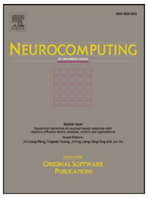
Neurocomputing 275 (2018) 1871–1883

Contents lists available at ScienceDirect

Neurocomputing

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



EEG feature descriptors and discriminant analysis under Riemannian Manifold perspective

Chuong H. Nguyen*, Panagiotis Artemiadis

School for Engineering of Matter, Transport and Energy, Arizona State University, Tempe, AZ 85287 USA

ARTICLE INFO

Article history:
Received 10 January 2017
Revised 21 September 2017
Accepted 21 October 2017
Available online 31 October 2017

Communicated by Nianyin Zeng

Keywords:
Brain Computer Interface (BCI)
Discriminant analysis
Electroencephalography (EEG)
Multi-class
Multi-kernel
Riemannian Manifold
Relevance Vector Machine

ABSTRACT

This paper presents a framework to classify motor imagery in the context of multi-class Brain Computer Interface based on electroencephalography (EEG). Covariance matrices are extracted as the EEG signal descriptors, and different dissimilarity metrics on the manifold of Symmetric Positive Definite (SPD) matrices are investigated to classify these covariance descriptors. Specifically, we compare the performance of the Log Euclidean distance, Stein divergence, Kullback–Leibler divergence and Von Neumann divergence. Furthermore, inspired from the conventional Common Spatial Pattern, discriminant analysis performed directly on the SPD manifold using different mentioned metrics are proposed to improve the classification accuracy. We also propose a new feature, namely Heterogeneous Orders Relevance Composition (HORC), by combining different relevance matrices, such as Covariance, Mutual Information or Kernel Matrix under the Tensor Framework and Multiple Kernel fusion. Multi-Class Multi-Kernel Relevance Vector Machine is adopted to provide a sparse classifier and Bayesian confidence prediction. Finally, we compare the performance of total 16 methods on the dataset IIa of the BCI Competition IV. The results shows that the mentioned dissimilarity metrics perform quite equally on the original manifold, whereas the proposed discrimination methods can improve the accuracy by 3–5% on the reduced dimension manifold.

1. Introduction

Brain Computer Interface (BCI) applications allow human to communicate and control computer aided systems using electrical activity recorded from the brain. A typical process of BCI system is to capture and analyze the electrical brain signals, extract their distinguished features and classify the mental tasks.

Scalp Electroencephalography (EEG) is commonly used as a noninvasive method to capture the brain's electrical activity. Over the past decade, a variety of EEG features has been proposed for many specific BCI applications. Several important EEG features include amplitude values of EEG signals, band powers, power spectral density, autoregressive and adaptive autoregressive parameters, mixed time-frequency representations, time-frequency synthesized spatial patterns, spatial deconvolution, inverse model-based features, and extreme energy ratio, where details of the aforementioned features can be found from the publications reviewed in [1–3].

However, these aforementioned methods are sub-optimal ways to extract the features. First, the energy of the scalp EEG signals is simultaneously distributed in 3 domains: Time–Space–Frequency. Hence, their original feature space is a 3-dimensional tensor. Unfortunately, the classical approaches in one way or another extract the feature descriptor into a vector in Euclidean space. This is rooted from the fact that these descriptors rely on some statistic parameters, such as mean, variance, median, which are defined as scalars. Furthermore, pattern recognition and dissimilarity metrics between features are built only for vectors in Euclidean space. Thus, the classical approaches fail to notice a very distinctive characteristic of data: their structure, or more specific, the manifolds and the interrelation across the tensor dimensions. Recently, data treatment based on the concept of manifold and tensor analysis have been proved to be more effective and adopted in many applications. Geometric control [4] has been extensively studied to model and control mechanical system dynamics under the concept of Riemannian Manifolds. In computer vision, covariance matrix [5] is considered as a specific class of Riemannian Manifolds and currently is the state-of-the art descriptor used for object and action recognition in video. In [6], Barachant et al. obtained very promising results by using covariance matrix as the EEG descriptor and adopting the Log Euclidean distance to discriminate among

* Corresponding author.
E-mail addresses: chuong.h.nguyen@asu.edu (C.H. Nguyen), panagiotis.artemiadis@asu.edu (P. Artemiadis).

https://doi.org/10.1016/j.neucom.2017.10.013
0925-2312/© 2017 Elsevier B.V. All rights reserved.

them. In [7], Phan and Cichocki proposed a Fisher Discrimination Analysis for higher-order dimension tensor, and also achieved encouraging results. In this paper, we revise the classical EEG features under the perspective of Riemannian Manifold and tensor analysis, and conduct an empirical study to compare the performance of different approaches.

Furthermore, classical approaches often rely only a single feature, i.e. power spectrum, to characterize the mental tasks. This is acceptable for some simple binary motor imagery tasks, i.e. right–left hand control, mostly studied in current BCI research. However, in general brain activities are very complicated and hardly be represented just by a single feature. In contrast, an appropriate combination of different features may provide different perspectives to the signals and hence be more distinctive. The common approach is to concatenate all the feature vectors into one long vector and utilize dimension reduction techniques, such as Principle Component Analysis (PCA), Linear Discriminant Analysis (LDA) or Canonical Correlation Analysis (CCA) to remove noisy and redundant features. However, these approaches often require the features to be in the same range and are not suitable for heterogeneous features, e.g. combination of features in the vector form with binary or histogram form. Furthermore, the dimension reduction techniques are designed independently from the classifiers, hence they do not take into account the bias of trained classifiers [8]. In this paper, we utilize a multi-kernel learning method, namely multi-class multi-kernel Relevance Vector Machine [9], to promote a framework for more efficient fusion of EEG features.

This paper is organized as follows. Section 2 introduces notation and basic concepts used in the paper. A review of EEG features is presented in Section 3. Section 4 revises the classical features under the perspective of Riemannian Manifold, and extends the concepts of discriminant analysis from Euclidean to Riemannian using different kinds of manifold distance. Heterogeneous Orders Relevance Composition (HORC) is introduced in Section 5. In Section 6, the multi-class multi-kernel machine learning approach proposed in [9] is adopted for feature fusion and recognition. Section 7 presents the experimental results and discussion. Section 8 concludes the paper and discusses future work.

2. Preliminary

This section establishes notations and definitions used in the paper. We denote m, $\boldsymbol{m}$, $\boldsymbol{M}$ and $\underline{\boldsymbol{M}}$ as a scalar, vector, matrix and tensor form, respectively. Let $\mathbb{R}^n$ ($\mathbb{C}^n$) be an n dimension real (complex) space, $\mathbf{1}_n \in \mathbb{R}^n$ be a vector with all entries equal to 1, and $\boldsymbol{I}_n \in \mathbb{R}^{n\times n}$ be the identity matrix. $E\{\boldsymbol{x}\}$ is the expected value of $\boldsymbol{x}$ and diag($\boldsymbol{x}$) is a diagonal matrix constructed from $\boldsymbol{x}$.

$\boldsymbol{A}^{\mathrm{T}}$ denotes the (conjugate) transpose of $\boldsymbol{A}$, and vec($\boldsymbol{A}$) is the vectorizing operator on matrix $\boldsymbol{A}$. If $\boldsymbol{A}$ is symmetric then vec($\boldsymbol{A}$) only takes an upper half of the matrix. We denote $\|\boldsymbol{v}\|_p$ and $\|\boldsymbol{v}\|$ as the $\mathcal{L}_p$ norm and $\mathcal{L}_2$ norm of a vector $\boldsymbol{v}$, respectively. $\|A\|_F$ denotes the Frobenius norm of matrix $\boldsymbol{A}$.

Definition 2.1. An $n \times n$ matrix $\boldsymbol{A}$ is Symmetric Positive Definite (SPD) if $\boldsymbol{A} = \boldsymbol{A}^{\mathrm{T}}, \boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} > 0, \forall \boldsymbol{x} \neq 0$. Equivalently, the eigenvalues of $\boldsymbol{A}$, denoted as $\lambda(\boldsymbol{A})$, are positive.

Definition 2.2. An $n \times d$ matrix $\boldsymbol{A}$ is *orthogonal* if its columns are orthogonal unit vector, i.e. $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} = \boldsymbol{I}_d$.

Definition 2.3. $\boldsymbol{A}^k$, $\exp(\boldsymbol{A})$ and $\log(\boldsymbol{A})$ of matrix $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ are defined through its eigenvalues $\boldsymbol{\Lambda}$ and eigenvectors $\boldsymbol{U}$ as

$$\boldsymbol{A}^k \triangleq \boldsymbol{U}\mathrm{diag}\left(\left[\lambda_1^k, \ldots, \lambda_n^k\right]\right)\boldsymbol{U}^{\mathrm{T}} = \boldsymbol{U}\boldsymbol{\Lambda}^k\boldsymbol{U}^{\mathrm{T}},$$

$$\exp(\boldsymbol{A}) \triangleq \boldsymbol{U}\mathrm{diag}\left(\left[e^{\lambda_1}, \ldots, e^{\lambda_n}\right]\right)\boldsymbol{U}^{\mathrm{T}} = \boldsymbol{U}e^{\boldsymbol{\Lambda}}\boldsymbol{U}^{\mathrm{T}},$$

$$\log(\boldsymbol{A}) \triangleq \boldsymbol{U}\mathrm{diag}([\log(\lambda_1), \ldots, \log(\lambda_n)])\boldsymbol{U}^{\mathrm{T}} = \boldsymbol{U}\log(\boldsymbol{\Lambda})\boldsymbol{U}^{\mathrm{T}}.$$

Definition 2.4. $\boldsymbol{x}|\mu, \alpha \sim \mathcal{N}(\boldsymbol{x}|\mu, \alpha^{-1})$ denotes that the random variable $\boldsymbol{x}$ follows a Gaussian distribution with the mean μ and variance $\sigma^2 = \alpha^{-1}$, i.e., its probability $P(\boldsymbol{x}|\mu, \alpha) = \mathcal{N}(\boldsymbol{x}|\mu, \alpha^{-1})$.

2.1. Multiple kernel learning for heterogeneous feature fusion [10]

Let $\{\boldsymbol{x}_i, l(\boldsymbol{x}_i)\}_{i=1}^n$ be a set of labeled patterns where $\boldsymbol{x}_i \in \mathcal{X}$ is a feature of a sample i and $l(\boldsymbol{x}_i) \in \{\pm 1\}$ is its output label. For a chosen feature map $\phi : \mathcal{X} \to \boldsymbol{R}^m$ assuming that a set $\{\phi(\boldsymbol{x}_i), l(\boldsymbol{x}_i) = -1\}$ can be linearly separated from $\{\phi(\boldsymbol{x}_j), l(\boldsymbol{x}_j) = 1\}$, the training process for classification attempts to find an optimal hyperplane $\boldsymbol{a} \in \boldsymbol{R}^m$ such that

$$y(\boldsymbol{x}) = \boldsymbol{a}^{\mathrm{T}}\phi(\boldsymbol{x}) + w_o \in \mathbb{R}, \qquad \text{s.t:} \qquad y(\boldsymbol{x})l(\boldsymbol{x}) > 0. \tag{1}$$

The solution's principle is to minimize the cost function

$$J(\boldsymbol{a}, w_0) = \frac{1}{2}\sum_{i=1}^n \left(\boldsymbol{a}^{\mathrm{T}}\phi(\boldsymbol{x}_i) + w_o - t_i\right)^2 + \frac{\lambda}{2}h\left(\boldsymbol{a}^2\right),$$

where $h(\boldsymbol{a}^2) > 0$ is a constraint function on $\boldsymbol{a}$, and λ is the Lagrange multiplier. By setting $\frac{\partial J}{\partial \boldsymbol{a}} = 0$, we obtain

$$\boldsymbol{a} = \sum_{i=1}^n \underbrace{-\left(\frac{\partial h}{\partial \boldsymbol{a}^2}\right)^{-1} \frac{\boldsymbol{a}^{\mathrm{T}}\phi(\boldsymbol{x}_i) + w_o - t_i}{\lambda}}_{w_i} \phi(\boldsymbol{x}_i) = \sum_{i=1}^n w_i\phi(\boldsymbol{x}_i). \tag{2}$$

Substituting (2) to (1) yields the *dual form of optimization*

$$y(\boldsymbol{x}) = \sum_{i=1}^n w_i\phi^{\mathrm{T}}(\boldsymbol{x})\phi(\boldsymbol{x}_i) + w_o = \sum_{i=1}^n w_i k(\boldsymbol{x}, \boldsymbol{x}_i) + w_o = \boldsymbol{w}^{\mathrm{T}}\Phi(x), \tag{3}$$

where $k(\boldsymbol{x}, \boldsymbol{x}_i) = \phi^{\mathrm{T}}(\boldsymbol{x})\phi(\boldsymbol{x}_i) \in \mathbb{R}$ is called *kernel at* $\boldsymbol{x}_i$, $\boldsymbol{w} = [w_0, \ldots, w_n]^{\mathrm{T}}$ and $\Phi(x) = [1,\ k(\boldsymbol{x}, \boldsymbol{x}_i), \ldots, k(\boldsymbol{x}, \boldsymbol{x}_n)]^{\mathrm{T}} \in \mathbb{R}^{n+1}$.

Eq. (3) is referred as the "kernel trick" as it embeds the feature from the original space $\mathcal{X}$ to the *Reproducing Kernel Hilbert space*. Hence, if a sample $\boldsymbol{x}$ is represented by a set of m features $\{\boldsymbol{x}^{(j)}\}_{j=1}^m$, where each $\boldsymbol{x}^{(j)}$ lies in its own space $\mathcal{X}_j$ equipped with a map $\phi_j(\boldsymbol{x})$, the weighted feature map $\phi(\boldsymbol{x}) = [\sqrt{\beta_1}\phi_1(\boldsymbol{x}), \ldots, \sqrt{\beta_m}\phi_m(\boldsymbol{x})]^{\mathrm{T}}$ yields the kernel at $\boldsymbol{x}_i$ at

$$k(\boldsymbol{x}, \boldsymbol{x}_i) = \sum_{j=1}^m \beta_j k_j(\boldsymbol{x}, \boldsymbol{x}_i), \quad k_j(\boldsymbol{x}, \boldsymbol{x}_i) = \phi_j^{\mathrm{T}}(\boldsymbol{x})\phi_j(\boldsymbol{x}_i). \tag{4}$$

Thus, multi-kernels function in form of linear combination of different kernels provides a clever way to combine heterogeneous features. In practice, the explicit map $\phi(\boldsymbol{x})$ are mostly avoided by directly defining the kernel. For example, the Gaussian kernel is commonly used:

$$k(\boldsymbol{x}_1, \boldsymbol{x}_2) = e^{-\gamma d^2(\boldsymbol{x}_1, \boldsymbol{x}_2)},$$

where $d(\boldsymbol{x}_1,\ \boldsymbol{x}_2)$ is the distance between two points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ defined in its original space $\mathcal{X}$, i.e. $d(\boldsymbol{x}_1, \boldsymbol{x}_2) = \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|$ for $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^q$. Using Gaussian kernel implies that the dimension of the mapped feature space is infinite, $\phi : \mathcal{X} \to \boldsymbol{R}^{+\infty}$ while still limits the dimension of $\boldsymbol{w}$ by the number of training samples, i.e. $\boldsymbol{w} \in \mathbb{R}^{n+1}$. Hence, multi-kernels function efficiently combines heterogeneous features.

