On the Comparisons of Decorrelation Approaches for Non-Gaussian Neutral Vector Variables

Ma, Zhanyu; Lu, Xiaoou; Xie, Jiyang; Yang, Zhen; Xue, Jing-Hao; Tan, Zheng-Hua; Xiao, Bo ; Guo, Jun

Published in:
I E E E Transactions on Neural Networks and Learning Systems

DOI (link to publication from Publisher):
10.1109/TNNLS.2020.2978858

Publication date:
2023

Document Version
Accepted author manuscript, peer reviewed version

On the Comparisons of Decorrelation Approaches for Non-Gaussian Neutral Vector Variables

Zhanyu Ma, *Senior Member, IEEE*, Xiaoou Lu, Jiyang Xie, *Student Member, IEEE*, Zhen Yang, *Member, IEEE*, Jing-Hao Xue, Zheng-Hua Tan, *Senior Member, IEEE*, Bo Xiao, and Jun Guo

***Abstract*— As a typical non-Gaussian vector variable, a neutral vector variable contains nonnegative elements only, and its l_1-norm equals one. In addition, its neutral properties make it significantly different from the commonly studied vector variables (e.g., the Gaussian vector variables). Due to the aforementioned properties, the conventionally applied linear transformation approaches [e.g., principal component analysis (PCA) and independent component analysis (ICA)] are not suitable for neutral vector variables, as PCA cannot transform a neutral vector variable, which is highly negatively correlated, into a set of mutually independent scalar variables and ICA cannot preserve the bounded property after transformation. In recent work, we proposed an efficient nonlinear transformation approach, i.e., the parallel nonlinear transformation (PNT), for decorrelating neutral vector variables. In this article, we extensively compare PNT with PCA and ICA through both theoretical analysis and experimental evaluations. The results of our investigations demonstrate the superiority of PNT for decorrelating the neutral vector variables.**

***Index Terms*— Decorrelation, neutral vector variable, neutrality, non-Gaussian, nonlinear transformation.**

Manuscript received July 19, 2019; revised December 25, 2019; accepted March 3, 2020. This work was supported in part by the National Key R&D Program of China under Grant 2019YFF0303300 and under Subject II No. 2019YFF0303302, in part by the National Science and Technology Major Program of the Ministry of Science and Technology under Grant 2018ZX03001031, in part by the National Natural Science Foundation of China (NSFC) under Grant 61773071, Grant 61922015, Grant U19B2036, and Grant 61671030, in part by the Beijing Academy of Artificial Intelligence (BAAI) under Grant BAAI2020ZJ0204, in part by the Beijing Nova Programme Interdisciplinary Cooperation Project under Grant Z191100001119140, in part by the Beijing Municipal Natural Science Foundation under Grant L172030 and Grant 19L2020, in part by the Importation and Development of High-Caliber Talents Project of Beijing Municipal Institutions under Grant CIT&TCD20190308, in part by the Scholarship from China Scholarship Council (CSC) under Grant CSC 201906470049, and in part by the BUPT Excellent Ph.D. Students Foundation under Grant CX2019109. *(Corresponding authors: Zhanyu Ma; Zhen Yang.)*

Zhanyu Ma, Jiyang Xie, Bo Xiao, and Jun Guo are with the Pattern Recognition and Intelligent System Laboratory, School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: mazhanyu@bupt.edu.cn).

Xiaoou Lu and Jing-Hao Xue are with the Department of Statistical Science, University College London, London WC1E 6BT, U.K.

Zhen Yang is with the College of Computer Science, Faculty of Information Technology, Beijing University of Technology, Beijing 100022, China (e-mail: yangzhen@bjut.edu.cn).

Zheng-Hua Tan is with the Department of Electronic Systems, Aalborg University, 9220 Aalborg, Denmark.

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

Digital Object Identifier 10.1109/TNNLS.2020.2978858

I. Introduction

DECORRELATION of a random vector variable plays an essential role in multivariate data analysis, signal processing, pattern recognition, and machine learning [1]–[3]. It can transform a correlated vector variable into a set of mutually uncorrelated scalar/subvector variables. That is, although the covariance matrix of the vector variable may not be diagonal, the covariance matrix of the resultant scalar variables can be made diagonal by a decorrelation transform; in other words, the correlations between the variables have been removed by the decorrelation transform.

A process closely related to decorrelation is called whitening, which removes not only the correlations between variables but also the variances of variables, transforming the original covariance matrix into an identity matrix. To achieve the whitening of a vector variable, there are many linear transforms, including the Mahalanobis transform, the Cholesky decomposition, and the eigendecomposition of the precision matrix (i.e., the inverse of the covariance matrix) [4]. However, using whitening transforms for decorrelation has a limitation. After whitening transformation, every uncorrelated scalar variable has unit variance; this means that the uncorrelated scalar variables are not distinguishable from each other in terms of variance (or "energy"). It is possible to further recover the original variances (on the diagonal entries of the original covariance matrix) to the uncorrelated variables [4], but this also means that the distribution of the variance over the elements of the vector variable does not change after transformation. A distributional change like the concentration of variance, such that the resultant uncorrelated scalar variables can be better distinguished, is often desirable in practice for tasks, such as data compression, dimension reduction, and feature selection. To this end, one can resort to linear orthogonal transforms.

Linear orthogonal transforms, including the renowned Fourier transform, discrete cosine transform, and Karhunen–Loève transform, are not only able to decorrelate the elements of a vector variable to various extents but also able to concentrate the "energy" (in terms of variance) of the vector in a small number of scalar variables obtained from the transformation [5]. Hence, linear orthogonal transforms are widely used to decorrelate a vector variable.

The Karhunen–Loève transform, also better known as principal component analysis (PCA) [6], among others, is a

ubiquitously applied linear orthogonal transformation method that can decorrelate a vector variable into a set of uncorrelated scalar variables. Moreover, if the original vector variable follows a multivariate Gaussian distribution, PCA can yield a set of mutually independent scalar variables. By applying eigenvalue analysis to the covariance matrix of vectors, PCA linearly maps the original vector into space spanned by the covariance matrix's eigenvectors [1]. If we treat the eigenvalue as the "energy" of the corresponding variable and select K eigenvectors that correspond to the top K eigenvalues as the representative features, PCA serves as a feature selection/dimension reduction approach to the vector [6]. The PCA-based feature selection/dimension reduction approach (and its extended versions, e.g., kernel PCA [7], [8]), which can also be considered as low-rank matrix approximation, has been widely applied in face recognition [9], [10], speech enhancement [11], text analysis [12], blind source separation [13], source coding [14], and so on.

In order to get mutually independent variables with PCA, the multivariate Gaussian assumption is usually applied to the original vector. However, it is uncommon to have true Gaussian distributed data in real-life applications [15]. For example, the gray or color pixel values in image processing [16], the rating scores to an item in a recommendation system [17], and the genome-wide DNA methylation level value in bioinformatics [18] are all strictly bounded and distributed in the interval [0, 1]. The speech signal's spectrum coefficients are distributed as $x \in (0, +\infty)$, which is semi-bounded [19]. The l_2-norms of the spatial fading correlation and the yeast gene expressions [20] are all equal to 1, and such data convey directional property (i.e., $\|\mathbf{x}\|_2 = 1$). Another type of data is the proportional/compositional data [21], which are nonnegative and have a l_1-norm equal to one. The aforementioned data all have asymmetric or constrained distributions [22], and they do not match the natural definition of the Gaussian distribution (i.e., the definition domain is unbounded, and the distribution shape is symmetric). Hence, these data are non-Gaussian distributed [23]. Recently, it has been demonstrated in many studies that explicitly utilizing the non-Gaussian characteristics can significantly improve the practical performance [16], [19], [20], [23]–[25]. Applying PCA to non-Gaussian distributed data can only get uncorrelated but also not independent variables, and therefore, the consequent performance, which requires the variables' mutual independence, will be decreased [23], [25], [26].

Independent component analysis (ICA) can decorrelate any vector variable (observed data) into a set of mutually independent scalar variables (data sources) [27], [28], with the assumption that the data sources are mutually independent and non-Gaussian distributed. Hence, applying ICA to non-Gaussian distributed vectors can lead to not only decorrelation but also independence. However, ICA is computationally costly because it requires several preprocessing steps, including centering, whitening, and/or dimension reduction before implementation [29]. ICA has been widely applied in several fields, such as face recognition [30], blind source separation [31], and wireless communications [32].

Neutral vector variables [33], [34] are a typical non-Gaussian vector variable. The non-Gaussian properties of neutral vector variables are as follows: 1) all the elements in a neutral vector variable are nonnegative; and 2) the l_1-norm of a neutral vector variable equals one. The neutral vector variable has been widely applied in many real-life applications. In biological research, the neutral vector had been applied to data on bone composition in rats and scute growth in turtles [33]. To describe the characteristics of the proportional data/compositional data, neutral vector variable has been extensively applied in document analysis [35], [36], image processing [37], and speech signal processing [38], [39]. A typical distribution for modeling the distribution of a neutral vector variable is the Dirichlet distribution [40]. As a classical method for constructing nonparametric models, several Dirichlet distribution-based Dirichlet process models have been proposed for the purpose of feature selection [41], [42], cognitive radios [43], [44], and so on. In order to explicitly explore the properties of the neutral-like data,[1] the Dirichlet distribution and the corresponding Dirichlet mixture model (DMM) have been applied to model the underlying distributions of such data [25], [45], [46]. The Bayesian estimation of DMM with variational inference, which provides an analytically tractable solution for parameter estimation, has been proposed in [47].

The neutral vector variable can be considered as a point process distributed variable in the plane of $\sum_{i=1}^{N} x_i = 1$. Both of them are used for analyzing bounded data. However, the point process focuses on discussing spatial and temporal relationships between data points and is mainly for modeling data with three types: 1) sequential data in continuous time [48], [49]; 2) spatial representations of locations [50], [51]; and 3) spatiotemporal data [52], [53], while the neutral vector variable can be applied for modeling not only spatiotemporal data but also other data without temporal and spatial correlations. Thus, the point process distributed variable can be considered as a special case of the neutral vector variable in the fields of applications.

Obviously, directly applying PCA to neutral vector variable can only yield uncorrelated variables. The mutual independence, which is required in many cases, is not guaranteed due to the non-Gaussian properties. With linear projection, the Dirichlet component analysis (DCA) was proposed to replace PCA for the Dirichlet variable decorrelation and dimension reduction [54]. Although DCA preserves the relevant constraints among the elements of the vector variable, it can only guarantee that the mapped components are decorrelated as much as possible. Mutual independence cannot be obtained by DCA either. With ICA, mutually independent scalar variables can be obtained after decorrelation. However, the bounded property cannot be preserved.

By explicitly exploring the completely neutral property [34], we have proposed a special nonlinear transformation strategy, namely, the parallel nonlinear transformation (PNT),

[1] "Neutral-like" data denote data simply satisfying the nonnegative and unit l_1-norm properties. However, these data may not have all the neutral vector variable's properties.

to decorrelate the neutral vector variable into a set of mutually independent scalar variables or a set of mutually independent subvector variables [25], [55]. The PNT has been successfully applied in many areas, such as speech linear predictive coding (LPC) model quantization [25] and feature selection for EEG signal classification [26].[2]

For neutral vector variable decorrelation, PNT, PCA, and ICA have several similarities: 1) all of them transform a vector variable into a set of uncorrelated scalar variables and 2) by yielding uncorrelated variables, they can all serve as feature selection methods. However, there are also some dissimilarities among these methods.

1) PCA and ICA are linear transformations, while PNT is nonlinear.
2) PCA is optimal[3] for Gaussian vector variables, ICA is optimal for any non-Gaussian sources, and PNT is optimal for neutral vector variables.
3) Neither PCA or ICA can preserve bounded support property, while PNT preserves it.
4) The eigenvalue analysis is the prerequisite for conducting linear transformation in PCA, several preprocessing steps are required for ICA, while PNT does not require the computation of statistical properties in its implementation. Hence, it is of sufficient interest to conduct extensive comparisons among these strategies for the neutral vector variables.

Several improved variants of PCA or ICA exist, such as nonlinear PCA [56], fast robust PCA [57], kernel PCA [58], kernel ICA [59], and binary ICA [28]. However, the purpose of this article is to analyze and compare the fundamental decorrelation methods for neutral vector variables, rather than involving the improved variants of them. Hence, we compare only the proposed PNT with the original PCA or ICA.

