Riemannian Optimization and Approximate Joint Diagonalization for Blind Source Separation
Florent Bouchard, Jérôme Malick, Marco Congedo

To cite this version:
Florent Bouchard, Jérôme Malick, Marco Congedo. Riemannian Optimization and Approximate Joint Diagonalization for Blind Source Separation. IEEE Transactions on Signal Processing, 2018, 66 (8), pp.2041-2054. 10.1109/TSP.2018.2795539. hal-01875788

HAL Id: hal-01875788
https://hal.science/hal-01875788
Submitted on 15 Oct 2018

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.
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 definite (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

\[ x(t) = As(t), \]

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 pre-whitening 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 arbitrarit.

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 developed in the pioneering article [5]. Given a set \( \{C_{k}\} \) of \( K \) symmetric matrices, we suppose that for all \( 1 \leq k \leq K \)

\[ C_{k} = A \Lambda_{k} A^{T} + N_{k}, \]

where matrices \( \Lambda_{k} \) 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 \( \{BC_{k}B^{T}\} \). 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

\[ f_{F}(B) = \sum_{k} w_{k} \left| \left| BC_{k}B^{T} - \text{ddiag}(BC_{k}B^{T}) \right| \right|_{F}^{2}, \] (3)

where \( w_{k} \) are positive weights, \( \left| \left| \cdot \right| \right|_{F} \) denotes the Frobenius norm and \( \text{ddiag}(\cdot) \) 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

\[ f_{\text{KL}}(B) = \sum_{k} w_{k} \log \frac{\det(\text{ddiag}(BC_{k}B^{T}))}{\det(BC_{k}B^{T})}, \] (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

$$\det(BB^T) = 1.$$  

(5)

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

$$d\text{diag}(BB^T) = I_n,$$ 

(6)

where $I_n$ is the identity matrix. Note that in [14], [23], the constraint is $d\text{diag}(BB^T) = I_n$ 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 constraint, it was introduced in [13], [16] and can be written as

$$\sum_{k=1}^K d\text{diag}(BC_kB^T)^2 = I_n.$$ 

(7)

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

$$\arg\min_{B \in \mathcal{M} \subset \text{GL}_n} f(B).$$ 

(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 $\text{GL}_n$ 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 reasoning 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

$$f(B) = \sum_k w_k d(BC_kB^T, \Lambda_k(B)),$$ 

(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 $\Lambda_k(B) = d\text{diag}(BC_kB^T)$. 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 $\alpha$-divergence, log-Euclidean, natural Riemannian and Wasserstein distances. In previous research, only the Frobenius distance, left Kullback-Leibler measure and log-det $\alpha$-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 algo-
rithms 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 AID problem.

Finally, section V draws several conclusions and perspectives. We discuss possible extensions of the Riemannian optimization framework and perspectives for future AID research, especially on the need to better understand the links between AID 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 readability, 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 constraints (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 $\xi_i$ in the tangent space $T_{B_i}\mathcal{M}$ 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 $\xi_i$ at $B_i$. The descent direction $\xi_i$ 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), (6) and (7) in section II-B. For better readability, 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 $S \in S_{++}^n$ (set of SPD matrices) and $U \in O_n$ (set of orthogonal matrices) such that $B = SU$. It follows that GL$_n$ is “equivalent” to the product manifold $\mathcal{P}_n = S_{++}^n \times O_n$, which we call the polar manifold. More precisely, $\mathcal{P}_n$ is diffeomorphic to GL$_n$ with the mapping $\pi: \mathcal{P}_n \rightarrow$ GL$_n$ such that $\pi(S,U) = SU$. In the following, $B$ denotes a couple $(S,U)$ and $B$ denotes the corresponding matrix $\pi(B)$. 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 $f_{\pi B}$ or $f_{\pi^{-1} B}$).

