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. Additionally, its neutral properties make it significantly different from the commonly studied vector variables (*e.g.*, Gaussian vector variables). Due to the aforementioned properties, the conventionally applied linear transformation approaches (*e.g.*, principal component analysis (PCA), 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, the parallel nonlinear transformation (PNT), for decorrelating neutral vector variables. In this paper, 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*—Neutral vector variable, neutrality, decorrelation, nonlinear transformation, non-Gaussian**

I. Introduction

Decorrelation of a random vector variable plays an essential role in multivariate data analysis, signal processing, pattern recognition and machine learning [1]–[4]. It can transform a correlated vector variable into a set of mutually uncorrelated scalar/sub-vector 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 whitening of a vector variable, there are many linear transforms including the Mahalanobis transform, Cholesky decomposition and eigen-decomposition of the precision matrix (*i.e.* the inverse of the covariance matrix) [2]. However, using whitening transforms for decorrelation has a limitation. After whitening transformation, every uncorrelated scalar variable has unit variance; this means 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 [2], but this also means the distribution of the variance over the elements of the vector variable does not change after transformation. A distributional change like 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.

Karhunen-Loève transform, also better-known as principal component analysis (PCA) [6], among others, is an 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 a space spanned by the covariance matrix's eigenvectors [1]. If we treat the eigenvalue as the "energy" of 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], [14], source coding [15], [16], etc.

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 [17]. For example, the grey or color pixel values in image processing [18], the

Z. Ma, J. Xie, B. Xiao, and J. Guo are with the Pattern Recognition and Intelligent System Lab., Beijing University of Posts and Telecommunications, Beijing, China.

X. Lu and J.-H. Xue are with the Department of Statistical Science, University College London, London, United Kingdom.

Z. Yang is with the College of Computer Science, Faculty of Information Technology, Beijing University of Technology, Beijing, China.

Z.-H. Tan is with the Department of Electronic Systems, Aalborg University, Aalborg, Denmark.

The corresponding authors are Zhanyu Ma and Zhen Yang.

rating scores to an item in a recommendation system [19], [20], and the genome-wide DNA methylation level value in bioinformatics [21], [22] 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 [23]. The l_2 norms of the spatial fading correlation and the yeast gene expressions [24] 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 [25], which are nonnegative and have a l_1 norm equal to one. The aforementioned data all have asymmetric or constrained distributions [26] and they do not match the natural definition of Gaussian distribution (*i.e.*, the definition domain is unbounded and the distribution shape is symmetric). Hence, these data are non-Gaussian distributed [27]. Recently, it has been demonstrated in many studies that explicitly utilizing the non-Gaussian characteristics can significantly improve the practical performance [18], [23], [24], [27]–[29]. Applying PCA to non-Gaussian distributed data can only get uncorrelated but *not* independent variables, and therefore, the consequent performance, which requires the variables' mutual independence, will be decreased [27], [29], [30].

Independent component analysis (ICA) can decorrelate any vector variable (observed data) into a set of mutually independent scalar variables (data sources) [31], [32], 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 [33]. ICA has been widely applied in several fields, such as face recognition [34], blind source separation [35], and wireless communications [36].

Neutral vector variables [37], [38] are a typical non-Gaussian vector variable. The non-Gaussian properties of a neutral vector variables are: 1) all the elements in a neutral vector variable are nonnegative; and 2) the l_1 norm of a neutral vector variable equals one. 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 [37]. To describe the characteristics of the proportional data/compositional data, neutral vector variable has been extensively applied in document analysis [39], [40], image processing [41], and speech signal processing [42], [43]. A typical distribution for modeling the distribution of a neutral vector variable is the Dirichlet distribution [44]. As a classical method for constructing non-parametric models, several Dirichlet distribution based Dirichlet process models have been proposed for the purpose of feature selection [45], [46], cognitive radios [47], [48], *etc*. 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 [29], [49], [50]. Bayesian estimation of DMM with variational inference, which provides analytically tractable solution for parameter estimation, has been proposed in [51].

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 modelling data with three types: 1) Sequential data in continuous time [52], [53], 2) spatial representations of locations [54], [55], and 3) spatio-temporal data [56], [57], while the neutral vector variable can be applied for modelling not only spatio-temporal 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, Dirichlet component analysis (DCA) was proposed to replace PCA for Dirichlet variable decorrelation and dimension reduction [58]. Although DCA preserves the relevant constraints among the elements of the vector variable, it can only guarantee that the mapped component 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 [38], we have proposed a special nonlinear transformation strategy, namely the parallel nonlinear transformation (PNT), to decorrelate the neutral vector variable into a set of mutually independent scalar variables or a set of mutually independent sub-vector variables [29], [59]. The PNT has been successfully applied in many areas, such as speech linear predictive coding (LPC) model quantization [29] and feature selection for EEG signal classification[2] [30].

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; 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; 2) neither PCA or ICA can perserve bounded support property while PNT preserves it; 4) eigenvalue analysis is the prerequisite for conducting linear transformation in PCA,

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

[2] Part of the work in the submitted manuscript (The RBF-SVM+PCA and the RBF-SVM+PNT results in Fig. 7(c)-7(f)) has been published in [30]. Focusing on the general framework for decorrelating completely neutral vector, this paper introduces the concept of 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 [30] 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.

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 [60], fast robust PCA [61], kernel PCA [62], kernel ICA [63], and binary ICA [32]. However, the purpose of this paper 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 work can be summarized as follow:

- We provide a through study of the so-called PNT decorrelation strategy for non-Gaussian neutral vector variable, which is optimal, preserves the non-Gaussian properties, and does not need to calculate the statistical properties during operation.
- 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 paper are organized as follows: in Sec. II, we briefly introduce the neutral vector and its related concepts and properties. The details of PNT, PCA, and ICA will be provided in Sec. III. Extensive comparisons among these methods, with theoretical analysis and data evaluations, will be conducted in Sec. IV. We will draw some conclusions in Sec. V.

II. Neutral Vector Variable

Assuming 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 $\mathbf{w}_k = \frac{1}{1-s_k}\mathbf{x}_{k2}$ (*i.e.*, $\mathbf{x}_{k1} \perp \mathbf{w}_k$), for $1 \le k \le K$ [37], [38], 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 [37], [64]. 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 [59]:

***Property** 2.1 (Mutual Independence):* For completely neutral vector variable $\mathbf{x}$, define $z_k = \frac{x_k}{1-s_{k-1}}$ and $z_1 = x_1$, we have $z_1, z_2, \ldots, z_K$ are mutually independent.

***Property** 2.2 (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** 2.3 (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.

