Riemannian approaches in Brain-Computer Interfaces: a review

Florian Yger, Maxime Berar, Fabien Lotte

Riemannian approaches in Brain-Computer Interfaces: a review

Florian Yger, Maxime Berar and Fabien Lotte

***Abstract*—Although promising from numerous applications, current Brain-Computer Interfaces (BCIs) still suffer from a number of limitations. In particular, they are sensitive to noise, outliers and the non-stationarity of ElectroEncephaloGraphic (EEG) signals, they require long calibration times and are not reliable. Thus, new approaches and tools, notably at the EEG signal processing and classification level, are necessary to address these limitations. Riemannian approaches, spearheaded by the use of covariance matrices, are such a very promising tool slowly adopted by a growing number of researchers. This article, after a quick introduction to Riemannian geometry and a presentation of the BCI-relevant manifolds, reviews how these approaches have been used for EEG-based BCI, in particular for feature representation and learning, classifier design and calibration time reduction. Finally, relevant challenges and promising research directions for EEG signal classification in BCIs are identified, such as feature tracking on manifold or multi-task learning.**

***Index Terms*—Riemannian geometry, Brain-Computer Interface (BCI), covariance matrices, subspaces, source extraction, Electroencephalography (EEG), classification**

I. Introduction

BRAIN-COMPUTER INTERFACES (BCIs) enable their users to interact with computers via brain activity only, this activity being typically measured by ElectroEncephaloGraphy (EEG) [1]. For instance, a BCI can enable a user to move a cursor leftwards or rightwards on a computer screen, by imagining left or right hand movements respectively [2]. BCIs have proven very promising, e.g., to provide communication to severely paralyzed users [3], as a new control device for gaming [4] or to design adaptive human-computer interfaces that can react to the user's mental state [5], to name a few [6]. However, most of these applications are prototypes and current BCI are still scarcely used outside laboratories.

The main reason that prevents EEG-based BCIs from being widely used is their low robustness and reliability [1][7]. Indeed, current BCIs too often recognize erroneous commands from the user, which results in rather low accuracy and information transfer rate, even in laboratory conditions [7][8]. Moreover, EEG-based BCIs are very sensitive to noise, e.g., user motions [9], as well as to the non-stationarity of EEG signals [10]. Indeed, a BCI calibrated in a given context is very likely to have much lower performances when used in another context, see, e.g., [11][12]. Finally, in addition to this reliability issue, BCIs also suffer from long calibration times. This is due to the need to collect numerous training EEG examples from each target user, to calibrate the BCI specifically for this user, to maximize performances [13].

Therefore, for BCIs to be usable in practice, they must be robust across contexts, time and users, and with calibration times as short as possible. These challenges can be addressed at multiple levels, e.g., at the neuroscience level, by identifying new neurophysiological markers that are more reliable than the ones currently used, at the human level, by training users to gain accurate and stable control over the EEG patterns they produce [14] or at the signal processing level, by building features and classifiers that are robust to context changes, and that can be calibrated with as little data as possible. Regarding EEG signal processing, some recent results suggest that a new family of approaches is very promising to address the multiple challenges mentioned above: Riemannian approaches [15][16][17]. These approaches enable the direct manipulation of EEG signal covariance matrices and subspaces, with an appropriate and dedicated geometry, the Riemannian geometry. They have recently shown their superiority to other classical EEG signal processing approaches based on feature vector classification, by being the winning methods on a couple of recent brain signal classification competitions, notably the "'DecMEG2014"' (https://www.kaggle.com/c/decoding-the-human-brain) and the "'BCI challenge 2015"' (http://neuro.embs.org/2015/bci-challenge/) competitions.

Since then, these approaches have witnessed an increased enthusiasm from the research community, and have been used to explore new feature representations, to learn features and design robust classifiers as well as to generate artificial EEG data [15][18][19][17][20]. Thus, time is now ripe to review what these methods are, what they can already contribute to BCI design, and how they should be further explored in the future. This is what this review paper proposes.

This paper is organized as follows: Section II first presents the standard design of a BCI, to illustrate how it employs covariance matrices. Then, Section III proposes a brief tutorial on Riemannian geometry, that makes possible the direct manipulation of such covariance matrices. Then Section IV reviews how such approaches have been used for EEG-based BCI, in particular for subspace methods (e.g., spatial filtering, see Section IV-A) and for covariance methods (to represent and classify EEG signals, to learn metrics or reduce calibration time, see Section IV-B). Finally, Section V identifies relevant challenges and promising research directions for EEG signal classification based on Riemannian approaches.

F. Yger is with the laboratory LAMSADE, Université Paris-Dauphine, Place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, France, e-mail: florian.yger@dauphine.fr.

M. Berar is with the laboratory LITIS, Université de Rouen, Avenue de l'Université, 76800 Saint Etienne du Rouvray, France e-mail: maxime.berar@univ-rouen.fr.

F. Lotte is with Potioc / LaBRI, Inria Bordeaux Sud-Ouest, 200 avenue de la vieille tour, 33405, Talence Cedex, France e-mail: fabien.lotte@inria.fr.

II. Standard BCI design

A prominent type of BCIs is oscillatory activity-based BCIs, that exploit amplitude changes in EEG oscillations. They notably include motor imagery-based BCIs. Such a BCI is typically designed around the Common Spatial Patterns (CSP) algorithm to optimize spatial filters and the Linear Discriminant Analysis (LDA) classifier. CSP aims at learning spatial filters such that the variance of spatially filtered signals is maximized for one class and minimized for the other class [21]. Formally, optimizing CSP spatial filters w (w being a column vector) consists in extremizing the following function:

$$J_{CSP}(w) = \frac{w^T C_1 w}{w^T C_2 w} \tag{1}$$

where C_j is the average spatial covariance matrix[1] of the band-pass filtered EEG signals from class j, and $(.)^\top$ denotes the transposition. Typically, these spatial covariance matrices are obtained by computing the spatial covariance matrix S_i^j from each trial Z_i^j from class j, and then averaging them:

$$C_j = \frac{1}{N_j}\sum_i^{N_j} S_i^j = \frac{1}{N_j}\sum_i^{N_j} Z_i^j {Z_i^j}^\top \tag{2}$$

with N_j the number of trials in class j and $Z_i^j \in \mathbb{R}^{N_c \times N_s}$ is the j^{th} EEG trial from class i, with N_s the number of samples in a trial, and N_c the number of channels. Note that EEG signals are usually band-pass filtered and thus have a zero mean. Equation 1 being a generalized Rayleigh quotient, it can be extremized by Generalized Eigen Value Decomposition (GEVD) of the average covariance matrices C_1 and C_2 [21]. The spatial filters which maximize/minimize $J_{CSP}(w)$ are the eigenvectors corresponding to the largest and smallest eigenvalues of this GEVD, respectively. It is common to select 3 pairs of CSP filters w_i, corresponding to the 3 largest and smallest eigenvalues [21]. Once the filters w_i are obtained, a feature f_i is computed as $f_i = log(w_i^T S w_i)$, where S is the current trial covariance matrix. The LDA classifier uses a linear hyperplane to separate feature vectors from two classes [22]. The intercept b and normal vector a of this hyperplane are computed as follows:

$$a = C^{-1}(\mu_1 - \mu_2) \text{ and } b = -\frac{1}{2}(\mu_1 + \mu_2)^T a \tag{3}$$

with μ_1 and μ_2 being the mean feature vectors for each class and C the covariance matrix of both classes. It is interesting to note that both CSP and LDA require estimating covariance matrices. Although other types of BCI such as Event-Related Potentials (ERP)-based BCI may not manipulate covariance matrices as explicitly, we will see later that they can also be represented using covariance matrices (see Section IV-B1). As mentioned earlier, Riemannian geometry provides tool to directly manipulate those covariance matrices, without the need for spatial filters. The next section provides an introductory tutorial on this Riemannian geometry and associated tools.