3. Literature review

3.1. Temporal–spatial–frequential decomposition

In feature extraction process, we aim to extract from the recored EEG signals the most salient characteristics that are correlated to the observed outcome. Since EEG signal captures brain's electrical activity, its energy is distributed over three domains: time, spatial and frequency. This section summarizes the main techniques to decompose the signals.

3.1.1. Temporal filters

This consists of several preprocessing steps. First, the EEG signal is band-bass filtered into a suitable narrow frequency band. For example, frequency band for motor imagery is often found in 7–30 Hz. After that, artifacts due to EOG such as eye-blinking need to be removed. Finally, since the brain activation can only maintained in a short period of time, the recorded EEG in one trial can be separated into several possibly overlapped segments. Heuristically, a time segment can be from 1–2 s.

3.1.2. Spatial filters

The recorded EEG signals $\boldsymbol{X}(t) = [x_1(t), \dots, x_N(t)]$ from N channels can be thought as "results echoed" from n unknown sources $\boldsymbol{S}(t) = [s_1(t), \dots, s_n(t)]$ and assumed to be a linear combination as $\boldsymbol{X}(t) = \boldsymbol{WS}(t)$, where $\boldsymbol{W} \in \mathbb{R}^{N\times n}$ is the linear fusion matrix [11]. Since $\boldsymbol{S}(t)$ is unknown, additional assumptions must be imposed to cast the constraints on the optimal problem of estimating $\boldsymbol{W}$. Once $\boldsymbol{W}$ is determined from the calibration process, $\boldsymbol{S}(t)$ can be obtained for an incoming $\boldsymbol{X}(t)$. Among several available approaches, Independent Component Analysis (ICA) [11], Canonical Correlation Analysis (CCA) [12] and standardized Low Resolution Electromagnetic Tomography (sLORETA) [13] have been successfully used in BCI applications.

3.1.3. Frequential filters

Among several approaches for time-frequency analysis, such as autoregressive model (ARM), Short-Time Fourier Transform (STFT) and wavelet transform (WT), discrete wavelet transform (DWT) is proved to be more effective to characterize EEG as it can handle non-stationary signals [2,7,14,15]. However, the performance of WT critically depends on the similarity between the shape of input signal and that of the chosen basis function. Thus, using one fixed WT basis function may not optimally capture the dynamics and nonlinearity of the brain signal. Recently, Hilbert Huang Transform (HHT) [16] offers an alternative time-frequency tool, in which the basis function is adaptively constructed by the data itself. Hence, HHT is more attractive to EEG processing than the conventional mentioned frequency transformation methods [17].

3.2. Features in Euclidean space

After being time–space–frequency filtered, the input EEG is transformed to $\boldsymbol{X}(t) \in \mathbb{R}^{N_c\times N_b\times T_s}$, where N_c is the number of channels, N_b is number of frequency bands and T_s is the number of sampling points. From here, distinguished feature can be extracted to recognize the mental tasks.

Classical approaches found in the literature, e.g. in [1,3,14,15,18] and their references, share the same principle: the features are extracted from the original n-dimension space into a 1-dimension feature vector. A typical framework involves extracting statistic measurements along the temporal dimension, and then vectorizing them along the spatial and frequential dimension. Finally, feature selection is utilized to remove any redundant and irrelevant elements. Hence, classification can be performed based on the dissimilarity between the features vectors, which can be measured by several well-established metrics in Euclidean space.

Let $\boldsymbol{a}_{i,k}(t) \in \mathbb{R}^{T_s}$ and $\varphi_{i,k}(t) \in \mathbb{R}^{T_s}$ be the amplitude and phase of the analytic signal $\boldsymbol{x}_{i,k} = a_{i,k}\angle\varphi_{i,k}(t) \in \mathbb{C}^{T_s}$ at the channel i and sub-band frequency k where $i = \{1, \dots, N_c\}, k = \{1, \dots, N_b\}$.

3.2.1. Features commonly used in BCI

Power spectrum (e), mean coefficients (μ), and standard deviation (σ) of individual sub-band $\boldsymbol{x}_{i,\,k}(t)$ is computed as:

$$e_{i,k} = \boldsymbol{a}_{i,k}^{\mathrm{T}}\boldsymbol{a}_{i,k}, \quad \boldsymbol{\mu}_{i,k} = \frac{\mathbf{1}^{\mathrm{T}}\boldsymbol{a}_{i,k}}{T_s}, \quad \sigma_{i,k}^2 = \frac{\left(\boldsymbol{a}_{i,k} - \boldsymbol{\mu}_{i,k}\right)^{\mathrm{T}}\left(\boldsymbol{a}_{i,k} - \boldsymbol{\mu}_{i,k}\right)}{T_s - 1}.$$

The final feature vector is a concatenation of all individual components, e.g. $\boldsymbol{e} = [e_{1,1} \dots e_{i,k} \dots e_{N_c,N_b}]^{\mathrm{T}}$.

Maximum cross-correlation (R) [19,20] between $\boldsymbol{x} \in \mathbb{C}^{T_s}$ and $\boldsymbol{y} \in \mathbb{C}^{T_s}$ is defined as

$$R_{xy} = \max_{k=1}^{T_s} |\rho(k)|, \quad \rho(k) = \sum_{t=0}^{T_s-k-1} \boldsymbol{x}(t+k)\boldsymbol{y}(t).$$

The correlation matrix $\boldsymbol{R} \in \mathbb{R}^{\bar{n}\times\bar{n}}$ is obtained by computing R_{xy} across all channels i and sub-band k using amplitude and phase i.e. $\boldsymbol{R}_a(j,l) = R_{\boldsymbol{a}_j\boldsymbol{a}_l}$ and $\boldsymbol{R}_\phi(j,l) = R_{\phi_j\phi_l}$.

Coherence (COH) [21,22] is the auto-correlation at a specific frequency bank k across the channels

$$coh(\boldsymbol{x}_{i,k}, \boldsymbol{x}_{j,k}) = \frac{\boldsymbol{x}_{i,k}^{\mathrm{T}}\boldsymbol{x}_{j,k}}{T_s - 1} = \frac{\sum_{t=0}^{T_s} \boldsymbol{a}_{i,k}(t)\boldsymbol{a}_{j,k}(t)e^{i(\varphi_{i,k}(t)-\varphi_{j,k}(t))}}{T_s - 1}.$$

The Coherence matrix $COH \in \mathbb{R}^{N_c\times N_c}$ is typically defined as the maximal coherence magnitude among all frequency bands

$$COH(i,j) = \max_{k=1}^{N_B} \left|coh\left(\boldsymbol{x}_{i,k}, \boldsymbol{x}_{j,k}\right)\right|.$$

Covariance (COV) [6] is a special case of Coherence matrix, where only the amplitude is considered

$$COV = \frac{\boldsymbol{AA}^{\mathrm{T}}}{T_s - 1} \in \mathbb{R}^{\bar{n}\times\bar{n}}.$$

Phase Locking Value (PLV) [23,24] is a special case of coherence when only the phase information is considered and the amplitude is set to 1

$$plv\left(\boldsymbol{x}_{i,k}, \boldsymbol{x}_{j,k}\right) = \frac{1}{T_s}\left|\sum_{t=0}^{T_s} e^{i\left(\varphi_{i,k}(t)-\varphi_{j,k}(t)\right)}\right|.$$

It can be seen that all the aforementioned features are the variants of cross-correlation matrix, which essentially captures the linear relationships between the channels. The coherence matrix *COH* and phase locking value matrix *PLV* are commonly used to construct graphs of brain-functional connectivity [22]. Among them, COH and COV are SPD. However, the final feature vector in classical approach is often obtained by vectorizing the upper half of the matrices. Mapping the feature into the Euclidean space yields sub-optimal results since it ignores this unique structure.

3.3. Feature selection and dimension reduction

There are several techniques to reduce the features' dimension, either through a linear mapping, such as Principle Component Analysis (PCA), Linear Discrimination Analysis (LDA), Local Preserving Projection (LPP) and Local Fisher Discrimination Analysis (LFDA) [25,26], or by performance ranking, such as Fisher Score [15] or Mutual Information [27], or a combination of them.

Among the linear transform techniques, *Common Spatial Pattern* (CSP) [28,29] has been successfully used in BCI contest. CSP seeks for a linear transform $\boldsymbol{W} \in \mathbb{R}^{N_C\times m}$ composed of m spatial filters $w_j \in \mathbb{R}^{N_c}$ that maps the original data $\boldsymbol{X} \in \mathbb{R}^{N_c\times T_s}$ to another space $\boldsymbol{Y} = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}$ in which the following Rayleigh quotient is extremized

$$J(\boldsymbol{w}_j) = \frac{E\left\{\boldsymbol{Y}_j^1\boldsymbol{Y}_j^{1^{\mathrm{T}}}\right\}}{E\left\{\boldsymbol{Y}_j^2\boldsymbol{Y}_j^{2^{\mathrm{T}}}\right\}} = \frac{\boldsymbol{w}_j^{\mathrm{T}}E\left\{\boldsymbol{X}_1\boldsymbol{X}_1^{\mathrm{T}}\right\}\boldsymbol{w}_j}{\boldsymbol{w}_j^{\mathrm{T}}E\left\{\boldsymbol{X}_2\boldsymbol{X}_2^{\mathrm{T}}\right\}\boldsymbol{w}_j} = \frac{\boldsymbol{w}_j^{\mathrm{T}}\boldsymbol{C}_1\boldsymbol{w}_j}{\boldsymbol{w}_j^{\mathrm{T}}\boldsymbol{C}_2\boldsymbol{w}_j}, \tag{5}$$

where $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are samples belong to classes 1 and 2, and $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ are the average of the covariance of each class. The optimal solution for $\boldsymbol{w}_j$ are then the eigenvectors of $\boldsymbol{C}_2^{-1}\boldsymbol{C}_1$ which correspond to the largest and smallest eigenvalues of $\boldsymbol{C}_2^{-1}\boldsymbol{C}_1$. After CSP transformation, the variance of the data is used as feature descriptor.

Among the performance ranking techniques, *Mutual Information (MI)* is the most general and statistically confident index. MI between two random variables X and Y, where X is often denoted the features and Y is the corresponding labels, measures their mutual dependence and is defined by Shannon's formula as

$$I(X,Y) = \sum_{x\in X}\sum_{y\in Y} p(xy)\log\frac{p(xy)}{p(x)p(y)} = H(X) - H(X|Y), \qquad (6)$$

where $H(x)$ and $H(X|Y)$ are the Shannon Entropy and Conditional Entropy defined, respectively, as

$$H(X) = -\sum_{x\in X} p(x)\log p(x),$$

$$H(X|Y) = -\sum_{y\in Y} p(y)\log p(x)\sum_{x\in X} p(x|y)\log p(x|y),$$

and $p(x)$, $p(y)$, $p(xy)$ and $p(x|y)$ are the probability density function (PDF) of X, Y and their joint PDF and conditional PDF respectively. Hence, a feature X can be selected based on its MI with the class label Y. Despite that MI is powerful, simple and intuitive, estimating MI is not easy since the PDFs are unknown. In practice, these PDFs are often estimated either by assumption of foreknown distribution, such as Gaussian, or by constructing their discrete histograms. In [30], the authors combined MI and Joint Approximate Diagonalization to extend CSP to a Multiclass-CSP application.

4. Feature in Riemannian Manifold and tensor

One problem of classical features is that they might discard discriminative information hidden in the original high dimension space. Recently, developments in manifold geometry and tensor analysis promote a new trend in feature descriptors. That is, the dissimilarity between features can be measured directly in the original high dimension space, which allows unveiling hidden features overlooked by the classical approaches. In this section, we summarize some basic definitions of manifolds and tensors, more details can be found in [31]. Then, some successful methods on using manifold and tensor features are analyzed.

4.1. Features in natural manifold

Definition 4.1. A *topological space* (X, N) is a set of points $X = \{x\}$ equipped with a *neighborhood function* N, which assigns each point x to a subset $N(x) \subset X, N(x) \neq \varnothing$.

Definition 4.2. A function f: $X \to Y$ between two topological spaces (X, N_X) and (Y, N_Y) is called *homeomorphism* if f is bijection, continuous and its inversion function f^{-1} is also continuous. Then, X and Y are called *homeomorphic*.

Definition 4.3. A *n-dimensional manifold* (X, N) is a topological space if each point $x \in X$ has a neighborhood $N(x)$ homeomorphic to Euclidean space $\mathbb{R}^n$ by a function $f : N(x) \to \mathbb{R}^n$. A *differentiable manifold* is a manifold equipped with a globally differentiable function f.

Definition 4.4. At each point x of the differentiable manifold X, one can attach a *Tangent Space* $T_X(x)$ that consists of real *tangent vectors* of all possible curves passing through x.

Definition 4.5. A *Riemannian Manifold* is a differentiable manifold equipped with a *smoothly varying inner product* on each Tangent Space.

Definition 4.6. A *geodesic distance* between two points on the manifold is the length of the shortest curve (called *geodesic*) connecting the two points.

Symmetric Positive Definite (SPD) Matrix belongs to a special Riemannian Manifold, often denoted by Sym_n^+. Several dissimilarity metrics are proposed to estimate the distance in Sym_n^+ as follow.

