Motor Imagery Classification based on Bilinear Sub-Manifold Learning of Symmetric Positive-Definite Matrices

Xiaofeng Xie, Zhu Liang Yu, Haiping Lu, Zhenghui Gu and Yuanqing Li

Abstract

In motor imagery brain-computer interfaces (BCIs), the symmetric positive-definite (SPD) covariance matrices of electroencephalogram (EEG) signals carry important discriminative information. In this paper, we intend to classify motor imagery EEG signals by exploiting the fact that the space of SPD matrices endowed with Riemannian distance is a high-dimensional Riemannian manifold. To alleviate the overfitting and heavy computation problems associated with conventional classification methods on high-dimensional manifold, we propose a framework for intrinsic sub-manifold learning from a high-dimensional Riemannian manifold. Considering a special case of SPD space, a simple yet efficient bilinear sub-manifold learning (BSML) algorithm is derived to learn the intrinsic sub-manifold by identifying a bilinear mapping that maximizes the preservation of the local geometry and global structure of the original manifold. Two BSML-based classification algorithms are further proposed to classify the data on a learned intrinsic sub-manifold. Experimental evaluation of the classification of EEG revealed that the BSML method extracts the intrinsic sub-manifold approximately 5× faster and with higher classification accuracy compared with competing algorithms. The BSML also exhibited strong robustness against a small training dataset, which often occurs in BCI studies.

Index Terms

Electroencephalography (EEG), motor imagery, classification algorithms, information geometry, covariance matrices, dimensionality reduction.

X. Xie, Z. L. Yu, Z. Gu and Y. Li are with the College of Automation Science and Engineering, South China University of Technology, Guangzhou, China, 510641. E-mail: zlyu@scut.edu.cn

H. Lu is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, China.

I. Introduction

BRAIN-computer interfaces (BCIs) provide a new way to translate human intentions into external device commands. BCIs can be used as communication tools for the disabled or as man-machine interface games for healthy people [1]. Many BCI systems have been designed to exploit different types of electroencephalogram (EEG) modalities. In this paper, we will focus on a motor imagery BCI system, in which a trained subject can voluntarily produce an EEG by imagining movements of different parts of the body. Two of the major challenges in motor imagery BCIs are the efficient extraction and correct classification of EEG features.

For the classification of motor imagery signals, common spatial pattern (CSP) [2] is used most frequently as the spatial filter for feature extraction. Taking left/right hand motor imagery as an example, CSP maximizes the variance of one-hand trials while minimizing the variance of the others. Covariance matrices are utilized to obtain the spatial filter. Because the space of symmetric positive-definite (SPD) covariance matrices endowed with Riemannian distance is a Riemannian manifold [3], EEG classification on high-dimensional Riemannian manifolds has recently received increasing attention to improve the performance of the EEG classification [4], [5]. For example, in [5], two algorithms are proposed. One algorithm compares the minimum Riemannian distance between an unlabeled data point and the Riemannian means of labeled data points using the concept of Riemannian geodesic distance. The other algorithm maps all data points in the Riemannian manifold into its tangent space, which is known as the best hyper-plane [6] for classification, and then applies classification methods developed in Euclidean space to the tangent space.

In general, classification in high-dimensional space is subject to overfitting and bias in statistical estimations, particularly for a small training dataset [7], which often occurs in BCI research. The computational cost of these algorithms is another serious limitation. Dimensionality reduction is a promising means of addressing these problems. The goal of dimensionality reduction is to identify a more compact representation of the high-dimensional space. One of the most important nonlinear dimensionality reduction techniques, manifold learning [8], learns the potential intrinsic low-dimensional embedding of the high-dimensional data space. Most manifold learning algorithms attempt to obtain low-dimensional embedding such that proximal data points in high-dimensional space remain proximal and distant data points in high-dimensional space remain distant.

Two canonical approaches in manifold learning are globally mapping methods and locally preserving methods. Global methods, such as isometric feature mapping (Isomap) [9], diffusion maps [10] and

Riemannian manifold learning (RML) [11], tend to identify the global representations of high-dimensional space by preserving the geodesic distance, which is the shortest distance between data points on manifold. One of the earliest global methods, Isomap, uses the shortest path in the graph to approximate the real geodesic, and then uses a multi-dimensional scaling (MDS) [12] algorithm to reduce dimensionality while preserving the approximated geodesic distance. Because the distance between nearby points is calculated as the Euclidean distance, local information of neighbors is lost for sparsely sampled data. Many extensions of Isomap have been proposed to address this limitation, such as incremental Isomap [13], conformal Isomap and landmark Isomap [14]. Diffusion maps [10] replaces the geodesic distance with the diffusion distance and reduces the dimensionality by selecting the first several non-trivial eigenvalues of the transition probability matrix. RML [11] computes a Riemannian normal coordinate chart using PCA projection and then represents the data point in the low-dimensional normal coordinate chart by solving a quadratically constrained linear least squares problem. The low-dimensional representation is computed by preserving radial geodesic distances and angles. Because the shortest paths are exploited to approximate geodesic curves, the RML may have a large error, particularly when the data points are sparsely sampled.

By contrast, local methods, such as locally linear embedding (LLE) [15], Laplacian eigenmaps [16], Hessian eigenmaps [17], manifold charting [18], local tangent space alignment (LTSA) [19] and adaptive manifold learning [20], attempt to preserve the local information of high-dimensional space based on the assumption that each data point and its neighbors are homomorphic to an open subset of Euclidean space. One type of local method is designed to preserve the relationship among proximal data points. LLE [15] characterizes the local geometry in the neighborhood of each data point by linear reconstruction of the data point from its neighbors. Laplacian eigenmaps [16] and Hessian eigenmaps [17] both seek a map in which proximal points in the high-dimensional space are mapped close together in the low-dimensional embedding. These local methods use the eigenfunctions of different operators, the Laplace Beltrami operator and Hessian matrix. The other type of local method is designed to obtain more simple coordinate systems. The manifold chart [18] decomposes the sampled data space into locally linear low-dimensional patches and merges these patches into a single low-dimensional coordinate system. LTSA [19] identifies the tangent space for each locally linear patch of manifold and then aligns those tangent spaces to obtain a parameterization of manifold. An extension of the LTSA algorithm, adaptive manifold learning [20], modifies the minimization model in LTSA and adaptively selects the neighbors of each data point. The local methods capture information only from the local patch and ignore the information of the global structure of data in processing.

Although many global and local methods have been proposed to identify low-dimensional embedding, they are mainly designed for a general manifold and few of these methods use the information of the manifold from which the original data were sampled. Without information on the geodesic of the unknown manifold, most global methods learn the low-dimensional embedding by approximating the geodesic distance, which results in a representation bias. In many applications of pattern recognition, the data can be represented by covariance matrices, which are SPD matrices. Because the space of the SPD matrices endowed with Riemannian distance is a Riemannian manifold [3], in this paper, we focus on examining a type of Riemannian manifold, in which the space of SPD matrices is endowed with explicit geodesic distance.

Considering the space of SPD matrices in motor imagery BCIs, we propose a novel dimensionality reduction method, bilinear sub-manifold learning (BSML). The main difference between BSML and other manifold learning methods is that BSML directly preserves the pairwise Riemannian geodesic distance between data points instead of approximating the geodesic distance, as shown in Fig. 1. Bilinear learning algorithms have been recently proposed for BCI applications. In motor imagery systems, popular bilinear methods include discriminative filter bank CSP (DFBCSP) [21], which simultaneously optimizes the spatial and temporal filters, and the more recent method of separable common spatio-spectral pattern (SCSSP) [22], which seeks the spatio-spectral features by matrix-variate Gaussian model. In particular, bilinear methods are also used for event-related potential (ERP) classification [23], [24], where the discriminant information of ERP signal is obtained by learning a spatial matrix and a temporal matrix collaboratively. Most of these studies have achieved great success in BCI applications. However, the above bilinear methods identify two projection matrices on Euclidean space and ignore that the covariance matrix lies on a Riemannian manifold. BSML algorithm learns two projection matrices using Riemannian geometry.

The major contributions of this paper are threefold.

1) A novel BSML is proposed for dimensionality reduction of the SPD matrices space in motor imagery BCIs. Calculation of the intrinsic sub-manifold is formulated as an eigenvalue problem. The sub-manifold is efficiently extracted by minimizing the Riemannian geodesic distance loss between any pair of data points on the original manifold and its intrinsic sub-manifold. Our method is specifically designed for the space of SPD matrices, and differs from the RML [11], which addresses arbitrary data space. The BSML can be considered as an extension of CSP on covariance matrices in measure of Riemannian distance.
2) Two classification algorithms, minimum distance to sub-manifold mean (MDSM) and tangent space

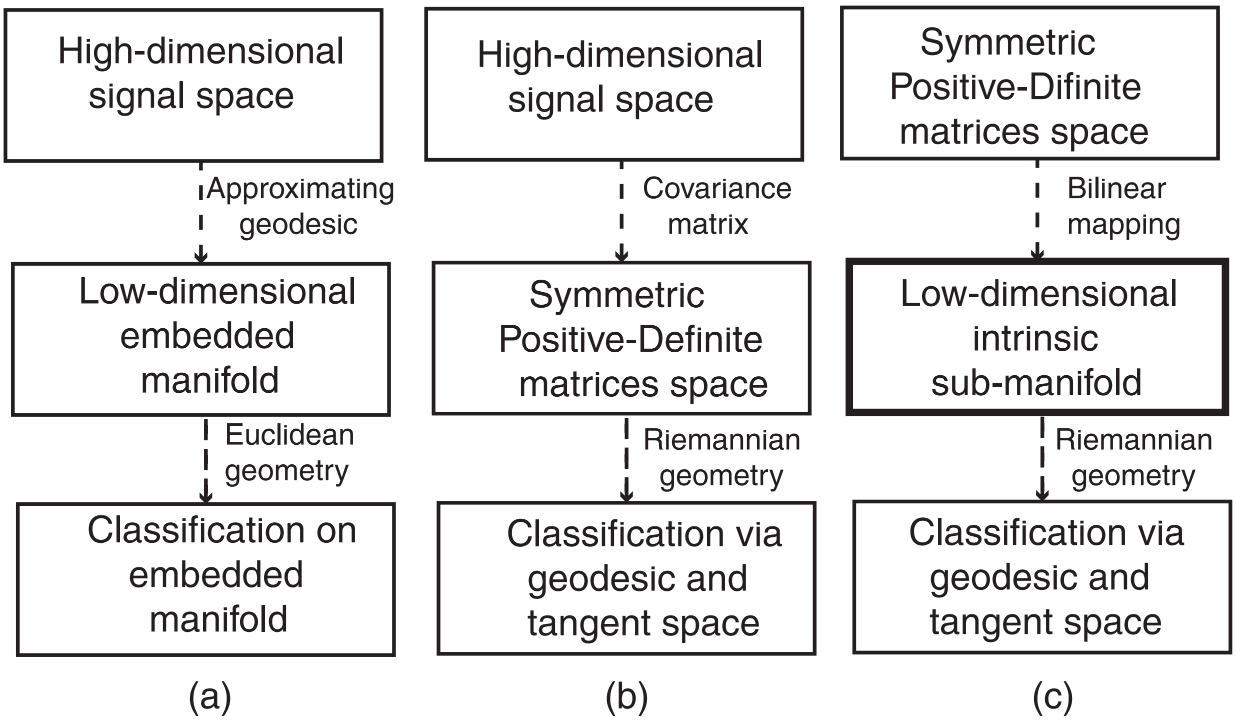

High-dimensional signal space
Approximating geodesic
Low-dimensional embedded manifold
Euclidean geometry
Classification on embedded manifold
(a)
High-dimensional signal space
Covariance matrix
Symmetric Positive-Definite matrices space
Riemannian geometry
Classification via geodesic and tangent space
(b)
Symmetric Positive-Difinite matrices space
Bilinear mapping
Low-dimensional intrinsic sub-manifold
Riemannian geometry
Classification via geodesic and tangent space
(c)

Figure 1. Related work and proposed framework. a) the framework of general manifold learning; b) classification on a high-dimensional Riemannian manifold; c) the framework of Bilinear Sub-Manifold Learning, classification on a low-dimensional sub-manifold.

of sub-manifold (TSSM), are proposed to function on the extracted Riemannian sub-manifold. Higher classification performance is obtained for motor imagery BCIs.

3) For small sample sizes, i.e., when the ratio of the number of training samples to the number of features is small, the BSML algorithm can efficiently alleviate the overfitting problem, as supported by experimental results.

