Neurocomputing 112 (2013) 172–178

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Classification of covariance matrices using a Riemannian-based kernel for BCI applications

Alexandre Barachant [a,b,*], Stéphane Bonnet [a], Marco Congedo [b], Christian Jutten [b]

[a] CEA-LETI, MINATEC Campus, 17 rue des Martyrs, F-38054 Grenoble, France
[b] Team ViBS (Vision and Brain Signal Processing), GIPSA-Lab, CNRS, Grenoble Universities, Domaine Universitaire, F-38402 Saint Martin d'Hères, France

ARTICLE INFO

Available online 7 March 2013

Keywords:
Brain–computer interfaces
Covariance matrix
Kernel
Support vector machine
Riemannian geometry

ABSTRACT

The use of spatial covariance matrix as a feature is investigated for motor imagery EEG-based classification in brain–computer interface applications. A new kernel is derived by establishing a connection with the Riemannian geometry of symmetric positive definite matrices. Different kernels are tested, in combination with support vector machines, on a past BCI competition dataset. We demonstrate that this new approach outperforms significantly state of the art results, effectively replacing the traditional spatial filtering approach.

1. Introduction

Brain–computer interfaces (BCIs) based on motor imagery have been well studied in the literature. For this type of BCI, the electrophysiological source of BCI control is based on spontaneous signals induced by voluntary changes in the dynamics of brain oscillations. In this context, *motor imagery* (MI), the kinaesthetic imagination of actual body movement, results in the event related synchronization/desynchronization (which are defined for specific band-pass regions) of large cortical patches in the sensorimotor cortex in the μ and β frequency bands [14].

The standard approach in MI-based EEG signal classification is to perform band-pass filtering, spatial filtering and linear classification, generally using Fisher's Linear Discriminant Analysis (LDA). The most popular spatial filtering algorithm is named Common Spatial Pattern (CSP) [15]. This spatial filtering algorithm can be seen as a data-driven dimension reduction method that aims at promoting variance differences between two conditions. In this fashion covariance matrices are handled in the Euclidean space without considerations about the curvature of the space of symmetric positive definite (SPD) matrices to which they belong.

This paper provides a simple way to take into account the Riemannian geometry for EEG signal classification. This approach has been successfully applied in the past on radar signal processing and image processing [19]. Furthermore, a new kernel is derived by establishing a connection with the Riemannian geometry of SPD matrices. Similar approaches have been applied [9,20], leading to the definition of different kernels depending on the Riemannian metric considered.

This kernel is tested in combination with support vector machines, although we could have applied the kernel trick to other classifiers like logistic regression. The presented results demonstrate the benefit of the proposed approach. The distinct advantage of the present method is that it can be applied directly, avoiding the need of spatial filtering [2].

The paper is organized as follows: In Section 2, a new kernel for covariance matrices, based on the Riemannian geometry, is introduced. In Section 3, this kernel is applied in the context of SVM classification. In Section 4, results on BCI competition dataset are provided. Finally, Section 5 concludes on the benefits of this method. This paper is an extended version of the work presented in Barachant et al. [3].

2. A new kernel for symmetric positive definite matrices

2.1. Introduction

EEG signals are often analyzed on short-time segments called trials. Let $\mathbf{X} \in R^{E \times T}$ be such a trial, with E being the number of electrodes and T the epoch duration expressed in the number of samples. We further assume that the different EEG signals have been band-pass filtered, for instance using a filter between 8 and 35 Hz in order to take into account the μ (8–14 Hz) and β (14–30 Hz) frequency bands.

For each trial $\mathbf{X}_p$ of known class $y_p \in \{-1,1\}$, one can estimate the spatial covariance matrix of the EEG random signal by the

[*] Corresponding author at: Team ViBS (Vision and Brain Signal Processing), GIPSA-Lab, CNRS, Grenoble Universities, Domaine Universitaire, F-38402 Saint Martin d'Hères, France. Tel.: +33 65 1690611.
E-mail address: alexandre.barachant@gmail.com (A. Barachant).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.12.039

$E \times E$ sample covariance matrix (SCM): $\mathbf{C}_p = 1/(T-1)\mathbf{X}_p\mathbf{X}_p^T$. The space of SPD $E \times E$ matrices will be denoted $P(E)$ in the rest of the paper. Moreover, note that SCM is sensitive to outliers so that either robust covariance estimation techniques or regularization can be applied to improve the estimator [11].

It is a common practice in MI-based BCI to use spatial filtering for dimension reduction and variance ratio enhancement between EEG trials coming from different motor classes [6]. The log-variances of the spatially filtered signals are then used as input features for a linear classification, usually achieved by a LDA. The Common Spatial Patterns (CSP) [15] are for instance successfully applied as a method to extract relevant features for the classification of EEG trials recorded during two motor imagery tasks. This technique aims at simultaneously diagonalizing the two intra-class covariance matrices obtained in the two conditions. This observation motivated us to investigate the direct use of spatial covariance matrix as input feature for EEG-based BCI signal classification.

2.2. Spatial covariance matrix as feature: a direct approach

If one is interested in using a covariance matrix as a feature in a classifier, a natural choice consists in vectorizing it in order to process this quantity as a vector and then use any vector-based classification algorithms. Due to symmetry, we consider the following modified half-vectorization operator that stacks, with appropriate weighting, the upper triangular part of $\mathbf{C} \in P(E)$ into a $(E+1)E/2 \times 1$ column vector:

$$\mathrm{vect}(\mathbf{C}) = [C_{1,1}; \sqrt{2}C_{1,2}; C_{2,2}; \sqrt{2}C_{1,3}; \sqrt{2}C_{2,3}; C_{3,3}; \ldots ; C_{E,E}] \tag{1}$$

