Detecting the Number of Clusters in n-Way Probabilistic Clustering

Zhaoshui He, Andrzej Cichocki, *Senior Member, IEEE*,
Shengli Xie, *Senior Member, IEEE*, and Kyuwan Choi

Abstract—Recently, there has been a growing interest in multiway probabilistic clustering. Some efficient algorithms have been developed for this problem. However, not much attention has been paid on how to detect the number of clusters for the general n-way clustering ($n \geq 2$). To fill this gap, this problem is investigated based on n-way algebraic theory in this paper. A simple, yet efficient, detection method is proposed by eigenvalue decomposition (EVD), which is easy to implement. We justify this method. In addition, its effectiveness is demonstrated by the experiments on both simulated and real-world data sets.

Index Terms—Multiway clustering, probabilistic clustering, hypergraph, parallel factor analysis (PARAFAC), model order selection, multiway array, higher order tensor, supersymmetric tensors, affinity arrays, enumeration of clusters, estimation of PARAFAC components, principal components enumeration.

1 Introduction

It is known that the relationships among real-world objects are usually more complex than pairwise in many applications. Simply approximating the complex relationships as pairwise can probably lead to the loss of information. For this reason, *hypergraghs* have been employed to describe the complex relationships among the data [1], where the hypergraph edges can connect more than two vertices [2], for example, a hypergraph with T hypervertices and C_T^n hyperedges, where n is the hyperedge degree. Hypervertices correspond to the objects in real-world problems and hyperedges correspond to the vertex subsets which represent the complex relationships among n objects of interest [1], [3], [4]. By hypergraghs, the complex relationships can be represented by a multiway array (or tensor) and we can analyze them by tensor computation using mathematical tools.

The pairwise (or two-way) clustering has been extensively studied up to now [5], [6], [7], [8], [9]. Based on hypergraph theory, pairwise clustering has been generalized to multiway in the past several years. Compared with conventional pairwise clustering, multiway clustering can exploit the multiway relations beyond pairwise [1], [3], [4]. Furthermore, it enables us to maximize the intracluster similarities and minimize the intercluster similarities among the objects more efficiently. Nowadays, multiway clustering is receiving more and more attention in data processing and machine learning. So far, several efficient multiway clustering methods have been developed. The hyperspectral clustering method was studied by Zhou et al. [1]; multiway clustering on relation graphs was discussed by Banerjee et al. [4]; Shashua et al. originally developed a multiway probabilistic clustering method by supersymmetric nonnegative tensor factorization (SS-NTF) [3].

However, in the existing works, there are not many discussions about the issue of choosing the number of clusters for multiway clustering. In most of them, the cluster number is simply given as a priori.

Following Shashua and Zass's pioneering works [3], [10], blind detection of cluster number for n-way probabilistic clustering is studied in this paper. And an efficient detection algorithm is developed by searching the gap in the ordered eigenvalues of the covariance matrix of an observed sequence, which works well for both pairwise clustering and multiway when the noise eigenvalues are approximately equal. The rest of this paper is organized as follows: The problem formulation of n-way probabilistic clustering is given in Section 2. In Section 3, we discuss the principle for selecting the scaling parameter such that the n-way probabilistic clustering can be approximately formulated to a parallel factor analysis (PARAFAC) problem. The algorithm for estimating the number of clusters is presented in detail in Section 4. Some existing methods for determining the number of clusters in pairwise clustering are briefly reviewed in Section 5. The demonstrations and experiments are given in Section 6. Finally, we conclude this paper in Section 7.

- Z. He is with the Laboratory for Advanced Brain Signal Processing, RIKEN Brain Science Institute, Wako-shi, Saitama 3510198, Japan, and the School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510640, China. E-mail: he_shui@tom.com.
- A. Cichocki is with the Laboratory for Advanced Brain Signal Processing, RIKEN Brain Science Institute, Wako-shi, Saitama 3510198, Japan, the System Research Institute, Polish Academy of Sciences (PAN), Warsaw 00-901, Poland, and the Department of Electrical Engineering, Warsaw University of Technology, Warsaw 00-661, Poland. E-mail: cia@brain.riken.jp.
- S. Xie is with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510640, China. E-mail: adshlxie@scut.edu.cn.
- K. Choi is with the Department of Computational Brain Imaging, ATR Computational Neuroscience Laboratories, 2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288, Japan. E-mail: kyuwanchoi@gmail.com.

Manuscript received 12 Nov. 2008; revised 16 Mar. 2009; accepted 16 Aug. 2009; published online 7 Jan. 2010.
Recommended for acceptance by M. Meila.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2008-11-0787.
Digital Object Identifier no. 10.1109/TPAMI.2010.15.

2 Problem Formulation of n-Way Probability Clustering

2.1 Mathematical Notations

The notations used in this paper are mostly consistent with previous publications in the area of tensor decompositions [11], [12], [13]. Scalars are denoted by italic letters (e.g., a, b, $\ldots, A$, B, $\ldots, \alpha$, β, $\ldots$), column vectors by italic lowercase boldface letters ($\boldsymbol{a}$, $\boldsymbol{b}$, $\ldots$), matrices by italic boldface capital letters ($\boldsymbol{A}$, $\boldsymbol{B}$, $\cdots$), and tensors are written as calligraphic capital letters ($\mathcal{A}$, $\mathcal{B}$, $\cdots$). The (i,j)-element of a matrix $\boldsymbol{A}$ is denoted by a_{ij} or $a_{i,j}$. Similarly, the $(t_1,\ldots,t_n)$-element of an nth-order tensor $\mathcal{A}$ is denoted by $a_{t_1,\ldots,t_n}$ or $(\mathcal{A})_{t_1,\ldots,t_n}$. The *Kronecker product* [11] of matrices $\boldsymbol{A} \in \mathbb{R}^{I\times J}$ and $\boldsymbol{B} \in \mathbb{R}^{K\times L}$ is denoted by $\boldsymbol{A}\otimes\boldsymbol{B}$ and given by

$$\boldsymbol{A}\otimes\boldsymbol{B} \triangleq \begin{bmatrix} a_{11}\boldsymbol{B} & \cdots & a_{1J}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{I1}\boldsymbol{B} & \cdots & a_{IJ}\boldsymbol{B} \end{bmatrix}.$$

The *Khatri-Rao product* [11] of $\boldsymbol{A} \in \mathbb{R}^{I\times K}$ and $\boldsymbol{B} \in \mathbb{R}^{J\times K}$ is defined as

$$\boldsymbol{A}\odot\boldsymbol{B} \triangleq [\boldsymbol{a}_1\otimes\boldsymbol{b}_1,\ \ldots, \boldsymbol{a}_K\otimes\boldsymbol{b}_K], \tag{1}$$

where $\boldsymbol{A} = [\boldsymbol{a}_1,\ldots,\boldsymbol{a}_K]$, $\boldsymbol{B} = [\boldsymbol{b}_1,\ldots,\boldsymbol{b}_K]$.

2.2 n-Way Probability Clustering

Let $\boldsymbol{X} = [\boldsymbol{x}_1,\ldots,\boldsymbol{x}_T]$ be an observed sample matrix, which contains T data points $\boldsymbol{x}_t \in \mathbb{R}^m$, $t = 1,\ldots,T$. K clusters are, respectively, denoted by $\Omega_1,\ldots,\Omega_K$. The objective of probabilistic clustering is to estimate the assignment probability matrix $\boldsymbol{P} = (p_{t,k})_{T\times K}$ and then assign $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_T$ into K clusters $\Omega_1,\ldots,\Omega_K$ according to their probability, where $p_{t,k} = \Pr(\boldsymbol{x}_t \in \Omega_k)$ is the probability that the sample $\boldsymbol{x}_t$ is partitioned into the cluster Ω_k. To accomplish this task, we must identify the cluster number K at first.

Let's make the following assumptions:

Assumption 1. *Given $\boldsymbol{X}$, K clusters $\Omega_1,\ldots,\Omega_K$ are mutually disjoint, i.e., $\Omega_1 \perp \ldots \perp \Omega_K \mid \boldsymbol{X}$;*

Assumption 2. *T observed points $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_T$ are mutually independent statistically.*

Define the *indicator function*,

$$I_{t_1,\ldots,t_n} = \begin{cases} 1, & \boldsymbol{x}_{t_1},\ldots,\boldsymbol{x}_{t_n} \text{ are in the same cluster,} \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

where $t_1,\ldots,t_n \in \{1,\ldots,T\}$.

Denote $g_{t_1,\ldots,t_n}$ as the probability that the samples $\boldsymbol{x}_{t_1},\ldots,\boldsymbol{x}_{t_n}$ are partitioned into the same cluster. If any two of $t_1,\ldots,t_n$ are not pairwise equal to each other, from (2), we have

$$\begin{aligned} g_{t_1,\ldots,t_n} &= \Pr(I_{t_1,\ldots,t_n} = 1) = \Pr\left(\bigcup_{k=1}^{K}\bigcap_{i=1}^{n}(\boldsymbol{x}_{t_i} \in \Omega_k)\right) \\ &= \sum_{k=1}^{K}\Pr\left(\bigcap_{i=1}^{n}(\boldsymbol{x}_{t_i} \in \Omega_k)\right). \end{aligned} \tag{3}$$

From Assumption 2, we have

$$\Pr\left(\bigcap_{i=1}^{n}(\boldsymbol{x}_{t_i} \in \Omega_k)\right) = \prod_{i=1}^{n}\Pr(\boldsymbol{x}_{t_i} \in \Omega_k) = \prod_{i=1}^{n} p_{t_i,k}, \tag{4}$$

where $k = 1,\ldots,K$. Combining (3) and (4), we can get

$$g_{t_1,\ldots,t_n} = \sum_{k=1}^{K} p_{t_1,k}\cdots p_{t_n,k} = \sum_{k=1}^{K}\prod_{i=1}^{n} p_{t_i,k}. \tag{5}$$

Note that (5) only covers those indices $t_1,\ldots,t_n$ such that $t_1,\ldots,t_n$ are pairwisely unequal. For the sake of our algorithmic implementation later, we need to extend (5) for the more general cases where certain indices $t_1,\ldots,t_n$ can be *identical*. For example, $g_{1,1,2}$, where $t_1 = t_2 = 1, t_3 = 2$. For this purpose, when certain indices $t_1,\ldots,t_n$ are identical, we give a special supplemental definition for them as

$$g_{t_1,\ldots,t_n} \triangleq \sum_{k=1}^{K} p_{t_1,k}\cdots p_{t_n,k} = \sum_{k=1}^{K}\prod_{i=1}^{n} p_{t_i,k}. \tag{6}$$

By the supplemental definition (6), we have

$$g_{t_1=1,t_2=1,t_3=2} = \sum_{k=1}^{K} p_{1,k}^2 \cdot p_{2,k}.$$

Although the physical interpretation of (6) is not consistent with (3), where $g_{t_1,\ldots,t_n} = \Pr(I_{t_1,\ldots,t_n} = 1)$ is interpreted as a membership probability, the algebraic structure of (6) is consistent with (5). By incorporating (6), (5) holds for all indices $t_1,\ldots,t_n \in \{1,\ldots,T\}$, which results in the n-way supersymmetric PARAFAC model [14], [15], [16] of $T\times\cdots\times T$ tensor $\mathcal{G}$ with respect to $\boldsymbol{P}$ as follows:

$$\mathcal{G} = \mathcal{I}\times_1 \boldsymbol{P}\times_2\cdots\times_n \boldsymbol{P}, \tag{7}$$

where $\mathcal{I}$ stands for a *superidentity tensor*, i.e., $(\mathcal{I})_{t_1,\ldots,t_n} = 1$ iff $t_1 = \cdots = t_n$; otherwise, $(\mathcal{I})_{t_1,\ldots,t_n} = 0$ and the operator "$\times_i$" denotes the ith-*mode multiplication* of a Tucker model ($i = 1,\ldots,n$) [11].

Then, the probabilistic clustering can be cast into an n-way supersymmetric PARAFAC fitting problem [14], [17], [18], [19]:

$$\min_{\boldsymbol{P}}\|\mathcal{G} - \mathcal{I}\times_1\boldsymbol{P}\times_2\cdots\times_n\boldsymbol{P}\|_2^2, \tag{8}$$

where $\|\cdot\|_2$ denotes the *Frobenius norm* of a tensor [12].

3 Computing the Probability Affinity for n-Way Probability Clustering

The probability affinity tensor $\mathcal{G}$ is unknown in (8). Let $\mathcal{V} \in \mathbb{R}^{T\times\cdots\times T}$ be the n-way distance/dissimilarity tensor of an n-way clustering, computed from the input data points $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_T$.

3.1 Estimating the Probability Affinity $\widehat{\mathcal{G}}$ from the n-Way Distance Affinity $\mathcal{V}$

Remark 1. The construction of n-way distance tensor $\mathcal{V}$ is flexible and not unique. Usually, it depends on the practical applications [3]. In principle, we attempt to construct $\mathcal{V} = (v_{t_1,\ldots,t_n})_{T\times T}$ approximately satisfying the conditions

TABLE 1
The Monotonically Decreasing Function $f(z) = e^{-z}$

z	0	1	2	**3**	4	5
$f(z)$	1	0.3679	0.1353	**0.0498**	0.0183	0.0067

$$\begin{cases} v_{t_1,\ldots,t_n} = 0, \ \boldsymbol{x}_{t_1}, \ldots, \boldsymbol{x}_{t_n} \text{ belong to the same cluster,} \\ v_{t_1,\ldots,t_n} > 0, \ \text{otherwise,} \end{cases} \tag{9}$$

as much as it can. For point clustering in euclidean space, the n-way distance $v_{t_1,\ldots,t_n}$ with respect to the n-tuple of points $\boldsymbol{x}_{t_1}, \ldots, \boldsymbol{x}_{t_n}$ usually can be given as:

$$v_{t_1,\ldots,t_n} = \sum_{1 \le i < j \le n} \left\| \boldsymbol{x}_{ti} - \boldsymbol{x}_{tj} \right\|_2, \tag{10}$$

where $\mathcal{V} = (v_{t_1,\ldots,t_n})_{T\times\cdots\times T}$.