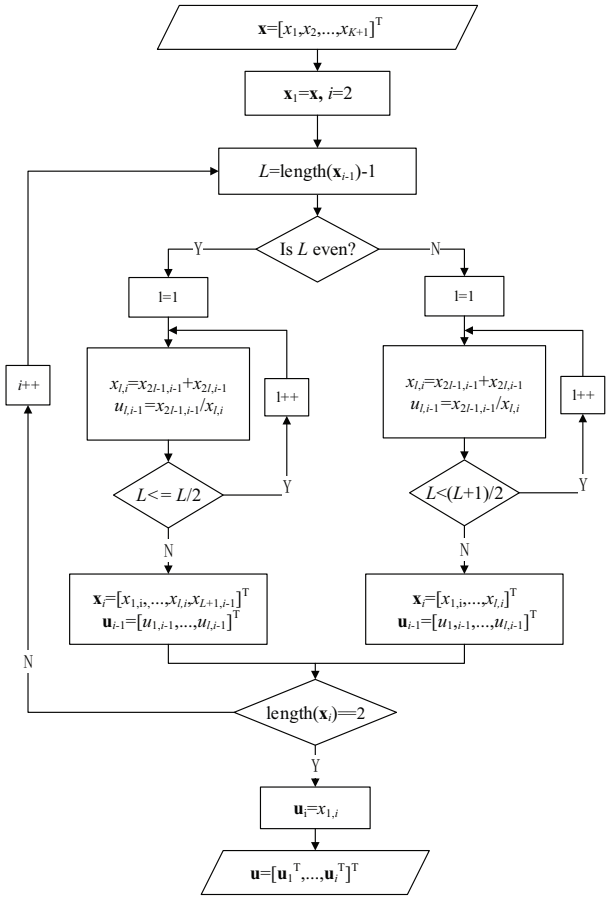


Fig. 1. Flow chart of PNT.

The Dirichlet variable is a typical case of neutral vector variable [1], [65], 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

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

The covariance matrix of the Dirichlet distribution is [66]

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

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

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

III. Decorrelation Approaches

Both PCA and ICA are commonly known for the communities of signal processing, pattern recognition, machine learning, *etc*. Due to the limitation of space, we skip the introduction to the technical details of these two methods and

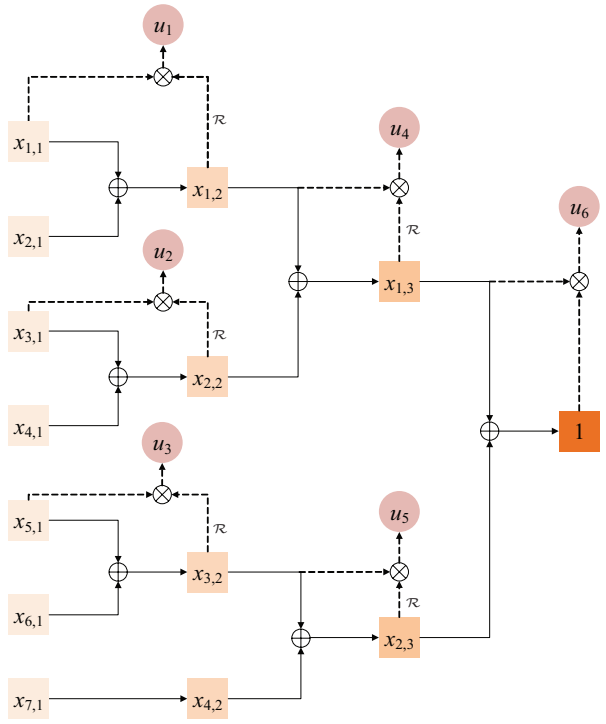


Fig. 2. An example of PNT with $K = 6$. The transformed coefficients are $u_1 = x_{1,1}/x_{1,2}$, $u_2 = x_{3,1}/x_{2,2}$, $u_3 = x_{5,1}/x_{3,2}$, $u_4 = x_{1,2}/x_{1,3}$, $u_5 = x_{3,2}/x_{2,3}$, and $u_6 = x_{1,3}$. $\mathcal{R}$ represents the reciprocal operation.

focus on PNT in this paper. Detailed information of PCA and ICA can be found in, *e.g.*, [1].

With the aforementioned properties, a neutral vector variable exhibits a particular type of statistical independence among its elements [37]. In order to explicitly explore such type of independence, we proposed a so-called parallel nonlinear transformation (PNT) scheme to transform a neutral vector variable into a set of mutually independent scalar variables [59]. 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 mutually independence has been presented in [59]. An example for applying PNT to a 7-dimensional (*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 [59].

Note that the proposed PNT scheme can be simply implemented by iterative element-wise 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 Discussions

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 Sec. III, a neutral vector can be transformed to a set of *mutually independent* scalar variables by PNT, in a nonliner 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 our best knowledge, the fastest method for eigenvalue analysis so-far is the method proposed by Luk et al. [68]. With the method proposed in [68], 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 robust performance. However, one drawback of the algorithms designed for carrying out ICA is the high computational load required in implementation [69]. Generally speaking, algorithms for ICA requires centering, whitening, and dimension reduction as the preprocessing steps for the purpose of facilitating calculation. As mentioned in [33], 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.

[4]According to T. Minka's Lightspeed Matlab toolbox [67].

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	$\times$	$\mathcal{O}(MK^2)$	$\times$

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 neutral vector is its bounded support property. It is usually required such property can be preserved after transformation. The proposed PNT method meets this requirement with its division operation. Neither PCA nor ICA can perserve 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) Discussions: The summary of the aforementioned theoretical comparisons are listed in Tab. 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.

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 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 parts.

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) [70], [71] was calculated to evaluate the mutual independence after decorrelation. The conventionally used Pearson correlation coefficient [72], [73] 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 [74]. 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

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

Taking the doubly centered distances, we have

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

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

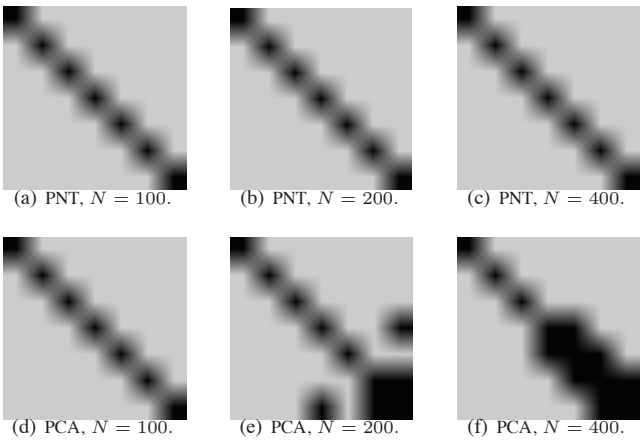
$$DC = \sqrt{\frac{\sum_{i,j=1}^{N} A_{ij} B_{ij}}{\sqrt{\sum_{i,j=1}^{N} A_{ij}^2 \sum_{i,j=1}^{N} B_{ij}^2}}}. \tag{5}$$

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:

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 two steps a large number of times, the p-value for this permutation test is the proportion of the DC values in step 2 that 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 paper.

