Contents lists available at ScienceDirect

Neurocomputing

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

Q2 Kernel learning over the manifold of symmetric positive definite matrices for dimensionality reduction in a BCI application

Q1 Khadijeh Sadatnejad, Saeed Shiry Ghidary*

Computer Engineering and Information Technology Department, Amirkabir University of Technology, Hafez Ave., Tehran, Iran

ARTICLE INFO

Article history:
Received 19 April 2015
Received in revised form
26 September 2015
Accepted 22 November 2015
Communicated by W.Wu

Keywords:
Brain computer interface
Nonlinear dimensionality reduction
Kernel learning
Riemannian geometry.

ABSTRACT

In this paper, we propose a kernel for nonlinear dimensionality reduction over the manifold of Symmetric Positive Definite (SPD) matrices in a Motor Imagery (MI)-based Brain Computer Interface (BCI) application. The proposed kernel, which is based on Riemannian geometry, tries to preserve the topology of data points in the feature space. Topology preservation is the main challenge in nonlinear dimensionality reduction (NLDR). Our main idea is to decrease the non-Euclidean characteristics of the manifold by modifying the volume elements. We apply a conformal transform over data-dependent isometric mapping to reduce the negative eigen fraction to learn a data dependent kernel over the Riemannian manifolds. Multiple experiments were carried out using the proposed kernel for a dimensionality reduction of SPD matrices that describe the EEG signals of dataset IIa from BCI competition IV. The experiments show that this kernel adapts to the input data and leads to promising results in comparison with the most popular manifold learning methods and the Common Spatial Pattern (CSP) technique as a reference algorithm in BCI competitions. The proposed kernel is strong, particularly in the cases where data points have a complex and nonlinear separable distribution.

1. Introduction

In brain–computer Interface systems that use motor imagery, brain activity is usually captured in the form of EEG signals and is transferred to an external device [27]. Extracting information from EEG signals is carried out by using different pattern recognition methods involving feature extraction, dimensionality reduction, and classification [32,49,51] to ultimately determine the user's mental state [28,36].

Several techniques are available for extracting features from EEG signals [25,4,7,8]. A common spatial pattern algorithm [38,7] and a spatial covariance matrix of a signal [4,5,9] are two major approaches to represent EEG signals in BCI applications. CSP can be considered to be a dimensionality reduction technique that learns spatial filters that maximize class separability. A spatial covariance matrix of the EEG signal, which lies in the space of symmetric positive definite matrices, can be formulated as a connected Riemannian manifold [2]. In recent years, methods using a spatial covariance matrix have attracted considerable attention [10,4,5,9].

In BCI application, samples are usually represented by large feature vectors. Therefore, these problems suffer from the curse of dimensionality [28]. Different research efforts have attempted to overcome the problem of the curse of dimensionality in the BCI literature. Zhang et al. [51] introduced Spatial-Temporal Discriminant Analysis (STDA) as a multiway extension of Linear Discriminant Analysis (LDA). They attempted to maximize the discrimination between two classes by finding two projections from the spatial and temporal information [52]. These projections reduce the dimensionality of the features that feed into the discriminant analysis. To overcome the problems of the curse of dimensionality and the bias-variance tradeoff for Event-Related Potential (ERP) classification in BCI applications, Zhang et al. [50] introduced Aggregation of Sparse Linear Discriminant Analysis (ASLDA). They introduced a sparse LDA to reduce the dimensionality. For this purpose, sparse discriminant vectors were learned by solving a l1-regularized Least Squares Regression (LSR). Sparse CSP that uses a linear combination of a subset of channels was introduced by Goksu et al. [20]. They proposed a generalized eigenvalue decomposition based on a greedy search to identify multiple sparse eigenvectors to compute spatial projections. They showed the effectiveness of the sparse CSP in comparison with the traditional CSP by examining the datasets in the BCI competition (2005). Wu et al. [46] used a statistical framework to provide a spatio-temporal representation of the EEG trials. They modeled the variance of source signals as random variables and proposed a

* Corresponding author. Tel.: +98 2164542737.
E-mail addresses: sadatnejad@aut.ac.ir (K. Sadatnejad), shiry@aut.ac.ir (S. Shiry Ghidary).

http://dx.doi.org/10.1016/j.neucom.2015.11.065
0925-2312/© 2015 Elsevier B.V. All rights reserved.

hierarchical Bayesian model for retrieving the inter-trial variability of amplitude in a sparse way to provide a reduced representation of data [46].

In the case of representing EEG signals by spatial covariance matrices, although this representation reduces the length of the descriptors in comparison with the raw EEG, this reduction is not sufficient to overcome the curse of dimensionality. Dimensionality reduction over the space of SPD matrices by considering the Riemannian geometry of the SPD matrices has difficulties in comparison with treating the points as Euclidean objects [4]. Formulating covariance matrices as a connected Riemannian manifold [4] leads to a nonlinear relationship between observations and latent variables. Therefore, NLDR techniques are required to reduce the dimensionality over this manifold. Several techniques are adapted to the cases where the relationships between observations and latent variables are nonlinear [24]. Popular NLDR techniques, such as locally linear embedding (LLE) [39], local tangent space alignment (LTSA) [48], Laplacian Eigenmap (LE) [6], and Isomap [41], have been applied to the manifolds. However, these techniques all have shortcomings on the manifold of SPD matrices. These shortcomings stem from ignoring the geometrical structure of the manifold (i.e., living the manifold in the non-Euclidean space and performing computations by assuming that the data points are embedded in Euclidean space) [17].

In this paper, we attempt to overcome the curse of dimensionality in the SPD matrix space in BCI applications by learning a kernel that is adapted to the manifold by considering the Riemannian geometry of the manifolds. The main contribution of this paper is learning a kernel by minimizing a measure that shows the non-Euclidean characteristics of the manifold by changing the volume elements, while preserving the geometry, of the input space. This minimization is especially useful in the cases where the data points lie on a manifold with a nonzero intrinsic curvature. The proposed kernel, when applied in multi-dimensional scaling [21], provides an NLDR technique that is well adapted to the manifold of SPD matrices.

The rest of the paper is organized as follows. In Section 2, we describe mathematical preliminaries that are required for learning over Riemannian manifolds and understanding the proposed modifications in the feature space. Section 3 provides more details on learning a data-dependent kernel by preserving geometry. Section 4 reports our experiments on a BCI data set. Our findings are discussed in Section 5, and concluding notes are mentioned in Section 6.

2. Preliminaries

In this section we describe basic concepts of Riemannian geometry that are necessary to understand our proposed approach. We review the metric applied in the SPD matrix space, its associated log and exp map, and the kernel functions from a geometrical point of view [22,23].