Without loss of generality, a $\sqrt{2}$ coefficient is applied on the non-diagonal elements of $\mathbf{C}$ in order to conserve equality of norms $\|\mathbf{C}\|_F = \|\mathrm{vect}(\mathbf{C})\|_2$. The reverse operation is defined in a straightforward manner by unvect($\mathbf{x}$). Such approach has been for instance investigated in Farquhar [8] and Reuderink et al. [16]. In their work, they have demonstrated that CSP-like spatial filtering followed by linear classification can be performed in one single step in a high-dimensional space by considering the vectorized form of the covariance matrix as feature for classification. Indeed, the classification score function $h(\cdot)$ obtained by applying the linear classifier $(\mathbf{u},b)$ on the temporal variances $\boldsymbol{\sigma}^2$ of the spatially filtered EEG signals, using the spatial filter $\mathbf{W}$, can be rewritten as

$$\begin{aligned} h(\boldsymbol{\sigma}^2) &= \langle \mathbf{u},\boldsymbol{\sigma}^2 \rangle + b = \sum_k u_k \mathbf{w}_k^T \mathbf{C} \mathbf{w}_k + b \\ &= \mathrm{tr}(\mathrm{diag}(\mathbf{u})\mathbf{W}^T\mathbf{C}\mathbf{W}) + b \\ &= \langle \mathbf{U},\mathbf{C} \rangle_F + b \end{aligned} \tag{2}$$

with $\mathbf{U} = \mathbf{W}\,\mathrm{diag}(\mathbf{u})\mathbf{W}^T$, tr the trace operator and $\langle \cdot , \cdot \rangle_F$ the Frobenius inner product. This equation results in the definition of a new linear classifier (vect($\mathbf{U}$),b) which can be directly applied on the vectorized covariance matrices vect($\mathbf{C}$).

2.3. Spatial covariance matrix as feature: a kernel approach

Riemannian tools: In this paper, we propose a kernel approach that will operate differently on the spatial covariance matrices. A spatial covariance matrix is by construction symmetric and if sufficient data have been used to estimate it, it will also be positive definite.

The direct vectorization of these matrices do not take into account the relationships that link between them the coefficients of the symmetric and positive definite (SPD) matrices. In addition, the vectorized covariance matrices do not follow a normal distribution, so that a large number of popular classification algorithms, like the Linear Discriminant Analysis (LDA) which is optimal for Gaussian distributions, are less effective.

We note that the space of SPD $E \times E$ matrices $P(E)$ forms a differentiable manifold $\mathcal{M}$ of dimension $E^\star = E\,(E+1)/2$. Thus, the correct manipulation of these matrices relies on a special branch of differential geometry, namely the Riemannian geometry [5]. The proposed method consists in taking into account the Riemannian geometry of the space of covariance matrices $P(E)$. Indeed at each point $\mathbf{C}$ (i.e. each covariance matrix in our case) of the manifold $\mathcal{M}$, a scalar product can be defined in the associated *tangent space* $\mathcal{T}_{\mathbf{C}}\mathcal{M}$. This tangent space is Euclidean and locally homomorphic to the manifold and Riemannian distance computations in the manifold can be well approximated by Euclidean distance computations in the tangent space. Let $\mathbf{S}_1$ and $\mathbf{S}_2$ be the two tangent vectors[1] (i.e. two symmetric matrices in our case), the scalar product in the tangent space at $\mathbf{C}$ can be defined by the relation:

$$\langle \mathbf{S}_1,\mathbf{S}_2 \rangle_{\mathbf{C}} = \mathrm{tr}(\mathbf{S}_1\mathbf{C}^{-1}\mathbf{S}_2\mathbf{C}^{-1}). \tag{3}$$

Furthermore, the *logarithmic map* projects locally all covariance matrices $\{\mathbf{C}_p\}_{p=1}^{P}$, onto the tangent plane by

$$\mathbf{S}_p = \mathrm{Log}_{\mathbf{C}}(\mathbf{C}_p) = \mathbf{C}^{1/2}\ \mathrm{logm}(\mathbf{C}^{-1/2}\mathbf{C}_p\mathbf{C}^{-1/2})\mathbf{C}^{1/2} \tag{4}$$

where logm$(\cdot)$ denotes the logarithm of a matrix [5]. The logarithm of a diagonalizable matrix $\mathbf{A} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1}$ is defined as $\mathrm{logm}(\mathbf{A}) = \mathbf{V}\mathbf{D}'\mathbf{V}^{-1}$, where the diagonal elements are given by $d'_{i,i} = \log(d_{i,i})$. Furthermore, the inverse operation that projects an element of the tangent space back to the manifold, namely the *exponential map*, is defined by

$$\mathbf{C}_p = \mathrm{Exp}_{\mathbf{C}}(\mathbf{S}_p) = \mathbf{C}^{1/2}\ \mathrm{expm}(\mathbf{C}^{-1/2}\mathbf{S}_p\mathbf{C}^{-1/2})\mathbf{C}^{1/2} \tag{5}$$

where expm$(\cdot)$ denotes the exponential of a matrix. Fig. 1 illustrates this process.

Riemannian-based kernel: According to these Riemannian geometry tools, it is possible to project locally each covariance matrix into the tangent plane, and use this new space to manipulate the projected covariance matrices. For the direct manipulation of covariance matrices in the manifold, refer to Barachant et al. [4].

Let us consider the scalar product, which is defined in this tangent plane at $\mathbf{C}_{\mathrm{ref}}$, and that is given by (3). We propose to use the following explicit mapping function on the covariance matrices:

$$\phi(\mathbf{C}) = \mathrm{Log}_{\mathbf{C}_{\mathrm{ref}}}(\mathbf{C}) \tag{6}$$

together with the Riemannian-based kernel

$$\begin{aligned} k_R(\mathrm{vect}(\mathbf{C}_i),\mathrm{vect}(\mathbf{C}_j);\mathbf{C}_{\mathrm{ref}}) &= \langle \phi(\mathbf{C}_i),\phi(\mathbf{C}_j) \rangle_{\mathbf{C}_{\mathrm{ref}}} \\ &= \mathrm{tr}[\mathrm{Log}_{\mathbf{C}_{\mathrm{ref}}}(\mathbf{C}_i)\mathbf{C}_{\mathrm{ref}}^{-1}\ \mathrm{Log}_{\mathbf{C}_{\mathrm{ref}}}(\mathbf{C}_j)\mathbf{C}_{\mathrm{ref}}^{-1}] \\ &= \mathrm{tr}[\mathrm{logm}(\mathbf{C}_{\mathrm{ref}}^{-1/2}\mathbf{C}_i\mathbf{C}_{\mathrm{ref}}^{-1/2})\ \mathrm{logm}(\mathbf{C}_{\mathrm{ref}}^{-1/2}\mathbf{C}_j\mathbf{C}_{\mathrm{ref}}^{-1/2})]. \end{aligned} \tag{7}$$