Given $\mathcal{V}$ and a positive number Δ, according to [3], we can estimate the probability affinity tensor $\mathcal{G}$ for (8) using the Gaussian kernel function

$$\hat{g}_{t_1,\ldots,t_n} = e^{-\frac{v^2_{t_1,\ldots,t_n}}{\Delta^2}} \longrightarrow g_{t_1,\ldots,t_n}, \tag{11}$$

i.e., $\widehat{\mathcal{G}} = (\hat{g}_{t_1,\ldots,t_n})_{T\times\cdots\times T} \longrightarrow \mathcal{G}$.

As in [3], [10], [20], after $\widehat{\mathcal{G}}$ is computed by (11), we can obtain a normalized version of $\widehat{\mathcal{G}}$ by n-stochastic normalization.

3.2 Principles for Selecting the Parameter Δ

Let us begin with the definition of "*the ideal probability clustering.*"

Definition 1. *A probability clustering is said to be "ideal" if its probability assignment matrix $\boldsymbol{P} = (p_{t,k})_{T\times K}$ satisfies the following binary conditions:*

$$\begin{cases} p_{t,k} = 1, \ \textit{the data point } \boldsymbol{x}_t \textit{ is from the cluster } \Omega_k, \\ p_{t,k} = 0, \ \textit{otherwise.} \end{cases} \tag{12}$$

Denote $\mathcal{E}$ to be the estimation error of $\mathcal{G}$ by $\widehat{\mathcal{G}}$, i.e.,

$$\widehat{\mathcal{G}} = \mathcal{G} + \mathcal{E} = \mathcal{I} \times_1 \boldsymbol{P} \times_2 \cdots \times_n \boldsymbol{P} + \mathcal{E}, \tag{13}$$

where $\mathcal{E} = (\varepsilon_{t_1,\ldots,t_n})_{T\times\cdots\times T} \in \mathbb{R}^{T\times\cdots\times T}$ and

$$\begin{aligned} \varepsilon_{t_1,\ldots,t_n} &= \varepsilon(v_{t_1,\ldots,t_n}, \Delta) \\ &= \hat{g}_{t_1,\ldots,t_n} - \sum_{k=1}^{K} p_{t_1,k} \times \cdots \times p_{t_n,k}. \end{aligned} \tag{14}$$

Proposition 1. *For an ideal n-way probability clustering, $\hat{g}_{t_1,\ldots,t_n}$ is given by (11). If the distance tensor $\mathcal{V}$ is properly selected to exactly satisfy the conditions (9), then we have*

$$|\varepsilon_{t_1,\ldots,t_n}| \le e^{-\frac{v^2_{t_1,\ldots,t_n}}{\Delta^2}}. \tag{15}$$

The proof is in Appendix A. Proposition 1 can yield the following corollary:

Corollary 1. *For an ideal n-way probability clustering, estimating probability affinity tensor $\widehat{\mathcal{G}}$ by (11) and supposing that the distance tensor $\mathcal{V}$ satisfies the conditions (9), then we have*

$$\lim_{\Delta \to 0_+} \widehat{\mathcal{G}} = \mathcal{I} \times_1 \boldsymbol{P} \times_2 \cdots \times_n \boldsymbol{P}. \tag{16}$$

Proof. The result readily follows from Proposition 1. □

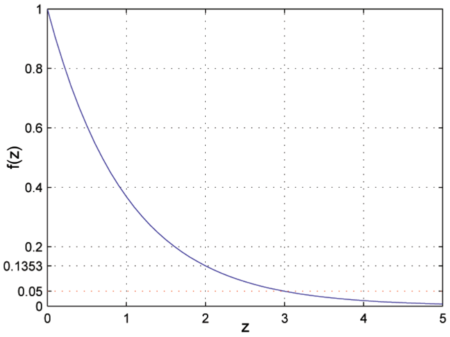


Fig. 1. The function $f(z) = e^{-z}$ is monotonically decreasing in the interval $[0, 5]$.

From (16), we can perform n-way clustering by solving the following n-way approximation problem:

$$\min_{\boldsymbol{P}} \|\widehat{\mathcal{G}} - \mathcal{I} \times_1 \boldsymbol{P} \times_2 \cdots \times_n \boldsymbol{P}\|_2^2. \tag{17}$$

Before the discussion of the selection of Δ^2, it is necessary to mention the proposition:

Proposition 2. *If an n-way distance tensor $\mathcal{V}$ satisfies the condition (9), the percentage $\gamma(\mathcal{V})$ of zero entries of $\mathcal{V}$ is*

$$\gamma(\mathcal{V}) = \frac{\sum_{k=1}^{K} \#\{\Omega_k\}^n}{T^n} \times 100\%, \tag{18}$$

where $\#\{\cdot\}$ accounts for the number of the elements of a set $\{\cdot\}$.

Proof. Please refer to Appendix B. □

To better analyze the error bound of (15), consider the function $f(z) = e^{-z}$. From Table 1 and Fig. 1, it can be seen that it is monotonically decreasing as z increases in the interval $[0, +\infty)$. When $z \ge 3$, $f(z) < 0.05$. Thus, the approximation error $\varepsilon_{t_1,\ldots,t_n}$ in (15) will be less than 0.05 as long as $\frac{v^2_{t_1,\ldots,t_n}}{\Delta^2} \ge 3$. In practice, this condition is not difficult to satisfy.

Therefore, we suggest carefully selecting the parameter Δ^2 such that $\frac{v^2_{t_1,\ldots,t_n}}{\Delta^2} \ge 3$ for most of entries $v_{t_1,\ldots,t_n}$ of $\mathcal{V}$. To meet this requirement, we can set Δ^2 by calculating the percentiles of the sequence $\{v_{t_1,\ldots,t_n} : t_1, \ldots, t_n = 1, \ldots, T\}$.

Let $R(\alpha)$ be the αth percentile of the entries in $\{v_{t_1,\ldots,t_n} : t_1, \ldots, t_n = 1, \ldots, T\}$, i.e.,

$$\frac{\#\{v_{t_1,\ldots,t_n} : v_{t_1,\ldots,t_n} \ge R(\alpha)\}}{\#\{v_{t_1,\ldots,t_n} : t_1, \ldots, t_n = 1, \ldots, T\}} \ge \alpha\%. \tag{19}$$

From the above discussion, it is guaranteed that (100-α) percent entries $v_{t_1,\ldots,t_n}$ in $\mathcal{V}$ will satisfy $\frac{v^2_{t_1,\ldots,t_n}}{\Delta^2(\alpha)} \ge 3$ if we set $\Delta^2(\alpha) \le \frac{R(\alpha)}{3}$. For simplicity, it is set as $\Delta^2(\alpha) = \frac{R(\alpha)}{3}$ in this paper.

On the other side, Δ^2 should not be too small. The extremely small Δ^2 will probably force $\widehat{\mathcal{G}} \longrightarrow 0_+$ in (11). From Proposition 2, we suggest setting the percentile parameter α to satisfy $\alpha > \gamma(\mathcal{V})$.

Therefore, our principle for Δ^2 can be outlined as

$$\begin{cases} \alpha > \gamma(\mathcal{V}), \\ \Delta^2(\alpha) = \dfrac{R(\alpha)}{3}. \end{cases} \qquad (20a)(20b)$$

For three-way clustering or higher-order array clustering (i.e., $n \geq 3$), usually the value of $\gamma(\mathcal{V})$ is quite small (e.g., $\gamma(\mathcal{V}) < 5\%$). In this case, it is suggested to set $\alpha = 100 \times 0.1^{n-2}$ ($n \geq 3$) for the case of no prior knowledge.

For two-way clustering, $\gamma(\mathcal{V})$ can sometimes be over 10 percent when there are few clusters (i.e, $K \leq 4$). As an example, please refer to the two-way clustering case of Example 5 in the experimental section, where there are a total of $T = 32$ samples and $K = 8$. In this case, it is not convenient to set Δ^2 by giving α in (20a) and (20b). From our experience, for two-way clustering, Δ^2 can be empirically chosen as

$$\Delta^2 = \frac{S^2}{\beta}, \qquad (21)$$

where the parameter β can take value in a relatively wide interval $\beta \in [3, 20]$ (typically, $\beta = 10$) and

$$S^2 = \frac{1}{|\mathcal{V}|} \sum_{t_1,\ldots,t_n} v^2_{t_1,\ldots,t_n},$$

where $|\mathcal{V}|$ denotes the cardinality of $\mathcal{V}$.

4 Detecting the Cluster Number by Estimating the Number of Components in PARAFAC Models

For a perfectly fitted PARAFAC model (16), we have $K = \mathrm{rank}(\lim_{\Delta\to 0_+} \widehat{\mathcal{G}})$. So, mathematical determination of the cluster number requires identifying the rank of tensor "$\lim_{\Delta\to 0_+} \widehat{\mathcal{G}}$" or identifying the number of components in the PARAFAC model (16).

4.1 Finding the Cluster Number by Determining the Number of Components in PARAFAC Models

An important issue in multilinear algebra is the determinacy of the number of components for the PARAFAC model. This problem has been studied recently in [21], [22], [23], [24], [25], [26], [27], [28]. It should be noted that determining the number of certain specifics given PARAFAC models can probably be NP-hard [13], [22], [29], [30]. Even for a three-way array, its rank can largely exceed the size of the array in all dimensions.

Fortunately, here our task is not NP-hard because, theoretically, $\mathrm{rank}(\lim_{\Delta\to 0_+} \widehat{\mathcal{G}}) = K \leq T$. For this special case, recent attention to this problem has mainly focused on two types of methods: the PARAFAC/Tucker-decomposition-based methods [21], [23], [24], [25], [26], [28] and the R-dimensional (R-D) model selection methods [27], [28].

The PARAFAC/Tucker-decomposition-based methods start by computing a set of candidate models (by PARAFAC/Tucker decomposition) for a range of values k from K_{min} to K_{max}, which is assumed to contain the true/optimal K. Then, the number of components is determined according to an appropriate model selection criterion. These methods include DIFFIT [23], Fast DIFFIT [25], CORCONDIA [24], Threshold-CORCONDIA [28], convex-hull-based method [21], [26], etc.

In addition, several R-D model selection methods were proposed in [27]: R-dimensional exponential fitting test (R-D EFT), R-D AIC, and R-D MDL. Basically, they are based on the concept of "global eigenvalues," which are the combinations of eigenvalues of different unfoldings. The R-D methods were tested on PARAFAC data sets and compared with other methods in [28] in which it was shown that R-D EFT performed best. But, R-D EFT has a limitation on the type of noise, which should be only white Gaussian noise.

All of these methods mentioned above can be potentially used to estimate the rank of $\lim_{\Delta\to 0_+} \widehat{\mathcal{G}}$ in (16). However, the PARAFAC/Tucker-decomposition-based methods need to repeatedly perform n-way PARAFAC/Tucker decomposition for computing its n factors. Since the PARAFAC/Tucker decomposition is time-consuming, the PARAFAC/Tucker-decomposition-based methods are extremely expensive computationally. As for the R-D methods, their computational cost is cheaper than that of the former. But, they also cannot avoid the computation of PARAFAC/Tucker decomposition. Next, we introduce an eigenvalue-decomposition (EVD)-based scheme to estimate the number of clusters, which is much more simple.

4.2 Tensor Matricization

This section discusses the tensor unfolding (or called matrix representation) $\boldsymbol{G} = \mathrm{mat}(\mathcal{G})$ of an n-way tensor $\mathcal{G}$. The tensor unfolding is not unique, i.e., Kofidis and Regalia performed the square matrix unfolding for even number order tensor (i.e., $n = 2r$) in [31]. For simplicity, we unfold the tensor $\mathcal{G}$ to be a matrix $\boldsymbol{G}$ such that $\boldsymbol{G}$ has a linear factor with respect to the matrix $\boldsymbol{P}$. In detail, $\boldsymbol{G} = \mathrm{mat}(\mathcal{G})$ is given in this work as

$$(\boldsymbol{G})_{t_1, t_2 + (t_3-1)T + \cdots + (t_n-1)T^{n-2}} = (\mathcal{G})_{t_1,\ldots,t_n} = g_{t_1,\ldots,t_n}, \qquad (22)$$

$$\boldsymbol{G} = \left[\underbrace{\begin{matrix} g_{1,1,1,1} & \cdots & g_{1,T,1,1} \\ \vdots & \ddots & \vdots \\ g_{T,1,1,1} & \cdots & g_{T,T,1,1} \end{matrix}}_{T\times T} \quad \underbrace{\begin{matrix} g_{1,1,2,1} & \cdots & g_{1,T,2,1} \\ \vdots & \ddots & \vdots \\ g_{T,1,2,1} & \cdots & g_{T,T,2,1} \end{matrix}}_{T\times T} \quad \begin{matrix} \ldots \\ \ldots \\ \ldots \end{matrix} \quad \underbrace{\begin{matrix} g_{1,1,T,T} & \cdots & g_{1,T,T,T} \\ \vdots & \ddots & \vdots \\ g_{T,1,T,T} & \cdots & g_{T,T,T,T} \end{matrix}}_{T\times T} \right]. \qquad (23)$$

where $\boldsymbol{G}$ is $T \times T^{n-1}$. As an example, a $T \times T \times T \times T$ tensor $\mathcal{G}$ is unfolded by (22) as (23) at the bottom of this page.