2.1. Riemannian geometry

The Riemannian metric on the Riemannian manifolds is a positive definite metric that takes two tangent vectors as inputs and generates a real number, which is a generalization of the inner product, and allows the similarity or dissimilarity of two points on the manifold to be measured [13,16,45]. A common invariant Riemannian metric on the tangent space of the SPD matrices [15,33,34] is defined as

$$< y,z >_X = \mathrm{trace}(X^{-\frac{1}{2}} y X^{-1} z X^{-\frac{1}{2}}) \tag{1}$$

where X denotes a point on the manifold and y and z show tangent vectors in the tangent space formed at point X.

The length of the curves along the manifold is computed by integrating the metric tensor along the curve, which connects two points on the manifold [13,26]. The geodesic, which is the local distance-minimizing curve over the manifold of SPD matrices associated with a metric from Eq. (1), is computed as

$$d_G{}^2(X,Y) = < \log_X(Y), \log_X(Y) >_X = \mathrm{trace}\left(\left(\log^2\left(X^{-1/2} Y X^{-1/2}\right)\right)\right. \tag{2}$$

where X and Y are two points on the manifold, $\log_X(Y)$ is the Riemannian log map of point Y to the tangent space formed at point X, and d_G denotes the geodesic distance on the manifold of the SPD matrices [42]. The Riemannian log map projects a point on the manifold to a point in tangent space. Its inverse is Riemannian $\exp_X(y)$, which projects a tangent vector $y \in T_X M$ into a point Y on the manifold.

The Riemannian exponential and logarithmic mappings associated to the metric of Eq. (1) are defined as

$$\exp_X(y) = X^{1/2} \exp\left(X^{-\frac{1}{2}} y X^{-\frac{1}{2}}\right) X^{1/2} \tag{3}$$

$$\log_X(Y) = X^{1/2} \log\left(X^{-\frac{1}{2}} Y X^{-\frac{1}{2}}\right) X^{1/2} \tag{4}$$

where exp and log are matrix exponential and logarithmic functions that are calculated as:

$$\begin{aligned} \exp \Sigma &= \sum_{k=0}^{\infty} \frac{\Sigma^k}{k!} = U \exp(D) U^T, \Sigma = UDU^T \\ \log \Sigma &= \sum_{k=1}^{\infty} (-1)^{k-1} \frac{(\Sigma - I)^k}{k} = U \log(D) U^T,\ \Sigma = UDU^T \end{aligned} \tag{5}$$

Eq. (5) assumes that Σ is decomposed into eigenvalues and vectors. Note that exp operator on the matrices always exists, while the log operator is defined only on symmetrical matrices with positive eigenvalues [15].

2.2. Kernel geometry

Kernel function $K(.,.)$ corresponds to the inner product in a high dimensional space H.

$$K(x,x') = \varphi(x).\varphi(x') \tag{6}$$

where φ is a projection of the input space S into the higher dimensional space H. The kernel function $K(.,.)$ induces a Riemannian metric to S using mapping φ, which is computed as [1,45]

$$g_{ij}(x,x') = \frac{\partial}{\partial x_i} \frac{\partial}{\partial x'_j} K(x,x')|_{x=x'} \tag{7}$$

where x_i denotes ith basis of vector x. Eq. (7) is written in Einstein summation notation. The volume element corresponding to the induced metric in input space is computed as [45]

$$dV = \sqrt{g(x)} dx_1 \ldots dx_n \tag{8}$$

where $g(x)$ represents the determinant of the matrix whose elements are g_{ij} and dV denotes the volume element. The expression $\sqrt{g(x)}$ is a factor that controls the expansion and contraction of volume elements [44].

2.3. Kernel principal component analysis

Kernel Principal Component Analysis (KPCA) (Algorithm 1) [40], which is widely used in dimensionality reduction and denoising applications, is a nonlinear generalization of principal component analysis (PCA) [19]. Classical PCA is designed to reduce dimensionality in the cases where the manifold is linearly

Please cite this article as: K. Sadatnejad, S. Shiry Ghidary, Kernel learning over the manifold of symmetric positive definite matrices for dimensionality reduction in a BCI application, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.11.065

embedded in the observation space. KPCA, which is composed of the kernel trick and PCA, provides the prerequisites of its later component by linearizing the manifold using the former component. KPCA projects data into a feature space implicitly using feature mapping $\varphi(x_i)$ and computes the pairwise scalar product between the mapped data in feature space *G* by using the kernel function. PCA is reformulated into an equivalent metric MDS version that is applied to the data projected in feature space. Finding an appropriate kernel, which considers the geometry of the input space to linearize the manifold in the feature space, is not a trivial problem. An inappropriate projection that does not provide these conditions would lead to the inadequacy of KPCA in nonlinear dimensionality reduction.

Algorithm 1. Kernel PCA algorithm [24]

1. Compute the matrix of scalar products, *S*, or the matrix of squared Euclidean distances, *D*, depending on the chosen kernel from observations *Y*.
2. Compute the matrix of kernel values *G*.
3. Centralize the projected points in the feature space.
4. Decompose the centralized into eigenvalues and eigenvectors, $G=U\Lambda U^{T}$.
5. A P-dimensional representation of *Y* is obtained by computing $\tilde{X}=I_{P\times N}\Lambda^{1/2}U^{T}$.

3. Data-dependent geometry preserving kernel

In this section, we describe our proposed data-dependent kernel, which adapts to the geometry of data points lying on the manifold of SPD matrices. We first describe an isometric kernel over the manifold of SPD matrices and show the drawbacks of this mapping and then rectify the isometric kernel by learning an appropriate conformal transformation. Some important notations that are used in this section are listed in Table 1.

3.1. Isometric kernel

The main goal of the NLDR methods is preserving the geometry during the mapping of observations to a low dimensional space. To relate the geometry of the observed data to the structure of the latent variables, two available choices are isometric and conformal embeddings [12]. Isometric embedding preserves the geometry by preserving the geodesic distances. This embedding results in preserving the geometrical structure of the manifold and the distribution of class labels over the manifold, which is influenced by the similarity of objects in the representation space (due to the compactness hypothesis). As preserving geodesic distances has an influence on the efficiency of learning methods, we construct our proposed kernel based on an isometric kernel.

To compute this kernel over the manifold of SPD matrices, considering the Riemannian geometry of the manifold, we apply a double centering algorithm [11] to the matrix of geodesic distances between data points, which is computed as follows:

$$K_0=-\frac{1}{2}Jd_G{}^2J$$
$$J=I_{N\times N}-\frac{1}{N}1_N\times 1_N{}^T \qquad (9)$$

where K_0 is the inner product matrix, d_G is the matrix of geodesic distances, and *N*denotes the number of data points. The expression $I_{N\times N}$ is an $N\times N$ identity matrix, and 1_N is a column vector where the elements are 1.