- *Riemannian distance* d_{af} *[32]* defines the geodesic distance between two SPD $\boldsymbol{S}_i$ and $\boldsymbol{S}_j$ as

  $$d_{\mathrm{R}}(\boldsymbol{S}_1,\boldsymbol{S}_2) \triangleq \|\log(\boldsymbol{S}_i^{-1}\boldsymbol{S}_2)\| = \sqrt{\sum_{i=1}^{n}\log^2\lambda_i},$$

  where $\lambda_i = \mathrm{eig}_i(\boldsymbol{S}_1^{-1}\boldsymbol{S}_2)$. This metric is invariant to an affine transformation and inversion. However, solving the generalized eigenvector is very computationally expensive for practical application.
- *Tangent Space distance.* The Riemannian metric can be approximated by the distance between tangent vectors through a common reference point $\boldsymbol{C}$. The tangent vector $\bar{\boldsymbol{S}}_i$ of a point $\boldsymbol{S}_i$ at the reference point $\boldsymbol{C}$ is defined as.

  $$\bar{\boldsymbol{S}}_i = \log_{\boldsymbol{C}}\boldsymbol{S}_i \triangleq \log\left(\boldsymbol{C}^{-\frac{1}{2}}\boldsymbol{S}_i\boldsymbol{C}^{-\frac{1}{2}}\right).$$

  This $\log_{\boldsymbol{C}}$ mapping defines a Lie group equipped by the multiplication and inverse operators [33] as

  $$\boldsymbol{S}_1 \odot \boldsymbol{S}_2 = \exp\left(\log_C\boldsymbol{S}_1 + \log_C\boldsymbol{S}_2\right), \quad \boldsymbol{S}^{-1} = \exp\left(-\log_C\boldsymbol{S}\right).$$

  This Lie group forms a Hilbert inner product between $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ in the Sym_D^+ manifold as

  $$\langle\boldsymbol{S}_1,\boldsymbol{S}_2\rangle_{\boldsymbol{C}} = \mathrm{tr}(\log_{\boldsymbol{C}}(\boldsymbol{S}_1)\log_{\boldsymbol{C}}(\boldsymbol{S}_2)),$$

  and the distance between $\boldsymbol{S}_i$ and $\boldsymbol{S}_j$ are derived as

  $$d_{TS}^2(\boldsymbol{S}_1,\boldsymbol{S}_2)_{\boldsymbol{C}} \triangleq \|\bar{\boldsymbol{S}}_i - \bar{\boldsymbol{S}}_j\|_F^2 = \mathrm{tr}\left(\left(\bar{\boldsymbol{S}}_i - \bar{\boldsymbol{S}}_j\right)\left(\bar{\boldsymbol{S}}_i - \bar{\boldsymbol{S}}_j\right)^{\mathrm{T}}\right),$$

  To obtain a good approximation with Rienmannian geodesics, the reference point $\boldsymbol{C}$ needs to be close to the two points. Hence, $\boldsymbol{C}$ is heuristically selected as the geometric mean of the point set $\{\boldsymbol{S}_i\}$. However, mapping to the Tangent Space flattens the manifold and does not preserve the true geodesic distance. Furthermore, for a set of non stationary points, the mean $\boldsymbol{C}$ shifts over time, hence the mean $\boldsymbol{C}$ needs to be iteratively re-estimated for any new collected data point $\boldsymbol{S}_t$ [6,34].
- *Log-Euclidean distance [35]* selects the reference point $\boldsymbol{C}$ at the identity matrix $\boldsymbol{I}$, hence the distance is simplified as

  $$d_{LE}^2(\boldsymbol{S}_1,\boldsymbol{S}_2) \triangleq \|\log(\boldsymbol{S}_1) - \log(\boldsymbol{S}_2)\|_F^2.$$

  In fact, if the dataset is first whitened by a map $\hat{\boldsymbol{S}}_i = \boldsymbol{C}^{-\frac{1}{2}}\boldsymbol{S}_i\boldsymbol{C}^{-\frac{1}{2}}$ where $\boldsymbol{C}$ can be the geodesic mean, the Tangent Space distance is reduced to Log Euclidean distance. Intuitively, Log-Euclidean distance first maps the SPDs from the Riemannian Manifold to the Euclidean space by the log operator, then compute the Euclidean distance.
- *Kullback–Leibler (KL) divergence [36]* is not a geodesics but instead based on informative geometry. If two random multivariate samples $\boldsymbol{X}_1 \in \mathbb{R}^{d\times T}$ and $\boldsymbol{X}_2 \in \mathbb{R}^{d\times T}$ are assumed to be Gaussian distribution, i.e.

  $$\boldsymbol{X}_1 \sim \mathcal{N}_1(\mu_1,\boldsymbol{S}_1), \quad \boldsymbol{X}_2 \sim \mathcal{N}_2(\mu_2,\boldsymbol{S}_2),$$

  where μ_i and $\boldsymbol{S}_i$ are the mean and covariance of $\boldsymbol{X}_i$ respectively, the KL divergence from $\mathcal{N}_1(\mu_1,\boldsymbol{S}_1)$ to $\mathcal{N}_2(\mu_2,\boldsymbol{S}_2)$ is defined as

  $$d_{KL}^2(\mathcal{N}_1|\mathcal{N}_2) \triangleq \frac{1}{2}\left(\mathrm{tr}\left(\boldsymbol{S}_2^{-1}\boldsymbol{S}_1\right) + \Delta_\mu^{\mathrm{T}}\boldsymbol{S}_2^{-1}\Delta_\mu + \ln\left(\frac{\det\boldsymbol{S}_2}{\det\boldsymbol{S}_1}\right) - d\right),$$

  where $\Delta_\mu \triangleq \mu_2 - \mu_1$. Since the KL divergence is asymmetric, its symmetric distance is defined as $d_{KL}^2(\boldsymbol{S}_1,\boldsymbol{S}_2) \triangleq d_{KL}^2(\mathcal{N}_1|\mathcal{N}_2) + d_{KL}^2(\mathcal{N}_2|\mathcal{N}_1)$. For $\mu_2 = \mu_1 = 0$, the KL distance is simplified to

  $$d_{KL}^2(\boldsymbol{S}_1,\boldsymbol{S}_2) \triangleq \frac{1}{2}\mathrm{tr}\left(\boldsymbol{S}_1^{-1}\boldsymbol{S}_2 + \boldsymbol{S}_2^{-1}\boldsymbol{S}_1\right) - d. \qquad (7)$$

- *Stein divergence [37] or LogDet Divergence [36]* defines the distance between two SPD $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ as

  $$d^2_{SD}(\boldsymbol{S}_1, \boldsymbol{S}_2) \triangleq \log\det\left(\frac{\boldsymbol{S}_1 + \boldsymbol{S}_2}{2}\right) - \frac{\log\det(\boldsymbol{S}_1\boldsymbol{S}_2)}{2}.$$

  Similar to Kullback–Leibler (KL) divergence, the distance d_{SD} is not geodesic.
- *Von Neumann divergence [36]* defines quantum relative entropy between two SPD covariance matrices as

  $$d^2_{VN}(\boldsymbol{S}_1|\boldsymbol{S}_2) \triangleq \mathrm{tr}(\boldsymbol{S}_1(\log(\boldsymbol{S}_1) - \log(\boldsymbol{S}_2)) - \boldsymbol{S}_1 + \boldsymbol{S}_2).$$

  Note that d_{VN} is also asymmetric. Hence, one can define the symmetric distance version as $d^2_{VN}(\boldsymbol{S}_1, \boldsymbol{S}_2) \triangleq d^2_{VN}(\boldsymbol{S}_1|\boldsymbol{S}_2) + d^2_{VN}(\boldsymbol{S}_2|\boldsymbol{S}_1)$

  $$d^2_{VN}(\boldsymbol{S}_1, \boldsymbol{S}_2) = \mathrm{tr}((\boldsymbol{S}_1 - \boldsymbol{S}_2)(\log(\boldsymbol{S}_1) - \log(\boldsymbol{S}_2))).$$

4.2. Manifold discrimination analysis and connection with common spatial pattern

4.2.1. Common spatial pattern revisited under manifold distance

In [38], Samek et al. unveiled that the classical CSP [28] and their variants can be casted into a unified framework based on Kullback–Leibler divergence. Specifically, let $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2 \in \mathbb{R}^{D\times D}$ are the mean covariance matrix of classes 1 and 2, $\boldsymbol{C} = (\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)$, and $\boldsymbol{P} = \boldsymbol{C}^{-\frac{1}{2}}$ is the whitening transform matrix, so that

$$\tilde{\boldsymbol{\Sigma}}_1 + \tilde{\boldsymbol{\Sigma}}_2 = \boldsymbol{I}, \quad \tilde{\boldsymbol{\Sigma}}_1 = \boldsymbol{P}\boldsymbol{\Sigma}_1\boldsymbol{P}^{\mathrm{T}}, \quad \tilde{\boldsymbol{\Sigma}}_2 = \boldsymbol{P}\boldsymbol{\Sigma}_2\boldsymbol{P}^{\mathrm{T}}.$$

The spatial filter $\boldsymbol{V} = \boldsymbol{I}_d\boldsymbol{R} \in \mathbb{R}^{d\times D}, \boldsymbol{R}\boldsymbol{R}^{\mathrm{T}} = \boldsymbol{I}$ is searched to maximize the distance between the means of each class

$$\mathcal{L}(\boldsymbol{V}) \triangleq (1-\lambda)d^2_{KL}\left(\boldsymbol{V}^{\mathrm{T}}\tilde{\boldsymbol{\Sigma}}_1\boldsymbol{V}, \boldsymbol{V}^{\mathrm{T}}\tilde{\boldsymbol{\Sigma}}_2\boldsymbol{V}\right) - \lambda\boldsymbol{\Delta}, \tag{8}$$

where d_{KL} is the Kullback–Leibler (KL) divergence given in (7), and Δ is the regularization defined as

$$\boldsymbol{\Delta} \triangleq \frac{1}{N_1 + N_2}\sum_{c=1}^{2}\sum_{i=1}^{N_c} d^2_{KL}\left(\boldsymbol{V}^{\mathrm{T}}\tilde{\boldsymbol{\Sigma}}^i_c\boldsymbol{V}, \boldsymbol{V}^{\mathrm{T}}\tilde{\boldsymbol{\Sigma}}_c\boldsymbol{V}\right),$$

where $\tilde{\boldsymbol{\Sigma}}^i_c$ are the covariance matrix of trial i in the class c which has total N_c trials. The optimal $\boldsymbol{V}$ can be solved iteratively by the gradient descent method. In [38], the authors prove that $\mathrm{span}(\boldsymbol{W}) = \mathrm{span}(\boldsymbol{V})$, where $\boldsymbol{W}$ is the conventional CSP coefficients given in (5).

4.2.2. Connection between CSP and LDA in tangent space and Log-Euclidean space

If one replaces the Kullback–Leibler divergence in the cost function (8) by another distance, e.g. Log Euclidean, so that

$$\mathcal{L}(\boldsymbol{V}) \triangleq d^2_{TS}\left(\boldsymbol{V}^{\mathrm{T}}\tilde{\boldsymbol{\Sigma}}_1, \boldsymbol{V}^{\mathrm{T}}\tilde{\boldsymbol{\Sigma}}_2\right) - \frac{\lambda}{1-\lambda}\sum_{c=1}^{2}\sum_{i=1}^{N_c}\frac{d^2_{TS}\left(\boldsymbol{V}^{\mathrm{T}}\tilde{\boldsymbol{\Sigma}}^i_c, \boldsymbol{V}^{\mathrm{T}}\tilde{\boldsymbol{\Sigma}}_c\right)}{N_1 + N_2}.$$

where the first term is to maximize the distance between the classes' means, and the second term is to minimize the sum of distances between the samples within each class.

This is the exact meaning of LDA in the Euclidean space, hence the CSP accomplished with Log Euclidean distance is equivalent to the Tangent Space LDA proposed in [6]. In short, these approaches have identical meaning with the only difference in defining the manifold distances.

4.2.3. Discriminant analysis on Riemannian Manifold

Let $f(\boldsymbol{V}, \boldsymbol{S}_i) : \mathrm{Sym}^+_{\mathrm{D}} \mapsto \mathrm{Sym}^+_{\mathrm{d}}$ be the function that maps $\boldsymbol{S}_i$ from the original manifold D to the lower dimension, more discriminable one, i.e. $d < D$, using the projector $\boldsymbol{V}$. If Stein divergence or Kullback divergence is used, it can be defined as

$$f(\boldsymbol{V}, \boldsymbol{S}_i) \triangleq \boldsymbol{V}^{\mathrm{T}}\boldsymbol{S}_i\boldsymbol{V}.$$

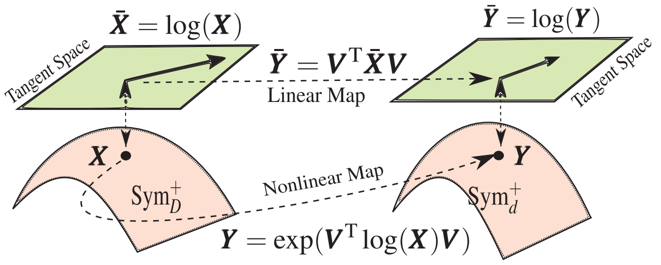

$\bar{\boldsymbol{X}} = \log(\boldsymbol{X})$
$\bar{\boldsymbol{Y}} = \log(\boldsymbol{Y})$
Tangent Space
$\bar{\boldsymbol{Y}} = \boldsymbol{V}^{\mathrm{T}}\bar{\boldsymbol{X}}\boldsymbol{V}$
Linear Map
Tangent Space
$\boldsymbol{X}$
Sym^+_D
Nonlinear Map
$\boldsymbol{Y}$
Sym^+_d
$\boldsymbol{Y} = \exp(\boldsymbol{V}^{\mathrm{T}}\log(\boldsymbol{X})\boldsymbol{V})$

Fig. 1. Dimension reduced mapping from $\mathrm{Sym}^+_D \mapsto \mathrm{Sym}^+_d$ under Log Euclidean operator.

Table 1
Between and within weighted matrices.

Condition	$y_i = y_j = c$	$y_i \neq y_j$
$\boldsymbol{W}^{(w)}_{i,j}$	$a_{i,j}\frac{1}{N_c}$	0
$\boldsymbol{W}^{(b)}_{i,j}$	$b_{i,j}(\frac{1}{N} - \frac{1}{N_c})$	$\frac{1}{N}$

Table 2
Affinity coefficients.

	FDA	kNN	Heat kernel	Local scaling
$a_{i,\,j}$	1	N_c/N	$e^{-\gamma d^2(\boldsymbol{S}_i,\boldsymbol{S}_j)}$	$e^{-\gamma_i\gamma_j d^2(\boldsymbol{S}_i,\boldsymbol{S}_j)}$
$b_{i,\,j}$	1	0	$e^{-\gamma d^2(\boldsymbol{S}_i,\boldsymbol{S}_j)}$	$e^{-\gamma_i\gamma_j d^2(\boldsymbol{S}_i,\boldsymbol{S}_j)}$

In case of the Log Euclidean distance, there are two approaches. First, the mapping can be defined as $f(\boldsymbol{V}, \boldsymbol{S}_i) \triangleq \boldsymbol{V}^{\mathrm{T}}\boldsymbol{s}_i$, where $\boldsymbol{s}_i$ is the vectorized Logarithm mapping of $\boldsymbol{S}_i$ to Euclidean space. Second, if one prefers to preserve the SPD structure, the transforming can also be defined by the nonlinear map:

$$f(\boldsymbol{V}, \boldsymbol{S}_i) \triangleq \exp\left(\boldsymbol{V}^{\mathrm{T}}\bar{\boldsymbol{X}}\boldsymbol{V}\right), \quad \bar{\boldsymbol{X}} = \log(\boldsymbol{X}), \tag{9}$$

which leads to a linear transform in its Tangent Space as illustrated in Fig. 1. It can be proved that the projected points defined by the mapping (9) also form another Lie Group (Appendix A).