The remainder of the paper is organized as follows. In Section II, some basic concepts of SPD matrices space are briefly reviewed. In Section III, we derive a framework for intrinsic mapping for the problem of dimensionality reduction. Based on this framework, we propose a simple method for intrinsic sub-manifold learning for the SPD matrices space, i.e., the BSML method. Two classification algorithms are proposed on the intrinsic manifold. Extensive experimental results are provided in Section IV to demonstrate the effectiveness of the proposed method. Finally, some conclusions are provided in Section V.

II. Data Model and Basic Concepts of SPD Matrices Space

A. Data Model of BCI

The recorded EEG signal of motor imagery BCIs is a multi-lag/multi-channel signal

$$\mathbf{X}(t) = [\mathbf{x}(t), ..., \mathbf{x}(t + L - 1)] \in \mathbb{R}^{N \times L} \tag{1}$$

where N and L indicate the number of channels and sampled points, respectively. The vector $\mathbf{x}(t) = [x_1(t), ..., x_N(t)]^T \in \mathbb{R}^N$ is the snapshot vector. In motor imagery BCIs, the second order statistical information of $\mathbf{X}(t)$ often provides discriminative information for brain states. Among the second order statistical information [25], the spatial covariance matrix of EEG data is widely used in motor imagery BCIs [26]. In this paper, the spatial covariance matrix of EEG data $\mathbf{P}(t)$, which is represented by its sample covariance matrix (SCM), is defined as

$$\mathbf{P}(t) = \frac{1}{L-1}\mathbf{X}(t)\mathbf{X}^T(t). \tag{2}$$

Generally, the classification of $\mathbf{P}(t)$ is always directly performed in Euclidean space [27], where the Euclidean distance of SCM is used. The SCM is a SPD matrix. Because the space of SPD matrices endowed with Riemannian distance is a differentiable Riemannian manifold, Riemannian geometry can be used to analyze SCM. Many concepts and tools, such as Riemannian distance, tangent space and Riemannian mean, which are briefly reviewed in the following section, can be readily applied in the classification of SCM.

B. Basic Concepts of SPD Matrices Space

Denoting the space of symmetric matrices

$$\mathcal{S}(N) = \left\{\mathbf{P} \in \mathbb{R}^{N \times N}, \mathbf{P} = \mathbf{P}^T\right\} \tag{3}$$

and the space of positive-definite matrices

$$\mathcal{P}(N) = \left\{\mathbf{P} \in \mathbb{R}^{N \times N}, \mathbf{u}^T\mathbf{P}\mathbf{u} > 0, \forall \mathbf{u} \in \mathbb{R}^N\right\}, \tag{4}$$

the space of SPD matrices is defined as

$$\mathcal{SPD}(N) = \mathcal{S}(N) \cap \mathcal{P}(N). \tag{5}$$

The SPD matrix lies on a differentiable Riemannian manifold [3]. Thus, many of the mathematical concepts defined in Riemannian geometry can be applied to $\mathcal{SPD}(N)$.

The Riemannian distance $\delta_R(\mathbf{P}_1, \mathbf{P}_2)$ is the minimum length of the curve connecting $\mathbf{P}_1$ and $\mathbf{P}_2$ on a Riemannian manifold [27]. There are many possible mathematical definitions of the Riemannian distance [28]. In this paper, we adopt the Riemannian distance between two matrices $\mathbf{P}_1, \mathbf{P}_2 \in \mathcal{SPD}(N)$ as [3]

$$\delta_R(\mathbf{P}_1, \mathbf{P}_2) = \left\|\log({\mathbf{P}_1}^{-1}\mathbf{P}_2)\right\|_F = \left[\sum_{i=1}^{N} \log^2 \beta_i\right]^{\frac{1}{2}} \tag{6}$$

where $|| \cdot ||_F$ is the Frobenius norm of a matrix and β_i is the i-th real eigenvalue of ${\mathbf{P}_1}^{-1}\mathbf{P}_2$. The Riemannian distance poses three fundamental properties of metric space: positivity, symmetry and triangle inequality [3]. One of the most important properties of the Riemannian distance is the invariance of linear transformation [3]

$$\delta_R(\mathbf{P}_1, \mathbf{P}_2) = \delta_R(\mathbf{W}^T\mathbf{P}_1\mathbf{W}, \mathbf{W}^T\mathbf{P}_2\mathbf{W}) \tag{7}$$

where the transformation matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$ is invertible.

The tangent space, as a Euclidean space, is an important space in the analysis of a Riemannian manifold. Before defining the tangent space, we first introduce the logarithmic mapping operator and exponential mapping operator

$$\begin{aligned} Log_{\mathbf{P}}(\mathbf{P}_i) &= \mathbf{S}_i = \mathbf{P}^{\frac{1}{2}} \log(\mathbf{P}^{-\frac{1}{2}}\mathbf{P}_i\mathbf{P}^{-\frac{1}{2}})\mathbf{P}^{\frac{1}{2}}, \\ Exp_{\mathbf{P}}(\mathbf{S}_i) &= \mathbf{P}_i = \mathbf{P}^{\frac{1}{2}} \exp(\mathbf{P}^{-\frac{1}{2}}\mathbf{S}_i\mathbf{P}^{-\frac{1}{2}})\mathbf{P}^{\frac{1}{2}}. \end{aligned} \tag{8}$$

$Log_{\mathbf{P}}(\cdot)$ is a mapping from the manifold to the tangent space at $\mathbf{P}$, whereas $Exp_{\mathbf{P}}(\cdot)$ is a mapping from the tangent space at $\mathbf{P}$ to manifold. These two operators are a pair of one-to-one mapping operators between the Riemannian manifold and the tangent space. The logarithm $\log(\mathbf{P})$ and exponential $\exp(\mathbf{P})$ of a SPD matrix $\mathbf{P}$ in (8) are defined as follows. If the eigenvalue decomposition of $\mathbf{P}$ is $\mathbf{P} = \mathbf{U}diag(\sigma_1, ...\sigma_n)\mathbf{U}^T$, where $\sigma_1, ...\sigma_n$ are the eigenvalues of $\mathbf{P}$ and $\mathbf{U}$ is the eigenvector matrix, then $\log(\mathbf{P}) = \mathbf{U}diag(\log(\sigma_1), ... \log(\sigma_n))\mathbf{U}^T$ and $\exp(\mathbf{P}) = \mathbf{U}diag(\exp(\sigma_1), ... \exp(\sigma_n))\mathbf{U}^T$.

Because the tangent space $\mathcal{T}_{\mathbf{P}}(N) = \{Log_{\mathbf{P}}(\mathbf{P}_i), \mathbf{P}_i \in \mathcal{SPD}(N)\}$ is a space of symmetric matrices, there are only $N(N+1)/2$ independent elements. We can find a minimal representation of the tangent space $\mathcal{T}_{\mathbf{P}}(N)$ at $\mathbf{P}$ as a vector space [6]

$$\mathcal{T}(N) = \left\{\mathbf{s}_i = upper(\mathbf{P}^{-\frac{1}{2}} Log_{\mathbf{P}}(\mathbf{P}_i)\mathbf{P}^{-\frac{1}{2}}) \in \mathbb{R}^{N(N+1)/2}\right\} \tag{9}$$

where upper$(\cdot)$ operator retains the upper triangular portion of the symmetric matrix and vectorizes it.

An important property of the tangent space is that the Riemannian distance from any point $\mathbf{P}_i$ to the point $\mathbf{P}$ can be calculated as the Euclidean distance on the tangent space at $\mathbf{P}$ [5]

$$\delta_R(\mathbf{P}, \mathbf{P}_i) = \|\mathbf{s}_i - 0\|_2 \tag{10}$$

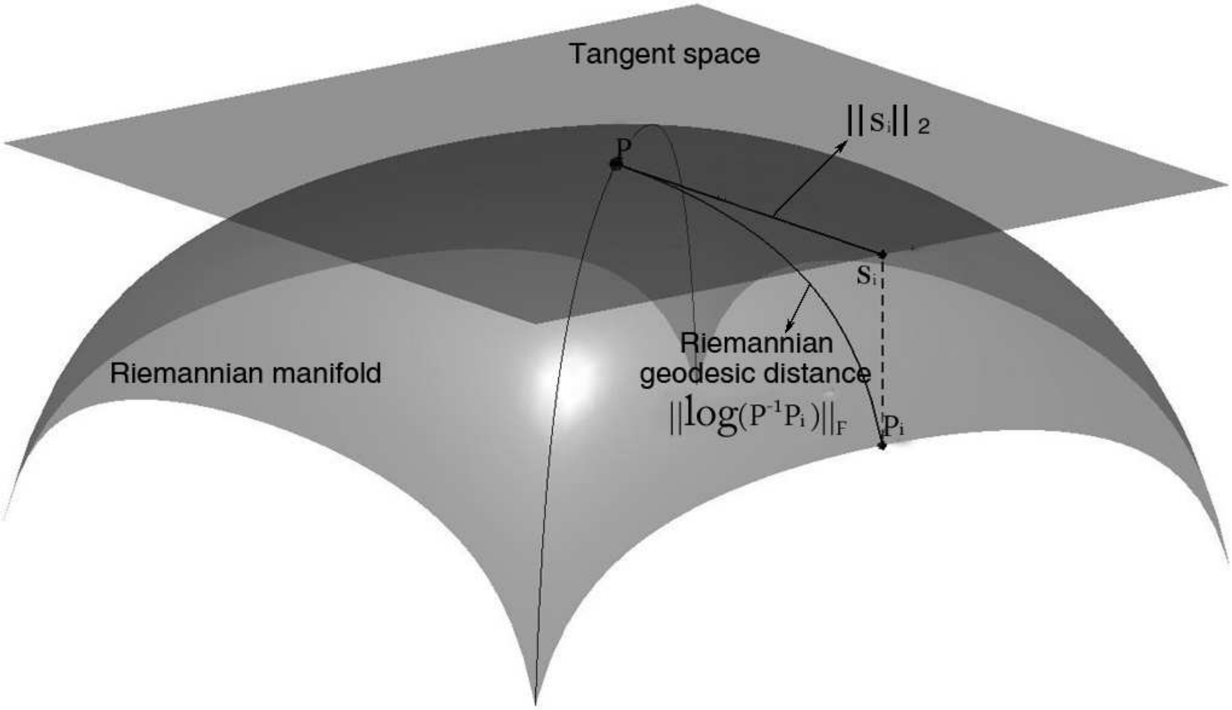


Figure 2. An illustration of the Riemannian manifold and tangent space.

where $\mathbf{s}_i \in \mathcal{T}(N)$ is the vector in tangent space corresponding to $\mathbf{P}_i \in \mathcal{SPD}(N)$. The Riemannian manifold and tangent space are illustrated in Fig. 2.

The mean of SPD matrices plays an important role in classification and is defined as the point $\mathbf{P}_R \in \mathcal{SPD}(N)$, which has a minimum sum of the squared distances to all SPD matrices in dataset C

$$\mathbf{P}_R = \arg \min_{\mathbf{P} \in \mathcal{SPD}(N)} \sum_{\mathbf{P}_i \in C} \delta^2{}_R(\mathbf{P}, \mathbf{P}_i). \tag{11}$$

In the literature [29], the mean $\mathbf{P}_R$ is also referred to as the Riemannian mean. Directly calculating the Riemannian mean is not easy, and an alternative method is to calculate it using the relationship between the Euclidean distance and Riemannian distance [30]. While the Riemannian mean is calculated, the tangent space at the Riemannian mean is the best hyperplane in which the distance loss is minimal for classification [6].

III. Proposed Methods

Similar to the classification on Euclidean space, the classification on the Riemannian manifold (see [31]) also suffers from longstanding problems of high-dimensionality, such as overfitting and bias in statistical estimations, particularly in the case of a small sample setting. To address these problems, we introduce a dimensionality reduction approach to identify a low-dimensional intrinsic sub-manifold that maximizes the preservation of the local geometry and global structure of the original manifold.

A. Framework of Intrinsic Mapping

Suppose operator f is a smooth mapping that maps a data point on a Riemannian manifold of N dimensions to a Riemannian manifold of M dimensions:

$$f : \Omega^N \to \Omega^M (N > M). \tag{12}$$

If Ω^M is a subset of Ω^N, then Ω^M is an embedded sub-manifold of Ω^N [28]. The embedded sub-manifold is modeled locally on the standard embedding of $\mathbb{R}^M$ into $\mathbb{R}^N$, identifying $\mathbb{R}^M$ with the subset $\{(x^1, ...x^M,\ x^{M+1}, ...x^N)|x^{M+1} = ... = x^N = 0\}$ of $\mathbb{R}^N$.