To compute d_G, a collection of tangent spaces is implicitly formed at different points, and in each implicit tangent space, the geodesic distances along the radial geodesics are computed using Eq. (2). Eq. (2) calculates the dissimilarity between the basepoint of a tangent space and other points that are mapped to that tangent space, which is the same as the actual value along the manifold. Therefore, we have implicitly formed a tangent space at a point X_i on the manifold. The distances between the basepoint of the tangent space, X_i, and the projection of any other points, X_j, in that tangent space, which is denoted by $dG(i,j)$, is computed by Eq. (2), which is equal to the geodesic distance between X_i and X_j. Iterating this procedure for every point as the basepoint of an implicit tangent space and using Eq. (2) results in an $N\times N$ dimensional matrix of geodesic distances, d_G. The resulting matrix d_G would represent the actual dissimilarity between all pairs of samples. Because $dG(i,j)=dG(j,i)$, Algorithm 2 would eliminate redundant computations for computing *dG*.

Algorithm 2. Compute the matrix of the geodesic distances over the manifold of SPD matrices.

```
For $i=1:N-1$
  (Implicitly form a tangent space, $T_{Xi}(M)$, at point X$_i$of the manifold M)
  For $j=i+1:N$
      (Implicitly project $X_j$to $T_{Xi}(M)$)
      Compute $d_G(i,j)$ using Eq. (2) between $X_i$ and $X_j$
      (Note that $d_G(j,i)=d_G(i,j)$)
  End
      $d_G(i,i)=0$
End
```

Applying an isometric kernel over the manifolds with nonzero intrinsic curvatures (i.e., non-developable manifolds) leads to an indefinite kernel [14,35]. The negative eigenvalues of the resulting Gram matrix are the consequence of the nonlinear structure of the manifold and application of the Riemannian metric. This indefinite kernel leads to suboptimal solutions in classification problems and may transfer data points into pseudo-Euclidean space in dimensionality reduction applications [14,35]. Removing the negative eigenvalues of the Gram matrix over the manifold of SPD matrices

Table 1
Some important notations that are used in Section 3.

Notation	Description	Notation	Description
K	Kernel matrix	λ_i	*i*th eigenvalue of resulting Gramian matrix.
g_{ij}	Riemannian metric	$c(x)$	Conformal transformation of *x*
d_G	Matrix of geodesic distances	α_i,δ	Unknown parameters of conformal transform
K_0	An isometric kernel matrix which is computed based on d_G	$C_{N\times N}$	A diagonal matrix in which its diagonal elements are conformal transforms of train samples
φ	A feature mapping from input space, *S*, to a high dimensional space *H*	$C'_{M\times M}$	A diagonal matrix in which its diagonal elements are conformal transforms of test samples

Please cite this article as: K. Sadatnejad, S. Shiry Ghidary, Kernel learning over the manifold of symmetric positive definite matrices for dimensionality reduction in a BCI application, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.11.065

may lead to the overlapping of the data points and consequently missing local information. In classification applications, the increased class overlap that occurs by removing negative eigenvalues can cause a decline in performance.

3.2. Conformal mapping

As stated in Section 3.1, due to the importance of the negative eigenvalues of the isometric kernel over the manifolds with non-zero intrinsic curvatures, we might not be able to remove them to preserve the geometry. However, we might be able to manipulate the isometric kernel using a geometry-preserving transform that decreases the negative eigenvalues.

Our main aim is learning a kernel that leads to changing the metric so that it modifies the volume elements to decrease the non-Euclidean characteristics while preserving the geometry. As mentioned in Eq. (8), the volume element is proportional to a factor that is computed based on the Riemannian metric induced by feature mapping φ in the input space. Therefore, modifying the kernel leads to changes in the induced metric and, consequently, the volume element. Our choice for modifying the metric is applying a conformal transformation that preserves the local geometry by preserving the local angles. The transformation can be defined as:

$$k(x_i,x_j)=c(x_i)k_0(x_i,x_j)c(x_j) \tag{10}$$

where k_0 is called the basic kernel and $c(x_i)$ denotes a conformal transform of x_i. In this study,$c(x_i)$ is defined by the following formula [47]:

$$c(x)=\alpha_0+\sum_{i=1}^{N}\alpha_i e^{-\delta||x-a_i||^2} \tag{11}$$

where α_i and δ denote unknown parameters that should be tuned using an optimization process;a_is are called empirical cores, which can be selected randomly or based on the geometry of the training dataset; and N denotes the number of cores.

The desired kernel is achieved by learning the unknown parameters of Eq. (10) so that they would decrease the negative eigen fraction (NEF) of the resulting kernel. The NEF is the result of the nonlinear structure of the manifold and is used to quantify the non-Euclidean characteristics of the manifold. The NEF is defined as:

$$\mathrm{NEF}=\frac{\sum_{\lambda_i<0}|\lambda_i|}{\sum_i|\lambda_i|} \tag{12}$$

where λ_i is the ith eigenvalue of the kernel matrix, which is computed as

$$\begin{aligned}&K=C\times K_0\times C,\\&C=diag([c(x_1),\ldots,c(x_N)]\end{aligned} \tag{13}$$

where K is the proposed kernel matrix that depends on α_i and δ parameters, K_0 is an $N\times N$ isometric kernel matrix over the training set, and C is an $N\times N$diagonal matrix with diagonal elements $c(x_i)$s. The expression $c(x_i)$ denotes the conformal transformation of x_i, which is computed using Eq. (11). The parameters $x_i$are training samples, and N denotes the number of training samples.

Tuning the unknown parameters of the proposed kernel is performed in an iterative process using a genetic algorithm, which is a heuristic technique for optimization [30,37]. Our solution space is an array of parameters of the model, including the weight of different cores and variance parameters. The NEF of the Gram matrix of proposed kernel over the training dataset is used as the fitness function for evaluating the chromosomes as the solutions of the optimization problem. The stopping criterion should be set to lead to the reduction of the negative eigen fraction of our proposed kernel over the training set. The stopping criterion is set as a fraction of the negative eigen fraction of the isometric kernel, which should be rectified by learning an appropriate conformal transform in our proposed kernel. The resulting δ and α_i parameters are used to compute similarity values between the test samples and the training set as the following

$$\begin{aligned}&K_{test}=C'_{M\times M}\times K_{0_{M\times N}}\times\ C^T_{N\times N}\\&C'=diag\left(\left[c(x'_1),\ldots,c(x'_M)\right]\right.\\&C=diag([c(x_1),\ldots,c(x_N)]\end{aligned} \tag{14}$$

where K_{test} is a $M\times N$ matrix that shows the similarity between the test and training samples. $K_{0M\times N}$ is a $M\times N$ matrix that denotes the isometric kernel matrix between the test and the training samples. C and C' are diagonal matrices where their diagonal elements are conformal transforms of training and test samples, respectively, and N and M are the number of training and test samples. x'_i and x_i denotes the ith test and train samples, respectively.

