KERNEL-BASED TENSOR PARTIAL LEAST SQUARES FOR RECONSTRUCTION OF LIMB MOVEMENTS

Qibin Zhao[1]*, Guoxu Zhou*[1]*, Tulay Adali*[2]*, Liqing Zhang*[3]*, Andrzej Cichocki*[1]

[1] RIKEN Brain Science Institute, Japan
[2] University of Maryland Baltimore County, MD, USA
[3] Shanghai Jiao Tong University, China

ABSTRACT

We present a new supervised tensor regression method based on multi-way array decompositions and kernel machines. The main issue in the development of a kernel-based framework for tensorial data is that the kernel functions have to be defined on tensor-valued input, which we define based on multi-mode product kernels and probabilistic generative models. This strategy enables a range of machine learning methods to take into account the underlying multilinear structure during the learning process. Based on the defined kernels for tensorial data, we develop a kernel-based tensor partial least squares approach for regression. The effectiveness of our method is demonstrated by a real-world application, i.e., the reconstruction of 3D movement trajectories from electrocorticography signals recorded from a monkey brain.

Index Terms— Tensor, Kernels, Partial Least Squares, ECoG, Motion trajectory

1. INTRODUCTION

Tensors (also called multiway arrays) are generalization of vectors and matrices to higher dimensions with corresponding multilinear operators. The theory and algorithms of tensor decomposition (or factorization techniques) have been extensively investigated in the past decade, see e.g. [1, 2], and successfully applied to problems in unsupervised learning and exploratory data analysis. Tensor decompositions typically enable us to capture the structure of the data, which is usually available as *a priori* information on the data nature, and hence might provide advantages over matrix factorizations. Machine learning methods have been increasingly used for the analysis of neural/medical data, such as functional magnetic resonance (fMRI), electrocorticography (ECoG) and electroencephalography (EEG) data, and have emphasized the need to take the structural information of original data into account. To this end, tensor representation is natural and efficient for such multiway structural data, meanwhile its corresponding learning techniques should explicitly exploit the *a priori* information of data structure and capture the underlying multiway relations, resulting in useful decompositions with good generalization ability. Kernel methods, on the other hand, have proven successful in many applications, providing an efficient way to solve nonlinear problems [3] by mapping input data space into a high dimensional feature space, where the problem becomes linearly solvable. Recent research has addressed the incorporation of kernel concept into tensor decompositions [4], which aims to bring together the desirable properties of kernel methods and tensor decompositions for significant performance gain when the data are structured and nonlinear dependencies do exist.

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 [5, 6, 7]. In order to predict response variables $\mathbf{Y}$ from independent variables $\mathbf{X}$, PLS finds a set of latent variables (also called latent vectors, score vectors or components) by projecting both $\mathbf{X}$ and $\mathbf{Y}$ onto a new subspace, while at the same time maximizing the pairwise covariance between the latent variables of $\mathbf{X}$ and $\mathbf{Y}$. There are many variations of the PLS model including the orthogonal projection on latent structures (O-PLS) [8], biorthogonal PLS (BPLS) [9], recursive partial least squares (RPLS) [10], and nonlinear PLS [11]. An extension of PLS to tensor data is N-way PLS (N-PLS) that decomposes the independent and dependent tensor into rank-one tensors, subject to maximum pairwise covariance of the latent vectors, resulting in enhanced stability, resilience to noise, and intuitive interpretation of the results [12, 13]. The tensor decomposition used within N-PLS is Canonical Decomposition/Parallel Factor Analysis (CANDECOMP/PARAFAC or CP), which makes N-PLS inherit both the advantages and limitations of CP. In [14], the generalized mutilinear regression model, called Higher-Order Partial Least Squares (HOPLS), is introduced such that a tensor $\boldsymbol{\mathcal{Y}}$ can be predicted from a tensor $\boldsymbol{\mathcal{X}}$ by projection onto a low-dimensional common latent subspace. Owing to the better fitness ability of the orthogonal Tucker model as compared to CP and the flexibility of the block Tucker model [15], HOPLS is demonstrated to be a promising multilinear subspace regression framework that provides an optimal tradeoff between fitness and model complexity and enhanced predictive ability in general.

In this study, we introduce a novel supervised learning method, called kernel tensor partial least squares (KTPLS), for tensor regression problems using kernel machines. The key issue in developing a kernel-based framework for tensorial data is the definition of the kernel function on the tensor-valued input, which we define based on multi-mode product kernels and probabilistic generative models [16]. In addition, we apply KTPLS to electrocorticography (ECoG) signals recorded from a monkey for reconstruction of 3D movement trajectories.

