Multi-way Multi-level Kernel Modeling for Neuroimaging Classification

Lifang He[1*], Chun-Ta Lu[2], Hao Ding[3], Shen Wang[2], Linlin Shen[1†], Philip S. Yu[2,4], Ann B. Ragin[5]
[1]Shenzhen University, Shenzhen, China
[2]University of Illinois at Chicago, Chicago, IL, USA
[3]Beijing University of Posts and Telecommunications, Beijing, China
[4]Tsinghua University, Beijing, China
[5]Northwestern University, Chicago, IL, USA
{lifanghescut, haoding.tourist}@gmail.com, llshen@szu.edu.cn
{clu29, swang224, psyu}@uic.edu, ann-ragin@northwestern.edu

Abstract

Owing to prominence as a diagnostic tool for probing the neural correlates of cognition, neuroimaging tensor data has been the focus of intense investigation. Although many supervised tensor learning approaches have been proposed, they either cannot capture the nonlinear relationships of tensor data or cannot preserve the complex multi-way structural information. In this paper, we propose a Multi-way Multi-level Kernel (MMK) model that can extract discriminative, nonlinear and structural preserving representations of tensor data. Specifically, we introduce a kernelized CP tensor factorization technique, which is equivalent to performing the low-rank tensor factorization in a possibly much higher dimensional space that is implicitly defined by the kernel function. We further employ a multi-way nonlinear feature mapping to derive the dual structural preserving kernels, which are used in conjunction with kernel machines (e.g., SVM). Extensive experiments on real-world neuroimages demonstrate that the proposed MMK method can effectively boost the classification performance on diverse brain disorders (i.e., Alzheimer's disease, ADHD, and HIV).

1. Introduction

In many neurological and neuropsychiatric disorders, brain involvement, including irreversible loss of brain tissue and deterioration in cognitive function [14], has deleterious consequences for judgment and function. For brain disease diagnosis and detection of early anomalies, recent years have witnessed an intensive development in noninvasive neuroimaging techniques, *e.g.*, functional Magnetic Resonance Imaging (fMRI) and structural Diffusion Tensor Imaging (DTI). A neruoimaging sample is naturally a third-order tensor consisting of 3D voxels, and each voxel contains an intensity value that is proportional to the strength of the signal emitted by the corresponding location in the brain volume [4]. Since the 3D images are often specified in high-dimensional space (exponential to the dimensionality of each mode in the tensor), traditional vector-based methods prone to overfitting, especially for small sample size problems [5, 35]. How to extract compact and discriminative representations from the original tensor data has drawn significant attention in the study of neuroimaging analysis.

In the literature, several supervised tensor learning algorithms have been proposed. Early approaches which can preserve tensor structures are based upon linear models [8, 9, 12, 37], while the neuroimaging data is usually not linearly separable. Recently, several tensor-based kernel methods have been developed [25, 28, 29, 36]. Most of them focus on learning kernel via vector/matrix unfolding along each mode of the tensor data. However, the multi-way structural information such as the spatial arrangement of voxels will be lost in the unfolding procedures.

In order to capture the underlying patterns in tensor data, tensor factorization methods have been widely used [4, 17, 19]. However, most of the existing works either focus on the unsupervised exploratory analysis or to deal with linear tensor-based models. Recently, [13] employed the CANDECOMP/PARAFAC (CP) factorization to foster the use of kernel methods for supervised tensor learning. Although the CP factorization provides a good approximation to the original tensor data, it only concerned with multilinear formulas. Thus, the nonlinear relationships in the tensor object can hardly be modeled.

In this paper, we propose a novel Multi-way Multi-level Kernel (MMK) model to learn discriminative, nonlinear and

*This work was done while the first author was at the University of Illinois at Chicago.
†Corresponding author.

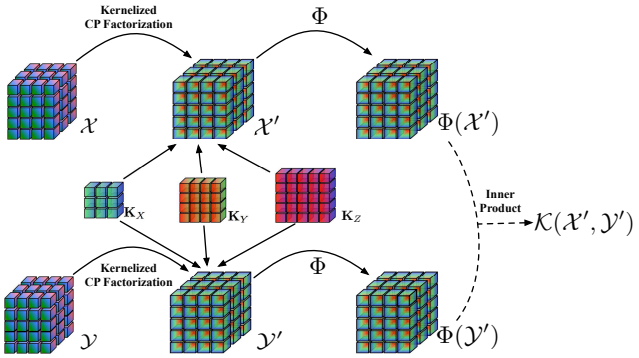


Figure 1. Overview of MMK. Given two input tensor data $\mathcal{X}$ and $\mathcal{Y}$, MMK first applies kernelized CP factorization to extract the nonlinear representations $\mathcal{X}'$ and $\mathcal{Y}'$, by sharing the kernel matrices of each mode, respectively, $\mathbf{K}_X$, $\mathbf{K}_Y$, and $\mathbf{K}_Z$ during the decomposition process. Then the extracted representations are embedded into the dual structural preserving kernel.

structural preserving representations of 3D neuroimaging data for brain disease classification. Figure 1 illustrates the proposed MMK model. Different from conventional methods, our approach is based upon kernelized tensor factorization that can fully capture the multi-way nonlinear structures of tensor data. Inspired by the Representer's Theorem [26], we present a new scheme of kernelized CP (KCP) factorization, which leverages the prior knowledge of feature correlation of each mode to extract a nonlinear representation from implicitly defined kernel space. Furthermore, we employ a multi-way nonlinear feature mapping to derive the dual structural preserving kernels in the tensor product feature space. The derived kernels are used in conjuction with kernel machines (*e.g.*, SVM) to solve the tensor data classification problems.

The main contributions of this work are summarized as follows:

- We propose KCP factorization as a new and systematic method for decomposing tensors of arbitrary order, which casts kernel learning methods into the framework of CP factorization. The proposed KCP can effectively capture the complex nonlinear relationships between tensor modes.
- We introduce MMK model that integrates KCP factorization and tensor kernel methods for practical tensor data classification.
- Extensive empirical studies on neurological disorder prediction for three different diseases (Alzheimer's disease, ADHD and brain damage by HIV) demonstrate the proposed approach significantly outperforms other related state-of-the-art classification methods.

The rest of the paper is organized as follows. We briefly review on related works of kernel learning and tensor factorization in Section 2. We introduce the preliminary concepts in Section 3. Then we give the problem formulation and present the proposed MMK approach in section 4. In Section 5, we report the experiment results. Finally, we conclude the paper in Section 6.