Since a valid scalar product is used, Mercer's condition that each kernel should respect is verified. Using trace definition, the proposed kernel defined in (7) can be reformulated conveniently as follows:

$$\begin{aligned} k_R(\mathrm{vect}(\mathbf{C}_i),\mathrm{vect}(\mathbf{C}_j);\mathbf{C}_{\mathrm{ref}}) &= \mathrm{tr}[\mathbf{C}_{\mathrm{ref}}^{-1/2}\ \mathrm{Log}_{\mathbf{C}_{\mathrm{ref}}}(\mathbf{C}_i)\mathbf{C}_{\mathrm{ref}}^{-1/2}\mathbf{C}_{\mathrm{ref}}^{-1/2}\ \mathrm{Log}_{\mathbf{C}_{\mathrm{ref}}}(\mathbf{C}_j)\mathbf{C}_{\mathrm{ref}}^{-1/2}] \\ &= \langle \tilde{\mathbf{S}}_i,\tilde{\mathbf{S}}_j \rangle_F \\ &= \mathrm{vect}(\tilde{\mathbf{S}}_i)^T\mathrm{vect}(\tilde{\mathbf{S}}_j) \end{aligned} \tag{8}$$

where we have introduced the symmetric matrix

$$\tilde{\mathbf{S}}_i = \mathbf{C}_{\mathrm{ref}}^{-1/2}\ \mathrm{Log}_{\mathbf{C}_{\mathrm{ref}}}(\mathbf{C}_i)\mathbf{C}_{\mathrm{ref}}^{-1/2} = \mathrm{logm}(\mathbf{C}_{\mathrm{ref}}^{-1/2}\mathbf{C}_i\mathbf{C}_{\mathrm{ref}}^{-1/2}). \tag{9}$$

We note that this equivalence allows us another possible kernel-free implementation of the proposed approach by first transforming covariance matrices $\{\mathbf{C}_p\}_{p=1}^{P}$ into symmetric matrices $\tilde{\mathbf{S}}_p$,

[1] An element of the tangent space is usually called tangent vector.

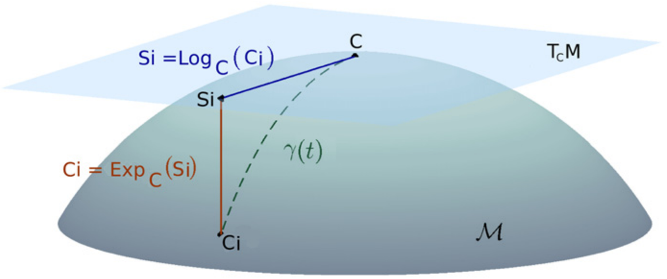

Fig. 1. Manifold $\mathcal{M}$ and the corresponding local tangent space $\mathcal{T}_{\mathbf{C}}\mathcal{M}$ at $\mathbf{C}$. The logarithmic map $\mathrm{Log}_{\mathbf{C}}(\cdot)$ projects the matrix $\mathbf{C}_i \in \mathcal{M}$ into the tangent space. The exponential map $\mathrm{Exp}_{\mathbf{C}}(\cdot)$ projects the element of the tangent space $\mathbf{S}_i$ back to the manifold.

using Eq. (9) and then performing linear classification on the half-vectorized form of these matrices. We will come back to this point in Section 3.2.

2.4. The choice of a reference SPD matrix

$\mathbf{C}_{\text{ref}}$ is a free parameter of the proposed kernel that defines the point in the manifold $\mathcal{M}$ where the tangent plane is computed. The choice $\mathbf{C}_{\text{ref}} = \mathbf{I}_E$, where $\mathbf{I}_E$ denotes the identity matrix of size $E \times E$, yields for instance the log-Euclidean kernel, denoted k_{LE}:

$$\begin{aligned} k_{LE}(\text{vect}(\mathbf{C}_i),\text{vect}(\mathbf{C}_j)) &= k_R(\text{vect}(\mathbf{C}_i),\text{vect}(\mathbf{C}_j);\mathbf{I}_E) \\ &= \text{tr}[\text{logm}(\mathbf{C}_i)\ \text{logm}(\mathbf{C}_j)] \end{aligned} \tag{10}$$

This kernel is again interpreted as a Frobenius inner product and has been investigated in Wang et al. [20].

Another choice for the reference matrix $\mathbf{C}_{\text{ref}}$ is the average of the whole set of covariance matrices. A practical choice is to consider the arithmetic mean of the P labelled covariance matrices $\{\mathbf{C}_p\}_{p=1}^{P}$:

$$\mathfrak{A}(\mathbf{C}_1,\ldots,\mathbf{C}_P) = \frac{1}{P}\sum_{p=1}^{P}\mathbf{C}_p \tag{11}$$

In this paper, we propose to use the *geometric mean* introduced in [13]. Such a matrix is defined as follows:

$$\mathfrak{G}(\mathbf{C}_1,\ldots,\mathbf{C}_P) = \underset{\mathbf{C}}{\text{argmin}}\sum_{p=1}^{P}\delta_R^2(\mathbf{C},\mathbf{C}_p) \tag{12}$$

The Riemannian distance between two SPD matrices is defined as

$$\delta_R(\mathbf{C}_1,\mathbf{C}_2) = \|\log(\mathbf{C}_1^{-1}\mathbf{C}_2)\|_F = \left[\sum_{i=1}^{E}\log^2\lambda_i\right]^{1/2} \tag{13}$$

where $\{\lambda_i\}_{i=1}^{E}$ are the real eigenvalues of $\mathbf{C}_1^{-1}\mathbf{C}_2$. Note the connection between the Riemannian distance and the CSP [1].

As mentioned in Tuzelarticle et al. [19], the geometric mean is the point where the mapping on the tangent space leads to the better local approximation of the manifold. This observation motivated our choice of the geometric mean for the reference point.

The geometric mean $\mathfrak{G}(\mathbf{C}_1,\ldots,\mathbf{C}_P)$ can be computed efficiently by an iterative procedure consisting in projecting the covariance matrices in the tangent space, estimating the arithmetic mean in the tangent space and projecting the arithmetic mean back in the manifold. Then iterate the three above steps until convergence.