[1]throughout this manuscript, average covariance matrices estimated over several trials will be denoted as C, whereas covariance matrices estimated from a single trial will be denoted as S

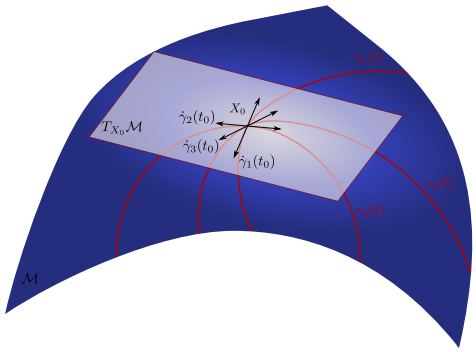

Fig. 1. On the differential manifold $\mathcal{M}$, the tangent space at X_0 is the set of the velocities $\dot\gamma(0)$ of the curves $\gamma(t)$ passing through X_0 at $t = 0$.

III. Concepts of Riemannian geometry

To phrase it in an almost over-simplistic way, Riemannian geometry is the branch of mathematics that studies smoothly curved spaces that locally behave like Euclidean spaces. Although it may look like an exotic mathematical tool, the concept of Riemannian manifold is more common than one can expect as the Earth is an example of a Riemannian manifold and we will use this as a pedagogical example in what follows. As this section will only give basic concepts about Riemannian geometry, the reader can refer to [23], [24] for an in depth exploration of the topic.

A. Riemannian manifolds

Riemannian manifolds are defined as the result of imbricated mathematical structures in the manner of Russian dolls.

First, we need to define *topological* manifolds, which are spaces in which every point has a neighborhood homeomorphic to $\mathbb{R}^n$. To simplify, a topological manifold can be seen as a space that locally looks flat.

Endowed with a differential structure -also called atlas- (i.e. a set of bijections called charts between a collection of subsets of the topological manifold and a set of open subsets of $\mathbb{R}^n$), the topological manifold becomes a *differential* manifold. Within differential manifolds, there exist *smooth* manifolds which are differential manifolds for which the transitions between maps are smooth. To simplify again, this step aims at giving rules for locally translating a point on the manifold to its linear approximation. Those rules are local but on a smooth manifold, the rules slightly change from one point to another.

On every point of a smooth differential manifold, the notion of tangent space can be defined as the velocity of the curves passing the point as illustrated in Fig. 1. A *Riemannian* manifold is then a real smooth manifold equipped with an inner product on the tangent space at each point.

For any Riemannian manifold, there exists a pair of mappings transporting points from the manifold to any given tangent space and vice versa. More precisely, the Exponential mapping transports a tangent vector (i.e. a point in a tangent space) to the manifold and the Logarithmic mapping is locally defined to transport a point in the neighborhood of a point to the tangent space defined at this point. As a consequence, Riemannian manifolds can be locally approximated by Euclidean

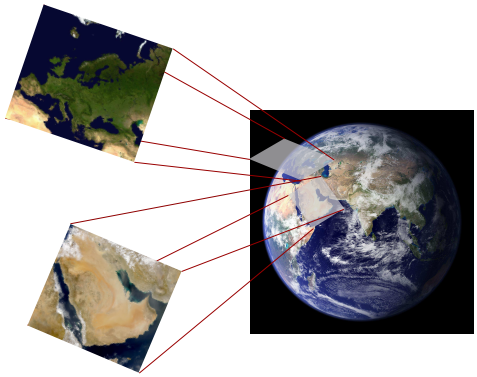

Fig. 2. The Earth can be seen as a Riemannian manifold as it can be locally linearly approximated. On earth and for Riemannian manifolds in general, a map is only valid locally and implies deformations when used further.

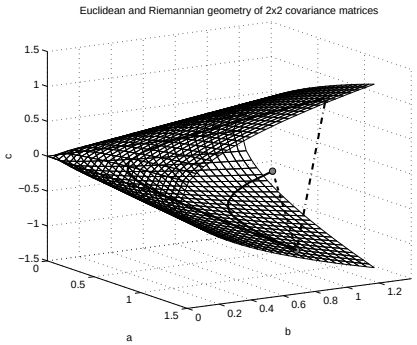

Fig. 3. Comparison between Euclidean (straight dashed lines) and Riemannian (curved solid lines) distances measured between 2×2 SPD matrices.

spaces via their tangent spaces, however deformations occur for points mapped far from where the tangent point is defined.

As depicted in Fig. 2, in our example of the Earth, the topological manifold is the sphere that is locally homeomorphic to $\mathbb{R}^2$. Defining a differential structure on the Earth boils down to defining geographic maps and to gather them in an atlas. Once a scalar product is defined on each map, distances can be computed locally and then extended to a global notion of distance computed along curves on the manifold, called geodesics. On an Earth map, the deformations of the shapes of countries far from the center of the map results from the approximation made by choosing a tangent plan defined at the center of the map.

For example, as described in [25], on the set of Symmetric Positive Definite (SPD) matrices $\mathcal{P}_n$ -that will be presented next-, the tangent space at a point X, $T_X\mathcal{P}_n$, is the space of symmetric matrices. Then, one choice (for making a Riemannian manifold out of this space) is to equip every tangent space with the following metric:

$$\forall A, B \in T_X\mathcal{P}_n \quad \langle A, B \rangle_X = \mathrm{Tr}\left(X^{-1}AX^{-1}B\right).$$

From this choice of metric follows the definition of a distance in Eq. 8. Then, any symmetric matrix A belonging to $T_X\mathcal{P}_n$, the tangent space at X, can be mapped on $\mathcal{P}_n$ (with the reciprocal operation) as

$$S_A = \exp_G(A) = X^{\frac{1}{2}} \exp\left(X^{-\frac{1}{2}} A X^{-\frac{1}{2}}\right) X^{\frac{1}{2}}, \quad (4)$$

$$A = \log_X(S_A) = X^{\frac{1}{2}} \log\left(X^{-\frac{1}{2}} S_A X^{-\frac{1}{2}}\right) X^{\frac{1}{2}} \quad (5)$$

with $\log(.)$ and $\exp(.)$ the matrix logarithm and exponential.

Given a differential manifold, a common way to create a Riemannian manifold is to embed the tangent space with the usual scalar product of the ambient space. Doing so creates a Riemannian submanifold. However, for some cases (like the space of SPD matrices), several scalar products are available, leading as we will see, to different Riemannian geometries.

B. Bestiary of manifolds for BCI

When dealing with EEG signals, one has to manipulate matrices under certain types of constraints. Most of the classical constraints used in BCI applications are smooth. Hence, the space under constraints can be interpreted as a smoothly curved space. Equipped with the proper mathematical structure, those spaces can be handled as Riemannian manifold. In terms of applications, Stiefel and Grassman manifolds are well-suited to model subspaces methods as in [26], [27], [28], [20] and Symmetric Positive Definite (SPD) matrices naturally models covariance matrices [15], [19], [16].

1) Manifolds of subspaces: In BCI applications, the most prominent constraints are orthogonality constraints (that occur in every eigenproblem for example). More formally, the space of orthonormal matrices is defined as the Stiefel manifold $St(n,p) = \{X \in \mathbb{R}^{n\times p} | X^\top X = \mathbb{I}_{n\times p}\}$. At every point of this space, we have the tangent space $T_X St(n,p) = \{V \in \mathbb{R}^{n\times p} | X^\top V + V^\top X = 0\}$ equipped with the usual Euclidean scalar product of $\mathbb{R}^{n\times p}$.