The contribution of this article can be summarized as follows.

1) We provide a thorough study of the so-called PNT decorrelation strategy for the non-Gaussian neutral vector variable, which is optimal, preserves the non-Gaussian properties, and does not need to calculate the statistical properties during operation.
2) Intensive comparisons between the proposed PNT and the conventionally used PCA and ICA have been conducted. Theoretical analysis and synthesized and real data evaluations demonstrate the effectiveness and the robustness of the proposed method.

The remaining parts of this article are organized as follows. In Section II, we briefly introduce the neutral vector and its related concepts and properties. The details of PNT, PCA, and ICA will be provided in Section III. Extensive comparisons among these methods, with theoretical analysis and data evaluations, will be conducted in Section IV. We will draw some conclusions in Section V.

[2]Part of the work in the submitted manuscript [the RBF-SVM+PCA and the RBF-SVM+PNT results in Fig. 7(c)–(f)] has been published in [26]. Focusing on the general framework for decorrelating a completely neutral vector, this article introduces the concept of a completely neutral vector and demonstrates the advantages (by comparing with PCA and ICA) of this framework with both synthesized data and real-life data applications. In contrast, the work in [26] is only a use case of the proposed methods.

[3]Hereby, "optimal" means that the transformation can yield not only uncorrelated but also mutually independent scalar variables.

II. Neutral Vector Variable

Assume that we have a random vector variable $\mathbf{x} = [x_1, x_2, \ldots, x_K, x_{K+1}]^{\mathrm{T}}$, where $x_k > 0$ and $\sum_{k=1}^{K+1} x_k = 1$. Let $\mathbf{x}_{k1} = [x_1, \ldots, x_k]^{\mathrm{T}}$ and $\mathbf{x}_{k2} = [x_{k+1}, \ldots, x_{K+1}]^{\mathrm{T}}$. The vector $\mathbf{x}_{k1}$ is neutral if $\mathbf{x}_{k1}$ is independent of $\mathbf{w}_k = (1/(1 - s_k))\mathbf{x}_{k2}$ (i.e., $\mathbf{x}_{k1} \perp \mathbf{w}_k$), for $1 \le k \le K$ [33], [34], where $s_k = \sum_{i=1}^{k} x_i$ and $s_0 = 0$. If for all k, $\mathbf{x}_{k1}$ are neutral, then $\mathbf{x}$ is defined as a completely neutral vector variable [33], [60]. A (completely) neutral vector variable with $(K + 1)$ elements has K degrees of freedom.

A completely neutral vector variable has the following relatively proportional properties [55]:

Property 1 (Mutual Independence): For completely neutral vector variable $\mathbf{x}$, define $z_k = \frac{x_k}{1-s_{k-1}}$ and $z_1 = x_1$, and we have that $z_1, z_2, \ldots, z_K$ are mutually independent.

Property 2 (Aggregation Property): For a completely neutral vector variable $\mathbf{x}$, when adding any adjacent elements x_r and x_{r+1} together, the resulting K-dimensional vector $\mathbf{x}^{r\uplus r+1} = [x_1, \ldots, x_r + x_{r+1}, \ldots, x_{K+1}]$ is a completely neutral vector again.

Property 3 (Exchangeable Property): For a completely neutral vector variable $\mathbf{x}$, if any arbitrarily permuted version of $\mathbf{x}$ is still completely neutral, then this vector variable is exchangeably completely neutral.

For the convenience of expression, we use "neutral vector variable" to represent the term "completely neutral vector variable" for short.

The Dirichlet variable is a typical case of neutral vector variable [1], [61], and it contains nonnegative elements with summation equals one. The probability density function of a $(K + 1)$-dimensional Dirichlet distribution, given parameter vector $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_{K+1}]^{\mathrm{T}}$, is defined as

$$\mathbf{Dir}(\mathbf{x}; \boldsymbol{\alpha}) = \frac{\Gamma\left(\sum_{k=1}^{K+1} \alpha_k\right)}{\prod_{k=1}^{K+1} \Gamma(\alpha_k)} \prod_{k=1}^{K+1} x_k^{\alpha_k - 1}, \quad x_k \ge 0, \quad \sum_{k=1}^{K+1} x_k = 1, \ \alpha_k > 0. \tag{1}$$

The covariance matrix of the Dirichlet distribution is [62]

$$\mathrm{Cov}[\mathbf{x}]_{i,j} = \begin{cases} \dfrac{\alpha_i(s - \alpha_i)}{s^2(s+1)}, & i = j \\ \dfrac{-\alpha_i \alpha_j}{s^2(s+1)}, & i \ne j \end{cases} \tag{2}$$

where $s = \sum_{k=1}^{K+1} \alpha_k$. Obviously, the covariance matrix of the Dirichlet vector variable is negatively correlated (off-diagonal elements are negative), which reflects the proportional property of the neutral vector variable.

In summary, a neutral vector variable should satisfy the following:

1) nonnegative elements and unit l_1-norm;
2) relatively proportional properties;
3) negatively correlated covariance matrix.

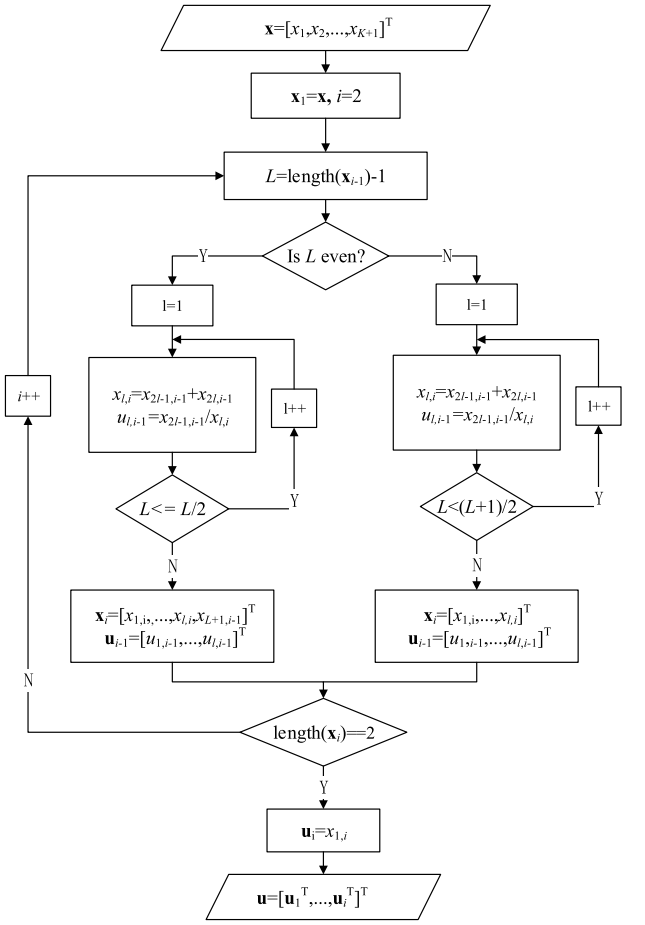


Fig. 1. Flowchart of PNT.

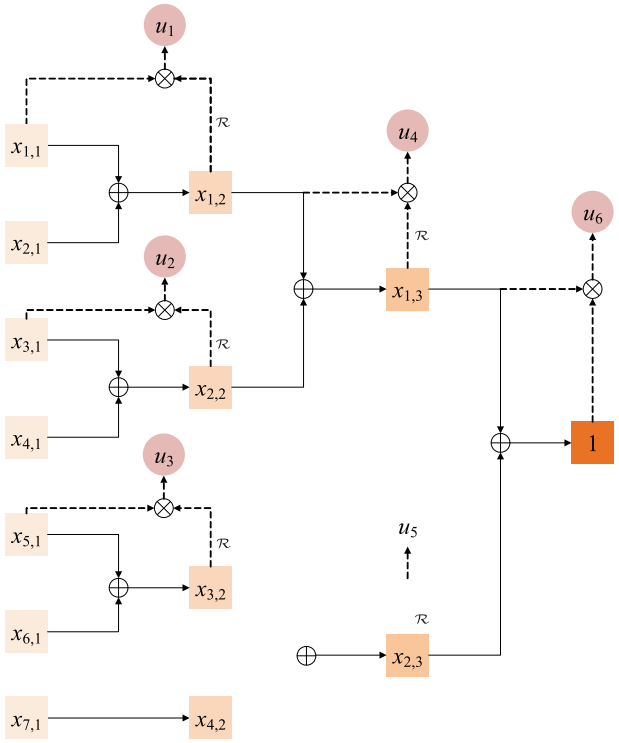


Fig. 2. Example of PNT with $K = 6$. The transformed coefficients are $u_1 = x_{1,1}/x_{1,2}$, $u_2 = x_{3,1}/x_{2,2}$, $u_3 = x_{5,1}/x_{3,2}$, $u_4 = x_{1,2}/x_{1,3}$, $u_5 = x_{3,2}/x_{2,3}$, and $u_6 = x_{1,3}$. $\mathcal{R}$ represents the reciprocal operation.

III. Decorrelation Approaches

Both PCA and ICA are commonly known for the communities of signal processing, pattern recognition, machine learning, and so on. Due to the limitation of space, we skip the introduction to the technical details of these two methods and focus on PNT in this article. Detailed information about PCA and ICA can be found in [1].

With the aforementioned properties, a neutral vector variable exhibits a particular type of statistical independence among its elements [33]. In order to explicitly explore such type of independence, we proposed a so-called PNT scheme to transform a neutral vector variable into a set of mutually independent scalar variables [55]. For a neutral vector variable, PNT carries out a nonlinear transformation according to the procedure illustrated in Fig. 1.

For a $(K+1)$-dimensional neutral vector variable, K mutually independent scalar variables, each of which is distributed in the interval $[0, 1]$, can be obtained. The proof of mutual independence has been presented in [55]. An example of applying PNT to a 7-D (i.e., $K = 6$) neutral vector variable is shown in Fig. 2. A fast implementation of PNT (FPNT), which involves zero-padding, was introduced in [55].

Note that the proposed PNT scheme can be simply implemented by iterative elementwise summation and division operations. No statistical information of the variables, e.g., covariance matrix, is required. In other words, unlike PCA or ICA, which needs to get eigenvalues and eigenvectors in advance, the PNT can be carried out based on the neutral vector variable itself.

IV. Experimental Results and Discussion

PNT is a nonlinear decorrelation method specially designed for neutral vector variables. Meanwhile, PCA or ICA is a typically and widely applied decorrelation method, which can also be applied to neutral vector variables. Hence, in terms of decorrelation performance for neutral vector variables, it is of sufficient interest to conduct extensive comparisons for these two methods, with theoretical analysis, synthesized data evaluation, and real data evaluation.

A. Comparisons With Theoretical Analysis

1) Mutual Independence: The importance of independence arises in many applications. With the scheme introduced in Section III, a neutral vector can be transformed to a set of mutually independent scalar variables by PNT, in a nonlinear manner. PCA can be applied to transform any vector variable, with a linear manner, to a set of uncorrelated scalar variables. However, PCA can yield mutually independent scalar variables only when the vector variable is multivariate Gaussian. With ICA, a neutral vector variable can be transformed into a set of mutually independent scalar variables as well, which is due to the principles of ICA.

Hence, in terms of mutual independence, PNT and ICA are optimal for neutral vector variables.

2) Computational Complexity: In practical applications, the computational complexity of decorrelation is usually an essential concern. We now compare the computational complexities of PNT, PCA, and ICA.

PNT can be conducted in a parallel manner. According to the algorithm described in Fig. 1, it requires at most $\lceil \log_2(K+1) \rceil$ iterations. Within each iteration, about $L/2$ summations and $L/2$ divisions with an even L or $(L+1)/2$ summations and $(L+1)/2$ divisions with an odd L are needed. Therefore, if we treat the summation as one floating-point operation and the division as eight times of that,[4] the computational complexity for PNT is $\mathcal{O}(K \log K)$ since $L = K$ at the first iteration and L will reduce to (approximately) half in each of the consequent iteration.