2. Related Work

Tensor factorization is a higher-order extension of matrix factorization that elicit intrinsic multi-way structure and capture the underlying patterns in tensor data. There are many excellent works for tensor factorization [3, 4, 17, 38] The two most commonly used factorization techniques are CP and Tucker [17, 27, 32]. Compared to Tucker, CP is more frequently used due to its properties of uniqueness and simplicity [12, 15, 31]. A comprehensive survey on tensor factorization can be found in [17]. However, the existing works are mainly based on multilinear factorization schemes, and are difficult to model nonlinear relationships and to discover complex patterns in data.

Supervised tensor learning has been extensively studied in recent years. For example, [31] proposed a supervised tensor learning framework, which extends the standard linear SVM learning framework to tensor patterns by constructing multilinear models. Under this learning framework, several linear tensor-based models [2, 9, 12, 18, 37] are developed, whereas the problem of how to build nonlinear models directly on tensor data has not been well studied. In order to apply kernel modeling for tensor data, several works [25, 28, 29, 36] have been presented to convert the tensors into vectors (or matrices), which are then used to construct kernels. However, this kind of conversion will result in the following two problems: 1) break the natural multi-way structure and correlation in the original data [7, 22]; 2) lead to the curse of dimensionality and small sample size problems [11, 35]. The problem of how to build kernel modeling directly on tensor data has not been well studied. Most recent attempt in this direction is related to CP factorization proposed in [13], while it has the same drawback as CP factorization.

3. Preliminaries

In this section, we briefly introduce some preliminary knowledge from tensor algebra. For a deeper introduction to the concepts and terminology, we refer to [17]. Table 1 summarizes the important symbols for handy reference.

3.1. Notation and Basic Operations

To facilitate the distinction, scalars are denoted by lowercase letters $(a, b, \cdots; \alpha, \beta, \cdots)$, vectors by bold lowercase letters $(\mathbf{a}, \mathbf{b}, \cdots)$, matrices by bold uppercase letters

Table 1. Important Notations

Symbol	Definition and Description
a or α	lowercase letter represents scale
$\mathbf{a}$	boldface lowercase letter represents vector
$\mathbf{A}$	boldface uppercase letter represents matrix
$\mathcal{A}$	calligraphic letter represents tensor
$\otimes$	denotes the outer product
$\odot$	denotes the Khatri-Rao product
$\langle\cdot,\cdot\rangle$	denotes the inner product
$[\![\cdot]\!]$	denotes the CP factorization
$\Phi(\cdot)$	denotes the feature mapping
$\kappa(\cdot,\cdot)$	denotes the kernel function

$(\mathbf{A}, \mathbf{B}, \cdots)$, and tensors by calligraphic letters $(\mathcal{A}, \mathcal{B}, \cdots)$. The order of a tensor is the number of dimensions, also known as modes or ways. An N-th order tensor is represented as $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, where I_n is the cardinality of its n-th mode, $n \in \{1, 2, \cdots, N\}$. An element of a vector $\mathbf{a}$, a matrix $\mathbf{A}$, or a tensor $\mathcal{A}$ is denoted by a_i, $a_{i,j}$, $a_{i,j,k}$, etc., depending on the number of modes. The *outer product*, *Khatri-Rao product* and *inner product* are denoted by $\otimes$, $\odot$ and $\langle\cdot,\cdot\rangle$, respectively.

Given two tensors $\mathcal{X} = \mathbf{x}^{(1)} \otimes \cdots \otimes \mathbf{x}^{(N)}$ and $\mathcal{Y} = \mathbf{y}^{(1)} \otimes \cdots \otimes \mathbf{y}^{(N)}$, it holds that

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \prod_{i=1}^{N} \langle \mathbf{x}^{(i)}, \mathbf{y}^{(i)} \rangle \tag{1}$$

Matricization or *unfolding* is the process to transform a tensor into a matrix such that all of the columns along a certain mode are rearranged to form a matrix [16]. The mode-n matricization of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is denoted by $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times \Pi_{i \neq n}^{N} I_i}$, which can be obtained by permuting the dimensions of $\mathcal{X}$ as $[I_n, I_1, \cdots, I_{n-1}, I_{n+1}, \cdots, I_N]$ and then reshaping the permuted tensor into a matrix of size $I_n \times \Pi_{i \neq n}^{N} I_i$ [38].

3.2. CP Factorization

CANDECOMP/PARAFAC (CP) factorization is a widely used technique for exploring and extracting the underlying structure of the multi-way data, which is critical to the development of our proposed method. Basically, given an N-th order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, CP factorization approximates this tensor by N *loading* matrices $\mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(N)}$, such that

$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \otimes \cdots \otimes \mathbf{a}_r^{(N)} = [\![\mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(N)}]\!] \tag{2}$$

where $[\![\cdot]\!]$ is defined as the CP factorization operator for shorthand and each loading matrix $\mathbf{A}^{(n)} = [\mathbf{a}_1^{(n)}, \cdots, \mathbf{a}_R^{(n)}]$, $n \in \{1, 2, \cdots, N\}$ is of size $I_n \times R$, and R is referred to as the *rank* of the tensor $\mathcal{X}$ [6], indicating the number of factors.

To obtain the CP factorization $[\![\mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(N)}]\!]$, the objective is to minimize the following estimation error:

$$\mathcal{L} = \min_{\mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(N)}} \|\mathcal{X} - [\![\mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(N)}]\!]\|_F^2 \tag{3}$$

However, $\mathcal{L}$ is not jointly convex w.r.t. $\mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(N)}$. A widely used optimization technique is the Alternating Least Squares (ALS) algorithm, which alternatively minimize $\mathcal{L}$ for each variable while fixing the other, that is,

$$\mathbf{A}^{(n)} \leftarrow \arg\min_{\mathbf{A}^{(n)}} \|\mathbf{X}_{(n)} - \mathbf{A}^{(n)} (\odot_{i \neq n}^{N} \mathbf{A}^{(i)})^{\mathrm{T}}\|_F^2 \tag{4}$$

where $\odot_{i \neq n}^{N} \mathbf{A}^{(i)} = \mathbf{A}^{(N)} \odot \cdots \mathbf{A}^{(n-1)} \odot \mathbf{A}^{(n+1)} \cdots \odot \mathbf{A}^{(1)}$.

