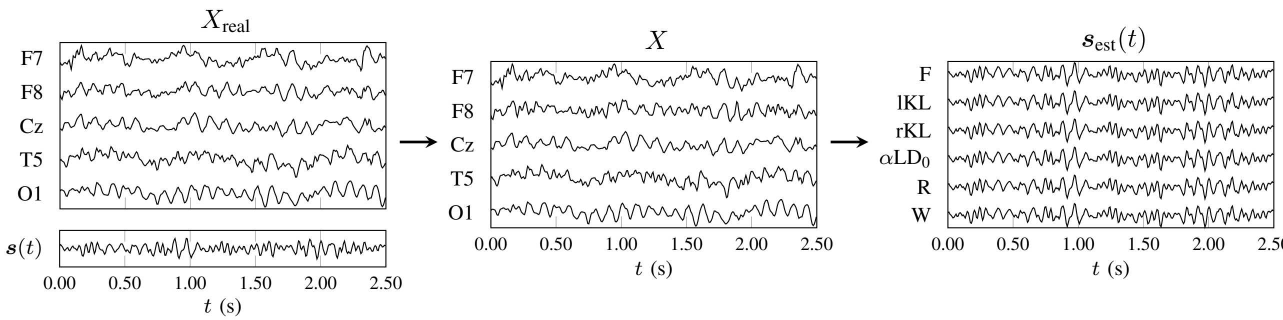

Fig. 10: Example of the BSS on subject 11 with $\sigma = 0.05$, for which the recovery of the synthetic source is particularly good, despite the low signal to noise ratio. Left: 2.5 seconds of the real EEG recording (top, data of only 5 of the 19 available electrodes are shown) and the synthetic source signal (bottom). Middle: 2.5 seconds of the mixing process of the real EEG recording and the synthetic source signal. Right: 2.5 seconds of the estimated source signal for different criteria.

their arithmatic mean and we initialize all the algorithms with the identity matrix. See figure 10 for an example of the BSS pipeline.

To estimate the performance of the algorithms, we use an index closely related to the Pearson correlation, which measures the linear dependance between two vectors. It is defined as

$$I_{\mathrm{P}}(\boldsymbol{u}, \boldsymbol{v}) = 1 - \frac{|\operatorname{cov}(\boldsymbol{u}, \boldsymbol{v})|}{\sigma_{\boldsymbol{u}} \sigma_{\boldsymbol{v}}}, \tag{49}$$

where $|\cdot|$ denotes the absolute value, $\operatorname{cov}(\cdot)$ returns the covariance of its arguments, $\sigma_{\boldsymbol{u}}$ and $\sigma_{\boldsymbol{v}}$ are the standard deviations of vectors $\boldsymbol{u}$ and $\boldsymbol{v}$. I_{P} gives a value in $[0, 1]$ with zero indicating perfect collinearity of the arguments. To determine the performance of the spatial recovery, we use the index $I_{\mathrm{P}}(\boldsymbol{a}, \boldsymbol{a}_{\text{est}})$, where $\boldsymbol{a}_{\text{est}}$ is the componnent of $A_{\text{est}} = B^{-1}$ best matching $\boldsymbol{a}$. We also measure the quality of the recovered waveform $\boldsymbol{s}_{\text{est}}$ in $S_{\text{est}} = BX$ with the index $I_{\mathrm{P}}(\boldsymbol{s}, \boldsymbol{s}_{\text{est}})$, where $\boldsymbol{s}_{\text{est}}$ is the source process best matching $\boldsymbol{s}$.

1) Riemannian optimization: Again, we start by checking that Riemannian optimization is appropriate for AJD by comparing the performance of the classical algorithms *NOJoB*, *uwedge* and *jadiag* with the ones of the corresponding Riemannian versions F-IP-a, F-SP-c and lKL-SP-a. The ANOVAs show that the differences in the performance of Riemannian and classical algorithms are not significant (data not shown). This confirms that Riemannian optimization is a viable solution for AJD and BSS.

2) Constraints and AJD optimization algorithms: Then, we study the influence of the constraints and optimization strategies. On these pseudo-real data, we only observe significant differences for the Wasserstein distance (figure 11). The ANOVAs show that significant differences are found between algorithms based on the Wasserstein distance only for the waveform recovery index and $\sigma = \{0.1, 0.2\}$. Post-hoc comparisons reveal that W-SP-a displays better performance as compared to the other algorithms.