The geometry of $\mathcal{P}_n$ follows from those of $S_{++}^n$ and $O_n$, 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 $T_{B}\mathcal{P}_n$ of $\mathcal{P}_n$ at $B = (S,U)$ is $T_B S_{++}^n \times T_B O_n$, i.e.,

$$T_B\mathcal{P}_n = \{((\xi_S,\xi_U) \in \mathcal{E}_n: \xi_S^T = \xi_S, U \xi_U^T + \xi_U U^T = 0)\} \quad (10)$$

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

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

where $\text{tr}(\cdot)$ 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 $P_B$ to the metric (11) defined from $\mathcal{E}_n$ onto $T_B\mathcal{P}_n$ such that, for $Z = (Z_S,Z_U)$,

$$P_B(Z) = (\text{sym}(Z_S), Z_U - U\text{sym}(U^T Z_U)), \quad (12)$$

where $\text{sym}(\cdot)$ 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 $P_B(Z)$ respectively (the same type of notations are used for gradients, Hessians and retractions). One then obtain the Riemannian gradient and Hessian in $T_B\mathcal{P}_n$ of $f$ at $B$ in $\mathcal{P}_n$, from the Euclidean ones in $\mathcal{E}_n$ with expressions

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

and

$$\text{Hess}_{\mathcal{P}_n} f(S)[\xi] = SP_S(\text{Hess}_{\mathcal{E}_n} f(S)[\xi])S$$

$$+ P_S(\xi_S P_S(\text{grad}_{\mathcal{E}_n} f(S)))S,$$

$$\quad \text{Hess}_{\mathcal{P}_n} f(U)[\xi] = P_U(\text{Hess}_{\mathcal{E}_n} f(U)[\xi])$$

$$- P_U(\xi_U U^T \text{sym}(U^T \text{grad}_{\mathcal{E}_n} f(U))). \quad (14)$$

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 $\text{grad}_{\text{GL}} f$ and $\text{Hess}_{\text{GL}} f$:
Proposition 1. Let $B = (S, U)$ and $\xi = (\xi_S, \xi_U)$. Then,
\[
\text{grad}_{\xi_B} f(B) = (\text{grad}_{GL} f(\pi(B)) U^T, S \text{grad}_{GL} f(\pi(B))).
\]
\[
\text{Hess}_{\xi_B} f(S)[\xi] = \text{Hess}_{GL} f(\pi(B)) [\xi_S U + S \xi_U] U^T + \text{grad}_{GL} f(\pi(B)) \xi_U^T,
\]
\[
\text{Hess}_{\xi_B} f(U)[\xi] = S \text{Hess}_{GL} f(\pi(B)) [\xi_S U + S \xi_U] + \xi_S \text{grad}_{GL} f(\pi(B)).
\]

Finally a retraction $R_B$ on $\mathcal{P}_n$ at $B$ is properly defined by the exponential map on $\mathcal{P}_n$, which is given, for all $\xi = (\xi_S, \xi_U)$ in $T_B\mathcal{P}_n$, by
\[
R_B(\xi) = \exp_B^n(\xi) = \left(\exp^{S^{++}}_S(\xi_S), \exp^{O^2}_U(\xi_U)\right),
\]
where $\exp^{S^{++}}_S(\xi_S) = S^{1/2} \exp(S^{-1/2} \xi_S S^{-1/2}) S^{1/2}$ is the exponential map on $S^{++}_n$ ($\exp(\cdot)$ is the standard matrix exponential) and $\exp^{O^2}_U(\xi_U)$ is the exponential map on $O_n$, 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 $\det(\pi(B) \pi(B)^T) = 1$. It leads to the subspace of $\mathcal{P}_n$
\[
SP_n = \{(S, U) \in \mathcal{P}_n : \det(S) = 1\},
\]
which we call the special polar manifold. Constraint (6) becomes $d\text{diag}(\pi(B) \pi(B)^T) = I_n$ in $\mathcal{P}_n$. It leads to the subspace of $\mathcal{P}_n$
\[
\mathcal{O}P_n = \{(S, U) \in \mathcal{P}_n : d\text{diag}(S^2) = I_n\},
\]
which we name the oblique polar manifold. Finally, constraint (7) leads to the subspace
\[
\mathcal{T}P_n = \left\{(S, U) \in \mathcal{P}_n : \sum_k d\text{diag}(BC_k B^T)^2 = I_n\right\},
\]
which we refer to as the intrinsic polar manifold. In proposition 2, we will get the first ingredient needed for Riemannian optimization, i.e. we define the Riemannian geometry of these three subspaces.

