

Functional Brain Network Classification With Compact Representation of SICE Matrices

Jianjia Zhang*, Luping Zhou, *Senior Member, IEEE*, Lei Wang, *Senior Member, IEEE*,
and Wanqing Li, *Senior Member, IEEE*

Abstract—Recently, a sparse inverse covariance estimation (SICE) technique has been employed to model functional brain connectivity. The inverse covariance matrix (SICE matrix in short) estimated for each subject is used as a representation of brain connectivity to discriminate Alzheimers disease from normal controls. However, we observed that direct use of the SICE matrix does not necessarily give satisfying discrimination, due to its high dimensionality and the scarcity of training subjects. Looking into this problem, we argue that the intrinsic dimensionality of these SICE matrices shall be much lower, considering 1) an SICE matrix resides on a Riemannian manifold of symmetric positive definiteness matrices, and 2) human brains share common patterns of connectivity across subjects. Therefore, we propose to employ manifold-based similarity measures and kernel-based PCA to extract principal connectivity components as a compact representation of brain network. Moreover, to cater for the requirement of both discrimination and interpretation in neuroimage analysis, we develop a novel preimage estimation algorithm to make the obtained connectivity components anatomically interpretable. To verify the efficacy of our method and gain insights into SICE-based brain networks, we conduct extensive experimental study on synthetic data and real rs-fMRI data from the ADNI dataset. Our method outperforms the comparable methods and improves the classification accuracy significantly.

Index Terms—Alzheimer’s disease (AD) classification, brain network, kernel principal component analysis (PCA), preimage estimation, rs-fMRI, symmetric positive definite (SPD) kernel.

I. INTRODUCTION

AS an incurable and the most common form of dementia, Alzheimer’s disease (AD) affects tens of million people worldwide. Precise diagnosis of AD, especially at its early warning stage: Mild cognitive impairment (MCI), enables treatments to delay or even avoid cognitive symptoms, such as language disorder and memory loss [1]. However, this is a very challenging task. Conventional diagnosis of MCI based on clinical observations and structural imaging [2] can hardly achieve accurate diagnosis since the symptoms of MCI are often ambiguous and not necessarily related to structural alterations [3]. Recent studies show that the functional connectivity between some brain

regions of AD patients differs from that of normal aging. For example, compared with the healthy, AD patients have been found decreased functional connectivity between hippocampus and other brain regions, and MCI patients have been observed increased functional connectivity between the frontal lobe and other brain regions [4]. Therefore, detecting these abnormal alterations in functional connectivity of AD can bring significant benefits in identifying novel connectivity-based biomarkers to improve the diagnosis confidence and revealing the mechanism of AD to help the development of therapies.

Constructing and classifying functional brain networks based on resting-state functional magnetic resonance imaging (rs-fMRI) [5] holds great promise for functional connectivity analysis [6], [7]. rs-fMRI focuses on the low-frequency (<0.1 Hz) oscillations of blood-oxygen-level-dependent signal, which presents the underlying neuronal activation patterns of brain regions [8]–[10]. Many methods have been proposed to model brain connectivity based on the covarying patterns of rs-fMRI time series across brain regions. Two issues are generally involved: identifying network nodes and inferring the functional connectivity between nodes. The network nodes are often defined as anatomically separated brain regions of interest (ROIs) or alternatively as latent components in some data-driven methods, e.g., independent component analysis [11], [12], and clustering-based methods [13], [14]. Given a set of network nodes, the functional connectivity between two nodes is conventionally measured by the correlation coefficient of time series associated with the two nodes (e.g., the averaged time series from all voxels within a node) [15]–[17], and the brain network is then represented by a correlation matrix.

However, it has been argued that partial correlation could be a better choice since it measures the correlation of two nodes by regressing out the effects from all other nodes [18]. This often results in a more accurate estimate of network structure in comparison with those correlation-based methods. Sparse inverse covariance estimation (SICE) is a principled method for partial correlation estimation, which often produces a stable estimation with the help of the sparsity regularization [19]. The result of SICE is an inverse covariance matrix, and each of its off-diagonal entries indicates the partial correlation between two nodes. It has been widely used to model functional brain connectivity in [20]–[22]. For brevity, we call it “SICE matrix” throughout this paper.

SICE matrices can be used as a representation to classify brain networks. A direct approach could be to vectorize each SICE matrix into a feature vector, as in [16]. However, when using it to train a classifier to separate AD from normal controls (NC),

Manuscript received August 17, 2014; accepted January 24, 2015. Date of publication February 4, 2015; date of current version May 18, 2015. Asterisk indicates corresponding author

*J. Zhang is with the School of Computer Science and Software Engineering, University of Wollongong, Wollongong, N.S.W. 2522, Australia (e-mail: jz163@uowmail.edu.au).

L. Zhou, L. Wang, and W. Li are with the School of Computer Science and Software Engineering, University of Wollongong.

This paper contains supplementary material available online at <http://ieeexplore.ieee.org> (File size: 1 MB).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TBME.2015.2399495

the problem of “the curse of dimensionality” arises since the dimensionality of the vector (at the order of $d \times d^l$ for a network with d nodes, for example, $d = 90$ in our study) is usually much larger than the number of training subjects, which is often only tens for each class. This usually leads to poor performance of classification. An alternative approach is to summarize a $d \times d$ SICE matrix into lower dimensional graphical features, such as local clustering coefficient (LCC) [17] or hubs [23]. Nevertheless, these approaches have the risk of losing useful information contained in the SICE matrices. This paper aims to address the high dimensionality issue of these SICE matrices by extracting compact representation for classification.

As an inverse covariance matrix, an SICE matrix is symmetric positive definite (SPD). This inherent property restricts SICE matrices to a lower dimensional Riemannian manifold rather than the full $d \times d$ dimensional Euclidean space. In medical image analysis, the concept of Riemannian manifold has been widely used for DTI analysis [24], shape statistics [25], and functional-connectivity detection [6]. Moreover, considering the fact that brain connectivity patterns are specific and generally similar across different subjects, the SICE matrices representing the brain connectivity should concentrate on an even smaller subset of this manifold. In other words, the intrinsic degree of freedom of these SICE matrices shall be much lower than the apparent dimensions of $d \times d$. These two factors motivate us to seek a compact representation that better reflects the underlying distribution of the SICE matrices.

Principal component analysis (PCA), the commonly used unsupervised dimensionality reduction method, is a natural option for this task. However, a linear PCA is not expected to work well for manifold-constrained SICE matrices. Recently, advances have been made on measuring the similarity of SPD matrices considering the underlying manifold that they reside. In particular, a set of SPD kernels, e.g., Stein kernel [26] and Log-Euclidean kernel [27], have been proposed with promising applications [28], [29]. These kernels implicitly embed the Riemannian manifold of SPD matrices to a kernel-induced feature space $\mathcal{F}$. They offer better measure than their counterparts in Euclidean spaces and require less computation than Riemannian metric, as detailed in [26]. In this paper, we take advantage of these kernels to conduct a SPD-kernel-based PCA. This provides two advantages: 1) It produces a compact representation that can mitigate the curse of dimensionality and, thus, improves classification. 2) The extracted leading eigenvectors in $\mathcal{F}$ can reveal the intrinsic structure of the SICE matrices and, hence, assist brain network analysis.

While our approach introduced above could significantly improve the classification accuracy, another problem arises: how to interpret the obtained compact representation anatomically, or more specifically, can we visualize the principal connectivity components identified by a SPD-kernel PCA? This is important in neuroimage analysis, as it could possibly help to reveal the disease mechanisms behind. Since SPD-kernel PCA is implicitly carried out in the kernel-induced feature space $\mathcal{F}$, the ex-

tracted eigenvectors in $\mathcal{F}$ are not explicitly known and, therefore, cannot be readily used for anatomical analysis. A kernel preimage method has to be employed to recover these eigenvectors in the original input space. However, estimating the preimages of an object in $\mathcal{F}$ is challenging. Existing preimage methods [30], [31] require the knowledge of an explicit distance mapping between an input space and the feature space $\mathcal{F}$. Unfortunately, such an explicit distance mapping is intractable for SPD kernels and, thus, the existing preimage methods cannot be applied to our case. To solve this problem, we further propose a novel preimage method for the SPD kernels and use it to gain insight into SICE-based brain network analysis.

To verify our approach, we conduct an extensive experimental study on both synthetic dataset and rs-fMRI data from the benchmark dataset ADNI.² As will be seen, the results well demonstrate the effectiveness and advantages of our method. Specifically, the proposed compact representation obtained via the SPD-kernel PCA achieves superior classification performance to that from linear PCA and the graphical feature LCC. Also, the proposed preimage method can effectively recover in the original input space the principal connectivity components identified in a feature space and enables the visualization and anatomical analysis of these components.

In addition, we would like to point out that besides SICE matrices, the proposed method can be seamlessly applied to the correlation matrices previously mentioned, because they are also SPD. We focus on SICE matrices in this paper because SICE matrices model the partial correlations, which enjoy theoretical advantages, and generally admit more stable connectivity in comparison with correlation [32].

This paper is a significant extension of our previous work reported in a workshop paper [33]. The extension is made in three aspects: 1) More SPD kernels are investigated in this version. As demonstrated, different SPD kernels consistently achieve superior classification performance, which indicates the generality of the proposed method; 2) new experiments are conducted on a specifically designed synthetic data to show the characteristics of the proposed preimage method and its effectiveness; 3) in addition to the k -nearest neighbor (k -NN) classifier, this version includes support vector machines (SVM) as a classifier to evaluate the classification performance.

