Multilinear and Nonlinear Generalizations of Partial Least Squares: An Overview of Recent Advances

Qibin Zhao, Liqing Zhang and Andrzej Cichocki

Abstract—Partial Least Squares (PLS) is an efficient multivariate statistical regression technique that has proven to be particularly useful for analysis of highly collinear data. To predict response variables $\mathbf{Y}$ from independent variables $\mathbf{X}$, PLS attempts to find a set of common orthogonal latent variables by projecting both $\mathbf{X}$ and $\mathbf{Y}$ onto a new subspace respectively. As an increasing interest in multiway analysis, the extension to multilinear regression model are also developed with the aim to analyzing two multidimensional tensor data. In this article, we overview the PLS related methods including linear, multilinear and nonlinear variants and discuss the strength of the algorithms. Since Canonical Correlation Analysis (CCA) is another similar technique with aim to extract the most correlated latent components between two datasets, we also briefly discuss the extension of CCA to tensor space. Finally, several examples are given to compare these methods with respect to the regression and classification performance.

Index Terms—Tensor decomposition, Partial least squares (PLS), Canonical Correlation Analysis (CCA), Electrocorticogram (ECoG), Kernel machines.

1 Introduction

The modern machine learning methodologies have been increasingly used to analyse the relationship between behavioral data and neuroscience data, such as functional magnetic resonance imaging (fMRI), electrocorticography (ECoG) and electroencephalography (EEG). Furthermore, due to recent improvements in neuroscience scanning technology, there has been an increasing interest in the analysis of various factors using cross-domain multiple sources. Tensors (also called multiway arrays) have been proven to be a natural and efficient representation for modeling such high-dimensional structured data. In particular, tensor subspace learning methods have been shown to outperform their corresponding vector subspace methods, including multilinear principal component analysis (PCA), multilinear discriminant analysis, and higher-order partial least squares (HO-PLS) [1]. Tensor-based techniques allow us to take into account the structure of data representation in model learning. The corresponding tensor subspace regression and classification attracted increasingly interest in computer vision, machine learning and neuroscience fields.

The Partial Least Squares (PLS) is a well-established framework for estimation, regression and classification, whose objective is to predict a set of dependent variables (responses) from a set of independent variables (predictors) through the extraction of a small number of latent variables. One member of the PLS family is Partial Least Squares Regression (PLSR) - a multivariate method, is proven to be particularly suited for highly collinear data [2], [3]. There are many variations of PLS model such as orthogonal projection on latent structures (O-PLS) [4], Biorthogonal PLS (BPLS) [5], recursive partial least squares (RPLS) [6], [7], nonlinear PLS [8], [9]. Penalized regression methods are also popular for simultaneous variable selection and coefficient estimation by imposing e.g. L2 or L1 constraints on the regression coefficients. Algorithms of this kind are ridge regression [10] and lasso [11].

In this paper, we reviewed the standard PLS, multilinear PLS and kernel-based tensor PLS methods in terms of modelling and algorithms. The objective is to provide a tutorial of the relevant topics by discussing the strengths of the algorithms. The article is structured as follows. In Section 2, the notation and notions related to multi-way data analysis are introduced. In Section 3, the extensions of PLS to tensor space are presented followed by a brief overview of canonical correlation analysis (CCA) in Section 4. In Section 5, the relationships among these algorithms are discussed together with some recent research directions. Simulation results on real-world applications are presented in Section 6, followed by conclusions in Section 7.

2 Preliminaries and Notations

In this paper, Nth-order tensors (*multi-way arrays*) are denoted by underlined boldface capital letters, matrices

- *Q. Zhao is with Laboratory for Advanced Brain Signal Processing, Brain Science Institute, RIKEN, Saitama, Japan and Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China.*
- *L. Zhang is with MOE-Microsoft Laboratory for Intelligent Computing and Intelligent Systems and Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China.*
- *A. Cichocki is with Laboratory for Advanced Brain Signal Processing, Brain Science Institute, RIKEN, Saitama, Japan and Systems Research Institute in Polish Academy of Science.*

(*two-way arrays)* by boldface capital letters, and vectors by boldface lower-case letters. The ith entry of a vector $\mathbf{x}$ is denoted by x_i, element (i,j) of a matrix $\mathbf{X}$ is denoted by x_{ij}, and element $(i_1, i_2, \ldots, i_N)$ of an Nth-order tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ by $x_{i_1 i_2 \ldots i_N}$ or $(\underline{\mathbf{X}})_{i_1 i_2 \ldots i_N}$. Indices typically range from 1 to their capital version, e.g., $i_N = 1, \ldots, I_N$. The mode-n matricization of a tensor is denoted by $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times I_1 \cdots I_{n-1} I_{n+1} \cdots I_N}$. The nth matrix in a sequence is denoted by a superscript in parentheses, i.e., $\mathbf{A}^{(n)}$.

The *n-mode product* of a tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_n \times \cdots \times I_N}$ and matrix $\mathbf{A} \in \mathbb{R}^{J_n \times I_n}$is denoted by $\underline{\mathbf{Y}} = \underline{\mathbf{X}} \times_n \mathbf{A} \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N}$ and is defined as:

$$y_{i_1 i_2 \ldots i_{n-1} j_n i_{n+1} \ldots i_N} = \sum_{i_n} x_{i_1 i_2 \ldots i_n \ldots i_N} a_{j_n i_n}. \quad (1)$$

while the n-mode product of Nth-order tensor $\underline{\mathbf{X}}$ and a vector $\mathbf{a} \in \mathbb{R}^{I_n}$ is an $N-1$th tensor denoted by $\underline{\mathbf{Y}} = \underline{\mathbf{X}} \bar{\times}_n \mathbf{a} \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N}$ and defined by

$$y_{i_1 i_2 \cdots i_{n-1} i_{n+1} \cdots i_N} = \sum_{i_n} x_{i_1 i_2 \cdots i_n \cdots i_N} a_{i_n} \quad (2)$$

The *rank-*$(R_1, R_2, ..., R_N)$ *Tucker model* [12] is a tensor decomposition defined as follows:

$$\underline{\mathbf{Y}} \approx \underline{\mathbf{G}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \cdots \times_N \mathbf{A}^{(N)}, \quad (3)$$

where $\underline{\mathbf{G}} \in \mathbb{R}^{R_1 \times R_2 \times .. \times R_N}$ is the *core tensor* and $\mathbf{A}^{(n)} = [\mathbf{a}_1^{(n)} \mathbf{a}_2^{(n)} \cdots \mathbf{a}_{R_n}^{(n)}] \in \mathbb{R}^{I_n \times R_n}$ are the *factor matrices*. When the factor matrices are restricted to be orthonormal this model is called multilinear singular value decomposition (MSVD). A useful property of MSVD is that it can be computed directly from data by applying SVD to each mode of the tensor, while keeping the left singular matrices as the factor matrices. The core tensor can then be computed as $\underline{\mathbf{G}} = \underline{\mathbf{Y}} \times_1 \mathbf{A}^{(1)T} \times_2 \mathbf{A}^{(2)T} \times_3 \cdots \times_N \mathbf{A}^{(N)T}$.

The canonical polyadic decomposition (CPD) [13], [14], [15], [16], [17] became prominent in Chemistry [18] and is defined as a sum of rank-one tensors:

$$\underline{\mathbf{Y}} \approx \sum_{r=1}^{R} \lambda_r \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \cdots \circ \mathbf{a}_r^{(N)}, \quad (4)$$

where the symbol '$\circ$' denotes the outer product of vectors, $\mathbf{a}_r^{(n)}$ is the column-r vector of matrix $\mathbf{A}^{(n)}$, and λ_r are scalars. This notation suggests the definition of *tensor rank*: we say that a tensor is *rank-*R if the minimal value of r in (4), providing a perfect fit to $\underline{\mathbf{Y}}$, is R. The CP model can also be represented by (3), under the condition that the core tensor is super-diagonal, i.e., $R = R_1 = R_2 = \cdots = R_N$ and $g_{r_1 r_2, \ldots, r_N} = 0$ if $r_n \neq r_m$ for all $n \neq m$.

The *inner product of two tensors* $\underline{\mathbf{A}}, \underline{\mathbf{B}} \in \mathbb{R}^{I_1 \times I_2 \ldots \times I_N}$ is defined by $\langle \underline{\mathbf{A}}, \underline{\mathbf{B}} \rangle = \sum_{i_1 i_2 \ldots i_N} a_{i_1 i_2 \ldots i_N} b_{i_1 i_2 \ldots i_N}$, and the squared Frobenius norm by $\|\underline{\mathbf{A}}\|_F^2 = \langle \underline{\mathbf{A}}, \underline{\mathbf{A}} \rangle$.