The full algorithm, derived from Moakher et al. [13], is given by the Algorithm 1.

Algorithm 1. Mean of P SPD matrices.

```
Input: $\Omega$ a set of $P$ SPD matrices $\mathbf{C}_p \in P(E)$ and $\epsilon > 0$.
Output: $\mathfrak{G}$ the estimated geometric mean in $P(E)$ of $\Omega$.
1. Initialize $\mathfrak{G}^{(1)} = \frac{1}{P}\sum_{p=1}^{P}\mathbf{C}_p$
2: repeat
3:     $\overline{\mathbf{S}} = \frac{1}{P}\sum_{p=1}^{P}\mathrm{Log}_{\mathfrak{G}^{(t)}}(\mathbf{C}_p)$ {Arithmetic mean in the tangent space}
4:     $\mathfrak{G}^{(t+1)} = \mathrm{Exp}_{\mathfrak{G}^{(t)}}(\overline{\mathbf{S}})$
5: until $\|\overline{\mathbf{S}}\|_F < \epsilon$
6: return $\mathfrak{G}^{(t+1)}$
```

2.5. Equivalence between kernels

Suppose that the trials $\mathbf{X}_p$ have been transformed so that $\tilde{\mathbf{C}}_p = \mathbf{C}_{\text{ref}}^{-1/2}\mathbf{C}_p\mathbf{C}_{\text{ref}}^{-1/2}$. This transformation implies that the trials have been spatially whitened according to $\tilde{\mathbf{X}}_p = \mathbf{C}_{\text{ref}}^{-1/2}\mathbf{X}_p$. Then it is easy to prove that

$$\begin{aligned} k_R(\text{vect}(\mathbf{C}_i),\text{vect}(\mathbf{C}_j);\mathbf{C}_{\text{ref}}) &= \text{tr}[\text{logm}(\tilde{\mathbf{C}}_i)\ \ \text{logm}(\tilde{\mathbf{C}}_j)] \\ &= k_R(\text{vect}(\tilde{\mathbf{C}}_i),\text{vect}(\tilde{\mathbf{C}}_j);\mathbf{I}_E) \end{aligned} \tag{14}$$

This result shows that it is equivalent to either apply Riemannian-kernel on native covariance matrices $k_R(\cdot,\cdot;\mathbf{C}_{\text{ref}})$ or apply Riemannian-kernel on “whitened” trials $k_R(\cdot,\cdot;\mathbf{I}_E)$. Interestingly, this approach was considered without justification in Reuderink et al. [16] where observations were first whitened before applying the classification on the vectorized covariance matrices.

3. Riemannian-based kernel in SVM classification

3.1. SVM formulation

Support Vector Machine (SVM) is a popular linear classifier in BCI applications [12]. Given a set of labelled feature vectors $\{(\mathbf{x}_p,y_p)\}_{p=1}^{P}$, this classification technique seeks to separate data by finding a hyperplane (with normal vector $\mathbf{w}$) that maximizes the margin, i.e. the distance between the hyperplane and the nearest points from each class, called *support vectors*. SVM is known to possess good generalization properties and to perform well in high-dimensional feature space. We refer to [17] and references herein for detailed discussion on SVM.

The decision function will be based on the sign of

$$h(\mathbf{x}) = b + \sum_{p=1}^{P}\alpha_p y_p\langle\mathbf{x}_p,\mathbf{x}\rangle = b + \langle\mathbf{w},\mathbf{x}\rangle \tag{15}$$

where $\langle\cdot,\cdot\rangle$ denotes the Euclidean scalar product between vectors. The term b is a bias and the $\{\alpha_p\}_{p=1}^{P}$ are the Lagrangian multipliers associated to the dual optimization problem [17]. Both quantities are estimated by quadratic programming. Most of the α_p's are null except for the support vectors. If data are not separable in their native space, a mapping can be applied on the feature vector $\mathbf{x}$ to another (high-dimensional) transformed space in which they are linearly separable. The transformation ϕ is generally non-linear and the decision function can be rewritten as

$$h(\mathbf{x}) = b + \sum_{p=1}^{P}\alpha_p y_p\langle\phi(\mathbf{x}_p),\phi(\mathbf{x})\rangle_{\mathcal{H}} \tag{16}$$

where $\mathcal{H}$ is a reproducing kernel Hilbert space where the dot product is defined. The associated kernel $k(\cdot,\cdot)$ is defined by $k(\mathbf{x}_i,\mathbf{x}_j) = \langle\phi(\mathbf{x}_i),\phi(\mathbf{x}_j)\rangle_{\mathcal{H}}$. In most SVM applications, the function ϕ is not explicitly expressed and solely the kernel is used. A usual kernel is the Gaussian kernel: $k(\mathbf{x}_i,\mathbf{x}_j) = \exp[-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2]$. Finally the margin maximization cost function is often penalized with the introduction of slack variables whose effects are controlled by a hyper-parameter, named λ. This allows us to define soft margins where some training feature vectors can be reasonably misclassified.

3.2. SVM applied on covariance matrix

According to the previous section, we will use the Riemannian kernel in the context of SVM classification. Since the $\phi(\cdot)$ transformation is known, Eq. (6), two equivalent implementations are either

1. half-vectorize spatial covariance matrices and apply on them the Riemannian-based kernel SVM, defined in (7) or
2. transform spatial covariance matrices into new matrices according to (9), half-vectorize these symmetric matrices and apply the linear SVM on them.

Both implementations will produce the same result and the best formulation is a matter of convenience depending upon the chosen SVM library. In this paper, we use the SVM and Kernel Matlab toolbox written by Canu et al. [7] with the second implementation.

3.3. Adaptive kernel formulation

In the context of BCI it is well known that there is a large variability in the feature distributions between sessions, i.e. records done on different days [18]. This variability should be taken into account when we apply a classifier trained on one session, for example the training session, to another session, for example the test session. This adaptation should be done in an unsupervised way and is usually achieved by adjusting the bias of the classifier [10]. Our main point is that within the Riemannian framework this adaptation is easily achieved by updating the reference point used for the tangent space mapping to fit the new data.