With different choices of f, there exist many sub-manifolds. Our target is to identify the intrinsic sub-manifold, which is the sub-manifold that maximizes the preservation of the local geometry and global structure of the original manifold. Because the local geometry and global structure can be represented by the metric structure of the manifold, we define the intrinsic mapping by minimizing the Riemannian geodesic distance loss between any pair of data points on the original manifold and its intrinsic sub-manifold as

$$f_{opt} = \arg\min_{f} \int\limits_{\Omega} |\delta_R(\rho_i, \rho_j) - \delta_R(f(\rho_i), f(\rho_j))| d\Omega \tag{13}$$

where ρ_i is a point on the manifold and $f(\rho_i)$ is the mapped point on its sub-manifold. The sub-manifold learned from (13) is defined as the intrinsic Riemannian sub-manifold, as illustrated in Fig. 3.

With different types of original manifolds, the optimal intrinsic mapping f of (13) has many variations, such as complicated nonlinear mapping and linear mapping. In this paper, considering the SPD matrices space, based on simple geometric intuition and to facilitate implementation, we characterize the intrinsic mapping operator f as a bilinear mapping with mapping matrix $\mathbf{W}_s \in \mathbb{R}^{M\times N}$. Once $\mathbf{P} \in \mathbb{R}^{N\times N}$ and the mapping matrix $\mathbf{W}_s$ are given, we can directly obtain the mapped SPD matrix, $\mathbf{P}_c = \mathbf{W}_s\mathbf{P}\mathbf{W}_s{}^T \in \mathbb{R}^{M\times M}$. This mapped SPD matrices space is also a differentiable Riemannian sub-manifold [3].

The intrinsic mapping matrix $\mathbf{W}_s$ can be obtained by minimizing the distance loss as

$$\mathbf{W}_{sopt} = \arg\min_{\mathbf{W}_s} \sum_{\mathbf{P}_i, \mathbf{P}_j \in C} \left|\delta_R(\mathbf{P}_i, \mathbf{P}_j) - \delta_R(\mathbf{W}_s\mathbf{P}_i\mathbf{W}_s{}^T, \mathbf{W}_s\mathbf{P}_j\mathbf{W}_s{}^T)\right| \tag{14}$$

where $\mathbf{C}$ is the experimental dataset of matrices in $\mathcal{SPD}(N)$.

It should be noted that (14) is a non-convex problem and is difficult to solve. In this paper, (14) is approximated as a simple eigenvalue optimization problem.

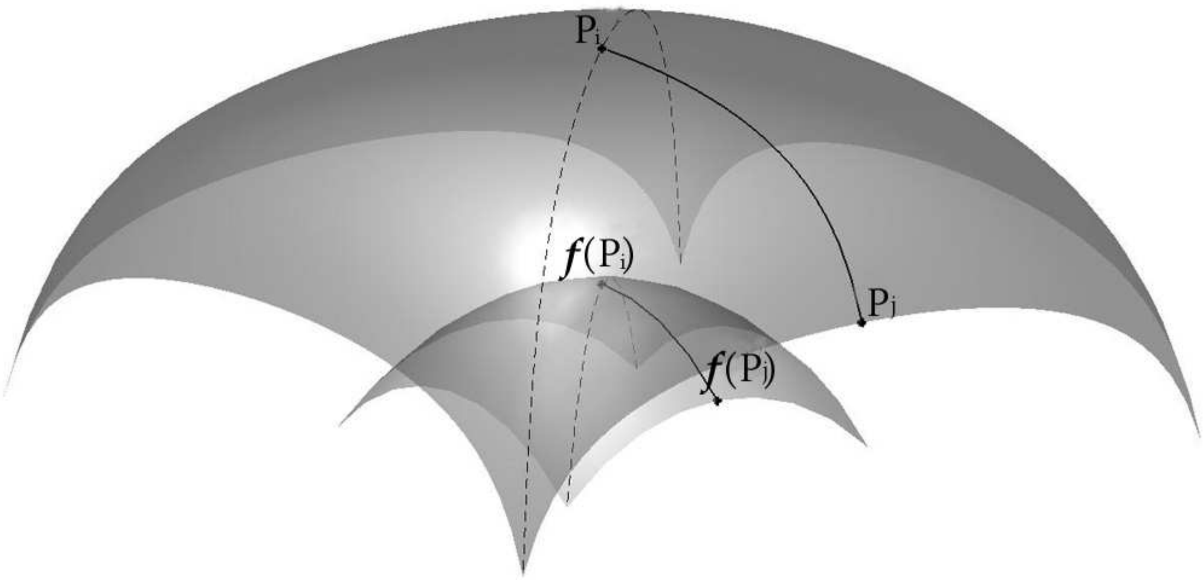


Figure 3. An illustration of the Riemannian manifold (larger one) and intrinsic Riemannian sub-manifold (smaller one).

B. Bilinear Sub-Manifold Learning Algorithm

Normally, (14) is an intractable problem and the optimal solution is difficult to identify. In this section, considering the two-class classification problem, i.e., left/right motor imagery problem in BCI, the BSML algorithm is proposed to solve (14) approximately. As shown in Appendix I, (14) can be approximated as

$$\mathbf{W}_{sopt} = \arg\min_{\mathbf{W}_s} \left| \delta_R(\mathbf{P}_{R1}, \mathbf{P}_{R2}) - \delta_R(\mathbf{W}_s \mathbf{P}_{R1} {\mathbf{W}_s}^T, \mathbf{W}_s \mathbf{P}_{R2} {\mathbf{W}_s}^T) \right|. \tag{15}$$

where $\mathbf{P}_{R1}, \mathbf{P}_{R2}$ are the Riemannian means of two-class datasets

$$\begin{aligned} \mathbf{P}_{R1} &= \arg\min_{\mathbf{P}} \sum_{\mathbf{P}_i \in C_1} \delta_R^2(\mathbf{P}, \mathbf{P}_i) \\ \mathbf{P}_{R2} &= \arg\min_{\mathbf{P}} \sum_{\mathbf{P}_i \in C_2} \delta_R^2(\mathbf{P}, \mathbf{P}_i) \end{aligned} \tag{16}$$

and C_1, C_2 represent the datasets of class 1 and class 2, respectively. Because the Riemannian mean generalizes naturally to a finite set of SPD matrices, the approximated problem (15) attempts to preserve Riemannian distance between the Riemannian means of two-class datasets instead of the entire set of data points.

As shown in Appendix II, we can identify an invertible matrix $\mathbf{W}$ that jointly diagonalizes $\mathbf{P}_{R1}, \mathbf{P}_{R2}$ and satisfies

$$\mathbf{W}\mathbf{P}_{R1}\mathbf{W}^T + \mathbf{W}\mathbf{P}_{R2}\mathbf{W}^T = \mathbf{I}. \tag{17}$$

The corresponding eigenvalues $\lambda_{j1}, \lambda_{j2} (j = 1, \cdots, N)$ of the diagonal matrices $\mathbf{W}\mathbf{P}_{R1}\mathbf{W}^T$ and $\mathbf{W}\mathbf{P}_{R2}\mathbf{W}^T$ are subject to $\lambda_{j1} + \lambda_{j2} = 1$ [2], [32]. After obtaining the transformation matrix $\mathbf{W}$, the mapping matrix $\mathbf{W}_s \in \mathbb{R}^{M\times N}$ can be constructed by choosing different combination of row vectors from the transformation matrix $\mathbf{W}$. According to the invariance of linear transformation and the definition of the Riemannian distance, the optimization problem (15) is expressed as

$$\{\lambda\}_{opt} = \arg\min_{\{\lambda_i\}} \left| \sqrt{\sum_{j=1}^{N} \log^2(\frac{\lambda_{j1}}{1-\lambda_{j1}})} - \sqrt{\sum_{i=1}^{M} \log^2(\frac{\lambda_i}{1-\lambda_i})} \right| \tag{18}$$

where $\{\lambda_i, i \in [1, M]\}$ is subset of $\{\lambda_{j1}, j \in [1, N]\}$. For different dimensions M, the solution of (18) can be obtained by choosing the first M eigenvalues that are far from 0.5. Each combination of corresponding row vectors from the transformation matrix $\mathbf{W}$ is a mapping matrix $\mathbf{W}_s \in \mathbb{R}^{M\times N}$.

The problem remaining here is the selection of the dimension M of the intrinsic sub-manifold. Similar to [9], we estimate M from the elbow of the relative error of the Riemannian distances before-and-after mapping for different dimensions of sub-manifold. The relative error E_r is defined as

$$E_r = 1 - \left(\frac{\delta_R\left(\mathbf{W}_s\mathbf{P}_{R1}\mathbf{W}_s{}^T, \mathbf{W}_s\mathbf{P}_{R2}\mathbf{W}_s{}^T\right)}{\delta_R\left(\mathbf{P}_{R1}, \mathbf{P}_{R2}\right)} \right). \tag{19}$$

The intrinsic dimensionality is located at the largest curvature of the error curve at which the error curve ceases to decrease significantly with increasing dimensionality [9]. The pseudo-code of the BSML is given in Algorithm 1.

Algorithm 1 Bilinear sub-manifold learning (BSML)

```
Input: The training SPD matrix samples $\mathbf{P}_{Tr} \in \mathbb{R}^{N\times N}$;
Output: The optimal dimensionality $M_s$, and intrinsic mapping matrix $\mathbf{W}_s \in \mathbb{R}^{M_s\times N}$;
1: Calculate the Riemannian means $\mathbf{P}_{R1}$, $\mathbf{P}_{R2}$ from the training data as (16);
2: Calculate the $N \times N$ normalization matrix $\mathbf{W}$
   $(\mathbf{P}_{R1} + \mathbf{P}_{R2})^{-1}\mathbf{P}_{R1} = \mathbf{W}\Sigma_1\mathbf{W}^T, \mathbf{W} \in \mathbb{R}^{N\times N}$ (See Append. II);
3: For different numbers $M$, select $M$ eigenvalues $\{\lambda_i, i \in [1, M]\}$ far from 0.5 from $\{\lambda_{j1}, j \in [1, N]\}$,
   and construct mapping matrix $\mathbf{W}_s \in \mathbb{R}^{M\times N}$ as the corresponding submatrix of $\mathbf{W}$;
4: Select the optimal dimensionality $M_s$ of the intrinsic sub-manifold based on the error curve (19);
```

C. Classification on the Intrinsic Riemannian Sub-manifold

Two classification algorithms, e.g., minimum distance to Riemannian mean (MDRM) and tangent space linear discriminant analysis (TS+LDA), have been proposed on the high-dimensional Riemannian manifold by utilizing the concepts of geodesic distance and tangent space [5]. The intrinsic sub-manifold learned by BSML captures the information of high-dimensional Riemannian manifold. Because classification on the low-dimensional sub-manifold can alleviate the overfitting and heavy computation problems, in this paper, we propose two classification algorithms for the sub-manifold based on MDRM and TS+LDA.

The first proposed algorithm is named minimum distance to sub-manifold mean (MDSM) and is based on MDRM [5]. Once training datasets are mapped to the $M_s \times M_s$ intrinsic sub-manifold by the BSML algorithm, the Riemannian mean of each class on the intrinsic sub-manifold can be calculated. For the testing SPD matrices mapped onto the intrinsic sub-manifold, the minimum distance of the testing matrix to all Riemannian means is calculated and the label of the testing matrix can be assigned according the minimum distance. MDSM is presentd in Algorithm 2.

Algorithm 2 Minimum distance to sub-manifold mean (MDSM)

```
Input: Training and testing SPD datasets $\mathbf{P}_{Tr}$,$\mathbf{P}_{Te}$;
Output: Label of testing data;
1: Obtain the optimal dimensionality and intrinsic mapping matrix by BSML, $[M_s, \mathbf{W}_s] = BSML(\mathbf{P}_{Tr})$.
2: Map data onto the intrinsic sub-manifold of size $M_s \times M_s$ as $\mathbf{P}_{cTr} = \mathbf{W}_s \mathbf{P}_{Tr} \mathbf{W}_s^T, \mathbf{P}_{cTe} = \mathbf{W}_s \mathbf{P}_{Te} \mathbf{W}_s^T$;
3: Calculate the Riemannian mean of each class,
   $\mathbf{P}_{cTr1} = \arg\min_{\mathbf{P}} \sum_{\mathbf{P}_i \in C_1} \delta^2{}_R(\mathbf{P}, \mathbf{P}_i)$
   $\mathbf{P}_{cTr2} = \arg\min_{\mathbf{P}} \sum_{\mathbf{P}_i \in C_2} \delta^2{}_R(\mathbf{P}, \mathbf{P}_i)$
   where $C_1$, $C_2$ are the subsets of $\mathbf{P}_{cTr}$ of different classes;
4: Assign a label to the testing data according to the minimum distance to the Riemannian means. For each testing data $\mathbf{P}_j \in \mathbf{P}_{cTe}$, we obtain
   Label=1 if $\delta_R(\mathbf{P}_j, \mathbf{P}_{cTr1}) \leq \delta_R(\mathbf{P}_j, \mathbf{P}_{cTr2})$
   Label=2 if $\delta_R(\mathbf{P}_j, \mathbf{P}_{cTr1}) > \delta_R(\mathbf{P}_j, \mathbf{P}_{cTr2})$;
```

The second proposed algorithm is named tangent space of sub-manifold (TSSM). Suppose $\mathbf{P}_R$ is the Riemannian mean of all data points, including training and testing data points on the intrinsic Riemannian sub-manifold. We could obtain a particular tangent space at the Riemannian mean. Each point on the intrinsic sub-manifold can be projected onto the tangent space. Because the tangent space is a Euclidean vector space, two classical classification algorithms, linear discriminant analysis (LDA) [33] and support vector machine (SVM) [34], are applied for classification. According to the classification used, the two methods are called TSSM+LDA and TSSM+SVM, respectively. The pseudo-code of TSSM is given in Algorithm 3.