Implementation of PCA generally contains two stages: 1) eigenvalue analysis of the covariance matrix and 2) linear mapping of the vector via eigenvectors. To the best of our best knowledge, the fastest method for eigenvalue analysis so far is the method proposed by Luk and Qiao [64]. With the method proposed in [64], the computational cost of eigenvalue analysis is about $\mathcal{O}(K^2 \log K)$ for a $K \times K$ covariance matrix. For the linear mapping, multiplying a vector with the eigenvector matrix has a computational cost around $\mathcal{O}(K^2)$. Therefore, the computational cost for PCA is, on average, $\mathcal{O}(K^2 \log K)$.

In terms of source separation, ICA has a robust performance. However, one drawback of the algorithms designed for carrying out ICA is the high computational load required in implementation [65]. Generally speaking, algorithms for ICA require centering, whitening, and dimension reduction as the preprocessing steps for the purpose of facilitating calculation. As mentioned in [29], the computational cost for ICA is $\mathcal{O}(MK^2)$, where M denotes the number of iterations required. This indicates that the convergence of ICA depends on the number of iterations as well.

As PNT avoids the eigenvalue analysis/whitening for PCA/ICA, the computational complexity is significantly reduced. For neutral vector variable decorrelation, PNT has less computational cost than both PCA and ICA.

3) Preservation of Non-Gaussian Property: An important property of a neutral vector is its bounded support property. It is usually required that such property can be preserved after transformation. The proposed PNT method meets this requirement with its division operation. Neither PCA nor ICA can preserve the bounded support property,[5] as there is no constraint applied during transformation to ensure the resultant scalar variables (uncorrelated or independent) have unconstrained support range.

In terms of non-Gaussian property preservation only, PNT is capable and, thus, outperforms PCA and ICA.

4) Discussion: The summary of the aforementioned theoretical comparisons is listed in Table I. It is observed that PCA and ICA both have more computational complexity than the PNT method. ICA usually has a larger computational cost than PCA since M is a number larger than $\log K$. Meanwhile, ICA needs many iterations to converge, and analytically tractable solution does not exist. In terms of non-Gaussianity, PNT is the only one that preserves the bounded support property.

TABLE I

PROPERTIES OF PNT, PCA, AND ICA FOR DECORRELATION OF N SAMPLES. SEE TEXT FOR ANALYSIS

Method	Analytically tractable solution	Computational complexity	NG property preservation
PNT	$\surd$	$\mathcal{O}(K \log K)$	$\surd$
PCA	$\surd$	$\mathcal{O}(K^2 \log K)$	$\times$
ICA	$\times$	$\mathcal{O}(MK^2)$	$\times$

In summary, for neutral vector variables, PNT performs better than PCA and ICA, in terms of decorrelation, computational complexity, and non-Gaussianity preservation. Compared with PNT and PCA, ICA does not have an analytically tractable solution. Therefore, ICA algorithms typically resort to iterative procedures with either difficulties or high computational load. Moreover, although ICA can yield mutually independent scalar variables (PNT can do this as well for neutral vector variable), it cannot preserve the NG property and is not a "suitable" method for fair comparisons. Hence, we compare only PNT and PCA in the following.

B. Comparisons Through Synthesized Data Evaluation

1) Decorrelation Effect on Neutral Vector Variables: Vectors generated from a Dirichlet distribution are completely neutral. In order to illustrate the decorrelation effect of the PNT and PCA on neutral vector variables, we generated vectors from a given Dirichlet distribution with parameter $\boldsymbol{\alpha} = [3, 5, 15, 9, 12, 8, 7, 20]^{\mathrm{T}}$. PNT and PCA were applied to this generated data set, respectively.

In order to measure the decorrelation effect quantitatively, the distance correlation (DC) [66], [67] was calculated to evaluate the mutual independence after decorrelation. The conventionally used the Pearson correlation coefficient [68], [69] can only measure correlations between two random variables. Unlike the Pearson correlation coefficient, the DC is zero if and only if the random variables are mutually statistically independent [70]. Given a set of paired samples $(X_n, Y_n),\ n = 1, \ldots, N$, all pairwise Euclidean distances a_{ij} and b_{ij} are calculated as

$$a_{ij} = \|X_i - X_j\|, \quad b_{ij} = \|Y_i - Y_j\|,\ i, j = 1, \ldots, N. \tag{3}$$

Taking the doubly centered distances, we have

$$A_{ij} = a_{ij} - \bar{a}_{i\cdot} - \bar{a}_{\cdot j} + \bar{a}_{\cdot\cdot}, \quad B_{ij} = b_{ij} - \bar{b}_{i\cdot} - \bar{b}_{\cdot j} + \bar{b}_{\cdot\cdot} \tag{4}$$

where $\bar{a}_{i\cdot}$ denotes the mean of the ith row, $\bar{a}_{\cdot j}$ is the mean of the jth column, and $\bar{a}_{\cdot\cdot}$ stands for the grand mean of the matrix. The same definitions apply to $\bar{b}_{i\cdot}$, $\bar{b}_{\cdot j}$, and $\bar{b}_{\cdot\cdot}$. The DC is calculated as

$$\mathrm{DC} = \sqrt{\frac{\sum_{i,j=1}^{N} A_{ij} B_{ij}}{\sqrt{\sum_{i,j=1}^{N} A_{ij}^2 \sum_{i,j=1}^{N} B_{ij}^2}}}. \tag{5}$$

In order to evaluate the statistical significance of the DC, a permutation test is employed. The p-value for the permutation test is calculated as follows.

[4]According to T. Minka's Lightspeed MATLAB toolbox [63].

[5]Some kernel methods can be applied to preserve the bounded support property; however, it is out of the scope of this article.

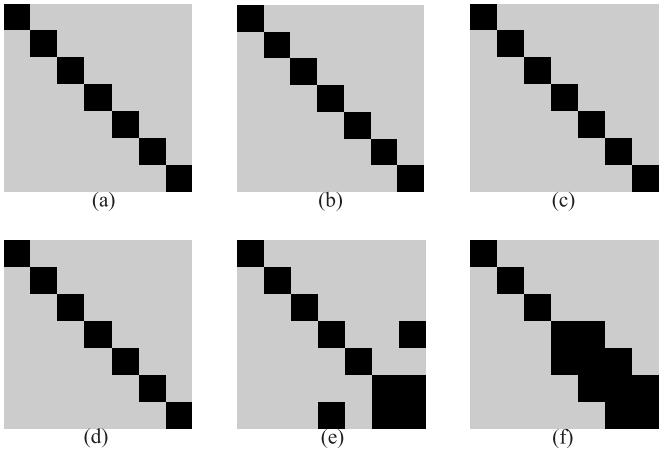


Fig. 3. Decorrelation performances of PNT and PCA measured with p-values. See text for details. (a) PNT, $N = 100$. (b) PNT, $N = 200$. (c) PNT, $N = 400$. (d) PCA, $N = 100$. (e) PCA, $N = 200$. (f) PCA, $N = 400$.

1) For the original data (X_n, Y_n), create a new data set (X_n, Y_{n^*}), where n^* denotes a permutation of the set $\{1, \ldots, N\}$. The permutation set is selected randomly as drawing without replacement.
2) Calculate a DC for the randomized data.
3) Repeat the above-mentioned two steps a large number of times, and the p-value for this permutation test is the proportion of the DC values in step 2, which are larger than the DC from the original data.

The null hypothesis, in this case, is that the two variables involved are independent of each other (the DC is 0). When the corresponding p-value is smaller than 0.05, the null-hypothesis is rejected so that these two variables are not independent (but could still be uncorrelated). Hence, p-value greater than 0.05 indicates mutual independence. We choose the significance level as 0.05 in this article.

The decorrelation performance, with different amounts of generated data, is illustrated in Fig. 3. PNT and PCA were applied to transform the generated vectors, respectively. The p-values of the transformed data were calculated. We chose 0.05 as the threshold to encode the p-values to black (if p-value is smaller than 0.05, which indicates dependence) or gray (otherwise).

When the amount of data is small (e.g., $N = 100$), the generated data cannot reveal obvious complete neutral properties. Hence, both PCA and PNT perform well, and they can decorrelate such a "semi"neutral vector variable into a set of mutually independent scalar variables.

As the amount of generated data increases, clear complete neutrality can be expected. It can be observed that PNT always transforms a neutral vector variable into a set of mutually independent scalar variables [the diagonal elements of the p-value matrix are smaller than 0.05, and all the off-diagonal elements are larger than 0.05, as shown in Fig. 3(a)–(c)]. PCA does not perform well in terms of yielding mutually independent scalar variables when applied to neutral vector variables [some of the off-diagonal elements are smaller than 0.05, as shown in Fig. 3(e) and (f)].

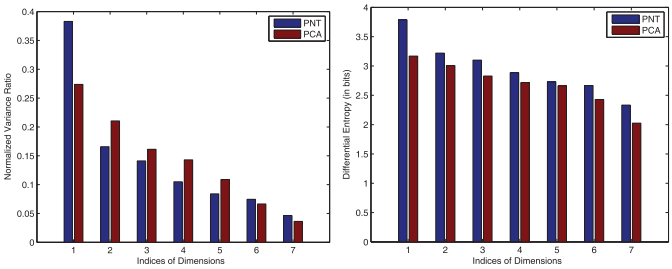


Fig. 4. Effect of PNT versus PCA on energy distribution. (a) Comparisons of normalized variance ratio. (b) Comparisons of differential entropy.

Similar performances can be obtained when choosing other parameter settings, and we show only one example. For neutral vector decorrelation, PNT outperforms PCA.

2) Effect on Energy Distribution: In pattern recognition applications, getting a set of independent/uncorrelated variables from a correlated vector variable is helpful for feature selection. Given the independent/uncorrelated features, we can select features to construct a new subspace, in which it is easier to distinguish data according to their labels.[6] It is generally useful to select the dimensions that have relatively large variances such that the multimodality of the data distribution is preserved. From the perspective of information theory, feature selection always favors the dimensions with relatively large differential entropies. In this article, we treat either variance or differential entropy as the "energy" of the dimension. In this case, the feature selection task aims at selecting the dimensions with relatively large energies.

With similar Dirichlet parameter settings as in Section IV-B1, we generated 5000 vectors from a Dirichlet distribution. After applying PNT and PCA on these data, separately, we compared the energy distributions yielded by these two schemes. The variances of the scalar variables after transformation are first normalized to have a unit l_1-norm and then sorted in the descending order. The normalized variance distributions obtained via PNT and PCA are shown in Fig. 4(a). We also calculated the differential entropies of each dimension after PNT and PCA transformations. The differential entropies obtained from each scheme were sorted in descending order as well. Comparisons of differential entropies are shown in Fig. 4(b).

For feature selection, it is usually preferred to have energies concentrated at a few dimensions. The largest normalized variance ratio (1st dimension) in the PNT scheme is larger than that in the PCA scheme. A similar phenomenon is also observed for the differential entropy case. This indicates that PNT can make better energy concentration than PCA when applying them to decorrelate neutral vector variables.

In order to make fair comparisons for the aforementioned energy distributions, we defined a so-called "flatness coefficient (FC)" as the measurement. The FC for the normalized

[6]For the classification task, each data sample has a class label. These labels are known for the training set and unknown for the test set. For the clustering task, we assume that the class labels are the missing underlying variables that need to be estimated.