The *n-mode cross-covariance* between an Nth-order tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_n \times \cdots \times I_N}$ and an Mth-order tensor $\underline{\mathbf{Y}} \in \mathbb{R}^{J_1 \times \cdots \times I_n \times \cdots \times J_M}$ with the same size I_n on the nth-mode, denoted by $\mathrm{COV}_{\{n;n\}}(\underline{\mathbf{X}}, \underline{\mathbf{Y}}) \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N \times J_1 \times \cdots \times J_{n-1} \times J_{n+1} \times \cdots \times J_M}$, is defined as

$$\underline{\mathbf{C}} = \mathrm{COV}_{\{n;n\}}(\underline{\mathbf{X}}, \underline{\mathbf{Y}}) = < \underline{\mathbf{X}}, \underline{\mathbf{Y}} >_{\{n;n\}}, \quad (5)$$

where the symbol $< \bullet, \bullet >_{\{n;n\}}$ represents an n-mode multiplication between two tensors, and is defined as

$$c_{i_1, \ldots, i_{n-1}, i_{n+1} \ldots i_N, j_1, \ldots, j_{n-1} j_{n+1} \ldots j_M} = \sum_{i_n=1}^{I_n} x_{i_1, \ldots, i_n, \ldots, i_N} y_{j_1, \ldots, i_n, \ldots, j_M}. \quad (6)$$

3 Partial Least Squares

3.1 Linear PLS

The PLS regression was originally developed for econometrics by H. Wold [19], [20] in order to deal with collinear predictor variables. For this case, the ordinary least squares regression fails due to the ill-conditioning of data matrices. The usefulness of PLS in chemical applications was illuminated by the group of S. Wold [21], [22], after some initial work by Kowalski *et al*. [23]. Currently, the PLS regression is being widely applied in chemometrics, sensory evaluation, industrial process control, and more recently, in the analysis of functional brain imaging data [24], [25], [26], [27], [28].

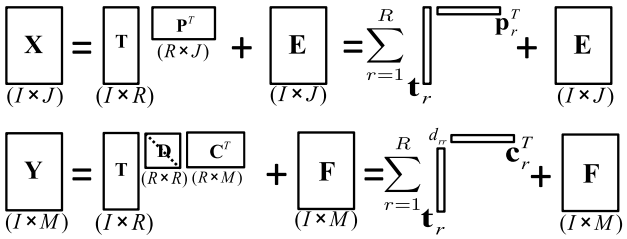


Fig. 1. The PLS model: data decomposition as a sum of rank-one matrices.

The principle behind PLS is to search for a set of latent vectors by performing a simultaneous decomposition of $\mathbf{X} \in \mathbb{R}^{I \times J}$ and $\mathbf{Y} \in \mathbb{R}^{I \times M}$ with the constraint that these components explain as much as possible of the covariance between $\mathbf{X}$ and $\mathbf{Y}$. As illustrated in Fig. 1, this can be formulated as

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E} = \sum_{r=1}^{R} \mathbf{t}_r \mathbf{p}_r^T + \mathbf{E}, \quad (7)$$

$$\mathbf{Y} = \mathbf{U}\mathbf{C}^T + \mathbf{F} = \sum_{r=1}^{R} \mathbf{u}_r \mathbf{c}_r^T + \mathbf{F}, \quad (8)$$

where $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_R] \in \mathbb{R}^{I \times R}$ consists of R orthonormal latent variables from $\mathbf{X}$, and $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_R] \in \mathbb{R}^{I \times R}$ are latent variables from $\mathbf{Y}$ having maximum covariance with $\mathbf{T}$ column-wise. The matrices $\mathbf{P}$ and $\mathbf{C}$ represent loadings and $\mathbf{E}, \mathbf{F}$ are respectively the residuals for $\mathbf{X}$ and $\mathbf{Y}$. In order to find the first set of components, the classical PLS algorithm is to optimize the two sets of weights $\mathbf{w}, \mathbf{c}$ so as to satisfy

$$\max_{\{\mathbf{w}, \mathbf{c}\}} [\mathbf{w}^T \mathbf{X}^T \mathbf{Y} \mathbf{c}]^2, \quad \text{s. t.} \quad \mathbf{w}^T \mathbf{w} = 1, \mathbf{c}^T \mathbf{c} = 1. \quad (9)$$

The latent variables are then given by $\mathbf{t} = \mathbf{X}\mathbf{w}/\|\mathbf{X}\mathbf{w}\|$ and $\mathbf{u} = \mathbf{Y}\mathbf{c}$. Here, two assumptions are made: i) the latent variables $\{\mathbf{t}_r\}_{r=1}^{R}$ are good predictors of $\mathbf{Y}$; ii) a linear inner relation between the latent variables $\mathbf{t}$ and $\mathbf{u}$ exists; i.e., $\mathbf{U} = \mathbf{T}\mathbf{D} + \mathbf{Z}$ where $\mathbf{D}$ is a linear relation diagonal matrix and $\mathbf{Z}$ denotes the matrix of Gaussian i.i.d. residuals. Upon combining it with the decomposition of $\mathbf{Y}$, (8) can be written as

$$\mathbf{Y} = \mathbf{T}\mathbf{D}\mathbf{C}^T + (\mathbf{Z}\mathbf{C}^T + \mathbf{F}) = \mathbf{T}\mathbf{D}\mathbf{C}^T + \mathbf{F}^*, \qquad (10)$$

where $\mathbf{F}^*$ is the residual matrix. Thus (10) indicates that the problem boils down to finding common latent variables $\mathbf{T}$ that explain the variance of both $\mathbf{X}$ and $\mathbf{Y}$.

3.2 Multilinear PLS

The N-way PLS (N-PLS), illustrated in Fig. 2, was developed by Bro [29] as a multi-way extension of standard PLS, which decomposes a multi-way tensor $\underline{\mathbf{X}}$ based on the CP model, to predict response variables $\mathbf{Y}$. For a three-way tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I\times J\times K}$ and a multivariate matrix $\mathbf{Y} \in \mathbb{R}^{I\times M}$ with elements x_{ijk} and y_{im} respectively, $\underline{\mathbf{X}}$ is decomposed into one latent vector $\mathbf{t} \in \mathbb{R}^{I\times 1}$ and two loading vectors $\mathbf{p} \in \mathbb{R}^{J\times 1}$ and $\mathbf{q} \in \mathbb{R}^{K\times 1}$, i.e., one loading vector per mode. The decomposition model for $\underline{\mathbf{X}}$ is given by

$$\underline{\mathbf{X}} = \sum_{r=1}^{R} \mathbf{t}_r \circ \mathbf{p}_r \circ \mathbf{q}_r + \underline{\mathbf{E}}, \quad \mathbf{Y} = \sum_{r=1}^{R} d_{rr}\, \mathbf{t}_r \mathbf{c}_r^T + \mathbf{F} \qquad (11)$$

and the objective is to find the vectors $\mathbf{p}_r, \mathbf{q}_r$ and $\mathbf{c}_r$ that satisfy

$$\begin{aligned} \{\mathbf{p}_r, \mathbf{q}_r, \mathbf{c}_r\} &= \arg\max_{\mathbf{p}_r,\mathbf{q}_r,\mathbf{c}_r} [\mathrm{cov}(\mathbf{t}_r, \mathbf{u}_r)], \\ \text{s. t.} \quad \mathbf{t}_r &= \underline{\mathbf{X}} \bar{\times}_1 \mathbf{p}_r \bar{\times}_2 \mathbf{q}_r, \mathbf{u}_r = \mathbf{Y}\mathbf{c}_r \\ \text{and} \quad \|\mathbf{p}_r\|_2^2 &= \|\mathbf{q}_r\|_2^2 = \|\mathbf{c}_r\|_2^2 = 1. \end{aligned} \qquad (12)$$

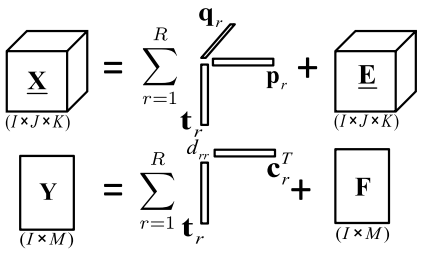


Fig. 2. The N-PLS model: data decomposition as a sum of rank-one tensors and a sum of rank-one matrices.