3) Criteria: Finally, we look at the influence of the choice of the divergence and target matrices. The ANOVAs reveal no significant differences and a high variability in the results is observed (data not shown). Depending on the data, it is not always the same criterion that performs the best. This is likely due to high inter-variability of the individual background EEG.

In summary, the differences observed on the pseudo-real data do not match those we havve found on the simulated data. However, we still have that all approaches are not equivalent when looking at subjects separately. Thus, practionners should try different possibilities in order to determine what criterion is more appropriate for their data. The Riemannian framework we propose in this paper can be convenient for this testing.

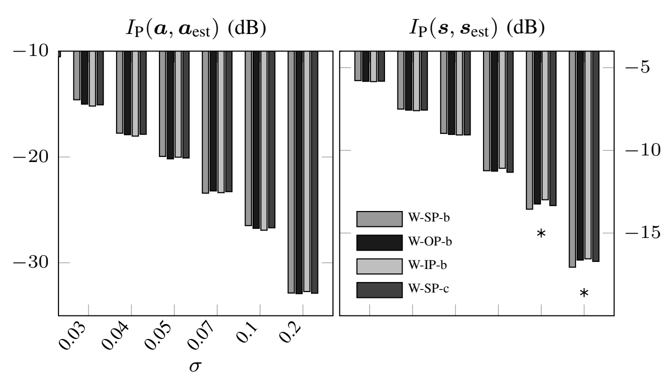

Fig. 11: Median of the spatial mixing (left) and waveform (right) recovery performance of algorithms based on the Wasserstein distance over 84 pseudo-real EEG datasets with respect to the noise parameter σ. $*$: significative differences when performing a one-way anova ($p < 5.10^{-4}$) on the performance of all algorithms.

V. Conclusions and perspectives

In this article, we developed the first complete Riemannian optimization framework suited for BSS and AJD handling three classically used constraints. It can be used providing only the Euclidean gradient and possibly Hessian of the objective function of interest. Furthermore, Riemannian optimization offers a large panel of general optimization algorithms [24] and research on this topic is very active, see *e.g.* [51], [52]. Besides, we also provide a thorough study of the AJD problem of SPD matrices from an information geometry point of view subsuming previous research on this topic, and bringing new insights and original methods based on criteria that have not been considered before. Finally, our numerical experiments show that Riemannian optimization yields equivalent results as compared to classical AJD algorithms. The interest of Riemannian optimization resides in the modularity and flexibility it offers: we proposed a unified framework allowing to design new BSS methods and to compare with others, in a simple manner by just specifying the criteria and constraints in problem (8). The experiments suggest that there is not a best AJD criterion in general and that the different AJD criteria are not always equivalent. In this sense, our analysis brings new information, while further specific studies are needed to understand what AJD criterion should be prefered depending on the data at hand.

This work opens new perspectives on the different topics we have investigated. Concerning the Riemannian optimization framework, we first notice that the generalization to complex or rectangular matrices is straightforward. Moreover, we have used the polar decomposition, however other matrix factorizations may be investigated since they possess different properties as shown in [53]. We also believe that the choice of metric (11) is crucial and that different possibilities must be considered. Indeed, the *natural gradient* approach [7], [17], [18], [22] is based on endowing GL_n with a Riemannian metric adapted to BSS and *information theory* [17]. A well chosen metric can also allow to embed the *non-holonomic constraint* [7], [15], [21], [22] in a Riemannian quotient manifold [24], which has never been done properly as far as we know.

AJD criteria are characterized by the choice of a divergence and target diagonal matrices. It remains to better understand the respective influence of these two constituents and other associations besides those we have considered here may yield useful results. This might help explaining the differences between the criteria observed in the results and finding a way to combine the different criteria in order to increase the robustness and accuracy of the results. Another direction for further investigation concerns the porperties of the AJD criteria. Further research is needed to analyze all desirable properties and their importance for AJD. As pointed out in [9], there are links between AJD and centers of mass of SPD matrices. This latter topic is a well studied field [27]–[32] and studying these links can lead to a better understanding of the AJD problem. Furthermore, developing methods that simulataneously solve the AJD problem and find a center of mass of a set of SPD matrices might yield more robust and accurate results for both problems.