4. Methodology

In a typical neuroimaging classification task, we are given a collection of M training examples $\{\mathcal{X}_i, y_i\}_{i=1}^{M} \subset X \times Y$, where $\mathcal{X}_i \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is the input of the i-th neuroimaging sample and y_i is the corresponding class label. The task of neuroimaging classification is to find a function $f : X \to Y$ that correctly predicts the label of an unseen neuroimaging sample $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. In the kernel learning scenario, this problem can be formulated as the following optimization problem:

$$f^* = \arg\min_{f \in H} \left(\frac{C}{M} \sum_{i=1}^{M} V\left(f(\mathcal{X}_i), y_i\right) + \|f\|_H^2 \right) \tag{5}$$

where C is used to control the trade-off between the empirical risk and the regularization term, H is a set of functions forming a Hilbert space (the hypothesis space), and $V(\cdot)$ is a prescribed loss function that measures how well f fits the data. By using the *Representer Theorem* [26], we classically obtain:

$$f^*(\mathcal{X}) = \sum_{i=1}^{M} c_i \kappa(\mathcal{X}_i, \mathcal{X}) \tag{6}$$

where c_i is the coefficient and $\kappa(\cdot,\cdot)$ is a positive definite (reproducing) kernel function, defined by $\kappa : X \times X \to \mathbb{R}$ with $\kappa(\mathcal{X}_i, \mathcal{X}_j) = \langle \Phi(\mathcal{X}_i), \Phi(\mathcal{X}_j) \rangle$, where $\Phi : X \to H$ is a feature mapping function.

Notice that the kernel function becomes the only domain specific module of the learning system. In this context, it is therefore essential to design a kernel that adequately encapsulates all information necessary for prediction. In the following, we first propose a kernelized CP factorization model to learn the latent structural features. We then show how to design a good kernel by leveraging the extracted latent structural features.

4.1. Kernelized CP Factorization

Although tensor provides a natural representation for multi-way data, there is no guarantee that it will be effective for kernel learning. Learning will only be successful

if the regularities that underlie the data can be discerned by the kernel function. From the previous analysis of the multi-way data, we noted that the essential information contained in the tensor is embedded in its multi-way structure. Therefore, an important aspect of tensor based kernel learning is to represent tensor by key structural features easier to manipulate, and design kernels on such features.

According to the principle of CP factorization, it is designed to conserve the original multi-way structure of the tensor object and provide more compact and meaningful representations. Nevertheless, it only provides a good approximation – rather than the most relevant-non-redundant (*i.e.*, representative, discriminative and non-redundant) features. Besides, the standard CP factorization is only concerned with multilinear formulas, and thus it is difficult to capture the nonlinear relationships in the tensor object. To leverage the success of kernel learning, we tailor a kernelized CP (KCP) factorization, on making use of tensor characteristics and mode-specific knowledge.

For simplicity and without loss of generality, we consider a third-order tensor $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$. Inspired by [1], it is instructive to look at the three-way structure of $\mathcal{X}$ as a function of three variables x, y, z, living in measurable spaces X, Y, Z, respectively. To obtain a kernel version of a given CP factorization $\mathcal{X} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$, where the tensor element $x_{i,j,k} = \sum_{r=1}^{R} a_{i,r}b_{j,r}c_{k,r}$, we define low-rank functions g belonging to the following family

$$\begin{aligned}\mathcal{G}_R := \{g : X \times Y \times Z \to\ & \mathbb{R} \\ g(x,y,z)\ \mapsto\ & \sum_{r=1}^{R} a_r(x)b_r(y)c_r(z) \\ s.t.\ a_r(x) \in H_X,\ & b_r(y) \in H_Y,\ c_r(z) \in H_Z\} \quad (7)\end{aligned}$$

where H_X, H_Y and H_Z are Hilbert spaces constructed from specified kernels $\kappa_X(\cdot,\cdot)$, $\kappa_Y(\cdot,\cdot)$ and $\kappa_Z(\cdot,\cdot)$, in X, Y and Z, respectively. Here x, y, z can be seen as indices of mode.

By recursively using the *Representer Theorem*, we have

$$\begin{aligned}a_r(x) &= \textstyle\sum_{i=1}^{I} \alpha_{i,r}\kappa_X(x_i, x), \\ b_r(y) &= \textstyle\sum_{j=1}^{J} \beta_{j,r}\kappa_Y(y_j, y), \\ c_r(z) &= \textstyle\sum_{k=1}^{K} \gamma_{k,r}\kappa_Z(z_k, z).\end{aligned}$$

Defining vectors $\boldsymbol{\kappa}_X^{\mathrm{T}}(x) := [\kappa_X(x_1, x), \cdots, \kappa_X(x_I, x)]$, $\boldsymbol{\kappa}_Y^{\mathrm{T}}(y) := [\kappa_Y(y_1, y), \cdots, \kappa_Y(y_J, y)]$, and $\boldsymbol{\kappa}_Z^{\mathrm{T}}(z) := [\kappa_Z(z_1, z), \cdots, \kappa_Z(z_K, z)]$, along with matrices $\mathbf{A} \in \mathbb{R}^{I\times R} : a_{i,r} := \alpha_{i,r}$, $\mathbf{B} \in \mathbb{R}^{J\times R} : b_{j,r} := \beta_{j,r}$, and $\mathbf{C} \in \mathbb{R}^{K\times R} : c_{k,r} := \gamma_{k,r}$, it follows that

$$\begin{aligned}g(x,y,z) &= \sum_{r=1}^{R} a_r(x)b_r(y)c_r(z) \\ &= \sum_{r=1}^{R} (\boldsymbol{\kappa}_X^{\mathrm{T}}(x)\mathbf{a}_r)(\boldsymbol{\kappa}_Y^{\mathrm{T}}(y)\mathbf{b}_r)(\boldsymbol{\kappa}_Z^{\mathrm{T}}(z)\mathbf{c}_r) \quad (8)\end{aligned}$$

Algorithm 1 Kernelized CP (KCP) Factorization

Input: Tensor object $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$ and rank R
Output: $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$
1: Allocate $\mathbf{K}_X, \mathbf{K}_Y, \mathbf{K}_Z$
2: Calculate matrices $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$ by using the standard CP factorization
3: Set $\mathbf{A} \leftarrow \mathbf{K}_X^{-1}\mathbf{A}$
4: Set $\mathbf{B} \leftarrow \mathbf{K}_Y^{-1}\mathbf{B}$
5: Set $\mathbf{C} \leftarrow \mathbf{K}_Z^{-1}\mathbf{C}$