Algorithm 3 Tangent space of sub-manifold (TSSM)

```
Input: Training and testing SPD datasets $\mathbf{P}_{Tr}$,$\mathbf{P}_{Te}$;
Output: Label of testing data;
Obtain the optimal dimensionality and intrinsic mapping matrix by BSML, $[M_s, \mathbf{W}_s] = BSML(\mathbf{P}_{Tr})$.
Map data onto the intrinsic sub-manifold of size $M_s \times M_s$ as $\mathbf{P}_{cTr} = \mathbf{W}_s \mathbf{P}_{Tr} \mathbf{W}_s^T, \mathbf{P}_{cTe} = \mathbf{W}_s \mathbf{P}_{Te} \mathbf{W}_s^T$;
Calculate the Riemannian mean of all data points as
    $\mathbf{P}_R = \arg\min_{\mathbf{P}} \sum_{\mathbf{P}_i} \delta^2{}_R(\mathbf{P}, \mathbf{P}_i)$ , $\mathbf{P}i \in \mathbf{P}_{cTr} \bigcup \mathbf{P}_{cTe}$;
Project data onto the tangent space
    $\mathbf{s}_{Tr}$= upper($\mathbf{P}_R^{-\frac{1}{2}} Log(\mathbf{P}_{cTr}) \mathbf{P}_R^{-\frac{1}{2}}) \in \mathbb{R}^{\frac{M_s(M_s+1)}{2}}$
    $\mathbf{s}_{Te}$= upper($\mathbf{P}_R^{-\frac{1}{2}} Log(\mathbf{P}_{cTe}) \mathbf{P}_R^{-\frac{1}{2}}) \in \mathbb{R}^{\frac{M_s(M_s+1)}{2}}$;
Apply the LDA or SVM classifier to the tangent space and for $\mathbf{s}_i \in \mathbf{s}_{Te}$,
    Label=LDA($\mathbf{s}_i$,$\mathbf{s}_{Tr}$) or Label=SVM($\mathbf{s}_i$,$\mathbf{s}_{Tr}$).
```

Table I

COMPARISON OF THE CONFIGURATIONS OF COMPETITION DATA AND IN-HOUSE BCI DATA.

	Dataset IIa of BCI competition IV	In-house dataset
Number of subjects	9	12
Number of channels	22	64
Number of classes	4	2
Trials per class	144	234
Number of training/testing trials	288/288	234/234
Sampling rate	250Hz	250Hz
Filter bank	bandpass 8-30Hz	bandpass 8-30Hz

IV. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of the proposed algorithms on EEG signals from motor imagery BCIs.

A. Experimental Setup

Data description: The n-channel EEG data used in the experiments are BCI competition IV motor imagery data [35] and our in-house experimental data. The major experimental configurations of these two datasets are shown in Table I.

1) Dataset IIa of BCI competition IV was recorded from 9 subjects who performed four types of motor imagery tasks (right hand, left hand, foot and tongue imagined movements). The recorded signals

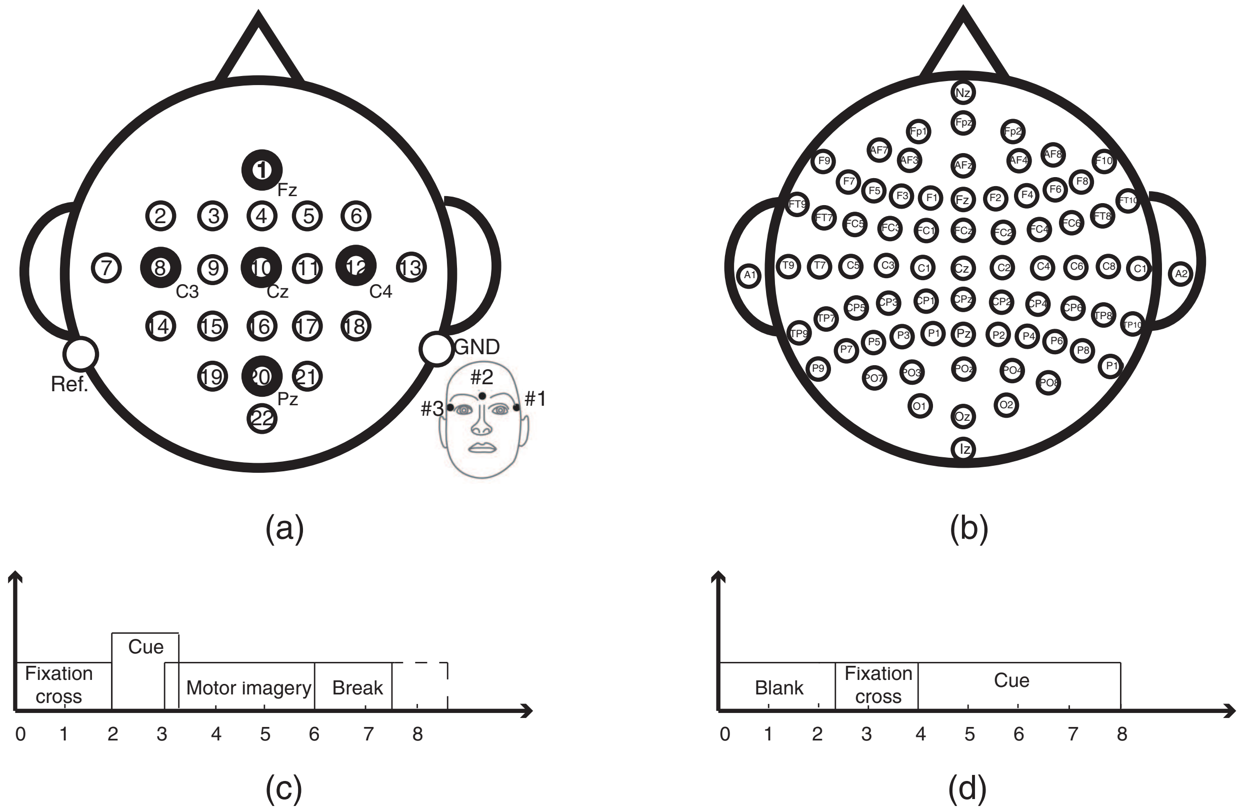
consisted of 22 EEG channels with channel configuration as shown in Fig. 4 (a). The protocol of the experiment was given as follows. In the initial time ($0 - 2s$), a short acoustic warning tone was presented. After two seconds ($2s$), a cue in the form of an arrow pointing left, right, down or up appeared and remained on the screen from $2s$ to $3.25s$. This prompted the subjects to perform the motor imagery task until the fixation cross disappears from the screen at $6s$. Lastly, there was a short break that lasted for 1.5s. The paradigm is illustrated in Fig. 4 (c). The time interval of processed data was restricted to the time segment between $3.75s$ and $5.75s$ during which the subject performed the mental tasks. For each subject and mental task, there were 72 training and 72 testing trials. Thus, the overall number of training/testing trials for each subject was 288/288. The EEG signals were sampled with a sampling rate 250Hz and filtered by a $8-30$Hz bandpass filter to analyze the μ and β rhythms.

2) Our in-house EEG data were recorded from 12 subjects with 64 EEG channels. The configuration of the 64 EEG channels is shown in Fig. 4 (b). The protocol of the in-house experiment was given as follows. Two mental tasks, i.e., left/right hand imaged movements, were required to perform the in-house BCIs. In the initial interval ($0 - 2.25s$), the screen remained blank. A cross appeared on the screen to attract the subject's visual fixation from $2.25s$ to $4s$. From $4s$ to $8s$, a left/right arrow cue was shown and the subject performed the required task. The paradigm is illustrated in Fig. 4 (d). The time interval for the processed data was restricted to the time segment between $5s$ and $7s$. For each subject and each task, there were 117 training and testing trials. The overall number of training/testing trials for each subject was 234/234. The EEG signals were sampled with a sampling rate of 250Hz and filtered by a $8 - 30$Hz bandpass filter.

Algorithms evaluated: We evaluated the proposed algorithms, MDSM, TSSM+LDA and TSSM+SVM against the following four competing algorithms.

1) CSP+LDA: CSP [36] followed by LDA [33] was applied for motor imagery classification.
2) CSP+SVM: CSP [36] followed by SVM [34] was applied for motor imagery classification.
3) MDRM: Minimum distance to Riemannian mean was used for classification on the high-dimensional Riemannian manifold [5].
4) TS+LDA: LDA was applied to high-dimensional tangent space for classification [5].

Parameter setting: The number of selected variables of TS+LDA was set to 10 as suggested in [5]. The number of CSP spatial filter was set to 8 as suggested in [36]. The kernel of SVM was selected as radial basis function (RBF) and the regularization parameter of SVM was set as 0.8 [34]. Electrooculography



(a)

(b)

(c)

(d)

Figure 4. The configuration of EEG sensors. a) BCI competition IV (#1,#2,#3 are the positions of EOG channels.); b) In-house BCI system ; c) timing scheme of the paradigm for dataset II of competition IV; d) timing scheme of the paradigm for in-house dataset.

(EOG) artifact was removed by linear regression method [37] as $\mathbf{Y}(t) = \mathbf{X}(t) - \mathbf{K}\mathbf{U}(t)$, where $\mathbf{X}(t) \in \mathbb{R}^{22\times L}$ is the EOG-contaminated EEG signal, $\mathbf{Y}(t) \in \mathbb{R}^{22\times L}$ is EEG signal with EOG removed, $\mathbf{U}(t) \in \mathbb{R}^{3\times L}$ is the EOG signal and $\mathbf{K} \in \mathbb{R}^{22\times 3}$ is weighting matrix which is estimated by $\mathbf{K} = \mathbf{C}_{XU}\mathbf{C}_{UU}^{-1}$ [38], where $\mathbf{C}_{XU}$ is the cross-covariance matrix of $\mathbf{X}(t)$ and $\mathbf{U}(t)$, and $\mathbf{C}_{UU}$ is the auto-covariance matrix of the EOG signal $\mathbf{U}(t)$.

B. Experimental Results

Three experiments were performed to evaluate the algorithms. Experiment I had a normal setting and Experiment II had a small sample data setting. Experiment III studied the computational load of each algorithm.

1) Results of Experiment I: Because cross-validation can test the model in the training phase and provide insights on how the model will generalize an independent test dataset, in this paper, we first tested

Table II

Comparison of classification accuracies on dataset IIa via 30-fold cross-validation.

Subject	Method						
	TSSM+LDA	TSSM+SVM	MDSM	TS+LDA[5]	MDRM[5]	CSP+LDA[33]	CSP+SVM[34]
S01	81.8	80	82.2	80.5	77.8	78.3	76.3
S02	62.5	58.7	57.4	51.3	44.1	44.7	50.7
S03	88.8	86.3	84.8	87.5	76.8	82.2	85.1
S04	63.7	68.2	62.2	59.3	54.9	59.1	52.9
S05	62.9	60.3	62.5	45	43.8	39.7	48.8
S06	58.5	59.2	58.5	55.3	47.1	50.1	49.2
S07	86.6	84.4	83.7	82.1	72	81	78.1
S08	85.1	84.0	80.3	84.8	75.2	68.5	77.4
S09	90	89.6	85.1	86.1	76.6	77.4	82.2
mean±std	**75.5±13.2**	**74.5±12.7**	**73.0±12.3**	70.2±17.1	63.2±15.2	64.6±16.6	66.7±15.7

the performance of the proposed algorithms on dataset IIa of BCI competition IV and in-house dataset with a 30-fold cross-validation procedure [5]. No parameters need to be set for the proposed methods. The training dataset of competition IV and in-house BCI were randomly divided into 30 subsets of equal size. In each run, 29 subsets were used as training data and a single subset was used as the validation data.