The rest of the paper is organized as follows. Section II reviews the SICE algorithm and the manifold structure of SPD matrices. Section III details the proposed SPD-kernel PCA and the preimage method. Section IV presents the experimental results on synthetic and real rs-fMRI datasets, and finally, Section V concludes this paper.

II. RELATED WORK

A. Constructing Brain Network Using SICE

Let $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M\}$ be a time series of length M , where $\mathbf{x}_i$ is a d -dimensional vector, corresponding to an observation of d brain nodes. Following the literature of SICE [19], [21], $\mathbf{x}_i$ is

¹To be precise, the dimensionality of the vector is $\frac{d(d-1)}{2}$ because the SICE matrix is symmetric and its diagonal entries are not used.

²<http://adni.loni.usc.edu>.

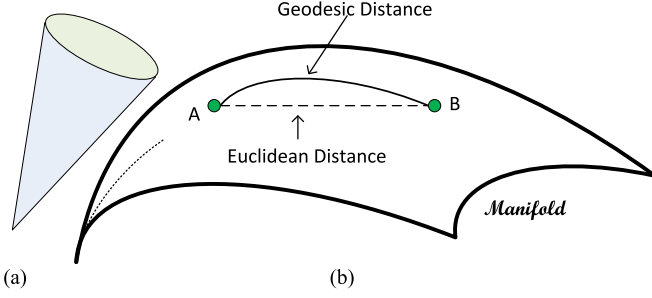


Fig. 1. Illustration of the Riemannian manifold of SPD matrices. (a) Sym_d^+ forms a closed, self-dual convex cone, which is a Riemannian manifold in the Euclidean space $\mathbb{R}^{d \times d}$ [26]. (b) To measure the distance between two SICE matrices A and B, Euclidean distance is not accurate since it does not consider the special geometry of the manifold structure. Instead, geodesic distance, which is defined as the shortest curve connecting A and B on the manifold, is more accurate.

assumed to follow a Gaussian distribution $\mathcal{N}(\mu, \Sigma)$. Each off-diagonal entry of Σ^{-1} indicates the partial correlation between two nodes by eliminating the effect of all other nodes. Σ_{ij}^{-1} will be zero if nodes i and j are independent of each other when conditioned on the other nodes. In this sense, Σ_{ij}^{-1} can be interpreted as the existence and strength of the connectivity between nodes i and j . The estimation of $\mathbf{S} = \Sigma^{-1}$ can be obtained by maximizing the penalized log-likelihood over positive definite matrix $\mathbf{S}$ ($\mathbf{S} \succ 0$) [19], [21]:

$$\mathbf{S}^* = \arg \max_{\mathbf{S} \succ 0} \log(\det(\mathbf{S})) - \text{tr}(\mathbf{C}\mathbf{S}) - \lambda \|\mathbf{S}\|_1 \quad (1)$$

where $\mathbf{C}$ is the sample-based covariance matrix; $\det(\cdot)$, $\text{tr}(\cdot)$, and $\|\cdot\|_1$ denote the determinant, trace, and the sum of the absolute values of the entries of a matrix. $\|\mathbf{S}\|_1$ imposes sparsity on $\mathbf{S}$ to achieve more reliable estimation by considering the fact that a brain region often has limited direct connections with other brain regions in neurological activities. The tradeoff between the degree of sparsity and the log-likelihood estimation of $\mathbf{S}$ is controlled by the regularization parameter λ . Larger λ makes $\mathbf{S}^*$ more sparse. The maximization problem in (1) can be efficiently solved by the off-the-shelf packages, such as SLEP [34].

B. SPD Matrices

The resulting SICE matrix $\mathbf{S}^*$ obtained by (1) is SPD since it is an estimation of inverse covariance matrix. Let Sym_d^+ denotes the $d \times d$ SPD matrices set: $\text{Sym}_d^+ = \{\mathbf{A} | \mathbf{A} = \mathbf{A}^T \ \forall \mathbf{x} \in \mathbb{R}^d, \mathbf{x} \neq \mathbf{0}, \mathbf{x}^T \mathbf{A} \mathbf{x} > 0\}$.

As illustrated in Fig. 1(a), Sym_d^+ forms a closed self-dual convex cone, which is a Riemannian manifold in the Euclidean space $\mathbb{R}^{d \times d}$ [26]. To effectively measure the similarity between two SICE matrices, as in Fig. 1(b), methods that respect the geodesic distance rather than Euclidean distance should be used [27]. To directly measure the geodesic distance for SPD matrices on the manifold, affine-invariant Riemannian metrics (AIRMs) were proposed in [35] and [24]. However, there are two issues: 1) The computational cost of AIRMs is high because it intensively uses matrix inverse, square roots, and logarithms [27], [28]; 2) more importantly, the linear algorithms, e.g., SVM, that

are developed in Euclidean spaces cannot be directly applied to SPD matrices lying on a manifold [29]. To address these issues, kernel method [26], [27], has been adopted to measure the similarity between SPD matrices. It measures the similarity by implicitly mapping the Riemannian manifold of SPD matrices onto a high-dimensional kernel-induced feature space $\mathcal{F}$, where linear algorithms can be generalized. The manifold structure is well incorporated in the mapping by utilizing distance functions that are specially designed for SPD matrices. Also, kernel methods are often computationally more efficient than AIRMs because the intensive use of matrix inverse, square roots, and logarithms in AIRMs can be avoided or reduced [27].

III. PROPOSED METHOD

A. SICE Representation Using SPD-Kernel Based PCA

In spite of individual variation, human brains do share common specific connectivity patterns across different subjects. Therefore, the SICE matrices used to represent brain networks shall have similar structures across subjects. This makes them be further restricted into a small subset of the Riemannian manifold of SPD matrices, with a limited degree of freedom. Inspired by this observation, we aim to extract a compact representation of these SICE matrices for better classification and analysis. PCA is a commonly used technique to generate a compact representation of data by exploring a subspace that can best represent the data. Therefore, PCA is a natural choice for our task. However, linear PCA is not expected to work well for the SICE matrices because it does not consider the manifold structure. Consequently, we adopt kernel PCA [37] and integrate SPD kernels for similarity measure. This effectively accounts for the manifold structure of SICE matrices when exploring the subspace of the data. Our method is elaborated as follows.

The SICE method is applied to N subjects to obtain a training set $\{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_N\} \subset \text{Sym}_d^+$, where $\mathbf{S}_i$ is the SICE matrix for the i th subject. We define the kernel mapping $\Phi(\cdot): \text{Sym}_d^+ \mapsto \mathcal{F}$, which cannot be explicitly solved but implicitly induced by a given SPD kernel. As an extension of PCA, kernel PCA generalizes linear PCA to a kernel-induced feature space $\mathcal{F}$. For the self-containedness of this paper, we briefly describe Kernel PCA as follows and the details can be found in [37]. Without loss of generality, it is assumed that $\Phi(\mathbf{S}_i)$ is centered, i.e., $\sum_{i=1}^N \Phi(\mathbf{S}_i) = \mathbf{0}$, and as in [37], this can be easily achieved by simple computation with kernel matrix. Then, a $N \times N$ kernel matrix $\mathbf{K}$ can be obtained with each entry $\mathbf{K}_{ij} = \langle \Phi(\mathbf{S}_i), \Phi(\mathbf{S}_j) \rangle = k(\mathbf{S}_i, \mathbf{S}_j)$. Kernel PCA first performs the eigen decomposition on the kernel matrix: $\mathbf{K} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$. The i th column of $\mathbf{U}$, denoted by $\mathbf{u}_i$, corresponds to the i th eigenvector, and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N)$, where λ_i corresponds to the i th eigenvalue in a descending order. Let Σ_Φ denote the covariance matrix computed by $\{\Phi(\mathbf{S}_i)\}_{i=1}^N$ in $\mathcal{F}$. The i th eigenvector of Σ_Φ can be expressed as

$$\mathbf{v}_i = \frac{1}{\sqrt{\lambda_i}} \Phi \mathbf{u}_i \quad (2)$$

where $\Phi = [\Phi(\mathbf{S}_1), \Phi(\mathbf{S}_2), \dots, \Phi(\mathbf{S}_N)]$. Analogous to linear PCA, for a given SICE matrix $\mathbf{S}$, $\Phi(\mathbf{S})$ can then be projected onto

the top m eigenvectors to obtain an m -dimensional principal component vector

$$\alpha = \mathbf{V}_m^\top \Phi(\mathbf{S})$$

where $\mathbf{V}_m = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m]$. Note that the i th component of α , denoted by α_i , is $\mathbf{v}_i^\top \Phi(\mathbf{S})$. With the kernel trick, it can be computed as

$$\alpha_i = \mathbf{v}_i^\top \Phi(\mathbf{S}) = \frac{1}{\sqrt{\lambda_i}} \mathbf{u}_i^\top \Phi^\top \Phi(\mathbf{S}) = \frac{1}{\sqrt{\lambda_i}} \mathbf{u}_i^\top \mathbf{k}_S \quad (3)$$

where $\mathbf{k}_S = [k(\mathbf{S}, \mathbf{S}_1), k(\mathbf{S}, \mathbf{S}_2), \dots, k(\mathbf{S}, \mathbf{S}_N)]^\top$. Once α is obtained as a new representation for each SICE matrix, an SVM or k -NN classifier can be trained on α with class labels.