TABLE II

FC AND KLD COMPARISONS. $\boldsymbol{\alpha}_2$ IS THE SWITCHED VERSION OF $\boldsymbol{\alpha}_1$, WHERE THE SWITCHED ELEMENTS ARE HIGHLIGHTED WITH UNDERLINE. FC_V AND FC_E DENOTE FC CALCULATED BASED ON THE NORMALIZED VARIANCE RATIO AND DIFFERENTIAL ENTROPY, RESPECTIVELY. THE SAME DEFINITION APPLIES TO KLD

	FC_V		KLD_V		FC_E		KLD_E	
	PNT	PCA	PNT	PCA	PNT	PCA	PNT	PCA
$\boldsymbol{\alpha}_1$	0.1142	0.0801	0.2204	0.1726	0.0225	0.0201	0.0103	0.0091
$\boldsymbol{\alpha}_2$	0.0658	0.0790	0.1014	0.1697	0.0172	0.0185	0.0065	0.0077

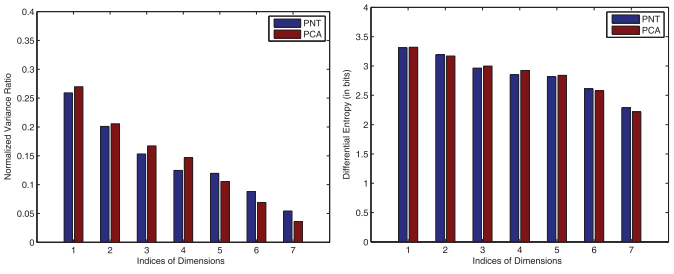
$\boldsymbol{\alpha}_1 = [3, 5, 15, 9, 12, 8, 7, 20]^{\mathrm{T}}$, $\boldsymbol{\alpha}_2 = [\underline{15}, 5, \underline{3}, 9, 12, 8, 7, 20]^{\mathrm{T}}$

variance ratio case is defined as the standard deviation as

$$\mathrm{FC} = \sqrt{\frac{1}{K-1}\sum_{k=1}^{K-1}(\mathrm{nvar}_k - \mathrm{nvar}_{\mathrm{mean}})^2} \tag{6}$$

where nvar_k is the normalized variance ratio for the kth dimension and $\mathrm{nvar}_{\mathrm{mean}}$ is the mean of all the ratios. A large FC means the energy distribution to be nonflat. Therefore, the larger the FC, the better the scheme. In addition to FC, the Kullback–Leibler divergence (KLD) of the energy distribution from the uniform distribution is also calculated as a metric to measure how likely the energy distribution is uniformly distributed. Larger KLD indicates better energy distribution. The ratios of variance/differential entropies are treated as a probability distribution in the KLD calculation. The FCs and KLDs for PNT and PCA are listed in Table II. In the first row of Table II, all the FCs and KLDs (under both the normalized variance ratio and differential entropy cases) obtained via PNT are larger than those obtained via PCA, respectively. With such observations, we conclude that PNT can yield a feature distribution, which is favorable in feature selection. Feature selection performance for real data will be presented in Section IV-C.

According to the nonlinear transformation procedure (the summation and division operations), the results of PNT depend on the order of dimensions in the neutral vector variable (however, PCA will not be affected by the permutation of dimensions). With the exchangeable property, any permuted version of a neutral vector variable can also be optimally decorrelated by PNT. Hence, the order of dimensions has significant effect on the resulting energy distribution. In order to demonstrate such effect, we repeat the abovementioned procedures with a Dirichlet distribution, where the parameter setting is $\boldsymbol{\alpha}_2 = [\underline{15}, 5, \underline{3}, 9, 12, 8, 7, 20]^{\mathrm{T}}$. This is a permuted version of $\boldsymbol{\alpha}_1 = [3, 5, 15, 9, 12, 8, 7, 20]^{\mathrm{T}}$ by switching the first and third elements. A set of 5000 data samples were generated from this Dirichlet distribution. The aforementioned energy distribution evaluation procedure was applied to these data. The effect of PNT and PCA on energy distribution are shown in Fig. 5, where the largest normalized variance ratio in the PNT scheme is smaller than that in the PCA scheme. Meanwhile, the differential entropy in PNT is also smaller than that obtained via PCA. Comparing with the procedure (with $\boldsymbol{\alpha}_1$), this observation yields opposite comparison results on energy distribution. Moreover, when comparing the FCs and KLDs (listed in the second row of Table VI), PNT



(a) (b)

Fig. 5. Effect of PNT versus PCA on energy distribution, with the first and third dimensions switched. (a) Comparisons of normalized variance ratio. (b) Comparisons of differential entropy.

underperforms PCA in resulting in a more favorable feature distribution.

With $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$, we have obtained opposite performance rankings of the two methods, only by permuting the neutral vector variable. This indicates that the permutation of the neutral vector variable (the order of neutral vector elements) has an effect on the energy distribution after applying PNT. It remains future work to design a strategy to find the optimal permuted version of a neutral vector variable such that the energy distribution obtained by PNT is the best among all the possible permutations.

3) Decorrelation Effect on Neutral-Like Vector Variables: Definition: A vector $\mathbf{x}$ of dimension $(K+1)$ is referred to as a neutral-like vector if $x_k, k = 1, 2 \ldots, K+1$ satisfies $x_k \geq 0$ and $\sum_{k=1}^{K+1} x_k = 1$.

The neutral vector is a subtype of compositional data. Compositional data are commonly present in real problems so testing the performance of PNT in such a more general data class is important. In this section, we extend our experiment to the compositional data. Compositional data may not satisfy neutral vector's neutrality properties, so we call this kind of vector variables neutral-like variables. In order to illustrate the decorrelation effect of PNT and PCA on neutral-like vector variables, we implement an experiment, which is similar to the experiment in Section IV-B1, on a neutral-like data set (i.e., logistic normal distributed data).

Definition: A $(K+1)$ part composition $\mathbf{x} = [x_1, \ldots, x_{K+1}]^{\mathrm{T}}$ is said to have a K dimensional additive logistic normal (LN) distribution $L_K(\mu, \Sigma)$ when $\mathbf{y} = [y_1, \ldots, y_K]^{\mathrm{T}}$ (where $y_i = \log(x_i/(x_{K+1})), i = 1, 2, \ldots, K$) follows a K-dimensional normal distribution $N_K(\mu, \Sigma)$.

The logistic normal distributed data can have either fully negative (FN) covariance matrix or partially negative (PN) covariance matrix, which is more flexible in topic model applications [71]. We generated two data sets, one with an FN covariance matrix and one with a PN covariance matrix, each with 400 samples ($N = 400$), from two logistic normal distributions with sample covariance matrices shown in Table III(a) and (e).

In order to investigate whether PCA and PNT can reduce the mutual dependence evaluated by DC, we first computed the DCs of the original data, PCA and PNT were then applied to transform the data separately, and, finally, the DCs of the

TABLE III

SAMPLE COVARIANCE MATRICES OF LN-DISTRIBUTED DATA AND DC MATRICES OF THE ORIGINAL AND THE TRANSFORMED DATA. (A) FN: COVARIANCE MATRIX (in $\times 10^{-3}$). (B) FN: DC MATRIX, ORIGINAL. (C) FN: DC MATRIX, WITH PNT. (D) FN: DC MATRIX, WITH PCA. (E) PN: COVARIANCE MATRIX (in $\times\ 10^{-3}$). (F) PN: DC MATRIX, ORIGINAL. (G) PN: DC MATRIX, WITH PNT. (H) PN: DC MATRIX, WITH PCA

(a)

$$\begin{bmatrix} 43.87 & -7.42 & -7.10 & -6.91 & -7.28 & -7.22 & -7.16 & -0.79 \\ & 44.73 & -7.24 & -7.15 & -7.45 & -7.23 & -7.42 & -0.81 \\ & & 43.43 & -6.87 & -7.20 & -7.00 & -7.20 & -0.81 \\ & & & 42.72 & -7.07 & -7.09 & -6.88 & -0.75 \\ & & & & 44.20 & -7.17 & -7.13 & -0.91 \\ & & & & & 43.63 & -7.11 & -0.81 \\ & & & & & & 43.80 & -0.88 \\ & & & & & & & 5.76 \end{bmatrix}$$

(b)

$$\begin{bmatrix} 1.00 & 0.26 & 0.16 & 0.21 & 0.19 & 0.19 & 0.17 \\ & 1.00 & 0.27 & 0.27 & 0.35 & 0.29 & 0.26 \\ & & 1.00 & 0.17 & 0.18 & 0.30 & 0.30 \\ & & & 1.00 & 0.22 & 0.17 & 0.17 \\ & & & & 1.00 & 0.24 & 0.22 \\ & & & & & 1.00 & 0.87 \\ & & & & & & 1.00 \end{bmatrix}$$

(c)

$$\begin{bmatrix} 1.00 & 0.06 & 0.08 & 0.10 & 0.21 & 0.07 & 0.23 \\ & 1.00 & 0.07 & 0.06 & 0.17 & 0.08 & 0.15 \\ & & 1.00 & 0.06 & 0.06 & 0.90 & 0.49 \\ & & & 1.00 & 0.08 & 0.13 & 0.08 \\ & & & & 1.00 & 0.07 & 0.27 \\ & & & & & 1.00 & 0.50 \\ & & & & & & 1.00 \end{bmatrix}$$

(d)

$$\begin{bmatrix} 1.00 & 0.53 & 0.31 & 0.29 & 0.23 & 0.18 & 0.18 \\ & 1.00 & 0.50 & 0.36 & 0.28 & 0.19 & 0.20 \\ & & 1.00 & 0.487 & 0.29 & 0.17 & 0.19 \\ & & & 1.00 & 0.46 & 0.26 & 0.25 \\ & & & & 1.00 & 0.47 & 0.45 \\ & & & & & 1.00 & 0.33 \\ & & & & & & 1.00 \end{bmatrix}$$

(e)

$$\begin{bmatrix} 0.54 & 0.22 & 0.01 & 0.06 & -0.87 & 0.01 & 0.00 & 0.01 \\ & 84.99 & 0.24 & -0.42 & -85.15 & 0.07 & 0.00 & 0.05 \\ & & 0.48 & 0.05 & -0.81 & 0.00 & 0.00 & 0.01 \\ & & & 4.72 & -4.54 & 0.04 & 0.00 & 0.10 \\ & & & & 91.71 & -0.14 & 0.00 & -0.19 \\ & & & & & 0.04×10^{-1} & 0.00 & 0.00 \\ & & & & & & 3.12×10^{-7} & 0.00 \\ & & & & & & & 0.01 \end{bmatrix}$$

(f)

$$\begin{bmatrix} 1.00 & 0.26 & 0.16 & 0.21 & 0.19 & 0.19 & 0.17 \\ & 1.00 & 0.27 & 0.27 & 0.35 & 0.29 & 0.26 \\ & & 1.00 & 0.17 & 0.18 & 0.30 & 0.30 \\ & & & 1.00 & 0.22 & 0.17 & 0.17 \\ & & & & 1.00 & 0.24 & 0.22 \\ & & & & & 1.00 & 0.87 \\ & & & & & & 1.00 \end{bmatrix}$$

(g)

$$\begin{bmatrix} 1.00 & 0.06 & 0.08 & 0.10 & 0.21 & 0.07 & 0.23 \\ & 1.00 & 0.07 & 0.06 & 0.17 & 0.08 & 0.15 \\ & & 1.00 & 0.06 & 0.06 & 0.90 & 0.49 \\ & & & 1.00 & 0.08 & 0.13 & 0.08 \\ & & & & 1.00 & 0.07 & 0.27 \\ & & & & & 1.00 & 0.50 \\ & & & & & & 1.00 \end{bmatrix}$$

(h)

$$\begin{bmatrix} 1.00 & 0.53 & 0.31 & 0.29 & 0.23 & 0.18 & 0.18 \\ & 1.00 & 0.50 & 0.36 & 0.28 & 0.19 & 0.20 \\ & & 1.00 & 0.487 & 0.29 & 0.17 & 0.19 \\ & & & 1.00 & 0.46 & 0.26 & 0.25 \\ & & & & 1.00 & 0.47 & 0.45 \\ & & & & & 1.00 & 0.33 \\ & & & & & & 1.00 \end{bmatrix}$$

TABLE IV

COMPARISONS OF AVERAGE DCS

Cov. matrix \ Average DC	Raw data	PNT	PCA
FN	0.18	0.12 (↓)	0.27 (↑)
PN	0.34	0.21 (↓)	0.46 (↑)

(a) (b) (c) (d)

Fig. 6. p-values of PNT and PCA on FN and PN logistic normal data, respectively. The significance level is 0.05. (a) FN and PNT. (b) FN and PCA. (c) PN and PNT. (d) PN and PCA.

transformed data obtained by PCA and PNT were computed. The DC matrices of the original and transformed data are shown in Table III(b)–(d) and (f)–(h), respectively.

It can be observed that, for neutral-like vector variables, most of the DCs were reduced by PNT. In contrast, most of the DCs were increased after PCA. The average DCs before and after transformation are listed in Table IV. From these results, we can conclude that PCA is incapable of reducing neutral-like vector variable's dependence as measured by DC, while PNT is capable of to some extent. Similar to Section IV-B1, we implemented a permutation test, and the experimental results of the p-value matrix are shown in Fig. 6, for PNT and PCA, respectively. From Fig. 6, we can observe that PNT outperforms PCA in terms of mutual independence measured by DC although some p-values are less than 0.05. (In contrast, for PCA, almost all p-values are equal to zero, which means that the null hypothesis of mutual independence was rejected.) With other logistic normal distribution's parameter settings, similar results can also be obtained.