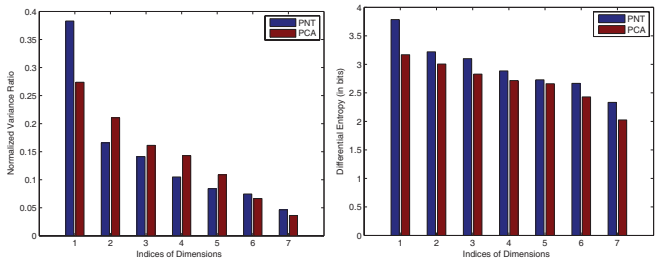
The decorrelation performance, with different amounts of generated data, are illustrated in Fig. 3. PNT and PCA were applied to transform the generated vectors, respectively. The



(a) PNT, $N = 100$. (b) PNT, $N = 200$. (c) PNT, $N = 400$.
(d) PCA, $N = 100$. (e) PCA, $N = 200$. (f) PCA, $N = 400$.

Fig. 3. Decorrelation performances of PNT and PCA measured with p-values. See text for details.

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



(a) Comparisons of normalized variance ratio. (b) Comparisons of differential entropy.

Fig. 4. Effect of PNT vs. PCA on energy distribution.

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 dependency) or grey (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), 3(b), and 3(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 Fig. 3(f).)

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 multi-modality 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 paper, 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 Sec. IV-B1, we generated $5,000$ 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 firstly normalized to have a unit l_1-norm and then sorted in 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).

[6]For classification task, each data sample has a class label. These labels are known for the training set and unknown for the test set. For 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

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

For feature selection, it is usually preferred to have energies concentrated at a few dimensions. The largest normalized variance ratio (1^{st} dimension) in the PNT scheme is larger than that in the PCA scheme. 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 variance ratio case is defined as the standard deviation as

$$\text{FC} = \sqrt{\frac{1}{K-1}\sum_{k=1}^{K-1}(nvar_k - nvar_{\text{mean}})^2}, \tag{6}$$

where $nvar_k$ is the normalized variance ratio for the k^{th} dimension and $nvar_{\text{mean}}$ is the mean of all the ratios. A large FC means the energy distribution to be non-flat. 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 probability distribution in the KLD calculation. The FCs and KLDs for PNT and PCA are listed in Tab. II. In the first row of Tab. 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 Sec. 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 have significant effect on the resulting energy distribution. In order to demonstrate such effect, we repeat the above procedures

TABLE III
SAMPLE COVARIANCE MATRICES OF LN-DISTRIBUTED DATA AND DC MATRICES OF THE ORIGINAL AND THE TRANSFORMED DATA.

(a) FN: covariance matrix (in $\times 10^{-3}$)

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

(b) FN: DC matrix, original

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

(c) FN: DC matrix, with PNT

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

(d) FN: DC matrix, with PCA

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

(e) PN: covariance matrix (in $\times 10^{-3}$)

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

(f) PN: DC matrix, original

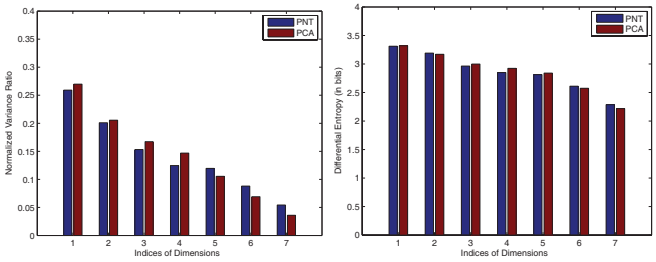
$$\begin{bmatrix} 1.00 & 0.26 & 0.16 & 0.21 & 0.19 & 0.19 & 0.17 \\ & 1.00 & 0.27 & 0.27 & 0.35 & 0.29 & 0.26 \\ & & 1.00 & 0.17 & 0.18 & 0.30 & 0.30 \\ & & & 1.00 & 0.22 & 0.17 & 0.17 \\ & & & & 1.00 & 0.24 & 0.22 \\ & & & & & 1.00 & 0.87 \\ & & & & & & 1.00 \end{bmatrix}$$

(g) PN: DC matrix, with PNT

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

(h) PN: DC matrix, with PCA

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



(a) Comparisons of normalized variance ratio. (b) Comparisons of differential entropy.

Fig. 5. Effect of PNT vs. PCA on energy distribution, with the 1^{st} and the 3^{rd} dimensions switched.

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 1^{st} and the 3^{rd} elements. A set of $5,000$ 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 Tab. VI), PNT 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 neutral vector variable (the order of neutral vector elements) has 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 $x_k, k = 1, 2..., K+1$, satisfies $x_k \geq 0$ and $\sum_{k=1}^{K+1} x_k = 1$.

Neutral vector is a sub-type 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 Sec. IV-B1, on a neutral-like dataset (*i.e.*, logistic normal distributed data).

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

The logistic normal distributed data can have an either fully negative (FN) covariance matrix or partially negative (PN) covariance matrix, which is more flexible in topic model applications [75]. 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 Tab. III(a) and III(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 transformed data obtained by PCA and PNT were computed. The DC matrices of the original and transformed data are shown in Tab. III(b)- III(d) and Tab. III(f)- III(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 Tab. 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. Similarly to Sec. 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,

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) FN & PNT (b) FN & PCA (c) PN & PNT (d) PN & PCA

Fig. 6. The p-values of PNT and PCA on FN and PN logistic normal data, respectively. The significance level is 0.05.

for PCA, almost all p-values are equal to zero, which means the null hypothesis of mutual independence was rejected.) With other logistic normal distribution's parameter settings, similar results can also be obtained.

In the experiments above, 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 [76], [77]. A BCI system connects persons with the 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 [78]–[80], 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 [78], 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 paper are from the BCI competition III [81]. The data set contains two types of actions: a subject performed imagined movement of 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 which 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 set and test set, 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 $3,000$ samples were provided.