Proposition 2. Subspaces $SP_n$, $\mathcal{O}P_n$ and $\mathcal{T}P_n$ are connected submanifolds of $\mathcal{P}_n$. The tangent spaces of $SP_n$, $\mathcal{O}P_n$ and $\mathcal{T}P_n$ at $B$ are subspaces of $T_B\mathcal{P}_n$ characterized for $\xi = (\xi_S, \xi_U)$ by equations
\[
\begin{array}{l}
SP_n: \text{tr}(S^{-1} \xi_S) = 0 \\
\mathcal{O}P_n: d\text{diag}(S \xi_S) = 0 \\
\mathcal{T}P_n: d\text{diag}(B Q) = 0
\end{array}
\]
where $\dot{B} = \xi_S U + S \xi_U$ is the derivative of $B = \pi(B)$ in direction $\xi$ and $Q = \sum_k C_k B^T d\text{diag}(B C_k B^T)$. Finally, manifolds $SP_n$, $\mathcal{O}P_n$ and $\mathcal{T}P_n$ 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 $SP_n$, $\mathcal{O}P_n$ and $\mathcal{T}P_n$ 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 manifold $\mathcal{E}_n$ with (12).

Lemma 1. The projection maps $P_B^S$, $P_B^O$ and $P_B^T$ on $T_BSP_n$, $T_B\mathcal{O}P_n$ and $T_B\mathcal{T}P_n$ are respectively given, for $Z = (Z_S, Z_U)$ in $T_B\mathcal{P}_n$, by
\[
\begin{align*}
SP_n: & \quad (Z_S - \frac{1}{n} \text{tr}(S^{-1} Z_S) S, Z_U) \\
\mathcal{O}P_n: & \quad (Z_S - P_S(S^2 \Delta_O S), Z_U) \\
\mathcal{T}P_n: & \quad (Z_S - SP_S(U Q \Delta_T) S, Z_U - P_U(S \Delta_T Q^T))
\end{align*}
\]
where $\Delta_O$ is the unique diagonal matrix solution to $d\text{diag}(P_S(S^2 \Delta_O S)) S = d\text{diag}(Z_S S)$ and $\Delta_T$ is the unique diagonal matrix solution to $d\text{diag}(SP_S(U Q \Delta_T) B + P_U(S \Delta_T Q^T)) Q)
\]
\[
= d\text{diag}((S Z_S + Z_S U) Q) - W_T),
\]
where $W_T = P_S(S^2 \Delta_O \xi_S + (\xi_S S + S \xi_S) \Delta_O S) S + P_S(S^2 \Delta_O S) \xi_S$.

Lemma 2. The derivative $\Delta_O$ of $\Delta_O$ defined in (19) at $S$ in the direction $\xi_S$ is the unique diagonal matrix solution to $d\text{diag}(P_S(S^2 \Delta_O S)) S = d\text{diag}(Z_S \xi_S - W_O)$ where
\[
W_O = P_S(S^2 \Delta_O \xi_S + (\xi_S S + S \xi_S) \Delta_O S) S + P_S(S^2 \Delta_O S) \xi_S.
\]