Follow the convention of formulating the discriminant analysis problems [25], one can define the *between class* and *with-in class* cost functions as

$$D^{(w)} \triangleq \frac{1}{2}\sum_{i,j\neq i}^{n}\boldsymbol{W}^{(w)}_{i,j}d^2\left(f(\boldsymbol{V}, \boldsymbol{S}_i), f(\boldsymbol{V}, \boldsymbol{S}_j)\right), \tag{10a}$$

$$D^{(b)} \triangleq \frac{1}{2}\sum_{i,j\neq i}^{N}\boldsymbol{W}^{(b)}_{i,j}d^2\left(f(\boldsymbol{V}, \boldsymbol{S}_i), f(\boldsymbol{V}, \boldsymbol{S}_j)\right). \tag{10b}$$

where $d(f(\boldsymbol{S}_i), f(\boldsymbol{V}, \boldsymbol{S}_j))$ is the corresponding Riemannian distance, and $\boldsymbol{W}^{(w)}_{i,j}$ and $\boldsymbol{W}^{(b)}_{i,j}$ are the within and between weighted matrices characterizing the relation between the samples. $\boldsymbol{W}^{(w)}_{i,j}$ and $\boldsymbol{W}^{(b)}_{i,j}$ are defined in Table 1.

There are several manners to define the weight $a_{i,\,j}$ and $b_{i,\,j}$, such as Fisher Discriminant Analysis (FDA), k-Nearest Neighbor (kNN), Heat kernel (HK), and Local Scaling (LS) [25] which are specified in Table 2. In the case of kNN, only k nearest neighbors of the sample $\boldsymbol{S}_i$ are considered to apply the conditions. In the heat kernel, $\gamma > 0$ is the tuning parameter. In the local scaling, $\gamma_i^{-1} = d(\boldsymbol{S}_i, \boldsymbol{S}^{(k)}_i)$ where $\boldsymbol{S}^{(k)}_i$ is the kth nearest neighbor of $\boldsymbol{S}_i$. Heuristicly, $k = 7$ is recommended [25].

For the chosen weights $\boldsymbol{W}^{(w)}_{i,j}$ and $\boldsymbol{W}^{(b)}_{i,j}$, we seek for an orthonormal matrix $\boldsymbol{V}$, $\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V} = \boldsymbol{I}_d$, that simultaneously minimizes the *with-in class* $D^{(w)}$ and maximizes the *between class* $D^{(b)}$ cost functions, $\boldsymbol{V} = \arg\min_{\boldsymbol{V}}\mathcal{L}(\boldsymbol{V})$, where

$$\mathcal{L}(\boldsymbol{V}) \triangleq D^{(w)} - D^{(b)} \quad \text{or} \quad \mathcal{L}(\boldsymbol{V}) \triangleq \frac{D^{(w)}}{D^{(b)}}. \tag{11}$$

Table 3
Transform map and Jacobian matrix.

Distance	Transform map	Jacobian $\frac{\partial d^2}{\partial \boldsymbol{V}}(\boldsymbol{X}, \boldsymbol{Y})$
Log Euclid	$\hat{\boldsymbol{X}} = \boldsymbol{V}^{\mathrm{T}} \log(\boldsymbol{X})\boldsymbol{V}$	$4(\log(\boldsymbol{X}) - \log(\boldsymbol{Y}))\boldsymbol{V}(\hat{\boldsymbol{X}} - \hat{\boldsymbol{Y}}),$
LogDet	$\hat{\boldsymbol{X}} = \boldsymbol{V}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{V}$	$(\boldsymbol{X}\boldsymbol{V}\hat{\boldsymbol{X}}^{-1} - \boldsymbol{Y}\boldsymbol{V}\hat{\boldsymbol{Y}}^{-1})(\hat{\boldsymbol{X}} - \hat{\boldsymbol{Y}})(\hat{\boldsymbol{X}} + \hat{\boldsymbol{Y}})^{-1}$
Kullback–Leibler	$\hat{\boldsymbol{X}} = \boldsymbol{V}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{V}$	$(\boldsymbol{X}\boldsymbol{V}\hat{\boldsymbol{X}}^{-1} - \boldsymbol{Y}\boldsymbol{V}\hat{\boldsymbol{Y}}^{-1})(\hat{\boldsymbol{X}}\hat{\boldsymbol{Y}}^{-1} - \hat{\boldsymbol{Y}}\hat{\boldsymbol{X}}^{-1})$
Von Neumann	$\hat{\boldsymbol{X}} = \boldsymbol{V}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{V}$	$2(\Delta + \Delta^{\mathrm{T}}),\ \Delta = 2(\boldsymbol{X} - \boldsymbol{Y})\boldsymbol{V}\boldsymbol{V}^{\mathrm{T}}(\log(\boldsymbol{X}) - \log(\boldsymbol{Y}))\boldsymbol{V}$

This optimization problem can be solved by the conjugate gradient descent on Grassmann manifold method [39]. The gradient $\nabla_{\boldsymbol{V}}\mathcal{L}(\boldsymbol{V})$ on the manifold at an iterative step is

$$\nabla_{\boldsymbol{V}}\mathcal{L}(\boldsymbol{V}) = \left(\boldsymbol{I}_D - \boldsymbol{V}\boldsymbol{V}^{\mathrm{T}}\right)\sum_{i,j\neq i}^{n} \frac{\partial\mathcal{L}(\boldsymbol{V})}{\partial d_{ij}^2}\frac{\partial d_{ij}^2}{\partial \boldsymbol{V}},$$

where the Jacobian $\frac{\partial d^2}{\partial \boldsymbol{V}}$ of the squared distance for each metrics is given in Table 3 (proof in Appendix A). The optimization on manifold can be solved efficiently by the Manopt toolbox [40].

The advantage of the proposed discriminant analysis on Riemannian Manifold is that it is applied for multiple classes in a natural way while the conventional CSP is only applied to binary classification problem [38,41]. Hence, it avoids the burden of designing one-vs.-one strategy of CSP.

It is worth to emphasize that Log Det and Kullback Leibler distance are invariant to a linear transform of any full rank matrix $\boldsymbol{P} \in \mathbb{R}^{D\times D}$, i.e.

$$d(\boldsymbol{S}_1, \boldsymbol{S}_2) = d\left(\boldsymbol{P}\boldsymbol{S}_1\boldsymbol{P}^{\mathrm{T}}, \boldsymbol{P}\boldsymbol{S}_2\boldsymbol{P}^{\mathrm{T}}\right).$$

Hence, under the purpose of increasing distance between classes, any blind source separation, such as ICA or CCA, are unnecessary. Therefore, the discrimination can only be improved if $d < D$ when assuming that the noise and artifact effects contained in $D - d$ components are removed.

4.3. Tensor discriminant analysis

Different with manifold, tensor is a multiple dimension array data without strictly constrained structure. Basic definitions and notations of tensor are summarized as follow.

Definition 4.7. An *m-order tensor* is a multidimensional data $\underline{\boldsymbol{X}} \in \mathbb{C}^{n_1\times n_2\times\cdots n_m}$. The tensor $\underline{\boldsymbol{X}}$ can be *unfolded* at a *mode* i into a matrix $\boldsymbol{X}_{(i)} \in \mathbb{C}^{n_i\times\bar{n}_i}$, where $\bar{n}_i = \frac{1}{n_i}\prod_{j=1}^{m} n_j$.

Definition 4.8. The product of a tensor $\underline{\boldsymbol{X}}$ and a matrix $\boldsymbol{V} \in \mathbb{C}^{d\times n_i}$ along the mode-i is denoted as

$$\underline{\boldsymbol{Y}} = \underline{\boldsymbol{X}} \times_i \boldsymbol{V} \Leftrightarrow \boldsymbol{Y}_{(i)} = \boldsymbol{V}\boldsymbol{X}_{(i)}, \qquad \underline{\boldsymbol{Y}} \in \mathbb{C}^{n_1\cdots\times n_{i-1}\times d\times n_{i+1}\cdots\times n_m}.$$

Definition 4.9. An *m-order tensor* $\underline{\boldsymbol{X}} \in \mathbb{C}^{n_1\times n_2\cdots\times n_m}$ can be decomposed by a set of matrix $\boldsymbol{V}_i \in \mathbb{C}^{n_i\times d_i}$ as

$$\underline{\boldsymbol{X}} = \underline{\boldsymbol{G}} \times \{\boldsymbol{V}\} = \underline{\boldsymbol{G}} \times_1 \boldsymbol{V}_1 \cdots \times_m \boldsymbol{V}_m, \qquad \underline{\boldsymbol{G}} \in \mathbb{C}^{d_1\times d_2\cdots\times d_m}.$$

where $\underline{\boldsymbol{G}}$ is the *core tensor*.

All the classical dimension reduction techniques, such as PCA, LPP, or LDA, can be extended to Tensor decomposition [7,42,43]. For example, for the case of second order tensor (matrix) $\boldsymbol{X}_i \in \mathbb{R}^{d_1\times d_2}$, tensor discriminant analysis is formulated as

$$\boldsymbol{V} = \underset{\boldsymbol{V}=\{\boldsymbol{V}_1,\boldsymbol{V}_2\}}{\arg\min} \sum_{i,j\neq i}^{n} \boldsymbol{W}_{i,j}\|\boldsymbol{V}_1^{\mathrm{T}}\boldsymbol{X}_i\boldsymbol{V}_2 - \boldsymbol{V}_1^{\mathrm{T}}\boldsymbol{X}_j\boldsymbol{V}_2\|_F^2, \tag{12}$$

where $\boldsymbol{W}_{ij} = \boldsymbol{W}_{ij}^{w} - \boldsymbol{W}_{ij}^{b}$ is the weight defined similarly in Table 1. In principle, the set of factorization matrix $\{\boldsymbol{V}\}$ are obtained by iteratively finding one mode $\boldsymbol{V}_i$ while fixing other modes. Consequently, the problem for a single mode is simplified to the conventional linear discrimination analysis that can be solved easily by eigenvector decomposition. The optimal $\{\boldsymbol{V}\}$ are selected as the eigenvectors corresponding to the maximal eigenvalues. Hence, an incoming feature $\underline{\boldsymbol{X}}$ can be decomposed to the core tensor $\underline{\boldsymbol{G}}$, where $d_i < n_i$. The final feature is obtained by vectorizing $\underline{\boldsymbol{G}}$ followed by a feature selection. In general, performing dimension reduction directly on tensor space yields better discriminant than that on the Euclidean since it exploits the dissimilarity across dimensions [44].

5. Heterogeneous order relevance composition

Covariance matrix or cross-correlation is a simple and effective method to capture the linear relationship between two multivariate random variables. However, beyond the linear relationship, there may also exist the nonlinear or conditional dependence between them.

As noticed by Wang et al. [45], Covariance matrix can be interpreted as a linear kernel, i.e. the dot product of the feature vector in its original space, e.g. $COV(\boldsymbol{X}) = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$. Hence, this definition can be generalized to any nonlinear kernel, i.e. the dot product in the Hilbert space, e.g., $\boldsymbol{K}(\boldsymbol{X}) = \langle\phi(\boldsymbol{X}), \phi(\boldsymbol{X})\rangle$. Other advantages of Kernels are that they always satisfy the SPD condition regardless of the input vector's dimension and they do not require the explicit map $\phi(\boldsymbol{X})$.

We investigate several commonly used Kernels as follows:

- Linear kernel (covariance): $COV(i, j) = \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j$.
- Polynominal kernel: $K_P(i, j) = \left(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j + a\right)^d, a > 0$.
- Gaussian kernel: $K_G(i, j) = exp\left(-\gamma\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right), \gamma > 0$.
- Rational quadratic: $K_Q(i, j) = \left(1 + \gamma\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right)^{-d}, \gamma > 0$.
- Mutual information (MI)):

$$I(i, j) = \sum_{a\in\boldsymbol{x}_i}\sum_{b\in\boldsymbol{x}_j} p(ab)\log\frac{p(ab)}{p(a)p(b)},$$

To compute MI, we can use the k-neighbors approach [46] implemented in the Information Theoretical Estimators (ITE) toolbox provided by the Szabo [47]. The Mutual Information Matrix (MIM) is not always SPD as pointed out by Jakobsen [48]. However, the counter examples are very specific, while the author also claims that MIM is very often SPD in practice. In our experiment dataset, by using the Energy-weighted MI, i.e. $K_{MI}(i, j) = (e_{x_i}e_{x_j})^{1/2}I(i, j)$, it is always SPD.

To combine these heterogeneous relationships, one can just simply concatenate all matrices into a high-dimension vector. However, considering that the relevance matrices are all symmetric, we proposed two methods, named Heterogeneous Order Relevance Composition (HORC) Tensor and Kernel, to combine these features more efficient as follows:

Definition 5.1. Given a set of m symmetric relevance matrices $\{\boldsymbol{S}_i\}_{i=1}^m, \boldsymbol{S}_i \in \mathbb{R}^{d\times d}$, the *HORC Tensor* $\boldsymbol{H}_T$ is extracted as follows.

$$\boldsymbol{H}_T = [\boldsymbol{H}_1 \ldots \boldsymbol{H}_m]^{\mathrm{T}} \in \mathbb{R}^{m\times l}, \quad l = d(d+1)/2, \tag{13}$$

$$\boldsymbol{H}_i = \begin{cases} \mathrm{vec}(\log_{\mathrm{C}}(\boldsymbol{S}_{\mathrm{i}})), & \text{if} \quad \boldsymbol{S}_{\mathrm{i}} \in \mathrm{Sym}_{\mathrm{d}}^{+}, \\ \mathrm{vec}(\boldsymbol{S}_i), & \text{otherwise.} \end{cases} \tag{14}$$

In HORC Tensor, each column captures different order relevances of a pairwise EEG channels while each row contains a unique relationship across all pairwise channels. Thus, it forms a meaningful tensor, and dissimilarity between HORC features can be performed using Tensor Discriminant Analysis. In practice, to avoid the expensive computation when $l \gg m$, we first reshape the HORC features into a three order tensor $\mathbb{R}^{m\times\frac{d}{2}\times(d+1)}$ or $\mathbb{R}^{m\times\frac{d+1}{2}\times d}$.

Definition 5.2. Given a set of m symmetric relevance matrices $\{\boldsymbol{S}_i\}_{i=1}^{m}, \boldsymbol{S}_i \in \mathbb{R}^{d\times d}$, the *HORC Kernel* $\boldsymbol{H}_K$ is defined as follows.

$$\boldsymbol{H}_K = \sum \beta_{ij} K_j\big(\gamma_j, H_i\big)$$

where H_i is defined in (14), K_j is the Gaussian kernel with kernel width γ_j, and β_{ij} is the optimal weighted of each kernel.

The optimal β_{ij} can be obtained from the multi-kernel Relevant Vector Machine, which performs feature selection and fusion simultaneously.

6. Multi-Class Multi-Kernel relevance vector machine

The aforementioned features can be optimally combined using the Multi-Kernel Learning approach introduced in Section 2.1. Among several kernel-based classifiers, such as Kernel PCA, Kernel Fisher Discrimination Analysis, and Support Vector Machine (SVM) [10], Relevant Vector Machine (RVM) is selected due to the following advantages.