In the abovementioned experiments, PNT can significantly reduce the DCs although the transformed data may not be fully mutually independent, and it outperforms PCA in this sense.

C. Comparisons With Real Data Evaluation

1) EEG Signal Classification: As a typical signal that can reflect the brain activities, the electroencephalogram (EEG) signal is the most studied and applied one in the design of a BCI system [72], [73]. A BCI system connects persons with external devices by recording and analyzing signals through a communication pathway. For those who suffer from neuromuscular diseases, a BCI system plays an important role in assisting them to communicate with others.

In order to classify the EEG signal properly, various types of features have been proposed. The marginal discrete wavelet transform (mDWT) vector, among others, has been widely adopted [74]–[76], as the elements in a DWT vector reveal features related to the transient nature of the EEG signal. To make the DWT vector insensitive to time alignment [74], the marginalization operation is applied. Therefore, the mDWT vector contains nonnegative elements and has unit l_1-norm, which is a type of "neutral-like" data.

The EEG signal data used in this article are from the BCI competition III [77]. The data set contains two types of actions: a subject performed an imagined movement of the left small finger or the tongue. The classification task is then a binary one. The electrical brain activity was picked up during these trials using an 8×8 ECoG platinum electrode grid that was placed on the contralateral (right) motor cortex. In total, 64 channels of EEG signals were obtained. For each channel, several trials of the imaginary brain activity were recorded. In total, 278 trials were recorded as the labeled training set, and 100 trials were recorded as the labeled test set. In both the training and test sets, the data are evenly recorded for each imaginary movement. All the data were labeled according to their ground truth. For each trial, 64 channel data of length 3000 samples were provided.

1) *Channel Selection:* The aforementioned EEG signals were recorded from 64 independent channels, and these channels were located on different positions of the scalp. Although it is commonly recognized that the classification accuracies are highly correlated with/dependent on the channels (i.e., recording positions), it is not clear which channels are more relevant to the imaginary tasks than the rest [78]. Hence, we applied two criteria,

namely, the Fisher ratio (FR) [79] and the generalization error estimation (GEE) [26], to select the relevant channels such that the irrelevant channels, which would be considered as noise for the task of classifications, can be discarded from the data set. The channels are ranked with FR or GEE, and the best m channels can be selected for the classification task. More details for channel selection can be found in [26], [55].

2) *Feature Selection:* The selection of relevant features that correlate with class label plays an essential role in EEG signal classification [26], [55], [80]. For each of the aforementioned channels, the dimensionality of the extracted mDWT feature vector is 5. Assuming that the mDWT feature vectors from one channel are neutral, we applied the PNT algorithm to transform the mDWT vectors into a set of 4-D vectors, each of which contains mutually independent scalar elements. The obtained four dimensions were sorted according to their variance in the descending order. With the new order, we selected the relevant D ($D \leq 4$) dimensions for the classification task. The abovementioned procedure was applied to both the mDWT vectors from the training and test sets.

With the abovementioned channel and feature selection procedures, the support vector machine (SVM) [81], [82] with radial basis function (RBF) kernel was applied to this binary classification task. With LIBSVM toolbox [81], we adjusted the parameters in the RBF-SVM so that the cross validation of training accuracy is the highest. We calculated the classification accuracies of the test data set to evaluate the feature selection strategy. To make comparisons with PCA, a conventional PCA was also applied to transform the mDWT vectors. The mDWT vectors in the test set were transformed with the eigenvectors obtained from the training set. The relevant dimensions were selected according to their variances (eigenvalues). An RBF-SVM was also designed and tuned for the PCA-selected features.

The classification accuracies are summarized in Table V. The classification results were obtained with the top m channels (ranked via FR or GEE). For each channel, the most relevant D features (ranked via variance) were selected. In total, we obtained $(m \times D)$-dimensional feature vector to train the RBF-SVM. It can be observed that the RBF-SVM+PNT yields the highest recognition accuracies, both for FR case and GEE cases.

Fig. 7 shows the classification results obtained with top m channels and different amounts of relevant dimensions. For each channel, the most relevant D dimensions were selected and concatenated to an $(m \times D)$-dimensional supervector as a classification feature. Generally speaking, channel selection improves the classification results by skipping the irrelevant channels. From Fig. 7(a)–(f), it can be observed that the RBF-SVM+PNT method outperforms both the benchmark RBF-SVM and the RBF-SVM+PCA method when m is smaller than 17, 26, 23, 27, 29, and 27. The highest classification rates for different methods all happen in this range. The abovementioned facts demonstrate that the proposed nonlinear transformation strategy can indeed improve the classification accuracy by decorrelation and feature selection. Moreover,

TABLE V

SUMMARY OF BEST CLASSIFICATION RATES. $D = 4$ IS THE CASE WITH LINEAR/NONLINEAR TRANSFORMATION BUT WITHOUT FEATURE SELECTION. m DENOTES THE NUMBER OF CHANNELS THAT HAVE BEEN SELECTED ACCORDING TO FR OR GEE

Channel selection	Classifier	Best performance
FR	RBF-SVM (no transformation)	$72\%(m = 25)$
	RBF-SVM+PCA ($D = 4$)	$73\%(m = 6)$
	RBF-SVM+PCA ($D = 3$)	$72\%(m = 5, 6, 7)$
	RBF-SVM+PCA ($D = 2$)	$73\%(m = 7)$
	RBF-SVM+PCA ($D = 1$)	$59\%(m = 7)$
	RBF-SVM+PNT ($D = 4$)	$\mathbf{75\%(m = 17)}$
	RBF-SVM+PNT ($D = 3$)	$74\%(m = 15, 18, 19, 20)$
	RBF-SVM+PNT ($D = 2$)	$\mathbf{75\%(m = 19, 20)}$
	RBF-SVM+PNT ($D = 1$)	$69\%(m = 11)$
GEE	RBF-SVM (no transformation)	$72\%(m = 12, 17, 27)$
	RBF-SVM+PCA ($D = 4$)	$72\%(m = 10, 11)$
	RBF-SVM+PCA ($D = 3$)	$72\%(m = 11)$
	RBF-SVM+PCA ($D = 2$)	$72\%(m = 11, 12)$
	RBF-SVM+PCA ($D = 1$)	$59\%(m = 22, 26, 27, 28)$
	RBF-SVM+PNT ($D = 4$)	$74\%(m = 10, 25)$
	RBF-SVM+PNT ($D = 3$)	$75\%(m = 4, 5, 7)$
	RBF-SVM+PNT ($D = 2$)	$\mathbf{77\%(m = 4)}$
	RBF-SVM+PNT ($D = 1$)	$71\%(m = 16)$

it also shows that, for neutral-like data, the PNT-based nonlinear transformation is more preferable than the conventionally applied PCA-based linear transformation. As m increases, the classification performance decreases due to the fact that more noisy channels are involved in the classifier. Interestingly, when only one dimension ($D = 1$) is selected from each channel [see Fig. 7(g) and (h)], both the RBF-SVM+PNT and the RBF-SVM+PCA perform worse than the benchmark method. This is because these two methods ignored too many dimensions so that valuable information for classification is also discarded. However, the RBF-SVM+PNT still has a higher classification rate than that obtained by the RBF-SVM+PCA. This further supports our hypothesis that the PNT-based nonlinear transformation method is better than the PCA-based linear transformation for the neutral-like data.

In summary, with the nonnegative and unit l_1-norm properties, we assumed that the mDWT vectors are neutral-like vectors and applied PNT and PCA, separately, to them as feature selection methods. Experimental results demonstrate that feature selection via PNT significantly improves the classification accuracy, for both FR and GEE cases.

2) Reconstruction of LPC Model: In speech coding, efficient transmission of the LPC model plays an essential role [83]. There exist many representations of the LPC parameters, such as the reflection coefficients (RCs), the arcsine RCs (ASRC), the log-area ratios (LARs), the immittance spectral frequencies (ISFs), and the line spectral frequencies (LSFs) [25], [83]. The LSF representation, among others, is the most commonly used one because it has a relatively uniform spectral sensitivity [84], [85]. By explicitly exploiting the boundary and the order properties, the LSF vector can be linearly transformed into the so-called LSF differences vector (ΔLSF). The ΔLSF vector has less variability, and the range is more compact compared with the absolute LSF

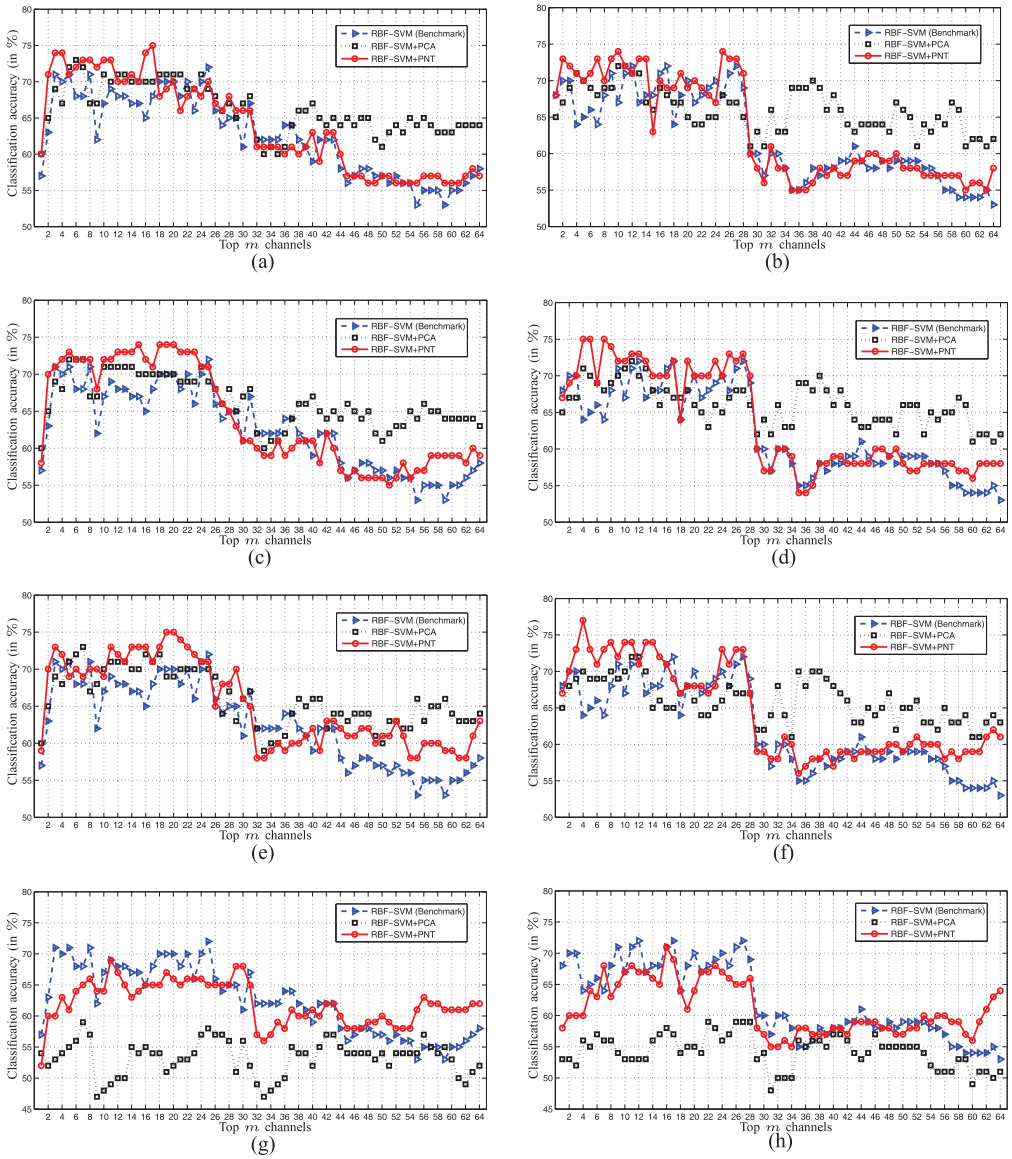


Fig. 7. Classification accuracy comparisons of RBF-SVM (benchmark), RBF-SVM+PCA, and RBF-SVM+PNT. The RBF-SVM+PCA and the RBF-SVM+PNT results in Fig. 7(c)–(f) have been reported in [26]. (a) Channel selection with FR and $D = 4$. (b) Channel selection with GEE and $D = 4$. (c) Channel selection with FR and $D = 3$. (d) Channel selection with GEE and $D = 3$. (e) Channel selection with FR and $D = 2$. (f) Channel selection with GEE and $D = 2$. (g) Channel selection with FR and $D = 1$. (h) Channel selection with GEE and $D = 1$.