Because dataset IIa of BCI competition IV is a four-class problem, in this paper, the one-versus-one strategy was used to extend the two-class classification algorithms for such a case. A total of 4(4-1)/2=6 binary classifiers and 6 mapping matrices $\mathbf{W}_s$ were learned from the training dataset. The simple majority voting scheme was applied to obtain the final label. Table II presents the accuracies of classification of all studied algorithms. The proposed methods, MDSM, TSSM+LDA and TSSM+SVM not only have higher mean accuracy on classification, but also have lower standard deviations on classification accuracy. These lower standard deviations indicate that the proposed methods are more robust against the variance of subjects than the other methods.

Table III shows the results of 30-fold cross-validation on the in-house dataset. It indicates that the proposed methods have better performance, e.g., higher mean accuracies and lower standard deviations, than the others. Since the standard deviation values are relatively large compared with the difference between mean performances in Table II and III, it is necessary to provide statistical significance analysis on the cross-validation results.

Table III

Comparison of classification accuracies on in-house dataset via 30-fold cross-validation.

Subject	Method						
	TSSM+LDA	TSSM+SVM	MDSM	TS+LDA[5]	MDRM[5]	CSP+LDA[33]	CSP+SVM[34]
A01	100	100	100	98.1	98.1	97.7	100
A02	95.5	93.3	93.3	93.3	94.6	81.6	80.0
A03	100	100	100	100	87.8	95.6	100
A04	94.8	94.8	94.9	86.1	84.6	84.6	90.3
A05	97.7	95.5	95.6	94.0	93.3	85.0	76.6
A06	92.3	94.8	94.8	86.1	81.5	80.7	80.7
A07	90.9	87.8	93.6	89.1	87.2	72.7	84.1
A08	87.8	84.8	93.9	87.2	85.4	81.8	84.1
A09	93.9	93.9	93.9	83.6	81.8	84.1	86.3
A10	84.8	90.9	84.8	83.1	89.1	88.6	88.6
A11	96.9	96.9	100	92.7	92.7	97.7	90.9
A12	84.0	82.6	79.7	75.8	69.5	68.4	80.4
mean±std	**93.3±5.4**	**92.9±5.5**	**93.7±6.0**	89.1±6.8	87.1±7.5	84.9±9.1	86.8±7.5

For the results in Table II and III, the analysis of variance (ANOVA) was first used to show the significant difference of the studied methods. The paired T-tests were then adopted to further prove the significant performance improvement of the proposed methods, compared to the other methods with the same classifier but different feature extractor. The one-way ANOVA results for Table II ($p = 2.0e^{-11} < 0.05$) and Table III ($p = 2.1e^{-7} < 0.05$) indicates that the performances of the studied methods are significantly different. From the paired T-test results shown in Table IV, it is clear that the proposed methods yield significantly higher performance than the competing methods.

To obtain a more sophisticated analysis of the 30-fold cross-validation experiment, we calculated the confusion matrix of all studied algorithms corresponding to the results in Table II. A graphical representation of the confusion matrix is also shown in Fig. 5. For a fair comparison, we used the average of the confusion matrices of all subjects. Compared with TS+LDA and MDRM, MDSM and TSSM have larger diagonal elements and smaller non-diagonal elements of the confusion matrix. This result indicates that the proposed methods greatly improve the classification performance of each class.

The Kappa coefficient is commonly used as a performance measure for the dataset IIa of competition IV [39]. Kappa coefficient scales from 0 to 1 linearly onto the range between random and perfect

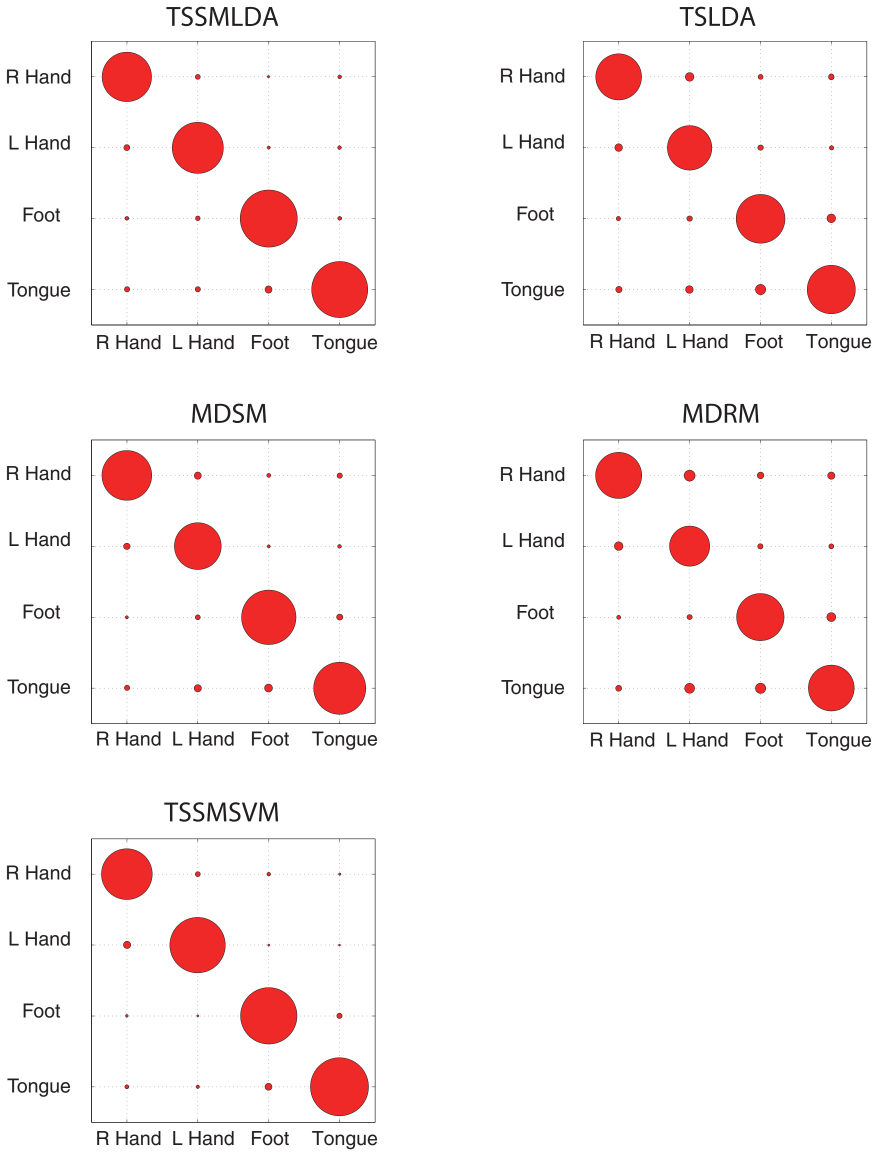


Figure 5. Comparison of the confusion matrices of the studied algorithms on dataset IIa of BCI competition IV. The diameter (size) of the circle indicates the magnitude of the corresponding entry in the confusion matrix.

Table IV

PAIRED T-TEST RESULTS FOR THE PROPOSED METHODS VERSUS COMPETING METHODS BASED ON TABLE II AND TABLE III.

Paired T-test	Dataset IIa (Table II)	In-house BCI (Table III)
	p-value	p-value
TSSM+LDA vs. TS+LDA	* (0.0227)	† (0.0015)
TSSM+LDA vs. CSP+LDA	† (0.0014)	† (0.0013)
TSSM+SVM vs. CSP+SVM	†† (5.1e-4)	** (0.0057)
MDSM v.s. MDRM	†† (1.7e-4)	† (0.0021)

Note: $\sim$ nonsignificant, * $p \leq 0.05$, ** $p \leq 0.01$, † $p \leq 0.005$, †† $p \leq 0.001$

classification. Kappa coefficient [39] is defined as

$$K_{appa} = \frac{(p_o - p_e)}{(1 - p_e)} \tag{20}$$

where p_o is the proportion of observed agreement (equivalent to the average classification accuracy rate over all the classes), p_e is the proportion of chance expected agreement (defined as $p_e = \mathbf{m}_c \times \mathbf{m}_r^T \times \Sigma^2$, where $\mathbf{m}_c$ and $\mathbf{m}_r$ represent row vectors containing elements as the sums of columns and the sums of the rows of the confusion matrix, respectively, and Σ are the sums of all elements in the confusion matrix). In Table V, we also reported the results using the Kappa coefficient as a performance index for dataset IIa of BCI competition IV. The performances of the top three winners of competition IV (1^{st}, 2^{nd} and 3^{rd} of Competition IV) are included in the comparison. In particular, TSSM+LDA achieved a mean value of 0.593 and the TSSM+SVM method achieved a mean value of 0.571. The MDSM method achieved a mean value of 0.568, thus ranking sixth in Table V. Moreover, to show the robustness against artifacts, the performance of the proposed algorithms without EOG removal are also presented in Table V. These results reveal that EOG removal is important for BCI system to improve classification performance. However, the proposed methods demonstrate strong robustness against artifacts. Without EOG removal, they have slight performance degradation but still produce superior performance.

In Table VI, to enrich the performance comparison, we presented the classification accuracies of the studied methods on our in-house BCI dataset. The proposed algorithm, MDSM, exhibits approximately 4% improvement on classification compared to MDRM. Similarly, TSSM+LDA exhibits approximately 2% improvement compared to TS+LDA. From the paired T-test results, e.g., MDSM vs. MDRM ($p = 0.003$), TSSM+LDA vs. TS+LDA ($p = 0.07$), TSSM+SVM vs. CSP+SVM ($p = 0.014$) and TSSM+LDA vs.

Table V

COMPARISON OF THE KAPPA VALUES OF DIFFERENT METHODS ON DATASET IIA OF BCI COMPETITION IV FOR PREDICTION.

Method	Mean Kappa	Subject								
		S01	S02	S03	S04	S05	S06	S07	S08	S09
TSSM+LDA	**0.593**	0.77	0.33	0.77	0.51	0.35	0.36	0.71	0.72	0.83
TSSM+LDA(without EOG removal)	**0.584**	0.76	0.32	0.77	0.50	0.34	0.35	0.70	0.70	0.81
TSSM+SVM	**0.571**	0.70	0.32	0.75	0.54	0.32	0.34	0.70	0.69	0.77
1^{st} of Competition IV	0.570	0.68	0.42	0.75	0.48	0.40	0.27	0.77	0.75	0.61
TS+LDA[5]	0.567	0.74	0.38	0.72	0.50	0.26	0.34	0.69	0.71	0.76
MDSM	**0.568**	0.72	0.34	0.74	0.49	0.34	0.34	0.71	0.70	0.73
TSSM+SVM(without EOG removal)	**0.564**	0.70	0.31	0.74	0.54	0.31	0.34	0.69	0.68	0.75
MDSM(without EOG removal)	**0.562**	0.71	0.33	0.73	0.49	0.34	0.33	0.70	0.70	0.73
MDRM[5]	0.521	0.75	0.37	0.66	0.53	0.29	0.27	0.56	0.58	0.68
2^{nd} of Competition IV	0.520	0.69	0.34	0.71	0.44	0.16	0.21	0.66	0.73	0.69
3^{rd} of Competition IV	0.310	0.38	0.18	0.48	0.33	0.07	0.14	0.29	0.49	0.44

CSP+LDA ($p = 0.015$), it is clear that only the performance improvement of TSSM+LDA vs. TS+LDA is not significant. However, the $p = 0.07$ is very close to 0.05. Based on the results for the competition and in-house datasets, we can conclude that the proposed methods significantly improved classification performance compared to the other algorithms. These improvements might be attributable in part to the ability of the intrinsic sub-manifold learned by BSML to capture the major geometric information of the original manifold and relief of the overfitting problem to some extent by the dimensionality reduction.

We also compared the dimensionality reduction performance of BSML and the state-of-the-art manifold learning algorithms. The 2-dimensional embeddings learned by Isomap, LLE, CSP and BSML are shown in Fig. 6. The distributions of learned features in Fig. 6 clearly indicate that the features learned by BSML have high separability, supporting the possibility of high classification performance for BSML based methods.

The proposed BSML method can be considered as an extension of CSP. The BSML also learned spatial patterns form the experimental covariance matrices. In Fig. 7, the topographic maps of spatial filters learned by the CSP and BSML are shown. We can see that the spatial filters learned by BSML have larger difference between electrodes C_3 and C_4, which cover the area dedicated to the right-hand and left-hand imagery movements. The major difference between BSML and CSP is the Riemannian

Table VI

COMPARISION OF CLASSIFICATION ACCURACIES ON IN-HOUSE BCI DATASET FOR PREDICTION.