Proposition 3. *Unfolding the tensor $\mathcal{G}$ in (7) to be the matrix $\boldsymbol{G}$ by (22), we have*

$$\boldsymbol{G} = \boldsymbol{P} \cdot (\underbrace{\boldsymbol{P} \odot \cdots \odot \boldsymbol{P}}_{n-1 \text{ matrices}})^T \in \mathbb{R}^{T \times T^{n-1}}. \qquad (24)$$

Proof. The proof is given in Appendix C. □

4.3 Detecting Cluster Number by Searching the Gap in the Ordered Eigenvalue Sequence

In (24), $\mathrm{rank}(\boldsymbol{P}) = K$. So, the dimension of the *cluster subspace* spanned by the columns of $\boldsymbol{P}$ is also K, whereas its orthogonal complement is *noise subspace* whose dimension is $T - K$. This algebraic structure allows us to detect the dimension of cluster subspace by EVD.

Perform EVD as

$$\mathrm{EVD}\left(\frac{1}{T^{n-1}}\boldsymbol{G}\boldsymbol{G}^T\right) = \boldsymbol{U}\Lambda \boldsymbol{U}^T, \tag{25}$$

where $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_T)$. Suppose that the T eigenvalues are sorted to be $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_T$. The $(T - K)$ smallest eigenvalues should be zeros, i.e.,

$$\lambda_1 \geq \cdots \geq \lambda_K > 0 = \lambda_{K+1} = \cdots = \lambda_T. \tag{26}$$

Heuristically, we can detect the cluster number by counting the number of nonzero eigenvalues. However, the exact eigenvalues $\lambda_1, \ldots, \lambda_T$ are not available because the true $\mathcal{G}$ is not known in practice but replaced by its approximation $\widehat{\mathcal{G}}$ estimated from $\mathcal{V}$. Correspondingly, $\boldsymbol{G}$ in (24) will be substituted with its estimation $\widehat{\boldsymbol{G}}$. To compute $\widehat{\boldsymbol{G}}$, perform the matrix representation operator $\mathrm{mat}(\cdot)$ on the tensor (13) in the same manner as (22) to

$$\mathrm{mat}(\widehat{\mathcal{G}}) = \mathrm{mat}(\mathcal{G}) + \mathrm{mat}(\mathcal{E}),$$

i.e.,

$$\widehat{\boldsymbol{G}} = \boldsymbol{G} + \boldsymbol{E}, \tag{27}$$

where

$$(\widehat{\boldsymbol{G}})_{t_1,t_2+(t_3-1)T+\cdots+(t_n-1)T^{n-2}} = (\widehat{\mathcal{G}})_{t_1,\ldots,t_n} = \widehat{g}_{t_1,\ldots,t_n}, \tag{28}$$

and the error term $\boldsymbol{E} = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{T^{n-1}}] \in \mathrm{IR}^{T\times T^{n-1}}$ is given by

$$(\boldsymbol{E})_{t_1,t_2+(t_3-1)T+\cdots+(t_n-1)T^{n-2}} = (\mathcal{E})_{t_1,\ldots,t_n} = \varepsilon_{t_1,\ldots,t_n}. \tag{29}$$

Proposition 4. *Suppose that the approximation errors* $(\varepsilon_{t_1,\ldots,t_n})_{T\times\cdots\times T}$ *in (14) are mutually independent for different subscript sets* $\{t_1, \ldots, t_n\}$ *and follow the identical Gaussian distribution* $N(0, \sigma_\varepsilon^2)$, *where* σ_ε^2 *is the variance. Then,*

$$\frac{1}{T^{n-1}}\widehat{\boldsymbol{G}}\widehat{\boldsymbol{G}}^T \longrightarrow \boldsymbol{U}\cdot\boldsymbol{\Lambda}\cdot\boldsymbol{U}^T + \sigma_\varepsilon^2\boldsymbol{I}, \tag{30}$$

where $\boldsymbol{I}$ *is an* identity matrix.

Proof. The proof is given in Appendix D. □

Applying EVD on (30), we can obtain

$$\mathrm{EVD}\left(\frac{1}{T^{n-1}}\widehat{\boldsymbol{G}}\widehat{\boldsymbol{G}}^T\right) = \boldsymbol{U}\widehat{\Lambda}\boldsymbol{U}^T \longrightarrow \boldsymbol{U}\cdot\boldsymbol{\Lambda}\cdot\boldsymbol{U}^T + \sigma_\varepsilon^2\boldsymbol{I},$$

i.e.,

$$\boldsymbol{U}\widehat{\Lambda}\boldsymbol{U}^T \longrightarrow \boldsymbol{U}\cdot\left(\boldsymbol{\Lambda} + \sigma_\varepsilon^2\boldsymbol{I}\right)\cdot\boldsymbol{U}^T, \tag{31}$$

where

$$\widehat{\boldsymbol{\Lambda}} \longrightarrow \boldsymbol{\Lambda} + \sigma_\varepsilon^2\boldsymbol{I}.$$

So, we have

$$\widehat{\lambda}_t \longrightarrow \lambda_t + \sigma_\varepsilon^2,\ t = 1, \ldots, T. \tag{32}$$

Comparing (26) with (32), hopefully we can obtain

$$\widehat{\lambda}_1 \geq \cdots \geq \widehat{\lambda}_K > \widehat{\lambda}_{K+1} = \cdots = \widehat{\lambda}_T = \sigma_\varepsilon^2. \tag{33}$$

Equation (33) implies that there will exist a noticeable gap between $\widehat{\lambda}_K$ and $\widehat{\lambda}_{K+1}$ if $\widehat{\lambda}_K$ is significantly larger than $\widehat{\lambda}_{K+1}$. Then, intuitively, we can detect the cluster number by finding this gap.

Remark 2. Based on (33), theoretically, many model selection techniques can be used to detect the gap in the sequence $\{\widehat{\lambda}_k\}_{k=1}^T$, e.g., Bayesian information criterion (BIC) [32], the Laplace method [32], Stein's unbiased risk estimator (SURE) [33], Radoi and Quinquis's method [34], exponential fitting test (EFT) [35], [36], modified EFT (M-EFT) [27], etc. In most of such situations, these methods work well. However, the problem of n-way clustering is very special because $K \lll T$. In this special case, we observed in the experiments that the performance of BIC, Laplace method, and SURE was poor in n-way clustering. The EFT and M-EFT need to compute threshold coefficients. Next, we develop a computationally more efficient method for this problem.

Compute the differences of eigenvalues

$$\nabla\widehat{\lambda}_t = \widehat{\lambda}_t - \widehat{\lambda}_{t+1}, \tag{34}$$

where $t = 1, \ldots, T - 1$. From (33), we have

$$\nabla\widehat{\lambda}_{K+1} = \nabla\widehat{\lambda}_{K+2} = \cdots = \nabla\widehat{\lambda}_{T-1} = 0. \tag{35}$$

To detect the cluster gap, compute the variance of the sequence $\{\nabla\widehat{\lambda}_i\}_{i=k}^{T-1}$ as

$$\widehat{\sigma}_k^2 = \frac{1}{T-k}\sum_{i=k}^{T-1}\left(\nabla\widehat{\lambda}_i - \frac{1}{T-k}\sum_{i=k}^{T-1}\nabla\widehat{\lambda}_i\right)^2, \tag{36}$$

where $k = 1, \ldots, T - 1$. From (35) and (36), it is easy to check that

$$\begin{cases} \widehat{\sigma}_k^2 > 0,\ k = 1, \ldots, K, \\ \widehat{\sigma}_k^2 = 0,\ k = K+1, \ldots, T-1 \end{cases}. \tag{37}$$

Further, define a Second ORder sTatistic of the Eigenvalues (SORTE) as follows:

$$\mathrm{SORTE}(k) = \begin{cases} \dfrac{\widehat{\sigma}_{k+1}^2}{\widehat{\sigma}_k^2}, & \widehat{\sigma}_k^2 > 0, \\ +\infty, & \widehat{\sigma}_k^2 = 0, \end{cases} \tag{38}$$

where $k = 1, \ldots, T - 2$. Then, from (33) and the definition (38), we have

$$\begin{cases} \mathrm{SORTE}(k) > 0,\ k = 1, \ldots, K-1, \\ \mathrm{SORTE}(k) = 0,\ k = K, \\ \mathrm{SORTE}(k) = +\infty,\ k = K+1, \ldots, T-3, \\ \mathrm{SORTE}(k) = 0,\ k = T-2. \end{cases} \tag{39}$$

According to (39), we can perform the model selection by the following criterion:

$$\widehat{K} = \arg\min_{k=1,\ldots,T-3} \mathrm{SORTE}(k). \tag{40}$$

The detailed procedure for cluster number detection is described in Algorithm 1.

Algorithm 1. Detecting the number of clusters K for an n-way probabilistic clustering (n-way SORTE method)

```
1. Construct the $n$-way distance tensor $\mathcal{V}$ for $T$ data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$. For the point clustering, we can compute $\mathcal{V}$ by (10).
2. Compute the scaling parameter $\Delta^2$ according to the discussions suggested in Section 3.2.
3. Compute the probability affinity tensor $\widehat{\mathcal{G}}$ by (11) and normalize it;
4. Compute the matrix $\widehat{\boldsymbol{G}}$ from $\widehat{\mathcal{G}}$ by (28);
5. Carry out EVD decomposition on the matrix $\frac{1}{T^{n-1}}\widehat{\boldsymbol{G}}\widehat{\boldsymbol{G}}^T$ in (30) and get $\widehat{\boldsymbol{\Lambda}} = (\widehat{\lambda}_1, \ldots, \widehat{\lambda}_T)^T$ by (31) such that $\widehat{\lambda}_1 \geq \widehat{\lambda}_2 \geq \cdots \geq \widehat{\lambda}_T$;
6. Compute $\widehat{\sigma}_k^2$, $k = 1, \ldots, T-1$ by (36);
7. If necessary, set a proportion $p$ and compute the truncation parameter $J \leq T-3$ by (42).
8. Compute $\mathrm{SORTE}(k)$, $k = 1, \ldots, J$ by (38);
9. Detect $K$ in the sequences $\{\mathrm{SORTE}(k)\}_{k=1}^J$ according to the criterion (40).
```

It is worth mentioning that (33) will probably not exactly hold; rather, in most situations, it will be

$$\widehat{\lambda}_1 \geq \cdots \geq \widehat{\lambda}_K \geq \sigma_\varepsilon^2 + \delta > \sigma_\varepsilon^2 \geq \widehat{\lambda}_{K+1} \geq \cdots \geq \widehat{\lambda}_T > 0, \tag{41}$$

where δ is a positive number. A typical example is that $\widehat{\lambda}_{K+1}, \ldots, \widehat{\lambda}_T$ could be collinear, i.e., $\widehat{\lambda}_{K+1} - \widehat{\lambda}_{K+2} = \widehat{\lambda}_{K+2} - \widehat{\lambda}_{K+3} = \cdots = \widehat{\lambda}_{T-1} - \widehat{\lambda}_T = \text{const} \geq 0$, where (33) is a special case of this problem. Interestingly, the SORTE criterion (40) still works because we still have $\nabla\widehat{\lambda}_{K+1} = \nabla\widehat{\lambda}_{K+2} = \cdots = \nabla\widehat{\lambda}_T$ in this case and all equations from (37) to (39) hold in this case.

Remark 3. To suppress noise and reject possible outliers, we suggest considering only the first J SORTE values $\mathrm{SORTE}(k)$, $k = 1, \ldots, J$, and discarding the remaining $T-3-J$ ones in (38) by choosing an appropriate parameter $J \leq T-3$ such that the collected cumulative energy of the largest J eigenvalues is above a large percentage p, i.e.,

$$J = \arg\min_{J=1,\ldots,T-3} \left\{ J : \frac{\sum_{t=1}^J \widehat{\lambda}_t}{\sum_{t=1}^T \widehat{\lambda}_t} > p \right\}. \tag{42}$$

Empirically, the proportion parameter p usually can sufficiently approximate to 100 percent, for example, $p = 99\%$ (even $p > 99.99\%$). This truncation operation is optional.

After the cluster number K is estimated, we can estimate the assignment probability matrix $\boldsymbol{P}$ from (17) by PARAFAC decomposition [11], [12], [15], [37], or, more precisely, by solving an SS-NTF problem [3]. Then, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ can be assigned to K cluster $\Omega_1, \ldots, \Omega_K$ according to $\boldsymbol{P}$. Let $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K$ be the corresponding clustering centers. If necessary, we can estimate them by

$$\boldsymbol{c}_k = \frac{\sum_{t=1}^T p_{t,k}\boldsymbol{x}_t}{\sum_{t=1}^T p_{t,k}}, \ k = 1, \ldots, K.$$

5 Some Related Methods for Cluster Number Detection

Determining the number of clusters in a data set is a fundamental problem in cluster analysis [9], [38], [39], [40], [41], [42], [43], [44], [45]. Although this problem is still largely unresolved, numerous methods have been suggested for it. But, as a whole, these methods are mostly available for the pairwise clustering. They can be roughly divided into two categories: the information-theoretic methods [44], [46], [47], [48] and the gap-based methods [42], [45].

5.1 Information-Theoretic Methods

The information-theoretic methods assume that $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ follow a Gaussian mixture model (GMM):

$$\Pr(\boldsymbol{x}) = \sum_{k=1}^K w_k N(\boldsymbol{x} \in \mathbb{R}^{m\times 1} | \boldsymbol{\mu}_k, \Sigma_k), \tag{43}$$

where the weight coefficients $w_k \geq 0$, $k = 1, \ldots, K$, and $\sum_{k=1}^K w_k = 1$. $N(\boldsymbol{x}|\boldsymbol{\mu}_k, \Sigma_k)$, $k = 1, \ldots, K$, respectively, denote K probability density functions (PDF) of m-variate Gaussian/Normal distribution with mean vector $\boldsymbol{\mu}_k$ and covariance matrix Σ_k. So, the parameter set with K clusters should be

$$\Theta(K) = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K; \Sigma_1, \ldots, \Sigma_K; w_1, \ldots, w_K\}. \tag{44}$$

Usually, we can estimate the parameters $\Theta(K)$ by *maximizing likelihood* [47], [48], i.e.,

$$\widehat{\Theta}(K) = \arg\begin{cases} \max\limits_{\Theta(K)} L(\Theta(K)) \\ \text{subject to}: \ w_k \geq 0, k = 1, \ldots, K, \\ \qquad\qquad \sum_{k=1}^K w_k = 1, \end{cases} \tag{45}$$

where the likelihood function $L(\Theta(K))$ is given by

$$L(\Theta(K)) = \log\prod_{t=1}^T \Pr(\boldsymbol{x}_t) = \sum_{t=1}^T \log\sum_{k=1}^K w_k N(\boldsymbol{x}_t|\boldsymbol{\mu}_k, \Sigma_k). \tag{46}$$

A common tool to optimize (45) is the *expectation-maximization* (EM) algorithm [8], [49].