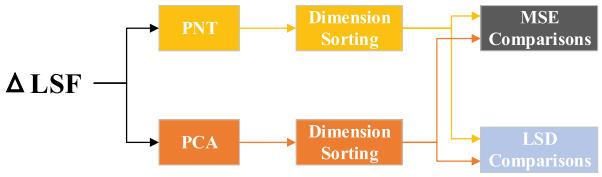


Fig. 8. Diagram of LPC reconstruction performance comparisons.

value [25], [39], [86]. It contains nonnegative elements and has unit l_1-norm, and it is natural to model the underlying distribution of the ΔLSF vectors with a DMM [25]. Recent studies demonstrated that, with DMM modeling, the performance of related applications can be significantly improved, such as LSF quantization in transmission [25], [39] and LSF vector estimation in packet networks [38]. This is because the ΔLSF vector has neutral-like property, and the Dirichlet variable is a typical neutral vector.

In this article, we study the performance of PNT for the LPC model reconstruction. The TIMIT data set [87] was used for evaluation. The speech data from the TIMIT database have a sampling rate of 16 kHz, and LPC parameters were extracted and transformed to LSF/ΔLSF vector.[7] With window length of 25 ms and step size of 20 ms, approximately, 964k LSF/ΔLSF vectors were extracted from the database. The Hann window was applied to each frame.

According to [25], the LSF vector is 16-D, and the corresponding ΔLSF vector is 17-D (with degrees of

[7]The details of transformation from LPC to LSF/ΔLSF (and its inverse transformation) can be found in [25].

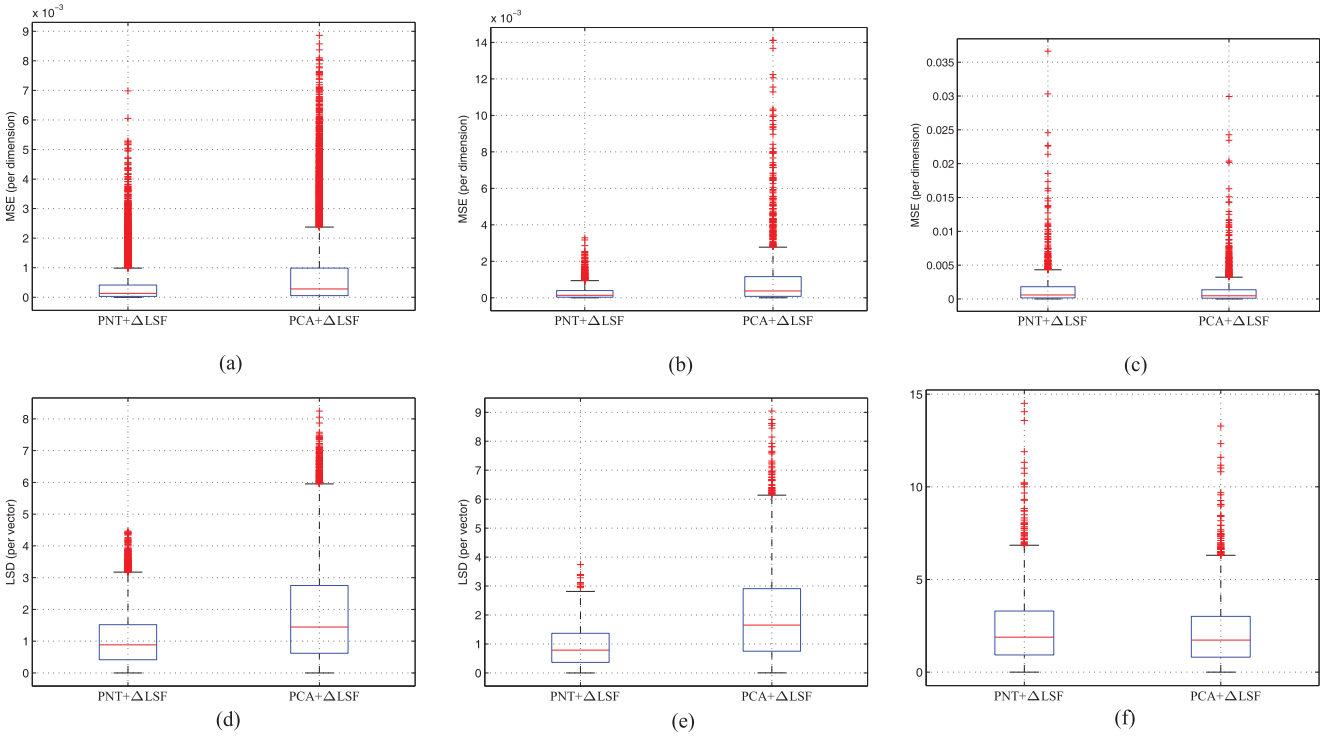


Fig. 9. Comparisons of LPC reconstruction performances via boxplots. The missing dimensions are ♯1, ♯8, and ♯16, respectively. (a) Reconstruction performance with first dimension (the one with the largest variance) estimated. Boxplot of mse. (b) Reconstruction performance with the eighth dimension estimated. Boxplot of mse. (c) Reconstruction performance with 16th (the one with the smallest variance) dimension estimated. Boxplot of mse. (d) Reconstruction performance with first dimension (the one with the largest variance) estimated. Boxplot of LSD. (e) Reconstruction performance with the eighth dimension estimated. Boxplot of LSD. (f) Reconstruction performance with the 16th dimension (the one with the smallest variance) estimated. Boxplot of LSD.

TABLE VI

FC AND KLD COMPARISONS FOR ENERGY DISTRIBUTIONS OF TRANSFORMED ΔLSF VECTORS. FC_V AND FC_E DENOTE FC CALCULATED BASED ON THE NORMALIZED VARIANCE RATIO AND DIFFERENTIAL ENTROPY, RESPECTIVELY. THE SAME DEFINITION APPLIES TO KLD

FC_V		KLD_V		FC_E		KLD_E	
PNT	PCA	PNT	PCA	PNT	PCA	PNT	PCA
0.0393	0.0372	**0.3884**	0.3130	**0.0148**	0.0054	**0.0389**	0.0039

freedom $K = 16$). For the ΔLSF parameters, we applied the proposed PNT algorithm to obtain a set of 16-D scalars. With the assumption that the ΔLSF vector is neutral vectors, the resultant scalars are mutually independent. These scalars are sorted in the descending order according to their variances. The FC and KLD comparisons for energy distribution yielded by applying PNT and PCA on ΔLSF parameters, respectively, are listed in Table VI.

We evaluate the robustness of the decorrelation strategy with the following steps.

1) The ΔLSF vectors are decorrelated by the PNT method, and the decorrelated dimensions are sorted according to their variances in descending order.
2) Assume that some dimensions are missing during transmission, and we replace these dimensions by their corresponding mean values.
3) Reconstruct the LPC model and evaluate the distortion between the original model and the reconstructed one.

Two metrics, namely, the mean squared error (mse) and the log spectral distortion (LSD), are used to measure the distortion. The mse between the original ΔLSF vector and the reconstructed one is calculated as

$$\text{mse} = \frac{1}{N}\sum_{n=1}^{N}(\Delta\text{LSF}_n - \widehat{\Delta\text{LSF}}_n)^2 \tag{7}$$

where ΔLSF_n and $\widehat{\Delta\text{LSF}}_n$ denote the original and reconstructed ΔLSF vectors, respectively. With the original/reconstructed ΔLSF vectors, the corresponding LPC models can be obtained. The LSD between the original and reconstructed LPC models is evaluated as

$$\text{LSD}_n = \sqrt{\frac{1}{F_s}\int_0^{F_s}\left[10\log_{10}P_n(f) - 10\log_{10}\widehat{P}_n(f)\right]^2 df} \tag{8}$$

where n is the index of the vector, F_s is the sampling frequency in Hz, and $P_n(f)$ and $\widehat{P}_n(f)$ are the original and quantized LPC power spectra of the nth vector. $P(f)$ and $\widehat{P}(f)$ are calculated as

$$P_n(f) = 1/|A_n(e^{j2\pi f/F_s})|^2, \quad A(z) = 1 + \sum_{k=1}^{K} a_k z^{-k}$$
$$\widehat{P}_n(f) = 1/|\widehat{A}_n(e^{j2\pi f/F_s})|^2, \quad \widehat{A}(z) = 1 + \sum_{k=1}^{K} \widehat{a}_k z^{-k} \tag{9}$$

TABLE VII

COMPARISONS OF RECONSTRUCTION PERFORMANCE OF THE LPC MODEL WITH DIFFERENT DECORRELATION METHODS. FOR THE STUDENT'S t-TEST, THE SIGNIFICANT LEVEL FOR THE NULL HYPOTHESIS THAT PNT AND PCA ARE SIMILAR METHODS IS 0.05

Metric	Method	Missing Dimension							
		♯1	♯2	♯3	♯4	♯5	♯6	♯7	♯8
MSE (in 10^{-4})	PNT	**3.31**	**3.38**	**3.26**	**2.53**	**2.92**	**3.25**	**1.58**	**3.00**
	PCA	5.50	16.58	8.97	5.14	6.71	11.30	5.38	9.55
LSD (in dB)	PNT	**1.06**	**0.93**	**0.91**	**0.82**	**1.01**	**0.92**	**0.88**	**0.92**
	PCA	1.48	2.78	1.93	1.57	1.65	2.11	1.49	2.03
p-value	MSE	8.77×10^{-23}	2.35×10^{-119}	1.05×10^{-56}	2.06×10^{-32}	1.35×10^{-45}	1.44×10^{-72}	1.56×10^{-59}	7.46×10^{-68}
	LSD	4.19×10^{-38}	3.53×10^{-266}	7.14×10^{-137}	5.85×10^{-102}	5.15×10^{-70}	2.06×10^{-158}	4.71×10^{-78}	5.81×10^{-156}
Metric	**Method**	**Missing Dimension**							
		♯9	♯10	♯11	♯12	♯13	♯14	♯15	♯16
MSE (in 10^{-4})	PNT	**7.67**	**7.70**	9.05	**10.02**	24.07	25.42	72.62	14.00
	PCA	19.99	21.95	**3.04**	17.20	**22.55**	**15.55**	**4.64**	**12.00**
LSD (in dB)	PNT	**2.22**	**1.69**	1.79	**1.57**	3.36	3.46	5.50	2.31
	PCA	3.04	2.25	**1.19**	2.71	**3.27**	**2.72**	**1.45**	**2.15**
p-value	MSE	8.86×10^{-72}	2.15×10^{-8}	8.46×10^{-62}	3.41×10^{-24}	1.52×10^{-24}	3.60×10^{-69}	3.05×10^{-163}	5.3×10^{-3}
	LSD	9.76×10^{-50}	5.16×10^{-27}	1.38×10^{-50}	1.58×10^{-86}	1.64×10^{-24}	1.18×10^{-33}	0	5.4×10^{-3}

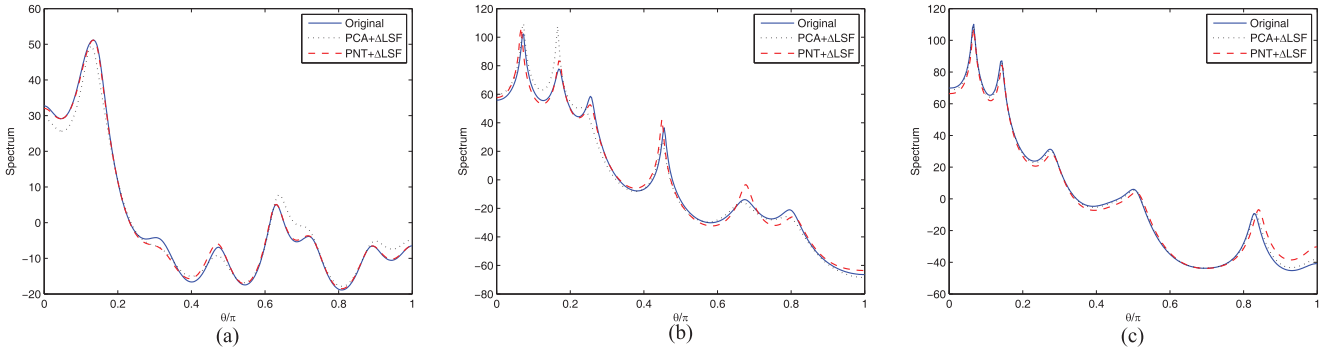


Fig. 10. Illustration of LPC spectrum reconstructions. The reported LSD value is for the selected frame (LPC vector). (a) Reconstructed spectrum with missing dimension ♯1. $\text{LSD}_{\text{PNT}} = 0.3352$, and $\text{LSD}_{\text{PCA}} = 1.0896$. (b) Reconstructed spectrum with missing dimension ♯8. $\text{LSD}_{\text{PNT}} = 0.8252$, and $\text{LSD}_{\text{PCA}} = 2.1809$. (c) Reconstructed spectrum with missing dimension ♯16. $\text{LSD}_{\text{PNT}} = 1.2125$, and $\text{LSD}_{\text{PCA}} = 0.9831$.

where a_k, $k = 1, \ldots, K$ are the corresponding LPC parameters. From the speech quality point of view, the LSD is the most preferred objective distortion measure in the literature [85], both for narrowband and wideband speech [88], [89]. In order to make comparisons with PCA, we applied PCA to the ΔLSF vectors as the method of decorrelation. After transformation, the aforementioned approaches were conducted to evaluate the reconstruction performance achieved by PCA. Fig. 8 shows the diagram of such procedures.