3.3 Higher-order PLS (HOPLS)

Another multilinear regression model, termed higher-order partial least squares (HOPLS) [1], [30], operates by modeling Nth-order independent tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I_1\times\cdots\times I_N}$ and an Mth-order dependent tensor $\underline{\mathbf{Y}} \in \mathbb{R}^{J_1\times\cdots\times J_M}$, having the same size on the first mode, i.e., $I_1 = J_1$ (see Fig. 3). This allows us to find the optimal subspace approximation of $\underline{\mathbf{X}}$, in which the independent and

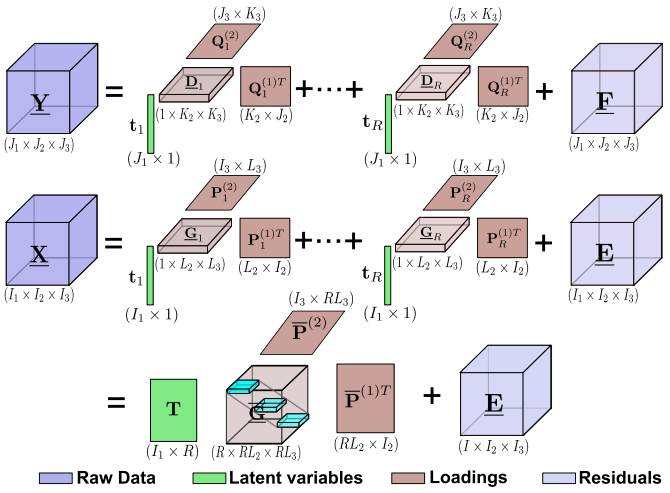


Fig. 3. Schematic diagram of the HOPLS model: approximating $\underline{\mathbf{X}}$ as a sum of rank-$(1, L_2, L_3)$ tensors. Approximation for $\underline{\mathbf{Y}}$ follows a similar principle with shared common latent components $\mathbf{T}$.

dependent variables share a common set of latent vectors on one specific mode (i.e., samples mode). More specifically, we assume $\underline{\mathbf{X}}$ is decomposed as a sum of rank-$(1, L_2, \ldots, L_N)$ Tucker blocks, while $\underline{\mathbf{Y}}$ is decomposed as a sum of rank-$(1, K_2, \ldots, K_M)$ Tucker blocks, which can be expressed as

$$\begin{aligned} \underline{\mathbf{X}} &= \sum_{r=1}^{R} \underline{\mathbf{G}}_r \times_1 \mathbf{t}_r \times_2 \mathbf{P}_r^{(1)} \times_3 \cdots \times_N \mathbf{P}_r^{(N-1)} + \underline{\mathbf{E}}_R, \\ \underline{\mathbf{Y}} &= \sum_{r=1}^{R} \underline{\mathbf{D}}_r \times_1 \mathbf{t}_r \times_2 \mathbf{Q}_r^{(1)} \times_3 \cdots \times_M \mathbf{Q}_r^{(M-1)} + \underline{\mathbf{F}}_R, \end{aligned} \qquad (13)$$

where R is the number of latent vectors, $\mathbf{t}_r \in \mathbb{R}^{I_1}$ is the r-th latent vector, $\left\{\mathbf{P}_r^{(n)}\right\}_{n=1}^{N-1} \in \mathbb{R}^{I_{n+1}\times L_{n+1}}$ and $\left\{\mathbf{Q}_r^{(m)}\right\}_{m=1}^{M-1} \in \mathbb{R}^{J_{m+1}\times K_{m+1}}$ are loading matrices on mode-n and mode-m respectively, and $\underline{\mathbf{G}}_r \in \mathbb{R}^{1\times L_2\times\cdots\times L_N}$ and $\underline{\mathbf{D}}_r \in \mathbb{R}^{1\times K_2\times\cdots\times K_M}$ are core tensors. By defining a latent matrix $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_R]$, mode-$n$ loading matrix $\overline{\mathbf{P}}^{(n)} = [\mathbf{P}_1^{(n)}, \ldots, \mathbf{P}_R^{(n)}]$, mode-$m$ loading matrix $\overline{\mathbf{Q}}^{(m)} = [\mathbf{Q}_1^{(m)}, \ldots, \mathbf{Q}_R^{(m)}]$ and core tensor $\overline{\mathbf{G}} = \mathrm{blockdiag}(\underline{\mathbf{G}}_1, \ldots, \underline{\mathbf{G}}_R) \in \mathbb{R}^{R\times RL_2\times\cdots\times RL_N}$, $\overline{\mathbf{D}} = \mathrm{blockdiag}(\underline{\mathbf{D}}_1, \ldots, \underline{\mathbf{D}}_R) \in \mathbb{R}^{R\times RK_2\times\cdots\times RK_M}$, the HOPLS model in (13) can be rewritten as

$$\begin{aligned} \underline{\mathbf{X}} &= \overline{\mathbf{G}} \times_1 \mathbf{T} \times_2 \overline{\mathbf{P}}^{(1)} \times_3 \cdots \times_N \overline{\mathbf{P}}^{(N-1)} + \underline{\mathbf{E}}_R, \\ \underline{\mathbf{Y}} &= \overline{\mathbf{D}} \times_1 \mathbf{T} \times_2 \overline{\mathbf{Q}}^{(1)} \times_3 \cdots \times_M \overline{\mathbf{Q}}^{(M-1)} + \underline{\mathbf{F}}_R, \end{aligned} \qquad (14)$$

where $\underline{\mathbf{E}}_R$ and $\underline{\mathbf{F}}_R$ are residuals after extracting R components. The core tensors $\overline{\mathbf{G}}$ and $\overline{\mathbf{D}}$ have a special block-diagonal structure (see Fig. 3) and their elements indicate the level of local interactions between the corresponding latent vectors and loading matrices.

Benefiting from the advantages of Tucker decomposition over the CP model [17], HOPLS promises to approximate data better than N-PLS. Specifically, HOPLS

differs substantially from the N-PLS model in the sense that the size of loading matrices is controlled by a hyperparameter, providing a tradeoff between fitness and model complexity. Note that HOPLS simplifies into N-PLS if we define $\forall n : \{L_n\} = 1$ and $\forall m : \{K_m\} = 1$.

The optimization of subspace transformation according to (13) will be formulated as a problem of determining a set of orthogonormal loadings and latent vectors. Since each term can be optimized sequentially with the same criteria based on deflation, in the following, we shall simplify the problem to that of finding the first latent vector $\mathbf{t}$ and two sequences of loading matrices $\mathbf{P}^{(n)}$ and $\mathbf{Q}^{(m)}$. Finally, if we define $< \underline{\mathbf{X}}, \underline{\mathbf{Y}} >_{\{1;1\}}$ as a mode-1 cross-covariance tensor $\underline{\mathbf{C}} = \text{COV}_{\{1;1\}}(\underline{\mathbf{X}}, \underline{\mathbf{Y}}) \in \mathbb{R}^{I_2 \times \cdots \times I_N \times J_2 \times \cdots \times J_M}$, the optimization problem can be finally formulated as

$$\max_{\{\mathbf{P}^{(\mathbf{n})}, \mathbf{Q}^{(\mathbf{m})}\}} \left\| [\![\underline{\mathbf{C}}; \mathbf{P}^{(1)T}, \ldots, \mathbf{P}^{(N-1)T}, \mathbf{Q}^{(1)T}, \ldots, \mathbf{Q}^{(M-1)T}]\!] \right\|_F^2$$
$$\text{s. t.} \quad \mathbf{P}^{(n)T}\mathbf{P}^{(n)} = \mathbf{I}_{L_{n+1}}, \mathbf{Q}^{(m)T}\mathbf{Q}^{(m)} = \mathbf{I}_{K_{m+1}}, \quad (15)$$

where $\mathbf{P}^{(n)}, n = 1, \ldots, N-1$ and $\mathbf{Q}^{(m)}, m = 1, \ldots, M-1$ are the parameters to optimize. This is equivalent to find the best subspace approximation of $\underline{\mathbf{C}}$ which can be obtained by rank-$(L_2, \ldots, L_N, K_2, \ldots, K_M)$ HOSVD on tensor $\underline{\mathbf{C}}$. To achieve this goal, the higher-order orthogonal iteration (HOOI) algorithm [15], [17], which is known to converge fast, is employed to find the parameters $\mathbf{P}^{(n)}$ and $\mathbf{Q}^{(m)}$ by orthogonal Tucker decomposition of $\underline{\mathbf{C}}$.