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 $(D = 1)$	$71\%(m = 16)$

- 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 [82]. Hence, we applied two criteria, namely the Fisher ratio (FR) [83] and the generalization error estimation (GEE) [30], 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 [30], [59].
- Feature Selection
  Selection of relevant features that correlate with class label plays an essential role in EEG signal classification [30], [59], [84]. For each of the aforementioned channels, the dimensionality of the extracted mDWT feature vector is 5. Assuming the mDWT feature vectors from one channel is neutral, we applied the PNT algorithm to transform the mDWT vectors into a set of 4-dimensional vectors, each of which contains mutually independent scalar elements. The obtained 4 dimensions were sorted according to their variance in descending order. With the new order, we selected the relevant D ($D \leq 4$) dimensions for classification task. The above procedure was applied to both the mDWT vectors from the training and test sets.

With the above channel and feature selection procedures, the support vector machine (SVM) [85], [86] with radial basis function (RBF) kernel was applied to this binary classification task. With LIBSVM toolbox [85], we adjusted the parameters in the RBF-SVM so that the cross validation of training accu-

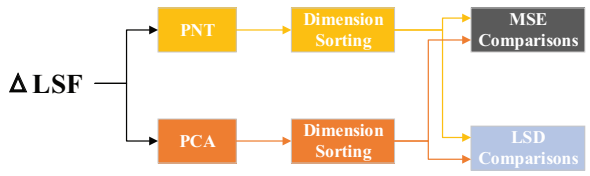


Fig. 8. Diagram of LPC reconstruction performance comparisons.

racy is the highest. We calculated the classification accuracies of the test dataset 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). A RBF-SVM was also designed and tuned for the PCA-selected features.

The classification accuracies are summarized in Tab. 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.

vspace-5mm Figure 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 super-vector as classification feature. Generally speaking, channel selection improves the classification results by skipping the irrelevant channels. From Fig. 7(a), 7(b), 7(c), 7(d), 7(e), and 7(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 above facts demonstrate that the proposed nonlinear transformation strategy can indeed improve the classification accuracy by decorrelation and feature selection. Moreover, 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 7(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 are also discarded. However, the RBF-SVM+PNT still has 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.

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

2) Reconstruction of LPC Model: In speech coding, efficient transmission of the linear predictive coding (LPC) model plays an essential role [87]. There exist many representations of the LPC parameters, such as the reflection coefficients (RC), the arcsine reflection coefficients (ASRC), the log-area ratios (LAR), the immittance spectral frequencies (ISF), and the line spectral frequencies (LSF) [29], [87]. The LSF representation, among others, is the most common used one, because it has a relatively uniform spectral sensitivity [88], [89]. By explicitly exploiting the boundary and the order properties, the LSF vector can be linearly transformed to the so-called LSF differences vector (ΔLSF). The ΔLSF vector has less variability and the range is more compact compared to the absolute LSF value [29], [90], [91]. 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 Dirichlet mixture model (DMM) [29]. Recent studies demonstrated that, with DMM modeling, the performance of related applications can be significantly improved, such as LSF quantization in transmission [29], [91], and LSF vector estimation in packet networks [42]. This is because that the ΔLSF vector has neutral-like property and Dirichlet variable is a typical neutral vector.

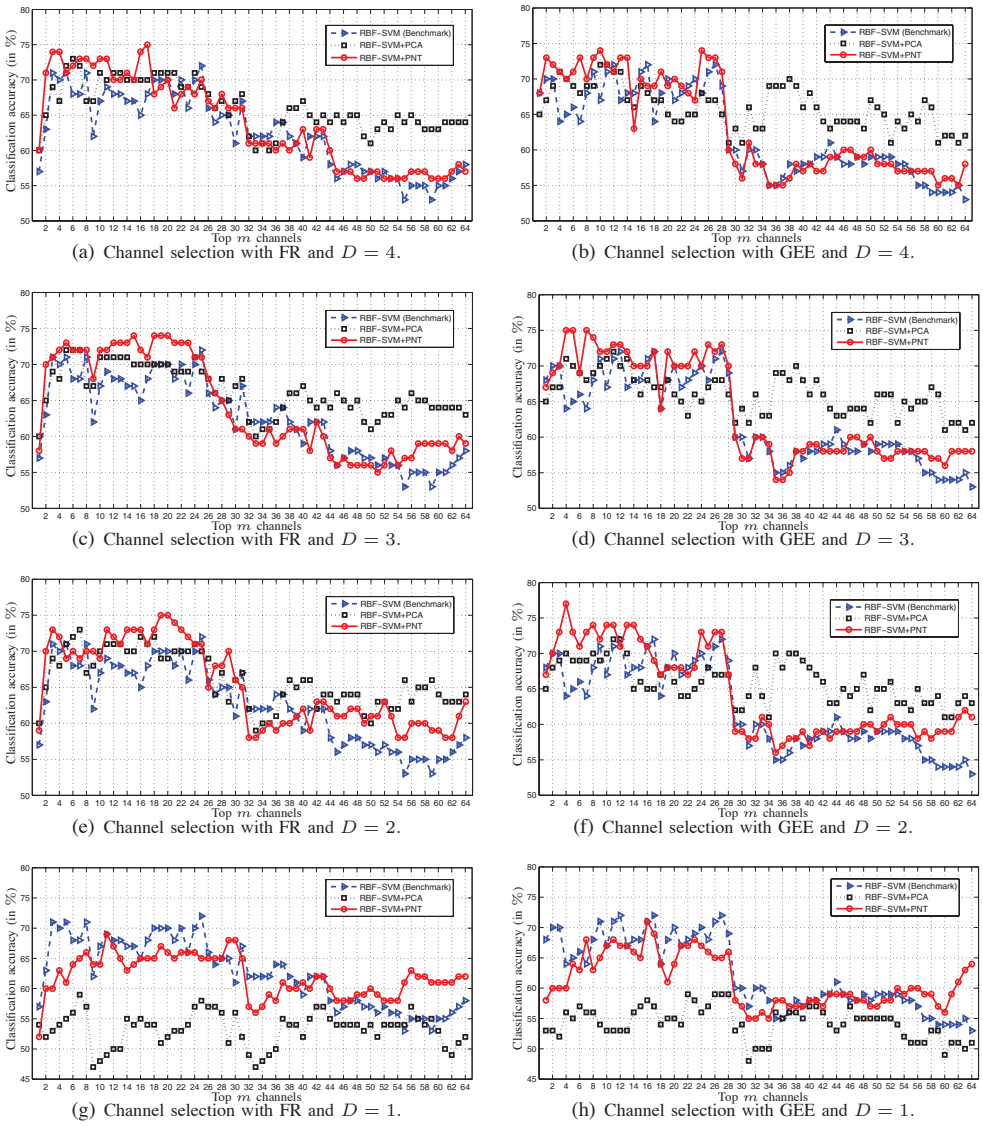
In this paper, we study the performance of PNT for the LPC model reconstruction. The TIMIT dataset [92] was used for evaluation. The speech data from the TIMIT database have a sampling rate of 16 kHz and LPC parameters were extracted and transformed to LSF/ΔLSF vector[7]. With window length of 25 milliseconds and step size of 20 milliseconds, approximate $964k$ LSF/ΔLSF vectors were extracted from the database. The Hann window was applied to each frame.