The method for learning the proposed data dependent kernel and for using it as a kernel in a dimensionality reduction procedure is described as Algorithm 3.

Algorithm 3. Dimensionality reduction over the manifold of SPD matrices.

1. Divide the training dataset into empirical cores, a_i which are selected randomly and a smaller training dataset.
2. Compute the isometric kernel matrix, K_0, over the training set, Eq. (9).
3. Learn a conformal transform using GA that uses NEF of K_{train} as a fitness function,
   where $K_{train}=C_{N\times N}\times K_{0N\times N}\times {C_{N\times N}}^T$,
   $C=diag([c(x_1),\ldots,c(x_N)]$.
4. Compute K_{test}, which is the similarity matrix between the training and test samples.
   $K_{test}=C'_{M\times M}\times K_{0M\times N}\times\ C^T{}_{N\times N}$, $C'=diag([c(x'_1),\ldots,c(x'_\mathrm{M})]$
5. Run the kernel PCA using the resulting data dependent kernel.

4. Evaluations

To assess the proposed kernel, we used it as a kernel in kernel PCA for dimensionality reduction. The experiments were run over data set IIa of the BCI competition IV [31]. The 1-Nearest Neighbor classifier (1-NN) is used to evaluate the proposed method in comparison with the most popular nonlinear dimensionality reduction techniques, as shown in Table 2. We have also made the comparison against CSP with the Linear Discriminant Analysis classifier (CSP+LDA), which is a reference method in BCI competitions.

4.1. Data set IIa, BCI competition IV

Data set IIa of BCI competition IV contains EEG signals that are captured from 9 subjects while performing four different motor imageries, including Left Hand (LH), Right Hand (RH), Foot (F), and Tongue (T) MIs. Twenty-two electrodes lying over the scalp are used for recording EEG signals. For each class and subject, 72 trials are recorded as training and test sets in different sessions. In this study, we assign each trial to one of the four specified classes. For each trial, the features are extracted from 0.5 s to 2.5 s after the cue that is used to perform MI by the subjects. The trials are band-pass filtered in 8–35 Hz using a 5th order Butterworth filter

Please cite this article as: K. Sadatnejad, S. Shiry Ghidary, Kernel learning over the manifold of symmetric positive definite matrices for dimensionality reduction in a BCI application, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.11.065

Table 2
Accuracy of kernel PCA (RBF and CILK), Isomap, LLE, LE, and LTSA with 1-NN classifier and CSP with LDA classifier on all pairs of MIs over dataset IIa, BCI competition IV.

LH/RH	S1	S2	S3	S4	S5	S6	S7	S8	S9	Mean ± STD
KPCA(CILK)	88.89	59.03	90.28	78.47	62.50	75.00	72.92	93.06	87.50	**78.63** ± 12.34
KPCA(RBF)	54.17	49.31	54.17	45.14	46.53	54.86	50.69	47.22	43.06	49.00 ± 3.94
Isomap	50.00	50.00	56.25	52.08	50.00	52.08	47.92	65.97	72.92	55.25 ± 8.55
LTSA	49.31	53.47	50.00	49.31	48.61	45.83	46.53	48.61	60.42	50.23 ± 4.39
LE	47.22	42.36	55.56	50.69	47.22	48.61	44.44	52.08	33.33	46.83 ± 6.24
LLE	60.42	47.92	84.03	64.58	54.86	59.03	53.47	84.72	76.39	65.05 ± 13.53
CSP+LDA	88.89	51.39	96.53	70.14	54.86	71.53	81.25	93.75	93.75	78.01 ± 17.01
LH/F	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**Mean ± STD**
KPCA(CILK)	97.92	86.81	96.53	81.94	70.14	75.69	88.89	85.42	97.22	**86.73** ± 9.73
KPCA(RBF)	44.44	54.86	54.86	48.61	44.44	52.08	52.08	59.72	50.00	51.23 ± 5.01
Isomap	63.19	53.47	64.58	70.14	50.00	61.11	50.69	50.00	63.89	62.34 ± 11.15
LTSA	47.92	47.92	54.17	57.64	40.28	54.17	44.44	54.86	47.22	49.85 ± 5.68
LE	44.44	49.31	59.03	45.14	47.22	54.17	45.14	50.00	45.83	48.92 ± 4.90
LLE	84.03	53.47	77.78	64.58	47.22	59.03	61.81	70.83	86.11	67.21 13.49
CSP+LDA	98.61	68.75	94.44	78.47	63.19	59.03	97.92	87.50	95.14	82.56 ± 15.63
LH/T	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**Mean ± STD**
KPCA(CILK)	96.53	71.53	93.75	85.42	77.78	75.00	87.50	88.89	97.92	**86.04** ± 9.50
KPCA(RBF)	50.00	45.83	47.22	44.44	52.78	45.83	58.33	50.69	64.58	51.08 ± 6.64
Isomap	69.44	45.83	65.97	65.97	48.61	51.39	68.75	65.97	79.17	62.34 ± 11.15
LTSA	45.83	54.17	46.53	47.92	55.56	47.22	57.64	55.56	59.03	52.16 ± 5.22
LE	47.92	51.39	58.33	44.44	47.42	55.56	45.14	51.39	45.83	49.71 ± 4.83
LLE	88.89	53.47	80.56	72.22	52.78	65.97	70.14	77.78	90.97	72.53 ± 13.70
CSP+LDA	98.61	67.36	94.44	86.81	68.75	71.53	95.14	90.97	95.14	85.42 ± 12.62
RH/F	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**Mean ± STD**
KPCA(CILK)	98.61	90.97	90.97	84.03	67.36	74.31	95.83	88.19	86.11	**86.26** ± 9.98
KPCA(RBF)	50.69	50.69	51.39	49.31	45.14	47.22	49.31	51.39	47.22	49.15 ± 2.19
Isomap	60.42	53.47	61.11	65.97	47.92	64.58	51.39	59.72	44.44	56.56 ± 7.55
LTSA	46.53	49.31	53.47	58.33	46.53	51.39	52.08	47.92	49.31	50.54 ± 3.78
LE	52.78	45.14	52.08	45.83	42.36	54.86	51.39	54.84	47.92	49.69 ± 4.53
LLE	84.72	58.33	79.17	61.11	43.06	60.42	70.83	67.36	54.17	64.35 ± 12.77
CSP+LDA	97.22	81.25	93.06	88.89	68.75	63.19	99.31	86.81	84.72	84.80 ± 12.20
RH/T	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**Mean ± STD**
KPCA(CILK)	98.61	90.97	97.22	85.42	72.92	75.69	95.83	87.50	89.58	**88.19** ± 9.05
KPCA(RBF)	53.47	47.22	51.39	59.72	48.61	57.64	53.47	54.17	43.75	52.16 ± 5.04
Isomap	70.14	45.83	62.50	63.89	48.61	53.47	68.06	63.19	67.36	60.34 ± 8.84
LTSA	47.92	57.64	51.39	54.17	47.22	50.69	59.03	52.78	48.61	52.16 ± 4.17
LE	52.78	50.69	55.56	51.39	50.69	49.31	57.64	56.94	47.22	52.47 ± 3.56
LLE	93.06	47.92	81.94	63.19	50.00	59.72	70.83	68.06	70.83	67.28 ± 14.37
CSP+LDA	100.00	63.89	96.53	85.42	65.28	65.97	97.22	91.67	81.94	83.10 ± 14.67
F/T	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**Mean ± STD**
KPCA(CILK)	81.94	89.58	79.86	72.22	73.61	69.44	80.56	86.11	89.58	**80.32** ± 7.39
KPCA(RBF)	54.86	49.31	45.14	59.72	47.22	48.61	47.92	48.61	45.83	49.69 ± 4.67
Isomap	50.69	49.31	51.39	44.44	50.69	47.22	63.19	61.11	68.06	54.01 ± 8.07
LTSA	52.78	48.61	45.14	51.39	47.22	45.83	53.47	53.47	55.56	50.39 ± 3.77
LE	50.00	47.92	44.44	63.89	51.39	47.92	46.53	52.78	55.56	51.16 ± 5.84
LLE	58.33	53.47	63.19	53.47	51.39	55.56	67.36	72.92	72.92	60.96 ± 8.45
CSP+LDA	69.44	69.44	69.44	56.94	70.83	67.36	81.25	82.64	88.89	72.91 ± 9.66