Using Eq. (8), the following fitting criterion holds:

$$\begin{aligned}\hat{g} :&= \arg\min_{g\in\mathcal{G}_R} \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} (x_{i,j,k} - g(x_i, y_j, z_k))^2 \\ &= \min_{\mathbf{A},\mathbf{B},\mathbf{C}} \|\mathcal{X} - [\![\mathbf{K}_X\mathbf{A}, \mathbf{K}_Y\mathbf{B}, \mathbf{K}_Z\mathbf{C}]\!]\|_F^2 \quad (9)\end{aligned}$$

where kernel matrices $\mathbf{K}_X = [\boldsymbol{\kappa}_X(x_1), \cdots, \boldsymbol{\kappa}_X(x_I)] \in \mathbb{R}^{I\times I}$, $\mathbf{K}_Y = [\boldsymbol{\kappa}_Y(y_1), \cdots, \boldsymbol{\kappa}_Y(y_J)] \in \mathbb{R}^{J\times J}$, and $\mathbf{K}_Z = [\boldsymbol{\kappa}_Z(z_1), \cdots, \boldsymbol{\kappa}_Z(z_K)] \in \mathbb{R}^{K\times K}$.

Eq. (9) reduces to the standard CP factorization when the side information is discarded by selecting $\kappa_X(\cdot,\cdot)$, $\kappa_Y(\cdot,\cdot)$ and $\kappa_Z(\cdot,\cdot)$ as Kronecker delta functions (*i.e.*, the corresponding kernel matrix is the identity matrix). Essentially, (9) yields the sought nonlinear low-rank approximation method for $g(x, y, z)$ in Eq. (8). We refer to Eq. (9) as KCP factorization, and matrices $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$ as latent factor matrices to distinguish from loading matrices of CP method. Note that we do not kernelize the tensor data $\mathcal{X}$, and we kernelize the memberships in the latent factors. Moreover, it is easy to extend this result to the higher-order case.

4.2. Optimization for KCP

Basically, Eq. (9) has the same structure as the CP factorization in Eq. (3), so we have a similar update rule for solving the KCP factorization. Here, we state the following theorem to efficiently solve Eq. (9). The overall algorithm is summarized in Algorithm 1.

Theorem 1. *Let $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ be an arbitrary tensor and assume we have a CP factorization of $\mathcal{X}$, $\mathcal{X} = [\![\mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(N)}]\!]$ such that Eq. (3) holds. Then the solution of the following problem*

$$\min_{\hat{\mathbf{A}}^{(1)},\cdots,\hat{\mathbf{A}}^{(N)}} \|\mathcal{X} - [\![\mathbf{K}_1\hat{\mathbf{A}}^{(1)}, \cdots, \mathbf{K}_N\hat{\mathbf{A}}^{(N)}]\!]\|_F^2 \quad (10)$$

is

$$\hat{\mathbf{A}}^{(n)} = \mathbf{K}_n^{-1}\mathbf{A}^{(n)} \quad (11)$$

where $n \in \{1, 2, \cdots, N\}$, $\mathbf{K}_n \in \mathbb{R}^{I_n\times I_n}$ is known positive definite matrix.

Proof. Let $\tilde{\mathbf{A}}^{(n)} = \mathbf{K}_n\hat{\mathbf{A}}^{(n)}$ for $n \in \{1, 2, \cdots, N\}$. When applying the ALS approach to solve Eq. (10), we have

$$\tilde{\mathbf{A}}^{(n)} \leftarrow \arg\min_{\tilde{\mathbf{A}}^{(n)}} \|\mathbf{X}_{(n)} - \tilde{\mathbf{A}}^{(n)}(\odot_{i\neq n}^{N}\tilde{\mathbf{A}}^{(i)})^{\mathrm{T}}\|_F^2 \quad (12)$$

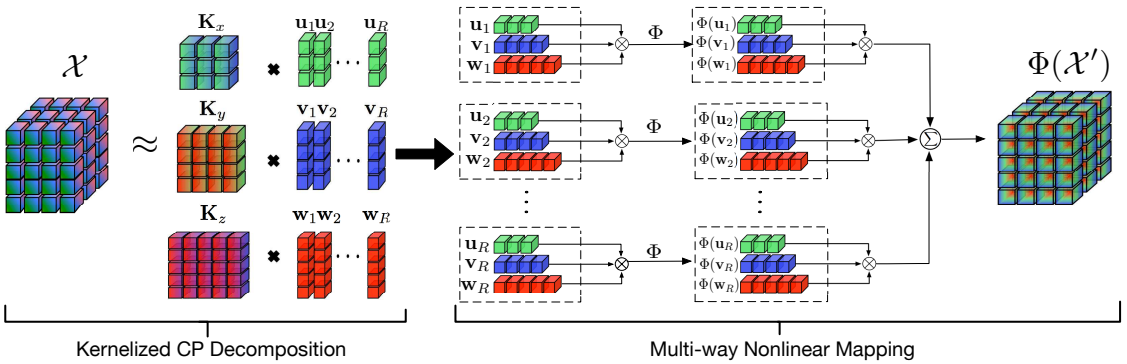


Figure 2. Multi-way Multi-level Kernel Modeling.

where $\odot_{i\neq n}^{N}\tilde{\mathbf{A}}^{(i)} = \tilde{\mathbf{A}}^{(N)}\odot\cdots\tilde{\mathbf{A}}^{(n-1)}\odot\tilde{\mathbf{A}}^{(n+1)}\cdots\odot\tilde{\mathbf{A}}^{(1)}$.

Since the objective function in (12) is strictly convex, a unique solution exists for this problem. That is, $\mathbf{A}^{(n)}$ is also the solution of Eq. (12) and $\tilde{\mathbf{A}}^{(n)} \triangleq \mathbf{A}^{(n)}$. Moreover, since $\mathbf{K}_n$ is a positive-definite matrix, its inverse exists. Thus we arrive at Theorem 1. □

According to Theorem 1, it turns out that the minimization of Eq. (9) can be solved by first computing the CP factorization of $\mathcal{X}$ and then obtaining the solution of latent factor matrices on the loading matrices. There are many efficient algorithms available for solving CP factorization [17], which facilitate the implementation of KCP factorization. We use the method (and code) of [24] to compute the CP factorization, which is based on the ALS algorithm, coupled with the line search scheme to speed up convergence.