The Stiefel manifold comprises several special cases -like the orthogonal group $O_p = St(p,p)$- and variants -like the generalized Stiefel manifold $St_G(n,p) = \{X \in \mathbb{R}^{n\times p} | X^\top G X = \mathbb{I}_{n\times p}\}$ [26]. Many pre-processing, blind source separation or classification methods currently used in practice (Principal Component Analysis (PCA), Canonical Correlation Analysis (CCA), ...) share a manifold structure of this kind. Moreover, equipped with a group structure over the space O_p, the Stiefel manifold becomes a quotient space named the Grassman manifold $Gr(n,p)$. This manifold is particularly recommended when a cost function is invariant over O_p in order to reduce the search space. For a complete overview about those manifolds, the interested reader can refer to [29] and [24].

2) Manifold of covariances matrices: The space of SPD matrices, noted $\mathcal{P}_n = \{X \in \mathbb{R}^{n\times n} | X = X^\top, X \succ 0\}$ and composed of symmetric matrices of strictly positive eigenvalues, can be successfully applied for manipulating covariance matrices from EEG signals. At every point of the space of SPD matrices, we have the tangent space $T_X\mathcal{P}_n$ homogeneous to the space of $n \times n$ symmetric matrices. Depending on the choice of scalar product to equip the tangent spaces, one Euclidean and two different Riemannian geometries are

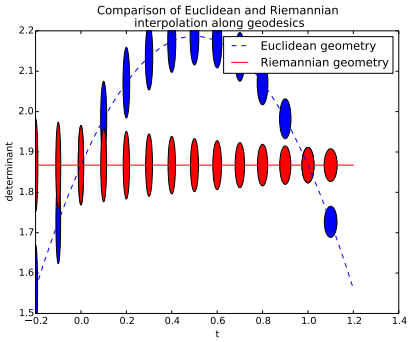

Fig. 4. Illustration of the swelling effect : the evolution of the determinant of matrices interpolationg two other matrices in the Euclidean sense (in blue) and in the Riemannian sense (in red). In the Euclidean interpolation, the determinant of the interpolation is bigger than the determinant of the points. In the Riemannian interpolation, the determinant is constant.

possible and subsequently two different Riemannian distances (and one Euclidean) can be defined:

- the Euclidean distance[2]

$$\delta_e\left(X_A, X_B\right) = \|X_A - X_B\|_{\mathcal{F}} = \sqrt{\langle X_A - X_B, X_A - X_B\rangle_{\mathcal{F}}}, \tag{6}$$

- the LogEuclidean distance [30], [19]

$$\delta_l(X_A, X_B) = \|\log\left(X_A\right) - \log\left(X_B\right)\|_{\mathcal{F}}, \tag{7}$$

- the Affine Invariant Riemannian Metric (AIRM) distance [25]

$$\delta_r(X_A, X_B) = \|\log(X_A^{-\frac{1}{2}} X_B X_A^{-\frac{1}{2}})\|_{\mathcal{F}}, \tag{8}$$

where the $\log(.)$ corresponds to the matrix logarithm.

The difference between the Euclidean and Riemannian geometries is illustrated in Fig. 3, where 2×2 SPD matrices are represented as points in $\mathbb{R}^3$. The positivity constraint is a cone, inside which SPD matrices lie strictly. The minimal path between the two points for Euclidean distance is a straight line, whereas the computed AIRM distances draw curves.

Despite its apparent simplicity, the Euclidean geometry has several drawbacks and is not always well suited for SPD matrices [31], [30], [32], which motivates the use of Riemannian geometries. For example, the Euclidean geometry induces some artifacts like the so-called *swelling effect* [30]. As illustrated in Fig. 4, this effect is observed in task as simple as averaging two matrices, where the determinant of the average is larger than any of the two matrices. Hence, such an effect can be particularly harmful for data analysis as it adds spurious variation to the data. Another drawback, illustrated in Fig. 3 and already noticed in [31], is the fact that Euclidean geometry for SPD matrices forms a non-complete space. Hence, in this Euclidean space interpolation between matrices is possible (but affected by the swelling effet), but extrapolation may produce uninterpretable indefinite matrices.

[2]Note that, in order to make it a Euclidean space, $\mathcal{P}_n$ is equipped with the Frobenius inner product $\langle X_A, X_B\rangle_{\mathcal{F}} = \mathrm{Tr}\left(X_A^\top X_B\right)$

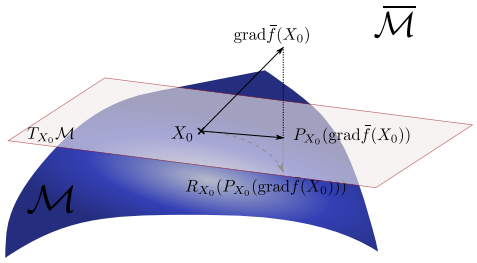

Fig. 5. Gradient descent on a Riemannian manifold : at the point X_0, the Euclidean gradient is projected on the tangent space at X_0 and then mapped back on the manifold $\mathcal{M}$.

As a consequence, some methods can be transformed into their Riemannian counter-part by substituting the Euclidean distance by a Riemannian distance. For example, in a metric space -like a Riemannian manifold-, the Fréchet mean extends the concept of mean[3]. It can be defined as the element $\bar{E}$ of the space minimizing the sum of its squared distances[4] δ to the points E_i of the dataset:

$$\bar{X} = \mathrm{argmin}_X \sum_{i=1}^{N} \delta^2(X_i, X). \tag{9}$$

Contrary to the Euclidean mean, the Fréchet mean usually does not have any closed-form solution and it must be obtained by solving the optimization problem in Eq 9. Such an averaging method has attracted a lot of attention from the optimization community and for example, for SPD matrices, several algorithms have been proposed [34], [35], [36].

C. Optimization on Riemannian manifolds

Optimization on matrix manifolds is by now a mature field with most of the classical optimization algorithms having a (Riemannian) geometric counterpart [24]. In spirit, it consists in optimizing a function over a curved space instead of optimizing it under a smooth constraint in a big Euclidean space. In Riemannian setting, descent directions are no longer straight lines, but curves on the manifold and at every step, an admissible solution is created. To account for this, the algorithms must be slightly modified. Fig. 5 sums up the general recipe for the implementation of Riemannian gradient descent for a function f:

1) At each iteration, at point X_t, transform the Euclidian gradient $D_{X_t} f$ into a Riemannian gradient $\nabla_{X_t} f$.
2) Perform a line search along geodesics at X_t in the direction $H = \nabla_{X_t} f$. To do so the Riemannian gradient $\nabla_{X_t} f$ is mapped to the geodesic using the exponential map. For computational reasons, an approximation of the exponential map, called a retraction, is often used instead [24].

The reader interested by the details about optimization can refer to [24] and to [29] for the special case of Stiefel and Grassman manifolds. Most of the examples given in [24] have been implemented efficiently in a MATLAB toolbox [37].

[3]In the litterature, the Fréchet mean is also sometimes called the Karcher mean or the geometric mean.

[4]Note that using a simple distance instead of the squared distance would give a Fréchet median [33].

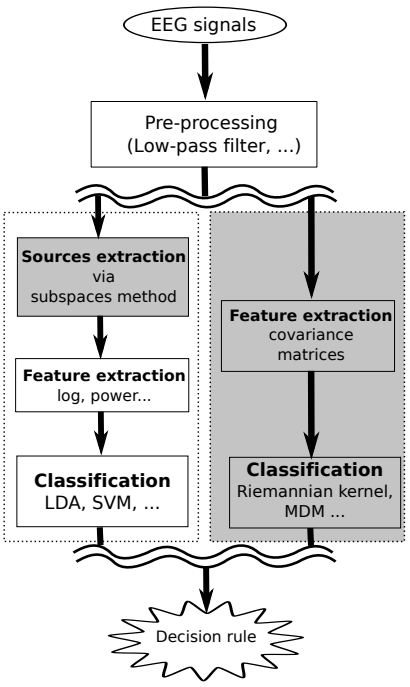

Fig. 6. Processing pipelines used in BCI, grey areas indicate where Riemannian approaches are or can be used. The left branch is the classical pipeline currently used, the right is the new Riemannian approach reviewed in this paper.