In this paper, we study four commonly used SPD kernels, namely, Cholesky kernel (CHK) [29], power Euclidean kernel (PEK) [29], Log-Euclidean kernel (LEK) [27], and Stein kernel (SK) [26]. The four kernels are all in a form of

$$k(\mathbf{S}_i, \mathbf{S}_j) = \exp(-\theta \cdot d^2(\mathbf{S}_i, \mathbf{S}_j)) \quad (4)$$

where $d(\cdot, \cdot)$ is a kind of distance between two SPD matrices. Different definitions of $d(\cdot, \cdot)$ lead to different kernels, and the distance functions in the four kernels are Cholesky distance [36], power Euclidean distance [36], Log-Euclidean distance [27], and root Stein divergence [26], respectively. They are introduced as follows.

1) *Cholesky Distance*: Cholesky distance measures the difference between $\mathbf{S}_i$ and $\mathbf{S}_j$ by

$$d(\mathbf{S}_i, \mathbf{S}_j) = \|(\mathbf{S}_i) - (\mathbf{S}_j)\|_F \quad (5)$$

where $(\mathbf{S})$ is a lower triangular matrix with positive diagonal entries obtained by the Cholesky decomposition of $\mathbf{S}$, that is, $\mathbf{S} = (\mathbf{S})(\mathbf{S})^\top$ and $\|\cdot\|_F$ denotes the Frobenius matrix norm.

2) *Power Euclidean Distance*: Power Euclidean distance between $\mathbf{S}_i$ and $\mathbf{S}_j$ is given by

$$d(\mathbf{S}_i, \mathbf{S}_j) = \frac{1}{p} \|\mathbf{S}_i^p - \mathbf{S}_j^p\|_F \quad (6)$$

where $p \in \mathbb{R}$. Note that $\mathbf{S}$, as a SPD matrix, can be eigen decomposed as $\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, and $\mathbf{S}^p$ can be easily computed by $\mathbf{S}^p = \mathbf{U}\mathbf{\Lambda}^p\mathbf{U}^\top$. In this paper, we set $p = 0.5$ since it achieves the best result in the literature [29], [36] and our experiments.

3) *Log-Euclidean Distance*: Log-Euclidean distance is defined as

$$d(\mathbf{S}_i, \mathbf{S}_j) = \|\log(\mathbf{S}_i) - \log(\mathbf{S}_j)\|_F \quad (7)$$

where $\log(\mathbf{S}) = \mathbf{U}\log(\mathbf{\Lambda})\mathbf{U}^\top$ and $\log(\mathbf{\Lambda})$ applies logarithm to each diagonal element of $\mathbf{\Lambda}$ to obtain a new diagonal matrix.

4) *Root Stein Divergence*: Root Stein divergence is the square root of Stein divergence, which is defined as

$$d(\mathbf{S}_i, \mathbf{S}_j) = \left[\log \left(\det \left(\frac{\mathbf{S}_i + \mathbf{S}_j}{2} \right) \right) - \frac{1}{2} \log (\det(\mathbf{S}_i \mathbf{S}_j)) \right]^{\frac{1}{2}}. \quad (8)$$

With root Stein divergence as the distance function, the θ in $k(\mathbf{S}_i, \mathbf{S}_j) = \exp(-\theta \cdot d^2(\mathbf{S}_i, \mathbf{S}_j))$ is a positive scalar within the range of $\{\frac{1}{2}, \frac{2}{2}, \frac{3}{2}, \dots, \frac{(d-1)}{2}\} \cup (\frac{(d-1)}{2}, +\infty)$ to guarantee SK to be a Mercer kernel [26].

The four distance functions and the corresponding kernels are summarized in Table I. They will be applied to SPD-kernel PCA to produce the principal component vector α .

B. Preimage Estimation

As will be shown in the experimental study, the principal components α extracted by the above SPD-kernel PCA offer promising classification performance. Note that α is fundamentally determined by the m leading eigenvectors $\mathbf{v}_1, \dots, \mathbf{v}_m$, which capture the underlying structure of SICE matrices and can be deemed as the building blocks of this representation of brain connectivity. Therefore, analyzing these eigenvectors is important for the understanding and interpretation of the obtained principal connectivity patterns. However, the eigenvectors are derived in $\mathcal{F}$ via the implicit kernel mapping $\Phi(\cdot)$ and, thus, are not readily used for analysis in the input space Sym_d^+ . To tackle this issue, we aim to develop a method that can project a data-point in the subspace spanned by the m leading eigenvectors in $\mathcal{F}$ back to the input space. This will allow the visualization of the principal connectivity patterns in the input space for interpretation. This is known as the “preimage” problem of kernel methods in the literature [30], [31], [38]. Unfortunately, existing preimage methods, such as those in [30] and [31], cannot be applied to our case, because they require an explicit mapping between the Euclidean distance in $\mathcal{F}$ and the Euclidean distance in the input space, which is unavailable when the SPD kernels are used. In the following, we develop a novel preimage method for the SPD kernels to address this issue.

Let $\Phi_m(\mathbf{S})$ denote the projection of $\Phi(\mathbf{S})$ into the subspace spanned by the m leading eigenvectors in $\mathcal{F}$, that is

$$\begin{aligned} \Phi_m(\mathbf{S}) &= \sum_{i=1}^m \alpha_i \mathbf{v}_i = \sum_{i=1}^m \frac{1}{\sqrt{\lambda_i}} \mathbf{u}_i^\top \mathbf{k}_S \cdot \frac{1}{\sqrt{\lambda_i}} \Phi \mathbf{u}_i \\ &= \sum_{i=1}^m \left[\mathbf{k}_S^\top \frac{1}{\lambda_i} \mathbf{u}_i \cdot \mathbf{u}_i^\top \Phi^\top \right]^\top = \Phi \mathbf{M} \mathbf{k}_S \end{aligned} \quad (9)$$

where $\mathbf{M} = \sum_{i=1}^m \frac{1}{\lambda_i} \mathbf{u}_i \mathbf{u}_i^\top$ and recall $\Phi = [\Phi(\mathbf{S}_1), \Phi(\mathbf{S}_2), \dots, \Phi(\mathbf{S}_N)]$ and $\mathbf{k}_S = [k(\mathbf{S}, \mathbf{S}_1), k(\mathbf{S}, \mathbf{S}_2), \dots, k(\mathbf{S}, \mathbf{S}_N)]^\top$. Our aim is to find a preimage $\hat{\mathbf{S}}$ in the original input space (that is, Sym_d^+) which best satisfies $\Phi(\hat{\mathbf{S}}) = \Phi_m(\mathbf{S})$. Considering the fact that Riemannian manifold is locally homeomorphic with a Euclidean space [39], we model $\hat{\mathbf{S}}$ by a linear combination³ of its neighboring SICE matrices in Sym_d^+ . Similar to the work in [30], we assume that if $\mathbf{S}_i$ and $\mathbf{S}_j$ are close in Sym_d^+ , then $\Phi(\mathbf{S}_i)$ and $\Phi(\mathbf{S}_j)$ shall also be close in $\mathcal{F}$. With this assumption, we can obtain the neighbors of $\hat{\mathbf{S}}$ in Sym_d^+ by finding the neighbors of $\Phi_m(\mathbf{S})$ in $\mathcal{F}$.

Specifically, $\hat{\mathbf{S}}$ is estimated as follows. First, we find a set of NN $\Omega = \{\mathbf{S}_j\}_{j=1}^L$ for $\hat{\mathbf{S}}$ from a training set $\{\mathbf{S}_i\}_{i=1}^N$ by sorting

³Using linear combination of neighbors may restrict the search space of preimage and could affect the reconstruction accuracy. Here, we use it for three reasons: i) our experiment on synthetic data (with ground truth) has demonstrated good reconstruction result; ii) using linear combination can significantly simplify the optimization problem of preimage estimation; iii) by using linear combination of neighbors, we can better enforce the constructed preimage to follow the underlying distribution of training samples.