The overall reconstruction performances are summarized in Table VII, and the corresponding (selected) boxplots are illustrated in Fig. 9. We randomly selected 20 000 ΔLSF vectors for evaluation and conducted 50 rounds of such simulations. The mean values are reported in this article.

It can be observed that, during transmission, decorrelation of the ΔLSF vector can significantly remove the correlation among elements, and therefore, the effect of packet loss (i.e., subvector/element loss in our case) is also reduced. With mse and LSD as the measurements for error, applying PNT to the ΔLSF vector achieves smaller error than PCA for a wide range of missing dimensions (i.e., ♯1–♯10 and ♯12). For the other dimension indices, PNT performs slightly worse than PCA although these dimensions are corresponding to relatively smaller variances (the dimensions are sorted according to their variances in descending order). This is due to the nonlinear transformation procedure of PNT. As demonstrated in Fig. 2, the elements with larger indices in the transformed vector $\mathbf{u}$ have relatively smaller variances (the distribution range is relatively compact). When taking the inverse PNT, the error caused by estimating these elements will be propagated in the following operations.[8] Hence, estimation errors in the dimensions with larger indices will have more influence than those occurring in the dimensions with smaller indices. Although PNT has the error propagation effect for the dimensions with larger indices, it still performs well for the decorrelation of the ΔLSF vectors in most cases. How to efficiently decrease the error propagation effect is an open problem for our future studies.

In order to demonstrate the statistical significance, we conducted the student's t-test for the null hypothesis that the two decorrelation methods are similar. This null hypothesis is rejected, and the p-values are listed in Table VII as well. Fig. 10 illustrates the comparisons of the original LPC spectrum, the reconstructed LPC spectrum via PNT, and the reconstructed LPC spectrum via PCA.

[8]With the example in Fig. 2, $x_{1,1} = u_1 \cdot u_4 \cdot u_6$, $x_{3,1} = u_2 \cdot (1 - u_4) \cdot u_6$, and $x_{5,1} = u_3 \cdot u_5 \cdot (1 - u_6)$. Therefore, the estimation error occurred in u_6 will have "global" effect, while the error in u_1 or u_2 only has "local" effect.

From the abovementioned analysis, we can conclude that when packet loss occurred and there is no estimation available, PNT outperforms PCA in the LPC model transmission.

V. Conclusion

A neutral vector variable is a typical non-Gaussian vector variable. By explicitly exploring the neutral properties, the so-called PNT has already been proposed for the purpose of efficient and effective decorrelation of the neutral vector variable. In this article, we studied and compared the PNT method with the conventionally applied PCA and ICA methods. Theoretical analysis and comparisons showed that PNT has the lowest computational complexity among all the three methods. It can also transform a highly negatively correlated neutral vector variable into a set of mutually independent scalar variables, as well as preserve the bounded support property. With real-life data evaluation, the advantages of the PNT method in EEG signal feature selection and speech model reconstruction were demonstrated with extensive experiments.

There remain several open problems for future work: 1) propose a strategy to find the optimal permuted version for neutral vector variables; 2) study the error propagation control strategy for the PNT method such that the reconstruction performance can be further improved; 3) similar to the improved version of PCA or ICA, an improved PNT is expected to be proposed such that the overall performance can also be improved; and 4) investigate more real-applications with the proposed PNT and its variants.

References

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[2] J. Yan, J. Wang, H. Zha, X. Yang, and S. Chu, "Consistency-driven alternating optimization for multigraph matching: A unified approach," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 994–1009, Mar. 2015.

[3] J. Yan, M. Cho, H. Zha, X. Yang, and S. M. Chu, "Multi-graph matching via affinity optimization with graduated consistency regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 6, pp. 1228–1242, Jun. 2016.

[4] A. Kessy, A. Lewin, and K. Strimmer, "Optimal whitening and decorrelation," 2015, *arXiv:1512.00809*. [Online]. Available: http://arxiv.org/abs/1512.00809

[5] R. Wang, *Introduction to Orthogonal Transforms*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[6] I. Jolliffe, *Principal Component Analysis* (Springer Series in Statistics). New York, NY, USA: Springer, 2002.

[7] C. Alzate and J. A. K. Suykens, "Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 2, pp. 335–347, Feb. 2010.

[8] K. W. Jorgensen and L. K. Hansen, "Model selection for Gaussian kernel PCA denoising," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 1, pp. 163–168, Jan. 2012.

[9] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "Subspace learning from image gradient orientations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2454–2466, Dec. 2012.

[10] D. Li, H. Zhou, and K.-M. Lam, "High-resolution face verification using pore-scale facial features," *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2317–2327, Aug. 2015.

[11] J. He, E.-L. Tan, and W.-S. Gan, "Linear estimation based primary-ambient extraction for stereo audio signals," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 2, pp. 505–517, Feb. 2014.

[12] J. Yan *et al.*, "Trace-oriented feature analysis for large-scale text data dimension reduction," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 7, pp. 1103–1117, Jul. 2011.

[13] G. Chabriel, M. Kleinsteuber, E. Moreau, H. Shen, P. Tichavsky, and A. Yeredor, "Joint matrices decompositions and blind source separation: A survey of methods, identification, and applications," *IEEE Signal Process. Mag.*, vol. 31, no. 3, pp. 34–43, May 2014.

[14] W. Zhou, M. Yang, X. Wang, H. Li, Y. Lin, and Q. Tian, "Scalable feature matching by dual cascaded scalar quantization for image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 159–171, Jan. 2016.

[15] S. Park, E. Serpedin, and K. Qaraqe, "Gaussian assumption: The least favorable but the most useful [lecture notes]," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 183–186, May 2013.

[16] Z. Ma and A. Leijon, "Bayesian estimation of beta mixture models with variational inference," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2160–2173, Nov. 2011.

[17] J. Liu, Y. Jiang, Z. Li, X. Zhang, and H. Lu, "Domain-sensitive recommendation with user-item subgroup analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 4, pp. 939–950, Apr. 2016.

[18] E. A. Houseman, K. T. Kelsey, J. K. Wiencke, and C. J. Marsit, "Cell-composition effects in the analysis of DNA methylation array data: A mathematical perspective," *BMC Bioinf.*, vol. 16, no. 1, p. 95, 2015.

[19] N. Mohammadiha, R. Martin, and A. Leijon, "Spectral domain speech enhancement using HMM state-dependent super-Gaussian priors," *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 253–256, Mar. 2013.

[20] J. Taghia, Z. Ma, and A. Leijon, "Bayesian estimation of the von-Mises Fisher mixture model with variational inference," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 9, pp. 1701–1715, Sep. 2014.

[21] C. Chen, W. Buntine, N. Ding, L. Xie, and L. Du, "Differential topic models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 2, pp. 230–242, Feb. 2015.

[22] T. M. Nguyen and Q. M. J. Wu, "A nonsymmetric mixture model for unsupervised image segmentation," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 751–765, Apr. 2013.

[23] Z. Ma, "Non-Gaussian statistical models and their applications," Ph.D. dissertation, School Elect. Eng., KTH-Roy. Inst. Technol., Stockholm, Sweden, 2011.

[24] Z. Ma and A. E. Teschendorff, "A variational Bayes beta mixture model for feature selection in dna methylation studies," *J. Bioinf. Comput. Biol.*, vol. 11, no. 4, Aug. 2013, Art. no. 1350005.

[25] Z. Ma, A. Leijon, and W. B. Kleijn, "Vector quantization of LSF parameters with a mixture of Dirichlet distributions," *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 9, pp. 1777–1790, Sep. 2013.

[26] Z. Ma, Z.-H. Tan, and J. Guo, "Feature selection for neutral vector in EEG signal classification," *Neurocomputing*, vol. 174, pp. 937–945, Jan. 2016.

[27] J. V. Stone, *Independent Component Analysis: A Tutorial Introduction*. Cambridge, MA, USA: MIT Press, 2004.

[28] H. Nguyen and R. Zheng, "Binary independent component analysis with or mixtures," *IEEE Trans. Signal Process.*, vol. 59, no. 7, pp. 3168–3181, Jul. 2011.

[29] V. Laparra, G. Camps-Valls, and J. Malo, "Iterative Gaussianization: From ICA to random rotations," *IEEE Trans. Neural Netw.*, vol. 22, no. 4, pp. 537–549, Apr. 2011.

[30] K.-C. Kwak and W. Pedrycz, "Face recognition using an enhanced independent component analysis approach," *IEEE Trans. Neural Netw.*, vol. 18, no. 2, pp. 530–541, Mar. 2007.

[31] I. Santamaria, "Handbook of blind source separation: Independent component analysis and applications (Common, P. and Jutten, C.; 2010) [book review]," *IEEE Signal Process. Mag.*, vol. 30, no. 2, pp. 133–134, Mar. 2013.

[32] L. R. Arnaut and C. S. Obiekezie, "Source separation for wideband energy emissions using complex independent component analysis," *IEEE Trans. Electromagn. Compat.*, vol. 56, no. 3, pp. 559–570, Jun. 2014.

[33] R. J. Connor and J. E. Mosimann, "Concepts of independence for proportions with a generalization of the Dirichlet distribution," *J. Amer. Stat. Assoc.*, vol. 64, no. 325, pp. 194–206, Mar. 1969.

[34] I. R. James and J. E. Mosimann, "A new characterization of the Dirichlet distribution through neutrality," *Ann. Statist.*, vol. 8, no. 1, pp. 183–189, Jan. 1980.

[35] J.-T. Chien and C.-H. Chueh, "Topic-based hierarchical segmentation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 55–66, Jan. 2012.

[36] Y. Zhuang, H. Gao, F. Wu, S. Tang, Y. Zhang, and Z. Zhang, "Probabilistic word selection via topic modeling," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 6, pp. 1643–1655, Jun. 2015.

[37] N. Bouguila and D. Ziou, "A Dirichlet process mixture of generalized Dirichlet distributions for proportional data modeling," *IEEE Trans. Neural Netw.*, vol. 21, no. 1, pp. 107–122, Jan. 2010.

[38] Z. Ma, R. Martin, J. Guo, and H. Zhang, "Nonlinear estimation of missing ΔLSF parameters by a mixture of Dirichlet distributions," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 6929–6933.

[39] Z. Ma, S. Chatterjee, W. Bastiaan Kleijn, and J. Guo, "Dirichlet mixture modeling to estimate an empirical lower bound for LSF quantization," *Signal Process.*, vol. 104, pp. 291–295, Nov. 2014.

[40] A. Sakowicz and J. Wesołowski, "Dirichlet distribution through neutralities with respect to two partitions," *J. Multivariate Anal.*, vol. 129, pp. 1–15, Aug. 2014.

[41] R. Huang, G. Yu, Z. Wang, J. Zhang, and L. Shi, "Dirichlet process mixture model for document clustering with feature partition," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 8, pp. 1748–1759, Aug. 2013.