Lemma 3. The derivative $\Delta_T$ of $\Delta_T$ defined in (20) at $B$ in the direction $\xi = (\xi_S, \xi_U)$ is the unique diagonal matrix solution to
\[
d\text{diag}(SP_S(U Q \Delta_T) B + P_U(S \Delta_T Q^T)) Q) = \text{diag}((S Z_S + Z_S U) Q + (\xi_S Z_S + Z_S \xi_U) Q - W_T),
\]
where $W_T = \xi_S E Q + S H Q + S E \dot{Q}$,
\[
E = P_S(U Q \Delta_T) B + P_U(S \Delta_T Q^T),
\]
\[
H = \frac{1}{2} (\xi_S \Delta_T Q^T + S \Delta_T \dot{Q}^T) + \frac{\Delta_T}{2} (\dot{Q}^T U^T B + Q^T \xi_U B + Q^T U^T \dot{B}).
\]

This leads to proposition 3 where the Riemannian gradients and Hessians of $f$ in $SP_n$, $\mathcal{O}P_n$ and $\mathcal{T}P_n$ 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 $\mathcal{GL}_n$ by applying proposition 1, (13) and (14) first.
The following functions or \( \tilde{\xi} \) are defined for retractions at \( \frac{\partial}{\partial\xi} \) with (19) of the criteria in GL matrices, given classical (Euclidean) gradients and Hessians \( \tilde{\Delta} \), which corresponds to a path on \( S_{n}^{++} \) (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 \( B C_{k} B^{T} \) according to \( d(\cdot, \cdot) \), which are defined as

\[
\Lambda_{k}(B) = \arg\min_{\Lambda \in \mathcal{D}_{++}} d(B C_{k} B^{T}, \Lambda).
\]  

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 \( \alpha \)-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 beginning 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 \( \Lambda_{k} \).
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 $S^{++}_n$, initial guess $B_0$ for $B$

**Output:** iterates $B_i$ of the estimated joint diagonalizer

1. Compute matrices $\{B_0 C_k B_0^T\}$ and set $i = 0$.
2. **while** not convergence
   - Obtain $B_{i+1}$ through update rule (a), (b) or (c) on $B_i$.
3. **Update rule (a):** direct strategy
   - Obtain a descent direction $\xi_i$ from the gradient and Hessian of (9) at $B_i$.
   - Compute $B_{i+1}$ through the retraction of $\xi_i$ at $B_i$.
4. **Update rule (b):** indirect strategy
   - Obtain a descent direction $\xi_i$ from the gradient and Hessian of (9) at $B_i$.
   - Compute $B_{i+1}$ through the retraction of $\xi_i$ at $B_i$.
5. **Update rule (c):** inverse indirect strategy
   - Compute matrices $\Lambda_k$.
   - Obtain a descent direction $\xi_i$ from the gradient and Hessian of (9) at $B_i$.
   - Compute $B_{i+1}$ through the retraction of $\xi_i$ at $B_i$.

![Fig. 3: Schematic illustration of the different optimization strategies in algorithm 1.](image)

(a) (b) (c)

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 $B \rightarrow \Lambda_k(B)$ 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

$$\tilde{f}(A) = \sum_k w_k d(BC_k B^T, A\Lambda_k A^T).$$  \quad (26)

Starting from $A_0 = I_n$, we perform one step in a descent direction of $\tilde{f}$ to obtain $A$ and update $B$ such as $B \leftarrow A^{-1} B$. Again, matrices $BC_k B^T$ get closer to $\Lambda_k(B)$ 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

$$\tilde{f}(A) = \sum_k w_k d(BC_k B^T, A\Lambda_k A^T).$$  \quad (25)

This leads to update rule (b) of algorithm 1. As illustrated in figure 3, at each iteration matrices $BC_k B^T$ 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

$$\tilde{f}(A) = \sum_k w_k d(BC_k B^T, A\Lambda_k A^T).$$  \quad (26)

Starting from $A_0 = I_n$, we perform one step in a descent direction of $\tilde{f}$ to obtain $A$ and update $B$ such as $B \leftarrow A^{-1} B$. Again, matrices $BC_k B^T$ get closer to $\Lambda_k(B)$ without taking into account how target matrices $\Lambda_k(B)$ are modified.