TABLE I
DEFINITION AND PROPERTIES OF DISTANCE FUNCTIONS ON Sym_d^+

Distance name	Formula	Range of θ in $k = \exp(-\theta \cdot d^2)$ to define a valid kernel	Kernel abbr. in the paper
Cholesky [36]	$d = \ (\mathbf{S}_1) - (\mathbf{S}_2)\ _F$	$\mathbb{R}^+$	CHK
Power-Euclidean [36]	$d = \frac{1}{p} \ \mathbf{S}_1^p - \mathbf{S}_2^p\ _F$	$\mathbb{R}^+$	PEK
Log-Euclidean [27]	$d = \ \log(\mathbf{S}_1) - \log(\mathbf{S}_2)\ _F$	$\mathbb{R}^+$	LEK
S-Divergence root [26]	$d = \left[\log \left(\det \left(\frac{\mathbf{S}_1 + \mathbf{S}_2}{2} \right) \right) - \frac{1}{2} \log (\det(\mathbf{S}_1 \mathbf{S}_2)) \right]^{\frac{1}{2}}$	$\theta \in \left\{ \frac{1}{2}, \frac{2}{2}, \frac{3}{2}, \dots, \frac{(d-1)}{2} \right\} \cup \left(\frac{(d-1)}{2}, +\infty \right)$	SK

the following distance:

$$\begin{aligned}
d^2(\Phi_m(\mathbf{S}), \Phi(\mathbf{S}_i)) & \\
&\triangleq \|\Phi_m(\mathbf{S}) - \Phi(\mathbf{S}_i)\|^2 \\
&= \|\Phi_m(\mathbf{S})\|^2 + \|\Phi(\mathbf{S}_i)\|^2 - 2\Phi_m(\mathbf{S})^\top \Phi(\mathbf{S}_i) \\
&= \left(\sum_{i=1}^m \alpha_i \mathbf{v}_i \right)^\top \left(\sum_{i=1}^m \alpha_i \mathbf{v}_i \right) + k(\mathbf{S}_i, \mathbf{S}_i) \\
&\quad - 2(\Phi \mathbf{M} \mathbf{k}_S)^\top \Phi(\mathbf{S}_i) \\
&= \sum_{i=1}^m \alpha_i^2 + k(\mathbf{S}_i, \mathbf{S}_i) - 2\mathbf{k}_S^\top \mathbf{M} \Phi^\top \Phi(\mathbf{S}_i) \\
&\quad \text{[By applying Eq.(3)]} \\
&= (\mathbf{k}_S^\top - 2\mathbf{k}_{S_i}^\top) \mathbf{M} \mathbf{k}_S + k(\mathbf{S}_i, \mathbf{S}_i). \tag{10}
\end{aligned}$$

This distance can be easily computed because it is fully represented by the kernel functions.

Second, we model the preimage $\hat{\mathbf{S}}$ by a convex (linear) combination of its neighbors as

$$\hat{\mathbf{S}} = \sum_{j=1}^L w_j \mathbf{S}_j \tag{11}$$

where $\mathbf{S}_j \in \Omega$, $w_j \geq 0$, and $\sum_{j=1}^L w_j = 1$. This convex combination guarantees the SPD of $\hat{\mathbf{S}}$ and also makes it be effectively constrained by its L neighbors. Defining $\mathbf{w} = [w_1, w_2, \dots, w_L]^\top$, we seek the optimal $\mathbf{w}$ by solving

$$\mathbf{w}^* = \arg \min_{\mathbf{w} \geq 0; \mathbf{w}^\top \mathbf{1} = 1} d^2 \left(\Phi_m(\mathbf{S}), \Phi \left(\sum_{\mathbf{S}_j \in \Omega} w_j \mathbf{S}_j \right) \right) \tag{12}$$

where $d^2(\Phi_m(\mathbf{S}), \Phi(\sum_{\mathbf{S}_j \in \Omega} w_j \mathbf{S}_j)) = d^2(\Phi_m(\mathbf{S}), \Phi(\hat{\mathbf{S}})) = (\mathbf{k}_S^\top - 2\mathbf{k}_{\hat{\mathbf{S}}}^\top) \mathbf{M} \mathbf{k}_S + k(\hat{\mathbf{S}}, \hat{\mathbf{S}})$ by applying (10) and (11). This optimization problem can be efficiently solved using gradient descent based algorithms. Note that (12) can be used to compute the preimage of any datapoint $\Phi_m(\mathbf{S})$ in $\mathcal{F}$. In addition, when estimating the preimage of a specific eigenvector $\mathbf{v}_i$, we can simply set $\Phi_m(\mathbf{S})$ as $\mathbf{v}_i$ and solve the same optimization

problem in (12). In this case, the objective function reduces to

$$\begin{aligned}
d^2(\mathbf{v}_i, \Phi(\mathbf{S}_i)) &= \|\mathbf{v}_i - \Phi(\mathbf{S}_i)\|^2 \\
&= \|\mathbf{v}_i\|^2 + \|\Phi(\mathbf{S}_i)\|^2 - 2\mathbf{v}_i^\top \Phi(\mathbf{S}_i) \\
&= 1 + k(\mathbf{S}_i, \mathbf{S}_i) - 2 \left(\frac{1}{\sqrt{\lambda_i}} \Phi \mathbf{u}_i \right)^\top \Phi(\mathbf{S}_i) \\
&= 1 + k(\mathbf{S}_i, \mathbf{S}_i) - \frac{2}{\sqrt{\lambda_i}} \mathbf{u}_i^\top \mathbf{k}_{S_i}. \tag{13}
\end{aligned}$$

Algorithm 1 outlines the proposed preimage algorithm.

Algorithm 1: Preimage estimation for $\Phi_m(\mathbf{S})$ in $\mathcal{F}$

Input: A training set $\{\mathbf{S}_i\}_{i=1}^N$, test data $\mathbf{S}$, m ;

Output: Preimage $\hat{\mathbf{S}}$

- 1: Find a set of L neighbors $\Omega = \{\mathbf{S}_j\}_{j=1}^L$ for $\hat{\mathbf{S}}$ by sorting $d^2(\Phi_m(\mathbf{S}), \Phi(\mathbf{S}_i))$, $i = 1, \dots, N$, according to (10);
 - 2: Solve (14) to obtain $\mathbf{w}^*$:
 $\mathbf{w}^* = \arg \min_{\mathbf{w} \geq 0; \mathbf{w}^\top \mathbf{1} = 1} d^2(\Phi_m(\mathbf{S}), \Phi(\sum_{\mathbf{S}_j \in \Omega} w_j \mathbf{S}_j))$;
 - 3: **return** $\hat{\mathbf{S}} = \sum_{j=1}^L w_j \mathbf{S}_j$.
-

IV. EXPERIMENTAL STUDY

A. Data Preprocessing and Experimental Settings

Rs-fMRI data of 196 subjects were downloaded from the ADNI website⁴ in June 2013. Nine subjects were discarded due to the corruption of data, and the remaining 187 subjects were preprocessed for analysis. After removing subjects that had problems in the preprocessing steps, such as large head motion, 156 subjects were kept, including 26 AD, 44 early MCI, 38 late MCI, 38 NC, and ten significant memory concern labeled by ADNI. We used the 38 NC and the 44 early MCI in this paper because our focus in this paper is to identify MCI at very early stage, which is the most challenging and significant task in AD prediction. The IDs of the 82 (38 NC and 44 early MCI) subjects are provided in the supplementary material. The data are acquired on a 3-T (Philips) scanner with TR/TE set as 3000/30

⁴<http://adni.loni.usc.edu>.

TABLE II
CLASSIFICATION ACCURACY (IN %) BY DIRECTLY USING SICE/SLR MATRICES AS FEATURES

	Linear kernel (vectorized [16])		LEK (proposed)		SK (proposed)		CHK (proposed)		PEK (proposed)	
	k -NN	SVM	k -NN	SVM	k -NN	SVM	k -NN	SVM	k -NN	SVM
SLR [17]	53.7	52.4	NA	NA	NA	NA	NA	NA	NA	NA
SICE	57.3	57.3	61.0	61.0	63.4	64.6	61.0	62.2	61.0	65.9

ms and flip angle of 80° . Each series has 140 volumes, and each volume consists of 48 slices of image matrices with dimensions 64×64 with voxel size of $3.31 \times 3.31 \times 3.31 \text{ mm}^3$. The pre-processing is carried out using SPM8⁵ and DPARSFA [40]. The first ten volumes of each series are discarded for signal equilibrium. Slice timing, head motion correction, and MNI space normalization are performed. Participants with too much head motion are excluded. The normalized brain images are warped into automatic anatomical labeling (AAL) [41] atlas to obtain 90 ROIs as nodes. By following common practice [15]–[17], the ROI mean time series are extracted by averaging the time series from all voxels within each ROI and then bandpass filtered to obtain multiple subbands as in [17].

The functional connectivity networks of 82 participants are obtained by the SICE method using SLEP [34], with the sparsity levels of $\lambda = [0.1 : 0.1 : 0.9]$. For comparison, constrained sparse linear regression (SLR) [17] is also used to learn functional connectivity networks with the same setting. Functional connectivity networks constructed by SICE and SLR are called “SICE matrices” and “SLR matrices,” respectively. To make full use of the limited subjects, a leave-one-out procedure is used for training and test. That is, each sample is reserved for test in turn, while the remaining samples are used for training. Both SVM and k -NN are used as the classifier to compare the classification accuracy of different methods. The parameters used in the following classification tasks of this rs-fMRI dataset, including the sparsity level λ , the subband of the time series, the number of eigenvectors m , and the regularization parameter of SVM are tuned by fivefold cross validation on the training set. θ in all the four SPD kernels is empirically set as 0.5, and the k of k -NN is set as 7.

B. Experimental Result

The experiment consists of three parts: 1) Evaluating the classification performance when the original SICE or SLR matrices are used as the features; 2) evaluating the classification performance when the compact representation of SICE or SLR matrices is used as the features; 3) investigating the effectiveness of the proposed preimage method.