[42] A. M. Dai and A. J. Storkey, "The supervised hierarchical Dirichlet process," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 2, pp. 243–255, Feb. 2015.

[43] M. Bkassiny, S. K. Jayaweera, and Y. Li, "Multidimensional Dirichlet process-based non-parametric signal classification for autonomous self-learning cognitive radios," *IEEE Trans. Wireless Commun.*, vol. 12, no. 11, pp. 5413–5423, Nov. 2013.

[44] X.-L. Huang, F. Hu, J. Wu, H.-H. Chen, G. Wang, and T. Jiang, "Intelligent cooperative spectrum sensing via hierarchical Dirichlet process in cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 5, pp. 771–787, May 2015.

[45] W. Fan, N. Bouguila, and D. Ziou, "Variational learning for finite Dirichlet mixture models and applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 5, pp. 762–774, May 2012.

[46] X. Zhou, J. Mateos, F. Zhou, R. Molina, and A. K. Katsaggelos, "Variational Dirichlet blur kernel estimation," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5127–5139, Dec. 2015.

[47] Z. Ma, P. K. Rana, J. Taghia, M. Flierl, and A. Leijon, "Bayesian estimation of Dirichlet mixture model with variational inference," *Pattern Recognit.*, vol. 47, no. 9, pp. 3143–3157, Sep. 2014.

[48] S. Xiao, J. Yan, M. Farajtabar, L. Song, X. Yang, and H. Zha, "Learning time series associated event sequences with recurrent point process networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 10, pp. 3124–3136, Oct. 2019.

[49] H. Mei and J. Eisner, "The neural hawkes process: A neurally self-modulating multivariate point process," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6757–6767.

[50] A. Miller, L. Bornn, R. P. Adams, and K. Goldsberry, "Factorized point process intensities: A spatial analysis of professional basketball," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 235–243.

[51] T. T. Pham, S. H. Rezatofighi, I. Reid, and T.-J. Chin, "Efficient point process inference for large-scale object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2837–2845.

[52] N. Mehrasa, A. A. Jyothi, T. Durand, J. He, L. Sigal, and G. Mori, "A variational auto-encoder model for stochastic point processes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3165–3174.

[53] M. M. Moradi and J. Mateu, "First- and second-order characteristics of spatio-temporal point processes on linear networks," *J. Comput. Graph. Statist.*, pp. 1–21, Dec. 2019.

[54] H.-Y. Wang, Q. Yang, H. Qin, and H. Zha, "Dirichlet component analysis: Feature extraction for compositional data," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 1128–1135.

[55] Z. Ma, J.-H. Xue, A. Leijon, Z.-H. Tan, Z. Yang, and J. Guo, "Decorrelation of neutral vector variables: Theory and applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 129–143, Jan. 2018.

[56] P. Howard, D. W. Apley, and G. Runger, "Distinct variation pattern discovery using alternating nonlinear principal component analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 156–166, Jan. 2018.

[57] N. Shahid, N. Perraudin, V. Kalofolias, G. Puy, and P. Vandergheynst, "Fast robust PCA on graphs," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 4, pp. 740–756, Jun. 2016.

[58] S. Z. Rizvi, J. Mohammadpour, R. Toth, and N. Meskin, "A kernel-based PCA approach to model reduction of linear parameter-varying systems," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 5, pp. 1883–1891, Sep. 2016.

[59] Y. Xiao, Z. Zhu, Y. Zhao, Y. Wei, and S. Wei, "Kernel reconstruction ICA for sparse representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 6, pp. 1222–1232, Jun. 2015.

[60] R. K. S. Hankin, "A generalization of the Dirichlet distribution," *J. Stat. Softw.*, vol. 33, no. 11, pp. 1–18, 2010.

[61] B. A. Frigyik, A. Kapila, and M. R. Gupta, "Introduction to the Dirichlet distribution and related processes," Dept. Elect. Eng., Univ. Washington, Seattle, WA, USA, Tech. Rep. 2010-0006, 2010.

[62] V. B. Balakrish, "Dirichlet distribution," *A Primer on Statistical Distributions*. Hoboken, NJ, USA: Wiley, 2005, p. 274.

[63] T. Minka. *The Lightspeed MATLAB Toolboxes*. Accessed: Mar. 20, 2020. [Online]. Available: https://github.com/tminka/lightspeed

[64] F. T. Luk and S. Qiao, *A Fast Singular Value Algorithm for Hankel Matrices*. Boston, MA, USA: American Mathematical Society, 2003, pp. 169–177.

[65] S. Shwartz, M. Zibulevsky, and Y. Y. Schechner, "Independent component analysis and blind signal separation," in *ICA Using Kernel Entropy Estimation With NlogN Complexity*. Berlin, Germany: Springer, 2004, pp. 422–429.

[66] G. J. Székely, M. L. Rizzo, and N. K. Bakirov, "Measuring and testing independence by correlation of distances," *Ann. Statist.*, vol. 35, no. 6, pp. 2769–2794, 2007.

[67] G. J. Székely and M. L. Rizzo, "Brownian distance covariance," *Ann. Appl. Statist.*, vol. 3, no. 4, pp. 1236–1265, Dec. 2009.

[68] K. Pearson, "Notes on regression and inheritance in the case of two parents," *Proc. Roy. Soc. London*, vol. 58, nos. 347–352, pp. 240–242, 1895.

[69] R. R. Wilcox, *Introduction to Robust Estimation and Hypothesis Testing*. New York, NY, USA: Academic, 2005.

[70] G. J. Székely and M. L. Rizzo, "On the uniqueness of distance covariance," *Statist. Probab. Lett.*, vol. 82, no. 12, pp. 2278–2282, Dec. 2012.

[71] D. M. Blei and J. D. Lafferty, "Correlated topic models," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2006, p. 147.

[72] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi, "A review of classification algorithms for EEG-based brain-computer interfaces," *J. Neural Eng.*, vol. 4, no. 2, p. R1, 2007.

[73] H. Cecotti and A. Graser, "Convolutional neural networks for P300 detection with application to brain-computer interfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 433–445, Mar. 2011.

[74] A. Subasi, "EEG signal classification using wavelet feature extraction and a mixture of expert model," *Expert Syst. Appl.*, vol. 32, no. 4, pp. 1084–1093, May 2007.

[75] Z. Ma, Z.-H. Tan, and S. Prasad, "EEG signal classification with super-Dirichlet mixture model," in *Proc. IEEE Stat. Signal Process. Workshop (SSP)*, Aug. 2012, pp. 440–443.

[76] Z. Xu, S. MacEachern, and X. Xu, "Modeling non-Gaussian time series with nonparametric Bayesian model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 2, pp. 372–382, Feb. 2015.

[77] *BCI Competition III*. Accessed: Mar. 20, 2020. [Online]. Available: http://www.bbci.de/competition/iii

[78] T. N. Lal *et al.*, "Support vector channel selection in BCI," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 1003–1010, Jun. 2004.

[79] W. Malina, "On an extended Fisher criterion for feature selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-3, no. 5, pp. 611–614, Sep. 1981.

[80] H.-I. Suk and S.-W. Lee, "A novel Bayesian framework for discriminative feature extraction in brain-computer interfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 286–299, Feb. 2013.

[81] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 17, pp. 1–27, 2011.

[82] X. Huang, L. Shi, and J. A. K. Suykens, "Support vector machine classifier with pinball loss," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 984–997, May 2014.

[83] P. Vary and R. Martin, *Digital Speech Transmission: Enhancement, Coding and Error Concealment*. Chichester, U.K.: Wiley, 2006.

[84] J. Li, N. Chaddha, and R. M. Gray, "Asymptotic performance of vector quantizers with a perceptual distortion measure," *IEEE Trans. Inf. Theory*, vol. 45, no. 4, pp. 1082–1091, May 1999.

[85] W. B. Kleijn and A. Ozerov, "Rate distribution between model and signal," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust. (WASPAA)*, Oct. 2007, pp. 243–246.

[86] F. K. Soong and B.-H. Juang, "Optimal quantization of LSP parameters using delayed decisions," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 1990, pp. 185–188.

[87] *Acoustic-Phonetic Continuous Speech Corpus*, DARPA-TIMIT, NIST, Gaithersburg, MD, USA, 1990.

[88] S. Chatterjee and T. V. Sreenivas, "Predicting VQ performance bound for LSF coding," *IEEE Signal Process. Lett.*, vol. 15, pp. 166–169, 2008.

[89] L. A. Ekman, W. B. Kleijn, and M. N. Murthi, "Regularized linear prediction of speech," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 1, pp. 65–73, Jan. 2008.

Zhanyu Ma (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the KTH Royal Institute of Technology, Stockholm, Sweden, in 2011.

From 2012 to 2013, he has been a Post-Doctoral Research Fellow with the School of Electrical Engineering, KTH Royal Institute of Technology. He has been an Associate Professor with the Beijing University of Posts and Telecommunications, Beijing, China, from 2014 to 2019. He has been an Adjunct Associate Professor with Aalborg University, Aalborg, Denmark, since 2015. He is currently a Full Professor with the Beijing University of Posts and Telecommunications. His research interests include pattern recognition and machine learning fundamentals with a focus on applications in computer vision, multimedia signal processing, and data mining.

Xiaoou Lu received the M.Sc. degree in computational statistics and machine learning from the University College London, London, U.K., in 2014, and the M.Sc. degree in mathematical finance from the University of York, York, U.K., in 2011. He is currently pursuing the Ph.D. degree with the Department of Statistical Science, University College London.

His research interests include compositional data analysis and machine learning.

Jiyang Xie (Student Member, IEEE) received the B.E. degree in information engineering from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2017, where he is currently pursuing the Ph.D. degree.

His research interests include pattern recognition and machine learning fundamentals with a focus on applications in image processing, data mining, and deep learning.

Zhen Yang (Member, IEEE) received the Ph.D. degree in signal processing from the Beijing University of Posts and Telecommunications, Beijing, China.

He is currently a Full Professor with the College of Computer Science, Faculty of Information Technology, Beijing University of Technology. He has published more than 30 articles in highly ranked journals and top conference proceedings. His research interests include data mining, machine learning, trusted computing, and content security.

Dr. Yang is also a Senior Member of the Chinese Institute of Electronics.

Jing-Hao Xue received the Dr.Eng. degree in signal and information processing from Tsinghua University, Beijing, China, in 1998, and the Ph.D. degree in statistics from the University of Glasgow, Glasgow, U.K., in 2008.

Since 2008, he has been a Lecturer and a Senior Lecturer with the Department of Statistical Science, University College London, London, U.K. His current research interests include statistical classification, high-dimensional data analysis, computer vision, and pattern recognition.

Zheng-Hua Tan (Senior Member, IEEE) has been a Professor with the Department of Electronic Systems, Aalborg University, Aalborg, Denmark, since May 2001. His research interests include speech and speaker recognition, noise-robust speech processing, multimedia signal and information processing, human–robot interaction, and machine learning.

Dr. Tan has served as an Editorial Board Member/Associate Editor for *Computer Speech and Language*, *Digital Signal Processing*, and *Computers & Electrical Engineering*. He was a Lead Guest Editor of the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING.

Bo Xiao was born in Changyi, Shandong, China, in 1975. He received the B.S. degree in image transmission and processing, the M.S. degree in computer science, and the Ph.D. degree in signal and information processing from the Beijing University of Posts and Telecommunications, Beijing China, in 1998, 2005, and 2009, respectively. He has been with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China, since 1998, where he has been an Associate Professor since 2010. From September 2018 to August 2019, he was a Visiting Scholar with the University of Windsor, Windsor, ON, Canada. He has coauthored five books and more than 40 journal articles and conference papers. His research interests include data mining, pattern recognition, deep learning, computer vision, and intelligent wireless communication.

Jun Guo received the B.E. and M.E. degrees from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1982 and 1985, respectively, and the Ph.D. degree from the Tohoku Gakuin University, Sendai, Japan, in 1993.

He is currently a Professor and the Vice President of BUPT. He has published over 200 articles on the journals and conferences, including *Science*, *Scientific Reports* (Nature), the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, *Pattern Recognition*, AAAI, CVPR, ICCV, and SIGIR. His research interests include pattern recognition theory and application, information retrieval, content-based information security, and bioinformatics.