IV. Riemannian approaches for BCI

BCIs are typically designed with a few standard steps to process and classify EEG signals, to translate them into a command [22]. Such processing pipeline, illustrated on Figure 6 (left) typically involves 1) preprocessing the signals, e.g., band-pass filtering them, 2) extracting sources, using spatial filters such as CSP, 3) extracting a feature vector from the resulting source signal, e.g., the signal bands power and 4) classifying this feature vector using a vector-based classifier such as LDA. As mentioned in Section II, this pipeline often involves estimating and manipulating covariance matrices, notably at the source extraction level. Thus, Riemannian approaches can already be used at that stage, e.g., to improve source extraction. This is reviewed in Section IV-A. Alternatively, since the BCI pipeline does involve covariance matrix estimation and manipulation, the whole processing pipeline can also be designed around such covariance matrices, using Riemannian geometry, by directly classifying those matrices. In other words, the full Riemannian BCI pipeline, illustrated on Figure 6 (right), consists in preprocessing the signals (as before), representing them as covariance matrices, and classifying them using covariance matrix-based classifiers, that exploit Riemannian geometry. It should be noted that this new processing pipeline is simpler as it does not require source extraction anymore. Indeed, the spatial information necessary for source extraction is already available in covariance matrix representations. This Riemannian pipeline is reviewed hereafter in Section IV-B.

A. Riemannian approaches for Subspaces methods

1) Source extraction: Sources can be extracted through PCA, CCA, or CSP. Those approaches enforce either an orthonormality, an uncorrelation or independance constraint on the sources making the Stiefel manifold and its variants natural tools for their modelisation.

Having extracted sources from EEG signals, the comparison of subspaces on Grassmann manifolds have been investigated in [38], [27], [28].

2) Going beyond CSP with Riemannian geometry: As previously stated, there has been a lot of questioning on how to link CSP based approaches to covariance based approaches. One attempt has been to cast the CSP framework into the information geometry [39]. In their core principle, CSP algorithms use covariance matrices of both classes of signals. Those covariance matrices reflect the mean covariance given the class of the signal. To do so, most algorithms rely on a Euclidean average for both classes of signals. Hence, other attempts have tried to include concepts of Riemannian geometry in CSP where covariance matrices are averaged.

The first approach [17] has consisted in replacing the Euclidean mean by a Riemannian averaging. Applying such a non-Euclidean averaging before a CSP algorithm seemed to improve the accuracy of the whole BCI system, at least when the number of sensors is not too high.

Considering that a CSP is a subspace, the authors of [40] have considered it as an element of a Grassman manifold. Then, using the geodesic on this space, they proposed a methodology to adapt the subspace to cope with dataset-shift between the training and the test phase.

B. Riemannian approaches for Covariances methods

1) Covariances as representation of the signals: As in most data processing pipelines, the feature extraction step of BCI system relies on *a priori* knowledge on the data and on the protocol for acquiring the signals. From the literature, we have identified two main ways of extracting features for EEG data:

- signal energy-based features
- sample based features.

The energy information is either extracted from the raw or preprocesed (typically by a band-pass or low-pass filter) signals or from extracted sources or just from a few sensors and is useful for protocols like Motor Imagery (MI), Steady-State Visual Evoked Potentials (SSVEP), respiratory state. The temporal information is usually used for ERP.

However, as shown in several papers, covariance matrices can be built[5] in order to extract either sample information or energy information. For example, for $Z \in \mathbb{R}^{n \times s}$ a multi-channel segment of signal and $T \in \mathbb{R}^{n \times s}$ a template signal, the following can be built:

- spatial covariance matrix: $C_s = \frac{1}{s} Z Z^\top$ - with the variance/power of electrodes on the diagonal,

[5]Note that only the Maximume Likelihood Estimator is shown here but the use of other estimators like [41], [42], [43], [44] could also be investigated.

- template-signal covariance: $C_T = \begin{pmatrix} TT^\top & TZ^\top \\ ZT^\top & ZZ^\top \end{pmatrix}$ - where the sample information is extracted on the off-diagonal blocks.
- filtered signal covariance: $C_f = \begin{pmatrix} Z_{f_1}Z_{f_1}^\top & \cdots & Z_{f_1}Z_{f_F}^\top \\ \vdots & \ddots & \vdots \\ Z_{f_F}Z_{f_1}^\top & \cdots & Z_{f_F}Z_{f_F}^\top \end{pmatrix}$ with the Z_f filtered versions (in frequency bands including the frequency of the targetted stimulations) of the original signal.

As far as one can tell from the literature, the approach described in [45] is one of the first attempts at classifying covariance matrices directly. Moreover, as the CSP is based on the extraction of sources from the class-covariance of the signals, there exists a strong link between the spatial covariance and the sources. Several attempts have been made for bridging the gap between CSP and the covariance based approaches [46], [47].

Since it extracts spatio-frequential information of the signals, spatial covariance matrices have been used for tasks involving Event-Related Desynchronization (ERD) such as MI [15], [19], [16] or respiratory state [48].

Using as template the mean target signal (e.g. a P300 signal in P300-speller application), template-signal covariance matrices have been used for ERP tasks [16], [49]. Then, using a filter bank with the frequencies of flickering objects, the covariance have been used for SSVEP [16], [50].

2) Metric learning: Metric learning consists in automatically tuning the parameter of a similarity or of a dissimilarity measure from the data. Usually, the learned metric aims at easing the task of classification. As reviewed in [51], it has been successfully applied to a great variety of data.

Naturally, the problem of learning the metric for data on a manifold has occured. For example, the seminal article [52] tackles the problem of metric learning for data on a manifold in the context of computer vision. Stemming from this approach, several methods have been tailored for the case of BCI.

To the best of our knowledge, Riemannian metric learning has only been proposed for data on the space of SPD matrices. In the case of [52], the proposed method consisted in finding a transformation that maps the data from the manifold $\mathcal{P}_n$ to the lower dimensional $\mathcal{P}_m$ while making the data more separable. This problem is then formulated as:

$$\min_W \sum_{i,j} A_{i,j}\delta^2\left(W^\top S_i W, W^\top S_j W\right)$$

with W a full-rank matrix of $\mathbb{R}^{n\times m}$ in the general formulation (but in practice, $W \in Gr(n,p)$) and A an affinity matrix exploiting the labels and the distances between points in $\mathcal{P}_n$. Several variants of this methods can be instantiated depending on the choice of Riemannian distance δ. This supervised dimensionality reduction has been applied on covariance matrices extracted from images and either followed by a simple Nearest-Neighbor (NN) classifier (based on the chosen δ) or by a combination of dictionary learning and NN classifier. On several datasets, the performances using the compressed matrices were at worst comparable with the original matrices.

Using the same parametrized mapping from $\mathcal{P}_n$ to $\mathcal{P}_m$ and a related cost function, a Riemannian version of the PCA algorithm tailored for SPD matrices has been proposed in [20]:

$$W = \underset{W\in Gr(n,p)}{\operatorname{argmax}} \sum_i \delta^2\left(W^\top S_i W, W^\top \bar{S} W\right). \quad (10)$$

In a metric space, the Fréchet variance extend the concept of variance in the same way the Fréchet extends the concept of mean. Then, it can be defined as $\sigma_\delta^2 = \frac{1}{N}\sum_{i=1}^N \delta^2(S_i, \bar{S})$. Hence, this cost can be interpreted as a PCA as it maximizes the Fréchet variance of the compressed samples.

In a sense, this geometry-aware PCA is also linked to subspace methods, as it looks for an orthonormal matrix W but on the contrary it keeps the symmetric positive definiteness of the matrices (although in a manifold of smaller dimension).