[28]. The covariance matrix of each trial is computed using Eq. (15).

$$C = \frac{1}{T-1} E \times E^T \tag{15}$$

where T shows the epoch duration; E is the $N \times T$ dimensional EEG signal, while N shows the number of channels used for recording EEGs; and C denotes the resulting $N \times N$ dimensional covariance matrix. In this data set, 22 channels have been used to record the EEG signals. We therefore have 22 × 22 dimensional descriptors.

4.2. Experiments

We evaluated the proposed kernel in a dimensionality reduction problem. Because CSP+LDA is appropriate for two-class classification problems, we ran our experiments over the pairs of two MIs [9] in Tables 2 and 3. Therefore, the signals are divided into LH/RH, LH/F, LH/T, RH/F, RH/T, and F/T subsets. In all of the experiments, different methods were trained using training trials and were evaluated on the test trials, except in Table 5, where we applied 10-fold cross-validation over the entire dataset.

Table 3
Mean accuracy and standard deviation of kernel PCA (CILK)+1-NN, MDM, and CSP+LDA on all pairs of MIs in BCI competition IV, dataset IIa

	KPCA(CILK) +1-NN	CSP+LDA	MDM
	Mean ± STD	**Mean ± STD**	**Mean ± STD**
LH/RH	**78.63** ± 12.34	78.01 ± 17.01	72.00 ± 30.00
LH/F	**86.73** ± 9.73	82.56 ± 15.63	85.41 ± 8.88
LH/T	**86.04** ± 9.50	85.42 ± 12.62	82.95 ± 12.21
RH/F	**86.26** ± 9.98	84.80 ± 12.20	83.33 ± 12.11
RH/T	**88.19** ± 9.05	83.10 ± 14.67	82.02 ± 11.50
F/T	**80.32** ± 7.39	72.91 ± 9.65	72.92 ± 7.92
Mean	**85.68** ± **8.68**	81.15 ± 12.00	81.15 ± 8.64

Popular dimensionality reduction techniques assume that data points are embedded in Euclidean space. Therefore, applying these methods over SPD matrices requires the conversion of the matrices to points of Euclidean space. For this purpose, the matrices need to be vectorized by stacking the columns of each matrix on top of each other and converting them to a column vector. Note that our proposed kernel receives the matrix of geodesic distance,

Please cite this article as: K. Sadatnejad, S. Shiry Ghidary, Kernel learning over the manifold of symmetric positive definite matrices for dimensionality reduction in a BCI application, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.11.065

Table 4
Performance of KPCA (CILK)+1-NN and the first 3 winners of the BCI competition 2008 on dataset IIa (4-class problem) in terms of the kappa value.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	Mean
First method	0.6800	0.4200	0.7500	0.4800	0.4000	0.2700	0.7700	0.7500	0.6100	0.5700
Second method	0.6900	0.3400	0.7100	0.4400	0.1600	0.2100	0.6600	0.73 00	0.6000	0.5200
Third method	0.3800	0.1800	0.4800	0.3300	0.0700	0.14 00	0.2900	0.49 00	0.4400	0.3100
KPCA (CILK)+1-NN	0.7407	0.4259	0.7407	0.4815	0.2315	0.2963	0.7454	0.7454	0.7130	0.5689

Table 5
Performance of KPCA with proposed kernel (CILK), [3,18], Gaussian, and polynomial kernels+LDA classifier in terms of the kappa value, according to a 10-fold cross-validation on dataset IIa, BCI competition IV.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	Mean
CILK	0.7079	0.4564	0.7612	0.4419	0.2600	0.3679	0.7887	0.7479	0.6127	0.5716
Harandi et al. [18]	0.5865	0.6059	0.6481	0.4237	0.1528	0.2718	0.7157	0.7420	0.5596	0.5229
Barachant et al. [3]	0.7917	0.4880	0.8151	0.3460	0.2285	0.2693	0.6539	0.7082	0.3304	0.5146
Gaussian	0.0139	0.0220	0.0078	0.0123	0.0148	0.0231	0.0407	0.0644	0.0147	0.0237
Polynomial	0.0550	0.0039	0.0203	0.0217	0.0217	0.0167	0.0167	0.0198	0.0158	0.0231

d_G, as the input and manipulates this matrix. Our kernel considers the geometry of the manifold by using Eq. (2) for computing the geodesic distances between SPD matrices (Algorithm 2). The vectorization that destroys the geometry of the manifold is not required for the proposed method.

We named the proposed kernel as a Conformal-Isometric Linearizing Kernel (CILK) and compared it with the RBF kernel (in a kernel PCA setup) as well as other popular NLDR techniques, including Isomap, LLE, LE, and LTSA. Drtoolbox [43] is used to implement these techniques. In this experiment, the number of neighbors needed to construct the graph in Isomap, LLE, and LE is determined empirically. On average, we chose the 20 nearest neighbors to construct the graphs. The dimensionality of the low dimensional space was determined experimentally by evaluating different dimensions and reporting the best results. In most cases, the best results were achieved for less than 50 dimensions. A wide range of values has been investigated to tune the variance parameter of the RBF kernel. K-fold cross-validation was applied over the training set, and the setting that led to the maximum average performance was used as our choice for evaluating the test set. In the case of CSP+LDA [9], three pairs of spatial filters were selected, which is a common setting in the BCI problem [7]. Solving the optimization problem that leads to our proposed method is performed by using GA, which is implemented using the Matlab genetic algorithm toolbox. Approximately 10% of the training data are devoted as the cores. The variance parameter is constrained to be a positive value, and the lower bound of the weight parameters is set to zero. The stopping criterion is set to a fraction (0.01) of the NEF of the empirical isometric kernel over the training set.