- RVM does not require a positive definite kernel. Thus, we can use Gaussian kernel on any geodesic distances. This is important since not all geometric distance yields SPD Gaussian kernel as noticed by Jayasumana et al. [49].
- The number of basis vectors[1] returned from RVM is much sparser than that of SVM. Hence, incoming data can be classified much faster. This advantage is very meaningful in practice since estimating manifold distances is much more computationally expensive relatively to Euclidean distance.
- RVM does not require a tuning process to avoid the over-fitting problem[2] induced in the other methods. This is critical since the size of training data in BCI applications is often very limited thus over-fitting very likely happens. Therefore, when using SVM, one must experimentally select the penalized parameters through a cross-validation process. However, doing so still cannot guarantee the hyperplane is safe to the outliers.
- RVM returns a probability of a sample belonging to a class. Hence, the results also provide the prediction confidence in contrast with true-or-false results returned from other classifiers.
- RVM is a multiple-class classifier, which is different from the strategy of one-vs.-all or one-vs.-one voting in other binary classifiers, such as SVM.

In this paper, we use the multi-class multi-kernel RVM (mRVM) fast version proposed by Psorakis et al. [9] Damoulas and Girolami [50], and the algorithm is summarized in the Appendix B.

7. Experiment and results

The aforementioned approaches are evaluated by using the datasets IIa from the BCI competition IV [51]. The datasets consist EEG signals from nine subjects, each was asked to perform four different motor imagery tasks: Left hand, right hand, tongue and foot. The EEG signals are recorded and sampled at the rate of 250 Hz using 22 electrodes. The experiment was conducted in two days, and 288 trials were recorded in each day. Each trial contained 7.5 s long samples, in which the trigger cue was shown in the period of [2–3.25] s. The subjects was asked to perform corresponding motor imagery after the cue and maintain for 3 s.

The same preprocessing steps with [6] are applied. Specifically, the signals are first bandpass filtered in [8–30] Hz using 5-order Butterworth filter, and the data epoch $\boldsymbol{X}$ is taken from 2.5 s to 4.5 s of the trial, which yields $\boldsymbol{X} \in \mathbb{R}^{N_c\times T_s}, N_c = 22, T_s = 500$. Missing values (denoted as *NaN* in the dataset) are replaced by its neighbor values, and any trial with more than 20 missing values are excluded. No EOG correction is performed.

In the following experiments, each reported result is an average over fifteen fold trials. In each trial, the whole dataset is randomly participated into two sets: a haft for training and a haft for testing. For each participant, mRVM is run three times to avoid the problem of falling into local maxima, and the highest result is reported for each trial. For Gaussian kernel $exp(-\gamma d^2(\boldsymbol{x}_i, \boldsymbol{x}_j))$, we use multiple kernel widths $\gamma \in [10^{-3}, 0.1]$ with 3–5 different values depending on the distance, and the optimal combination of the parameters are selected by mRVM algorithms.

In the first experiment, we evaluate the performance of each metric described in Section 4.1 using the mRVM classifier with the full dimension covariance matrix $COV \in \text{Sym}^{+}_{D=22}$. The result accuracy is reported in Table 4. As seen from the averaged accuracy across subjects, the performance of the metrics are quite similar. The Log Euclidean distance performs worst (e.g. 62%) as expected due to the unjustified usage of the Identity matrix as the reference point. In contrast, the Tangent Space distance, which is the Log Euclidean distance using the Geometric Mean as the reference point, yields similar results with Kullback–Leiber, Log Det and Von Neumann distance (e.g. 66%). The Tangent Space distance has a disadvantage since its performance depends on a suitable reference point, which may not work well for a scattering dataset or if the data is shifted over time. In contrast, other distances can be computed directly regardless the data point distribution.

To evaluate the computational cost, the algorithms depend on three main steps: (1) build the kernel for training dataset $\boldsymbol{K}_T \in \boldsymbol{R}^{288\times 288}$ using one of the mentioned distance and kernel type, (2) train the RVM classifier to extract the Relevant Vectors and their weights, (3) predict labels for the testing dataset. Table 5 shows the averaged computational time of the mentioned steps, and the averaged number of Relevant Vectors for one cross-validation partition. The test is conducted on a Computer with i7-3930 3.2 Ghz, 16 G RAM. The Log-Euclidean is the fastest since we only need to map the Covariance to the Euclidean space once, and hence can be precomputed efficiently. The Tangent Space distance is slower since it depends on the reference point, which needs to be recomputed for each dataset. The Kullback–Leiber, Log Det and Von Neumann are more computationally expensive. Hence, the time to construct Kernel and predict the labels is significantly longer.

The classification performance of the mRVM is illustrated in Fig. 2. The top four sub-figures show the probability of a test sample belongs to each class, and the bottom shows the classification results based on the maximum probability. The test samples are grouped from class 1 to class 4 for readability purpose. It can be seen that classification for Right Hand and Left Hand is almost perfect, as their probability is approximately at 100% confidence. The classification of tongue and foot is less consistent, and the misclassification often happens when the highest probability is just above 50%. This is important for robotics BCI application as we can neglect a BCI command if its highest probability is less than a certain threshold value. Furthermore, this prediction probability can also serve as the feedback to user.

[1] Basis vector is called “Support Vector” in SVM and “Relevant Vector” in RVM. These basis vectors help determine the classifier boundary.

[2] When a classifier performs very well in training but poorly in testing. This is due the hyperplane overfits to the outliers in the training set.

Table 4
Mean and standard deviation of the classification accuracy (%) using $COV \in \mathrm{Sym}^+_{D=22}$ for different metrics: Tangent Space (TS), Log Euclidean (LE), Log Det (LD), Kullback–Leiber (KL) and Von Neumann (VN) and kernels: Gaussian (Gauss) or Dot Product (dot Prod.). The highest and lowest accuracy are marked as **bold** and *italics*.

	TS + Gauss	TS + dot Prod.	LE + Gauss	LE + dot Prod	LD + Gauss	KL + Gauss	VN + Gauss
Subject 1	77.2 ± 3.8	*76.3* ± 2.0	77.1 ± 1.3	76.7 ± 1.7	78.5 ± 2.8	**79.2** ± 2.3	**79.2** ± 1.3
Subject 2	49.6 ± 2.5	50.0 ± 1.9	*46.7 ± 2.9*	49.8 ± 2.6	49.9 ± 2.1	49.8 ± 2.8	**51.1** ± 2.1
Subject 3	83.6 ± 1.1	83.2 ± 2.2	*77.7 ± 1.7*	80.8 ± 1.3	**84.0** ± 0.6	83.6 ± 1.9	83.7 ± 1.5
Subject 4	57.1 ± 2.8	56.6 ± 2.9	*53.7 ± 3.5*	55.8 ± 2.3	56.1 ± 1.3	**57.5** ± 2.1	54.2 ± 2.2
Subject 5	39.4 ± 2.9	39.4 ± 2.5	35.3 ± 1.5	*34.2 ± 1.9*	**40.3** ± 1.7	39.2 ± 4.3	38.4 ± 2.9
Subject 6	44.6 ± 2.3	40.0 ± 2.6	*37.5 ± 2.1*	39.9 ± 3.7	**44.7** ± 2.3	43.1 ± 4.4	42.0 ± 2.7
Subject 7	78.3 ± 1.9	77.6 ± 1.5	*72.3 ± 1.8*	75.9 ± 1.8	77.7 ± 2.4	76.6 ± 2.7	**78.7** ± 1.6
Subject 8	81.9 ± 1.6	81.3 ± 2.2	*77.7 ± 1.7*	79.9 ± 1.4	81.4 ± 2.5	81.3 ± 1.6	**82.6** ± 1.7
Subject 9	83.2 ± 1.3	82.0 ± 2.5	*80.3 ± 1.8*	81.3 ± 1.7	84.0 ± 1.2	**84.1** ± 1.6	**84.1** ± 1.7
Average	66.0 ± 2.3	65.2 ± 2.3	62.0 ± 2.0	63.8 ± 2.0	66.3 ± 1.9	66.0 ± 2.6	66.0 ± 2.0

Table 5
Averaged computing times (s) for constructing Kernel $\boldsymbol{K}_T \in \boldsymbol{R}^{288 \times 288}$ of the 288 training trials data, training RVM classifer and predicting labels for the 288 testing trials data. Number RVs is the average number of Relevant Vectors or Support Vectors selected from 288 training trials

	TS	TS	LE	LE	LD	KL	VN	TS -SVM
Average	+ Gauss	+ dot Prod.	+ Gauss	+ dot Prod	+ Gauss	+ Gauss	+ Gauss	+ Gauss
Kernel $\boldsymbol{K}_T$ (s)	1.11	1.1	0.03	0.026	2.49	2.52	2.68	1.11
Train classifier(s)	14.1	12.14	16.21	13.34	13.68	10.33	14.31	34.86
Number of RVs	13.3	14.52	16.98	15.7	13.5	14.2	16.2	613.1
Predict (s)	0.004	0.003	0.006	0.003	2.70	2.74	3.44	0.023
Total (s)	15.21	13.24	16.24	13.37	18.87	15.59	20.43	35.99

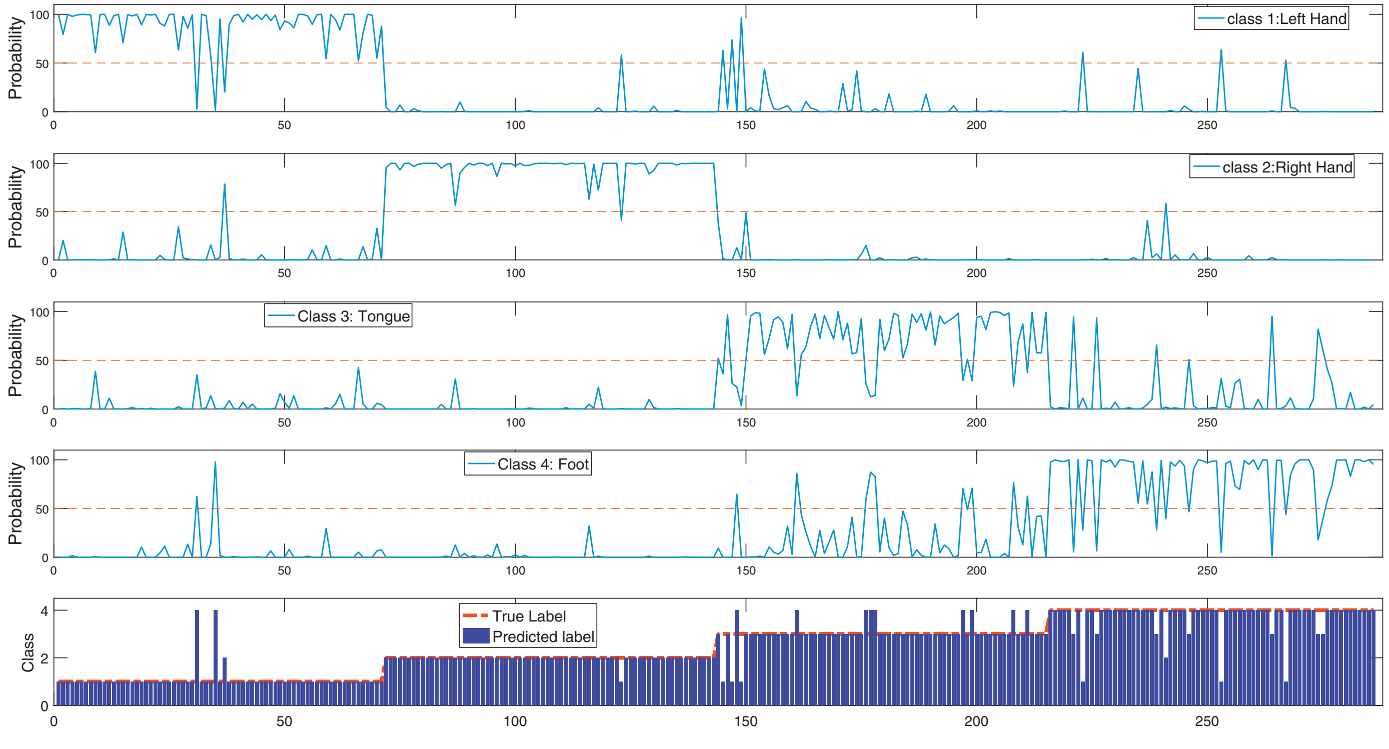


Fig. 2. Classification motor imagery task results and the corresponding confidence of Subject 3.

We also compare the performance of RVM with SVM classifier. Note that, among the considered metrics, only the Log-Euclidean or Tangent Space distance can yield a positive definite Gaussian kernel for all kernel width $\gamma > 0$ [49]. Thus, in this test, only these two metrics can be used with SVM. In our implementation, for each partitioned training dataset, SVM classifier is trained by 10-fold cross validation, and the optimal box-constraint C and kernel width γ are tuned by the Matlab's Bayesian Optimization function *bayesopt* [52]. The bound values are set to $C = [10^{-5}, 10^5]$, $\gamma = [10^{-5}, 10^5]$, and the Matlab function *fitcecoc* is used to train multiple-class SVM. Since there are $n = 4$ classes, we need to train total $n(n-1)/2 = 6$ binary one-vs.-one classifiers. The last column of Table 5 shows the computation time and the number of Support Vectors (SVs), which are significantly larger than that of RVM, due to the large number of binary classifiers combination and the cross-validation process to find the optimal parameters. Note that, while the training set only has 288 data points, SVM needs total 613 points (212%), many of which are redundant, to construct 6 classifier boundaries for 4 classes. In contrast, RVM only requires 14 points (5%) in average, which is very sparse.

Table 6
Mean and standard deviation of the classification accuracy using Tangent Space features and Support Vector Machine with Gaussian kernel.