According to [29], the LSF vector is 16-dimensional and the corresponding ΔLSF vector is 17-dimensional (with degrees of freedom $K = 16$). For the ΔLSF parameters, we applied the proposed PNT algorithm to obtain a set of 16-dimensional scalars. With the assumption that the ΔLSF vector are neutral vectors, the resultant scalars are mutually independent. These scalars are sorted in descend order according to their variances. The FC and KLD comparisons for energy distribution yield by applying PNT and PCA on ΔLSF parameters, respectively, are listed in Tab. VI.

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

1) The ΔLSF vectors are decorrelated by the PNT method, the decorrelated dimensions are sorted according to their

[7]The details of transformation from LPC to LSF/ΔLSF (and its inverse transformation) can be found in [29].



(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. 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), 7(d), 7(e), and 7(f) have been reported in [30].

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, $P_n(f)$ 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

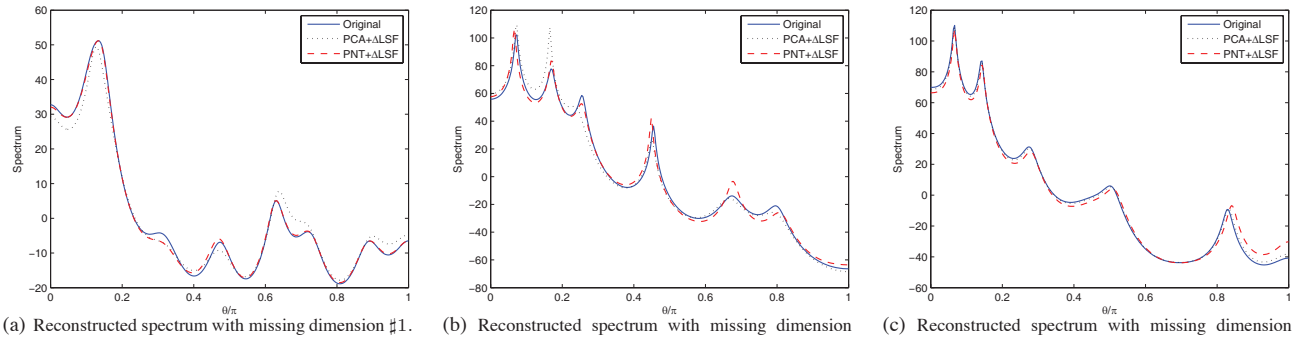
$$\begin{aligned} P_n(f) &= 1/|A_n(e^{j2\pi f/F_s})|^2,\ A(z) = 1 + \sum_{k=1}^{K} a_k z^{-k} \\ \widehat{P}_n(f) &= 1/|\widehat{A}_n(e^{j2\pi f/F_s})|^2,\ \widehat{A}(z) = 1 + \sum_{k=1}^{K} \widehat{a}_k z^{-k}, \end{aligned} \tag{9}$$

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 [89], both for narrowband and wideband speech [93], [94]. 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. Figure 8 shows the diagram of such procedures.

The overall reconstruction performances are summarized in Tab. VII and the corresponding (selected) boxplots are illustrated in Fig. 10. We randomly selected $20,000$ ΔLSF vectors

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}



(a) Reconstructed spectrum with missing dimension ♯1. $\text{LSD}_{\text{PNT}} = 0.3352$, $\text{LSD}_{\text{PCA}} = 1.0896$.

(b) Reconstructed spectrum with missing dimension ♯8. $\text{LSD}_{\text{PNT}} = 0.8252$, $\text{LSD}_{\text{PCA}} = 2.1809$.

(c) Reconstructed spectrum with missing dimension ♯16. $\text{LSD}_{\text{PNT}} = 1.2125$, $\text{LSD}_{\text{PCA}} = 0.9831$.

Fig. 9. Illustration of LPC spectrum reconstructions. The reported LSD value is for the selected frame (LPC vector).

for evaluation and conducted 50 rounds of such simulations. The mean values are reported in this paper.

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 occurred in the dimensions with smaller indices. Although PNT has the error propagation effect for the dimensions with larger indices, it still performs well for decorrelation of the ΔLSF vectors in most cases. How to efficiently decrease the error propagation effect is an open problem for our future studies.

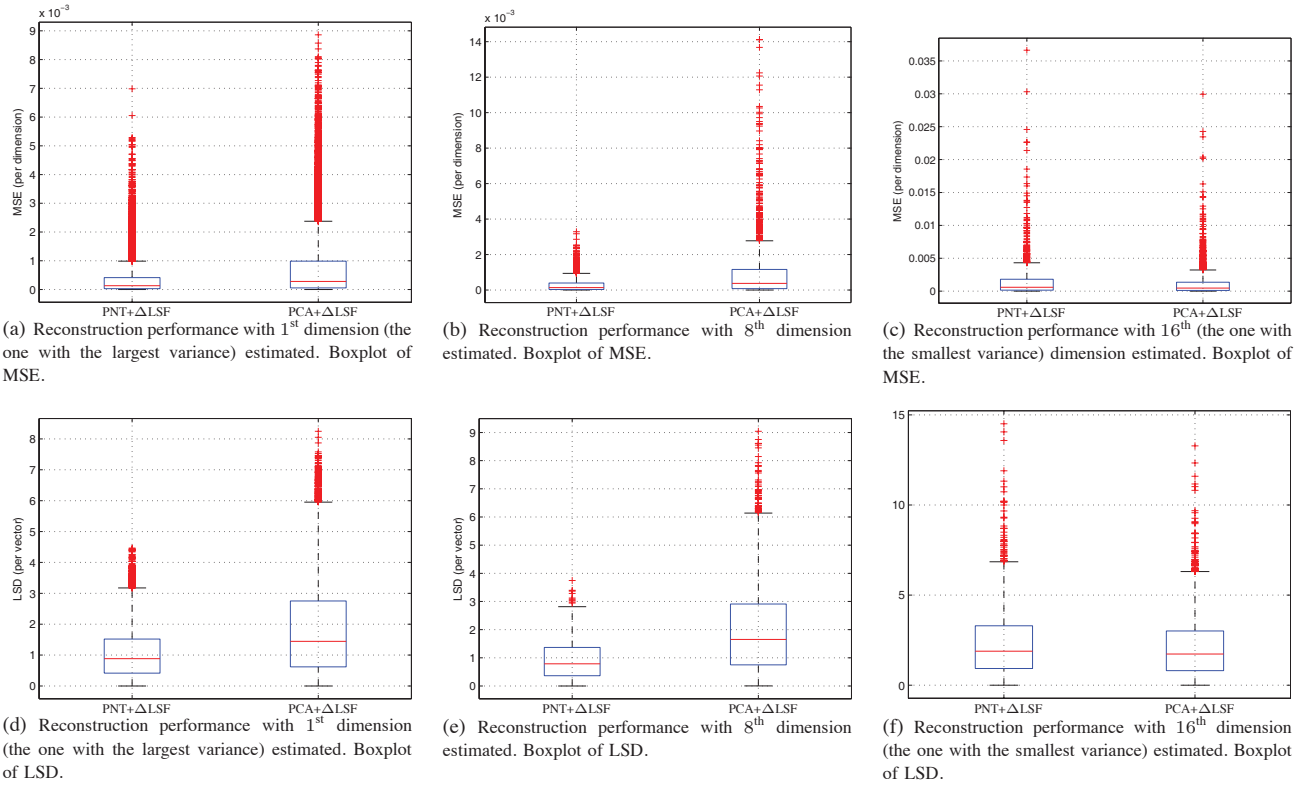
In order to demonstrate the statistical significance, we conducted the student's t-test for the null hypothesis that the two decorrelation methods are similar. This null hypothesis is rejected and the p-values are listed in Tab. VII as well. Figure 9 illustrates the comparisons of the original LPC spectrum, the reconstructed LPC spectrum via PNT, and the the reconstructed LPC spectrum via PCA.