This unsupervised dimensionality reduction technique has been applied as a preprocessing of covariance matrices from EEG data [53] and followed by a simple Minimum Distance to the Mean (MDM) classifier (described in the next section). With the AIRM distance, the geometric PCA obtained accuracy rates close to the MDM applied on uncompressed covariances and was also comparable to the classical baseline CSP+LDA. It is then a very promising research lead for bridging the gap between CSP-based approaches and covariance based approaches.

The approach described in [54] is another promising approach to bridge this gap. Contrary to the Riemannian PCA, it does not involve any dimensionality reduction but rather the learning of a metric through a mapping.

Following [25], the metric learning algorithm proposed in [54] uses a *congruent transform*, i.e., for any $S, G \in \mathcal{P}_n$,

$$\Gamma_{G^{-\frac{1}{2}}}(S) = G^{-\frac{1}{2}} S G^{-\frac{1}{2}}.$$

For a set of covariance matrices $\{S_1, \cdots, S_N\}$ and y the vector of their respective labels, the problem is formulated as :

$$\max_{G\in\mathcal{P}_d} \frac{\langle Uh(G)U, yy^\top\rangle_{\mathcal{F}}}{\|Uh(G)U\|_{\mathcal{F}}}, \quad (11)$$

where U is a centering matric [55] and $h(G)$ is the Gram matrix for k_G (the LogEuclidean metric):

$$\begin{aligned} h_{ij}(G) &= \operatorname{Tr}\left(\log\left(G^{-\frac{1}{2}} S_i G^{-\frac{1}{2}}\right)\log\left(G^{-\frac{1}{2}} S_j G^{-\frac{1}{2}}\right)\right) \\ &= \operatorname{Tr}\left(\log\left(\Gamma_{G^{-\frac{1}{2}}}(S_i)\right)\log\left(\Gamma_{G^{-\frac{1}{2}}}(S_j)\right)\right) \end{aligned}$$

This cost can be interpreted as the (centered) *kernel target alignment* (KTA) criterion [56], [57], [55] used on the LogEuclidean metric (applied on the tangent space at G). Contrary to the AIRM distance, the LogEuclidean distance (and its related metric) is not invariant to $\Gamma(.)$ and the Exponential Metric Increasing (EMI) property shown in [25], [58] illustrates the distortion implied by choosing G. Hence, this problem boils down to finding the reference point G of a tangent space where the distortion involved by the Logarithmic mapping is benefic for classification.

On the BCI data from [59], the LogEuclidean distance derived from the learned metric has shown better performances than other distances on $\mathcal{P}_n$ when used in a 1-NN classifier.

3) Classifiers: The Riemannian framework providing measures of distances between SPD matrices along the Riemannian manifold, such distances could be used to build classifiers that directly use SPD matrices as input. A simple yet efficient way to built such a classifier is the Minimum Distance to the Mean (MDM) approach [15]: the SPD matrix S_i representing the EEG trial i is assigned to the class k for which the average SPD matrix C^k is the closest from S_i according to a Riemmanian distance. Formally, such a Riemannian MDM classifier is built as follows:

1) **Classifier training:** Just compute the average C^k of the SPD matrices representing the trials S_i^k from each class k, using the Frechet mean from Equation 9.
2) **Classifier use:** To classify a new EEG trial, first compute the SPD matrix S_j that represents it (see Section IV-B1), and determine its class C^k as $k = \arg\,\mathrm{argmin}_k\, \delta^2(S_j, C^k)$.

This approach already makes full use of the Riemannian framework, and has been reported to result in classification accuracies equivalent to those obtained using the CSP spatial filtering algorithm with an LDA classifier [15]. Unfortunately, SPD matrices cannot be used as input to most of the typical classification algorithms used in BCI, such as LDA or Support Vector Machines (SVM) [22], since they use vectors as input. To benefit from both the Riemannian framework and available classification algorithms, Barachant et al proposed to project the SPD matrices onto the tangent space of the Riemannian manifold, where they can be vectorized and thus used as input to an LDA or SVM [15]. To do so, the average SPD matrix C of the whole training data set (all classes together) is first computed, and then each SPD matrix S_i is projected on the tangent space of the Riemannian manifold at point C, and vectorized as follows:

$$s_i = upper\left(C^{-\frac{1}{2}}\log_C(S_i)C^{-\frac{1}{2}}\right) \tag{12}$$

where $upper$ is an operator that keeps the upper triangular part of a symmetric matrix and vectorize it by applying a weight of 1 for elements on the diagonal, and a weight of $\sqrt{2}$ for off-diagonal elements. The resulting s_i being a vector, it can be used as input to any vector-based classifier such as LDA or SVM. However, this vector has dimension $n(n+1)/2$ (with n the number of rows/columns of the SPD matrices), which might be larger than the number of available training trials in many BCI contexts. Therefore, it is usually necessary to perform a feature selection step on these vectors to first reduce their dimensionality. By doing so, it was reported that using such vectors as input to an LDA significantly outperformed the classical CSP+LDA combination [15].

There is another alternative to use the Riemannian geometry and SPD matrices as input to a classifier, that does not require to vectorize such matrices: the so-called 'kernel trick' [60]. While most linear classification algorithms use as input a feature vector, it may be possible to substitute inner products in such algorithms by a kernel function, to be able to directly classify more structured data such as graphs, trees or SPD matrices, possibly non-linearly. A kernel can be seen as a similarity function between pairs of data points. Interestingly enough, a kernel k enables to efficiently compute the inner-products between two data points x and y, projected onto a different space using a mapping ϕ, without explicitly computing the projection $\phi(x)$ and $\phi(y)$, i.e., $k(x,y) = \langle\phi(x), \phi(y)\rangle$, with $\langle .,. \rangle$ being an inner product[6]. The kernel trick was first made popular for the SVM algorithm, whose decision function takes the following form:

$$h(x) = b + \sum_i \alpha_i y_i \langle x_i, x\rangle \tag{13}$$

with b the bias, y_i the class label (1 or -1) of training data x_i, x the data to classify, α_i the classifier weights obtained during training, and $h(x)$ the classifier output (positive for one class and negative for the other class). The formulation makes clear the use of an inner product between pairs of data points, which can thus be substituted by a kernel function, in order to map the data to a different space and/or work with structured data, and not only feature vectors. Applying the kernel trick to an SVM leads to the following decision function:

$$h(x) = b + \sum_i \alpha_i y_i \underbrace{\langle\phi(x_i), \phi(x)\rangle}_{k(x_i,x)} \tag{14}$$

This made it possible to use SVM to perform non-linear classification, e.g., by using Gaussian kernels, or to classify directly graphs or trees by using dedicated kernels for graphs or trees. Note that the kernel trick can be used on any algorithm based on an inner-product between pairs of data points (e.g., LDA, PCA, CCA), and is not restricted to the SVM. In any case, this kernel trick can thus be used to classify directly SPD matrices and exploiting Riemannian geometry by using a dedicated kernel for SPD matrices that is based on the Riemannian distance. Such a kernel was initially proposed by Barachant et al for BCI, in [19]. The so-called LogEuclidean kernel k_l^G is defined as the scalar product between two SPD matrices projected on the tangent space at point G:

$$k_l^G(S_i, S_j) = \mathrm{Tr}\left[\log(G^{-\frac{1}{2}}S_iG^{-\frac{1}{2}})\log(G^{-\frac{1}{2}}S_jG^{-\frac{1}{2}})\right] \tag{15}$$

Typically, the reference point G is defined as the mean SPD matrix on the training set (all classes together) [19]. Later, Yger et al, explored two other kernels on SPD matrices for BCI: the Stein kernel, which is closely related to the Riemannian distance, and the Euclidean kernel, that can be seen as a linear approximation of the LogEuclidean kernel [58]. Comparisons between these kernels used within an SVM and the standard BCI method for motor imagery (CSP+LDA) revealed that the LogEuclidean kernel led to the best classification accuracy, and even outperformed the standard BCI design [19][58]. Interestingly enough, normalizing the kernels also led to increased classification accuracies [58].