Fig. 2 shows the effect of this adaptation on the feature distribution in the tangent space. On the left, the feature distribution is represented for both classes with continuous lines (first session) and dashed lines (second session). We can clearly see a shift on the distribution means and a rotation of the ellipsoids. On the right, the same distributions are shown, but the reference point used for the second session is estimated with the data of the second session. This adaptation allows us to correct the shift between the distributions of the two sessions. As a consequence, we can use the same classifier for two different sessions without loss of performance.

A simple way to handle the variability between sessions is to estimate at the beginning of the session a new reference point, by arithmetic or geometric mean, projects data from the second session to a new tangent space and apply the classifier with unchanged parameters α_p and b. Since the support vectors should be projected using the reference point of the first session, this adaptation is easier to implement using the second implementation described in Section 3.2.

In other words, a linear SVM classifier is trained on features composed by the tangent space in $\mathbf{C}_{\text{ref,session:1}}$ representing data from the first session, and applied on features composed by the tangent space in $\mathbf{C}_{\text{ref,session:2}}$ representing data from the second session.

This will account for, among other sources of variability, the difference in electrode placement, the difference in baseline mental state, etc. As explained in Section 2.5, the mapping of data to the tangent space at the reference point can be interpreted as a whitening operation. Therefore, this is another important message of this paper, the adaptation of the reference point is equivalent to perform an adaptive whitening operation on the data. This adaptation is similar to the heuristic one presented by [16] where data are first whitened by covariance matrices estimated during previous trials.

4. Experiments

In order to evaluate the performance of the proposed method, we have compared the results to those obtained with the classical signal processing chain in MI-based BCI. This standard approach consists in band-pass filtering, spatial filtering (using four pairs of CSP spatial filters), log-variance feature extraction and Fisher's LDA classification [6]. This method is compared to linear SVM applied on half-vectorized covariance matrices (SVM vec) and to kernel-based SVMs (RK-SVM) according to (7). Three tangent planes have been considered: the first one at identity $\mathbf{I}_E$, the second one at the arithmetic mean $\mathfrak{A}$ and the last one at the geometric mean $\mathfrak{G}$. In addition, results for the adaptive kernel (ARK-SVM), described in Section 3.3, are also provided for reference point estimated using arithmetic mean and geometric mean.

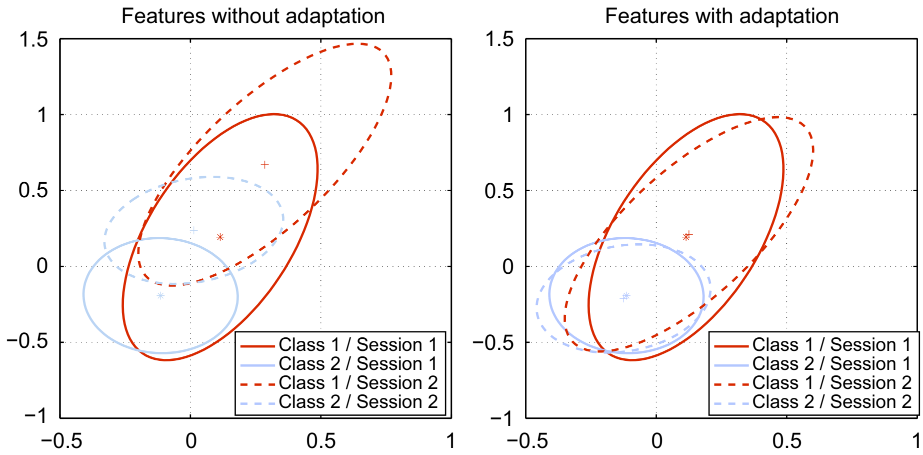

Fig. 2. A relevant choice of $\mathbf{C}_{\text{ref}}$ leads to a quasi invariant feature space across sessions. Left: representations of the feature distribution in the tangent plane for the two most discriminant dimensions, selected with a t-test, for two classes in two different sessions without adaptation of the reference point. The reference point used for the second session is estimated on the data of the first session (standard way). Right: the reference point of the second session is adapted using the data of the second session. Data from subject 1 from the BCI Competition IV, dataset IIa (see Section 4.1 for dataset description).

Table 1
Average classification accuracy (and standard deviation) across the nine subjects for six pairs of mental tasks.

Pair of mental tasks	Adaptive RK-SVM		RK-SVM			SVM vec	CSP+LDA
	$\mathbf{C}_{\text{ref}}=\mathfrak{G}$	$\mathbf{C}_{\text{ref}}=\mathfrak{A}$	$\mathbf{C}_{\text{ref}}=\mathfrak{G}$	$\mathbf{C}_{\text{ref}}=\mathfrak{A}$	$\mathbf{C}_{\text{ref}}=\mathbf{I}_E$		
LH/RH	**82.7** (**14.3**)	81.6 (14.8)	79.9 (13.4)	80.6 (13.2)	80.6 (12.8)	73 (14.8)	75.9 (19.2)
LH/BF	**89.5** (**10.7**)	88.3 (10.2)	87.3 (11.7)	85.8 (14)	85 (12.9)	78.2 (11.2)	80.8 (16.1)
LH/TO	**88.7** (**11.8**)	86.7 (11.9)	86.9 (11.9)	85.6 (12)	85 (12)	81 (13)	82.9 (16.8)
RH/BF	**87.3** (**11**)	87.1 (10.2)	85.9 (10.4)	83.6 (12.6)	80.5 (14.4)	77 (11.3)	84.2 (11.8)
RH/TO	**88.3** (**12.1**)	86.9 (11.6)	86 (12.1)	83.5 (12.3)	83.6 (12.6)	77.5 (14.1)	81.9 (15.6)
BF/TO	**79.5** (**9.6**)	77.6 (8.7)	77.2 (7.9)	74.2 (6.5)	75.5 (9.1)	68.5 (9.1)	73.4 (8.1)
Mean	**86** (**4**)	84.8 (4.2)	83.9 (4.2)	82.2 (4.4)	81.7 (3.6)	75.9 (4.4)	79.9 (4.3)

4.1. Datasets

The dataset IIa of BCI competition IV is used for analysis. Twenty-two electrodes are used (Fz, FC3, FC1, FCz, FC2, FC4, C5, C3, C1, Cz, C2, C4, C6, CP3, CP1, CPz, CP2, CP4, P1, Pz, P2, POz). The reference electrode is located on the left mastoid. A 8–35 Hz band-pass filter, using a fifth-order Butterworth filter, has been applied on the original EEG signals for all subjects and all methods. This dataset is composed of nine subjects who performed 576 trials of right-hand (RH), left-hand (LH), tongue (TO) and both feet (BF) motor imagery (i.e. 144 trials per class) in two different sessions, of 288 trials each, recorded on different days. Since CSP is designed for binary classification, we have evaluated the average performance per subject, for all six possible pairs of mental tasks: {LH/RH, LH/BF, LH/TO, RH/BF, RH/TO, BF/TO}.