3.4 Nonlinear tensor PLS

In this section, we introduce kernel-based tensor PLS (KTPLS) [31] as an extension of HOPLS to kernel spaces. Given N pairs of tensor observations $\{(\underline{\mathbf{X}}^{(n)}, \underline{\mathbf{Y}}^{(n)})\}_{n=1}^N$, $\underline{\mathbf{X}}^{(n)}$ denotes an Mth-order independent tensor and $\underline{\mathbf{Y}}^{(n)}$ denotes an Lth-order dependent tensor, which can be concatenated to form an $(M+1)$th-order tensor $\underline{\mathbf{X}} \in \mathbb{R}^{N \times I_1 \times \cdots \times I_M}$ and $(L+1)$th-order tensor $\underline{\mathbf{Y}} \in \mathbb{R}^{N \times J_1 \times \cdots \times J_L}$. We then let $\underline{\mathbf{X}}$, $\underline{\mathbf{Y}}$ to be mapped into the Hilbert space by $\phi : \quad \underline{\mathbf{X}}^{(n)} \longmapsto \phi\left(\underline{\mathbf{X}}^{(n)}\right)$. For simplicity, we denote $\phi(\underline{\mathbf{X}})$ by $\mathbf{\Phi}$ and $\phi(\underline{\mathbf{Y}})$ by $\mathbf{\Psi}$. KTPLS seeks tensor decompositions such that

$$\begin{aligned} \mathbf{\Phi} &= \underline{\mathbf{G}}_{\underline{\mathbf{X}}} \times_1 \mathbf{T} \times_2 \mathbf{P}^{(1)} \cdots \times_{M+1} \mathbf{P}^{(M)} + \underline{\mathbf{E}}_{\underline{\mathbf{X}}}, \\ \mathbf{\Psi} &= \underline{\mathbf{G}}_{\underline{\mathbf{Y}}} \times_1 \mathbf{U} \times_2 \mathbf{Q}^{(1)} \cdots \times_{L+1} \mathbf{Q}^{(L)} + \underline{\mathbf{E}}_{\underline{\mathbf{Y}}}, \\ \mathbf{U} &= \mathbf{TD} + \mathbf{E}_U, \end{aligned} \quad (16)$$

Since $\underline{\mathbf{G}}_{\underline{\mathbf{X}}} \times_2 \mathbf{P}^{(1)} \cdots \times_{M+1} \mathbf{P}^{(M)}$ denoted by $\widetilde{\underline{\mathbf{G}}}_{\underline{\mathbf{X}}}$ and $\underline{\mathbf{G}}_{\underline{\mathbf{Y}}} \times_2 \mathbf{Q}^{(1)} \cdots \times_{L+1} \mathbf{Q}^{(L)}$ denoted by $\widetilde{\underline{\mathbf{G}}}_{\underline{\mathbf{Y}}}$ can be represented as a linear combination of $\{\phi(\underline{\mathbf{X}}^{(n)})\}$ and $\{\phi(\underline{\mathbf{Y}}^{(n)})\}$ respectively, i.e., $\widetilde{\underline{\mathbf{G}}}_{\underline{\mathbf{X}}} = \mathbf{\Phi} \times_1 \mathbf{T}^T$ and $\widetilde{\underline{\mathbf{G}}}_{\underline{\mathbf{Y}}} = \mathbf{\Psi} \times_1 \mathbf{U}^T$, we only need to explicitly find the latent vectors of $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_R]$ and $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_R]$ with pairwise maximum covariance through solving an optimization problem sequentially, which is expressed by

$$\begin{aligned} \max_{\{\mathbf{w}_r^{(m)}, \mathbf{v}_r^{(l)}\}} \quad & [\text{cov}(\mathbf{t}_r, \mathbf{u}_r)]^2, \\ \text{where} \quad \mathbf{t}_r &= \mathbf{\Phi} \bar{\times}_2 \mathbf{w}_r^{(1)} \cdots \bar{\times}_{M+1} \mathbf{w}_r^{(M)}, \\ \mathbf{u}_r &= \mathbf{\Psi} \bar{\times}_2 \mathbf{v}_r^{(1)} \cdots \bar{\times}_{L+1} \mathbf{v}_r^{(L)}. \end{aligned} \quad (17)$$

Rewriting (17) in matrix form, it becomes $\mathbf{t}_r = \mathbf{\Phi}_{(1)} \widetilde{\mathbf{w}}_r, \mathbf{u}_r = \mathbf{\Psi}_{(1)} \widetilde{\mathbf{v}}_r$, which can be solved by kernelized version of the eigenvalue problem, i.e., $\mathbf{\Phi}_{(1)}\mathbf{\Phi}_{(1)}^T\mathbf{\Psi}_{(1)}\mathbf{\Psi}_{(1)}^T\mathbf{t}_r = \lambda\mathbf{t}_r$ and $\mathbf{u}_r = \mathbf{\Psi}_{(1)}\mathbf{\Psi}_{(1)}^T\mathbf{t}_r$ [8]. Note that $\mathbf{\Phi}_{(1)}\mathbf{\Phi}_{(1)}^T$ contains only the inner products between vectorized input tensors, which can be replaced by an $N \times N$ kernel matrix $\mathbf{K}_{\underline{\mathbf{X}}}$. Thus, we have $\mathbf{K}_{\underline{\mathbf{X}}}\mathbf{K}_{\underline{\mathbf{Y}}}\mathbf{t}_r = \lambda\mathbf{t}_r$ and $\mathbf{u}_r = \mathbf{K}_{\underline{\mathbf{Y}}}\mathbf{t}_r$. In order to take the multilinear structure into account, the kernel matrices should be computed using the kernel functions for tensors that will be discussed in the next section, i.e., $(\mathbf{K}_{\underline{\mathbf{X}}})_{nn'} = k\left(\underline{\mathbf{X}}^{(n)}, \underline{\mathbf{X}}^{(n')}\right)$ and $(\mathbf{K}_{\underline{\mathbf{Y}}})_{nn'} = k\left(\underline{\mathbf{Y}}^{(n)}, \underline{\mathbf{Y}}^{(n')}\right)$. Finally, the prediction of a novel data point $\underline{\mathbf{X}}^*$ can be achieved by

$$\mathbf{y}^{*T} = \mathbf{k}^{*T}\mathbf{U}(\mathbf{T}^T\mathbf{K}_{\underline{\mathbf{X}}}\mathbf{U})^{-1}\mathbf{T}^T\mathbf{Y}_{(1)}, \quad (18)$$

where $(\mathbf{k}^*)_n = k\left(\underline{\mathbf{X}}^{(n)}, \underline{\mathbf{X}}^*\right)$ and $\mathbf{y}^{*T}$ should be reorganized to tensor form $\underline{\mathbf{Y}}^*$.

The significance of (18) can be explained in several ways. First, it is a linear combination of N observations $\{\underline{\mathbf{Y}}^{(n)}\}$ with the coefficients $\mathbf{k}^{*T}\mathbf{U}(\mathbf{T}^T\mathbf{K}_{\underline{\mathbf{X}}}\mathbf{U})^{-1}\mathbf{T}^T$; the second interpretation is that y_j^* is predicted by a linear combination of N kernels, each one centered on a training point, i.e., $y_j^* = \sum_{n=1}^N \alpha_n k\left(\underline{\mathbf{X}}^{(n)}, \underline{\mathbf{X}}^*\right)$, where $\alpha_n = \left(\mathbf{U}(\mathbf{T}^T\mathbf{K}_{\underline{\mathbf{X}}}\mathbf{U})^{-1}\mathbf{T}^T\mathbf{Y}_{(1)}\right)_{nj}$. Finally, a third interpretation is that $\mathbf{t}^*$ is obtained by nonlinearly projecting $\underline{\mathbf{X}}^*$ onto the latent space, i.e., $\mathbf{t}^{*T} = \mathbf{k}^{*T}\mathbf{U}(\mathbf{T}^T\mathbf{K}_{\underline{\mathbf{X}}}\mathbf{U})^{-1}$, then $\mathbf{y}^{*T}$ is predicted by a linear regression against $\mathbf{t}^*$, i.e., $\mathbf{y}^{*T} = \mathbf{t}^{*T}\mathbf{C}$ where regression coefficient is $\mathbf{C} = \mathbf{T}^T\mathbf{Y}_{(1)}$. In general, to ensure the strict linear relationship between latent vectors and output in original spaces, the kernel function on data $\underline{\mathbf{Y}}$ is restricted to linear kernels.