Method	Mean accuracy	Subject											
		A01	A02	A03	A04	A05	A06	A07	A08	A09	A10	A11	A12
TSSM+SVM	**90.0**	100	98.7	100	95.1	98.7	87.0	92.9	87.7	83.9	78.5	78.9	78.6
TSSM+LDA	**89.8**	100	100	100	95.1	98.7	87.0	92.9	94.7	83.9	73.2	70.1	82.0
MDSM	**90.3**	100	96.1	97.5	91.9	98.7	100	89.4	85.9	87.5	78.5	80.7	77.7
TS+LDA[5]	88.0	100	97.4	100	88.7	97.4	85.5	91.2	84.2	82.1	71.4	75.4	79.4
MDRM[5]	86.6	100	96.1	95.0	85.4	93.5	85.4	87.7	80.7	82.1	75.0	71.9	76.9
CSP+LDA[33]	82.6	96.4	96.1	100	74.1	96.1	83.8	82.4	68.4	75.0	66.0	73.6	79.4
CSP+SVM[34]	85.2	98.2	93.5	100	77.4	98.7	87.0	89.4	78.9	82.1	67.8	70.1	79.4

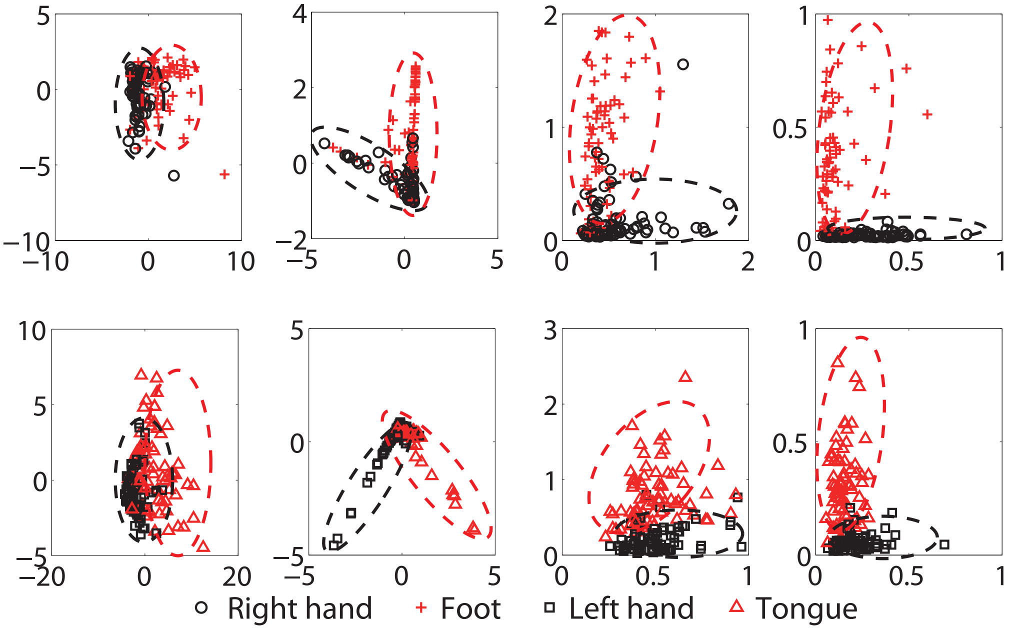


Figure 6. Comparison of 2-dimensional embedding learned from Isomap, LLE, CSP and BSML for subject 1 of dataset IIa of BCI competition IV. The features corresponding to the right hand and foot imaged movements are shown in the top row, and the features corresponding to the left hand and tongue imaged movements are shown in the bottom row.

distance used in BSML. Since Riemannian distance can better represents the relationship of covariance matrices, hence, the BSML is possible to learn a more precise pattern.

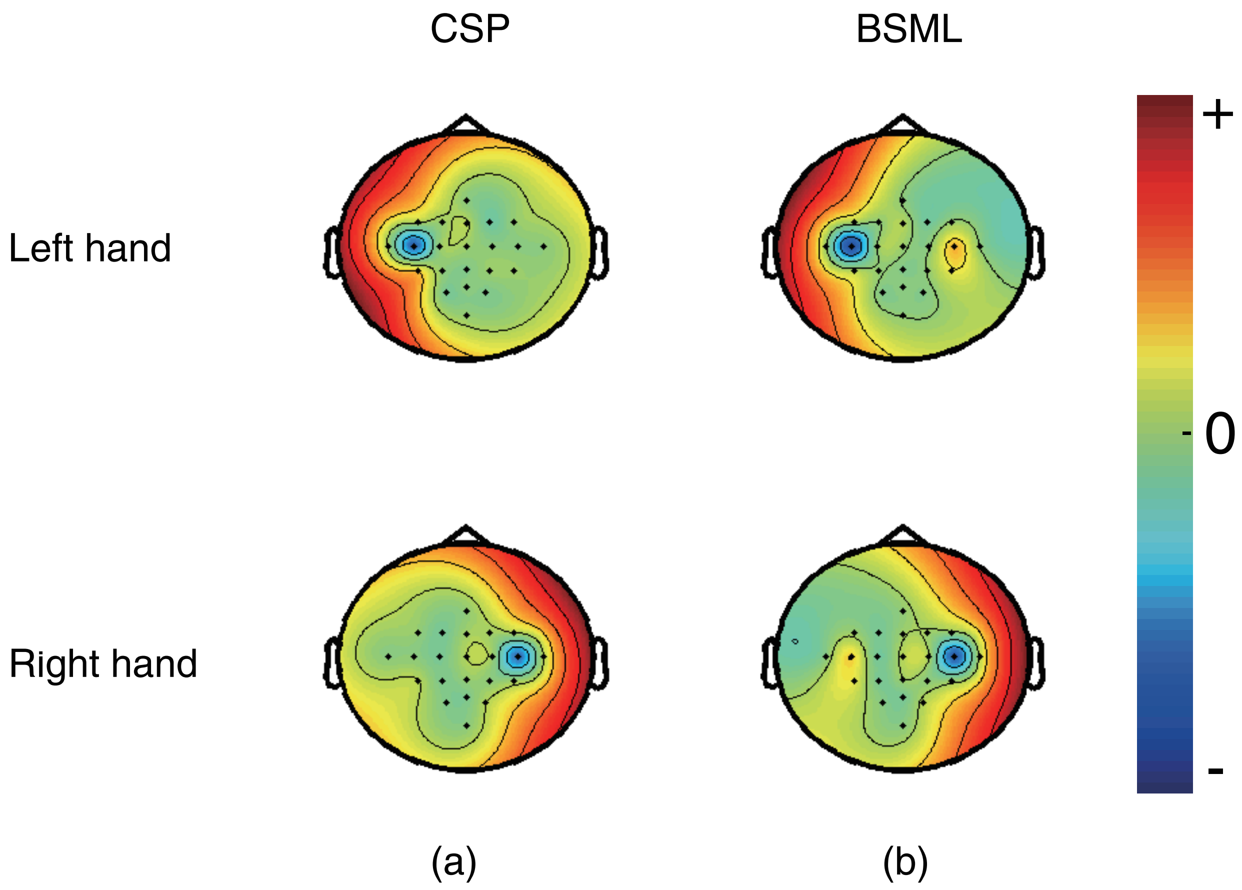


Figure 7. Topographic map of the CSP and BSML methods for the left/right hand motor imagery data from the subject 3 of dataset IIa of BCI competition IV. a) the first and last spatial filters of CSP; b) two row of normalization matrix **W** corresponding to the largest and smallest eigenvalues on $\{\lambda_{j1}, j \in [1, N]\}$.

2) Results of Experiment II: For many reasons, the training sets available in BCI applications are frequently small [40]. Reducing the number of training trials required for a specific task is an important objective in BCI feedback applications. Dimension reduction is a potential means of alleviating the problem of small training dataset. We performed experiments to evaluate the performance of BSML for a small training dataset. In this experiment, we only used 1/2,1/3 and 1/6 (i.e., 144, 96 and 48 trials) training samples of dataset IIa of competition IV. The average of 20 repeated experiments versus different sizes of training datasets are reported in Table VII. As the training sample size decreased from 1/2 to 1/6, the performances of all algorithms decreased. However, the proposed methods exhibited smaller performance

Table VII

COMPARISON OF CLASSIFICATION PERFORMANCE ON DATASET IIA WITH TRAINING DATA OF DIFFERENT SIZES FOR PREDICTION.

Number of sample	Method	Mean Kappa	Subject								
			S01	S02	S03	S04	S05	S06	S07	S08	S09
1/2 training sample (144 trials)	TSSM+LDA	**0.534**	0.71	0.28	0.75	0.40	0.30	0.30	0.67	0.61	0.79
	TSSM+SVM	**0.540**	0.72	0.32	0.75	0.42	0.26	0.26	0.73	0.61	0.74
	MDSM	**0.538**	0.74	0.28	0.72	0.40	0.32	0.31	0.68	0.65	0.75
	TS+LDA[5]	0.512	0.71	0.22	0.72	0.36	0.30	0.34	0.57	0.61	0.78
	MDRM[5]	0.490	0.74	0.27	0.63	0.37	0.28	0.30	0.56	0.58	0.68
1/3 training sample (96 trials)	TSSM+LDA	**0.513**	0.65	0.31	0.74	0.42	0.16	0.24	0.64	0.66	0.76
	TSSM+SVM	**0.515**	0.63	0.30	0.77	0.40	0.24	0.29	0.66	0.59	0.72
	MDSM	**0.519**	0.71	0.35	0.72	0.49	0.20	0.22	0.65	0.62	0.67
	TS+LDA[5]	0.487	0.66	0.31	0.75	0.38	0.19	0.19	0.55	0.61	0.68
	MDRM[5]	0.478	0.68	0.31	0.63	0.50	0.18	0.22	0.53	0.58	0.62
1/6 training sample (48 trials)	TSSM+LDA	**0.499**	0.62	0.29	0.78	0.34	0.21	0.23	0.62	0.55	0.80
	TSSM+SVM	**0.494**	0.63	0.27	0.75	0.37	0.21	0.23	0.63	0.55	0.76
	MDSM	**0.493**	0.66	0.29	0.73	0.42	0.24	0.22	0.61	0.54	0.69
	TS+LDA[5]	0.416	0.56	0.17	0.72	0.34	0.20	0.10	0.43	0.46	0.73
	MDRM[5]	0.410	0.58	0.21	0.64	0.40	0.12	0.15	0.44	0.49	0.64

degradation compared to the other methods. The paired T-tests of TSSM+LDA vs. TS+LDA ($p = 0.003$) and MDSM vs. MDRM ($p = 0.0005$) indicates that the performance improvement of the proposed methods in small sample sizes (1/6 size setting) is significant.

In Table VIII, the results for the in-house BCI dataset with only 1/6 of the training samples are shown. In each experiment, 39 trials randomly selected from 234 training trials were used as the training dataset. The reported results were the average of 20 repeated experiments. Compared with the results of the full training sample case in Table VI, MDSM exhibited 13.3% accuracy degradation, whereas MDRM exhibited 19% accuracy degradation. Similarly, TSSM+LDA exhibited approximately smaller performance degradation, 11.3% , compared to TS+LDA (15.9%). Compared to the other methods, the proposed methods also had smaller performance degradation. Thus, the proposed BSML-based classification methods are more robust for the small training dataset problem. Similar as the four paired comparison in Table IV, all the p-values of paired T-tests are smaller than 0.05, which reveals that the proposed methods have significantly high performance in small sample sizes.

Table VIII

Comparison of classification accuracies on in-house BCI dataset with 1/6 of the training dataset for prediction.

Method	Mean accuracy	Subject											
		A01	A02	A03	A04	A05	A06	A07	A08	A09	A10	A11	A12
TSSM+SVM	**76.5**	89.4	84.4	100	83.8	79.2	93.5	73.6	70.1	69.6	62.5	59.6	52.1
TSSM+LDA	**78.5**	94.2	80.1	100	97.4	89.2	53.5	88.9	64.3	72.5	65.3	73.1	63.8
MDSM	**77.0**	100	87.0	100	93.5	77.9	96.7	64.9	56.1	60.7	53.5	80.7	52.9
TS+LDA[5]	62.1	79.4	77.0	87.5	75.4	64.0	65.8	40.8	46.1	70.3	38.2	56.6	44.7
MDRM[5]	67.6	92.9	81.8	85.0	50.0	55.8	95.1	56.1	56.1	57.1	60.7	57.8	62.3
CSP+LDA[33]	63.3	77.1	48.0	95.0	53.2	61.0	64.5	75.4	57.8	55.3	57.1	50.8	64.1
CSP+SVM[34]	65.7	92.9	63.6	97.5	58.0	61.0	51.6	70.1	49.1	55.3	67.8	59.6	62.3

3) Results for Experiment III: Finally, to demonstrate the efficiency of the proposed methods, we compared the computational loads of the algorithms on the intrinsic sub-manifold with those of methods on the high-dimensional original manifold. The computational loads contain training and testing times. As shown in Fig. 8-9, the training times of MDSM and TSSM were shorter than those of MDRM and TS+LDA on both BCI competition IV and the in-house datasets because calculation of the Riemannian mean requires more time for high-dimensional manifolds. The testing times of MDSM and TSSM were nearly 5 times shorter than those of MDRM and TS+LDA.