The information-theoretic methods are implemented in three steps: in the first step, we roughly give the value range of K from K_{min} to K_{max} in which it is intended that the optimal K be contained; second, for each integer $k \in [K_{min}, K_{max}]$, we compute the parameters $\widehat{\Theta}(k)$ by (45) and obtain $L(\widehat{\Theta}(k))$ by (46); in the third step, we estimate K by

$$\widehat{K} = \arg\min_k\{J(\widehat{\Theta}(k)), k = K_{min}, \ldots, K_{max}\}, \tag{47}$$

where $J(\widehat{\Theta}(k))$ is an appropriate information-theoretic criterion. The popular information-theoretic criteria include Akaike's information criterion (AIC) [50], the consistent AIC (CAIC) [51], the minimum description length (MDL) criterion [52], [53], etc. They can be organized in a unified form as [47], [54]

$$J(\widehat{\Theta}(k)) = -2L(\widehat{\Theta}(K)) + A(T) \cdot D(k), \tag{48}$$

where $D(k) = (k-1) + k[m + m(m+1)/2]$ and $A(T)$ is $A(T) = 2$ for AIC [50], $A(T) = \log T + 1$ for CAIC [51], and $A(T) = \log T$ for MDL [48], [52].

In addition, under the information-theoretic framework, recently, there have been other modified cluster number detection methods, such as kernel MDL (KMDL) [48], BYY-HDS method [47], etc. KMDL is implemented by adapting MDL to kernel K-means clustering. In the BYY-HDS method, the information-theoretic criterion $J(\widehat{\Theta}(k))$ in (48) is replaced by the BYY-HDS criterion. By experiments, it was demonstrated that these methods can improve the performance to some extent [47], [48].

5.2 Gap-Based Detection Methods

Besides the information-theoretic methods, the gap/jump-based detection methods identify the number of clusters by searching a significant gap or jump in a properly defined monotonous index sequence. These works include Calinski and Harabasz's index [55], Hartigan's statistic [56], Krzanowski and Lai's index [39], Kaufman and Rousseeuw's *sihouette* statistic [9], Tibshirani et al.'s gap statistic [42], Sugar's jump method [44], Yan and Ye's weighted gap statistic [45], and so on. Here, we briefly introduce three relatively recent methods for later comparison in the experimental section.

The first one is the Gap method sugested by Tibshirani et al. [42], which finds the solution in the following procedure:

Step 1: Varying the total number of clusters from $k = 1$ to K_{max}, partition the observed data points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_T$ into k clusters $\Omega_1, \dots, \Omega_k$ by an appropriate clustering method (e.g., K-means).

Step 2: Compute the within-dispersion measures W_k, $k = 1, \dots, K_{max}$, by

$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r, \tag{49}$$

where $D_r = \sum_{t,t' \in \Omega_r} d_{tt'}$ is the sum of pairwise distance for all points in cluster Ω_r and $n_r = |\Omega_r|$ is the number of data points in Ω_r. $d_{tt'}$ denotes the distance between data points $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t'}$. The most common choice is the squared euclidean distance, $d_{tt'} = \|\boldsymbol{x}_t - \boldsymbol{x}_{t'}\|_2^2$.

Step 3: Given the prior/reference distribution[1] of $\log(W_k)$, compute its expectation $\mathbb{E}[\log(W_k)]$ and standard deviation $\mathbb{D}[\log(W_k)]$. Then, compute its (estimated) gap statistic

$$\mathrm{Gap}(k) = \mathbb{E}[\log(W_k)] - \log(W_k),\ k = 1, \dots, K_{max}. \tag{50}$$

Step 4: choose the number of clusters by the criterion:

$$\widehat{K} = \min_{k=1,\dots,K_{max}-1} \{k : \mathrm{Gap}(k) \geq \mathrm{Gap}(k+1) - \mathbb{D}[\log(W_{k+1})]\}. \tag{51}$$

The Gap method was very recently improved to the weighted Gap (WGap) method by Yan and Ye [45]. It was reported that the weighted gap method is more robust than the Gap method [45]. One of the main differences between them [45] is that the within-dispersion measure W_k in (49) is replaced by its corresponding weighted within-dispersion measure $\overline{W}_k$, given as

$$\overline{W}_k = \sum_{r=1}^{k} \frac{1}{2n_r} \overline{D}_r = \sum_{r=1}^{k} \frac{1}{2n_r(n_r - 1)} D_r. \tag{52}$$

Then, a weighted gap statistic is analogously defined as

$$\overline{\mathrm{Gap}}(k) = \mathbb{E}[\log(\overline{W}_k)] - \log(\overline{W}_k),\ k = 1, \dots, K_{max}. \tag{53}$$

Denote

$$D\overline{\mathrm{Gap}}(k) = \overline{\mathrm{Gap}}(k) - \overline{\mathrm{Gap}}(k-1)$$

and

$$DD\overline{\mathrm{Gap}}(k) = D\overline{\mathrm{Gap}}(k) - D\overline{\mathrm{Gap}}(k+1),$$

where $k = 2, \dots, K_{max} - 1$. The WGap method [45] identifies the cluster number according to the following criterion:

$$\widehat{K} = \arg \max_{k=2,\dots,K_{max}-1} DD\overline{\mathrm{Gap}}(k).$$

Another alternative method is the jump method proposed by Sugar and James [44]. It is also efficient and straightforward to implement. The jump method incorporates the average Mahalanobis distance,

$$\widehat{d}_K = \frac{1}{m} \min_{\boldsymbol{c}_1,\dots,\boldsymbol{c}_K} \mathbb{E}\left[(\boldsymbol{X} - \boldsymbol{c}_x)^T \boldsymbol{\Gamma}^{-1} (\boldsymbol{X} - \boldsymbol{c}_x)\right],$$

between $\boldsymbol{X}$ and $\boldsymbol{c}_x$, where $\boldsymbol{c}_1, \dots, \boldsymbol{c}_K$ is a set of candidate cluster centers and $\boldsymbol{c}_x$ is the one closest to $\boldsymbol{X}$. It is assumed that all clusters have the same covariance $\boldsymbol{\Gamma}$ [44]. The jump method estimates the K by

$$\widehat{K} = \arg \max_{k=2,\dots,T} \left[\widehat{d}_k^{-Y} - \widehat{d}_{k-1}^{-Y}\right],$$

where it is suggested that $Y = m/2$ in [44].

5.3 Some Properties of the Developed Cluster Number Detection Framework in Comparison to Above Existing Ones

The developed SORTE Algorithm 1 is also based on gap detection. However, differing from most of the existing methods, including the information-theoretic methods and the other gap methods, under our framework "clustering the data set $\boldsymbol{X}$" and "finding the number of clusters" are two separate procedures. Our SORTE method actually does not involve any clusterings at all. Additionally, our method is advantageous over the existing ones in the following aspects:

1. Interestingly, our SORTE method is not dependent on the dimension m of the data set $\boldsymbol{X}$. From Algorithm 1, it can be seen that the input data is only the distance affinity $\mathcal{V}$. The SORTE method gives a good answer (or scheme) to the open issue mentioned in [42]: how to carry out the gap test when the dimension m of the data is unknown and only pairwise dissimilarities are available.
2. The SORTE method does not necessarily work with any clustering algorithms. In other words, it is

[1] In [42], two prior distributions were considered: uniform distribution and uniform distribution over a box aligned with principal components of the data. Correspondingly, they are denoted as Gap/unif and Gap/pc, respectively.

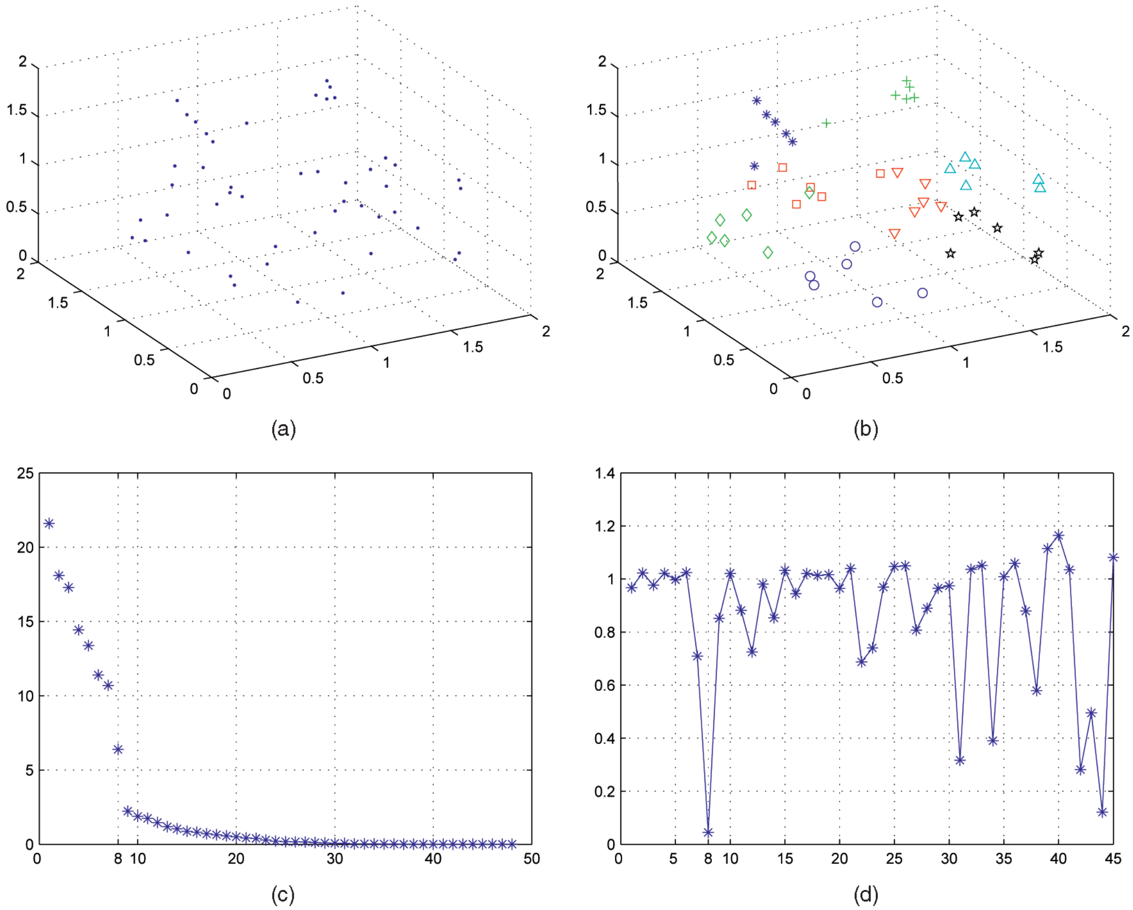


Fig. 2. Detecting the cluster number in a pairwise clustering ($\mathrm{SNR} = 16$ dB). (a) Eight clusters uniformly locate in above cube. (b) The same eight clusters labeled in different markers, respectively. (c) Forty-eight eigenvalues are sorted in descending order. (d) The SORTE curve. By searching the minimum SORTE-value, it is clear that we should have $K = 8$.

independent on the clustering methods. Many existing methods (e.g., KMDL [48] and the gap statistic method [42]) cannot work solely. They must collaborate with a certain clustering algorithm (e.g., K-means clustering), which directly leads to the problem that these cluster number detection algorithms will fail, if their collaborator cannot correctly find all clusters.

3. As mentioned previously, the SORTE method is available for the general n-way clustering, whereas the existing ones are limited to the pairwise clustering or cannot be conveniently extended to n-way case.
4. Even for the pairwise clustering, the SORTE method is more efficient in computation than the existing ones, especially when the cluster number is large. For information-theoretic methods, we need to repeatedly estimate the parameter set $\Theta(K)$ by (45), which involves a lot of nonlinear computations. Furthermore, the computational cost of the other gap-based methods is also much higher than that of our SORTE method. For them, we must also select an appropriate clustering algorithm first, and perform cluster analysis on the data set $\boldsymbol{X}$ at least K_{max} times in Step 1 by varying k from 1 to K_{max}.

In contrast, our method needs to compute Δ^2 for estimating $\widehat{\mathcal{G}}$ in (11). This additional computation is not necessary for other methods.

6 Experimental Results

In this section, we illustrate the performance of the cluster number detection algorithm by experiments. All data sets were tested in Matlab 7.1, and were run on an IBM laptop computer with Intel Pentium CPU 1.73 GHz under Windows XP Professional. In all experiments, the algorithm parameters are taken as follows: The cumulative proportion $p = 99\%$ in (42); for three-way or higher-order way cluster number detection, Δ^2 is set according to (20a) and (20b), where $\alpha = 100 \times 0.1^{n-2}$ $(n \geq 3)$ if no special mention is given; for two-way clustering, Δ^2 is given by (21) and $\beta = 10$.