1) *Classification Using Original SICE or SLR Matrices:* By applying the SICE or SLR method to the rs-fMRI data, we can obtain the SICE or SLR matrices as the representation of brain networks. These matrices can be directly used as features to train a classifier. A straightforward way is to vectorize the matrices into high-dimensional vectors as features as in [16], which are then used to train a linear SVM or k -NN with linear kernel

as the similarity measure to search NN to perform classification. Note that linear kernel is Euclidean distance-based similarity measure. As shown in the second and third columns in Table II (labeled by “linear kernel”), this method produces poor classification performance (lower than 60%) on both SICE and SLR matrices, be it k -NN or linear SVM is used as the classifier. Specifically, it only achieves 53.7% for the k -NN classifier using SLR matrices. When SICE matrices are used, the classification performance is only 57.3% too. The result does not change much when a linear SVM is used. The poor classification performance of this method is largely due to two issues: 1) The vectorization ignores the underlying structure of SICE matrices, and the linear kernel in SVM and in the k -NN classifier cannot effectively evaluate their similarity and distance; and 2) the “small sample size” problem occurs because the dimensionality of the resulting feature vectors is high, while the training samples are limited.

In order to effectively consider the manifold geometry of SICE matrices, we employ the four aforementioned SPD kernels to evaluate the similarity between SICE matrices and adopt k -NN and SVM classifiers with these kernels to perform classification. As seen in the columns under “LEK,” “SK,” “CHK,” “PEK” in Table II, the classification accuracy with respect to each SPD kernel is above 60%, which clearly outperforms that of their linear counterparts. In particular, PEK obtains 65.9% with SVM as the classifier, achieving an improvement of 8.6 percentage points over linear SVM. This well verifies the importance of considering the manifold structure of SICE matrices for the classification. Note that because SLR matrices are not necessarily SPD, the SPD kernels cannot be applied. Therefore, no classification result is reported in the row of “SLR” in Table II.

2) *Classification Using the Compact Representation:* In this experiment, we compare the classification performance of the compact representation obtained by the proposed SPD-kernel PCA, linear PCA, and the method computing LCC [17]. LCC, as a measure of local neighborhood connectivity for a node, is defined as the ratio of the number of existing edges between the neighbors of the node and the number of potential connections between these neighbors [42]. In this case, LCC can map a network, represented by a $d \times d$ adjacency matrix, to a d -dimensional vector, where d is the number of nodes in the network.

Table III shows the classification results when using the compact representation of SICE or SLR matrices using k -NN with Euclidean distance and linear kernel SVM. LCC achieves 65.9% for both SICE and SLR matrices with k -NN as the classifier. It is better than the result (53.7% and 57.3% in the second

⁵<http://www.fil.ion.ucl.ac.uk/spm/software/>.

TABLE III
CLASSIFICATION ACCURACY (IN %) OF COMPACT REPRESENTATION ON SICE/SLR MATRICES

	LCC		Linear PCA		LEK PCA (proposed)		SK PCA (proposed)		CHK PCA (proposed)		PEK PCA (proposed)	
	k -NN	SVM	k -NN	SVM	k -NN	SVM	k -NN	SVM	k -NN	SVM	k -NN	SVM
SLR [17]	65.9	64.6	67.1	65.9	NA	NA	NA	NA	NA	NA	NA	NA
SICE	65.9	63.4	67.1	68.3	69.5	69.5	72	73.2	68.3	70.7	72	73.2

TABLE IV
CLASSIFICATION ACCURACY (IN %) BY USING ORIGINAL SICE/SLR MATRICES AND PREIMAGES OF $\Phi_m(\mathbf{S})$ WITH k -NN

	SLR [17]	SICE	Preimages of SICE (LEK, proposed)	Preimages of SICE (SK, proposed)	Preimages of SICE (CHK, proposed)	Preimages of SICE (PEK, proposed)
Linear kernel	53.7	57.3	68.3	67.1	63.4	63.4
LCC	65.9	65.9	67.1	67.1	64.6	68.3

column of Table II) of directly using the original matrices, and is comparable to the result (65.9%) of applying PEK-SVM, the best one obtained in Table II. When linear PCA is applied to the vectorized SICE or SLR matrices to extract the top m principal components as features, the classification accuracy increases to 67.1% for both SICE and SLR matrices. This performance is better than LCC and all the methods in Table II. Such a result indicates the power of compact representation and also preliminarily justifies our idea of exploring the lower intrinsic dimensions of the SICE matrices. By further taking the SPD property into account and using the proposed SPD-kernel PCA to extract the compact representation, the classification accuracy is significantly boosted up to 73.2% for both SK-PCA and PEK-PCA, with SVM as the classifier. This achieves an improvement of 4.9 percentage points (73.2% versus 68.3%) over linear PCA and 7.3 percentage points (73.2% versus 65.9%) over LCC. These results well demonstrate that: 1) The obtained compact representation can effectively improve the generalization of the classifier in the case of limited training samples. 2) It is important to consider the manifold property of SICE matrices in order to obtain better compact representation. Cross referencing the SICE results in Tables II and III, SPD-kernel PCA achieves the best classification performance, i.e., 73.2%, obtaining an improvement of 15.9 percentage points over the linear kernel method (57.3%, in Table II).

3) *Investigating the Proposed Preimage Method:* The two goals of the preimage method, which is shown in Algorithm 1, is to estimate the preimage of 1) $\Phi_m(\mathbf{S})$, which is the projection of $\Phi(\mathbf{S})$ into the m leading eigenvectors in $\mathcal{F}$ and 2) one single eigenvector $\mathbf{v}_i$ of SPD-kernel PCA in $\mathcal{F}$.

The motivation of the first goal to recover the preimage of $\Phi_m(\mathbf{S})$ is inspired by the property of PCA. It is known that projecting data into the m leading eigenvectors discards the minor components, which often correspond to the data noise. Therefore, when an SICE matrix $\mathbf{S}$ is contaminated by noise (and it makes $\Phi(\mathbf{S})$ noisy), $\Phi_m(\mathbf{S})$ can be regarded as a “denoised” version of $\Phi(\mathbf{S})$. As a result, if the proposed preimage method

really works, the recovered preimage shall be closer to the true inverse covariance matrix than $\mathbf{S}$ is. In the literature, such a property has been extensively used for data and image denoising [43].

The proposed preimage method is performed on the real rs-fMRI data. Here, we aim to investigate if the preimages can boost the classification performance in comparison with the original SICE matrices based on the assumption that the preimage of $\Phi_m(\mathbf{S})$ can bring some kind of denoising effect. We first estimate the preimages of $\Phi_m(\mathbf{S}_i)$, $\mathbf{S}_i \in \{\mathbf{S}_i\}_{i=1}^{82}$ and redo classification using two methods: 1) *Linear kernel method.* As what we did in the second column of Table II, k -NN classifier is directly applied to the obtained preimages with linear kernel as the similarity measure; ii) *LCC method.* As what we did in the second column of Table III, LCC is extracted as a feature from the obtained preimages and apply k -NN classifier to LCC with Euclidean distance. The number of leading eigenvectors m is selected by cross validation from the range of $[1 : 5 : 80]$ on the training set while the number of neighbors L is empirically set as 20. In our experiment, we observe that 1) A larger L will make the optimization significantly more time consuming, while the performance of the method remains similar; 2) the selected value of m is usually in the range of $[15-35]$.

Table IV shows the classification result on the preimages of $\Phi_m(\mathbf{S}_i)$, $\mathbf{S}_i \in \{\mathbf{S}_i\}_{i=1}^{82}$, obtained on the real rs-fMRI data. The classification performance with the preimages when SK, LEK, and PEK are used can consistently outperform the classification performance with original SICE or SLR matrices using either linear kernel method or LCC method. Specifically, the performance of linear kernel method on SICE matrices is boosted to 68.3% (the fourth column, with preimages when LEK is used) from 57.3% (the third column). We believe that the improvement is due to that, by estimating the preimages of $\Phi_m(\mathbf{S}_i)$ in $\mathcal{F}$, the resulting matrices are more reliable than the original SICE matrices.