Table 2 shows the accuracy of classification over the LH/RH, LH/F, LH/T, RH/F, RH/T, and F/T pairs of MIs for each subject and the average for each pair of MIs. In this experiment, different dimensionality reduction techniques+1-NN classifier and CSP+LDA were trained over training trials and were evaluated during the test trials.

Fig. 1 illustrates the distribution of data points corresponding to the subjects in the LH/RH pair of MIs using isometric mapping into two dimensions. We use this figure to emphasize the relationship between the distribution of different classes and the efficiency of the proposed method in comparison with CSP+LDA, which are reported in Table 2. Comparing CSP+LDA and our proposed method shows 3 distinct states: our algorithm performs better, both methods behave similarly, and CSP+LDA is the superior algorithm. Fig. 1 illustrates the subjects that correspond to these three states in Fig. 1(a), (b), and (c), respectively.

A comparison of the kernel PCA with the CILK kernel and 1-NN classifier, Minimum Distance to Mean (MDM) [9], and CSP+LDA on the EEG signals of all of the pairs of MIs is reported in Table 3.

To compare the significance of CSP+LDA, KPCA (CILK)+1-NN, and MDM with respect to each other over the means of the accuracies of the pairs of MIs for each subject, the nonparametric Wilcoxon test is used. KPCA (CILK)+1-NN predicts a significantly better than CSP+LDA and MDM, with $p=0.028$ and $p=0.011$, respectively. However, MDM with $p>0.05$ shows an insignificant performance with respect to the CSP+LDA method.

Table 4 shows a comparison between the proposed method and the first three winners of the BCI competition 2008 on dataset IIa. The methods are evaluated on the test set, and the results are reported in terms of the kappa value. The proposed method, with an average performance of 0.5689, achieved second place in this experiment, with a very slight difference from the winner.

To verify the effectiveness of the proposed kernel, we compare it with other kernels (Table 5). The Riemannian kernel [18] and the kernel proposed by Barachant et al. [3] are kernels that consider the geometry of the SPD manifolds. We use the geometric mean as the reference point in Barachant's kernel. Gaussian and polynomial kernels, which are based on Euclidean geometry, are also compared. Experiments were performed by applying k-fold cross-validation to the total training and validation sets. The experimental results show the superiority of the proposed kernel in comparison with the competitors. The experimental results confirm the effectiveness of considering the geometry of the input space and the shortcoming of the kernels that rely on Euclidean geometry on the manifold of SPD matrices.

5. Discussion

In the experiments described in Section 4, the SPD matrices are formulated as a Riemannian manifold that lives in the non-Euclidean space. The experimental results on this manifold show the superiority of the proposed approach in comparison with popular manifold learning methods. The lower accuracy of the popular manifold learning methods, such as Isomap, LLE, LE, and LTSA, which are reported in Table 2, are a result of the inconsistency of the requirements of the above-mentioned methods for the geometry of the Riemannian manifold of SPD matrices. The decreased performance of those methods can be explained as

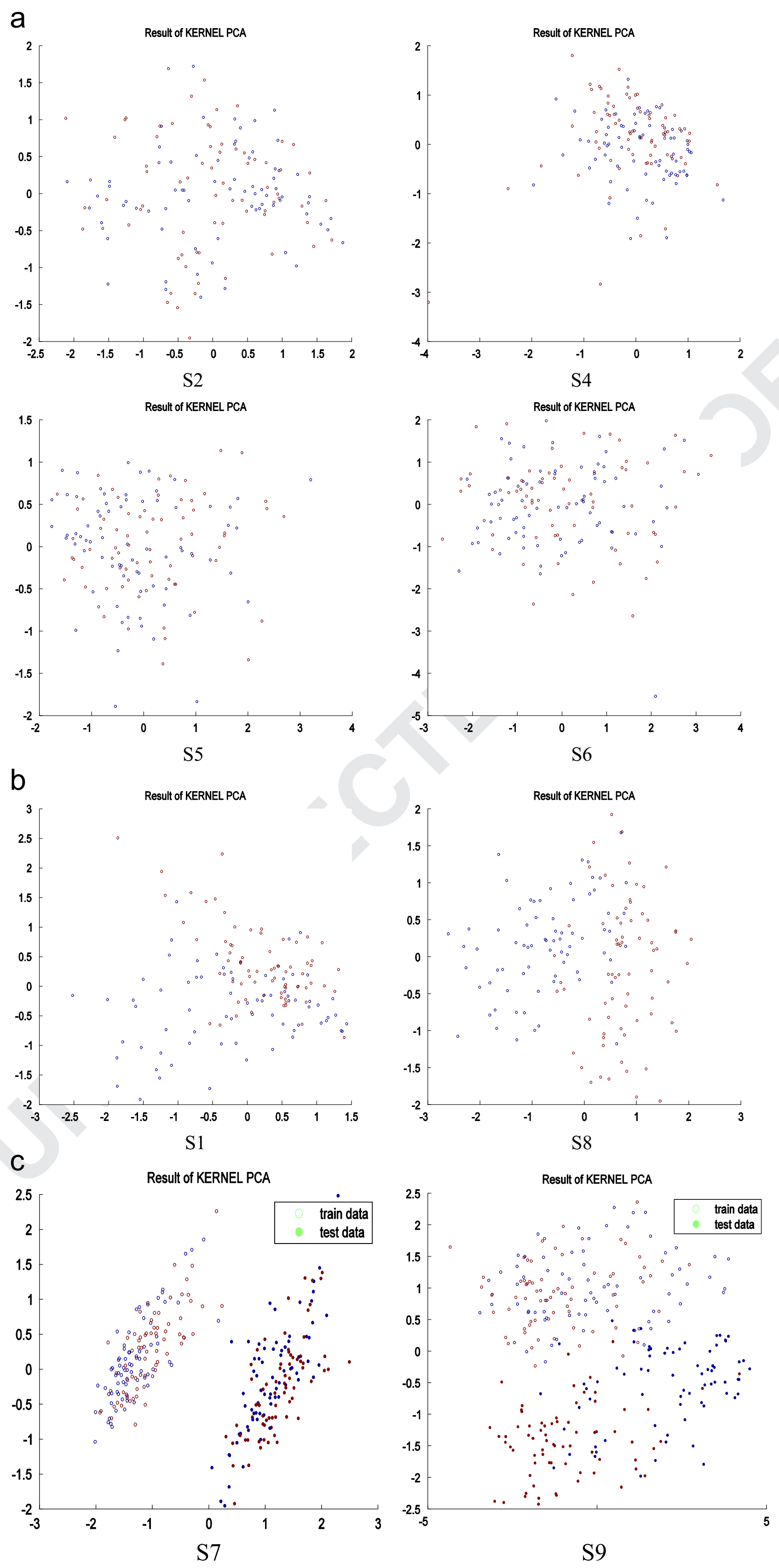

Fig. 1. Two dimensional representations of the subjects for LH/RH using kernel PCA with isometric kernel (a) subject nos. 2,4,5, and 6; (b) subject nos. 1 and 8, and (c) subject nos. 7 and 9.