6.1 Experiments on Synthetic Data

Example 1. Consider the following clustering problem (see Fig. 2): Eight clusters uniformly locate in a cube and are corrupted with white Gaussian noise; the signal-to-noise ratios (SNR)[2] are, respectively, listed in Table 2. There are six sample points in each cluster. So, in total, there are $T = 8 \times 6 = 48$ sample points (see Fig. 2a). This benchmark was performed with 100 Monte Carlo simulations by setting different seeds when generating the white Gaussian noise.

[2] Throughout this paper, the noise is measured using the definition of SNR: $\mathrm{SNR}(s, n) = 20 \cdot \log \frac{\|s\|_2}{\|n\|_2}$ [dB].

TABLE 2
The Percentages of Correct Detection (PoD) of 100 Monte Carlo Trials

SNR	18dB	17dB	16dB	15dB	14dB	13dB
Pairwise SORTE	100%	99%	98%	73%	25%	3%
Three-way SORTE	100%	100%	98%	88%	57%	21%
Four-way SORTE	100%	99%	98%	81%	53%	23%

Compute the n-way distance $\mathcal{V}$ using (10). We respectively applied the pairwise SORTE method, three-way SORTE method, and four-way SORTE method to each data set in different noise level. The percentages of correct detection (PoD) of 100 Monte Carlo tests are given in Table 2. Fig. 2, respectively, shows the scatter plot, ordered eigenvalues, and SORTE curve of one of the Monte Carlo simulations with $\mathrm{SNR} = 16$ dB. Due to noise, directly observing the scatter plot in Fig. 2a, it is very difficult to determine the number of clusters. However, it is much easier by our proposed scheme.

$$\boldsymbol{D} = \begin{bmatrix} 0.9065 & 0.1122 & 0.6014 & 0.4455 & 0.4057 & 0.6609 & 0.1654 & 0.4958 & 0.8522 & 0.1280 \\ 0.2378 & 0.9937 & 0.5656 & 0.8636 & 0.7590 & 0.3621 & 0.4856 & 0.6150 & 0.4832 & 0.9252 \\ 0.3489 & 0.0001 & 0.5643 & 0.2359 & 0.5093 & 0.6573 & 0.8584 & 0.6132 & 0.2008 & 0.3573 \end{bmatrix}. \tag{54}$$

It is observed in Table 2 that PoDs of three-way SORTE and four-way SORTE are higher than those of pairwise SORTE when the noise is relatively strong. The reason for this problem is that there is an important parameter Δ^2 in (11) which must be given in advance. In the case of pairwise clustering, Δ^2 can take a relatively large value in the very noisy scenarios when the number of clusters is small. So, it is difficult to precisely set the value for Δ^2. By (21), we can just give an approximate value for Δ^2 (but maybe not exactly optimal) in pairwise clustering. However, in the higher-way clustering, the value of Δ^2 is always small. In our experimental experience, the expression $\alpha = 100 \times 0.1^{n-2}$ $(n \geq 3)$ suggested in Section 3.2 works well and it can set a good value for Δ^2 which usually is sufficiently close to the optimal value. From experiments, we observed that, if the optima of Δ^2 can be provided, three SORTE methods (pairwise SORTE, three-way SORTE, and four-way SORTE) have almost equal PoDs. This can be seen in Table 3, in which all PoDs were obtained on the same data sets as in Table 2 by manually choosing the optimum for Δ^2 in (11) for each noise level: We first tried a series of candidate values for Δ^2 for each SORTE method and for each noise level, and followed by computing the PoD for every possible candidate value of Δ^2, then we chose the best one for Table 3 from those PoDs.

The truncation operation (42) can improve PoD in the noisy situations. In Table 2, we used the the cumulative proportion $p = 99\%$. Without using p but letting J be as great as $J = T - 3$ in (42), the results will be worse, especially when the noise is strong. We confirmed this by comparing Table 2 with Table 4. In Table 4, we reapplied the same methods on the same benchmarks as Table 2 without any help from parameter p.

TABLE 4
The PoDs of 100 Monte Carlo Trials without Using p by (42)

SNR	18dB	17dB	16dB	15dB	14dB	13dB
Pairwise SORTE	85%	78%	63%	34%	8%	1%
Three-way SORTE	96%	93%	84%	69%	31%	11%
Four-way SORTE	95%	89%	86%	64%	34%	16%

Incidentally, through this example we would like to roughly check whether all PARAFAC factors $\boldsymbol{P}_1, \dots, \boldsymbol{P}_n$ in (7) can be approximately equal. For this purpose, $\boldsymbol{P}_1, \dots, \boldsymbol{P}_n$ were computed by the alternating least-squares (ALS) method [16], [57], in each Monte Carlo trial. For the pairwise case, considering that the matrix factorization $\boldsymbol{G} = \boldsymbol{P}_1 \times \boldsymbol{P}_2^T$ is not unique, we solve $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ using the tool of nonnegative matrix factorization (NMF) [16], [58], [59] by imposing the nonnegativity constraints $\boldsymbol{P}_1 \geq 0$ and $\boldsymbol{P}_2 \geq 0$. Furthermore, to measure the differences among $\boldsymbol{P}_1, \dots, \boldsymbol{P}_n$, we define a relative squared error (RSE)

$$\mathrm{RSE}(\boldsymbol{P}_1, \dots, \boldsymbol{P}_n) = \frac{\frac{2}{n(n-1)} \sum_{1 \leq i < j \leq n} \|\boldsymbol{P}_i - \boldsymbol{P}_j\|_2^2}{\frac{1}{n} \sum_{i=1}^{n} \|\boldsymbol{P}_i\|_2^2},$$

where $\boldsymbol{P}_1, \dots, \boldsymbol{P}_n$ are normalized such that $\|\boldsymbol{P}_1\|_2 = \cdots = \|\boldsymbol{P}_n\|_2$.

For every n-way clustering in different noise levels, we respectively computed the RSE in each Monte Carlo simulation. The average RSEs of 100 Monte Carlo simulations are shown in Table 5. We can see that all average RSEs are small, which implies that we have $\boldsymbol{P}_1 \approx \boldsymbol{P}_2$ in pairwise clustering, $\boldsymbol{P}_1 \approx \boldsymbol{P}_2 \approx \boldsymbol{P}_3$ in three-way clustering, and $\boldsymbol{P}_1 \approx \boldsymbol{P}_2 \approx \boldsymbol{P}_3 \approx \boldsymbol{P}_4$ in four-way clustering.

Example 2. In this example, we compare our method with some existing ones for pairwise clustering. As mentioned in Section 5.3, the advantages of our method are obvious in terms of computational complexity and easy implementation. Here, we just simply compare them through three synthetic cluster benchmarks: a 5-cluster model, an 8-cluster model, and a 10-cluster model.

The data sets are generated in three steps. In the first step, a nonnegative 3×10 seed matrix $\boldsymbol{D}$, shown in (54), is generated randomly. The positions of its 10 columns, denoted as $\mathrm{d1}, \dots, \mathrm{d10}$, are labeled by different markers in Fig. 3. Second, based on matrix $\boldsymbol{D}$, three 3D point sequences with 200 samples are, respectively, generated: a 5-point sequence, an 8-point

TABLE 3
The Best PoDs of 100 Monte Carlo Trials by Manually Choosing the Optimum for Δ^2 in (11)

SNR	18dB	17dB	16dB	15dB	14dB	13dB
Best pairwise SORTE	100%	99%	97%	88%	60%	24%
Three-way SORTE	100%	100%	98%	88%	58%	24%
Four-way SORTE	100%	99%	98%	86%	55%	23%

TABLE 5
The Average Relative Squared Error (RSE) of 100 Monte Carlo Trials

SNR	18dB	17dB	16dB	15dB	14dB	13dB
Pairwise SORTE	0.060	0.073	0.085	0.097	0.113	0.129
Three-way SORTE	0.008	0.011	0.016	0.021	0.027	0.032
Four-way SORTE	0.017	0.022	0.023	0.034	0.037	0.041

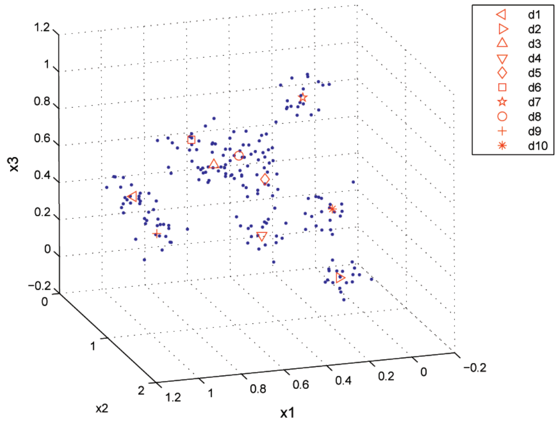


Fig. 3. Scatter plot of a 10-cluster data set with white Gaussian noise $\mathrm{SNR} = 30$ dB. Ten cluster centers are sequentially pointed out by different markers.

sequence, and a 10-point sequence. In each point sequence, there are 200 samples $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{200}]$, which are uniformly and randomly drawn from the columns of $\boldsymbol{D}$. Precisely, in the 5-point sequence, 200 samples in $\boldsymbol{Y}$ are uniformly and randomly drawn from the first five columns of $\boldsymbol{D}$, $\boldsymbol{Y}$ is analogously drawn from the first eight columns of $\boldsymbol{D}$ in the 8-point sequence, and all 10 columns in $\boldsymbol{D}$ are uniformly involved in the 10-point sequence. Finally, we obtain the ultimate observed data points $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{200}] = \boldsymbol{Y} + \boldsymbol{N}$ by adding a certain percent of white Gaussian noise $\boldsymbol{N} = [\boldsymbol{n}_1, \ldots, \boldsymbol{n}_{200}]$ to $\boldsymbol{Y}$. The SNRs are shown in Table 6. For each model and noise level, 100 Monte Carlo simulations were performed. As an example, the 10-cluster model with $\mathrm{SNR} = 30$ dB of one of Monte Carlo simulations is shown in Fig. 3.

To perform the comparative methods, the K-means algorithm was used to partition 200 data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{200}$ into k clusters $\Omega_1, \ldots, \Omega_k$, varying the cluster number k from $k = 1$ to $K_{max} = 15$. From the clustering results, we respectively estimated the cluster number for each data set under different noise levels using several recent methods. Meanwhile, we also tested our pairwise SORTE method for the same data sets in the same noise level. All PoDs of 100 Monte Carlo simulations are shown in Table 6. From this example, it can be seen that, relatively, the Gap methods, especially the Gap/unif, achieved a little worse estimation than the others. On the whole, the SORTE method slightly outperformed the existing ones. At least, it is comparable to others.

Example 3. The experimental setup totally consists of 200 observed sample points $\{\boldsymbol{x}_t\}_{t=1}^{200}$ in $\mathbb{R}^3$, which are randomly taken from five planes and contaminated by white Gaussian noise with SNR = 25 dB. By choosing different seeds to generate noise, 100 Monte Carlo tests were similarly conducted, where the scatter plot of $\{\boldsymbol{x}_t\}_{t=1}^{200}$ of one of Monte Carlo tests is shown in Fig. 4a. We would like to detect the appropriate plane number from $\{\boldsymbol{x}_t\}_{t=1}^{200}$.

Notice that combining with the origin O, any two different sample points $\boldsymbol{x}_{t_1}$ and $\boldsymbol{x}_{t_2}$ in $\{\boldsymbol{x}_t\}_{t=1}^{200}$ can produce a plane in $\mathbb{R}^3$. Totally nearly $200^2 = 40{,}000$ planes can be produced. However, only at most five candidates among

TABLE 6
Percentages of Correct Detection (PoD) of 100 Monte Carlo Tests in a Comparative Study

Method	SNR=100 dB	SNR=60 dB	SNR=50 dB	SNR=40 dB	SNR=30 dB	SNR=25 dB	SNR=22 dB
5-cluster model							
AIC	75%	75%	75%	75%	75%	73%	74%
CAIC	93%	95%	94%	95%	91%	95%	92%
MDL	92%	94%	93%	95%	91%	94%	92%
Gap/unif	2%	2%	2%	2%	2%	1%	1%
Gap/pc	31%	31%	30%	28%	27%	22%	18%
DDGap/unif	100%	100%	100%	100%	100%	100%	64%
DDGap/pc	100%	100%	100%	100%	100%	100%	64%
Jump	100%	100%	100%	100%	100%	100%	100%
SORTE	100%	100%	100%	100%	100%	100%	100%
8-cluster model							
AIC	63%	63%	63%	64%	64%	63%	33%
CAIC	87%	87%	87%	85%	84%	26%	2%
MDL	84%	83%	83%	82%	80%	46%	6%
Gap/unif	66%	66%	66%	66%	66%	65%	62%
Gap/pc	84%	84%	84%	84%	84%	84%	79%
DDGap/unif	99%	99%	99%	99%	55%	0%	0%
DDGap/pc	99%	99%	99%	99%	54%	0%	0%
Jump	100%	100%	100%	100%	100%	4%	4%
SORTE	100%	100%	100%	100%	100%	96%	75%
10-cluster model							
AIC	40%	40%	40%	40%	40%	40%	30%
CAIC	81%	81%	81%	81%	81%	58%	39%
MDL	77%	77%	77%	77%	77%	60%	40%
Gap/unif	83%	83%	83%	82%	83%	77%	67%
Gap/pc	95%	95%	95%	94%	95%	74%	67%
DDGap/unif	97%	97%	97%	97%	70%	65%	35%
DDGap/pc	97%	97%	97%	97%	70%	65%	35%
Jump	100%	100%	100%	100%	100%	67%	54%
SORTE	100%	100%	100%	100%	100%	82%	55%

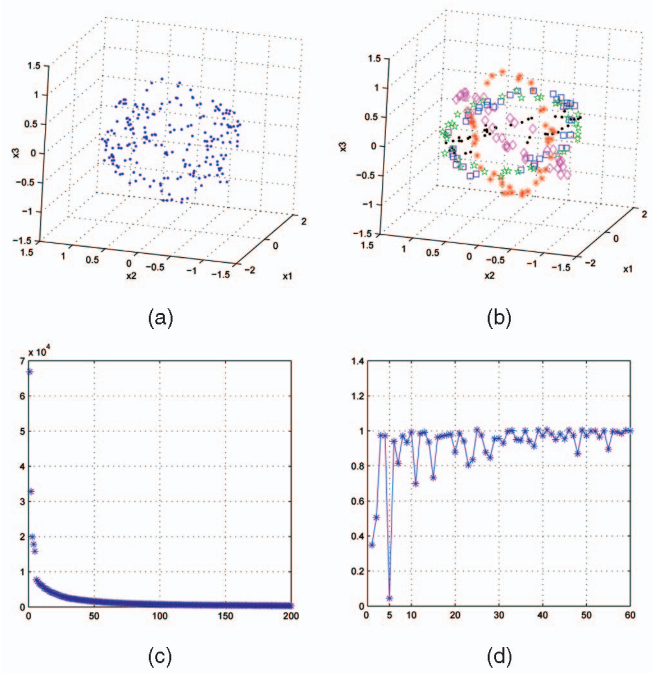


Fig. 4. Detect the number of planes hidden in three dimensions. It is shown that there are five planes. (a) The 3D scatter plot of 200 samples; (b) 200 data points from five planes are, respectively, indicated in different markers; (c) the ordered eigenvalues; (d) the first 60 SORTE values.

them potentially correspond to the true ones and the rest are spurious. To accomplish the detection, we must reject all fake planes. So, this problem is very challenging. It is a difficult task even for a human observer to precisely determine the number of planes if all observed samples are plotted in a completely blind manner as in Fig. 4a.