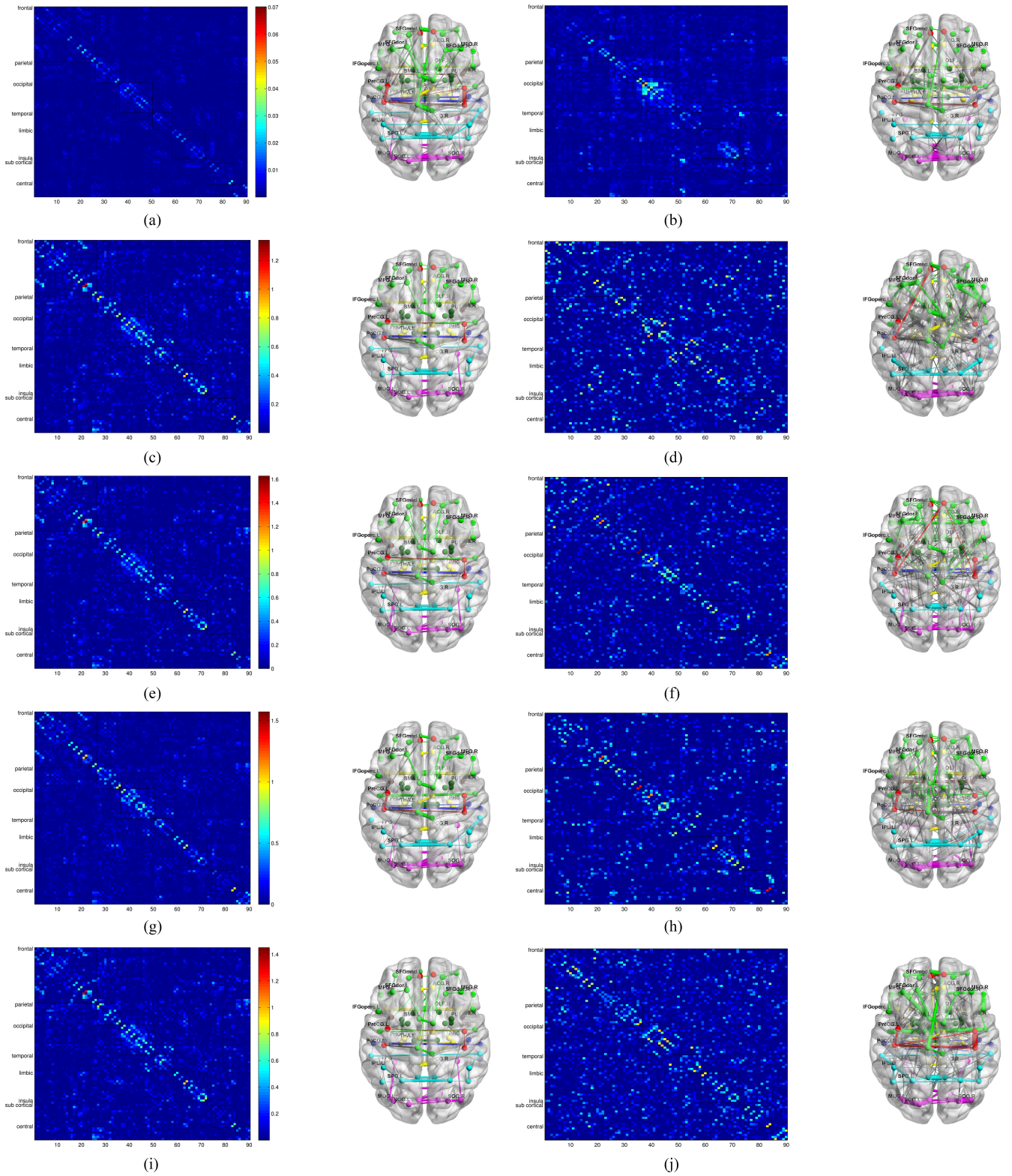


Fig. 2. Top two eigenvectors extracted in linear PCA (The first row), CHK PCA (The second row), PEK PCA (The third row), LEK PCA (The fourth row), and LEK PCA (The fifth row). (a) Eigenvector 1 of linear PCA. (b) Eigenvector 2 of linear PCA. (c) Eigenvector 1 of CHK PCA. (d) Eigenvector 2 of CHK PCA. (e) Eigenvector 1 of PEK PCA. (f) Eigenvector 2 of PEK PCA. (g) Eigenvector 1 of LEK PCA. (h) Eigenvector 2 of LEK PCA. (i) Eigenvector 1 of SK PCA. (j) Eigenvector 2 of SK PCA.

TABLE V
NAME AND LOBE OF EACH ROI IN FIG. 2

Lobe	ROI index	ROI name	ROI index	ROI name
frontal	1	Frontal_Sup_L	2	Frontal_Sup_R
	3	Frontal_Sup_Orb_L	4	Frontal_Sup_Orb_R
	5	Frontal_Mid_L	6	Frontal_Mid_R
	7	Frontal_Mid_Orb_L	8	Frontal_Mid_Orb_R
	9	Frontal_Inf_Oper_L	10	Frontal_Inf_Oper_R
	11	Frontal_Inf_Tri_L	12	Frontal_Inf_Tri_R
	13	Frontal_Inf_Orb_L	14	Frontal_Inf_Orb_R
	15	Supp_Motor_Area_L	16	Supp_Motor_Area_R
	17	Olfactory_L	18	Olfactory_R
	19	Frontal_Sup_Medial_L	20	Frontal_Sup_Medial_R
	21	Frontal_Mid_Orb_L	22	Frontal_Mid_Orb_R
	23	Rectus_L	24	Rectus_R
	25	Paracentral_Lobule_L	26	Paracentral_Lobule_R
parietal	27	Parietal_Sup_L	28	Parietal_Sup_R
	29	Parietal_Inf_L	30	Parietal_Inf_R
	31	SupraMarginal_L	32	SupraMarginal_R
	33	Angular_L	34	Angular_R
	35	Precuneus_L	36	Precuneus_R
occipital	37	Calcarine_L	38	Calcarine_R
	39	Cuneus_L	40	Cuneus_R
	41	Lingual_L	42	Lingual_R
	43	Occipital_Sup_L	44	Occipital_Sup_R
	45	Occipital_Mid_L	46	Occipital_Mid_R
temporal	47	Occipital_Inf_L	48	Occipital_Inf_R
	49	Fusiform_L	50	Fusiform_R
	51	Heschl_L	52	Heschl_R
	53	Temporal_Sup_L	54	Temporal_Sup_R
	55	Temporal_Mid_L	56	Temporal_Mid_R
limbic	57	Temporal_Inf_L	58	Temporal_Inf_R
	59	Cingulum_Ant_L	60	Cingulum_Ant_R
	61	Cingulum_Mid_L	62	Cingulum_Mid_R
	63	Cingulum_Post_L	64	Cingulum_Post_R
	65	Hippocampus_L	66	Hippocampus_R
insula	67	ParaHippocampal_L	68	ParaHippocampal_R
	69	Temporal_Pole_Sup_L	70	Temporal_Pole_Sup_R
	71	Temporal_Pole_Mid_L	72	Temporal_Pole_Mid_R
	73	Insula_L	74	Insula_R
sub cortical	75	Amygdala_L	76	Amygdala_R
	77	Caudate_L	78	Caudate_R
	79	Putamen_L	80	Putamen_R
	81	Pallidum_L	82	Pallidum_R
	83	Thalamus_L	84	Thalamus_R
central	85	Precentral_L	86	Precentral_R
	87	Rolandic_Oper_L	88	Rolandic_Oper_R
	89	Postcentral_L	90	Postcentral_R

Recall that the leading eigenvectors $\mathbf{v}_i$ in $\mathcal{F}$ capture the underlying structure of SICE matrices and can be deemed as the building blocks of the representation for brain connectivity. Thus, we estimate the preimage of top eigenvectors $\mathbf{v}_i$ in $\mathcal{F}$ for anatomical analysis. In this experiment, the preimages of the top two eigenvectors, which pose the most significant variance of SICE matrices in $\mathcal{F}$, are visualized in Fig. 2. The lobe, index, and name of each ROI in AAL [41] atlas are listed in Table V. We observe that: 1) compared with the eigenvectors in linear PCA, the eigenvectors obtained in the SPD-kernel PCA capture richer connection structures. Specifically, as seen from Fig. 2(a), the first eigenvector in linear PCA only presents very weak intralobe connections in frontal and occipital lobes. In contrast, the first eigenvector obtained by each of the SPD-kernel PCA well captures the intralobe connections in all the lobes. Especially, as indicated in Fig. 2(c), (e), (g), and (i), there are strong connections

at orbitofrontal cortex (ROI index: 8, 19–22), rectus gyrus (23, 24), occipital gyrus (43–48), temporal gyri (53–58), hippocampus (65–66), and temporal pole (69–72). Respecting the second eigenvector, the eigenvectors obtained by the SPD-kernels PCA [see Fig. 2(d), (f), (h), and (j)] incorporate both intralobe and inter-lobe connections while the eigenvector in linear PCA [see Fig. 2(b)] mainly captures only intraobe connections in occipital lobe; 2) the preimages obtained when different SPD kernels are used, as seen in Fig. 2(c)–(j), are very similar with each other with slight variation. This is expected since they all reflect the underlying manifold structure of SICE matrices. Further exploration of their clinical interpretation will be included in our future work.

C. Evaluation of the Preimage Method Using Synthetic Data

To further investigate the efficacy of the proposed preimage method, a synthetic dataset is specially designed for evaluation. The synthetic dataset is used for two purposes: 1) It allows the comparison between the recovered preimage of $\Phi_m(\cdot)$ and the ground truth inverse covariance matrix, which is not available for real rs-fMRI data; ii) by adjusting the parameters used to generate the synthetic data, the behavior of the proposed preimage method can be demonstrated. The synthetic data are generated by mimicking the following data generation process in practice.

- 1) Generate a set of 82 covariance matrices of the size of 90×90 , by sampling a Wishart distribution⁶ [44]. Let Σ_i ($i = 1, \dots, 82$) be the i th covariance matrix and its inverse Σ_i^{-1} will be used as a ground truth inverse covariance matrix;
- 2) A set of 130 vectors are randomly sampled from each normal distribution $\mathcal{N}(\mathbf{0}, \Sigma_i)$, where $i = 1, \dots, 82$;
- 3) Gaussian noise is added to each set of 130 vectors to simulate that the data are contaminated. The noise level is denoted by δ ;
- 4) A sample-based covariance matrix $\mathbf{C}$ is computed by using each set of the (noisy) 130 vectors and 82 covariance matrices are obtained in total. They are denoted as $\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_{82}$;
- 5) Apply the SICE method to each $\mathbf{C}_i$ to obtain the SICE matrix, and they are collectively denoted by $\{\mathbf{S}_i\}_{i=1}^{82}$. These SICE matrices form the synthetic dataset. Note that they are affected by the noise added in Step 3.