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

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

$$\tilde{f}(A) = \sum_k w_k d(BC_k B^T, A\Lambda_k A^T).$$  \quad (26)

Starting from $A_0 = I_n$, we perform one step in a descent direction of $\tilde{f}$ to obtain $A$ and update $B$ such as $B \leftarrow A^{-1} B$. Again, matrices $BC_k B^T$ get closer to $\Lambda_k(B)$ without taking into account how target matrices $\Lambda_k(B)$ are modified.

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

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

$$\tilde{f}(A) = \sum_k w_k d(BC_k B^T, A\Lambda_k A^T).$$  \quad (26)

Starting from $A_0 = I_n$, we perform one step in a descent direction of $\tilde{f}$ to obtain $A$ and update $B$ such as $B \leftarrow A^{-1} B$. Again, matrices $BC_k B^T$ get closer to $\Lambda_k(B)$ without taking into account how target matrices $\Lambda_k(B)$ are modified.

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

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

$$\tilde{f}(A) = \sum_k w_k d(BC_k B^T, A\Lambda_k A^T).$$  \quad (26)

Starting from $A_0 = I_n$, we perform one step in a descent direction of $\tilde{f}$ to obtain $A$ and update $B$ such as $B \leftarrow A^{-1} B$. Again, matrices $BC_k B^T$ get closer to $\Lambda_k(B)$ without taking into account how target matrices $\Lambda_k(B)$ are modified.

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

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

$$\tilde{f}(A) = \sum_k w_k d(BC_k B^T, A\Lambda_k A^T).$$  \quad (26)

Starting from $A_0 = I_n$, we perform one step in a descent direction of $\tilde{f}$ to obtain $A$ and update $B$ such as $B \leftarrow A^{-1} B$. Again, matrices $BC_k B^T$ get closer to $\Lambda_k(B)$ without taking into account how target matrices $\Lambda_k(B)$ are modified.

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

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

$$\tilde{f}(A) = \sum_k w_k d(BC_k B^T, A\Lambda_k A^T).$$  \quad (26)

Starting from $A_0 = I_n$, we perform one step in a descent direction of $\tilde{f}$ to obtain $A$ and update $B$ such as $B \leftarrow A^{-1} B$. Again, matrices $BC_k B^T$ get closer to $\Lambda_k(B)$ without taking into account how target matrices $\Lambda_k(B)$ are modified.

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

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

$$\tilde{f}(A) = \sum_k w_k d(BC_k B^T, A\Lambda_k A^T).$$  \quad (26)

Starting from $A_0 = I_n$, we perform one step in a descent direction of $\tilde{f}$ to obtain $A$ and update $B$ such as $B \leftarrow A^{-1} B$. Again, matrices $BC_k B^T$ get closer to $\Lambda_k(B)$ without taking into account how target matrices $\Lambda_k(B)$ are modified.
different (local) minima. However, by looking at the gradients of the criterion 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 readability, Euclidean gradients and Hessians of considered criteria are given in supplementary materials.

1) **Frobenius distance:** This distance is defined as

$$d_F^2(M, \Lambda) = \|M - \Lambda\|^2_F. \quad (27)$$

This is the distance on $S_{++}^n$ 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 $\Lambda = \text{ddiag}(M)$. 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 $f_1$ and defined for $\Lambda_k(B) = \text{ddiag}(BC_kB^T)$. For update rule (b), the criterion corresponding to (25) is denoted $f_B$. Finally, the criterion corresponding to (26) is denoted $f_2$ 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

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

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 $d_{KL}(M, \Lambda) = d_{KL}(M, \Lambda)$, the closest diagonal matrix to $M$ according to this measure is, again, $\Lambda = \text{ddiag}(M)$ [8]. Choosing $\Lambda_k(B) = \text{ddiag}(BC_kB^T)$ yields the historical log likelihood criterion given in (4) and denoted $f_{KL}$. 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 criterion will no longer depend on the data but only on the variable $B$ or $A$.