Finally, as explained in [62], it is possible to embed a distance in a radial kernel in order to benefit from the properties of both the kernel classifier and the embedded distance. However, as the distance must follow a technical condition -namely negative definiteness- that is incompatible with some Riemannian distances like the AIRM distance [63] but compatible with others [64], [65].

[6]For the sake of simplicity, we omit the details about the functional space on which the inner product is defined [61]

4) Calibration time reduction: As mentioned previously, current BCI systems suffer from long calibration times. This is due to the need to calibrate the classifier on each BCI user individually, and thus to collect numerous training data from each user [1][66]. Thus, to reduce BCI calibration, it is desirable to calibrate BCI classifiers with as little training data as possible. Several approaches have been explored to do so, such as subject-to-subject transfer, semi-supervised learning, or artificial data generation [13]. Riemannian approaches have been very recently shown to be useful tools for that objective.

In [13], Riemannian methods were used to perform subject-to-subject transfer. The idea is to improve the calibration of a classifier trained on a few data from a target subject, by regularizing it towards the data of other subjects, for which there are more data available. The Riemannian distance was used in [13] to find the distance between the average covariance matrix of the target subject (for each class) and the average covariance matrix of the other subjects available. Then, the target subject average covariance matrix was regularized towards that of the more similar subjects, as measured using the Riemannian distance, for both the CSP and LDA algorithm. This was shown to significantly improve classification performances when little training data was available for the target subject, and thus to reduce calibration time.

In [18], the Riemannian framework was used to generate artificial EEG data from the few available original EEG data, i.e., to perform data augmentation to ease the classifier calibration. Kalunga et al proposed to generate artificial covariance matrices by interpolating new matrices between two original matrices (of the same class), along the geodesic connecting them. This would thus densify the training set while remaining in its convex hull. Formally, this is done by creating artificial matrices $S(t)$, on the geodesic connecting two original matrices S_1 and S_2, as follows:

$$S(t) = \exp\left(t \log_{S_1}(S_2)\right) = S_1^{\frac{1}{2}} \left(S_1^{-\frac{1}{2}} S_2 S_1^{-\frac{1}{2}}\right)^t S_1^{\frac{1}{2}} \quad (16)$$

with $0 < t < 1$. Varying the value of t generates different interpolated covariance matrices along the geodesic connecting S_1 and S_2. The training data set can thus be augmented by generating multiple such artificial covariance matrices between each pair of original covariance matrices available, for each class. This approach was shown to significantly improve the classification performance of BCI based on SSVEP, notably for subjects with initially poor performances [18]. It was also shown to be a useful tool for BCI based on ERP - notably Error Potentials - to balance the number of training data per class in situation in which they were initially highly unbalanced - thus significantly improving classification accuracy [18].

V. Challenges and opportunities

As seen in the previous sections, Riemannian approaches are providing new ways to improve reliability and robustness at diverse levels of the pattern recognition pipeline (for features, classifiers or metrics). The resulting increase in quality is at the moment mostly focused on the classification task of BCI (i.e. converting signals into commands). The calibration procedure also benefits from the geometry of SPD matrices with more robust interpolations between data providing actually usable supplementary artificial instances.

However, it must be stated that Riemannian geometry has not solved every problem of BCI and encounters itself several limitations. For example, as noticed in [17], it seems that as the number of sensors rises (and so the bigger the dimension of the covariance matrix is), the worst the accuracy becomes. This may be due to the fact that, as the dimension rises, more samples are needed to build non-singular covariance matrices. When nearly singular covariance matrices are produced, they cannot be efficiently handled via Riemannian geometry. In such a case, the Euclidean geometry would outperform the Riemannian geometry. Another limitation to be remedied in order to quit laboratory is the real-time processing of EEG signals for the methods presented in this survey. Most of Riemannian approaches involve solving a computationally demanding Riemannian optimization problem [24] that could be leveraged by the use of stochastic gradient approaches adapted to manifolds [67] when a sufficient amount of data is available. Another question is how to incorporate any prior knowledge when using Riemannian approaches. Until now, only partial answers have been given, as for example, the choice of covariance estimators specific to an application (as explained in IV-B1). Moreover, whitening and metric learning are another way of imposing the extraction of useful information from covariance matrices but this the question still has to be studied carefully.

So far, Riemannian approaches are lacking comparisons with state-of-the-art BCI approaches on several problems such as out-of-lab BCI use and transfer learning. Hence, practical and mobile BCI will confront their users with a diversity of possible tasks in a non-controlled environment, what solutions could Riemannian approaches offer? At the representation level, dictionary learning methods, as well as sparse coding are interesting tools for BCI. Indeed, as demonstrated in [68], dictionary learning techniques can be used for taking into account both temporal and spatial dimension of the signals and by doing so, it can improve the performances of a BCI system. It would be then very appealing to take the Riemannian nature of covariance matrices or subspaces in order to build dictionaries. For now, some approaches have introduced dictionary for subspaces in the computer vision community [69] and metrics for comparing dictionaries of subspaces in the BCI communities [38], [27], [28] and other works are studying the extension of dictionary learning and sparse coding on manifolds [70], [71], [72], [73], [74]. Hence, the adaptation of those approaches to BCI, either in a context of transfer learning [75] or synthetic data generation [13] - using a learned dictionary instead of a fixed dictionary- are very promising research directions.

Hidden behind continuous classification of signals is the detection of changes, either between commands/tasks or more subtle change of context, related to the non-stationarity of EEG signals. Riemannian tracking methods could perform the adaptation of the representation still taking into account the geometry. For subspaces methods and their associated manifolds, numerous methods exist by the way of subspace tracking literature, and an example of its application with CCA

to BCI tasks segmentation can be found in [26]. Similarly, statistical tests and density estimation devised with the same regard for geometry are a necessary tool for change detection.

As observed in [19], [54], adapting the whitening of a dataset of covariance matrices leads to a dramatic increase in the accuracy of a Riemannian classifier. This means that despite its interesting results, covariance matrices are prone to non-stationarity. Using an importance-weighting technique, some work has been carried out in [76] in order to have an estimator of covariance matrices robust to a kind of non-stationarity -namely covariate-shift [77]-.This approach does not consider covariance matrices in a Riemannian setting and kernel density estimation [78] could be used in order to find an importance-weighted Riemannian estimator.

Close to this idea of treating the non-stationarity in the data is the notion of robustness to outliers. Indeed, in BCI as in any pattern recognition problem, the presence of outliers is prone to deteriorate the accuracy of the whole system. One way to make a method robust to outliers consists in trimming the dataset and to exclude extreme observations. This has been experimentally done in [79], [80] for SPD matrices. A theoretical analysis of those two algorithms could potentially give rise to a family of robust classifiers.

VI. Conclusion

This paper introduced the mathematical framework of the Riemannian geometry and identified its use-case in the BCI pipeline. In particular, a typical BCI processing pipeline involves the estimation and use of covariance matrices, which can be efficiently and rigorously manipulated with Riemannian geometry. Indeed, this review has identified that Riemannian approaches can be used to either improve current subspace methods used in BCI (e.g., CSP), or completely remove the need for such subspace methods by directly manipulating and classifying covariance matrices. In this respect, our review showed that EEG signals can be represented in various ways under the form of covariance matrices for BCI design, and that there are several classifiers that can handle such covariance matrices within a Riemannian framework. Moreover, Riemannian geometry has also been used to perform metric learning, to optimize distance based EEG classification, as well as to reduce BCI calibration time.