3.5 Kernel function for tensors

The kernels are considered as defining a topology implying the *a priori* knowledge about invariance in the input space. In this section, we discuss the kernels for tensor-valued inputs, which can take multiway structure into account for similarity measures. There are some valid reproducing kernels toward a straightforward generalization to Mth-order tensors, such as the kernel functions $k : \underline{\mathbf{X}} \times \underline{\mathbf{X}} \to \mathbb{R}$ given as

$$\begin{aligned} &\text{Linear kernel:} \quad k(\underline{\mathbf{X}}, \underline{\mathbf{X}}') = \langle \text{vec}(\underline{\mathbf{X}}), \text{vec}(\underline{\mathbf{X}}') \rangle, \\ &\text{Gaussian-RBF kernel:} \quad k(\underline{\mathbf{X}}, \underline{\mathbf{X}}') = \exp\left(-\frac{1}{2\beta^2}\|\underline{\mathbf{X}} - \underline{\mathbf{X}}'\|_F^2\right). \end{aligned} \quad (19)$$

In order to define the similarity measure that directly exploits multilinear algebraic structure of input tensors, Signoretto *et al.* [32], [33] proposed a tensorial

kernel exploiting algebraic geometry of spaces of tensors and a similarity measure between the different subspaces spanned by higher-order tensors. A product kernel can be defined by M factor kernels, e.g., $k(\underline{\mathbf{X}}, \underline{\mathbf{X}}') = \prod_{m=1}^{M} k\left(\mathbf{X}_{(m)}, \mathbf{X}'_{(m)}\right)$, where each factor kernel represents a similarity measure between mode-m matricization of two tensors.

In [31], a family of probabilistic product kernels for tensors are proposed based on generative models. More specifically, an Mth-order tensor observations are first mapped into an M-dimensional model space, then information divergence is applied as a similarity measure in the model space. The probabilistic tensor kernels can deal with multiway data with missing values and variable length. Since it provides a way to model one tensor from M different viewpoints that correspond to different low-dimensional vector space, multiway relations can be captured in the similarity measure. The similarity measure between two tensors $\underline{\mathbf{X}}$ and $\underline{\mathbf{X}}'$ in mode-m is defined as

$$S_m(\underline{\mathbf{X}}||\underline{\mathbf{X}}') = D\left(p(\mathbf{x}|\mathbf{\Omega}_m^{\underline{\mathbf{X}}})\|q(\mathbf{x}|\mathbf{\Omega}_m^{\underline{\mathbf{X}}'})\right), \tag{20}$$

where p, q represent probability density function for $\underline{\mathbf{X}}$ and $\underline{\mathbf{X}}'$ respectively and $D(p||q)$ is an information divergence between two distributions. $\Omega_m^{\underline{\mathbf{X}}}$ denotes the parameters of mode-m distribution of $\underline{\mathbf{X}}$, which depends on the model assumption. For simplicity, we assume Gaussian models for all modes of $\underline{\mathbf{X}}$, then Ω includes mean values and covariance matrix of that distribution. One popular information divergence is the *symmetric Kullback-Leibler (sKL)* divergence [34] Another possibility is the Jensen-Shannon (JS) divergence [35], [36] expressed by

$$D_{JS}(p||q) = \frac{1}{2}\mathrm{KL}(p||r) + \frac{1}{2}\mathrm{KL}(q||r), \tag{21}$$

where $\mathrm{KL}(\cdot||\cdot)$ denotes Kullback-Leibler divergence and $r(\mathbf{x}) = \frac{p(\mathbf{x})+q(\mathbf{x})}{2}$. Finally, a probabilistic product kernel for tensors is defined as

$$k(\underline{\mathbf{X}}, \underline{\mathbf{X}}') = \alpha^2 \prod_{m=1}^{M} \exp\left(-\frac{1}{2\beta_m^2} S_m(\underline{\mathbf{X}}||\underline{\mathbf{X}}')\right), \tag{22}$$

where α denotes a magnitude parameter and $[\beta_1, \ldots, \beta_M]$ are length-scales parameters. As isotropic RBF kernel, $\{\beta_m\}_{m=1}^{M}$ in (22) could also be the same. It can be shown that both sKL and JS divergences are non-negative and equal to zero when $p(\mathbf{x}) = q(\mathbf{x})$, while they do not fulfill the triangle inequality. However, it has been proven in [35], [36] that $[D_{sKL}(p||q)]^{\frac{1}{2}}$ and $[D_{JS}(p||q)]^{\frac{1}{2}}$ fulfill the triangle inequality thus is a metric, implying that *tensor kernel* defined in (22) is a metric kernel. In practice, because of the absence of closed-form solutions for probabilistic kernels, one may end up with a kernel matrix that is not positive-definite due to inaccuracies in the probabilistic density estimation.

An intuitive interpretation of kernel function in (22) is that Mth-order tensors are assumed to be generated from M generative models, the similarity of these models measured by information divergence are employed to provide a multiple kernel with well conditions. This kernel can effectively capture the statistical properties of tensors, which might be a powerful tool for multidimensional structured data analysis, such as video classification and multichannel ECoG feature extractions.

4 Canonical correlation Analysis

Canonical correlation analysis (CCA) is a method similar to PLS for determining the linear relationships between two multivatiates. However, unlike PLS, CCA is to maximize the mutual correlations in latent space. As a result, CCA has been successfully applied in various practical contexts, such as supervised dimensionality reduction, multi-view learning, and multilabel classification [37]. Kernel canonical correlation analysis (KCCA) has been proposed in [38], [39], while multilinear extension of CCA for third-order tensors was proposed in [40].

Let $\mathbf{X} \in \mathbb{R}^{I\times J}$ and $\mathbf{Y} \in \mathbb{R}^{I\times M}$ be two datasets of I observations, the linear CCA seeks two projections $\mathbf{w}, \mathbf{v}$ such that

$$\max_{\{\mathbf{w},\mathbf{v}\}} \quad \frac{\mathbf{w}^T\mathbf{X}^T\mathbf{Y}\mathbf{v}}{\sqrt{(\mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w})(\mathbf{v}^T\mathbf{Y}^T\mathbf{Y}\mathbf{v})}}. \tag{23}$$

When two datasets are represented by Nth-order tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I_1,\times I_2\cdots I_N}$ and Mth-order tensor $\underline{\mathbf{Y}} \in \mathbb{R}^{K_1,\times K_2\cdots K_M}$, with $I_1 = K_1$ denotes the number of samples, the tensor CCA seeks multilinear projections of two datasets respectively, resulting in that the correlation coefficient between them are maximised, that is

$$\begin{aligned} \max_{\{\mathbf{w}^{(n)},\mathbf{v}^{(m)}\}} \quad & \langle \underline{\mathbf{X}} \bar{\times}_{n\neq 1}\mathbf{w}^{(n)}, \underline{\mathbf{Y}} \bar{\times}_{m\neq 1}\mathbf{v}^{(m)}\rangle, \\ \text{subject to} \quad & \langle \underline{\mathbf{X}} \bar{\times}_{n\neq 1}\mathbf{w}^{(n)}, \underline{\mathbf{X}} \bar{\times}_{n\neq 1}\mathbf{w}^{(n)}\rangle = 1, \\ & \langle \underline{\mathbf{Y}} \bar{\times}_{m\neq 1}\mathbf{v}^{(m)}, \underline{\mathbf{Y}} \bar{\times}_{m\neq 1}\mathbf{v}^{(m)}\rangle = 1, \end{aligned} \tag{24}$$

where $\bar{\times}_{n\neq 1}$ denotes the multiplications of a tensor and a set of vectors in all mode-n except mode-1, resulting in a vector with the same length to mode-1 size of the original tensor.

KCCA is an nonlinear generalization of linear CCA using kernel trick while the kernel-based tensor CCA (KTCCA) [31] is a multiway generalization of KCCA. By employing the kernel techniques, these two algorithms can be formulated in a similar form. More specifically, given N observations of two random tensors denoted as $\underline{\mathbf{X}}$ and $\underline{\mathbf{Y}}$, the objective of KTCCA are given by

$$\max_{\{\mathbf{w},\mathbf{v}\}} \quad \frac{\mathbf{w}^T\mathbf{K}_{\underline{\mathbf{X}}}\mathbf{K}_{\underline{\mathbf{Y}}}\mathbf{v}}{\sqrt{(\mathbf{w}^T\mathbf{K}_{\underline{\mathbf{X}}}^2\mathbf{w})(\mathbf{v}^T\mathbf{K}_{\underline{\mathbf{Y}}}^2\mathbf{v})}}, \tag{25}$$

where the kernel matrices $\mathbf{K}_{\underline{\mathbf{X}}}, \mathbf{K}_{\underline{\mathbf{Y}}}$ are computed using the kernel function for tensorial data as defined in (22), $\{\mathbf{w}, \mathbf{v}\} \in \mathbb{R}^N$ are weight coefficients in kernel space corresponding to $\underline{\mathbf{X}}$ and $\underline{\mathbf{Y}}$, respectively. The solution can be obtained by solving $\mathbf{w} = \frac{1}{\lambda}\mathbf{K}_{\underline{\mathbf{X}}}^{-1}\mathbf{K}_{\underline{\mathbf{Y}}}\mathbf{v}$, $\mathbf{K}_{\underline{\mathbf{Y}}}^2\mathbf{v} - \lambda^2\mathbf{K}_{\underline{\mathbf{Y}}}^2\mathbf{v} = \mathbf{0}$, which has the same form as KCCA [39].