The parameters of each algorithm are first trained on the data of the first session and the second session is used to evaluate the generalization performance of the algorithms and the significance of the classification accuracy.

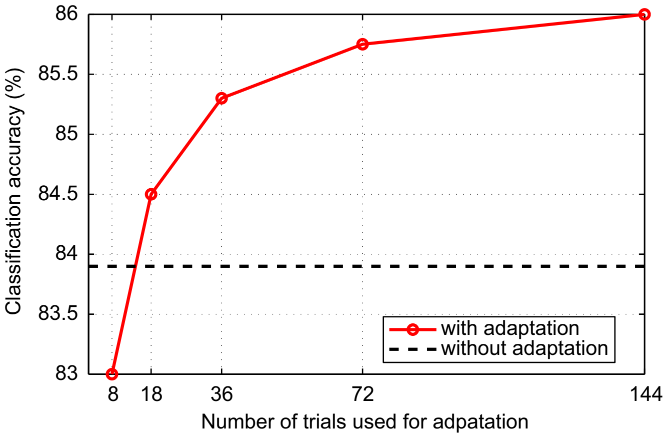

Fig. 3. Evolution of mean classification accuracy when an increasing number of trials is used to estimate the geometric mean of the test session. From 15 trials (10% of the test set), the performance becomes better than the performance obtained without adaptation.

4.2. Results

The hyperparameter λ in SVM is heuristically chosen equal to 10 in our experiments. Results are given in Table 1. As expected, the simple half-vectorization of the covariance matrices (SVM vec) leads to the worst performances. By ignoring the particular structure of the covariances matrices, the linear SVM is not able to outperform the CSP method. Kernel-based SVM approaches outperform the CSP method, whatever the matrix used as reference point. A mean classification accuracy of 86% can be obtained with the adaptive Riemannian kernel whereas the CSP method is limited to 80%.

To verify that gain in the performance was achieved due to the Riemannian framework, and not because of the use of a more advanced classifier, we also test CSP in combination with a SVM. The performance for CSP+SVM is 80.4% instead of 79.9% for CSP+LDA. Indeed, improvement brought by the Riemannian geometry is due to a better feature extraction by taking into account the non-linear information contained in the covariance matrices, which is usually discarded by the linear spatial filtering methods.

Locating the tangent plane at the geometric mean of all covariance matrices yields better results on average compared to the arithmetic mean or the log-Euclidean choice of Eq. (10).

Finally, the adaptation of the reference point between sessions provides the best results in all situations. However, the adaptation of the reference point uses all the data of the test session, so this procedure is not usable in online experiments. To overcome this problem, we can use trials from the beginning of the test session to estimate the reference point, or use an iterative estimation of this point. Fig. 3 shows the classification accuracy when different numbers of trials are used to estimate the geometric mean of the test session. With as few as 15 trials (10% of the test set), results are already better than the non-adaptive method RK-SVM.

The comparison between CSP and proposed kernel-based SVM method on individual sessions is illustrated in Fig. 4. It can be appreciated that, except for few localized pairs of mental tasks, RK-SVM (with or without updating the reference point) consistently outperforms CSP results. This remark is especially true for difficult binary classification cases (CSP performance below 80%) where the improvement brought by kernel-based SVM is significant. However, the improvement achieved by updating the reference point versus does not seem to be higher in the easiest cases.

We have also tested the significance of methods with respect to the reference method (CSP). Results are presented in Fig. 5. This figure illustrates the p-values obtained for the six pairs of mental tasks. These values are obtained with a one-tailed permutation test (1000 permutations) for paired sample for testing null hypothesis $\{\mathcal{H}_0 : \mu_1 = \mu_2\}$ and alternative hypothesis $\{\mathcal{H}_1 : \mu_1 > \mu_2\}$, where μ_1 is the mean of our tested method and μ_2 is the mean of the reference method. The SVM kernel with update of the reference point performs significantly better than the CSP ($p < 0.05$) for the six pairs of mental tasks.

The choice of the reference point impacts on the computational cost of the method. The Log-Euclidean kernel ($\mathbf{C}_{ref} = \mathbf{I}_E$) can be resumed as taking the logarithm of each matrices. The operator $\text{logm}(\cdot)$ involves only an eigenvector decomposition. Using an other matrix as reference point is more computationally

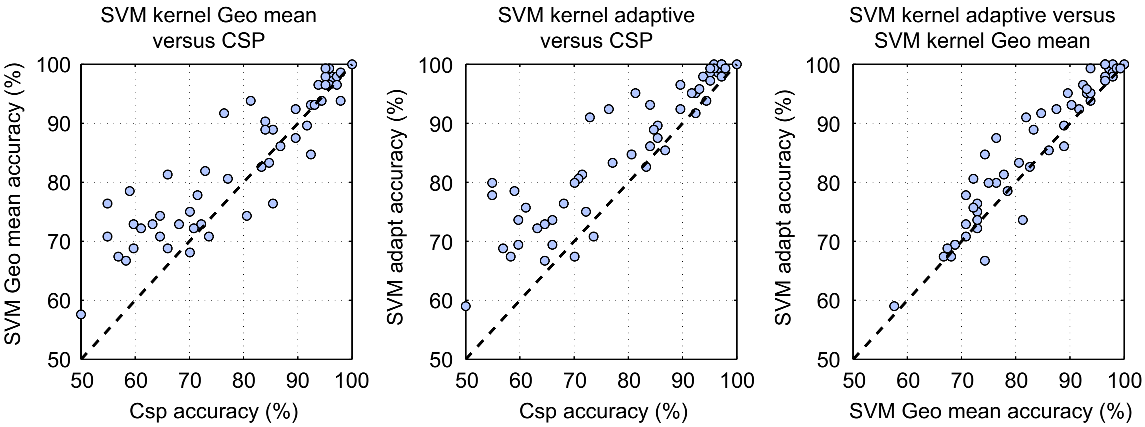

Fig. 4. Comparison of classification accuracy for each individual pair of mental tasks. From the left to the right: RK-SVM versus CSP, ARK-SVM versus CSP classification and RK-SVM with ARK-SVM.