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

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

$$d_{KL}(M, \Lambda, \Lambda) = \frac{1}{2}(d_{KL}(M, \Lambda) + d_{KL}(M, \Lambda)). \quad (29)$$

In this case, the closest diagonal matrix to $M$ is $\Lambda = \text{ddiag}(M)^{1/2} \text{ddiag}(M^{-1})^{-1/2}$ [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 $f_{KL}$ and defined for

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

The criteria corresponding to (25) and (26) are denoted $\tilde{f}_{KL}$ and $\tilde{f}_{KL}$, respectively.

3) **Log-det $\alpha$-divergence:** This divergence [29] is defined as

$$d_{\alpha,LD}(M, \Lambda) = \frac{4}{1 - \alpha^2} \log \det\left(\frac{1 - \alpha}{2} M + \frac{1 + \alpha}{2} \Lambda\right) \det(M)^{1/2} \det(\Lambda)^{1/2}. \quad (30)$$

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 $\alpha$ and when $\alpha \to -1$ and $\alpha \to 1$, it coincides with the right and left Kullback-Leibler measures presented previously. Furthermore, $\alpha=0$ 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 $S_{++}^n$ while being numerically cheaper [30]. As shown in [8], the closest diagonal matrix to $M$ according to this measure is the unique $\Lambda$ solution to

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

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 $\tilde{f}_{\alpha,LD}$ and $\tilde{f}_{\alpha,LD}$ corresponding to (25) and (26), respectively.

4) **Natural Riemannian distance:** As shown in [27], the natural Riemannian distance on $S_{++}^n$ is

$$d_R^2(M, \Lambda) = \left\|\log((\Lambda^{-1/2} M \Lambda^{-1/2})\right\|^2_F. \quad (32)$$

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 $\Lambda$ solution to

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

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 $f_R$ and $f_R$ corresponding to (25) and (26), respectively.
5) Log-Euclidean distance: This distance [31], [36] is given by
\[ \delta_{LE}(M, \Lambda) = \| \log(M) - \log(\Lambda) \|_F^2. \] (34)

If \( M \) and \( \Lambda \) 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 \( \Lambda \) 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 \( \Lambda = \exp(\text{ddiag}(\log(M))) \). It yields the criterion \( f_{LE} \) corresponding to \( (9) \) defined for \( \Lambda_k(B) = \exp(\text{ddiag}(\log(BC_kB^T))) \). 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 \( \tilde{f}_{LE} \) and \( \bar{f}_{LE} \), respectively.

6) Wasserstein distance: This distance is given by
\[ \delta_W(M, \Lambda) = \text{tr} \left( \frac{1}{2} (M + \Lambda) - (\Lambda^{1/2} M \Lambda^{1/2})^{1/2} \right). \] (35)

This distance has the major advantage of being defined when matrices \( M \) and \( \Lambda \) 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 \( S_n^{++} \), which is fully described in [32]. The closest diagonal matrix to \( M \) is \( \Lambda \) solution to
\[ \text{ddiag}(\Lambda^{1/2} M \Lambda^{1/2})^{1/2} = \Lambda. \] (36)

The proof of this result along with a method to find \( \Lambda \) are in supplementary materials. Again, we only consider update rules (b) and (c) and criteria corresponding to (25) and (26) are denoted \( \tilde{f}_W \) and \( \bar{f}_W \).

C. AID properties

By studying the AID problem, we can infer several desirable properties for the AID 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 \( \Sigma B \) are equivalent solutions for any non singular diagonal matrix \( \Sigma \) as explained in section I. Therefore, it is desirable that the criterion \( f \) takes the same values at \( B \) and \( \Sigma B \), i.e., that it satisfies the following

**Property 1.** The criterion \( f \) is said to be invariant by diagonal scaling if, for \( B \) in \( \text{GL}_n \), and any non singular diagonal matrix \( \Sigma \), we have
\[ f(B) = f(\Sigma B). \] (37)
Assuming that \( \Lambda_k(\Sigma B) = \Sigma \Lambda_k(B) \Sigma \), criteria with form (9) possess this property if the divergence satisfies
\[ d(\Sigma M \Sigma, \Lambda \Sigma^2) = d(M, \Lambda), \] (38)
for \( M \) in \( S_n^{++} \), \( \Lambda \) in \( D_n^{++} \) and a non singular diagonal matrix \( \Sigma \).