From the synthetic dataset $\{\mathbf{S}_i\}_{i=1}^{82}$, every $\mathbf{S}_i$ is selected in turn as the test data and the remainder are used as the training set. Algorithm 1 is then applied to estimate the preimage $\hat{\mathbf{S}}_i$ for $\Phi_m(\mathbf{S}_i)$. Then, the recovered preimage $\hat{\mathbf{S}}_i$ and the test data $\mathbf{S}_i$ are compared, respectively, with the ground truth inverse covariance matrix Σ_i^{-1} prepared in Step 1. This is to see whether $\hat{\mathbf{S}}_i$ is really closer to Σ_i^{-1} than $\mathbf{S}_i$. Following the literature [45], we use Kullback–Liebler (KL) divergence to compare $\hat{\mathbf{S}}_i$ (or $\mathbf{S}_i$) with Σ_i^{-1} . Given a pair of SPD matrices Σ_1 and Σ_2 , KL divergence measures the similarity of two Gaussians $\mathcal{N}(\mu_1, \Sigma_1)$ and $\mathcal{N}(\mu_2, \Sigma_2)$. It can

⁶The Wishart distribution is used as $\Sigma_i \sim \mathcal{W}_{90}(\Sigma_0, n)$, where $\Sigma_0 \in \text{Sym}_{90}^+$ is set as a block-wise covariance matrix for a better illustration of the result, and n is the degree of freedom set as 1000.

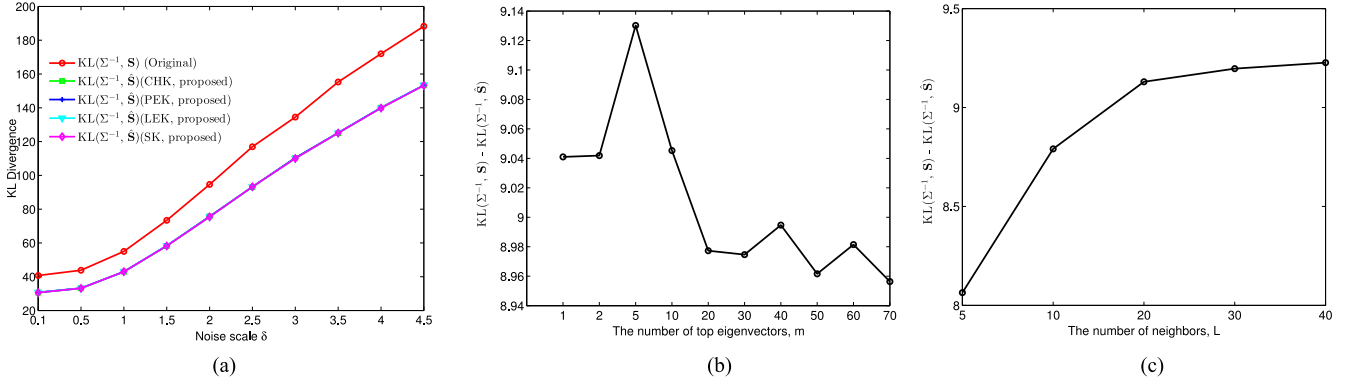


Fig. 3. Performance of the proposed preimage method on synthetic dataset. (a) Averaged KL divergence between the ground truth inverse covariance matrix Σ^{-1} and the original SICE matrix S (labeled by “original”) or the preimages $\hat{S}$ when four SPD kernels are used (labeled by “CHK,” “PEK,” “LEK,” and “SK,” respectively) at various noise levels with m and L set as 5 and 20, respectively. As indicated, the resulting KL divergence values corresponding to the four SPD kernels are consistently smaller than $KL(\Sigma^{-1}, S)$ at all noise levels. Moreover, the improvement of $KL(\Sigma^{-1}, \hat{S})$ over $KL(\Sigma^{-1}, S)$, i.e., $KL(\Sigma^{-1}, S) - KL(\Sigma^{-1}, \hat{S})$, becomes more significant with increase of δ . Note that the KL divergence values corresponding to the four kernels are similar and overlapped in the figure; (b) the improvement of the proposed preimage method (using SK) with various number of leading eigenvectors m when L is set as 20, and (c) the improvement of the proposed preimage method (using SK) with various number of neighbors L when m is set as 5.

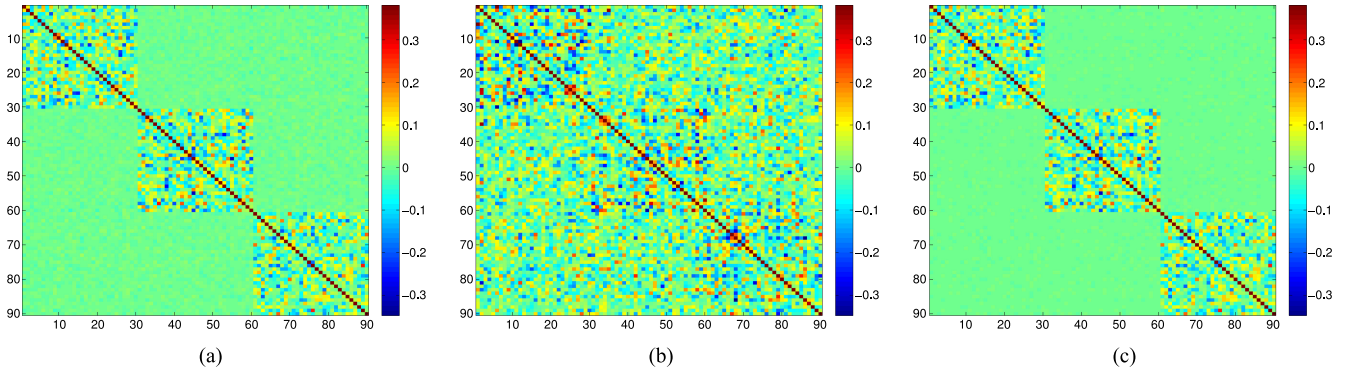


Fig. 4. Illustration of the result obtained by our proposed preimage method. (a) Shows a ground truth inverse covariance matrix Σ^{-1} , (b) plots the original SICE matrix S , and (c) shows the estimated preimage $\hat{S}$ of $\Phi_m(S)$. As seen, $\hat{S}$ is more similar to Σ^{-1} in comparison with S , indicating that the proposed preimage method brings some kind of denoising effect.

be used to measure the similarity between the two SPD matrices by relating them to the covariance matrices and setting the means as zero. KL divergence in our case is expressed as $KL(\Sigma_1, \Sigma_2) = \text{tr}(\Sigma_2^{-1}\Sigma_1) - \log \det(\Sigma_2^{-1}\Sigma_1) - d$, where d is the number of network nodes. It is nonnegative and a smaller divergence indicates that these two matrices are more similar.

The result is shown in Fig. 3. As seen in Fig. 3(a), $KL(\Sigma^{-1}, \hat{S})$ (averaged over all 82 test cases and with m and L set as 5 and 20, respectively.) is consistently lower than $KL(\Sigma^{-1}, S)$ for all the different noise levels and the SPD kernels used in the kernel PCA. This result suggests that the obtained preimage $\hat{S}$ is closer to the ground truth inverse covariance matrix Σ^{-1} in comparison with the original SICE matrix S . Relating back to the idea that we use to design this experiment, this result shows that the proposed preimage method indeed works. Also, the improvement of $KL(\Sigma^{-1}, \hat{S})$ over $KL(\Sigma^{-1}, S)$, i.e., $KL(\Sigma^{-1}, S) - KL(\Sigma^{-1}, \hat{S})$, becomes more significant with the increase of the noise level δ introduced in Step 3 of the synthetic data generation process. To demon-

strate the result obtained by the proposed pe-image method, an example is given in Fig. 4, where Fig. 4(a) shows a ground truth inverse covariance matrix Σ^{-1} , Fig. 4(b) plots the estimated SICE matrix S and Fig. 4(c) shows the preimage $\hat{S}$ of $\Phi_m(S)$. As seen, $\hat{S}$ is more similar to Σ^{-1} in comparison with S .

As indicated in Algorithm 1, the number of leading eigenvectors m and the number of neighbors L are two important parameters. We evaluate how the performance of the proposed preimage method will change with these two parameters. SK is taken as an example. Fig. 3(b) and (c) shows the improvement, i.e., $KL(\Sigma^{-1}, S) - KL(\Sigma^{-1}, \hat{S})$, of our method with different m and L , respectively. As seen in Fig. 3(b), when L is set as constant 20, the improvement first increases with m and then decreases, achieving the highest value when m is five. This is because the first several leading eigenvectors $\mathbf{v}_i$ in $\mathcal{F}$ represent the dominant network structures of the network while the following ones intend to characterize more detailed structures which are vulnerable to noise. As a result, with the increasing value of m , the components often correspond to noise.

Therefore, when $m > 5$, noisy components could be included, and this reduces the magnitude of the improvement. At the same time, note that the improvement does consistently hold although its magnitude is reduced. Fig. 3(c) shows that, when m is fixed at 5, the improvement with the increase of L becomes saturated when $L = 20$. This is because the constraint of $\sum_{j=1}^L w_j = 1$ in $\hat{\mathbf{S}} = \sum_{j=1}^L w_j \mathbf{S}_j$ (11) imposes the sparsity of w_j , limiting the actual number of neighbors $\mathbf{S}_j$ used to estimate $\hat{\mathbf{S}}$. Based on our experience, a relatively large initial number of L is recommended, e.g., one fourth of the number of training samples, and the constraint of $\sum_{j=1}^L w_j = 1$ will implicitly and automatically select a small set of $\mathbf{S}_j$ by setting most w_j as zero.