Fixing the indices i and k, define the set $\Gamma^{ik} = \{v_{ijk} : \ j = 1, \dots, i-1, i+1, \dots, T\}$. For convenience, sort the $T-1$ elements of set Γ^{ik} in an ascending order such that $v^{ik}_{(1)} \leq \cdots \leq v^{ik}_{(T-1)}$. In this example, the three-way distance affinity $\mathcal{V}$ is set as

$$v_{ijk} = \begin{cases} 0, & i = j = k, \\ \lambda_{\min}([\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k]^T [\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k]), & i \neq j, i \neq k, j \neq k, \\ \dfrac{1}{N} \displaystyle\sum_{j=1}^{N} v^{ik}_{(j)}, & \text{otherwise}, \end{cases}$$

where $i, j, k = 1, \dots, 200$, $N < T - 1$, and the operator $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue of a matrix. Here, we chose $N = 5$ in our experiment.

We applied the three-way SORTE method on $\{\boldsymbol{x}_t\}_{t=1}^{200}$ in each Monte Carlo trial. The PoD of 100 Monte Carlo tests is 97 percent. The ordered eigenvalues and SORTE sequence $\{\text{SORTE}(k)\}_{k=1}^{197}$ of one of Monte Carlo trials are, respectively, shown in Figs. 4c and 4d, both of which imply that the cluster number should be $\widehat{K} = 5$.

6.2 Experiments on the Real-World Data

Example 4. Detecting the number of distinct facial poses under varying illuminations: Given a collection of unlabeled faces $\{\boldsymbol{I}_t \in \mathbb{R}^{33\times 29}\}_{t=1}^{45}$ containing K different poses taken under varying illumination, we would like to determine the number of individual poses K. These face images, obtained from the same person, were downloaded from the Yale Face Database B [60] (see Fig. 5). To clearly show their difference, we put those faces, with the same pose but different illuminations, in the same row in Fig. 5. So, in total there are five rows corresponding to five individual poses, respectively.

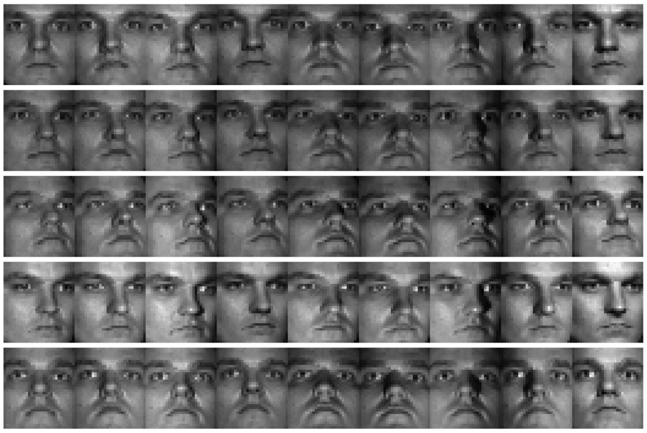


Fig. 5. Five sample facial poses, of the same person, under varying illumination conditions. These faces are drawn from the Yale database B. Each row shows the same pose with different illuminations.

By the method in [61], we first computed the *image gradient* $\nabla \boldsymbol{I}_t$ for each face and obtained $\{\nabla \boldsymbol{I}_t\}_{t=1}^{45}$. After this, we constructed the three-way distance affinity $\mathcal{V}$ as

$$v_{ijk} = \|\nabla \boldsymbol{I}_i - \nabla \boldsymbol{I}_j\|_2 + \|\nabla \boldsymbol{I}_i - \nabla \boldsymbol{I}_k\|_2 + \|\nabla \boldsymbol{I}_j - \nabla \boldsymbol{I}_k\|_2,$$

where $i, j, k = 1, \dots, 45$. Then, $\widehat{\mathcal{G}}$ was obtained by (11). Applying three-way cluster number detection algorithm 1 on $\widehat{\mathcal{G}}$, we get the results shown in Fig. 6 and obtain $K = 5$.

Example 5. Detect the number of different commands from a P300 EEG data set of a brain computer interface (BCI) [62], [63]. In this example, the P300 commands are, respectively, represented by different directional arrows (see Fig. 7a).

The subject focused attention on one of eight arrows in each epoch. Eight arrows was randomly selected during the session. But, for each selected arrow, the subject successively focused on it for 10 intensifications. In each intensification,

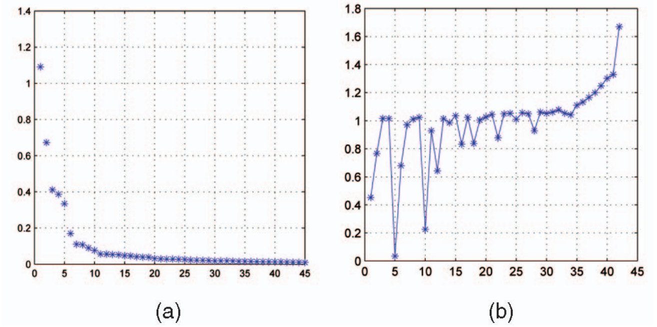
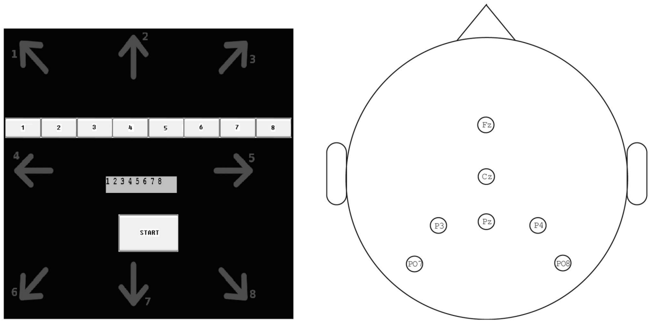


Fig. 6. Detecting pose number in three-way clustering. (a) The ordered eigenvalue sequence, in which the true gap is not very clear. (b) The SORTE sequence of three-way detection. Here, we can determine that the pose number is $K = 5$.



(a)

(b)

Fig. 7. The P300 experimental setup in Example 5. (a) Eight arrows on P300 screen. (b) Seven electrodes: Fz, Cz, Pz, P3, P4, PO7, and PO8.

the arrow was displayed for 70 ms. A 7-dimensional EEG data set $\{\boldsymbol{y}_t\}_{t=1}^{20,000}$ was collected by seven electrodes: Fz, Cz, Pz, P3, P4, PO7, and PO8. The objective is to detect how many individual arrows the subject had focused on during the experiment.

After preprocessing $\{\boldsymbol{y}_t\}_{t=1}^{20,000}$ and extracting the P300 features from it, we derived an 8×32 feature probability matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{32}]$.

Similarly, here all n-way distances $\mathcal{V}$ were given by (10). We respectively applied two-way, three-way, and four-way SORTE methods on $\boldsymbol{X}$. The detailed results are, respectively, shown in Fig. 8a, Fig. 8b, Fig. 8c, and Fig. 8d. It is shown that we correctly found the number of directional arrows in our experiments except for $\alpha = 10$ two-way detection. As mentioned at the end of Section 3, this is because $\alpha = 10$ violates the principle (20a) for this example in two-way clustering. From (18), we have

$$\gamma(\mathcal{V}; n) = \frac{\sum_{k=1}^{8} 4^n}{32^n} \times 100\% = \frac{1}{8^{n-1}} \times 100\%.$$

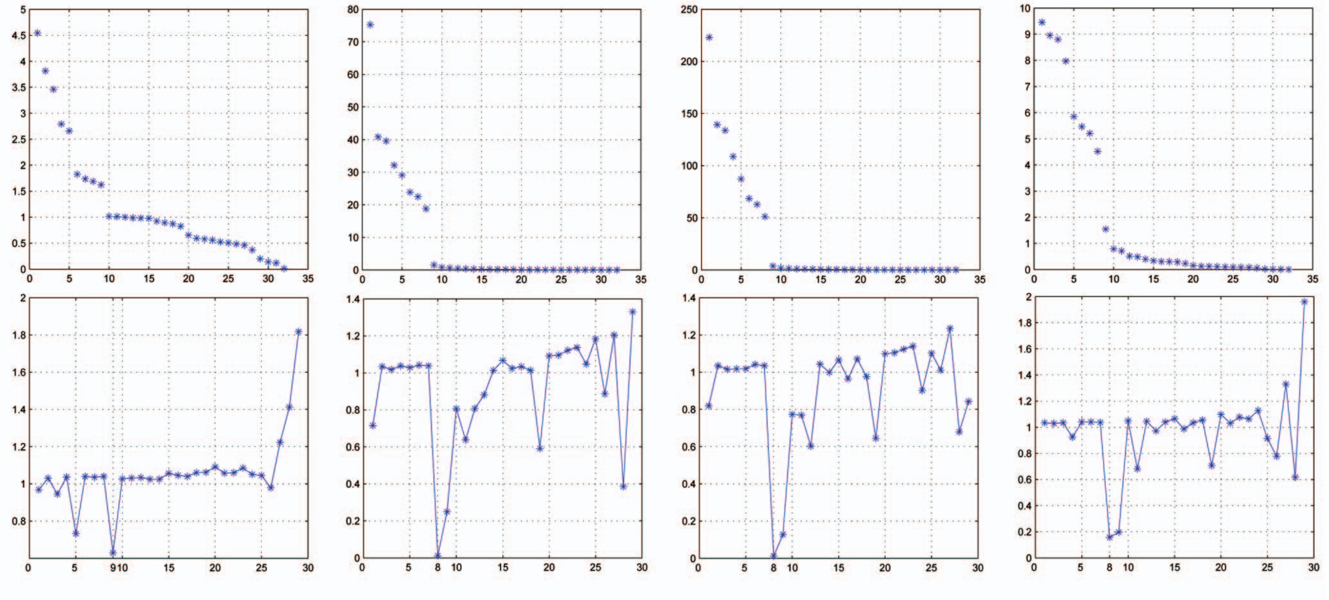
So, for two-way clustering (i.e, $n = 2$), $\gamma(\mathcal{V}) = 1/8 = 12.5\%$. According to (20a), the percentile α of the entries of $\mathcal{V}$ should satisfy $\alpha > \gamma(\mathcal{V}) = 12.5$. It was observed that for two-way detection, even $\alpha = 12.5$ failed. However, $\alpha = 12.6$ succeeded (see Fig. 8d).

7 Conclusions

The problem of detecting the cluster number K arises in many applications. If K is underestimated, it is impossible to achieve desirable clustering results. On the other hand, if K is overestimated, the clustering results are sensitive to noise. By precisely identifying the cluster number, it is not only helpful to reduce the computational complexity, but also to suppress noise.

The detection of the cluster number for the general n-way probabilistic clustering has been addressed in this paper. An efficient method has been developed for this problem. It is easy to implement and with low computational cost. It can be used as a preprocessing procedure in n-way probabilistic cluster analysis. Additionally, our discussions about the selection of the scaling parameter Δ^2 are helpful for the understanding of n-way probabilistic clustering [3].

In this paper, we mostly intend to develop a fast method to detect the number of clusters. Similarly to many conventional methods, the problem of cluster number detection addressed here also can be solved in a combined detection-estimation manner where one wants to estimate the number of the clusters K as well as the assignment probability matrix $\boldsymbol{P}$ in (17). Because of the computational complexity, this problem is usually solved in two steps: First, the number of clusters is detected, and then, with an estimate of cluster number $\widehat{K}$ at hand, $\boldsymbol{P}$ is updated again, where some criteria such as CORCONDIA and T-CORCONDIA can be exploited to this combined detection-estimation



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

Fig. 8. Identifying the number of P300 commands by two-way, three-way, and four-way detection, respectively. (a) Ordered eigenvalues and SORTE curve of two-way detection with $\alpha = 10$. For two-way detection, $\alpha = 10$ failed. (b) Three-way detection with $\alpha = 10$. (c) Four-way detection with $\alpha = 1$. (d) For two-way detection, $\alpha = 12.6$ found the solution correctly.

procedure [24], [30]. The evaluation of the corresponding criterion in this case involves the computation $\boldsymbol{P} = [\boldsymbol{p}_1, \ldots, \boldsymbol{p}_K]$ for every possible integer $K \in \{1, \ldots, T\}$. Solving this highly nonlinear problem for all values of K is computationally very expensive. Nevertheless, if one is ready to pay this price, the gain in terms of performance, especially in difficult situations such as strong noises, many outliers, small sample size, or huge overlapping, may be significant.