V. Conclusions

Dimensionality reduction methods for high-dimensional Riemannian manifold are important to address the over-fitting problem of classification. Most proposed dimension reduction methods are derived for general manifolds, and few exploit the structural information of the manifold from which the original data are sampled. Considering applications with SPD matrices, a BSML method was proposed to identify a linear sub-manifold of SPD data by maximizing the preservation of the Riemannian distance between data points. Three classification algorithms, i.e., MDSM, TSSM+LDA and TSSM+SVM, were proposed for the extracted intrinsic sub-manifold. Experimental results for EEG signals demonstrated that the proposed BSML can capture useful geometric information of the original manifold. The performance of the proposed MDSM and TSSM methods is superior to those of MDRM and TS+LDA, particularly when the training dataset is small. The proposed methods can also be applied to many other pattern recognitions with input data in the form of SPD matrices. Our future work will focus on constructing

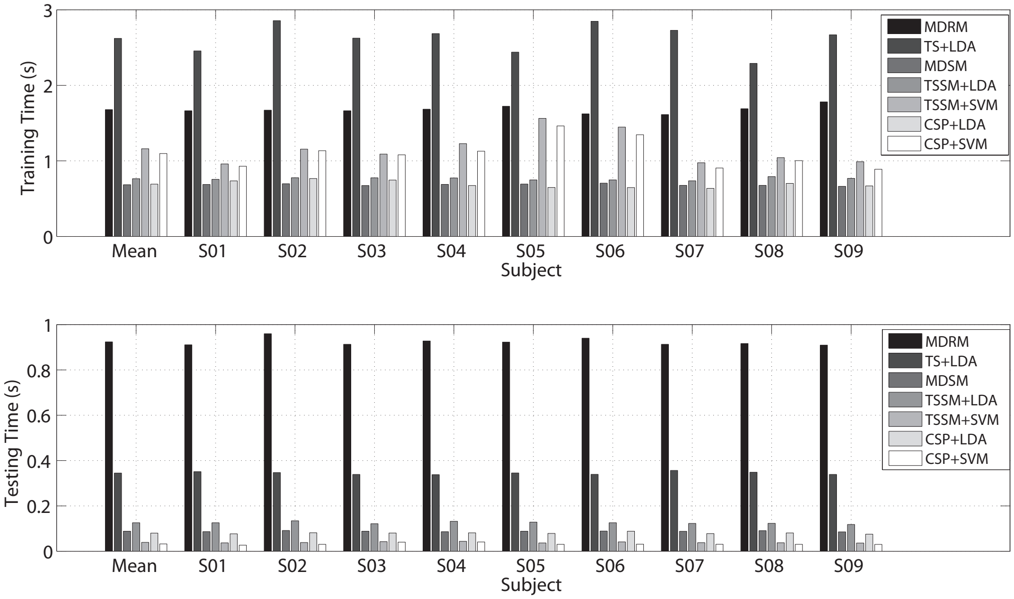


Figure 8. Comparison of the training and testing times of the studied algorithms on dataset IIa of BCI competition IV.

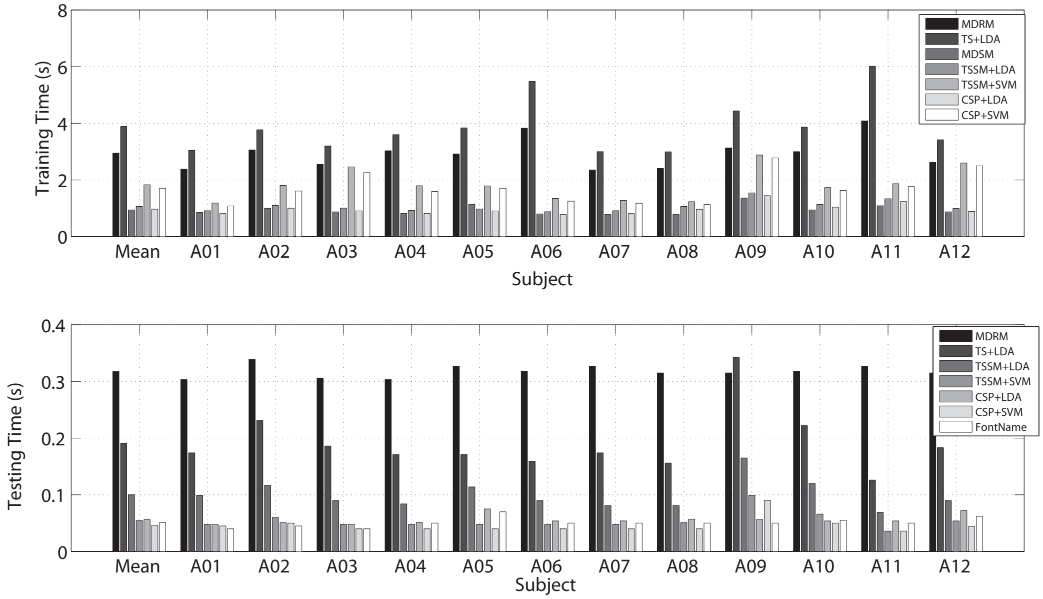


Figure 9. Comparison of the training and testing time of the studied algorithms on the in-house dataset.

nonlinear mapping, such as isometric mapping, instead of the bilinear mapping used in BSML.

Appendix

I. Approximation of (14)

For the two-class classification problem, the cost function of (14) is expressed as

$$\begin{aligned}
&\sum_{\mathbf{P}_i\in C_1,\mathbf{P}_j\in C_2}\left|\delta_R(\mathbf{P}_i,\mathbf{P}_j)-\delta_R(\mathbf{W}_s\mathbf{P}_i{\mathbf{W}_s}^T,\mathbf{W}_s\mathbf{P}_j{\mathbf{W}_s}^T)\right|\\
&+\sum_{\mathbf{P}_j\in C_1,\mathbf{P}_i\in C_2}\left|\delta_R(\mathbf{P}_i,\mathbf{P}_j)-\delta_R(\mathbf{W}_s\mathbf{P}_i{\mathbf{W}_s}^T,\mathbf{W}_s\mathbf{P}_j{\mathbf{W}_s}^T)\right|\\
&+\sum_{\mathbf{P}_i,\mathbf{P}_j\in C_1}\left|\delta_R(\mathbf{P}_i,\mathbf{P}_j)-\delta_R(\mathbf{W}_s\mathbf{P}_i{\mathbf{W}_s}^T,\mathbf{W}_s\mathbf{P}_j{\mathbf{W}_s}^T)\right|\\
&+\sum_{\mathbf{P}_i,\mathbf{P}_j\in C_2}\left|\delta_R(\mathbf{P}_i,\mathbf{P}_j)-\delta_R(\mathbf{W}_s\mathbf{P}_i{\mathbf{W}_s}^T,\mathbf{W}_s\mathbf{P}_j{\mathbf{W}_s}^T)\right|
\end{aligned}\tag{21}$$

where the first two items measure the mapping error among between-class samples, and the last two items measure the mapping error among within-class samples. According to the invariance of linear transformation (7), we have $\delta_R(\mathbf{P}_i,\mathbf{P}_j)\geq\delta_R(\mathbf{W}_s\mathbf{P}_i{\mathbf{W}_s}^T,\mathbf{W}_s\mathbf{P}_j{\mathbf{W}_s}^T)$ for all $\mathbf{W}_s$. Considering the classification problem, if we can guarantee the between-class distance, then the compression of within-class variance can be anticipated. Thus, the last two items can be ignored in optimization and the optimization problem (14) can be approximated as

$$\min_{\mathbf{W}_s}\sum_{\mathbf{P}_i\in C_1,\mathbf{P}_j\in C_2}\left|\delta_R(\mathbf{P}_i,\mathbf{P}_j)-\delta_R(\mathbf{W}_s\mathbf{P}_i{\mathbf{W}_s}^T,\mathbf{W}_s\mathbf{P}_j{\mathbf{W}_s}^T)\right|\tag{22}$$

As shown in Fig. 10, we regard $\mathbf{P}_a$ as the Riemannian mean of C_1 and $\mathbf{P}_b$ as Riemannian mean of C_2. Applying the triangular inequalities

$$\delta_R(\mathbf{P}_i,\mathbf{P}_b)-\delta_R(\mathbf{P}_j,\mathbf{P}_b)\leq\delta_R(\mathbf{P}_i,\mathbf{P}_j)\leq\delta_R(\mathbf{P}_i,\mathbf{P}_b)+\delta_R(\mathbf{P}_j,\mathbf{P}_b)\tag{23}$$

we have

$$\begin{aligned}
&\sum_{\mathbf{P}_i\in C_1,\mathbf{P}_j\in C_2}\delta_R(\mathbf{P}_i,\mathbf{P}_j),\\
&\leq|C_2|\sum_{\mathbf{P}_i\in C_1}\delta_R(\mathbf{P}_i,\mathbf{P}_b)+|C_1|\sum_{\mathbf{P}_j\in C_2}\delta_R(\mathbf{P}_j,\mathbf{P}_b),\\
&\leq|C_1||C_2|\delta_R(\mathbf{P}_a,\mathbf{P}_b)+|C_2|\sum_{\mathbf{P}_i\in C_1}\delta_R(\mathbf{P}_i,\mathbf{P}_a)+|C_1|\sum_{\mathbf{P}_j\in C_2}\delta_R(\mathbf{P}_j,\mathbf{P}_b)
\end{aligned}\tag{24}$$

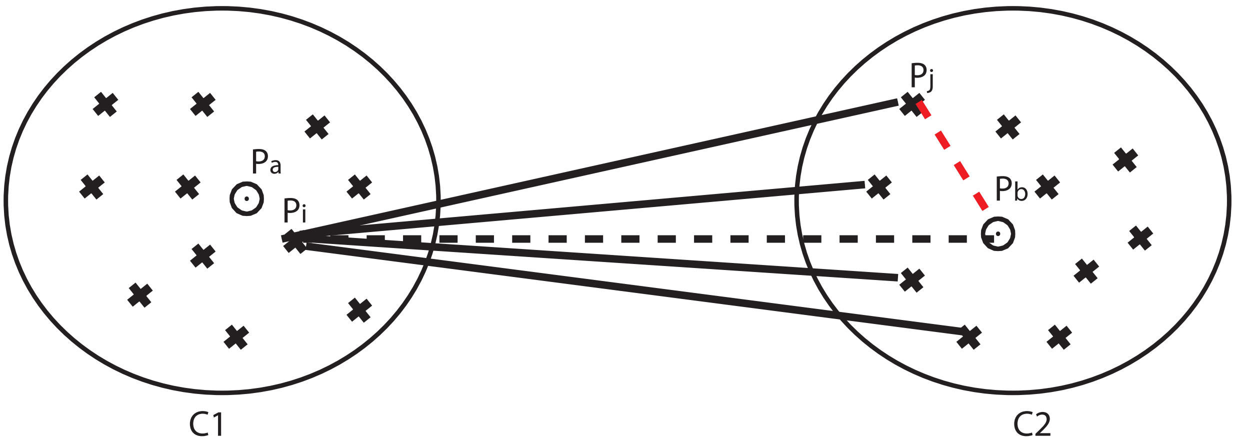


Figure 10. Approximation of between-class distance.

where $|C|$ is the cardinality of set C and

$$
\begin{aligned}
&\sum_{\mathbf{P}_i\in C_1,\mathbf{P}_j\in C_2} \delta_R(\mathbf{P}_i,\mathbf{P}_j),\\
&\geq |C_2|\sum_{\mathbf{P}_i\in C_1}\delta_R(\mathbf{P}_i,\mathbf{P}_b) - |C_1|\sum_{\mathbf{P}_j\in C_2}\delta_R(\mathbf{P}_j,\mathbf{P}_b),\\
&\geq |C_1||C_2|\delta_R(\mathbf{P}_a,\mathbf{P}_b) - |C_2|\sum_{\mathbf{P}_i\in C_1}\delta_R(\mathbf{P}_i,\mathbf{P}_a) - |C_1|\sum_{\mathbf{P}_j\in C_2}\delta_R(\mathbf{P}_j,\mathbf{P}_b)
\end{aligned} \tag{25}
$$

In summary,

$$
\begin{aligned}
&\left|\frac{1}{|C_1||C_2|}\sum_{\mathbf{P}_i\in C_1,\mathbf{P}_j\in C_2}\delta_R(\mathbf{P}_i,\mathbf{P}_j) - \delta_R(\mathbf{P}_a,\mathbf{P}_b)\right|,\\
&\leq \frac{1}{|C_1|}\sum_{\mathbf{P}_i\in C_1}\delta_R(\mathbf{P}_i,\mathbf{P}_a) + \frac{1}{|C_2|}\sum_{\mathbf{P}_j\in C_2}\delta_R(\mathbf{P}_j,\mathbf{P}_b).
\end{aligned} \tag{26}
$$