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 scaling [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 \( \text{GL}_n \) and any strictly positive scalars \( a_k \), we have
\[ f_{(a_k C_k)}(B) = f_{(C_k)}(B). \] (39)
Assuming that \( \Lambda_{a_k C_k}(B) = a_k \Lambda_{C_k}(B) \), (9) possesses this property if we have
\[ d(aM, a\Lambda) = d(M, \Lambda), \] (40)
for \( M \) in \( S_n^{++} \), \( \Lambda \) in \( D_n^{++} \) 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 \( \text{GL}_n \), we have
\[ f_{(C_k)}(B) = f_{(C_k^{-1})}(B^{-T}). \] (41)
For (9) to satisfy this property, we need to have
\[ d(M^{-1}, \Lambda^{-1}) = d(M, \Lambda), \] (42)
and \( \Lambda_{C_k}(B) = \Lambda_{C_k^{-1}}(B^{-T})^{-1}. \) (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 \( \text{GL}_n \), 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 \( \text{GL}_n \), we have
\[ f_{(WC_kW^T)}(B) = f_{(C_k)}(BW). \] (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.

<table>
<thead>
<tr>
<th>F</th>
<th>IKL</th>
<th>rKL</th>
<th>sKL</th>
<th>αLD</th>
<th>R</th>
<th>LE</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>prop. 1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>α ≠ 0</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>prop. 2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>prop. 3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>prop. 4</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

We see that only the symmetric Kullback-Leibler measure, the log-det \( \alpha \)-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 $\alpha$-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 $\Lambda_k$ 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 $\delta_F$ on the special polar manifold $SP_n^*$ with update rule (a) is denoted F-SP-a. For the log-det $\alpha$-divergence, we consider values $\alpha \in \{-0.5, 0, 0.5\}$. When needed, we indicate the value of $\alpha$ with a subscript, e.g. $\alpha$LD_0.

The stopping criterion for the iterate $B_i$ is defined as $\|B_{i-1}B_i^{-1}-I_n\|_F/n$. Its tolerance is set to $\varepsilon_{RCG} = 10^{-5}$ for RCG and $\varepsilon_{RTR} = 10^{-6}$ for RTR. The optimization is performed with the manifold toolbox [47] in Matlab. We compare our algorithms with previously published NOJoB [34], uvedge [14] and jadiag [6], which correspond to F-IP-a, F-SP-c and IKL-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 $5 \times 10^{-4}$, 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 $3 \times 10^{-5}$, which corresponds to $5 \times 10^{-4}$ 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]

$$C_k = AA_kA^T + \frac{1}{\sigma}E_kD_kE_k^T + \beta I_n, \quad (45)$$

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 $\sigma$ and $\beta = 10^{-3}$ define the expected signal to noise ratio and uncorrelated noise, respectively. Diagonal matrices $\Lambda_k$ and $\Delta_k$ hold signal and noise sources energies and have i.i.d. elements with the $p^{th}$ element drawn from a chi-squared distribution with expectation $n/\sigma^2$. 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]

$$I_{MA}(M) = \frac{1}{2n(n-1)} \sum_{p=q=1}^{n} \left( \frac{\sum_{q=1}^{n} |M_{pq}|}{\max_{1 \leq q \leq n} |M_{pq}|} - 1 \right) + \frac{1}{2n(n-1)} \sum_{p=1}^{n} \left( \frac{\sum_{q=1}^{n} |M_{qp}|}{\max_{1 \leq q \leq n} |M_{qp}|} - 1 \right), \quad (46)$$