Generally speaking, detecting the number of clusters is more challenging than the pure clustering problem under the condition that the number of clusters is given. From experiments, we find that our method and also some other existing methods work if the clusters in a data set are not very heavily overlapped. However, due to the fact that there is no clear definition of a "cluster," this problem will be difficult and also subjective to some extent if the data points of the observed samples are not clearly separated into groups, especially if there exist many overlapping samples between the hidden clusters [42], where different people might have different options about the number of distinct clusters.

Appendix A

Proof of Proposition 1

Proof. From (9) and (11), we have

$$\hat{g}_{t_1,\ldots,t_n} = \begin{cases} 1, & \boldsymbol{x}_{t_1}, \ldots, \boldsymbol{x}_{t_n} \text{ are in the same cluster,} \\ e^{-\frac{v^2_{t_1,\ldots,t_n}}{\Delta^2}}, & \text{otherwise.} \end{cases} \tag{55}$$

Since the n-way probability clustering is "ideal," the probability assignment matrix $\boldsymbol{P}$ satisfies the conditions (12), from which we can get

$$\begin{aligned} &\sum_{k=1}^{K} p_{t_1,k} \cdots p_{t_n,k} \\ &\quad = \begin{cases} 1, \ \boldsymbol{x}_{t_1}, \ldots, \boldsymbol{x}_{t_n} \text{ are in the same cluster,} \\ 0, \ \text{otherwise.} \end{cases} \end{aligned} \tag{56}$$

From (55) and (56), we have

$$\begin{aligned} &\varepsilon_{t_1,\ldots,t_n}| \\ &\quad = \left| \hat{g}_{t_1,\ldots,t_n} - \sum_{k=1}^{K} p_{t_1,k}, \ldots, p_{t_n,k} \right| \\ &\quad = \begin{cases} 0, & \boldsymbol{x}_{t_1}, \ldots, \boldsymbol{x}_{t_n} \text{ are in the same cluster,} \\ e^{-\frac{v^2_{t_1,\ldots,t_n}}{\Delta^2}}, & \text{otherwise.} \end{cases} \\ &\quad \leq \ e^{-\frac{v^2_{t_1,\ldots,t_n}}{\Delta^2}}. \end{aligned} \tag{57}$$

Thus, the proof is completed. □

Appendix B

Proof of Proposition 2

Proof. For the cluster Ω_k, the number of its entries is $\#\{\Omega_k\}$. From (9), for an arbitrary n-tuple sample $\boldsymbol{x}_{t_1}, \ldots, \boldsymbol{x}_{t_n}$, we have

$$v_{t_1,\ldots,t_n} = 0, \text{ if } \bigcup_{i=1}^{n} \boldsymbol{x}_{t_i} \subseteq \Omega_k.$$

So, the cluster Ω_k produces $\#\{\Omega_k\}^n$ zero entries of $\mathcal{V}$. Thus, there are, in total, $\sum_{k=1}^{K} \#\{\Omega_k\}^n$ zero entries in $\mathcal{V}$. Additionally, $\#\{v_{t_1,\ldots,t_n} : t_1, \ldots, t_n = 1, \ldots, T\} = T^n$. Hence, we can obtain (18). □

Appendix C

Proof of Proposition 3

Proof. From (22), $\forall \tau \in \{1, \ldots, T^{n-1}\}$, $\exists$ a set of unique $\{t_i \in \{1, \ldots, T\}\}_{i=2}^{n}$ such that

$$t_i = \left[\left(\lceil \tau / T^{i-2} \rceil - 1 \right) \mathrm{mod}\, T \right] + 1, \ i = 2, \ldots, n, \tag{58}$$

where $\lceil \cdot \rceil$ denotes the *ceil round* and mod is a *modulo operator*. Conversely, $\forall$ set of $\{t_i \in \{1, \ldots, T\}\}_{i=2}^{n}$, $\exists$ a unique $\tau \in \{1, \ldots, T^{n-1}\}$ such that

$$\tau = t_2 + (t_3 - 1)T + \cdots + (t_n - 1)T^{n-2}. \tag{59}$$

From (58) and (59), we know that the following mapping:

$$M : \tau \in \{1, \ldots, T^{n-1}\} \mapsto \{t_i\}_{i=2}^{n} \tag{60}$$

is bijective (or one-to-one), where $t_i \in \{1, \ldots, T\}$.

Next, notice that $\boldsymbol{P} = [\boldsymbol{p}_1, \ldots, \boldsymbol{p}_K]$. Denote

$$\boldsymbol{Q} = \underbrace{\boldsymbol{P} \odot \cdots \odot \boldsymbol{P}}_{n-1}. \tag{61}$$

By the definition (1) of the Khatri-Rao product, we have

$$\boldsymbol{Q} = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_K] = [\underbrace{\boldsymbol{p}_1 \otimes \cdots \otimes \boldsymbol{p}_1}_{n-1}, \ldots, \underbrace{\boldsymbol{p}_K \otimes \cdots \otimes \boldsymbol{p}_K}_{n-1}]. \tag{62}$$

The size of $\boldsymbol{Q}$ is $T^{n-1} \times K$. For an arbitrary integer $\tau \in \{1, \ldots, T^{n-1}\}$, there always exists an appropriate set of integers $\{t_i\}_{i=2}^{n}$ satisfying (59). From (1), (62), and (59), we have

$$\begin{aligned} (\boldsymbol{Q})_{\tau,k} = (\boldsymbol{q}_k)_{\tau,1} &= (\underbrace{\boldsymbol{p}_k \otimes \cdots \otimes \boldsymbol{p}_k}_{n-1})_{\tau,1} \\ &= (\boldsymbol{p}_k \otimes \cdots \otimes \boldsymbol{p}_k)_{t_2+(t_3-1)T+\cdots+(t_n-1)T^{n-2},1} \\ &= p_{t_2,k} \cdot p_{t_3,k} \cdots p_{t_n,k} \\ &= \prod_{i=2}^{n} p_{t_i,k}. \end{aligned} \tag{63}$$

Combining (5), (7), and (63), we have

$$\begin{aligned} (\boldsymbol{P} \cdot \boldsymbol{Q}^T)_{t_1,\tau} &= \sum_{k=1}^{K} p_{t_1,k} \cdot (\boldsymbol{Q})_{\tau,k} = \sum_{k=1}^{K} p_{t_1,k} \prod_{i=2}^{n} p_{t_i,k} \\ &= \sum_{k=1}^{K} \prod_{i=1}^{n} p_{t_i,k} = (\mathcal{G})_{t_1,\ldots,t_n} = g_{t_1,\ldots,t_n}. \end{aligned} \tag{64}$$

Equations (22) and (64) can yield

$$(\boldsymbol{P} \cdot \boldsymbol{Q}^T)_{t_1,\tau} = (\boldsymbol{G})_{t_1,t_2+(t_3-1)T+\cdots+(t_n-1)T^{n-2}}. \tag{65}$$

Let τ traverse all possible subscripts in $\{1, \dots, T^{n-1}\}$. Since the mapping (60) $M : \tau \mapsto \{t_2, \dots, t_n\}$ is bijective, from (65), we can derive

$$\boldsymbol{P} \cdot (\underbrace{\boldsymbol{P} \odot \cdots \odot \boldsymbol{P}}_{n-1})^T = \boldsymbol{P} \cdot \boldsymbol{Q}^T = \boldsymbol{G}.$$

□

Appendix D
Proof of Proposition 4

Proof. Since $(\varepsilon_{t_1,\dots,t_n})_{T\times\cdots\times T}$ are mutually independent for different subscript sets $\{t_1, \dots, t_n\}$ and follow the identical Gaussian distribution $N(0, \sigma_\varepsilon^2)$, we have

$$\mathbb{E}(e_{i,\tau}e_{j,\tau}) = \begin{cases} 0, & i \neq j, \\ \sigma_\varepsilon^2, & i = j, \end{cases} \tag{66}$$

where $\mathbb{E}(\cdot)$ denotes *mathematical expectation*, $e_{i,\tau} = (\boldsymbol{E})_{i,\tau}$ and $e_{j,\tau} = (\boldsymbol{E})_{j,\tau}$. By *Khinchin's laws of large numbers* [41], we have

$$\frac{1}{T^{n-1}} \sum_{\tau=1}^{T^{n-1}} e_{i,\tau}e_{j,\tau} = \frac{1}{T^{n-1}} (\boldsymbol{E}\boldsymbol{E}^T)_{i,j} \longrightarrow \mathbb{E}(e_{i,\tau}e_{j,\tau}) \tag{67}$$

because, usually, the number T^{n-1} is quite large (e.g., $T^{n-1} > 50$). Combining (66) with (67), we get

$$\frac{1}{T^{n-1}} \sum_{\tau=1}^{T^{n-1}} e_{i,\tau}e_{j,\tau} \longrightarrow \begin{cases} 0, & i \neq j, \\ \sigma_\varepsilon^2, & i = j, \end{cases} \tag{68}$$

i.e.,

$$\frac{1}{T^{n-1}} \boldsymbol{E}\boldsymbol{E}^T \longrightarrow \sigma_\varepsilon^2 \boldsymbol{I}_{T\times T}. \tag{69}$$

Moreover, by Khinchin's laws of large numbers again [41], we can derive

$$\frac{1}{T^{n-1}} \sum_{\tau=1}^{T^{n-1}} e_{i,\tau} \longrightarrow 0, \; i = 1, \dots, T. \tag{70}$$

Note that the probability affinity tensor $\mathcal{G}$ is bounded and $c = \max_{t_1,\dots,t_n} |g_{t_1,\dots,t_n}| \leq 1$. Thus, we have

$$\left| \frac{1}{T^{n-1}} \sum_{\tau=1}^{T^{n-1}} e_{i,\tau} g_{j,\tau} \right| \leq c \cdot \left| \frac{1}{T^{n-1}} \sum_{\tau=1}^{T^{n-1}} e_{i,\tau} \right| \longrightarrow 0, \tag{71}$$

for all $i, j = 1, \dots, T$. Hence,

$$\frac{1}{T^{n-1}} \boldsymbol{E}\boldsymbol{G}^T \longrightarrow 0 \text{ and } \frac{1}{T^{n-1}} \boldsymbol{G}\boldsymbol{E}^T \longrightarrow 0. \tag{72}$$

Equation (27) can yield

$$\begin{aligned} \frac{1}{T^{n-1}} \widehat{\boldsymbol{G}}\widehat{\boldsymbol{G}}^T = & \frac{1}{T^{n-1}} \boldsymbol{G}\boldsymbol{G}^T + \frac{1}{T^{n-1}} \boldsymbol{G}\boldsymbol{E}^T + \frac{1}{T^{n-1}} \boldsymbol{E}\boldsymbol{G}^T \\ & + \frac{1}{T^{n-1}} \boldsymbol{E}\boldsymbol{E}^T. \end{aligned} \tag{73}$$

From (25), (69), and (72), (73) can be simplified as

$$\frac{1}{T^{n-1}} \widehat{\boldsymbol{G}}\widehat{\boldsymbol{G}}^T \longrightarrow \boldsymbol{U} \cdot \boldsymbol{\Lambda} \cdot \boldsymbol{U}^T + \sigma_\varepsilon^2 \boldsymbol{I}.$$

□

Acknowledgments

The authors would like to thank the anonymous reviewers. Due to their comments and suggestions, both the technical quality and readability of this paper have been improved. The work is supported in part by the National Natural Science Foundation of China (Grants U0635001 and 60874061), National Basic Research Program of China (973 Program 2010CB731800), the Fundamental Research Funds for Central Univresity, SCUT (Grant x2xD2102070), and CPSF (Grants 20070410237, 200801248). Kyuwan Choi's work was partially supported by a grant from the National Institute of Information and Communications Technology project entitled "Multimodal integration for brain imaging measurements."

References

[1] D. Zhou, J. Huang, and B. Schölkopf, "Learning with Hypergraphs: Clustering, and Classification, Embedding," *Advances in Neural Information Processing Systems,* B. Schölkopf, J. Platt, and T. Hoffman, eds., vol. 19, pp. 1601-1608, MIT Press, 2007.

[2] C. Berge, *Hypergraphs,* first ed. North Holland, Aug. 1989.

[3] A. Shashua, R. Zass, and T. Hazan, "Multi-Way Clustering, Using Super-Symmetric Non-Negative Tensor Factorization," *Lecture Notes in Computer Science,* vol. 3954, pp. 595-608, Springer, July 2006.

[4] A. Banerjee, S. Basu, and S. Merugu, "Multi-Way Clustering on Relation Graphs," *Proc. SIAM Conf. Data Mining,* 2007.

[5] J.M. Buhmann and T. Hofmann, "A Maximum Entropy Approach to Pairwise Data Clustering," *Proc. 12th IAPR Int'l Conf. Pattern Recognition,* pp. 207-212, Oct. 1994.

[6] T. Hofmann and J. Buhmann, "Hierarchical Pairwise Data Clustering by Mean-Field Annealing," *Proc. Int'l Conf. Artificial Neural Networks,* pp. 197-202, 1995.

[7] T. Hofmann and J.M. Buhmann, "Pairwise Data Clustering by Deterministic Annealing," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 1, pp. 1-14, Jan. 1997.

[8] D.J.C. MacKay, *Information Theory, Inference and Learning Algorithms.* Cambridge Univ. Press, Sept. 2003.

[9] L. Kaufman and P.J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis.* Wiley Interscience, Mar. 2005.

[10] R. Zass and A. Shashua, "A Unifying Approach to Hard Probabilistic Clustering," *Proc. 10th IEEE Int'l Conf. Computer Vision,* vol. 1, pp. 294-301, 2005.