Finally, from the identified use-cases and applications of the Riemannian geometry, this review proposed potential research directions in order to leverage the existing limitations of BCI. Notably, more research is needed to identify new EEG representations as covariance matrices, that could be more robust, as well as new Riemannian classifiers that could deal with outliers and the EEG signals non-stationarity, possibly with adaptive methods. It would also be worth studying covariance matrices dictionary learning, for representation, classification and denoising, as well as change and feature tracking. Altogether, we are convinced that Riemannian approaches are very promising for BCI design and could become, in the future, the new standard for EEG signals classification. We hope that this review could help BCI designers to understand the values and principles of these methods, and that more scientists will join the research efforts in that direction.

Acknowledgment

The authors acknowledge the support of the Inria Project Lab BCI-LIFT and the French National Research Agency (REBEL project and grant ANR-15-CE23-0013-01).

References

[1] J. Wolpaw and E. Wolpaw, *Brain-computer interfaces: principles and practice*. Oxford University Press, 2012.
[2] J. R. Wolpaw, D. J. McFarland, G. W. Neat, and C. A. Forneris, "An EEG-based brain-computer interface for cursor control," *Electroenceph clinic neurophys*, vol. 78, pp. 252–259, 1991.
[3] G. Pfurtscheller, G. Müller-Putz, R. Scherer, and C. Neuper, "Rehabilitation with brain-computer interface systems," *IEEE Comp*, vol. 41, no. 10, pp. 58–65, 2008.
[4] D. Coyle, J. Principe, F. Lotte, and A. Nijholt, "Guest editorial: Brain/neuronal computer games interfaces and interaction," *IEEE Trans Comp Intell AI Games*, vol. 5, no. 2, pp. 77–81, 2013.
[5] T. Zander and C. Kothe, "Towards passive brain-computer interfaces: applying brain-computer interface technology to human-machine systems in general," *J Neur Eng*, vol. 8, 2011.
[6] J. van Erp, F. Lotte, and M. Tangermann, "Brain-computer interfaces: Beyond medical applications," *IEEE Comp*, vol. 45, pp. 26–34, 2012.
[7] B. Blankertz, C. Sannelli, S. Halder, E. Hammer, A. Kübler, K.-R. Müller, G. Curio, and T. Dickhaus, "Neurophysiological predictor of SMR-based BCI performance," *NeuroImage*, vol. 51, no. 4, pp. 1303–1309, 2010.
[8] C. Guger, B. Z. Allison, B. Großwindhager, R. Prückl, C. Hintermüller, C. Kapeller, M. Bruckner, G. Krausz, and G. Edlinger, "How many people could use an SSVEP BCI?" *Front Neurosc*, vol. 6, 2012.
[9] M. Fatourechi, A. Bashashati, R. Ward, and G. Birch, "EMG and EOG artifacts in brain computer interface systems: A survey," *Clin Neurophys*, vol. 118, pp. 480–494, 2007.
[10] D. Krusienski, M. Grosse-Wentrup, F. Galán, D. Coyle, K. Miller, E. Forney, and C. Anderson, "Critical issues in state-of-the-art braincomputer interface signal processing," *J Neur Eng*, vol. 8, no. 2, p. 025002, 2011.
[11] C. Mühl, C. Jeunet, and F. Lotte, "EEG-based workload estimation across affective contexts," *Front Neurosc*, vol. 8, p. 114, 2014.
[12] S. Brandl, J. Hohne, K.-R. Muller, and W. Samek, "Bringing BCI into everyday life: Motor imagery in a pseudo realistic environment," in *Proc IEEE NER*. IEEE, 2015, pp. 224–227.
[13] F. Lotte, "Signal processing approaches to minimize or suppress calibration time in oscillatory activity-based brain–computer interfaces," *Proc IEEE*, 2015.
[14] F. Lotte and C. Jeunet, "Towards improved BCI based on human learning principles," in *Proc 3rd Int BCI Winter Conf*, 2015.
[15] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Multiclass brain–computer interface classification by riemannian geometry," *IEEE Trans Biomed Eng*, vol. 59, no. 4, pp. 920–928, 2012.
[16] M. Congedo, A. Barachant, and A. Andreev, "A new generation of brain-computer interface based on riemannian geometry," *arXiv:1310.8115*, 2013.
[17] F. Yger, F. Lotte, and M. Sugiyama, "Averaging covariance matrices for EEG signal classification based on the CSP: an empirical study," in *Proc. EUSIPCO*, 2015.
[18] E. Kalunga, S. Chevallier, and Q. Barthélemy, "Data augmentation in riemannian space for brain-computer interfaces," in *Proc. ICML Workshop Stamlins*, 2015.
[19] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Classification of covariance matrices using a riemannian-based kernel for BCI applications," *Neurocomp*, vol. 112, pp. 172–178, 2013.
[20] I. Horev, F. Yger, and M. Sugiyama, "Geometry-aware principal component analysis for symmetric positive definite matrices," in *Proc. ACML*, 2015.
[21] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K.-R. Müller, "Optimizing spatial filters for robust EEG single-trial analysis," *IEEE Sig Proc Mag*, vol. 25, no. 1, pp. 41–56, 2008.
[22] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi, "A review of classification algorithms for EEG-based brain-computer interfaces," *J Neur Eng*, vol. 4, pp. R1–R13, 2007.
[23] S. Lang, *Differential and Riemannian Manifolds*, ser. Graduate Texts in Mathematics. Springer, 1995.
[24] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2009.
[25] R. Bhatia, *Positive Definite Matrices*. Princeton University Press, 2009.

[26] F. Yger, M. Berar, G. Gasso, and A. Rakotomamonjy, "Adaptive canonical correlation analysis based on matrix manifolds," in *Proc. ICML*, 2012.

[27] S. Chevallier, Q. Barthélemy, and J. Atif, "On the need for metrics in dictionary learning assessment," in *Proc. EUSIPCO*, 2014.

[28] ——, "Subspace metrics for multivariate dictionaries and application to EEG," in *proc. ICASSP*, 2014.

[29] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM J Matrix Anal Appl*, vol. 20, no. 2, pp. 303–353, 1998.

[30] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Geometric means in a novel vector space structure on symmetric positive-definite matrices," *SIAM J Mat Anal Appl*, vol. 29, no. 1, pp. 328–347, 2007.

[31] P. T. Fletcher, C. Lu, S. M. Pizer, and S. Joshi, "Principal geodesic analysis for the study of nonlinear statistics of shape," *IEEE Trans Med Imag*, vol. 23, no. 8, pp. 995–1005, 2004.

[32] S. Sommer, F. Lauze, S. Hauberg, and M. Nielsen, "Manifold valued statistics, exact principal geodesic analysis and the effect of linear approximations," in *Proc. ECCV*, 2010.

[33] P. T. Fletcher, S. Venkatasubramanian, and S. Joshi, "The geometric median on riemannian manifolds with application to robust atlas estimation," *NeuroImage*, vol. 45, no. 1, pp. S143–S152, 2009.

[34] M. Moakher, "A differential geometric approach to the geometric mean of symmetric positive-definite matrices," *SIAM J Mat Anal Appl*, vol. 26, no. 3, pp. 735–747, 2005.

[35] B. Jeuris, R. Vandebril, and B. Vandereycken, "A survey and comparison of contemporary algorithms for computing the matrix geometric mean," *Elec Trans Num Anal*, vol. 39, pp. 379–402, 2012.

[36] D. A. Bini and B. Iannazzo, "Computing the karcher mean of symmetric positive definite matrices," *Linear Algebra and its Applications*, vol. 438, no. 4, pp. 1700–1710, 2013.

[37] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, "Manopt, a Matlab toolbox for optimization on manifolds," *J Mach Learn Res*, vol. 15, pp. 1455–1459, 2014.

[38] S. Chevallier, Q. Barthélemy, and J. Atif, "Metrics for multivariate dictionaries," *arXiv:1302.4242*, 2013.