2. BACKGROUND AND NOTATION

Nth-order tensors (*multi-way arrays*) are denoted by calligraphy letters $\mathcal{X}$, matrices (*two-way arrays)* by boldface capital letters $\mathbf{X}$, and vectors by boldface lower-case letters $\mathbf{x}$. 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 $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ by $x_{i_1 i_2 \ldots i_N}$ or $(\mathcal{X})_{i_1 i_2 \ldots i_N}$. 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 $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times I_1 \cdots I_{n-1} I_{n+1} \cdots I_N}$, while the vectorization of a tensor is denoted as vec$(\mathcal{X})$. The *n-mode product* of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_n \times \cdots \times I_N}$ and matrix $\mathbf{A} \in \mathbb{R}^{J_n \times I_n}$is denoted by $\mathcal{Y} = \mathcal{X} \times_n \mathbf{A} \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N}$ and is defined as:

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

The *inner product of two tensors* $\mathcal{X}, \mathcal{X}' \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is defined by $\langle \mathcal{X}, \mathcal{X}' \rangle = \sum_{i_1 i_2 \ldots i_N} x_{i_1 i_2 \ldots i_N} x'_{i_1 i_2 \ldots i_N}$, and the squared Frobenius norm by $\|\mathcal{X}\|_F^2 = \langle \mathcal{X}, \mathcal{X} \rangle$.

The two most commonly used decompositions are the *Tucker* model and *CANDECOMP/PARAFAC* (CP) model, both of which can be regarded as higher-order generalizations of the matrix singular value decomposition (SVD). Let $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ denote an Nth-order tensor, then *Tucker* model is defined as follows:

$$\mathcal{X} = \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \cdots \times_N \mathbf{U}^{(N)} \qquad (2)$$

where $\mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_N}$ denotes the *core tensor* and $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times R_n}$ denotes the mode-n *factor matrix*. When all factor matrices $\{\mathbf{U}^{(n)}\}_{n=1}^N$ are orthonormal and the core tensor is all-orthogonal this model is called HOSVD [17] (see Fig. 1). When all the factor matrices have the same number of components, and the core tensor is super-diagonal, Tucker model simplifies to *CP* model. In general, *CP* model is considered to be a multilinear low-rank approximation while *Tucker* model is regarded as a multilinear subspace approximation.

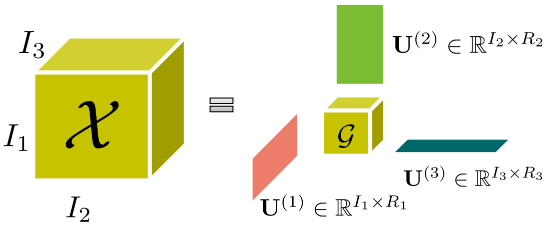
I_3
I_1
$\mathcal{X}$
I_2
$\equiv$
$\mathbf{U}^{(2)} \in \mathbb{R}^{I_2 \times R_2}$
$\mathcal{G}$
$\mathbf{U}^{(3)} \in \mathbb{R}^{I_3 \times R_3}$
$\mathbf{U}^{(1)} \in \mathbb{R}^{I_1 \times R_1}$

Fig. 1. The illustration of HOSVD scheme whose objective is to optimize orthonormal factor matrices $\{\mathbf{U}^{(n)}, n = 1, 2, 3\}$ and all-orthogonal core tensor $\mathcal{G}$.

3. KERNEL-BASED TENSOR PLS REGRESSION

Partial least squares (PLS) models two datasets by a generative process driven by a small number of latent variables. PLS regression actually consists of two steps: dimension reduction and linear regression. For N pairs of tensor observations $\{(\mathcal{X}^{(n)}, \mathcal{Y}^{(n)})\}_{n=1}^N$ where $\mathcal{X}^{(n)} \in \mathbb{R}^{I_1 \times \cdots \times I_L}$ is an Lth-order independent tensor and $\mathcal{Y}^{(n)} \in \mathbb{R}^{J_1 \times \cdots \times J_M}$ is an Mth-order dependent tensor, which can be concatenated as an $(L+1)$th-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{N \times I_1 \times \cdots \times I_L}$ and $(M+1)$th-order tensor $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{N \times J_1 \times \cdots \times J_M}$. The objective of HOPLS [18, 14] is to find the optimal tensor decompositions for $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$, yielding the maximum correlated latent vectors $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_R]$ and $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_R]$ from independent and dependent data respectively, i.e.