[11] H.A. Kiers, "Towards a Standardized Notation and Terminology in Multiway Analysis," *J. Chemometrics,* vol. 14, no. 3, pp. 105-122, 2000.

[12] L. De Lathauwer, "A Link between the Canonical Decomposition in Multilinear Algebra and Simultaneous Matrix Diagonalization," *SIAM J. Matrix Analysis and Applications,* vol. 28, no. 3, pp. 642-666, http://publi-etis.ensea.fr/2006/Del06, 2006.

[13] T.G. Kolda and B.W. Bader, "Tensor Decompositions and Applications," *SIAM Rev.,* vol. 51, no. 3, Sept. 2009.

[14] R.A. Harshman, "Foundations of the PARAFAC Procedure: Models and Conditions for an 'Explanatory'," *UCLA Working Papers in Phonetics,* vol. 16, pp. 1-84, 1970.

[15] R. Bro, "PARAFAC. Tutorial and Applications," *Chemometrics and Intelligent Laboratory Systems,* vol. 38, no. 2, pp. 149-171, Oct. 1997.

[16] A. Cichocki, R. Zdunek, A.H. Phan, and S.I. Amari, *Nonnegative Matrix and Tensor Factorizations.* Wiley, Nov. 2009.

[17] J.D. Carroll and J.-J. Chang, "Analysis of Individual Differences in Multidimensional Scaling via an n-Way Generalization of 'Eckart-Young' Decomposition," *Psychometrika,* vol. 35, no. 3, pp. 283-319, http://ideas.repec.org/a/spr/psycho/v35y1970i3p283-319.html, Sept. 1970.

[18] R.A. Harshman, "Determination and Proof of Minimum Uniqueness Conditions for PARAFAC1," UCLA Working Papers in Phonetics, vol. 22, pp. 111-117, 1972.

[19] R.A. Harshman, "PARAFAC2: Mathematical and Technical Notes," *UCLA Working Papers in Phonetics,* vol. 22, pp. 30-47, 1972.

[20] R. Zass and A. Shashua, "Doubly Stochastic Normalization for Spectral Clustering," *Advances in Neural Information Processing Systems,* B. Schölkopf, J. Platt, and T. Hoffman, eds., vol. 19, pp. 1569-1576, MIT Press, 2007.

[21] P.M. Kroonenberg and T.H.A. van der Voort, "Multiplicatieve Decompositie van Interacties bij Oordelen over de Werkelijkheidswaarde van Televisiefilms [Multiplicative Decomposition of Interactions for Judgements of Realism of Television Films]," *Kwantitatieve Methoden,* vol. 8, no. 23, pp. 117-144, 1987.

[22] J. Håstad, "Tensor Rank Is NP-Complete," *J. Algorithms,* vol. 11, no. 4, pp. 644-654, 1990.

[23] M.E. Timmerman and H.A.L. Kiers, "Three-Mode Principal Components Analysis: Choosing the Numbers of Components and Sensitivity to Local Optima," *British J. Math. and Statistical Psychology,* vol. 53, no. 1, pp. 1-16, 2000.

[24] R. Bro and H.A.L. Kiers, "A New Efficient Method for Determining the Number of Components in PARAFAC Models," *J. Chemometrics,* vol. 17, no. 5, pp. 274-286, 2003.

[25] H.A.L. Kiers and A. der Kinderen, "A Fast Method for Choosing the Numbers of Components in Tucker3 Analysis," *British J. Math. and Statistical Psychology,* vol. 56, no. 1, pp. 119-125, May 2003.

[26] E. Ceulemans and H.A.L. Kiers, "Selecting among Three-Mode Principal Component Models of Different Types and Complexities: A Numerical Convex Hull Based Method," *British J. Math. and Statistical Psychology,* vol. 59, no. 1, pp. 133-150, May 2006.

[27] J.P.C.L. da Costa, M. Haardt, F. Römer, and G. Del Galdo, "Enhanced Model Order Estimation Using Higher-Order Arrays," *Proc. 41st Asilomar Conf. Signals, Systems, and Computers,* pp. 412-416, Nov. 2007.

[28] J.P.C.L. da Costa, M. Haardt, and F. Römer, "Robust Methods Based on the HOSVD for Estimating the Model Order in PARAFAC Models," *Proc. Fifth IEEE Sensor Array and Multichannel Signal Processing Workshop,* pp. 510-514, July 2008.

[29] J.B. Kruskal, *Rank, Decomposition, and Uniqueness for 3-Way and N-Way Arrays.* North-Holland Publishing Co., 1989.

[30] P. Comon and J. ten Berge, "Generic and Typical Ranks of Three-Way Arrays," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing,* pp. 3313-3316, Apr. 2008.

[31] E. Kofidis and P.A. Regalia, "On the Best Rank-1 Approximation of Higher-Order Supersymmetric Tensors," *SIAM J. Matrix Analysis and Applications,* vol. 23, no. 3, pp. 863-884, 2002.

[32] T.P. Minka, "Automatic Choice of Dimensionality for PCA," *Advances in Neural Information Processing Systems,* T.K. Leen, T.G. Dietterich, and V. Tresp, eds, pp. 556-562, MIT Press, 2001.

[33] M.O. Ulfarsson and V. Solo, "Dimension Estimation in Noisy PCA with SURE and Random Matrix Theory," *IEEE Trans. Signal Processing,* vol. 56, no. 12, pp. 5804-5816, Dec. 2008.

[34] E. Radoi and A. Quinquis, "A New Method for Estimating the Number of Harmonic Components in Noise with Application in High Resolution Radar," *EURASIP J. Applied Signal Processing,* vol. 2004, no. 8, pp. 1177-1188, 2004.

[35] J. Grouffaud, P. Larzabal, and H. Clergeot, "Some Properties of Ordered Eigenvalues of a Wishart Matrix: Application in Detection Test and Model Order Selection," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing,,* vol. 5, pp. 2463-2466, 1996.

[36] A. Quinlan, J.-P. Barbot, P. Larzabal, and M. Haardt, "Model Order Selection for Short Data: An Exponential Fitting Test (EFT)," *EURASIP J. Advances in Signal Processing,* vol. 2007, pp. 1-11, 2007.

[37] A. Smilde, R. Bro, and P. Geladi, *Multi-Way Analysis: Applications in the Chemical Sciences.* John Wiley & Sons, Aug. 2005.

[38] G. Milligan and M. Cooper, "An Examination of Procedures for Determining the Number of Clusters in a Data Set," *Psychometrika,* vol. 50, no. 2, pp. 159-179, June 1985.

[39] W.J. Krzanowski and Y.T. Lai, "A Criterion for Determining the Number of Groups in a Data Set Using Sum-of-Squares Clustering," *Biometrics,* vol. 44, no. 1, pp. 23-34, Mar. 1988.

[40] M.F. Antonio Cuevas and R. Fraiman, "Estimating the Number of Clusters," *The Canadian J. Statistics,* vol. 28, no. 2, pp. 367-382, June 2000.

[41] W. Feller, *An Introduction to Probability Theory and Its Applications,* second ed., vol. 2. John Wiley & Sons, Jan. 1991.

[42] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the Number of Clusters in a Data Set via the Gap Statistic," *J. Royal Statistics Soc. (Series B),* vol. 63, no. 2, pp. 411-423, 2001.

[43] M.A.F. Figueiredo and A.K. Jain, "Unsupervised Learning of Finite Mixture Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 3, pp. 381-396, Mar. 2002.

[44] C.A. Sugar and G.M. James, "Finding the Number of Clusters in a Dataset: An Information-Theoretic Approach," *J. Am. Statistical Assoc.,* vol. 98, no. 463, pp. 750-763, Sept. 2003.

[45] M. Yan and K. Ye, "Determining the Number of Clusters Using the Weighted Gap Statistic," *Biometrics,* vol. 63, no. 4, pp. 1031-1037, Apr. 2007.

[46] P. Guo, P. Chen, and M. Lyu, "Cluster Number Selection for a Small Set of Samples Using the Bayesian Ying-Yang Model," *IEEE Trans. Neural Networks,* vol. 13, no. 3, pp. 757-763, Apr. 2002.

[47] X. Hu and L. Xu, "Investigation on Several Model Selection Criteria for Determining the Number of Cluster," *Neural Information Processing—Letters and Rev.,* vol. 4, no. 1, pp. 1-10, 2004.

[48] I.O. Kyrgyzov, O.O. Kyrgyzov, H. Maître, and M. Campede, "Kernel MDL to Determine the Number of Clusters," *Lecture Notes in Computer Science,* vol. 4571, pp. 203-217, Springer, 2007.

[49] A. Dempster, N. Laird, and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc., Series B,* vol. 39, no. 1, pp. 1-38, 1977.

[50] H. Akaike, "A New Look at the Statistical Model Identification," *IEEE Trans. Automatic Control,* vol. 19, no. 6, pp. 716-723, Dec. 1974.

[51] H. Bozdogan, "Model Selection and Akaike's Information Criterion (AIC): The General Theory and its Analytical Extensions," *Psychometrika,* vol. 52, no. 3, pp. 345-370, Sept. 1987.

[52] J. Rissanen, "Modelling by the Shortest Data Description," *Automatica,* vol. 14, pp. 465-471, 1978.

[53] A. Barron, J. Rissanen, and B. Yu, "The Minimum Description Length Principle in Coding and Modeling," *IEEE Trans. Information Theory,* vol. 44, no. 6, pp. 2743-2760, Oct. 1998.

[54] S.L. Sclove, "Some Aspects of Model-Selection Criterion," *Proc. First US/Japan Conf. Frontiers of Statistical Modeling: An Informational Approach,* H. Bozdogan, ed., vol. 2, pp. 37-67, 1994.

[55] T. Calinski and J. Harabasz, "A Dendrite Method for Cluster Analysis," *Comm. Statistics,* vol. 3, pp. 1-27, 1974.

[56] J.A. Hartigan, *Clustering Algorithms.* John Wiley & Sons, Apr. 1975.

[57] C.A. Andersson and R. Bro, "The n-Way Toolbox for MATLAB," *Chemometrics and Intelligent Laboratory Systems,* vol. 52, no. 1, pp. 1-4, 2000.

[58] D.D. Lee and H.S. Seung, "Learning the Parts of Objects by Nonnegative Matrix Factorization," *Nature,* vol. 401, no. 6755, pp. 788-791, 1999.

[59] D.D. Lee and H.S. Seung, "Algorithms for Non-Negative Matrix Factorization," *Advances in Neural Information Processing Systems,* T.K. Leen, T.G. Dietterich, and V. Tresp, eds., pp. 556-562, MIT Press, 2001.

[60] A. Georghiades, P. Belhumeur, and D. Kriegman, "From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 6, pp. 643-660, July 2001.

[61] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman, "Clustering Appearances of Objects under Varying Illumination Conditions," *Proc. 2003 IEEE CS Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 11-18, 2003.

[62] E. Donchin, K.M. Spencer, and R. Wijesinghe, "The Mental Prosthesis: Assessing the Speed of a P300-Based Brain-Computer Interface," *IEEE Trans. Rehabilitation Eng.,* vol. 8, no. 2, pp. 174-179, June 2000.

[63] F. Piccione, F. Giorgi, P. Tonin, K. Priftis, S. Giove, S. Silvoni, G. Palmas, and F. Beverina, "P300-Based Brain Computer Interface: Reliability and Performance in Healthy and Paralysed Participants," *Clinical Neurophysiology,* vol. 117, no. 3, pp. 531-537, Mar. 2006.

Zhaoshui He received the BS degree in applied mathematics from Hunan Normal University, Changsha, China, in 2000, and the PhD degree in electronics and information engineering from South China University of Technology, Guangzhou, China, in 2005. His research interests include blind signal processing, sparse representation, model selection, and clustering and their applications.

Andrzej Cichocki received the MSc (with honors), PhD, and Habilitate doctorate (DrSc) degrees, all in electrical engineering, from Warsaw University of Technology, Poland. He is the coauthor of four international books and monographs (two of them translated into Chinese): *Nonnegative Matrix and Tensor Factorizations* (J. Wiley, September 2009), *Adaptive Blind Signal and Image Processing* (J. Wiley, 2002), *MOS Switched Capacitor and Continuous-Time Integrated Circuits and Systems* (Springer-Verlag, 1989), and *Neural Networks for Optimization and Signal Processing* (J. Wiley and Teubner Verlag, 1993/1994) and author or coauthor of more than 200 papers. He is the editor-in-chief of the *Journal of Computational Intelligence and Neuroscience*. Currently, he is the head of the Laboratory for Advanced Brain Signal Processing in the RIKEN Brain Science Institute, Japan. He is a senior member of the IEEE.

Shengli Xie received the MS degree in mathematics from Central China Normal University, Wuhan, China, in 1992 and the PhD degree in control theory and applications from South China University of Technology, Guangzhou, China, in 1997. He is presently a full professor with the South China University of Technology. His research interests are broadly in automatic control and signal processing and mainly focus on blind signal processing, image processing, etc. He has authored or coauthored two monographs, a dozen of patents, and more than 70 scientific papers in journals and conference proceedings. He is a senior member of the IEEE.

Kyuwan Choi received the BA degree in electronic engineering from Korea University, Korea, in 2001, the MA degree in computer science from KAIST, Korea, in 2003, and the PhD degree (with honors) on brain-machine interfaces from Tokyo Institute of Technology, Japan, in 2007. From 2007 to 2009, he was a research scientist at the RIKEN Brain Science Institute, Japan. At RIKEN, he developed a wheelchair system that can be controlled with electroencephalography signals for people totally paralyzed from the neck down. Currently, he is researching brain information extraction using the NIRS-EEG system at ATR Computational Neuroscience Laboratories, Japan. His current research interests include brain-machine interface, biosignal processing, and human-computer interaction.