$\boldsymbol{S}_1$	$\boldsymbol{S}_2$	$\boldsymbol{S}_3$	$\boldsymbol{S}_4$	$\boldsymbol{S}_5$	$\boldsymbol{S}_6$	$\boldsymbol{S}_7$	$\boldsymbol{S}_8$	$\boldsymbol{S}_9$
81.5 ± 1.9	44.8 ± 10.3	84.5 ± 1.3	59.1 ± 2.9	42.2 ± 2.9	52.1 ± 2.3	75.9 ± 2.3	83.1 ± 1.9	84.4 ± 1.4

Table 7
Mean and standard deviation of the classification accuracy using $COV \in \text{Sym}^+_{d<22}$ for different metrics: vectorized Tangent Space (vec(TS)), vectorized Log Euclidean (vec(LE)), Tangent Space (TS), Log Det (LD), Kullback–Leiber (KL) and Von Neumann (VN) using Gaussian kernel. The highest accuracy are marked as **bold**.

	vec(TS)	vec(LE)	TS	LD	KL	VN	vec(TS) + LD/KL
Subject 1	80.4 ± 1.7	80.0 ± 2.1	80.2 ± 2.1	80.1 ± 2.1	81.8 ± 2.1	**82.3** ± 2.3	81.6 ± 1.6
Subject 2	47.6 ± 2.3	47.8 ± 1.9	51.3 ± 2.8	51.5 ± 2.3	50.7 ± 1.8	**51.6** ± 1.5	50.6 ± 1.7
Subject 3	**88.4** ± 1.4	84.8 ± 1.0	83.9 ± 1.0	86.9 ± 1.2	86.5 ± 1.5	85.6 ± 1.4	88.1 ± 1.5
Subject 4	62.1 ± 3.5	60.1 ± 2.2	58.3 ± 2.0	59.3 ± 1.6	57.2 ± 2.0	56.2 ± 2.6	**62.1** ± 2.5
Subject 5	43.2 ± 2.9	40.9 ± 2.0	39.2 ± 2.7	39.9 ± 1.7	36.9 ± 2.0	38.9 ± 2.1	**43.9** ± 1.8
Subject 6	**53.3** ± 2.4	49.3 ± 2.1	50.1 ± 2.2	51.4 ± 1.3	48.9 ± 1.3	45.0 ± 1.5	53.1 ± 2.7
Subject 7	77.9 ± 1.8	76.2 ± 2.0	80.4 ± 2.4	80.6 ± 2.1	**81.8** ± 1.9	81.0 ± 2.2	79.2 ± 2.0
Subject 8	**85.5** ± 1.6	82.3 ± 1.6	83.9 ± 1.8	82.9 ± 1.8	84.3 ± 1.5	83.1 ± 1.6	85.3 ± 1.3
Subject 9	86.7 ± 1.4	83.6 ± 2.1	82.4 ± 1.7	82.7 ± 0.8	83.0 ± 1.9	83.3 ± 2.0	**86.9** ± 1.4
Average	69.5 ± 2.0	67.2 ± 1.9	67.7 ± 2.1	68.4 ± 1.7	67.9 ± 1.8	67.4 ± 1.9	70.1 ± 1.8

In this test, SVM provides slightly higher accuracy, as reported in Table 6. The difference in the performance may due to the selection of the kernel widths. Specifically, while we implement the Bayesian Optimization function to tune the parameters for SVM in each training dataset, RVM only performs kernels fusion using a set of fixed kernel widths. However, notice that RVM does not suffer the over-fitting problem, while SVM relies mainly on tuning the box-constraint parameter *C*. This can be seen in Subject 2, where the over-fitting in SVM still happens in several cross-validation partition, as the minimal accuracy drops to 25.7%, i.e. close to the chance level.

In the second experiment, we evaluate the performance of discriminant analysis methods for each manifold distance (MDA) described in Section 4.2.3.[3] The manifold dimension is reduced to $\text{Sym}^+_{d=12}$ for Log Euclidean, Log Det and Von Neumann distance, and $\text{Sym}^+_{d=16}$ for Kullback–Leibler distance. The within and between weighted matrices $\boldsymbol{W}^{(w)}_{i,j}$ and $\boldsymbol{W}^{(b)}_{i,j}$ are constructed using Local Scaling method with $k = 7$, and the rational cost function is used. The results in terms of accuracy are reported in Table 7. In general, MDA not only reduces the dimension of the manifold but also improves the accuracy by 2–5%. For Tangent Space distance, we apply two techniques: the shrinkage LDA [53,54] on Euclidean space and the MDA on SPD. In this test, we found that the Tangent Vector combined with Shrinkage LDA yields the highest performance by boosting the accuracy approximately 5% comparing with that of the original dimension. The other MDA algorithms increases the accuracy by 2% in most subjects. However, we also observe that MDA occasionally does not improve or even decrease the accuracy such as the case of Subjects 5 and 9. The reason is that shrinkage LDA uses regularized Covariance while MDA does not. Hence, MDA is sensitive to the outliers.

The computational cost of the discriminant analysis algorithm depends on the four main steps: (1) reduce the dimension of the manifold from Sym^+_{22} to $\text{Sym}^+_{d<22}$ using one of the mentioned distance, (2) building the kernel for training dataset $\boldsymbol{K}_T \in \boldsymbol{R}^{288 \times 288}$ using the same distance, (2) train the RVM classifier to extract the Relevance Vectors and their weights, (3) predict labels for the testing dataset. Table 8 shows the averaged computational time of the mentioned steps, and the averaged number of Relevance Vectors for one cross-validation partition.

Table 8
Averaged computing times (s) for constructing Kernel $\boldsymbol{K}_T \in \boldsymbol{R}^{288 \times 288}$ of the 288 training trials data, training RVM classifer and predicting labels for the 288 testing trials data. Num. RVs is the average number of Relevance Vectors selected from 288 training trials.

Average	vec(TS)	vec(LE)	TS	LD	KL	VN
Dim. reduce (s)	2.05	0.94	98.29	219.62	226.35	176.43
Kernel $\boldsymbol{K}_T$ (s)	0.03	0.026	0.18	1.62	1.56	1.69
Train RVM (s)	10.18	6.09	11.56	10.66	9.43	9.01
Num. RVs	7.48	9.7	12.4	11.9	11.4	13.7
Predict (s)	0.002	0.004	0.19	1.7	1.66	2.08
Total	12.26	7.06	110.22	233.6	239.0	189.21

Table 9
Averaged accuracy across subjects using different relevant matrix types.

Kernel	Average
Poly	66.0 ± 1.9
Gauss	69.4 ± 1.9
R. quad	68.4 ± 1.7
M.I	58.9 ± 2.1
$HORC_T$	68.5 ± 2.1
$HORC_K$	70.3 ± 1.6

mRVM can also perform optimal feature fusion using multiple kernels framework. To illustrate this idea, we combine two kinds of distance: the Tangent Space LDA and Log Det or Kullback Leibler coupled with MDA. Note that although they use the same covariance matrix, the two features lie in different spaces and represent different perspectives: manifold geodesics and informative geometry. The result is reported in the second last column in Table 7. It can be seen that the combination approach yields equal or better accuracy relatively to each individual method.

For the HORC feature, we first investigate the performance of each individual component. For Polynomial kernel K_P, we select $a = 1, d = 2$. For Gaussian kernel K_G, we use the kernel width $\gamma = 10^{-4}$. For Rational Quadratic Kernel K_Q, we set $\gamma = 10^{-4}, d = 2$. For Mutual Information (MI) kernel, we use Shannon Entropy function with 8 neighborhood. In this experiment, we do not incorporate frequency information since the bandwidth of motor imagery is quite consistent and well described by α (8–12 Hz) and β (15–25 Hz) bands. As shown in Tables 9 and10, the prediction accuracy when using these kernels as the feature descriptors are quite

[3] We modified the source code published in [39] for our implementation.

Table 10
Mean and standard deviation of the classification accuracy using different kernel forms in place of covariance matrix with Tangent Space vector distance.

Kernel	$\boldsymbol{S}_1$	$\boldsymbol{S}_2$	$\boldsymbol{S}_3$	$\boldsymbol{S}_4$	$\boldsymbol{S}_5$	$\boldsymbol{S}_6$	$\boldsymbol{S}_7$	$\boldsymbol{S}_8$	$\boldsymbol{S}_9$
Poly	78.5 ± 1.5	45.8 ± 1.4	84.9 ± 1.5	60.8 ± 1.8	40.9 ± 1.8	41.2 ± 3.1	76.0 ± 2.6	80.1 ± 2.2	85.7 ± 1.5
Gauss	81.7 ± 2.1	49.9 ± 2.2	86.3 ± 1.0	62.1 ± 1.7	42.1 ± 3.2	54.5 ± 2.2	78.2 ± 1.0	84.4 ± 2.0	85.3 ± 1.9
R. quad	80.5 ± 1.8	50.0 ± 1.9	87.2 ± 1.1	61.2 ± 2.1	43.2 ± 1.3	53.6 ± 2.5	76.9 ± 2.0	81.0 ± 1.0	82.4 ± 1.8
M.I	72.2 ± 1.7	40.0 ± 2.0	75.7 ± 1.0	49.2 ± 2.8	32.6 ± 2.0	43.1 ± 2.9	63.5 ± 1.6	75.7 ± 2.0	78.1 ± 2.8
$HORC_T$	79.5 ± 1.8	48.6 ± 2.4	86.1 ± 1.8	60.2 ± 2.6	41.2 ± 3.6	52.7 ± 1.6	77.4 ± 2.1	84.5 ± 1.5	85.8 ± 1.3
$HORC_K$	81.7 ± 2.1	49.6 ± 1.5	87.8 ± 1.6	62.1 ± 3.0	42.6 ± 1.2	55.4 ± 1.9	80.6 ± 1.1	85.8 ± 1.2	86.9 ± 0.8

equivalent, except for the MI kernel. However, MI kernel's performance is still significantly above the chance level.

The last two rows of Table 10 shows the performance of the HORC descriptor, which combines the covariance matrix *COV*, Gaussian kernel $\boldsymbol{K}_G$, Rational Quadratic Kernels $\boldsymbol{K}_Q$ and Mutual Information kernel K_{MI}. These kernels are selected since they represent different order relevances, such as linear, nonlinear and statistic relationship. For Tensor discrimination analysis, we use Tensor HODA algorithm [7]. As illustrated in Tables 9 and10, the HORC feature yields equal or better performance than each single component. The HORC kernel yields higher accuracy than the HORC Tensor. This is because Tensor Discrimination Analysis only performs linear combination of the features. In contrast, mRVM is a sparse kernel learning that tends to suppress the less discriminative features before fusion, as we observe that the average kernel weights for *COV*, Gauss, Rational Quadratic and Mutual Info are 0.02, 0.65, 0.33 and 0.0, respectively.

Finally, Fig. 3 summarizes Cohen's Kappa values, with the maximum and minimum indication of the mentioned methods for each subject. The Kappa value is defined as

$$\kappa = \frac{P_a - P_c}{1 - P_c},$$

where P_a is the prediction accuracy and P_c is the chance level, i.e. 25% for 4 classes. Based on the Kappa values, we can see that all the prediction are significantly above the chance level.

8. Conclusion

In this paper, we revise the classical EEG features under the perspective of Riemannian Manifold and Tensor. An empirical study to investigate different dissimilarity metrics for covariance matrix, a special class of Riemannian Manifold, is conducted. Specifically, we compare the performance of Tangent Space, Log Euclidean, Log Det divergence, Kullback–Leibler and Von Neumann distance based on the accuracy of classifying four different Motor Imagery tasks. Furthermore, Common Spatial Pattern is generalized to the discriminant analysis in the Riemannian Manifold using different geometric distances. We also extend the Covariance matrix in the original space to the Kernel matrices, which capture different order relevance of the features in the Hilbert space. Two ways of combining these different Relevance matrices, named Heterogeneous Orders Relevance Composition (HORC) Tensor and Kernel, are also examined. The multi-class multi-kernel Relevance Vector Machines is promoted for classification since it offers several unique advantages. Based on Baysian Optimizing Principle, mRVM is a sparse classifier that avoids over-fitting problem and provides the prediction probability for multiple-class in a natural way. Especially, its kernel is not restricted by the Mercer condition, thus allows us to use any distance metrics on the Riemannian Manifold for any radial basic function kernel, such as the Gaussian kernel. Finally, a thorough study has been conducted to evaluate the performance of total sixteen techniques. Our future work will be combining reinforcement learning techniques with mRVM to perform online learning BCI based on the HORC features.

Acknowledgment

This work is supported by the U.S. Defense Advanced Research Projects Agency (DARPA) grant D14AP00068 and U.S. Air Force Office of Scientific Research (AFOSR) award FA9550-14-1-0149.

Appendix A. Proof for Jacobian of manifold distance

A1. Log Euclid distance

The nonlinear mapping (9) forms a Lie group [55] as there exists the multiplication and inverse operation in the projected space

$$\boldsymbol{Y}_i \odot \boldsymbol{Y}_j = \exp\left(\log(\boldsymbol{Y}_i) + \log(\boldsymbol{Y}_j)\right) = \exp\left(\boldsymbol{V}^{\mathrm{T}}(\bar{\boldsymbol{X}}_i + \bar{\boldsymbol{X}}_j)\boldsymbol{V}\right),$$
$$\boldsymbol{Y}^{-1} = \exp\left(-\log(\boldsymbol{Y})\right) = \exp\left(-\boldsymbol{V}^{\mathrm{T}}\bar{\boldsymbol{X}}\boldsymbol{V}\right),$$

where $\bar{\boldsymbol{X}} = \log(\boldsymbol{X})$). Since $\boldsymbol{X}$ and $\boldsymbol{Y}$ are SPD, $\boldsymbol{Y}_i \odot \boldsymbol{Y}_j$ and $\boldsymbol{Y}^{-1}$ are also SPD, which concludes the proof.

The Jacobian of $d^2_{LE}(\boldsymbol{Y}_i, \boldsymbol{Y}_j)$ w.r.t $\boldsymbol{V}$ is given as

$$\frac{\partial d^2_{LE}\left(\boldsymbol{Y}_i, \boldsymbol{Y}_j\right)}{\partial \boldsymbol{V}} = \frac{\partial \operatorname{tr}\left(\boldsymbol{V}^{\mathrm{T}}\left(\bar{\boldsymbol{X}}_i - \bar{\boldsymbol{X}}_j\right)\boldsymbol{V}\boldsymbol{V}^{\mathrm{T}}\left(\bar{\boldsymbol{X}}_i - \bar{\boldsymbol{X}}_j\right)\boldsymbol{V}\right)}{\partial \boldsymbol{V}}$$
$$= 4\left(\bar{\boldsymbol{X}}_i - \bar{\boldsymbol{X}}_j\right)\boldsymbol{V}\boldsymbol{V}^{\mathrm{T}}\left(\bar{\boldsymbol{X}}_i - \bar{\boldsymbol{X}}_j\right)\boldsymbol{V}.$$

A2. Log det distance

It follows Eq. (17) in [39] and the definition in Table 3 that the Jacobian of $d^2_{LE}(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Y}})$ w.r.t $\boldsymbol{V}$ can be obtained as

$$\frac{\partial d^2_{LD}\left(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Y}}\right)}{\partial \boldsymbol{V}} = -\boldsymbol{X}\boldsymbol{W}\hat{\boldsymbol{X}}^{-1} - \boldsymbol{Y}\boldsymbol{W}\hat{\boldsymbol{Y}}^{-1} + 2(\boldsymbol{X} + \boldsymbol{Y})\boldsymbol{W}(\hat{\boldsymbol{X}} + \hat{\boldsymbol{Y}})^{-1}$$
$$= \left(2(\boldsymbol{X}+\boldsymbol{Y})\boldsymbol{W} - \boldsymbol{X}\boldsymbol{W}\hat{\boldsymbol{X}}^{-1}\left(\hat{\boldsymbol{X}}+\hat{\boldsymbol{Y}}\right) - \boldsymbol{Y}\boldsymbol{W}\hat{\boldsymbol{Y}}^{-1}\left(\hat{\boldsymbol{X}}+\hat{\boldsymbol{Y}}\right)\right)\left(\hat{\boldsymbol{X}}+\hat{\boldsymbol{Y}}\right)^{-1}$$
$$= \left(\boldsymbol{X}\boldsymbol{W}\left(1 - \hat{\boldsymbol{X}}^{-1}\hat{\boldsymbol{Y}}\right) + \boldsymbol{Y}\boldsymbol{W}\left(1 - \hat{\boldsymbol{Y}}^{-1}\hat{\boldsymbol{X}}\right)\right)\left(\hat{\boldsymbol{X}} + \hat{\boldsymbol{Y}}\right)^{-1}$$
$$= \left(\boldsymbol{X}\boldsymbol{W}\hat{\boldsymbol{X}}^{-1} - \boldsymbol{Y}\boldsymbol{W}\hat{\boldsymbol{Y}}^{-1}\right)\left(\hat{\boldsymbol{X}} - \hat{\boldsymbol{Y}}\right))\left(\hat{\boldsymbol{X}} + \hat{\boldsymbol{Y}}\right)^{-1}.$$