5 Discussions

The standard PLS/CCA are well-known techniques for analyzing the linear relationships between two sets of multivariate data. Both of them can be used for feature extraction, regression and classification problems. However, when the original data can be represented naturally by multiway tensors such as multidimensional structured data, the multilinear generalization of PLS/CCA are more appropriate due to the power of effectively capturing the underlying interactions between different modes. From the optimization point of view, the multilinear versions of PLS/CCA are considered as constrained variants in the sense that the projection vectors can be approximated by rank-one tensors. For instance, given $\underline{\mathbf{X}}, \underline{\mathbf{Y}}$ with the same size of $10 \times 10 \times 10$, if we apply standard two-way PLS/CCA on the unfolding of original tensors, i.e., $\mathbf{X}, \mathbf{Y}$ of size 10×100, the linear transformations represented by $\mathbf{w} \in \mathbb{R}^{100}, \mathbf{v} \in \mathbb{R}^{100}$ are required to be optimized, while tensor PLS/CCA seek the multilinear transformations represented by $\mathbf{w}^{(1)}, \mathbf{w}^{(2)} \in \mathbb{R}^{10}$ and $\mathbf{v}^{(1)}, \mathbf{v}^{(2)} \in \mathbb{R}^{10}$ which is equivalent to applying linear transformation on unfolded $\mathbf{X}$ by $\mathbf{w}^{(1)} \otimes \mathbf{w}^{(2)}$. Therefore, the tensor PLS/CCA have some advantages on structured data analysis, and are less prone to overfitting problem when number of samples are relative small, because the model complexity is controlled to some extent by the rank-one approximation of projection vectors. However, if the original data does not contain the multiway structure, tensor PLS/CCA will obtain a model with higher fitting errors. Linear PLS/CCA are severely limited as they can only be applied to data that is linearly correlated while real world applications usually can not be expressed by a linear combinations of the input attributes, whereas nonlinear generalization of PLS and CCA are more powerful in this case. If we consider the tensor representations of original data, the recently proposed kernel-based tensor PLS/CCA are very powerful due to the ability of taking into account both the structure information and nonlinear relationships among the datasets.

There are many other variants of PLS/CCA, such as the extension to multiple datasets by multi-block PLS [41] and multi-view CCA [42]. To control the model complexity and prevent overfitting, the sparsity constraint can be enforced to obtain the latent components and to perform variable selection simultaneously [43]. On the other hand, PLS and CCA can be interpreted as Gaussian latent variable model under the probabilistic framework, such as probabilistic CCA [44] and probablistic PLSR [45], which results in the possibility of the fully Bayesian treatment of the model [46]. In addition, the sparse and robust versions of the probabilistic CCA model are introduced by [47], [48]. In our future work, the models and algorithms of tensor PLS and CCA under the Bayesian framework will be considered.

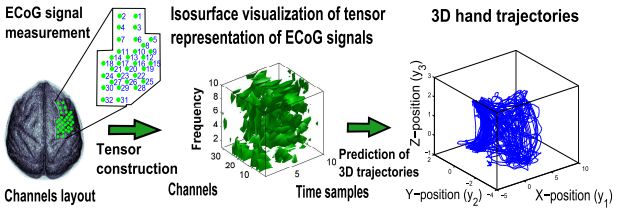


Fig. 4. The scheme for decoding of 3D hand movement trajectories from ECoG signals.

6 Experimental Results

6.1 Decoding of ECoG signals

ECoG-based decoding of 3D hand trajectories was demonstrated by means of classical PLS regression [27]. For the same datasets[1], in this study, several methods were applied for the prediction of limb movement trajectories in a 3D space based on ECoG signals recorded from monkey brains. The overall scheme of ECoG decoding is illustrated in Fig. 4. Specifically, 32 channels of ECoG signals were preprocessed by a band-pass filter from 0.1 to 600Hz and a spatial filter by common average reference. Motion marker locations were down-sampled to 20Hz. In order to extract features related to the 3D trajectory from ECoG signals, the Morlet wavelet transformation at 10 different center frequencies (10-150Hz, arranged in a logarithmic scale) was used to obtain the time-frequency representation. For each sample point of 3D trajectories, the most recent 1 second ECoG signals were transformed to time-frequency domains by means of the wavelet transformation. Finally, a third-order tensor of ECoG features $\underline{\mathbf{X}}$ (time samples $\times$ channels $\times$ time-frequency) was formed as an predictors. The movements of a monkey were captured by an optical motion capture system with reflective markers affixed to the left shoulder, elbows, wrists and hand, thus the responses were represented as a 3rd-order tensor $\underline{\mathbf{Y}}$ (i.e., samples $\times$ 3D positions $\times$ markers).

The advantage of HOPLS was better physical interpretation of the model. To investigate how the spatial, spectral, and temporal structure of ECoG data were used to create the regression model, loading vectors can be regarded as a subspace basis in spatial and time-frequency domains and latent variables were viewed as the coefficients. Fig. 5(A)(B) demonstrate spatial loadings $\mathbf{P}$ and time-frequency loadings $\mathbf{Q}$ employed in the decoding models for predicting hand trajectories. This way, we obtained a specific spatial distribution for each latent vector, which was valuable for investigating channel positions related to specific behaviors. With regard to time-frequency loadings, the β- and γ-band activities were most significant for encoding of movements; the duration of β-band was longer than γ-band. These findings also demonstrated that a high gamma band activity in the premotor cortex is associated with movement preparation, initiation and maintenance [50], illustrating

1. The datasets are freely available from neurotycho.org [49].

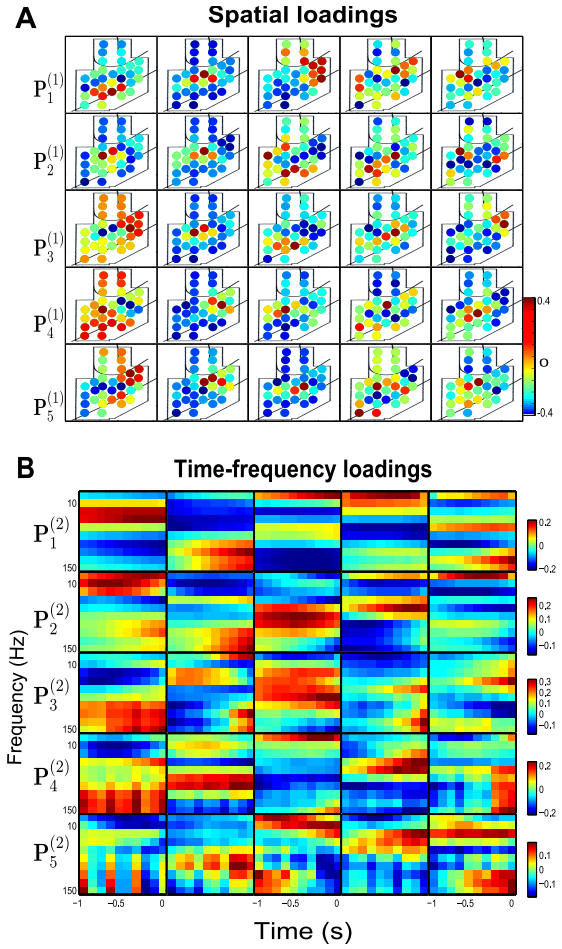
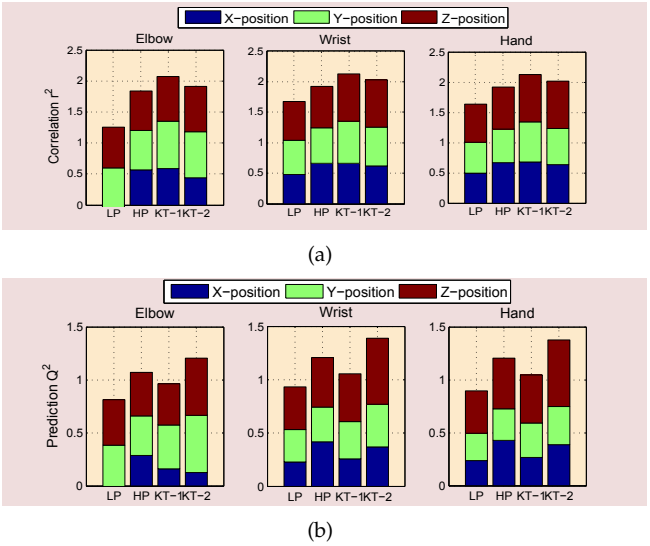


Fig. 5. Visualization of HOPLS model for $\underline{\mathbf{X}}$ decomposition. (A) Spatial loadings $\mathbf{P}_r^{(1)}$ corresponding to the first 5 latent vectors. Each row shows 5 significant loading vectors. Likewise, (B) depicts time-frequency loadings $\mathbf{P}_r^{(2)}$, with β and γ-band exhibiting significant contribution.