If we select $\mathbf{P}_1 = \arg\min_{\mathbf{P}_b}\sum_{\mathbf{P}_j\in C_2}\delta_R^2(\mathbf{P}_j,\mathbf{P}_b)$ and $\mathbf{P}_2 = \arg\min_{\mathbf{P}_a}\sum_{\mathbf{P}_i\in C_1}\delta_R^2(\mathbf{P}_i,\mathbf{P}_a)$, in other words, the Riemannian means of datasets, the between-class distance can be approximated as the distance between the means of two datasets, particularly when the within-class variance is much smaller compared with the between-class distance. Thus, for easy processing, we approximate the optimization problem (14) as

$$
\min_{\mathbf{W}_s}\left|\delta_R(\mathbf{P}_a,\mathbf{P}_b) - \delta_R(\mathbf{W}_s\mathbf{P}_a{\mathbf{W}_s}^T,\mathbf{W}_s\mathbf{P}_b{\mathbf{W}_s}^T)\right|. \tag{27}
$$

II. Solution of Joint Diagonalization

Suppose $\mathbf{P}_1$ and $\mathbf{P}_2$ are two symmetric positive-definite matrices. Essentially, the normalization of $\mathbf{P}_1$ and $\mathbf{P}_2$ is a problem of joint diagonalization which has been studied extensively. The joint diagonalization problem can be resolved by the following procedures:

1) Obtain a whitening matrix by Eigenvalue decomposition of $\mathbf{P}_1 + \mathbf{P}_2$

$$(\mathbf{P}_1 + \mathbf{P}_2) = \mathbf{U}\Sigma\mathbf{U}^T. \tag{28}$$

The whitening matrix is given as $\widetilde{\mathbf{U}} = \Sigma^{-\frac{1}{2}}\mathbf{U}^T$. Then, we have

$$\widetilde{\mathbf{U}}\mathbf{P}_1\widetilde{\mathbf{U}}^T + \widetilde{\mathbf{U}}\mathbf{P}_2\widetilde{\mathbf{U}}^T = \mathbf{I}. \tag{29}$$

2) Diagonalize of $\widetilde{\mathbf{U}}\mathbf{P}_1\widetilde{\mathbf{U}}^T$. Because it cannot be ensured that $\widetilde{\mathbf{U}}\mathbf{P}_1\widetilde{\mathbf{U}}^T$ is a diagonal matrix, we can find a diagonalization matrix $\mathbf{U}_1$ for $\widetilde{\mathbf{U}}\mathbf{P}_1\widetilde{\mathbf{U}}^T$ by applying eigenvalue decomposition as

$$\widetilde{\mathbf{U}}\mathbf{P}_1\widetilde{\mathbf{U}}^T = \mathbf{U}_1\Sigma_1{\mathbf{U}_1}^T. \tag{30}$$

3) Construct the transformation matrix as $\mathbf{W} = {\mathbf{U}_1}^T\widetilde{\mathbf{U}}$.

It is easy to prove that the transformation matrix $\mathbf{W}$ is subject to $\mathbf{W}\mathbf{P}_1\mathbf{W}^T + \mathbf{W}\mathbf{P}_2\mathbf{W}^T = \mathbf{I}$ and that both $\mathbf{W}\mathbf{P}_1\mathbf{W}^T$ and $\mathbf{W}\mathbf{P}_2\mathbf{W}^T$ are diagonal matrices.

An alternative method to obtain transformation matrix $\mathbf{W}$ is to apply eigenvalue decomposition to $(\mathbf{P}_1 + \mathbf{P}_2)^{-1}\mathbf{P}_1$ as $(\mathbf{P}_1 + \mathbf{P}_2)^{-1}\mathbf{P}_1 = \mathbf{W}\Sigma_1\mathbf{W}^T$. The obtained transformation matrix can be proven to be identical to the above method as follows:

$$\mathbf{P}_1\widetilde{\mathbf{U}}^T\mathbf{U}_1 = \widetilde{\mathbf{U}}^{-1}\mathbf{U}_1\Sigma_1 \tag{31}$$

$$(\mathbf{P}_1 + \mathbf{P}_2)^{-1} = \widetilde{\mathbf{U}}^T\widetilde{\mathbf{U}} \tag{32}$$

$$\widetilde{\mathbf{U}}^{-1}=(\mathbf{P}_1 + \mathbf{P}_2)\widetilde{\mathbf{U}^T} \tag{33}$$

$$(\mathbf{P}_1 + \mathbf{P}_2)^{-1}\mathbf{P}_1 = {\mathbf{U}_1}^T\widetilde{\mathbf{U}}\Sigma_1\mathbf{U}_1\widetilde{\mathbf{U}}^T = \mathbf{W}\Sigma_1\mathbf{W}^T. \tag{34}$$

ACKNOWLEDGMENTS

The authors acknowledge all of the anonymous reviewers for their constructive comments that helped to improve the quality of this paper. The authors also like to thank the authors of [5], Alexandre Barachant, for providing the Covariance Toolbox .

References

[1] B. Blankertz, M. Tangermann, C. Vidaurre, S. Fazli, C. Sannelli, S. Haufe, C. Maeder, L. Ramsey, I. Sturm, G. Curio, and K.-R. Müller, "The Berlin brain-computer interface: non-medical uses of BCI technology," *Front. Neurosci.*, vol. 4, no. 198, pp. 1–17, Dec. 2010.

[2] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K.-R. Müller, "Optimizing spatial filters for robust EEG single-trial analysis," *IEEE Signal Process. Mag.*, vol. 25, no. 1, pp. 41–56, Jan. 2008.

[3] W. Förstner and B. Moonen, "A metric for covariance matrices," in *Geodesy-The Challenge of the 3rd Millennium*, E. W. Grafarend, F. W. Krumm, and V. S. Schwarze, Eds. Berlin:Springer-Verlag, 2003, pp. 299–309.

[4] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Riemannian geometry applied to BCI classification," in *Latent Variable Analysis and Signal Separation*, V. Vigneron, V. Zarzoso, E. Moreau, R. Gribonval, and E. Vincent, Eds. Berlin:Springer-Verlag, 2010, pp. 629–636.

[5] ——, "Multiclass brain–computer interface classification by Riemannian geometry," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 4, pp. 920–928, Apr. 2012.

[6] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian detection via classification on Riemannian manifolds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1713–1727, Oct. 2008.

[7] Q. Jackson and D. Landgrebe, "An adaptive classifier design for high-dimensional data analysis with a limited training data set," *IEEE Trans. Geosci. Remote. Sens.*, vol. 39, no. 12, pp. 2664–2679, Dec. 2001.

[8] D. Freedman, "Efficient simplicial reconstructions of manifolds from their samples," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 10, pp. 1349–1357, Oct. 2002.

[9] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.

[10] S. Lafon and A. B. Lee, "Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1393–1403, Sep. 2006.

[11] T. Lin and H. Zha, "Riemannian manifold learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 5, pp. 796–809, May. 2008.

[12] I. Borg and P. J. Groenen, *Modern multidimensional scaling: theory and applications*. New York: Springer-Verlag, 2005, ch. 7, pp. 137–160.

[13] M. H. C. Law and A. K. Jain, "Incremental nonlinear dimensionality reduction by manifold learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 3, pp. 377–391, Mar. 2006.

[14] V. D. Silva and J. B. Tenenbaum, "Global versus local methods in nonlinear dimensionality reduction," in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds. MIT Press, 2003, pp. 705–712.

[15] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.

[16] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using Laplacianfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 328–340, Mar. 2005.

[17] D. L. Donoho and C. Grimes, "Hessian eigenmaps: locally linear embedding techniques for high-dimensional data," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 100, no. 10, pp. 5591–5596, Mar. 2003.

[18] M. Brand, "Charting a manifold," in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds. MIT Press, 2003, pp. 961–968.

[19] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimension reduction via local tangent space alignment," *SIAM J. Sci. Comput.*, vol. 26, no. 1, pp. 313–338, Jan. 2005.

[20] Z. Zhang, J. Wang, and H. Zha, "Adaptive manifold learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 2, pp. 253–265, Feb. 2012.

[21] H. Higashi and T. Tanaka, "Simultaneous design of FIR filter banks and spatial patterns for EEG signal classification," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 4, pp. 1100–1110, Apr. 2013.

[22] A. S. Aghaei, M. S. Mahanta, and K. N. Plataniotis, "Separable common spatio-spectral patterns for motor imagery BCI systems," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 1, pp. 15–29, Jan. 2016.

[23] K. Yu, K. Shen, S. Shao, W. Ng, and X. Li., "Bilinear common spatial pattern for single-trial ERP-based rapid serial visual presentation triage," *J Neural Eng*, vol. 9, no. 4, p. 046013, Aug. 2012.

[24] Y. Zhang, G. Zhou, Q. Zhao, J. Jin, X. Wang, and A. Cichocki, “Spatial-temporal discriminant analysis for ERP-based brain-computer interface,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 21, no. 2, pp. 233–243, Mar. 2013.

[25] A. K. Gupta and D. K. Nagar, *Matrix variate distributions*. CRC Press, 1999, ch. 2, pp. 55–82.

[26] S. T. Smith, “Covariance, subspace, and intrinsic Cramér-rao bounds,” *IEEE Trans. Signal Process.*, vol. 53, no. 5, pp. 1610–1630, May. 2005.

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

[28] J. M. Lee, *Introduction to smooth manifolds*. New York: Springer-Verlag, 2012, ch. 5, pp. 93–105.

[29] H. Karcher, “Riemannian center of mass and mollifier smoothing,” *Commun. Pur. Appl. Math.*, vol. 30, no. 5, pp. 509–541, Sep. 1977.

[30] P. T. Fletcher and S. Joshi, “Principal geodesic analysis on symmetric spaces: statistics of diffusion tensors,” in *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis*, M. Sonka, I. A. Kakadiaris, and J. Kybic, Eds. Berlin:Springer-Verlag, 2004, pp. 87–98.

[31] M. T. Harandi, M. Salzmann, and R. I. Hartley, “From manifold to manifold: Geometry-aware dimensionality reduction for SPD matrices,” in *13th European Conf. Comput. Vision (ECCV)*, Zurich, Sep. 2014, pp. 17–32.

[32] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, “Common spatial pattern revisited by Riemannian geometry,” in *Proc. 2010 IEEE Int. Workshop Multimedia Signal Process.(MMSP)*, Saint Malo, Otc. 2010, pp. 472–476.

[33] A. M. Martinez and A. C. Kak, “PCA versus LDA,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 228–233, Feb. 2001.

[34] H. Sun, Y. Xiang, Y. Sun, H. Zhu, and J. Zeng, “On-line EEG classification for brain-computer interface based on CSP and SVM,” in *3rd Int. Congr. Image and Signal Process. (CISP)*, Yantai, Oct. 2010, pp. 4105–4108.

[35] C. Brunner, R. Leeb, G. Müller-Putz, A. Schlögl, and G. Pfurtscheller, “BCI competition 2008 - graz data set a,” graz University of Technology, Austria.

[36] M. Grosse-Wentrup and M. Buss, “Multiclass common spatial patterns and information theoretic feature extraction,” *IEEE Trans. Biomed. Eng.*, vol. 55, no. 8, pp. 1991–2000, Aug. 2008.

[37] M. Fatourechi, A. Bashashati, R. K. Ward, and G. E. Birch, “EMG and EOG artifacts in brain computer interface systems: A survey,” *Clin. Neurophysiol.*, vol. 118, no. 3, pp. 480–494, Mar. 2007.

[38] A. Schlögl, C. Keinrath, D. Zimmermann, R. Scherer, R. Leeb, and G. Pfurtscheller, “A fully automated correction method of EOG artifacts in EEG recordings,” *Clin. Neurophysiol.*, vol. 118, no. 1, pp. 98–104, Jan. 2007.

[39] G. Townsend, B. Graimann, and G. Pfurtscheller, “A comparison of common spatial patterns with complex band power features in a four-class BCI experiment,” *IEEE Trans. Biomed. Eng.*, vol. 53, no. 4, pp. 642–651, Apr. 2006.

[40] H. Lu, H. L. Eng, C. Guan, K. Plataniotis, and A. Venetsanopoulos, “Regularized common spatial pattern with aggregation for EEG classification in small-sample setting,” *IEEE Trans. Biomed. Eng.*, vol. 57, no. 12, pp. 2936–2946, Dec. 2010.