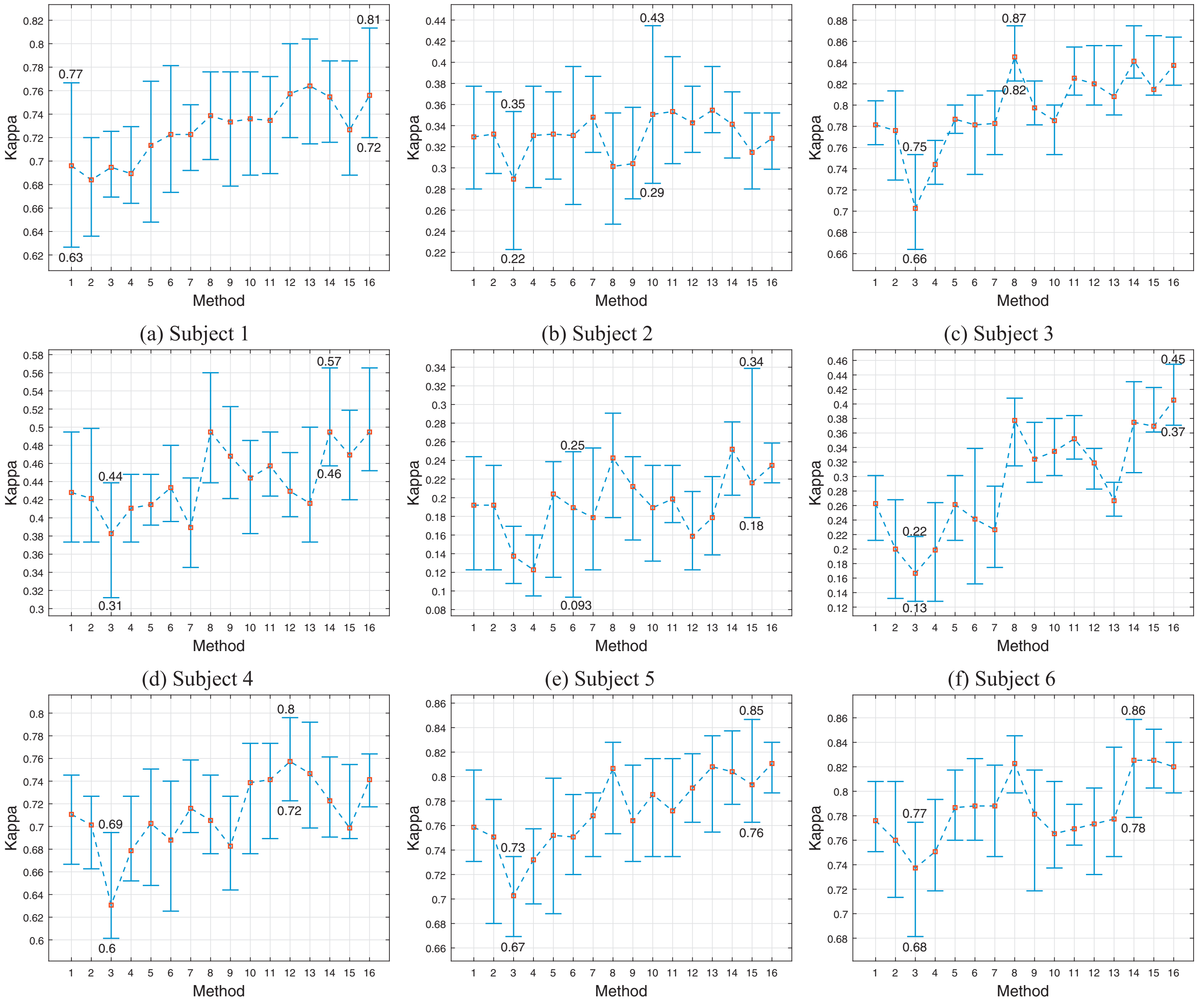
A3. Kullback Leibler distance

It follows Eq. (19) in [39] and the definition in Table 3 that the Jacobian of $d^2_{KL}(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Y}})$ w.r.t $\boldsymbol{V}$ can be obtained as

$$\frac{\partial d^2_{KL}\left(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Y}}\right)}{\partial \boldsymbol{V}} = \boldsymbol{X}\boldsymbol{W}\left(\hat{\boldsymbol{Y}}^{-1} - \hat{\boldsymbol{X}}^{-1}\hat{\boldsymbol{Y}}\hat{\boldsymbol{X}}^{-1}\right) + \boldsymbol{Y}\boldsymbol{W}\left(\hat{\boldsymbol{X}}^{-1} - \hat{\boldsymbol{Y}}^{-1}\hat{\boldsymbol{X}}\hat{\boldsymbol{Y}}^{-1}\right)$$
$$= \boldsymbol{X}\boldsymbol{W}\hat{\boldsymbol{X}}^{-1}\left(\hat{\boldsymbol{X}}\hat{\boldsymbol{Y}}^{-1} - \hat{\boldsymbol{Y}}\hat{\boldsymbol{X}}^{-1}\right) + \boldsymbol{Y}\boldsymbol{W}\hat{\boldsymbol{Y}}^{-1}\left(\hat{\boldsymbol{Y}}\hat{\boldsymbol{X}}^{-1} - \hat{\boldsymbol{X}}\hat{\boldsymbol{Y}}^{-1}\right)$$
$$= \left(\boldsymbol{X}\boldsymbol{W}\hat{\boldsymbol{X}}^{-1} - \boldsymbol{Y}\boldsymbol{W}\hat{\boldsymbol{Y}}^{-1}\right)\left(\hat{\boldsymbol{X}}\hat{\boldsymbol{Y}}^{-1} - \hat{\boldsymbol{Y}}\hat{\boldsymbol{X}}^{-1}\right).$$

A4. Von Neumann distance

Since there is no close form to compute $\frac{\partial \log(\boldsymbol{V}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{V})}{\partial \boldsymbol{V}}$, we utilize the trick in Lemma 6 [39], $\log(\boldsymbol{V}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{V}) \approx \boldsymbol{V}^{\mathrm{T}}\log(\boldsymbol{X})\boldsymbol{V}$, to approximate



(a) Subject 1 (b) Subject 2 (c) Subject 3

(d) Subject 4 (e) Subject 5 (f) Subject 6

(g) Subject 7 (h) Subject 8 (i) Subject 9

Fig. 3. Compare the Cohence's Kappa values between methods: 1. TS +Gauss, 2. TS + dot Prod., 3. LE + Gauss, 4. LE + dot Prod., 5. LD + Gauss, 6. Kullback + Gauss, 7. VN + Gauss, 8. Vec(TS) + Gauss, 9. vec(LE) + Gauss, 10. TS + Gauss, 11. LD + Gauss, 12. KL + Gauss, 13. VN + Gauss, 14. vec(TS) +KL/LD, 15. HORC Tensor, 16. HORC Kernel. The highest and lowest values are specified in each plot.

the Jacobian of $d^2_{VN}(\hat{\boldsymbol{X}},\hat{\boldsymbol{Y}})$ w.r.t $\boldsymbol{V}$ as

$$\frac{\partial d^2_{VN}(\hat{\boldsymbol{X}},\hat{\boldsymbol{Y}})}{\partial \boldsymbol{V}} = \frac{\partial \mathrm{tr}\big(\boldsymbol{V}^{\mathrm{T}}(\boldsymbol{X}_i-\boldsymbol{X}_j)\boldsymbol{V}\boldsymbol{V}^{\mathrm{T}}(\bar{\boldsymbol{X}}_i-\bar{\boldsymbol{X}}_j)\boldsymbol{V}\big)}{\partial \boldsymbol{V}} = \frac{\Delta+\Delta^{\mathrm{T}}}{2},$$

where $\Delta = 2(\boldsymbol{X}-\boldsymbol{Y})\boldsymbol{V}\boldsymbol{V}^{\mathrm{T}}(\log(\boldsymbol{X})-\log(\boldsymbol{Y}))\boldsymbol{V}$.

Appendix B. Summary of multi-class Relevant Vector Machine [9]

Let $\boldsymbol{X}=\{\boldsymbol{x}_i\}_{i=1}^{N}$ be a training set of N observations, each sample $\boldsymbol{x}_i$ has m features $\{\boldsymbol{x}_i^{(j)}\in\mathcal{X}_j\}_{j=1}^{m}$ in its feature space $\mathcal{X}_j$ and a corresponding label $l_i\in\{1,\dots,C\}$, where $C>1$ is the number of classes.

According to (3) and (4), we aim to build a model consisting of a multi-class hyperplane $\boldsymbol{W}\in\mathbb{R}^{N\times C}$ and a multi-kernel weighted vector $\boldsymbol{\beta}\in\mathbb{R}^m$ written as

$$\underbrace{\begin{bmatrix} y_1\\ \vdots\\ y_C\end{bmatrix}}_{\boldsymbol{y}(\boldsymbol{x})\in\mathbb{R}^C} = \underbrace{\begin{bmatrix} w_{11} & \dots & w_{1N}\\ \vdots & \vdots & \vdots\\ w_{C1} & \dots & w_{CN}\end{bmatrix}}_{\boldsymbol{W}^{\mathrm{T}}\in\mathbb{R}^{C\times(N)}} \underbrace{\begin{bmatrix} k_1(\boldsymbol{x}_1,\boldsymbol{x}) & \dots & k_m(\boldsymbol{x}_1,\boldsymbol{x})\\ \vdots & \vdots & \vdots\\ k_1(\boldsymbol{x}_N,\boldsymbol{x}) & \dots & k_m(\boldsymbol{x}_N,\boldsymbol{x})\end{bmatrix}}_{\boldsymbol{K}(\boldsymbol{x})\in\mathbb{R}^{(N)\times m}} \underbrace{\begin{bmatrix} \beta_1\\ \vdots\\ \beta_m\end{bmatrix}}_{\beta\in\mathbb{R}^m}$$

where each element $\boldsymbol{K}_{ij}(\boldsymbol{x})$ is the kernel function evaluated at the training sample $\boldsymbol{x}_i$ using the feature $\boldsymbol{x}_i^{(j)}$ and kernel function $k_j(\cdot,\cdot)$. $\boldsymbol{y}(\boldsymbol{x})=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{K}(\boldsymbol{x})\boldsymbol{\beta}$ is the response of the model to a data sample $\boldsymbol{x}$. For a training sample $\boldsymbol{x}_i\in\boldsymbol{X}$, the response is

$$\boldsymbol{y}(\boldsymbol{x}_i)=[y_1\dots y_c\dots y_C]^{\mathrm{T}},\qquad y_c=\begin{cases}1 & \text{if } l(x_i)=c,\\ 0 & \text{otherwise.}\end{cases}$$

and for a new sample $\boldsymbol{x}$, its label can be predicted as

$$l(\boldsymbol{x})=c,\quad \text{if}\quad \boldsymbol{y}_c(\boldsymbol{x})>\boldsymbol{y}_j(\boldsymbol{x})\quad \forall j\neq c. \tag{B.1}$$

mRVM finds the optimal parameter $\boldsymbol{W}$ and $\boldsymbol{\beta}$ using the Bayesian rule with the following probabilistic constraints.

First, the true label $t_i = l(x_i)$ is assumed to be the measure of the prediction $y(x)$ corrupted by a standardized normal noise $\epsilon \sim \mathcal{N}(0, 1)$, i.e. $l(\boldsymbol{x}) = y(\boldsymbol{x}) + \epsilon$, or

$$P\Big(t_i = c|\boldsymbol{x}_i, \boldsymbol{w}_c^{\mathrm{T}}, \beta\Big) = \mathcal{N}\Big(\boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{K}(\boldsymbol{x}_i)\beta, 1\Big). \tag{B.2}$$

Second, the over-fitting problem is solved by assuming that only a few sample in the training set are representative for its class, while the rest is redundant and safely ignored. This casts the sparsity on $\boldsymbol{W}$, which can be modeled as

$$P(\boldsymbol{W}|\boldsymbol{\alpha}) = \mathcal{N}\Big(\boldsymbol{W}|0, \boldsymbol{\alpha}^{-1}\Big). \tag{B.3}$$

Third, since some features are more important than the other, we can set $\sum_{i=1}^{m} \beta_i = 1,\ \beta_i > 0$, which implies a Dirichlet distribution of $\boldsymbol{\beta}$, i.e.

$$P\Big(\boldsymbol{\beta}|\boldsymbol{\rho}\Big) = \mathrm{Dir}\Big(\boldsymbol{\beta}|\rho_j\Big). \tag{B.4}$$

Hence, the maximal magnitudes of $\boldsymbol{W}$ and $\boldsymbol{\beta}$ are controlled by $\boldsymbol{\alpha}$ and $\boldsymbol{\rho}$, which are also enforced to follow Gamma distribution, i.e.

$$P(\alpha_{ci}|\tau_{ci}, \upsilon_{ci}) = \gamma(\alpha_{ci}|\tau_{ci}, \upsilon_{ci}),\ P(\boldsymbol{\rho}|\mu, \lambda) = \gamma(\boldsymbol{\rho}|\mu, \lambda). \tag{B.5}$$

The parameter $\Xi = [\tau, \upsilon, \mu, \lambda]$ can be automatically tuned as the arguments maximizing the evidence approximation, which is the following marginal likelihood function

$$P(l(\boldsymbol{X})|\boldsymbol{X}, \Xi) = P\Big(l(\boldsymbol{X})|\boldsymbol{X}, \boldsymbol{W}, \boldsymbol{\beta}\Big)P(\boldsymbol{W}|\boldsymbol{\tau}, \boldsymbol{\upsilon})P\Big(\boldsymbol{\beta}|\mu, \lambda\Big) \tag{B.6}$$

where $P(\boldsymbol{W}|\boldsymbol{\tau}, \boldsymbol{\upsilon}) = P(\boldsymbol{W}|\boldsymbol{\alpha})P(\boldsymbol{\alpha}|\boldsymbol{\tau}, \boldsymbol{\upsilon})$ and $P(\boldsymbol{\beta}|\mu, \lambda) = P(\boldsymbol{\beta}|\boldsymbol{\rho})P(\boldsymbol{\rho}|\mu, \lambda)$. Substituting (B.2)–(B.5) to (B.6), one can iteratively update Ξ by following the gradient descent $\partial P(l(\boldsymbol{X})|\boldsymbol{X}, \Xi)/\partial\Xi$. For a new Ξ, one can update the optimal parameters $\boldsymbol{W}^* = \mathrm{argmax}\ P(\boldsymbol{W}|\boldsymbol{\tau}, \boldsymbol{\upsilon})$ and $\boldsymbol{\beta}^* = \mathrm{argmax} P(\boldsymbol{\beta}|\mu, \lambda)$. The process runs iteratively until reaching some convergence conditions.