In the case of LE, which is a local method and uses a Laplacian matrix to represent the manifold, the deficiency is the result of approximating true geodesic distances by graph distances.

LLE, which attempts to preserve local linearity, computes a weight matrix to represent each data point as a linear combination of its neighbors. This aim is achieved by solving a least-squares problem in Euclidean space, while on the Riemannian manifold, solving an interpolation problem on the manifold is required.

LTSA needs to provide a local parameterization of the data points by relying on the assumption that data points are embedded in Euclidean space. The local coordinates around each point are computed by a Taylor series expansion in Euclidian space at the tangent space around the base point, which is computed using PCA. Because LTSA estimates the tangent space of the Riemannian manifold at a point using the available data samples in the neighborhood of the base point, the sampling conditions, such as the sampling extent and density, affect the estimated tangent space. Running PCA on some instances of the Riemannian manifold leads to inaccurate local information, which leads to poor classification results.

The comparison between RBF and CILK kernels, as shown in Table 2, demonstrates the significance of considering the geometry of the input data. As shown in Table 2, the proposed approach in some cases shows considerable superiority over the CSP+LDA method. Plotting samples in two dimensions using the isometric kernel, which is illustrated in Fig. 1(a), shows that the superiority of our proposed method (Table 2) corresponds to cases where different classes have complex non-linearly separable distributions. For linearly separable samples (Fig. 1(b)), our proposed method and CSP+LDA would achieve a similar performance (Table 2). For the cases where the training data do not provide a good covering over the feature space (Fig. 1(c)), CSP+LDA shows superiority, which is the result of using a discriminative classifier (Table 2).

As shown in Table 3, for all pairs of MIs, our proposed method results in higher accuracy with a smaller standard deviation in comparison with the CSP+LDA. The observed superiorities, especially in complex non-linearly separable cases, are the result of the strength of the local classifiers in these cases. The strength of these methods is strongly dependent on providing their prerequisites. 1-NN, which is used in our experiments, suffers from the curse of dimensionality. Overcoming the curse of dimensionality is a prerequisite for the 1-NN classifier, which is provided by reducing the dimensionality by decreasing the non-Euclidean characteristics while preserving the topology of the data points. The lower standard deviation that is achieved for the kernel PCA (CLIK)+1-NN is the result of the strength of the proposed approach in complex non-linearly separable cases.

6. Conclusions

In this paper, we propose a kernel for reducing dimensionality over manifolds with a known geometry (e.g., the manifold of SPD matrices). Preserving the geometrical structure of the manifold based on Riemannian geometry provides a kernel that is adapted to the manifold. The novelty of our algorithm is the modification of the volume elements to decrease the non-Euclidean characteristics of the manifold, which is represented by the negative eigen fraction of the resulting Gramian matrix in the feature space. Embedding to a lower dimensional space with this topology preserving mapping and using 1-NN for its classification leads to superior accuracy over the methods that are based on popular NLDR techniques and CSP +LDA. These superiorities are found, especially in the cases where samples have complex and nonlinear separable distribution. Considering the geometry of the input space and applying a classifier that relies on local information provides these superiorities in complex nonlinear separable cases, which leads to a lower standard deviation.

Acknowledgments

This work was supported by funding from the Institute for Cognitive Sciences and Technologies Council. The first author specially acknowledges Miss Alimardani for her help and useful discussions and A. Barachant for his useful guides. Q3

References

[1] S.I. Amari, S. Wu, Improving support vector machine classifiers by modifying kernel functions, Neural Netw. 12 (6) (1999) 783–789.
[2] A. Barachant, S. Bonnet, M. Congedo, C. Jutten, Riemannian geometry applied to BCI classification, Latent Variable Analysis and Signal Separation, Springer, Berlin Heidelberg (2010), p. 629–636.
[3] 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.
[4] A. Barachant, S. Bonnet, M. Congedo, C. Jutten, BCI signal classification using a Riemannian-based kernel, in: Proceedings of the 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2012), Michel Verleysen, April 2012, pp. 97–102.
[5] 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.
[6] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput. 15 (6) (2003) 1373–1396.
[7] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, K.R. Muller, Optimizing spatial filters for robust EEG single-trial analysis, IEEE Signal Process. Mag. 25 (1) (2008) 41–56.
[8] N. Brodu, F. Lotte, A. Lécuyer, Exploring two novel features for EEG-based brain–computer interfaces: multifractal cumulants and predictive complexity, Neurocomputing 79 (2012) 87–94.
[9] M. Congedo, A. Barachant, A. Andreev, A new generation of brain-computer interface based on Riemannian geometry arXiv Prepr. arXiv: 1310.8115, 2013.
[10] M. Congedo, A. Barachant, A special form of SPD covariance matrix for interpretation and visualization of data manipulated with Riemannian geometry, in: MaxEnt 2014, vol. 1641, AIP publishing LLC, January 2015, p. 495.
[11] T. Cox, M. Cox, Multidimensional Scaling, Chapman and Hall, London, 1994.
[12] V. De Silva, J.B. Tenenbaum, Global versus local methods in nonlinear dimensionality reduction, Adv. Neural Inf. Process. Syst. (2002) 705–712.
[13] T.K. Dey, K. Li, Cut locus and topology from surface point data, in: Proceedings of the Twenty-Fifth Annual Symposium on Computational Geometry, 2009, ACM, pp. 125–134.
[14] R.P. Duin, E. Pękalska, M. Loog, Non-Euclidean dissimilarities: Causes, embedding and informativeness. In: Similarity-Based Pattern Analysis and Recognition, Springer, London (2013), p. 13–44.
[15] W. Förstner, B. Moonen, A Metric For Covariance Matrices. In: Geodesy—The Challenge of the 3rd Millennium, Springer, Berlin Heidelberg (2003), p. 299–309.
[16] J. Gallier, Geometric Methods and Applications: For Computer Science and Engineering, Springer Science and Business Media, 2011.
[17] A. Goh, R. Vidal, Clustering and dimensionality reduction on Riemannian manifolds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008, pp. 1–7.
[18] M.T. Harandi, C. Sanderson, A. Wiliem, B.C. Lovell, Kernel analysis over Riemannian manifolds for visual recognition of actions, pedestrians and textures, in: Proceedings of the IEEE Workshop Applications of Computer Vision (WACV), 2012, pp. 433–439.
[19] I. Jolliffe, Principal Component Analysis, John Wiley and Sons, Ltd., 2002.
[20] F. Goksu, N.F. Ince, A.H. Tewfik, Greedy solutions for the construction of sparse spatial and spatio-spectral filters in brain computer interface applications, Neurocomputing 108 (2013) 69–78.
[21] Joseph B. Kruskal, Myron Wish, Multidimensional Scaling, Sage, 1978.
[22] J.M. Lee, Riemannian Manifolds: An Introduction to Curvature, Springer Science and Business Media, 1997.
[23] J. Jost, Riemannian Geometry and Geometric Analysis, Springer Science & Business Media, 2008.
[24] J.A. Lee, M. Verleysen, Nonlinear Dimensionality Reduction, Springer Science and Business Media, 2007.
[25] J. Li, Z. Struzik, L. Zhang, A. Cichocki, Feature Learning from Incomplete EEG with Denoising Autoencoder, Neurocomputing (2015).
[26] T. Lin, H. Zha, Riemannian manifold learning, IEEE Trans. Pattern Anal. Mach. Intell. 30 (5) (2008) 796–809.
[27] F. Lotte, C. Guan, Regularizing common spatial patterns to improve BCI designs: unified theory and new algorithms, IEEE Trans. Biomed. Eng. 58 (2) (2011) 355–362.
[28] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, A review of classification algorithms for EEG-based brain–computer interfaces, J. Neural Eng. 4 (2007).
[29] J. McCall, Genetic algorithms for modelling and optimisation, J. Comput. Appl. Math. 184 (1) (2005) 205–222.