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

V. CONCLUSIONS AND FUTURE WORK

A neutral vector variable is a typical non-Gaussian vector variable. By explicitly exploring the neutral properties, the so-called parallel nonlinear transformation (PNT) has already been proposed for the purpose of efficient and effective decorrelation of the neutral vector variable. In this paper, we studied and compared the PNT method with the conventionally applied principal component analysis (PCA) and independent component analysis (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.

[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, estimation error occurred in u_6 will have "global" effect while the error in u_1 or u_2 only has "local" effect.



(a) Reconstruction performance with 1st dimension (the one with the largest variance) estimated. Boxplot of MSE.

(b) Reconstruction performance with 8th 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 1st dimension (the one with the largest variance) estimated. Boxplot of LSD.

(e) Reconstruction performance with 8th dimension estimated. Boxplot of LSD.

(f) Reconstruction performance with 16th dimension (the one with the smallest variance) estimated. Boxplot of LSD.

Fig. 10. Comparisons of LPC reconstruction performances via boxplots. The missing dimensions are ♯1, ♯8, and ♯16, respectively.

There remains 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 as the improved version of PCA or ICA, an improved PNT is expected to be proposed such that the overall performance can also be improved; 4) investigate more real-applications with the proposed PNT and its variants.

REFERENCES

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[2] A. Kessy, A. Lewin, and K. Strimmer, "Optimal whitening and decorrelation," 2015, arXiv:1512.00809.

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

[4] J. Yan, M. Cho, H. Zha, X. Yang, and S. M. Chu, "Multi-graph matching via affinity optimization with graduated consistency regularization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 6, pp. 1228–1242, June 2016.

[5] R. Wang, *Introduction to Orthogonal Transforms*. Cambridge University Press, 2012.

[6] I. Jolliffe, *Principal Component Analysis*, ser. Springer Series in Statistics. Springer, 2002.

[7] C. Alzate and J. A. K. Suykens, "Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 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 Transactions on Neural Networks and Learning Systems*, vol. 23, no. 1, pp. 163–168, Jan 2012.

[9] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "Subspace learning from image gradient orientations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 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 Transactions on Image Processing*, 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 Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 2, pp. 505–517, Feb 2014.

[12] J. Yan, N. Liu, S. Yan, Q. Yang, W. Fan, W. Wei, and Z. Chen, "Trace-oriented feature analysis for large-scale text data dimension reduction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, pp. 1103–1117, July 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 Processing Magazine*, vol. 31, no. 3, pp. 34–43, May 2014.

[14] H. Guo, C. Qiu, and N. Vaswani, "An online algorithm for separating sparse and low-dimensional signal sequences from their sum," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4284–4297, Aug 2014.

[15] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2916–2929, Dec 2013.

[16] 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 Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 159–171, Jan 2016.

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

[18] Z. Ma and A. Leijon, "Bayesian estimation of beta mixture models with variational inference," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2160–2173, 2011.

[19] S. Jiang, X. Qian, J. Shen, Y. Fu, and T. Mei, "Author topic model-based collaborative filtering for personalized POI recommendations," *IEEE Transactions on Multimedia*, vol. 17, no. 6, pp. 907–918, June 2015.

[20] J. Liu, Y. Jiang, Z. Li, X. Zhang, and H. Lu, "Domain-sensitive recommendation with user-item subgroup analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 4, pp. 939–950, April 2016.

[21] P. W. Laird, "Principles and challenges of genome-wide DNA methylation analysis," *Nature Reviews Genetics*, pp. 191–203, 2010.

[22] E. A. Houseman, K. T. Kelsey, J. K. Wiencke, and C. J. Marsit, "Cell-composition effect in the analysis of dna methylation array data: a mathematical perspective," *BMC Bioinformatics*, vol. 16, no. 95, 2015.

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

[24] J. Taghia, Z. Ma, and A. Leijon, "Bayesian estimation of the von-Mises Fisher mixture model with variational inference," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 9, pp. 1701–1715, Sept 2014.

[25] C. Chen, W. Buntine, N. Ding, L. Xie, and L. Du, "Differential topic models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, pp. 230–242, Feb 2015.

[26] T. M. Nguyen and Q. M. J. Wu, "A nonsymmetric mixture model for unsupervised image segmentation," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 751–765, April 2013.

[27] Z. Ma, "Non-Gaussian statistical models and their applications," Ph.D. dissertation, KTH - Royal Institute of Technology, 2011.

[28] Z. Ma and A. E. Teschendorff, "A variational Bayes beta mixture model for feature selection in DNA methylation studies," *Journal of Bioinformatics and Computational Biology*, vol. 11, no. 4, 2013.

[29] Z. Ma, A. Leijon, and W. B. Kleijn, "Vector quantization of LSF parameters with a mixture of Dirichlet distributions," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, pp. 1777 – 1790, Sep. 2013.

[30] 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.

[31] J. V. Stone, *Independent component analysis: a tutorial introduction*. MIT Press, 2004.

[32] H. Nguyen and R. Zheng, "Binary independent component analysis with or mixtures," *IEEE Transactions on Signal Processing*, vol. 59, no. 7, pp. 3168–3181, July 2011.