Finally, for a new sample $\boldsymbol{x}$, its label can be predicted by (B.1) with the confidence

$$P\Big(l(\boldsymbol{x}) = c|\boldsymbol{X}, \boldsymbol{W}^*, \boldsymbol{\beta}^*\Big) = E_\epsilon\left[\prod_{i \neq c} \Phi(\epsilon + (\boldsymbol{w}_c^* - \boldsymbol{w}_i^*)^{\mathrm{T}}\boldsymbol{K}(x)\boldsymbol{\beta}^*\right],$$

where E_ϵ is the expectation along the variable ϵ. More details can be found in [9].[4]

References

[1] S. Sun, J. Zhou, A review of adaptive feature extraction and classification methods for EEG-based brain–computer interfaces, Proc. Int. J. Conf. Neural Netw. (2014) 1746–1753, doi:10.1109/IJCNN.2014.6889525.

[2] F. Lotte, A Tutorial on EEG Signal Processing Techniques for Mental State Recognition in Brain–Computer Interfaces, Springer London (2014).

[3] H. Yuan, B. He, Brain–Computer Interfaces Using Sensorimotor Rhythms: Current State and Future Perspectives, IEEE Trans. Biomed. Eng. 61 (5) (2015) 1425–1435, doi:10.1109/TBME.2014.2312397.Brain-Computer.

[4] A. Lewis, F. Bullo, Geometric Control of Mechanical Systems, 2005.

[5] O. Tuzel, F. Porikli, P. Meer, Pedestrian detection via classification on Riemannian manifolds., Pattern Anal. Mach. Intell. 30 (10) (2008) 1713–1727, doi:10.1109/TPAMI.2008.75.

[6] A. Barachant, S. Bonnet, M. Congedo, C. Jutten, Multiclass brain–computer interface classification by Riemannian geometry, IEEE Trans. Biomed. Eng. 59 (4) (2012) 920–928, doi:10.1109/TBME.2011.2172210.

[7] A.H. Phan, A. Cichocki, Tensor decompositions for feature extraction and classification of high dimensional datasets, Nonlinear Theory Appl. IEICE 1 (1) (2010) 37–68, doi:10.1587/nolta.1.37.

[8] C.C. Aggarwal, Data Classification: Algorithms and Applications, CRC Press, 2014.

[9] I. Psorakis, T. Damoulas, M.A. Girolami, Multiclass relevance vector machines: sparsity and accuracy, IEEE Trans. Neural Netw. 21 (10) (2010) 1588–1598, doi:10.1109/TNN.2010.2064787.

[10] C.M. Bishop, Pattern recognition, Mach. Learn. 128 (2006) 326–344.

[11] P. Comon, Independent component analysis, a new concept? Signal Process. 36 (3) (1994) 287–314, doi:10.1016/0165-1684(94)90029-9.

[12] M. Borga, H. Knutsson, A Canonical Correlation Approach to Blind Source Separation, Technical Report LiU-IMT-EX-0062, Department of Biomedical Engineering, Linköping University, Linköping, Sweden, October 2016.

[13] R.D. Pascual-Marqui, Standardized low-resolution brain electromagnetic tomography (sLORETA): technical details., Methods Find. Exp. Clin. Pharmacol. 24 (Suppl D) (2002) 5–12. 841 [pii].

[14] A.S. Al-fahoum, A.A. Al-fraihat, Methods of EEG signal features extraction using linear analysis in frequency and time-frequency domains, ISRN Neurosci. 2014 (2014) Article ID 730218, 7 pages, 2014. doi:10.1155/2014/730218.

[15] W. Ting, Y. Guo-zheng, Y. Bang-hua, S. Hong, EEG feature extraction based on wavelet packet decomposition for brain computer interface, Measurement 41 (6) (2008) 618–625, doi:10.1016/j.measurement.2007.07.007.

[16] N.E. Huang, Z. Shen, S.R. Long, M.C. Wu, H.H. Shih, Q. Zheng, N.-C. Yen, C.C. Tung, H.H. Liu, The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis, in: Proceedings of the 1998 Royal Society A: Mathematical, Physical and Engineering Science:, vol. 454, The Royal Society, 1998, pp. 903–995.

[17] N. Rehman, D.P. Mandic, Multivariate empirical mode decomposition, Proc. R. Soc. A-Math. Phys. Eng. Sci. 466 (2117) (2010) 1291–1302, doi:10.1098/rspa.2009.0502.

[18] F. Lotte, M. Congedo, A. Lecuyer, F. Lamarche, B. Arnaldi, A review of classification algorithms for EEG-based brain–computer interfaces, J. Neural Eng. 4 (2) (2007) R1–R13, doi:10.1088/1741-2560/4/2/R01.

[19] D.H. Krishna, I.A. Pasha, T.S. Savithri, Classification of EEG motor imagery multi class signals based on cross correlation, Proc. Comput. Sci. 85 (2016) 490–495, doi:10.1016/j.procs.2016.05.198.

[20] S. Siuly, Y. Li, Improving the separability of motor imagery EEG signals using a cross correlation-based least square support vector machine for brain–computer interface, IEEE Trans. Neural Syst. Rehabil. Eng. 20 (4) (2012) 526–538, doi:10.1109/TNSRE.2012.2184838.

[21] S. Martin, W. Rosenstiel, M. Bogdan, Using coherence for robust online brain–computer interface (BCI) Control, Commun. Comput. Inf. Sci. 438 (2014) 363–370, doi:10.1007/978-3-319-08672-9.

[22] A.M. Bastos, J.-M. Schoffelen, A tutorial review of functional connectivity analysis methods and their interpretational pitfalls, Front. Syst. Neurosci. 9 (January) (2016) 1–23, doi:10.3389/fnsys.2015.00175.

[23] B. Hamner, R. Leeb, M. Tavella, J. del R. Millán, Phase-based features for motor imagery brain–computer interfaces, in: Proceedings of the 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS) (Mi), 2011, pp. 2578–2581, doi:10.1109/IEMBS.2011.6090712.

[24] B. Shrestha, I. Vlachos, J.A. Adkinson, L. Iasemidis, Distinguishing motor imagery from motor movement using phase locking value and eigenvector centrality, in: Proceedings of the Thirty-Second Southern Biomedical Engineering Conference (SBEC 2016) (2016), pp. 107–108. doi:10.1109/SBEC.2016.46.

[25] M. Sugiyama, Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis, J. Mach. Learn. Res. 8 (2007) 1027–1061.

[26] M. Sugiyama, T. Ide, S. Nakajima, J. Sese, Semi-supervised local Fisher discriminant analysis for dimensionality reduction, Mach. Learn. 78 (1–2) (2010) 35–61, doi:10.1007/s10994-009-5125-7.

[27] G. Brown, A. Pocock, M.-J. Zhao, M. Lujan, Conditional likelihood maximisation: a unifying framework for mutual information feature selection, J. Mach. Learn. Res. 13 (2012) 27–66, doi:10.1016/j.patcog.2015.11.007.

[28] Y. Wang, S. Gao, X. Gao, Common spatial pattern method for channel selection in motor imagery based brain–computer interface, Eng. Med. Biol. 5 (2005) 5392–5395, doi:10.1109/IEMBS.2005.1615701.

[29] B. Blankertz, B. Blankertz, R. Tomioka, R. Tomioka, S. Lemm, S. Lemm, M. Kawanabe, M. Kawanabe, K.-R. Müller, K.-R. Müller, Optimizing spatial filters for robust eeg single-trial analysis, IEEE Signal Process. Mag. XX (2008) 1–12, doi:10.1109/MSP.2008.4408441.

[30] M. Grosse-Wentrup, M. Buss, Multiclass common spatial patterns and information theoretic feature extraction, IEEE Trans. Biomed. Eng. 55 (8) (2008) 1991–2000, doi:10.1109/TBME.2008.921154.

[31] I. Bengtsson, Lectures on the geometry of quantum mechanics, AIP Conf. Proc. 1360 (2011) 7–24, doi:10.1063/1.3599124.

[32] X. Pennec, P. Fillard, N. Ayache, A Riemannian framework for tensor computing, Int. J. Comput. Vis. 66 (5255) (2006) 41–66, doi:10.1007/s11263-005-3222-z.

[33] V. Arsigny, P. Fillard, X. Pennec, N. Ayache, Geometric means in a novel vector space structure on symmetric positive-definite matrices, SIAM J. Matrix Anal. Appl. 29 (1) (2007) 328–347, doi:10.1137/050637996.

[34] A. Barachant, S. Bonnet, M. Congedo, C. Jutten, Classification of covariance matrices using a Riemannian-based kernel for BCI applications, Neurocomputing 112 (2013) 172–178, doi:10.1016/j.neucom.2012.12.039.

[35] V. Arsigny, P. Fillard, X. Pennec, N. Ayache, Log-Euclidean metrics for fast and simple calculus on diffusion tensors, Magn. Reson. Med. 56 (2) (2006) 411–421, doi:10.1002/mrm.20965.

[36] B. Kulis, M.a. Sustik, I.S. Dhillon, Low-rank kernel learning with Bregman matrix divergences, J. Mach. Learn. Res. 10 (2009) 341–376.

[37] S. Sra, A new metric on the manifold of kernel matrices with application to matrix geometric means, Adv. Neural Inf. Process. Syst. 25 (2012) 144–152.

[38] W. Samek, M. Kawanabe, K.R. Muller, Divergence-based framework for common spatial patterns algorithms, IEEE Rev. Biomed. Eng. 7 (2014) 50–72, doi:10.1109/RBME.2013.2290621.

[4] The Matlab code implementation is published by the author Psorakis https://github.com/ipsorakis/mRVMs .

[39] M. Harandi, M. Salzmann, R. Hartley, Dimensionality reduction on SPD manifolds: the emergence of geometry-aware methods, IEEE Trans. Pattern Anal. Mach. Intell. 2006 (2016) 1–31.

[40] N. Boumal, B. Mishra, P.-A. Absil, R. Sepulchre, Manopt, a matlab toolbox for optimization on manifolds, J. Mach. Learn. Res. 15 (1) (2014) 1455–1459.

[41] F. Lotte, C. Guan, F. Lotte, C. Guan, S. Member, Regularizing common spatial patterns to improve BCI designs : theory and algorithms, IEEE Trans. Biomed. Eng. 58 (2) (2010) 355–362.

[42] X. He, D. Cai, P. Niyogi, Tensor subspace analysis, Adv. Neural Inf. Process. Syst. 18 (2006) 499.

[43] G. Dai, D.-Y. Yeung, Tensor embedding methods, AAAI (2006) 330–335.

[44] F. Cong, Q.H. Lin, L.D. Kuang, X.F. Gong, P. Astikainen, T. Ristaniemi, Tensor decomposition of EEG signals: a brief review, J. Neurosci. Methods 248 (2015) 59–69, doi:10.1016/j.jneumeth.2015.03.018.

[45] L. Wang, J. Zhang, L. Zhou, C. Tang, W. Li, Beyond covariance: feature representation with nonlinear kernel matrices, Proc. IEEE Int. Conf. Comput. Vis. 11–18 (2016) 4570–4578, doi:10.1109/ICCV.2015.519.

[46] A. Kraskov, H. Stögbauer, P. Grassberger, Estimating mutual information, Phys. Rev. E Stat. Nonlinear Soft Matter Phys. 69 (6) (2004) 1–16, doi:10.1103/PhysRevE.69.066138.

[47] Z. Szabó, Information theoretical estimators toolbox, J. Mach. Learn. Res. 15 (2014) 283–287.

[48] S.K. Jakobsen, Mutual information matrices are not always positive semidefinite, IEEE Trans. Inf. Theory 60 (5) (2014) 2694–2696, doi:10.1109/TIT.2014.2311434.

[49] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, M. Harandi, Kernel methods on Riemannian manifolds with Gaussian RBF kernels, IEEE Trans. Pattern Anal. Mach. Intell. 37 (12) (2015) 2464–2477, doi:10.1109/TPAMI.2015.2414422.

[50] T. Damoulas, M.A. Girolami, Probabilistic multi-class multi-kernel learning: o protein fold recognition and remote homology detection, Bioinformatics 24 (10) (2008) 1264–1270, doi:10.1093/bioinformatics/btn112.

[51] G.P.C. Brunner, R. Leeb, G.R. Muller-Putz, A. Schlogl, BCI Competition 2008 Graz Data Set A, Technical Report, Institute for Knowledge Discovery, and Institute for Human–Computer Interfaces Graz University of Technology, Austria, 2008, doi:10.1109/TBME.2004.827081.

[52] Matlab MathWorks, Optimize a Cross-Validated SVM Classifier Using Bayesian Optimization, https://www.mathworks.com/help/stats/bayesian-optimization-case-study.html, 2016.

[53] J. Schäfer, K. Strimmer, A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics., Stat. Appl. Genet. Mol. Biol. 4 (2005) 32, doi:10.2202/1544-6115.1175.

[54] B. Blankertz, S. Lemm, M. Treder, S. Haufe, K.R. Müller, Single-trial analysis and classification of ERP components – a tutorial, Neuroimage 56 (2) (2011) 814–825, doi:10.1016/j.neuroimage.2010.06.048.

[55] C. Xu, C. Lu, J. Gao, W. Zheng, T. Wang, S. Yan, Discriminative Analysis for Symmetric Positive Definite Matrices on Lie Groups, IEEE Trans. Circuits Syst. Video Technol. 25 (10) (2015) 1576–1585, doi:10.1109/TCSVT.2015.2392472.

Chuong Nguyen received a Ph.D. from ME department, Virginia Tech. He is currently a Postdoc researcher at Human Oriented Robotics and Control Lab in Arizona State University. His research interest areas are Brain Computer Interface, Adaptive Control and Rehabilitation Robotics.

Panagiotis Artemiadis received the Diploma and Ph.D. in mechanical engineering from National Technical University of Athens, Greece, in 2003 and 2009, respectively. From 2009 to 2011 he was a Postdoctoral Research Associate at the Newman Laboratory for Biomechanics and Human Rehabilitation, in the Mechanical Engineering Department, Massachusetts Institute of Technology, Boston, MA. Since 2011, he has been with Arizona State University, where he is currently an Associate Professor in the Mechanical and Aerospace Engineering Department, and director of the Human-Oriented Robotics and Control Lab. His research interests lie in the areas of robotics, control systems, system identification, brain-machine interfaces, rehabilitation robotics, neuro-robotics, orthotics, human motor control, mechatronics and human–robot interaction.