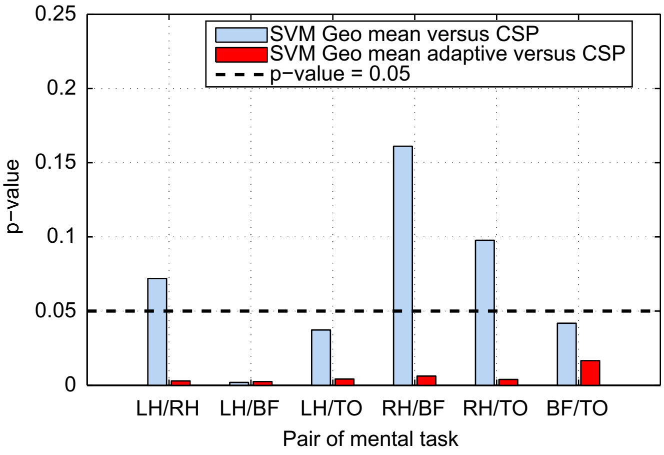

Fig. 5. p-Values for the six pairs of mental tasks in a subject-independent manner (see text for details).

Table 2
Average time in second for each method in training (training the classifier) and test (applying the classifier) stage.

Stage	Adaptive RK-SVM		RK-SVM			SVM vec	CSP+LDA
	$\mathbf{C}_{\text{ref}} = \mathfrak{G}$	$\mathbf{C}_{\text{ref}} = \mathfrak{A}$	$\mathbf{C}_{\text{ref}} = \mathfrak{G}$	$\mathbf{C}_{\text{ref}} = \mathfrak{A}$	$\mathbf{C}_{\text{ref}} = \mathbf{I}_E$		
Training (s)	1.37	0.35	1.37	0.35	0.24	0.1	0.2
Test (s)	1.23	0.25	0.26	0.19	0.08	0.05	0.05
Total (s)	2.6	0.60	1.6	0.54	0.32	0.15	0.25

demanding since it relies on the logarithmic map given by Eq. (4). The estimation of the geometric mean $\mathfrak{G}$ is based on an iterative procedure which involves the logarithmic mapping of all the covariances matrices in the training set at each iteration. Thus, this estimation has a high computational cost which is dependent both on the size of the matrices and the size of the training set. The use of arithmetic mean $\mathfrak{A}$ could be a good alternative to this estimation. The adaptive kernel implies again the estimation of a new geometric/arithmetic mean and this also increases the computational cost of the method. Finally, the choice of the reference point could be a trade-off between the expected performance and the available computational power, especially for online experiments.

Table 2 gives the average computation time in second for the different methods on a common computer.[2] The timing for training and test stage is given separately. As it can be seen, the AKR-SVM is 10 times slower than the CSP method. However, with only 2.6 s required to train the classifier and apply it on the 144 test trials, this difference is not critical.

5. Conclusion

This paper proposes a new kernel for classifying covariance matrices directly. The approach, based on the Riemannian geometry, is tested on a BCI competition dataset and outperforms significantly the conventional CSP method. The proposed kernel could be employed in different applications where covariance matrices are the main ingredients of the feature extraction process. More generally, this kernel could be employed in every paradigm where the underlying signals are characterized in terms of bandpower or where the classes engender a specific spatial covariance structure.

This work proves that the spatial filtering of electrodes can be by-passed without loss of performance using Riemannian geometry concepts. Thus, the feature extraction process is only based on unsupervised operations (temporal filtering and covariance estimation), which make the calibration phase easier and more robust to overfitting.

In addition, a simple way to handle inter-sessions variability is presented, consisting in the re-estimation of the reference point used for the tangent space mapping, and shows its effectiveness on the used dataset. Future work will investigate the online use of this algorithm and the use of the adaptive kernel as a possible way to deal with inter-subjects variability.

References

[1] A. Barachant, S. Bonnet, M. Congedo, C. Jutten, Common spatial pattern revisited by Riemannian geometry, in: IEEE International Workshop on Multimedia Signal Processing (MMSP) 2010, 2010, pp. 472–476.

[2] A. Barachant, S. Bonnet, M. Congedo, C. Jutten, Riemannian geometry applied to BCI classification, in: Ninth International Conference Latent Variable Analysis and Signal Separation (LVA/ICA 2010), Lecture Notes in Computer Science, vol. 6365, 2010, pp. 629–636.

[3] A. Barachant, S. Bonnet, M. Congedo, C. Jutten, BCI signal classification using a Riemannian-based kernel, in: Proceedings of the 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, April 2012, pp. 97–102.

[2] Processor intel SU4100 1.3 GHz dual core, memory 8 GB.

[4] A. Barachant, S. Bonnet, M. Congedo, C. Jutten, Multiclass brain–computer interface classification by Riemannian geometry, IEEE Trans. Biomed. Eng. 59 (April (4)) (2012) 920–928.

[5] M. Berger, A Panoramic View of Riemannian Geometry, Springer, 2003.

[6] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, K. Müller, Optimizing spatial filters for robust EEG Single-Trial analysis, IEEE Signal Process. Mag. 25 (2008) 41–56.

[7] S. Canu, Y. Grandvalet, V. Guigue, A. Rakotomamonjy, SVM and kernel methods matlab toolbox, Perception Systmes et Information, INSA de Rouen, Rouen, France, 2005.

[8] J. Farquhar, A linear feature space for simultaneous learning of spatio-spectral filters in BCI, Neural Networks 22 (November (9)) (2009) 1278–1285.

[9] M. Harandi, C. Sanderson, A. Wiliem, B.C. Lovell, Kernel analysis over Riemannian manifolds for visual recognition of actions, pedestrians and textures, in: IEEE Workshop on the Applications of Computer Vision (WACV'12), Colorado, 2012.

[10] M. Krauledat, M. Tangermann, B. Blankertz, K.R. Müller, Towards zero training for brain–computer interfacing, PLoS ONE 3 (8) (2008).