[39] W. Samek, M. Kawanabe, and K.-R. Muller, "Divergence-based framework for common spatial patterns algorithms," *IEEE Rev Biomed Eng*, vol. 7, pp. 50–72, 2014.

[40] X. Li, C. Guan, K. K. Ang, H. Zhang, and S. H. Ong, "Spatial filter adaptation based on geodesic-distance for motor EEG classification," in *Proc. IJCNN*, 2014, pp. 3859–3864.

[41] P. Rousseeuw and K. Van Driessen, "A fast algorithm for the minimum covariance determinant estimator," *Technometrics*, 1999.

[42] O. Ledoit and M. Wolf, "A well-conditioned estimator for large-dimensional covariance matrices," *J. Multivar. An.*, 2004.

[43] W. Samek and M. Kawanabe, "Robust common spatial patterns by minimum divergence covariance estimator," in *Proc. ICASSP*, 2014.

[44] D. Bartz and K.-R. Müller, "Covariance shrinkage for autocorrelated data," in *Proc. NIPS*, 2014.

[45] R. Tomioka and K. Aihara, "Classifying matrices with a spectral regularization," in *Proc. ICML*, 2007.

[46] N. Tomida, M. Yamagishi, I. Yamada, and T. Tanaka, "A reduced rank approach for covariance matrix estimation in EEG signal classification," in *Proc. EMBC*, 2014.

[47] L. Roijendijk, S. Gielen, and J. Farquhar, "Classifying regularised sensor covariance matrices: an alternative to CSP," *IEEE Trans Neural Syst Rehab*, 2015.

[48] X. Navarro-Sune, A. Hudson, F. Fallani, J. Martinerie, A. Witon, P. Pouget, M. Raux, T. Similowski, and M. Chavez, "Riemannian geometry applied to detection of respiratory states from EEG signals: the basis for a brain-ventilator interface," *arXiv:1601.03022*, 2016.

[49] A. Barachant and M. Congedo, "A plug&play P300 BCI using information geometry," *arXiv:1409.0107*, 2014.

[50] E. Kalunga, S. Chevallier, Q. Barthélemy, K. Djouani, Y. Hamam, and E. Monacelli, "From euclidean to riemannian means: Information geometry for SSVEP classification," in *Geometric Science of Information*. Springer, 2015, pp. 595–604.

[51] A. Bellet, A. Habrard, and M. Sebban, "A survey on metric learning for feature vectors and structured data," *arXiv:1306.6709*, 2013.

[52] M. Harandi, M. Salzmann, and R. Hartley, "From manifold to manifold: geometry-aware dimensionality reduction for SPD matrices," in *Proc. ECCV*, 2014.

[53] A. Schlögl, F. Lee, H. Bischof, and G. Pfurtscheller, "Characterization of four-class motor imagery EEG data for the BCI-competition 2005," *J Neur Eng*, vol. 2, no. 4, p. L14, 2005.

[54] F. Yger and M. Sugiyama, "Supervised logeuclidean metric learning for symmetric positive definite matrices," *arXiv:1502.03505*, 2015.

[55] C. Cortes, M. Mohri, and A. Rostamizadeh, "Algorithms for learning kernels based on centered alignment," *JMLR*, vol. 13, no. 1, pp. 795–828, 2012.

[56] N. Cristianini, A. Elisseeff, J. Shawe-Taylor, and J. Kandola, "On kernel-target alignment," in *Proc. NIPS*, 2001.

[57] M. Ramona, G. Richard, and B. David, "Multiclass feature selection with kernel gram-matrix based criteria," *IEEE Trans Neural Netw Learn Syst*, vol. 23, no. 10, Month 2012.

[58] F. Yger, "A review of kernels on covariance matrices for BCI applications," in *Proc. MLSP*, 2013.

[59] M. Naeem, C. Brunner, R. Leeb, B. Graimann, and G. Pfurtscheller, "Seperability of four-class motor imagery data using independent components analysis," *J Neur Eng*, vol. 3, no. 3, p. 208, 2006.

[60] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans Neur Netw*, vol. 12, no. 2, pp. 181–201, 2001.

[61] B. Scholkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[62] B. Haasdonk and C. Bahlmann, "Learning with distance substitution kernels," in *Pattern Rec*. Springer, 2004, pp. 220–227.

[63] A. Feragen, F. Lauze, and S. Hauberg, "Geodesic exponential kernels: When curvature and linearity conflict," in *Proc. CVPR*, 2015, pp. 3032–3042.

[64] S. Sra, "Positive definite matrices and the S-divergence," *arXiv preprint arXiv:1110.1773*, 2011.

[65] S. Jayasumana, R. Hartley, and M. Salzmann, "Kernels on riemannian manifolds," in *Riemannian Computing in Computer Vision*. Springer, 2016, pp. 45–67.

[66] F. Lotte, L. Bougrain, and M. Clerc, "Electroencephalography (EEG)-based brain-computer interfaces," in *Wiley Encycl Electric Electro Eng*. Wiley, 2015.

[67] S. Bonnabel, "Stochastic gradient descent on riemannian manifolds," *IEEE Trans Auto Control*, vol. 58, no. 9, pp. 2217–2229, 2013.

[68] Q. Barthélemy, C. Gouy-Pailler, Y. Isaac, A. Souloumiac, A. Larue, and J. I. Mars, "Multivariate temporal dictionary learning for EEG," *J Neurosci Meth*, 2013.

[69] M. Harandi, R. Hartley, C. Shen, B. Lovell, and C. Sanderson, "Extrinsic methods for coding and dictionary learning on grassmann manifolds," *Int J Comp Vis*, vol. 114, no. 2-3, pp. 113–136, 2015.

[70] A. Cherian and S. Sra, "Riemannian sparse coding for positive definite matrices," in *Proc. ECCV*. Springer, 2014, pp. 299–314.

[71] ——, "Riemannian dictionary learning and sparse coding for positive definite matrices," *arXiv preprint arXiv:1507.02772*, 2015.

[72] M. Harandi and M. Salzmann, "Riemannian coding and dictionary learning: Kernels to the rescue," in *Proc. CVPR*, 2015, pp. 3926–3935.

[73] M. Harandi, R. Hartley, B. Lovell, and C. Sanderson, "Sparse coding on symmetric positive definite manifolds using bregman divergences," *IEEE Trans Neural Netw Learn Syst*, 2015.

[74] S. Zhang, S. Kasiviswanathan, P. C. Yuen, and M. Harandi, "Online dictionary learning on symmetric positive definite manifolds with vision applications." in *Proc. AAAI*, 2015, pp. 3165–3173.

[75] H. Morioka, A. Kanemura, J.-i. Hirayama, M. Shikauchi, T. Ogawa, S. Ikeda, M. Kawanabe, and S. Ishii, "Learning a common dictionary for subject-transfer decoding with resting calibration," *NeuroImage*, 2015.

[76] A. Balzi, F. Yger, and M. Sugiyama, "Importance-weighted covariance estimation for robust common spatial pattern," *Patt Rec Lett*, vol. 68, pp. 139–145, 2015.

[77] M. Sugiyama and M. Kawanabe, *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT Press, 2012.

[78] E. Chevallier, E. Kalunga, and J. Angulo, "Kernel density estimation on spaces of gaussian distributions and symmetric positive definite matrices," *hal-archive*, 2015.

[79] A. Barachant, A. Andreev, and M. Congedo, "The riemannian potato: an automatic and adaptive artifact detection method for online experiments using riemannian geometry," in *TOBI Workshop IV*, 2013, pp. 19–20.

[80] T. Uehara, T. Tanaka, and S. Fiori, "Robust averaging of covariance matrices by riemannian geometry for motor-imagery brain–computer interfacing," in *Advances in Cognitive Neurodynamics*. Springer, 2016, pp. 347–353.