Finally, this work can be generalized to handle models beyond linear BSS. Indeed, both the Riemannian optimization framework and the information geometry approach can be adapted to *independent vector analysis* [54], *joint BSS* [34], *joint independent subspace analysis* [55] or *bilinear BSS* [56]. We will investigate the perspectives above and these generalizations in future research.

Acknowledgment

The authors would like to thank Octave Curmi and Ronald Phlypo for their help on the Riemannian geometry framework, Bijan Afsari for the helpful comments on Riemannian geometry and AJD, and Pedro Rodrigues for the autoregressive model. This work has been 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 CHESS 2012-ERC-AdG-320684.

References

[1] C. Jutten and J. Herault. Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal processing*, 24(1):1–10, 1991.

[2] P. Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

[3] P. Comon and C. Jutten. *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. Academic Press, 1st edition, 2010.

[4] J.-F. Cardoso and A. Souloumiac. Blind beamforming for non Gaussian signals. *IEEE Proceedings-F*, 140(6):362–370, dec 1993.

[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] B. Afsari. Sensitivity analysis for the problem of matrix joint diagonalization. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1148–1171, 2008.

[8] 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*, 2016.

[9] M. Congedo, B. Afsari, A. Barachant, and M. Moakher. Approximate joint diagonalization and geometric mean of symmetric positive definite matrices. *PLoS ONE*, 10(4):e0121423, 04 2015.

[10] D.-T. Pham and J.-F. Cardoso. Blind separation of instantaneous mixtures of nonstationary sources. *IEEE Transactions on Signal Processing*, 49(9):1837–1848, 2001.

[11] 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.

[12] A. Yeredor. Non-orthogonal joint diagonalization in the least-squares sense with application in blind source separation. *IEEE Transactions on Signal Processing*, 50(7):1545–1553, Jul 2002.

[13] M. Congedo and D.-T. Pham. Least-squares joint diagonalization of a matrix set by a congruence transformation. In *SinFra'09 - 2nd Singaporean-French IPAL Symposium*, pages 96–106, Feb 2009.

[14] P. Tichavskỳ and A. Yeredor. Fast approximate joint diagonalization incorporating weight matrices. *Signal Processing, IEEE Transactions on*, 57(3):878–891, 2009.

[15] A. Ziehe, P. Laskov, G. Nolte, and K.-R. Müller. A fast algorithm for joint diagonalization with non-orthogonal transformations and its application to blind source separation. *The Journal of Machine Learning Research*, 5:777–800, 2004.

[16] D.-T. Pham and M. Congedo. Least square joint diagonalization of matrices under an intrinsic scale constraint. In *Independent Component Analysis and Signal Separation*, pages 298–305. Springer, 2009.

[17] S.-I. Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

[18] A. Yeredor, A. Ziehe, and K.-R. Müller. Approximate joint diagonalization using a natural gradient approach. In *Independent Component Analysis and Blind Signal Separation*, pages 89–96. Springer, 2004.

[19] A.-J. van der Veen. Joint diagonalization via subspace fitting techniques. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, volume 5, pages 2773–2776, 2001.

[20] M. Joho, R. H. Lambert, and H. Mathis. Elementary cost functions for blind separation of non-stationary source signals. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, volume 5, pages 2793–2796, 2001.

[21] S.-I. Amari, T. Chen, and A. Cichocki. Nonholonomic orthogonal learning algorithms for blind source separation. *Neural computation*, 12(6):1463–1484, 2000.

[22] B. Afsari and P. S. Krishnaprasad. Some gradient based joint diagonalization methods for ICA. In *Independent Component Analysis and Blind Signal Separation*, pages 437–444. Springer, 2004.

[23] S. Dégerine and E. Kane. A comparative study of approximate joint diagonalization algorithms for blind source separation in presence of additive noise. *Signal Processing, IEEE Transactions on*, 55(6):3022–3031, 2007.

[24] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, USA, 2008.

[25] P.-A. Absil and K. A. Gallivan. Joint diagonalization on the oblique manifold for independent component analysis. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 5, pages 945–948, May 2006.

[26] 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.

[27] R. Bhatia. *Positive definite matrices*. Princeton University Press, 2009.

[28] 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.

[29] Z. Chebbi and M. Moakher. Means of Hermitian positive-definite matrices based on the log-determinant α-divergence function. *Linear Algebra and its Applications*, 436(7):1872–1889, 2012.