4.3. Multi-way Nonlinear Mapping

To incorporate prior known similarities across different data samples, we use the KCP factorization to extract the compact representations for each input tensor object, by sharing the kernel matrices of each mode. In this way, the discriminative latent factor matrices, which is beneficial to the classification task, is constructed. In the following, we illustrate how to use these extracted latent structural features to evolve a suitable kernel for neuroimaging classification.

Let the KCP factorization of $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I\times J\times K}$ be $\mathcal{X} = [\![\mathbf{K}_X\mathbf{A}, \mathbf{K}_Y\mathbf{B}, \mathbf{K}_Z\mathbf{C}]\!]$ and $\mathcal{Y} = [\![\mathbf{K}_X\mathbf{U}, \mathbf{K}_Y\mathbf{V}, \mathbf{K}_Z\mathbf{W}]\!]$, respectively. The latent factor matrices are of multi-mode and each factor matrix is associated with a mode. It is straightforward to express the latent factor matrices in a tensor fashion by means of the outer product operator. In this manner we will bring the latent factor matrices of different modes into one tensor, not only to bring their own interpretations, but also to include multi-way capabilities. We define the *latent* tensors for $\mathcal{X}$ and $\mathcal{Y}$ as $\mathcal{X}' = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ and $\mathcal{Y}' = [\![\mathbf{U}, \mathbf{V}, \mathbf{W}]\!]$, which have the same sizes as $\mathcal{X}$ and $\mathcal{Y}$. Inspired by the success of dual structure-preserving kernel (DuSK) method [13], we assume the latent tensors are mapped into a Hilbert space H by

$$\Phi : X \times Y \times Z \to \mathbb{R}^{H_X\times H_Y\times H_Z} \tag{13}$$

where $\mathbb{R}^{H_X\times H_Y\times H_Z}$ is a third-order tensor Hilbert space.

Based on the derivatives of the kernel function, it is important to note that the Hilbert space is a high-dimensional space of the original feature space, equipped with the same operations. Consequently, we can directly factorize tensor data in the Hilbert space the same as original feature space. This is equivalent to considering the following multi-way nonlinear mapping:

$$\Phi : \sum_{r=1}^{R}\mathbf{a}_r\otimes\mathbf{b}_r\otimes\mathbf{c}_r \mapsto \sum_{r=1}^{R}\Phi(\mathbf{a}_r)\otimes\Phi(\mathbf{b}_r)\otimes\Phi(\mathbf{c}_r) \tag{14}$$

In this respect, it corresponds to mapping the latent tensors into high-dimensional tensors that retain the multi-way structure. More generally, it can be seen as mapping the tensor data into tensor Hilbert space and then performing the CP factorization in the Hilbert space.

After mapping the latent tensor into the Hilbert space, which is also essentially the tensor product space, the kernel is just the standard inner product of tensors on that space. Hence, we can derive a dual structural preserving kernel function as follows:

$$\begin{aligned}
&\kappa(\mathcal{X}', \mathcal{Y}') = \kappa(\sum_{r=1}^{R}\mathbf{a}_r\otimes\mathbf{b}_r\otimes\mathbf{c}_r\ ,\ \sum_{r=1}^{R}\mathbf{u}_r\otimes\mathbf{v}_r\otimes\mathbf{w}_r)\\
&= \langle\Phi(\sum_{r=1}^{R}\mathbf{a}_r\otimes\mathbf{b}_r\otimes\mathbf{c}_r), \Phi(\sum_{r=1}^{R}\mathbf{u}_r\otimes\mathbf{v}_r\otimes\mathbf{w}_r)\rangle\\
&= \langle\sum_{r=1}^{R}\Phi(\mathbf{a}_r)\otimes\Phi(\mathbf{b}_r)\otimes\Phi(\mathbf{c}_r), \sum_{r=1}^{R}\Phi(\mathbf{u}_r)\otimes\Phi(\mathbf{v}_r)\otimes\Phi(\mathbf{w}_r)\rangle\\
&= \sum_{i=1}^{R}\sum_{j=1}^{R}\kappa(\mathbf{a}_i, \mathbf{u}_j)\kappa(\mathbf{b}_i, \mathbf{v}_j)\kappa(\mathbf{c}_i, \mathbf{w}_j)
\end{aligned} \tag{15}$$

By virtue of its derivation, we see that such a kernel function can take the multi-way structure within tensor flexibility into consideration. In general, this kernel is an extension

of the conventional kernels in the vector space to the tensor space, and each vector kernel function can be applied in this framework for tensor classification in conjunction with kernel machines (*e.g.*, SVM). Different kernel functions specify different hypothesis spaces or even different knowledge embeddings of the data and thus can be viewed as capturing different notions of correlations. In particular, it can be regarded as computing the inner product after l successive applications of the nonlinear mapping $\Phi(\cdot)$:

$$\kappa^l(\mathbf{x},\mathbf{y}) = \langle \underbrace{\Phi(\Phi(\cdots\Phi(\mathbf{x})))}_{l \text{ times}}, \underbrace{\Phi(\Phi(\cdots\Phi(\mathbf{y})))}_{l \text{ times}} \rangle \quad (16)$$

Intuitively, if the base kernel function $\kappa(\mathbf{x},\mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle$ mimics the computation in a single-level network, then the iterated mapping in Eq. (16) should mimic the computation in a multi-level network.

Finally, by putting everything together, we obtain the general version of our multi-way multi-level kernel (MMK) modeling, as illustrated in Figure 1 and Figure 2. Compared to DuSK method [13], which models the relationships between tensor samples by using a single-level kernel on the low-rank approximation (*i.e.*, loading matrices), the proposed method models the nonlinear and structural information not only between tensor samples but also within the tensor sample itself, by using a multi-way multi-level kernel on the latent structural features (*i.e.*, latent factor matrices).

5. Experiments and Results

In order to empirically evaluate the effectiveness of the proposed MMK approach in addressing the neuroimaging classification, we conduct extensive experiments on three real-life neuroimaging (fMRI) datasets and compare with eight existing state-of-the-art methods. In the following, we introduce the datasets used in our analysis and describe the experimental settings. Then we present the experimental results as well as the analysis.