[11] O. Ledoit, M. Wolf, A well-conditioned estimator for large-dimensional covariance matrices, J. Multivar. Anal. 88 (2) (2004) 365–411.

[12] F. Lotte, M. Congedo, A. Lcuyer, F. Lamarche, B. Arnaldi, A review of classification algorithms for EEG-based brain–computer interfaces, J. Neural Eng. 4 (2007).

[13] M. Moakher, A differential geometric approach to the geometric mean of symmetric positive-definite matrices, SIAM J. Matrix Anal. Appl. 26 (3) (2005) 735–747.

[14] G. Pfurtscheller, F.H. Lopes da Silva, Event-related EEG/MEG synchronization and desynchronization: basic principles, Clin. Neurophysiol. 110 (November (11)) (1999) 1842–1857.

[15] H. Ramoser, J. Muller-Gerking, G. Pfurtscheller, Optimal spatial filtering of single trial EEG during imagined hand movement, IEEE Trans. Rehabil. Eng. 8 (December (4)) (2000) 441–446.

[16] B. Reuderink, J. Farquhar, M. Poel, A. Nijholt, A subject-independent brain–computer interface based on smoothed, second-order baselining, in: 2011 Annual International Conference of the Engineering in Medicine and Biology Society, EMBC, September 2011, pp. 4600–4604.

[17] B. Schölkopf, A. Smola, Learning with kernels: support vector machines, regularization, optimization, and beyond, December 2001.

[18] P. Shenoy, M. Krauledat, B. Blankertz, R. Rao, K. Müller, Towards adaptive classification for BCI, J. Neural Eng. 3 (March (1)) (2006) 13–23.

[19] O. Tuzel, F. Porikli, P. Meer, Pedestrian detection via classification on Riemannian manifolds, IEEE Trans. Pattern Anal. Mach. Intell. 30 (10) (2008) 1713–1727.

[20] E. Wang, W. Guo, L. Dai, K. Lee, B. Ma, H. Li, Factor analysis based spatial correlation modeling for speaker verification, IEEE ISCSLP-2010, November 2010, pp. 166–170.

Alexandre Barachant was born in Chateauroux, France, in 1985. In 2012, he received the Ph.D. degree in signal processing from the Grenoble University, Grenoble, France. He has worked at CEA-LETI during his Ph.D. on the topic of brain–computer interfaces. He his actually a post-doc fellow at the Centre National de la Recherche Scientifique (CNRS) in the GIPSA Laboratory, Grenoble, France. His current research interests include statistical signal processing, Riemannian geometry and classification of neurophysiological recordings with applications to brain–computer interfaces.

Stéphane Bonnet is a Research Engineer who graduated from the Polytechnique National Institute of Grenoble (INPG), France, in 1996. He received his Ph.D. degree in image processing from the National Institute of Applied Sciences (INSA) of Lyon, France, in 2000. In 2001, he joined the CEA-LETI of Grenoble as a post-doctoral student to work on dynamic computed tomography reconstruction algorithms. He has led several projects focussed on motion sensor technology for Healthcare applications. He is currently involved in R&D in the area of information processing related to neurosciences applications ranging from MEA onboard spike detection to EEG-based brain–computer interfaces.

Marco Congedo was born in Bari, Italy, in 1972. In 2003 he obtained the Ph.D. degree in Biological Psychology with a minor in Statistics from the University of Tennessee, Knoxville. From 2003 to 2006 he has been a post-doc fellow at the French National Institute for Research in Informatics and Control (INRIA) and at France Telecom R&D, in France. Since 2007 Congedo is a Research Scientist at the Centre National de la Recherche Scientifique (CNRS) in the GIPSA Laboratory, Grenoble, France. Congedo has been the recipient of several awards, scholarships and research grants. He is interested in basic human electroencephalography (EEG) and magnetoencephalography (MEG), real-time neuroimaging (neurofeedback and brain–computer interface), EEG hyperscanning and multivariate statistical tools useful for EEG and MEG such as inverse solutions and blind source separation. Congedo is a Fellow of the International Society for Neurofeedback and Research and a Consulting Editor for the Journal of Neurotherapy.

Christian Jutten (M'03-SM'06-F'08) received the Ph.D. degree in 1981 and the Docteur ès Sciences degree in 1987 from the Institut National Polytechnique of Grenoble (France). He taught as an Associate Professor in the Electrical Engineering Department from 1982 to 1989. He was a visiting professor in Swiss Federal Polytechnic Institute in Lausanne in 1989, before became a full-time professor in University Joseph Fourier of Grenoble, more precisely in the sciences and technologies department: Polytech' Grenoble. He is currently the associate director of the Grenoble images, speech, signal and control laboratory (GIPSA, 300 people) and head of the Department Images-Signal (DIS) of this laboratory. For 25 years, his research interests are blind source separation, independent component analysis and learning in neural networks, including theoretical aspects (separability, source separation in nonlinear mixtures), applications in signal processing (biomedical, seismic, speech) and data analysis. He is author or co-author of more than 45 papers in international journals, 3 books, 16 invited papers and 140 communications in international conferences. He has been an associate editor of IEEE Trans. on Circuits and Systems (1994–1995), and a co-organizer for the first International Conference on Blind Signal Separation and Independent Component Analysis (Aussois, France, January 1999). He has been a scientific advisor for signal and images processing at the French Ministry of Research from 1996 to 1998 and for the French National Research Center from 2003 to 2006. He has been an associate editor of IEEE Trans. CAS from 1992 to 1994. He is a member of the technical committee "Blind signal Processing" of the IEEE CAS society and of the technical committee "Machine Learning for signal Processing" of the IEEE SP society. He is a reviewer of main international journals (IEEE Trans. on Signal Processing, IEEE Signal Processing Letters, IEEE Trans. on Neural Networks, Signal Processing, Neural Computation, Neurocomputing, etc.) and conferences in signal processing and neural networks (ICASSP, ISCASS, EUSIPCO, IJCNN, ICA, ESANN, IWANN, etc.). He received the EURASIP best paper award in 1992 and Medal Blondel in 1997 from SEE (French Electrical Engineering society) for his contributions in source separation and independent component analysis, and has been elevated as a Fellow IEEE and a senior Member of Institut Universitaire de France in 2008.