[33] V. Laparra, G. Camps-Valls, and J. Malo, "Iterative Gaussianization: From ICA to random rotations," *IEEE Transactions on Neural Networks*, vol. 22, no. 4, pp. 537–549, April 2011.

[34] K.-C. Kwak and W. Pedrycz, "Face recognition using an enhanced independent component analysis approach," *IEEE Transactions on Neural Networks*, vol. 18, no. 2, pp. 530–541, March 2007.

[35] I. Santamaria, "Handbook of blind source separation: Independent component analysis and applications (common, p. and jutten, ; 2010 [book review]," *IEEE Signal Processing Magazine*, vol. 30, no. 2, pp. 133–134, March 2013.

[36] L. R. Arnaut and C. S. Obiekezie, "Source separation for wideband energy emissions using complex independent component analysis," *IEEE Transactions on Electromagnetic Compatibility*, vol. 56, no. 3, pp. 559–570, June 2014.

[37] R. J. Connor and J. E. Mosimann, "Concepts of independence for proportions with a generalization of the Dirichlet distribution," *Journal of the American Statistical Association*, vol. 64, no. 325, pp. 194–206, 1969.

[38] I. R. James and J. E. Mosimann, "A new characterization of the Dirichlet distribution through neutrality," *The Annals of Statistics*, vol. 8, no. 1, pp. 183–189, 1980.

[39] J.-T. Chien and C.-H. Chueh, "Topic-based hierarchical segmentation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, pp. 55–66, Jan. 2012.

[40] Y. Zhuang, H. Gao, F. Wu, S. Tang, Y. Zhang, and Z. Zhang, "Probabilistic word selection via topic modeling," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 6, pp. 1643–1655, June 2015.

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

[42] Z. Ma, R. Martin, J. Guo, and H. Zhang, "Nonlinear estimation of missing ΔLSF parameters by a mixture of Dirichlet distributions," in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2014, pp. 6929–6933.

[43] Z. Ma, S. Chatterjee, W. B. Kleijn, and J.Guo, "Dirichlet mixture modeling to estimate an empirical lower bound for LSF quantization," *Signal Processing*, vol. 104, no. 291-295, 2014.

[44] A. Sakowicz and J. Wesoowski, "Dirichlet distribution through neutralities with respect to two partitions," *Journal of Multivariate Analysis*, vol. 129, pp. 1–15, 2014.

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

[46] A. M. Dai and A. J. Storkey, "The supervised hierarchical Dirichlet process," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, pp. 243–255, 2015.

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

[48] 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 Journal on Selected Areas in Communications*, vol. 33, no. 5, pp. 771–787, May 2015.

[49] 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.

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

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

[52] 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 Transactions on Neural Networks and Learning Systems*, vol. 30, no. 10, pp. 3124–3136, 2019.

[53] H. Mei and J. Eisner, "The neural hawkes process: a neurally self-modulating multivariate point process," in *neural information processing systems*, 2017, pp. 6757–6767.

[54] A. Miller, L. Bornn, R. P. Adams, and K. Goldsberry, "Factorized point process intensities: A spatial analysis of professional basketball," in *international conference on machine learning*, 2014, pp. 235–243.

[55] T. T. Pham, S. H. Rezatofighi, I. Reid, and T. Chin, "Efficient point process inference for large-scale object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2837–2845.

[56] N. Mehrasa, A. A. Jyothi, T. Durand, J. He, L. Sigal, and G. Mori, "A variational auto-encoder model for stochastic point processes." in *computer vision and pattern recognition*, 2019, pp. 3165–3174.

[57] M. M. Moradi and J. Mateu, "First- and second-order characteristics of spatio-temporal point processes on linear networks," *Journal of Computational and Graphical Statistics*, 2019.

[58] H.-Y. Wang, Q. Yang, H. Qin, and H. Zha, "Dirichlet component analysis: Feature extraction for compositional data," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2008.

[59] Z. Ma, J.-H. Xue, A. Leijon, Z.-H. Tan, Z. Yang, and J. Guo, "Decorrelation of neutral vector variables: Theory and applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 1, pp. 129–143, Jan. 2018.

[60] P. Howard, D. W. Apley, and G. Runger, "Distinct variation pattern discovery using alternating nonlinear principal component analysis," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 1, pp. 156–166, Jan 2018.

[61] N. Shahid, N. Perraudin, V. Kalofolias, G. Puy, and P. Vandergheynst, "Fast robust PCA on graphs," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 4, pp. 740–756, June 2016.

[62] S. Z. Rizvi, J. Mohammadpour, R. Tth, and N. Meskin, "A kernel-based PCA approach to model reduction of linear parameter-varying systems," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 5, pp. 1883–1891, Sep. 2016.

[63] Y. Xiao, Z. Zhu, Y. Zhao, Y. Wei, and S. Wei, "Kernel reconstruction ICA for sparse representation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 6, pp. 1222–1232, June 2015.

[64] R. K. S. Hankin, "A generalization of the Dirichlet distribution," *Journal of Statistical Software*, vol. 33, no. 11, pp. 1–18, 2010.

[65] B. A. Frigyik, A. Kapila, and M. R. Gupta, "Introduction to the Dirichlet distribution and related processes," Department of Electrical Engineering, University of Washington, Tech. Rep., 2010.

[66] V. B. Balakrish, *A Primer on Statistical Distributions*. John Wiley & Sons, 2005, ch. Dirichlet Distribution, p. 274.

[67] T. Minka, "The lightspeed matlab toolboxes." [Online]. Available: http://research.microsoft.com/en-us/um/people/minka/software/lightspeed/

[68] F. T. Luk and S. Qiao, *A fast singular value algorithm for Hankel matrices*. Boston, MA, USA: American Mathematical Society, 2003, pp. 169–177.

[69] S. Shwartz, M. Zibulevsky, and Y. Y. Schechner, *ICA Using Kernel Entropy Estimation with NlogN Complexity*. Springer Berlin Heidelberg, 2004, ch. Independent Component Analysis and Blind Signal Separation, pp. 422–429.