5.1. Data Collection and Preprocessing

We consider three resting-state fMRI datasets as follows:

- *Alzheimer's Disease (ADNI)*: This dataset is collected from the Alzheimer's Disease Neuroimaging Initiative[1], which consists of records of patients with Mild Cognitive Impairment (MCI) and Alzheimer's Disease (AD). We downloaded all 33 records of resting-state fMRI images and applied SPM8 [2] to preprocess data. For each individual, the first ten volumes were removed, functional images were realigned to the first volume, slice timing corrected, and spatially smoothed with an 8-mm FWHM Gaussian kernel and normalized to the MNI template. We then adopted REST[3] to perform temporally band-pass filtering ($0.01 - 0.08$ Hz) and remove the linear trend of time series. After this, we averaged each individual over time domain, resulting in 33 samples of size $61 \times 73 \times 61$. We treat AD+MCI as the negative class, and the normal brains as positive class. Finally, followed by [13], we scaled each individual to $[0, 1]$. Note that normalization is of extreme importance for group analyses, since the brain of every individual is different.

- *Human Immunodeficiency Virus Infection (HIV)*: This dataset is collected from Chicago Early HIV Infection Study in Northwestern University [33], which contains 83 fMRI brain images of patients with early HIV infection (negative) and normal controls (positive). We used the same preprocessing steps as in ADNI dataset, resulting in 83 samples of size $61 \times 73 \times 61$.

- *Attention Deficit Hyperactivity Disorder (ADHD)*: This dataset is collected from ADHD-200 global competition dataset[4], which contains the resting-state fMRI images of 200 subjects, either ADHD patients (negative) or normal controls (positive). We averaged each individual over time domain, resulting in 200 samples of size $58 \times 49 \times 47$.

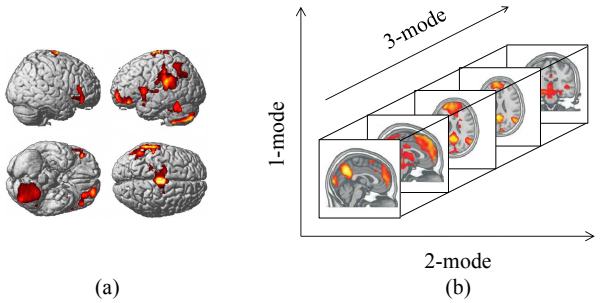


Figure 3. (a) Visualization of an fMRI image from four angles, (b) An illustration of third-order tensor of an fMRI image.

5.2. Baselines and Metrics

We consider Gaussian RBF kernel and SVM classifier as the constituents of our MMK method for comparison, and use the following eight methods as baselines.

- **SVM**: First we implemented a naive baseline, Gaussian-RBF kernel-based SVM, which is the most widely used vector-based method for classification. In the following methods, we use SVM with Gaussian RBF kernel as the classifier, if not stated explicitly.

[1] http://adni.loni.usc.edu/
[2] http://www.l.ion.ucl.ac.uk/spm/software/spm8/
[3] http://resting-fmri.sourceforge.net
[4] http://neurobureau.projects.nitrc.org/ADHD200/

Table 2. Summary of compared methods. C is the trade-off parameter, σ is the kernel width parameter, R is the rank of tensor factorization.

Method	SVM/SVM+PCA	K_{3rd} [25]	sKL [36]	FK [28]	DuSK [13]	STTK [23]	3D CNN [10]	MMK
Data Post-processing	Vectors	Vectors	Matrices	Matrices	3D Tensor	4D Tensor	3D Tensor	3D Tensor
Correlation Exploited	One-way	One-way	One-way	One-way	Multi-way	Multi-way	Multi-way	Multi-way
Kernel Explored	Single-level	Single-level	Single-level	Single-level	Single-level	Single-level	Multi-level	Multi-level
Parameters	C, σ	C, σ	C, σ	C, σ	C, σ, R	C, σ, R, ER_t	Many*	C, σ, R

- **SVM+PCA**: We also implemented a vector-based subspace learning algorithm, which first uses principal component analysis (PCA) to reduce the input dimension and then feeds into SVM model. This method is commonly used to deal with high-dimensional classification, in particular fMRI classificatioin [30, 34].

- $\mathbf{K}_{3rd}$: It is a vector unfolding based tensor kernel method, which aims at exploiting the input tensor along each mode to capture structural information and has been used to analyze fMRI data together with Gaussian RBF kernel [25].

- **sKL**: It is a matrix unfolding based tensor kernel method that defined based on the symmetric Kullback-Leibler divergence, and has been used to reconstruct 3D movement [36].

- **FK**: It is also a matrix unfolding based tensor kernel method, but defined based on multilinear singular value decomposition (MLSVD). The constituent kernels are from the class of Gaussian RBF kernels [28].

- **DuSK**: It is the most recent tensor kernel method based upon CP factorization, which has been used to analyze fMRI data together with Gaussian RBF kernel [13].

- **STTK**: It is a variant of the DuSK method for whole-brain fMRI classification, which views the fMRI data as 4D spatio-temporal objects observed under different conditions/sessions [23].

- **3D CNN**: It is a 3D convolutional neural network extended from 2D version [10], which uses the convolution kernel. The convolution kernel is the cubic filters learned from data, which has a small receptive field, but extends through the full depth of the input volume.

Table 2 summarizes the compared methods. We performed 5-fold cross-validation and used the classification accuracy as the evaluation measure. This process was repeated 50 times for all methods and the average reported as the result. We use LibSVM as a SVM tool. The optimal parameters for all methods were determined by grid search. The optimal trade-off parameter is selected from $C \in \{2^{-8}, 2^{-7}, \cdots, 2^{8}\}$, the kernel width parameter is selected from $\sigma \in \{2^{-8}, 2^{-7}, \cdots, 2^{8}\}$, the optimal rank R is selected from $\{1, 2, \cdots, 10\}$. The parameter ER_t for STTK is set by default to 0.2. The optimal parameter for 3D CNN, *i.e.*, receptive field (R), zero-padding (P), the input volume dimensions (Width $\times$ Height $\times$ Depth, or $W \times H \times D$) and stride length (S) are tuned following [10]. Covariance kernel matrices $\mathbf{K}_X, \mathbf{K}_Y, \mathbf{K}_Z$ are assumed known.