(a)

(b)

Fig. 6. The prediction performance for 3D movement trajectories recorded from Elbow, Wrist and Hand using four regression models including Linear PLS (LP), HOPLS (HP), KTPLS with Chordal distance based kernel (KT-1) and KTPLS with KL divergence based kernel (KT-2). The correlation coefficients r^2 between prediction and real data shown in (a) indicates that the best performance is obtained by TK-1 while evaluation of $Q^2 = 1 - \|\hat{\mathbf{y}} - \mathbf{y}\|^2 / \|\mathbf{y}\|^2$ showed in (b) indicates that TK-2 outperforms the other methods.

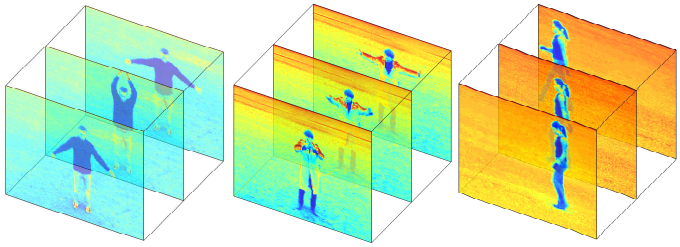


Fig. 7. Three examples of video sequences in tensor form for H-W, H-C and walking actions.

the effectiveness of HOPLS in interpreting the neurophysiological principles of movements encoding.

The dataset is divided into training set (10 minutes) and test set (5 minutes) and the selection of tuning parameters, such as number of latent components for HOPLS, PLS and kernel parameters for KTPLS, is performed by cross-validation on the training data. The predictive performances for the test set are shown in Fig. 6 demonstrating the superiority of KTPLS over linear PLS and HOPLS.

6.2 Video classification using KTCCA

Human action recognition in videos is of high interest for a variety of applications such as video surveillance, human-computer interface and video retrieval, where the most competing methods are based on motion estimation [51], local space-time interest points and visual code words [52], [53], [54], multiple classifiers [55], [56], sparse representation [57] and multiway tensor methods [40], [58]. Tensor representation enables us to directly analyze 3D video volume and encode global space-time structure information.

The effectiveness of KTCCA is demonstrated by video classifications on the KTH human action database[2] that contains six types (walking (W), running (R), jogging (J), boxing (B), hand-waving (H-W), and hand-clapping (H-C)) of human actions performed by 25 persons in four different scenarios (outdoors, outdoors with scale variation, outdoors with different clothes, and indoors). There are 600 video sequences in the dataset and three examples of video sequences represented as tensors are shown in Fig. 7. The space-time alignment on the human action is performed manually, then all video sequences are rescaled to 20×20×32. The dataset is divided with respect to the subjects into a training set (16 persons) and

2. http://www.nada.kth.se/cvap/actions/

TABLE 1
Confusion matrix for human action classification. The last column represents individual accuracy for each class.

	Walk	Run	Jog	Box	H-C	H-W	Acc.
Walk	36	0	0	0	0	0	100%
Run	0	35	1	0	0	0	97%
Jog	0	0	36	0	0	0	100%
Box	0	0	0	36	0	0	100%
H-C	0	0	0	0	36	0	100%
H-W	0	0	0	0	3	33	92%

a test set (9 persons). Each data example i.e., a video sequence, is represented by a 3rd-order tensor, and KTCCA is performed to find the shared latent space between training data and the corresponding class membership. The test data are then projected onto the latent space by model parameters learned from the training data, to obtain the discriminative components. Fig. 8 shows the test data in two-dimensional latent space, and six classes are clearly separated in the latent space indicating that KTCCA is able to capture the discriminative components very well. A simple k-nearest neighbor classifier (k-NN) is applied on lower-dimensional features for action classification and the confusion matrices on test set are shown in Table 1, in which rows correspond to the ground truth, and columns correspond to the classification results.

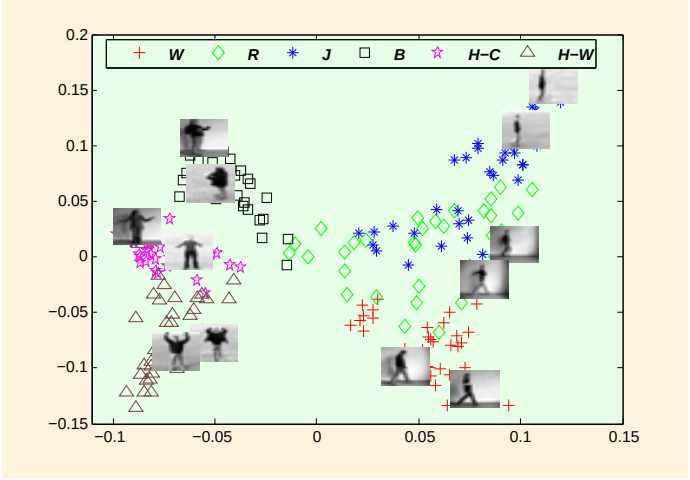


Fig. 8. Visualization of test dataset in two-dimensional KTCCA latent space. Observe that the first two components obtained from KTCCA are discriminative for action classification.

In addition, the leave-one-out performance is evaluated for comparison with the state-of-the-art methods on the KTH dataset. As shown in Table 2, KTCCA achieves the highest overall classification accuracy followed by product manifold (PM) [58], tensor CCA (TCCA) [40] and boosted exemplar learning (BEL) [56].

Acknowledgments

The work was supported in part by the National Natural Science Foundation of China (Grant No. 61202155) and JSPS KAKENHI (Grant No. 24700154). The work of second author was supported by the National Natural Science Foundation of China (Grant No. 91120305, 61272251).

7 Conclusions

In this paper, we reviewed PLS/CCA related methods including linear, multilinear, and nonlinear variants, especially focusing on tensor-based approaches. In addition, some recent advances about kernelization of tensor-based models are also discussed with supported experimental results. These methods may have some advantages over the traditional linear/nonlinear methods using matrix algebra in terms of the multidimensional structured data analysis. Several illustrative examples were provided to compare the performance with the state-of-the-art of methods that are relevant to this topic.

References

[1] Q. Zhao, C. Caiafa, D. Mandic, Z. Chao, Y. Nagasaka, N. Fujii, L. Zhang, and A. Cichocki, "Higher-order partial least squares (HOPLS): A generalized multi-linear regression method," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 35, no. 7, pp. 1660–1673, 2013.

[2] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis*, 3rd ed. New York: John Wiley & Sons, 2001.

[3] C. Dhanjal, S. Gunn, and J. Shawe-Taylor, "Efficient sparse kernel feature extraction based on partial least squares," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 8, pp. 1347–1361, 2009.

[4] J. Trygg and S. Wold, "Orthogonal projections to latent structures (O-PLS)," *Journal of Chemometrics*, vol. 16, no. 3, pp. 119–128, 2002.

[5] R. Ergon, "PLS score-loading correspondence and a bi-orthogonal factorization," *Journal of Chemometrics*, vol. 16, no. 7, pp. 368–373, 2002.

[6] E. Helland Hans *et al.*, "Recursive algorithm for partial least squares regression," *Chemometrics and Intelligent Laboratory Systems*, vol. 14, no. 1-3, pp. 129–137, 1992.

[7] S. Vijayakumar and S. Schaal, "Locally weighted projection regression: An O (n) algorithm for incremental real time learning in high dimensional space," in *Proceedings of the Seventeenth International Conference on Machine Learning*, vol. 1, 2000, pp. 288–293.

[8] R. Rosipal and L. Trejo, "Kernel partial least squares regression in reproducing kernel Hilbert space," *The Journal of Machine Learning Research*, vol. 2, p. 123, 2002.

[9] J. Ham, D. Lee, S. Mika, and B. Schölkopf, "A kernel view of the dimensionality reduction of manifolds," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, pp. 47–54.

[10] A. Hoerl and R. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, pp. 55–67, 1970.

[11] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[12] L. R. Tucker, "Implications of factor analysis of three-way matrices for measurement of change," in *Problems in Measuring Change*, C. W. Harris, Ed. University of Wisconsin Press, 1963, pp. 122–137.

[13] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an explanatory multimodal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, 1970.

[14] 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, pp. 283–319, 1970.

TABLE 2
Comparisons of leave-one-out classification accuracy on the KTH data set [31].