Please cite this article as: K. Sadatnejad, S. Shiry Ghidary, Kernel learning over the manifold of symmetric positive definite matrices for dimensionality reduction in a BCI application, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.11.065

[30] M. Naeem, C. Brunner, R. Leeb, B. Graimann, G. Pfurtscheller, Seperability of four-class motor imagery data using independent components analysis, J. Neural Eng. 3 (3) (2006) 208.

[31] L.F. Nicolas-Alonso, R. Corralejo, J. Gomez-Pilar, D. Álvarez, R. Hornero, Adaptive semi-supervised classification to reduce intersession non-stationarity in multiclass motor imagery-based brain–computer interfaces, Neurocomputing 159 (2015) 186–196.

[32] W. Rossman, Lie Groups: An Introduction Through Linear Groups, Oxford, 2002.

[33] X. Pennec, P. Fillard, N. Ayache, A Riemannian framework for tensor computing, Int. J. Comput. Vis. 66 (1) (2006) 41–66.

[34] E. Pekalska, P. Paclik, R.P. Duin, A generalized kernel approach to dissimilarity-based classification, J. Mach. Learn. Res. 2 (2002) 175–211.

[35] G. Pfurtscheller, C. Neuper, Motor imagery and direct brain–computer communication, Proc. IEEE 89 (7) (2001) 1123–1134.

[36] A. Popov, Genetic Algorithms for Optimization 2013. User Manual, Hamburg, 2005.

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

[38] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.

[39] B. Schölkopf, A. Smola, K.R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (5) (1998) 1299–1319.

[40] J.B. Tenenbaum, V. De Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.

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

[42] L.J. Van der Maaten, E.O. Postma, H.J. van den Herik, Dimensionality reduction: a comparative review, J. Mach. Learn. Res. 10 (1-41) (2009) 66–71.

[43] P. Williams, S. Li, J. Feng, S. Wu, A geometrical method to improve performance of the support vector machine, IEEE Trans. Neural Netw. 18 (3) (2007) 942–947.

[44] S. Wu, S.I. Amari, Conformal transformation of kernel functions: a data-dependent way to improve support vector machine classifiers, Neural Process. Lett. 15 (1) (2002) 59–67.

[45] W. Wu, Z. Chen, S. Gao, E.N. Brown, A hierarchical Bayesian approach for learning sparse spatio-temporal decompositions of multichannel EEG, NeuroImage 56 (4) (2011) 1929–1945.

[46] H. Xiong, M.N.S. Swamy, M.O. Ahmad, Optimizing the kernel in the empirical feature space, IEEE Trans. Neural Netw. 16 (2) (2005) 460–474.

[47] Z.Y. Zhang, H.Y. Zha, Principal manifolds and nonlinear dimensionality reduction via tangent space alignment, J. Shanghai Univ. (Engl. Ed.) 8 (4) (2004) 406–424.

[48] Y.U. Zhang, G. Zhou, J. Jin, X. Wang, A. Cichocki, Frequency recognition in SSVEP-based BCI using multiset canonical correlation analysis, Int. J. Neural Syst. 24 (04) (2014) 1450013.

[49] Y. Zhang, G. Zhou, J. Jin, Q. Zhao, X. Wang, A. Cichocki, Aggregation of sparse linear discriminant analyses for event-related potential classification in brain–computer interface, Int. J. Neural Syst. 24 (01) (2014) 1450003.

[50] Y. Zhang, G. Zhou, J. Jin, M. Wang, X. Wang, A. Cichocki, L1-regularized multiway canonical correlation analysis for SSVEP-based BCI, IEEE Trans. Neural Syst. Rehabil. Eng. 21 (6) (2013) 887–896.

[51] Y. Zhang, G. Zhou, Q. Zhao, J. Jin, X. Wang, A. Cichocki, Spatial-temporal discriminant analysis for ERP-based brain-computer interface, IEEE Trans. Neural Syst. Rehabil. Eng. 21 (2) (2013) 233–243.

Saeed Shiry Ghidary was born in Zanjan, Iran. He received his B.Sc degree in Electronic engineering and Msc in computer architecture from Amirkabir University of Technology in 1990 and 1994 respectively. He studied robotics and artificial intelligent systems at Kobe University and received his PhD in 2002. He has been an assistant Prof. at Amirkabir University of Technology since 2004. His research interests include Robotics, Machine learning, Machine vision, Cognitive science and Brain modeling.

Khadijeh Sadatnejad received her BS and MS degrees in computer engineering from the Shiraz University in 2003 and 2010. She is currently a PhD student at the Department of Computer Engineering, Amirkabir University of Technology. Her research interests are in Machine learning, Biomedical signal processing, and Machine vision. She has currently concentrated on learning over Riemannian manifold and its application in BCI.