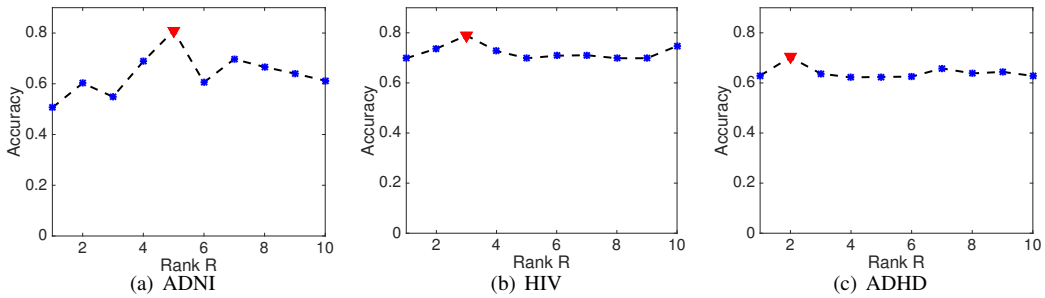
Table 3. Classification accuracy comparison (mean $\pm$ standard deviation)

	ADNI	HIV	ADHD
SVM	0.49 ± 0.02	0.70 ± 0.01	0.58 ± 0.00
SVM+PCA	0.50 ± 0.02	0.73 ± 0.03	0.63 ± 0.01
K_{3rd}	0.55 ± 0.01	0.75 ± 0.02	0.55 ± 0.00
sKL	0.51 ± 0.03	0.65 ± 0.02	0.50 ± 0.04
FK	0.51 ± 0.02	0.70 ± 0.01	0.50 ± 0.00
DuSK	0.75 ± 0.02	0.74 ± 0.00	0.65 ± 0.01
STTK	0.76 ± 0.02	0.76 ± 0.01	0.68 ± 0.01
3DCNN	0.52 ± 0.03	0.75 ± 0.02	0.68 ± 0.02
MMK	$\mathbf{0.81 \pm 0.01}$	$\mathbf{0.79 \pm 0.01}$	$\mathbf{0.70 \pm 0.01}$

5.3. Classification Performance

Table 3 shows the average classification accuracy and standard deviation of different methods on three datasets, where the best result is highlighted in bold type. From comparison results, we have the following observations.

First, the classification accuracy of each method on different dataset can be quite different. However, the proposed MMK method consistently outperforms all the other methods on all three datasets. This is mainly because MMK can learn the nonlinear latent subspaces embedded within the tensor together with considering a prior knowledge across different data samples, while the other methods fail to explore the nonlinear relationships in the tensor object. Moreover, it can be found that MMK significantly outperforms other methods on the ADNI dataset. The reason behind is that this data is extremely high dimensional but with small sample size. In neuroimaging task it is very hard for classification algorithms to achieve even moderate classification accuracy on ADNI dataset. In particular, as can be seen from the results, 3D CNN achieves a relatively lower accuracy on ADNI dataset, which is because the deep learning method needs a large number of training data to train the deep neural network. Medical neuroimaging datasets do not have enough training data because of privacy. Nevertheless, by making use of tensor properties, we are able to boost the classification performance. Further, it can be



(a) ADNI (b) HIV (c) ADHD

Figure 4. Test accuracy vs. R on three neuroimaging datasets.

found that MMK always performs better than DuSK, which empirically shows the effectiveness of feature extraction in high-dimensional tensor data rather than approximation.

Based on these results, we can conclude that unfolding tensor into vectors or matrices would lose the multi-way structural information within tensor data, leading to the degraded performance. While operation on tensors is much more effective than on vectors and matrices for high-dimensional tensor data analysis. In general, the experimental results demonstrate the effectiveness and considerable advantages of our proposed methods in the fMRI data classification study.

5.4. Parameter Sensitivity

Although the optimal values of the parameters in our proposed MMK are found by grid search, it is still important to see the sensitivity of MMK to the rank of KCP factorization R. To this end, in this section we demonstrate a sensitivity study over different $R \in \{1, 2, \cdots, 10\}$, where the optimal kernel width parameter and trade-off parameter are still selected from $C \in \{2^{-8}, 2^{-7}, \cdots, 2^{8}\}$ and $\sigma \in \{2^{-8}, 2^{-7}, \cdots, 2^{8}\}$ respectively. It is obvious that the efficiency of MMK is reduced when R is increased because a higher value of R implies that more items are included into kernel computations. Thus, we only demonstrate the variation in test accuracy over different R on three datasets. As shown in Figure 4, the rank parameter R has a significant effect on the test accuracy and the optimal value of R depends on the data. Despite that, the optimal value of R in general lies in the range $2 \leq R \leq 5$, which may provide a good guidance for selection of the R in advance.

In summary, the parameter sensitivity study indicates that the classification performance of MMK relies on parameter R and it is difficult to specify an optimal value for R in advance. However, in most cases the optimal value of R lies in a small range of values as demonstrated in [12] and it is not time-consuming to find it using the grid search strategy in practical applications.

6. Conclusion and Outlook

In this paper, we have introduced a multi-way multi-level kernel (MMK) method, with an application to neuroimaging classification. Different from conventional kernel methods, our approach is based on kernelized CP (KCP) factorization that casts kernel learning methods into the framework of CP factorization, such that the complex nonlinear relationships between tensor modes can be captured. The nonlinear representations extracted from the KCP are embedded in the dual structural preserving kernels, which are used in conjunction with SVM to solve the neuroimaging classification problems. Extensive empirical studies on three different neurological disorder prediction tasks demonstrated the superiority of the proposed approach over existing state-of-the-art methods.

In the future, we will explore situations in machine learning where tensor factorization are used, for example, multi-source and multi-task learning [20, 21], and examine how the KCP and MMK methods could be applied to improve the learning performance.

Acknowledgments

This work is supported in part by NSFC through grants 61503253, 61672357 and 61672313, NSF through grants IIS-1526499 and CNS-1626432, NIH through grant R01-MH080636, and the Science Foundation of Guangdong Province through grant 2014A030313556.

References

[1] J. A. Bazerque, G. Mateos, and G. B. Giannakis. Nonparametric low-rank tensor imputation. In *SSP*, pages 876–879, 2012.

[2] B. Cao, L. He, X. Kong, S. Y. Philip, Z. Hao, and A. B. Ragin. Tensor-based multi-view feature selection with applications to brain diseases. In *ICDM*, pages 40–49, 2014.

[3] B. Cao, L. He, X. Wei, M. Xing, P. S. Yu, H. Klumpp, and A. D. Leow. t-bne: Tensor-based brain network embedding. In *SDM*, 2017.