[30] S. Sra. Positive definite matrices and the s-divergence. *arXiv preprint arXiv:1110.1773*, 2013.

[31] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM journal on matrix analysis and applications*, 29(1):328–347, 2007.

[32] R. Bhatia, T. Jain, and Y. Lim. On the Bures-Wasserstein distance between positive definite matrices. *preprint*, 2017.

[33] F. Bouchard, J. Malick, and M. Congedo. Approximate joint diagonalization according to the natural Riemannian distance. In *13th International Conference on Latent Variable Analysis and Signal Separation*, pages 290–299. Springer, 2017.

[34] M. Congedo, R. Phlypo, and J. Chatel-Goldman. Orthogonal and non-orthogonal joint blind source separation in the least-squares sense. In *20th European Signal Processing Conference (EUSIPCO-2012)*, pages 1885–1889, 2012.

[35] P.-A. Absil and J. Malick. Projection-like retractions on matrix manifolds. *SIAM Journal on Optimization*, 22(1):135–158, 2012.

[36] P. Fillard, V. Arsigny, N. Ayache, and X. Pennec. A Riemannian framework for the processing of tensor-valued images. In *Deep Structure, Singularities, and Computer Vision*, pages 112–123. Springer, 2005.

[37] A. Goh and R. Vidal. Unsupervised Riemannian clustering of probability density functions. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 377–392. Springer, 2008.

[38] L. T. Skovgaard. A Riemannian geometry of the multivariate normal model. *Scandinavian Journal of Statistics*, pages 211–223, 1984.

[39] C. R. Rao. Information and accuracy attainable in the estimation of statistical parameters. *Bull. Calcutta Math. Soc*, 37(3):81–91, 1945.

[40] S.-I. Amari. *Differential-geometrical methods in statistics*. Springer, Heidelberg, 1985.

[41] D. C. Dowson and B. V. Landau. The Fréchet distance between multivariate normal distributions. *Journal of multivariate analysis*, 12(3):450–455, 1982.

[42] I. Olkin and F. Pukelsheim. The distance between two random vectors with given dispersion matrices. *Linear Algebra and its Applications*, 48:257–263, 1982.

[43] C. Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[44] A. Takatsu. On wasserstein geometry of the space of gaussian measures. *arXiv preprint arXiv:0801.2250*, 2009.

[45] L. Ning, X. Jiang, and T. T. Georgiou. Geometric methods for estimation of structured covariances. *arXiv preprint arXiv:1110.3695*, 2011.

[46] X. Jiang, Z.-Q. Luo, and T. T. Georgiou. Geometric methods for spectral analysis. *IEEE Transactions on Signal Processing*, 60(3):1064–1074, 2012.

[47] 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.

[48] M. Congedo, A. Barachant, and E. K. Koopaei. Fixed point algorithms for estimating power means of positive definite matrices. *IEEE Transactions on Signal Processing*, 65(9):2211–2220, May 2017.

[49] 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.

[50] M. Congedo. *EEG Source Analysis*. CNRS, University of Grenoble Alpes, Grenoble Institute of Technology, 2013.

[51] X. Yuan, W. Huang, P.-A. Absil, and K. A. Gallivan. A Riemannian limited-memory BFGS algorithm for computing the matrix geometric mean. *Procedia Computer Science*, 80:2147–2157, 2016.

[52] H. Zhang, S. J. Reddi, and S. Sra. Riemannian SVRG: Fast stochastic optimization on Riemannian manifolds. In *Advances in Neural Information Processing Systems*, pages 4592–4600, 2016.

[53] R. Bhatia. Matrix factorizations and their perturbations. *Linear Algebra and its applications*, 197:245–276, 1994.

[54] T. Adali, M. Anderson, and G.-S. Fu. Diversity in independent component and vector analyses: Identifiability, algorithms, and applications in medical imaging. *IEEE Signal Processing Magazine*, 31(3):18–33, 2014.

[55] D. Lahat and C. Jutten. Joint independent subspace analysis using second-order statistics. *IEEE Transactions on Signal Processing*, 64(18):4891–4904, 2016.

[56] M. Congedo, R. Phlypo, and D. T. Pham. Approximate joint singular value decomposition of an asymmetric rectangular matrix set. *IEEE Transactions on Signal Processing*, 59(1):415–424, Jan 2011.