KTCCA	TCCA [40]	PM [58]	pLSA/ISM [59]	WX/SVM [59]	BEL [56]	MIL [51]	pLSA/LDA [52]	LF/SVM [53]
97.83%	95.33%	97%	83.92%	91.6%	95.33%	87.7%	83.33%	71.72%

[15] L. De Lathauwer, B. De Moor, and J. Vandewalle, "On the Best Rank-1 and Rank-(R1, R2,..., RN) Approximation of Higher-Order Tensors," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000.

[16] L.-H. Lim and V. D. Silva, "Tensor rank and the ill-posedness of the best low-rank approximation problem," *SIAM Journal of Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1084–1127, 2008.

[17] T. Kolda and B. Bader, "Tensor Decompositions and Applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[18] A. Smilde, R. Bro, and P. Geladi, *Multi-way analysis with applications in the chemical sciences*. Wiley, 2004.

[19] H. Wold, "Soft modelling by latent variables: The non-linear iterative partial least squares (NIPALS) approach," *Perspectives in Probability and Statistics*, pp. 117–142, 1975.

[20] ——, "Soft modeling: the basic design and some extensions," *Systems Under Indirect Observation*, vol. 2, pp. 1–53, 1982.

[21] S. Wold, M. Sjostroma, and L. Erikssonb, "PLS-regression: a basic tool of chemometrics," *Chemometrics and Intelligent Laboratory Systems*, vol. 58, pp. 109–130, 2001.

[22] S. Wold, A. Ruhe, H. Wold, and W. Dunn III, "The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverses," *SIAM Journal on Scientific and Statistical Computing*, vol. 5, p. 735, 1984.

[23] B. Kowalski, R. Gerlach, and H. Wold, "Systems under indirect observation," *Chemical Systems under Indirect Observation*, pp. 191–209, 1982.

[24] A. McIntosh and N. Lobaugh, "Partial least squares analysis of neuroimaging data: applications and advances," *Neuroimage*, vol. 23, pp. S250–S263, 2004.

[25] A. McIntosh, W. Chau, and A. Protzner, "Spatiotemporal analysis of event-related fMRI data using partial least squares," *Neuroimage*, vol. 23, no. 2, pp. 764–775, 2004.

[26] N. Kovacevic and A. McIntosh, "Groupwise independent component decomposition of EEG data and partial least square analysis," *NeuroImage*, vol. 35, no. 3, pp. 1103–1112, 2007.

[27] Z. Chao, Y. Nagasaka, and N. Fujii, "Long-term asynchronous decoding of arm motion using electrocorticographic signals in monkeys," *Frontiers in Neuroengineering*, vol. 3, no. 3, 2010.

[28] L. Trejo, R. Rosipal, and B. Matthews, "Brain-computer interfaces for 1-D and 2-D cursor control: designs using volitional control of the EEG spectrum or steady-state visual evoked potentials," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 14, no. 2, pp. 225–229, 2006.

[29] R. Bro, "Multiway calibration. Multilinear PLS," *Journal of Chemometrics*, vol. 10, no. 1, pp. 47–61, 1996.

[30] Q. Zhao, C. F. Caiafa, D. P. Mandic, L. Zhang, T. Ball, A. Schulze-bonhage, and A. S. Cichocki, "Multilinear subspace regression: An orthogonal tensor decomposition approach," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., 2011, pp. 1269–1277.

[31] 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*, vol. 30, pp. 137–148, 2013.

[32] M. Signoretto, L. De Lathauwer, and J. A. Suykens, "A kernel-based framework to tensorial data analysis," *Neural networks*, vol. 24, no. 8, pp. 861–874, 2011.

[33] M. Signoretto, E. Olivetti, L. De Lathauwer, and J. Suykens, "Classification of multichannel signals with cumulant-based kernels," *IEEE Transactions on Signal Processing*, vol. 60, no. 5, pp. 2304–2314, 2012.

[34] P. Moreno, P. Ho, and N. Vasconcelos, "A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications," *Advances in Neural Information Processing Systems*, vol. 16, pp. 1385–1393, 2003.

[35] A. Chan, N. Vasconcelos, and P. Moreno, "A family of probabilistic kernels based on information divergence," *Univ. California, San Diego, CA, Tech. Rep. SVCL-TR-2004-1*, 2004.

[36] D. Endres and J. Schindelin, "A new metric for probability distributions," *IEEE Transactions on Information Theory*, vol. 49, no. 7, pp. 1858–1860, 2003.

[37] L. Sun, S. Ji, and J. Ye, "Canonical correlation analysis for multilabel classification: A least-squares formulation, extensions, and analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 194–200, 2011.

[38] K. Fukumizu, F. Bach, and A. Gretton, "Statistical convergence of kernel CCA," *Advances in Neural Information Processing Systems (NIPS)*, 2005.

[39] D. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Computation*, vol. 16, no. 12, pp. 2639–2664, 2004.

[40] T. Kim and R. Cipolla, "Canonical correlation analysis of video volume tensors for action categorization and detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 8, pp. 1415–1428, 2009.

[41] A. Krishnan, L. J. Williams, A. R. McIntosh, and H. Abdi, "Partial least squares (pls) methods for neuroimaging: a tutorial and review," *Neuroimage*, vol. 56, no. 2, pp. 455–475, 2011.

[42] J. Rupnik and J. Shawe-Taylor, "Multi-view canonical correlation analysis," in *Conference on Data Mining and Data Warehouses (SiKDD 2010)*, 2010, pp. 1–4.

[43] T.-Y. Liu, L. Trinchera, A. Tenenhaus, W. B. Dennis, A. Hero *et al.*, "Globally sparse PLS regression," in *Proceedings of the 7th International Conference on Partial Least Squares and Related Methods*, 2012.

[44] C. Fyfe and G. Leen, "Stochastic processes for canonical correlation analysis." in *ESANN*, 2006, pp. 245–250.

[45] S. Li, J. Gao, J. O. Nyagilo, and D. P. Dave, "Probabilistic partial least square regression: a robust model for quantitative analysis of raman spectroscopy data," in *Bioinformatics and Biomedicine (BIBM), 2011 IEEE International Conference on*. IEEE, 2011, pp. 526–531.

[46] A. Klami, S. Virtanen, and S. Kaski, "Bayesian canonical correlation analysis," *Journal of Machine Learning Research*, vol. 14, pp. 965–1003, 2013.

[47] C. Archambeau and F. R. Bach, "Sparse probabilistic projections," in *Advances in Neural Information Processing Systems*, 2008, pp. 73–80.

[48] J. Viinikanoja, A. Klami, and S. Kaski, "Variational bayesian mixture of robust cca models," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2010, pp. 370–385.

[49] Y. Nagasaka, K. Shimoda, and N. Fujii, "Multidimensional recording (MDR) and data sharing: An ecological open research and educational platform for neuroscience," *PLoS ONE*, vol. 6, no. 7, p. e22561, 2011.

[50] J. Rickert, S. de Oliveira, E. Vaadia, A. Aertsen, S. Rotter, and C. Mehring, "Encoding of movement direction in different frequency ranges of motor cortical local field potentials," *The Journal of Neuroscience*, vol. 25, no. 39, pp. 8815–8824, 2005.

[51] S. Ali and M. Shah, "Human action recognition in videos using kinematic features and multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 2, pp. 288–303, 2010.

[52] J. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *International Journal of Computer Vision*, vol. 79, no. 3, pp. 299–318, 2008.

[53] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proceedings of the 17th International Conference on Pattern Recognition, ICPR*, vol. 3. IEEE, 2004, pp. 32–36.

[54] M. Holte, B. Chakraborty, J. Gonzalez, and T. Moeslund, "A local 3-D motion descriptor for multi-view human action recognition from 4-D spatio-temporal interest points," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 5, pp. 553–565, 2012.

[55] Y. Song, Y. Zheng, S. Tang, X. Zhou, Y. Zhang, S. Lin, and T. Chua, "Localized multiple kernel learning for realistic human action

recognition in videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 9, pp. 1193–1202, 2011.

[56] T. Zhang, J. Liu, S. Liu, C. Xu, and H. Lu, "Boosted exemplar learning for action recognition and annotation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 7, pp. 853–866, 2011.

[57] T. Guha and R. Ward, "Learning sparse representations for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 8, pp. 1576–1588, 2012.

[58] Y. Lui, J. Beveridge, and M. Kirby, "Action classification on product manifolds," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 833–839.

[59] S. Wong, T. Kim, and R. Cipolla, "Learning motion categories using both semantic and structural information," in *Computer Vision and Pattern Recognition. CVPR'07*. IEEE, 2007, pp. 1–6.