[4] A. Cichocki. Tensor decompositions: a new concept in brain data analysis? *Journal of Control Measurement, and System Integration*, 6(7):507–517, 2013.

[5] J. V. Davis and I. S. Dhillon. Structured metric learning for high dimensional problems. In *KDD*, pages 195–203, 2008.

[6] L. De Lathauwer, B. De Moor, and J. Vandewalle. On the best rank-1 and rank-(r 1, r 2,..., rn) approximation of higher-order tensors. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1324–1342, 2000.

[7] X. Geng, K. Smith-Miles, Z.-H. Zhou, and L. Wang. Face image modeling by multilinear subspace analysis with missing values. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(3):881–892, 2011.

[8] T. Guo, L. Han, L. He, and X. Yang. A ga-based feature selection and parameter optimization for linear support higher-order tensor machine. *Neurocomputing*, 144:408–416, 2014.

[9] W. Guo, I. Kotsia, and I. Patras. Tensor learning for regression. *IEEE Transactions on Image Processing*, 21(2):816–827, 2012.

[10] A. Gupta, M. Ayhan, and A. Maida. Natural image bases to represent neuroimaging data. In *ICML*, pages 987–994, 2013.

[11] X. Han, Y. Zhong, L. He, S. Y. Philip, and L. Zhang. The unsupervised hierarchical convolutional sparse auto-encoder for neuroimaging data classification. In *BIH*, pages 156–166, 2015.

[12] Z. Hao, L. He, B. Chen, and X. Yang. A linear support higher-order tensor machine for classification. *IEEE Transactions on Image Processing*, 22(7):2911–2920, 2013.

[13] L. He, X. Kong, S. Y. Philip, A. B. Ragin, Z. Hao, and X. Yang. Dusk: A dual structure-preserving kernel for supervised tensor learning with applications to neuroimages. In *SDM*, 2014.

[14] R. Heaton, D. Clifford, D. Franklin, S. Woods, C. Ake, F. Vaida, R. Ellis, S. Letendre, T. Marcotte, J. Atkinson, et al. Hiv-associated neurocognitive disorders persist in the era of potent antiretroviral therapy charter study. *Neurology*, 75(23):2087–2096, 2010.

[15] A. Jukić, I. Kopriva, and A. Cichocki. Canonical polyadic decomposition for unsupervised linear feature extraction from protein profiles. In *EUSIPCO*, pages 1–5, 2013.

[16] T. G. Kolda. *Multilinear operators for higher-order decompositions*. United States. Department of Energy, 2006.

[17] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[18] I. Kotsia, W. Guo, and I. Patras. Higher rank support tensor machines for visual recognition. *Pattern Recognition*, 45(12):4192–4203, 2012.

[19] X. Liu, T. Guo, L. He, and X. Yang. A low-rank approximation-based transductive support tensor machine for semisupervised classification. *IEEE Transactions on Image Processing*, 24(6):1825–1838, 2015.

[20] C.-T. Lu, L. He, H. Ding, and P. S. Yu. Learning from multi-view structural data via structural factorization machines. *arXiv preprint arXiv:1704.03037*, 2017.

[21] C.-T. Lu, L. He, W. Shao, B. Cao, and P. S. Yu. Multilinear factorization machines for multi-task multi-view learning. In *WSDM*, pages 701–709, 2017.

[22] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. Mpca: Multilinear principal component analysis of tensor objects. *IEEE Transactions on Neural Networks*, 19(1):18–39, 2008.

[23] G. Ma, L. He, C.-T. Lu, P. S. Yu, L. Shen, and A. B. Ragin. Spatio-temporal tensor analysis for whole-brain fmri classification. In *SDM*, pages 819–827, 2016.

[24] D. Nion and L. De Lathauwer. An enhanced line search scheme for complex-valued tensor decompositions. application in ds-cdma. *Signal Processing*, 88(3):749–755, 2008.

[25] S. W. Park. Multifactor analysis for fmri brain image classification by subject and motor task. Electrical and computer engineering technical report, Carnegie Mellon University, 2011.

[26] B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. In *International Conference on Computational Learning Theory*, pages 416–426, 2001.

[27] W. Shao, L. He, and S. Y. Philip. Clustering on multi-source incomplete data via tensor modeling and factorization. In *PAKDD*, pages 485–497, 2015.

[28] M. Signoretto, L. De Lathauwer, and J. A. Suykens. A kernel-based framework to tensorial data analysis. *Neural networks*, 24(8):861–874, 2011.

[29] M. Signoretto, E. Olivetti, L. De Lathauwer, and J. A. Suykens. Classification of multichannel signals with cumulant-based kernels. *IEEE Transactions on Signal Processing*, 60(5):2304–2314, 2012.

[30] S. Song, Z. Zhan, Z. Long, J. Zhang, and L. Yao. Comparative study of svm methods combined with voxel selection for object category classification on fmri data. *PloS one*, 6(2):e17191, 2011.

[31] D. Tao, X. Li, X. Wu, W. Hu, and S. J. Maybank. Supervised tensor learning. *Knowledge and Information Systems*, 13(1):1–42, 2007.

[32] S. Wang, L. He, L. Stenneth, P. S. Yu, and Z. Li. Citywide traffic congestion estimation with social media. In *GIS*, page 34, 2015.

[33] X. Wang, P. Foryt, R. Ochs, J.-H. Chung, Y. Wu, T. Parrish, and A. B. Ragin. Abnormalities in resting-state functional connectivity in early human immunodeficiency virus infection. *Brain connectivity*, 1(3):207–217, 2011.

[34] S.-y. Xie, R. Guo, N.-f. Li, G. Wang, and H.-t. Zhao. Brain fmri processing and classification based on combination of pca and svm. In *IJCNN*, pages 3384–3389, 2009.

[35] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H.-J. Zhang. Multilinear discriminant analysis for face recognition. *IEEE Transactions on Image Processing*, 16(1):212–220, 2007.

[36] Q. Zhao, G. Zhou, T. Adali, L. Zhang, and A. Cichocki. Kernelization of tensor-based models for multiway data analysis: Processing of multidimensional structured data. *IEEE Signal Processing Magazine*, 30(4):137–148, 2013.

[37] H. Zhou, L. Li, and H. Zhu. Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, 108(502):540–552, 2013.

[38] S. Zhou, X. V. Nguyen, J. Bailey, Y. Jia, and I. Davidson. Accelerating online cp decompositions for higher order tensors. In *KDD*, 2016.