[70] G. J. Székely, M. L. Rizzo, and N. K. Bakirov, "Measuring and testing independence by correlation of distances," *Annals of Statistics*, vol. 35, no. 6, pp. 2769–2794, 2007.
[71] G. J. Székely and M. L. Rizzo, "Brownian distance covariance," *Annals of Applied Statistics*, vol. 3, no. 4, pp. 1233–1303, 2009.
[72] K. Pearson, "Notes on regression and inheritance in the case of two parents," in *Proceedings of the Royal Society of London*, 1895, pp. 240–242.
[73] R. R. Wilcox, *Introduction to robust estimation and hypothesis testing*. Academic Press, 2005.
[74] G. J. Székely and M. L. Rizzo, "On the uniqueness of distance covariance," *Statistics & Probability Letters*, vol. 82, no. 12, pp. 2278–2282, 2012.
[75] D. M. Blei and J. D. Lafferty, "Correlated topic models," in *Advances in Neural Information Processing Systems (NIPS)*, 2006.
[76] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi, "A review of classification algorithms for EEG-based brain-computer interfaces," *Journal of Neural Engineering*, vol. 4, no. 2, p. R1, 2007.
[77] H. Cecotti and A. Graser, "Convolutional neural networks for P300 detection with application to brain-computer interfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 433–445, March 2011.
[78] A. Subasi, "EEG signal classification using wavelet feature extraction and a mixture of expert model," *Expert Systems with Applications*, vol. 32, no. 4, pp. 1084 – 1093, 2007.
[79] Z. Ma, Z.-H. Tan, and S. Prasad, "EEG signal classification with super-dirichlet mixture model," in *Proceedings of IEEE Statistical Signal Processing Workshop*, Aug. 2012, pp. 440 – 443.
[80] Z. Xu, S. MacEachern, and X. Xu, "Modeling non-Gaussian time series with nonparametric Bayesian model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, pp. 372–382, Feb 2015.
[81] "BCI competition III," http://www.bbci.de/competition/iii.
[82] T. N. Lal, M. Schroder, T. Hinterberger, J. Weston, M. Bogdan, N. Birbaumer, and B. Scholkopf, "Support vector channel selection in BCI," *Biomedical Engineering, IEEE Transactions on*, vol. 51, no. 6, pp. 1003 –1010, Jun. 2004.
[83] W. Malina, "On an extended fisher criterion for feature selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-3, no. 5, pp. 611 –614, Sep. 1981.
[84] H.-I. Suk and S.-W. Lee, "A novel Bayesian framework for discriminative feature extraction in brain-computer interfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 286–299, Feb 2013.
[85] C.-C. Chang and C.-J. Lin., "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 17, pp. 1–27, 2011.
[86] X. Huang, L. Shi, and J. A. K. Suykens, "Support vector machine classifier with pinball loss," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 984–997, May 2014.
[87] P. Vary and R. Martin, *Digital Speech Transmission: Enhancement, Coding and Error Concealment*. Chichester, England: John Wiley & Sons, Ltd, 2006.
[88] J. Li, N. Chaddha, and R. M. Gray, "Asymptotic performance of vector quantizers with a perceptual distortion measure," *IEEE Trans. on Information Theory*, vol. 45, pp. 1082 – 1091, May 1999.
[89] W. B. Kleijn and A. Ozerov, "Rate distribution between model and signal," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2007, pp. 243–246.
[90] F. Soong and B. Juang, "Optimal quantization of LSP parameters using delayed decisions," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1990, pp. 185 –188.
[91] Z. Ma, S. Chatterjee, W. B. Kleijn, and J. Guo, "Dirichlet mixture modeling to estimate an empirical lower bound for LSF quantization," *Signal Processing*, vol. 104, no. 11, pp. 291–295, Nov. 2014.
[92] "DARPA-TIMIT," *Acoustic-phonetic continuous speech corpus, NIST Speech Disc 1.1-1*, 1990.
[93] S. Chatterjee and T. Sreenivas, "Predicting VQ performance bound for LSF coding," *IEEE Signal Processing Letters*, vol. 15, pp. 166 –169, 2008.
[94] L. A. Ekman, W. B. Kleijn, and M. N. Murthi, "Regularized linear prediction of speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 65 –73, Jan. 2008.

Zhanyu Ma is currently a full Professor at Beijing University of Posts and Telecommunications, Beijing, China. He received his Ph.D. degree in Electrical Engineering from KTH-Royal Institute of Technology, Sweden, in 2011. From 2012 to 2013, he has been a Postdoctoral research fellow in the School of Electrical Engineering, KTH, Sweden. He has been an Associate Professor at Beijing University of Posts and Telecommunications, Beijing, China from 2014 to 2019. He is also an adjunct Associate Professor at Aalborg University, Aalborg, Denmark, since 2015. His research interests include pattern recognition and machine learning fundamentals with a focus on applications in computer vision, multimedia signal processing, data mining. He is a Senior Member of IEEE.

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, ~~U.K.~~, in 2011. He is currently pursuing the Ph.D. degree at the Department of Statistical Science, University College London. His research interests include compositional data analysis and machine learning.

Jiyang Xie received his B.E. degree in information engineering from Beijing University of Posts and Telecommunications (BUPT), 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 is currently a full professor of College of Computer Science, Faculty of Information Technology, Beijing University of Technology. He received the PhD degree in signal processing from the Beijing University of Posts and Telecommunications. His research interests include data mining, machine learning, trusted computing, and content security. He has published more than 30 papers in highly ranked journals and top conference proceedings. He is a senior Member of the Chinese Institute of Electronics and a member of the IEEE.

Jing-Hao Xue received the Dr. Eng. degree in signal and information processing from Tsinghua University in 1998 and the Ph.D. degree in statistics from the University of Glasgow in 2008. Since 2008 he has worked in the Department of Statistical Science at University College London as a Lecturer and Senior Lecturer. His current research interests include statistical classification, high-dimensional data analysis, computer vision, and pattern recognition.

Zheng-Hua Tan is a Professor in the Department of Electronic Systems at 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. He has served as an Editorial Board Member/Associate Editor for Elsevier Computer Speech and Language, Elsevier Digital Signal Processing and Elsevier Computers and Electrical Engineering. He was a Lead Guest Editor for the IEEE Journal of Selected Topics in Signal Processing. He is a senior member of IEEE.

Bo Xiao was born in Changyi City, Shandong Province, China in 1975. He received his BS degree in image transmission and processing, MS degree in Computer Science and Ph.D. degree in Signal and Information Processing from Beijing University of Posts and Telecommunications, Beijing China, in 1998, 2005 and 2009, respectively. He works in the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China, since 1998, and 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 co-authored five books and more than 40 journal and conference papers. His research interests include data mining, pattern recognition, deep learning, computer vision, intelligent wireless communication etc.

Jun Guo received B.E. and M.E. degrees from Beijing University of Posts and Telecommunications (BUPT), China in 1982 and 1985, respectively, Ph.D. degree from the Tohuku-Gakuin University, Japan in 1993. At present he is a professor and a vice president of BUPT. His research interests include pattern recognition theory and application, information retrieval, content based information security, and bioinformatics. He has published over 200 papers on the journals and conferences including SCIENCE, Nature Scientific Reports, IEEE Trans. on PAMI, Pattern Recognition, AAAI, CVPR, ICCV, SIGIR, etc.