V. CONCLUSION

Recently, SICE has been used as a representation of brain connectivity to classify AD and NC. However, its high dimensionality can adversely affect the classification performance. Taking advantage of the SPD property of SICE matrices, we use SPD-kernel PCA to extract principal components to obtain a compact representation for classification. We also propose a preimage estimation algorithm, which allows visualization and analysis of the extracted principal connectivity patterns in the input space. The efficacy of the proposed method is verified by extensive experimental study on synthetic data and real rs-fMRI data from the ADNI.

In this paper, we specifically focus on unsupervised learning to explore compact representation without using class label information. Note that our framework can readily be extended to supervised case, such as kernel linear discriminant analysis, to explore discriminative representation. This will be studied in our future work.

REFERENCES

- [1] T. Musha *et al.*, "EEG markers for characterizing anomalous activities of cerebral neurons in NAT (neuronal activity topography) method," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 8, pp. 2332–2338, Aug. 2013.
- [2] C. Jack *et al.*, "Prediction of AD with MRI-based hippocampal volume in mild cognitive impairment," *Neurology*, vol. 52, no. 7, pp. 1397–1397, 1999.
- [3] J. Richiardi *et al.*, "Classifying minimally disabled multiple sclerosis patients from resting state functional connectivity," *NeuroImage*, vol. 62, no. 3, pp. 2021–2033, 2012.
- [4] R. Gould *et al.*, "Brain mechanisms of successful compensation during learning in Alzheimer disease," *Neurology*, vol. 67, no. 6, pp. 1011–1017, 2006.
- [5] X. Yang, "Evaluation of statistical inference on empirical resting state fMRI," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 4, pp. 1091–1099, Apr. 2014.
- [6] G. Varoquaux *et al.*, "Detection of brain functional-connectivity difference in post-stroke patients using group-level covariance modeling," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervention Conf.*, 2010, pp. 200–208.
- [7] K. Ugurbil, "Magnetic resonance imaging at ultrahigh fields," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 5, pp. 1364–1379, May 2014.
- [8] V. D. Heuvel *et al.*, "Exploring the brain network: A review on resting-state fMRI functional connectivity," *Eur. Neuropsychopharmacol.*, vol. 20, no. 8, pp. 519–534, 2010.
- [9] R. L. Buckner and J. L. Vincent, "Unrest at rest: Default activity and spontaneous network correlations," *NeuroImage*, vol. 37, no. 4, pp. 1091–1096, 2007.
- [10] M. D. Greicius *et al.*, "Functional connectivity in the resting brain: a network analysis of the default mode hypothesis," *Proc. Nat. Acad. Sci.*, vol. 100, no. 1, pp. 253–258, 2003.
- [11] V. Calhoun *et al.*, "A method for making group inferences from functional MRI data using independent component analysis," *Hum. Brain Mapping*, vol. 14, no. 3, pp. 140–151, 2001.
- [12] S. B. Katwal *et al.*, "Unsupervised spatiotemporal analysis of fMRI data using graph-based visualizations of self-organizing maps," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 9, pp. 2472–2483, Sep. 2013.
- [13] J. Damoiseaux *et al.*, "Consistent resting-state networks across healthy subjects," *Proc. Nat. Acad. Sci.*, vol. 103, no. 37, pp. 13848–13853, 2006.
- [14] M. van den Heuvel *et al.*, "Normalized cut group clustering of resting-state fMRI data," *PLoS One*, vol. 3, no. 4, p. e2001, 2008.
- [15] S. M. Smith *et al.*, "Functional connectomics from resting-state fMRI," *Trends Cognit. Sci.*, vol. 17, no. 12, pp. 666–682, 2013.
- [16] N. Leonardi *et al.*, "Principal components of functional connectivity: A new approach to study dynamic brain connectivity during rest," *NeuroImage*, vol. 83, pp. 937–950, 2013.
- [17] C.-Y. Wee *et al.*, "Constrained sparse functional connectivity networks for MCI classification," in *Proc. Med. Image Comput. Comput.-Assisted Intervention Conf.*, 2012, pp. 212–219.
- [18] S. M. Smith, "The future of fMRI connectivity," *NeuroImage*, vol. 62, no. 2, pp. 1257–1266, 2012.
- [19] J. Friedman *et al.*, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.
- [20] S. Huang *et al.*, "Learning brain connectivity of Alzheimers disease by exploratory graphical models," *NeuroImage*, vol. 50, pp. 935–949, 2010.
- [21] S. Huang *et al.*, "Learning brain connectivity of Alzheimer's disease by sparse inverse covariance estimation," *NeuroImage*, vol. 50, no. 3, pp. 935–949, 2010.
- [22] B. Ng *et al.*, "A novel sparse group Gaussian graphical model for functional connectivity estimation," *Inf. Process. Med. Imag.*, vol. 23, pp. 256–267, 2013.
- [23] O. Sporns *et al.*, "Identification and classification of hubs in brain networks," *PLoS One*, vol. 2, no. 10, p. e1049, 2007.
- [24] X. Pennec *et al.*, "A riemannian framework for tensor computing," *Int. J. Comput. Vis.*, vol. 66, no. 1, pp. 41–66, 2006.
- [25] P. T. Fletcher *et al.*, "Principal geodesic analysis for the study of nonlinear statistics of shape," *IEEE Trans. Med. Imag.*, vol. 23, no. 8, pp. 995–1005, Aug. 2004.
- [26] S. Sra, "A new metric on the manifold of kernel matrices with application to matrix geometric mean," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger, Eds., New York, NY: Curran Associates, Inc., 2012, pp. 144–152.
- [27] V. Arsigny *et al.*, (2006). Log-euclidean metrics for fast and simple calculus on diffusion tensors. *Magn. Reson. Med.*, [Online]. 56(2), pp. 411–421. Available: <http://dx.doi.org/10.1002/mrm.20965>
- [28] M. T. Harandi *et al.*, "Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach," in *Proc. Comput. Vis. Conf.*, 2012, pp. 216–229.
- [29] S. Jayasumana *et al.*, "Kernel methods on the riemannian manifold of symmetric positive definite matrices," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 2013, pp. 73–80.
- [30] J.-Y. Kwok and I.-H. Tsang, "The pre-image problem in kernel methods," *IEEE Trans. Neural Netw.*, vol. 15, no. 6, pp. 1517–1525, Nov. 2004.
- [31] Y. Rathi *et al.*, "Statistical shape analysis using kernel PCA," in *Proc. Electron. Imag. Int. Soc. Opt. Photon. Conf.*, 2006, pp. 60641B–60641B.
- [32] S. M. Smith *et al.*, "Network modelling methods for fMRI," *NeuroImage*, vol. 54, no. 2, pp. 875–891, 2011.
- [33] J. Zhang *et al.*, "Exploring compact representation of SICE matrices for functional brain network classification," presented at the Workshop Mach. Learn. Med. Imag. Boston, MA, USA, 2014.
- [34] J. Liu *et al.*, (2009). SLEP: Sparse Learning With Efficient Projections, Arizona State University. [Online]. Available: <http://www.public.asu.edu/~jye02/Software/SLEP>
- [35] W. Förstner and B. Moonen, "A metric for covariance matrices," in *Geodesy—The Challenge of the 3rd Millennium*. New York, NY, USA: Springer, 2003, pp. 299–309.
- [36] I. L. Dryden *et al.*, "Non-euclidean statistics for covariance matrices, with applications to diffusion tensor imaging," *Ann. Appl. Stat.*, vol. 3, pp. 1102–1123, 2009.
- [37] B. Schölkopf *et al.*, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, 1998.

- [38] L. Zhou *et al.*, "Identifying anatomical shape difference by regularized discriminative direction," *IEEE Trans. Med. Imag.*, vol. 28, no. 6, pp. 937–950, Jun. 2009.
- [39] O. Tuzel *et al.*, "Pedestrian detection via classification on riemannian manifolds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1713–1727, Oct. 2008.
- [40] Y. Chao-Gan and Z. Yu-Feng, "DPARSF: A MATLAB toolbox for pipeline data analysis of resting-state fMRI," *Front. Syst. Neurosci.*, vol. 4, pp. 1–7, 2010.
- [41] N. Tzourio-Mazoyer *et al.*, "Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain," *NeuroImage*, vol. 15, no. 1, pp. 273–289, 2002.
- [42] M. Kaiser, "A tutorial in connectome analysis: Topological and spatial features of brain networks," *NeuroImage*, vol. 57, no. 3, pp. 892–907, 2011.
- [43] S. Mika *et al.* (1999). Kernel PCA and de-noising in feature spaces, in *Proc. Conf. Adv. Neural Inf. Process. Syst. II*, pp. 536–542. [Online]. Available: <http://dl.acm.org/citation.cfm?id=340534.340729>
- [44] T. Tokuda *et al.*, "Visualizing distributions of covariance matrices," Columbia Univ., New York, NY, USA, Tech. Rep., 2011.
- [45] N. Städler, and P. Bühlmann, "Missing values: Sparse inverse covariance estimation and an extension to sparse regression," *Stat. Comput.*, vol. 22, no. 1, pp. 219–235, 2012.

Authors' photographs and biographies not available at the time of publication.