where $M = BA$, with $B$ the estimated joint diagonalizer and $A$ the true mixing matrix of the signal part in (45). Thus, $I_{MA}$ 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, uvedge and jadiag with the corresponding Riemannian versions F-IP-a, F-SP-c and IKL-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 $\alpha$-divergence (figure 6), the Riemannian distance (not
we do not show them here. Note that the optimization on $SP_n$ 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 $\alpha$-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 $\alpha LD_{a_1}$-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 $\sigma$. Post-hoc comparisons show that W-SP-b features lower performance for $n = 32$ and all values of $K$ and $\sigma$ 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 algorithm 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 $\alpha$-divergence for $\alpha = \{-0.5, 0, 0.5\}$ are plotted. The ANOVAs show significant differences for $n = 32$ and all values of $K$ and $\sigma$. Except for $\alpha LD_{a_1}$-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 $\sigma$ decreases. They also show that the only significant differences in the performance of $\alpha LD_{a_1}$-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 $\alpha$-divergence for $\alpha = -0.5$. Therefore, $\alpha LD_{a_1}$-IP-b appears to be the most accurate as compared to the algorithms based on the three Kullback-Leibler measures and the log-det $\alpha$-divergence for
\( \alpha = \{0, 0.5\} \).

In figure 9, we give the performance of algorithms corresponding to the Frobenius distance, the left and right Kullback-Leibler measures, the log-det \( \alpha \)-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 \( \alpha \)-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 \( SP_n \) appears advantageous only when associated with update rule (c), while with other update rules, \( TP_n \) 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 \( \alpha \)-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 2-32Hz and 16 seconds (\( T = 2048 \) samples) of each recording are used here. The real data are denoted \( X_{\text{real}} \).

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

\[
s(t) = 2r \cos \left( \frac{2\pi f}{f_s} \right) s(t - 1) - r^2 s(t - 2) + m(t),
\]

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 \( s = \tilde{s} + \tilde{s} \) in \( \mathbb{R}^{1 \times T} \), where \( \tilde{s} \) and \( \tilde{s} \) are simulated using (47) with \( f = 13 \)Hz and \( f = 21 \)Hz respectively. This source is projected on the electrodes with a random mixing vector \( \alpha \in \mathbb{R}^{n \times 1} \) to create the synthetic EEG \( X_{\text{sim}} = \alpha s \).

The BSS is performed on

\[
X = \sigma \frac{X_{\text{sim}}}{\|X_{\text{sim}}\|_F} + \frac{X_{\text{real}}}{\|X_{\text{real}}\|_F},
\]

where \( \sigma \) 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 \( u_k \) 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
real EEG datasets with respect to the noise parameter. For example, the performance of algorithms based on the Wasserstein distance over.

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 \( \sigma \). \(*\): significative differences when performing a one-way anova \( p < 5.10^{-4} \) on the performance of all algorithms.

their arithmetic 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 dependence between two vectors. It is defined as

\[
I_P(u, v) = 1 - \frac{|\text{cov}(u, v)|}{\sigma_u \sigma_v},
\]

where \( |\cdot| \) denotes the absolute value, \( \text{cov}(\cdot) \) returns the covariance of its arguments, \( \sigma_u \) and \( \sigma_v \) are the standard deviations of vectors \( u \) and \( v \). \( I_P \) 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 \( I_P(a, a_{est}) \), where \( a_{est} \) is the component of \( A_{est} = B^{-1} \) best matching \( a \). We also measure the quality of the recovered waveform \( s_{est} \) in \( S_{est} = BX \) with the index \( I_P(s, s_{est}) \), where \( s_{est} \) is the source process best matching \( s \).

1) Riemannian optimization: Again, we start by checking that Riemannian optimization is appropriate for AJD by comparing the performance of the classical algorithms NOJoB, u wedge and jading with the ones of the corresponding Riemannian versions F-IP-a, F-SP-c and ILK-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 significative 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 have found on the simulated data. However, we still have that all approaches are not equivalent when looking at subjects separately. Thus, practitioners 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.

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 preferred 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 $G_n$ 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 properties 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 simultaneously 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