$$\begin{aligned} \boldsymbol{\mathcal{X}} &= \mathcal{G}_{\mathcal{X}} \times_1 \mathbf{T} \times_2 \mathbf{P}^{(1)} \cdots \times_{L+1} \mathbf{P}^{(L)} + \mathcal{E}_{\mathcal{X}}, \\ \boldsymbol{\mathcal{Y}} &= \mathcal{G}_{\mathcal{Y}} \times_1 \mathbf{U} \times_2 \mathbf{Q}^{(1)} \cdots \times_{M+1} \mathbf{Q}^{(M)} + \mathcal{E}_{\mathcal{Y}}, \end{aligned} \qquad (3)$$

where $\left\{\mathbf{P}^{(l)}, \mathbf{Q}^{(m)}\right\}$ denote factor matrices on specific modes and $\{\mathcal{G}_{\mathcal{X}}, \mathcal{G}_{\mathcal{Y}}\}$ denote block structured core tensors. These parameters are learned sequentially from training data and then used to make a prediction by projecting a data example onto latent space and then making regression from the corresponding latent variables.

Let us now consider how the kernel-based approach can be exploited to obtain kernel-based tensor PLS (KTPLS). We shall assume the tensorial observations are mapped into the Hilbert space $\mathcal{F}$ by

$$\phi : \quad \mathcal{X}^{(n)} \rightarrow \phi\left(\mathcal{X}^{(n)}\right) \in \mathbb{R}^{F_1 \times \cdots \times F_L}. \qquad (4)$$

Note that the projected tensor $\phi\left(\mathcal{X}^{(n)}\right)$ has the same order L with $\mathcal{X}^{(n)}$, but the mode-l dimension is F_l or even an infinite dimension depending on the nonlinear function $\phi(\cdot)$. For dependent data $\mathcal{Y}$, we can choose to either apply nonlinear mapping or not. For simplicity, we denote $\phi(\boldsymbol{\mathcal{X}})$ by tensor $\boldsymbol{\Phi}$ and $\phi(\boldsymbol{\mathcal{Y}})$ by tensor $\boldsymbol{\Psi}$. The KTPLS model is formulated as

$$\begin{aligned} \boldsymbol{\Phi} &= \widetilde{\mathcal{G}}_{\mathcal{X}} \times_1 \mathbf{T} + \mathcal{E}_{\mathcal{X}}, \\ \boldsymbol{\Psi} &= \widetilde{\mathcal{G}}_{\mathcal{Y}} \times_1 \mathbf{U} + \mathcal{E}_{\mathcal{Y}}, \\ \mathbf{U} &= \mathbf{T}\mathbf{D} + \mathbf{E}_U, \end{aligned} \qquad (5)$$

where $\mathbf{D}$ is a diagonal matrix denoting *inner relation* between latent vectors $\mathbf{t}_r$ and $\mathbf{u}_r$, $\widetilde{\mathcal{G}}_{\mathcal{X}}, \widetilde{\mathcal{G}}_{\mathcal{Y}}$ are core tensors absorbing several factor matrices, both of which cannot be computed explicitly. Note that $\widetilde{\mathcal{G}}_{\mathcal{X}}$ can be obtained as $\mathbf{\Phi} \times_1 \mathbf{T}^T$ when $\mathbf{T}$ is orthogonal, implying that it can be represented as a linear combination of $\{\phi(\mathcal{X}^{(n)})\}$. Hence, instead of computing high-dimensional core tensors $\widetilde{\mathcal{G}}_{\mathcal{X}}, \widetilde{\mathcal{G}}_{\mathcal{Y}}$, we only need to explicitly estimate $\mathbf{T}, \mathbf{U}$ by solving an optimization problem sequentially for any rth component, which is expressed by

$$\max_{\{\mathbf{w}_r, \mathbf{v}_r\}} \quad [\mathrm{cov}(\mathbf{t}_r, \mathbf{u}_r)]^2, \tag{6}$$

$$\text{s. t.} \quad \mathbf{t}_r = \mathbf{\Phi}_{(1)}\mathbf{w}_r, \quad \mathbf{u}_r = \mathbf{\Psi}_{(1)}\mathbf{v}_r \quad \text{and} \quad r = 1, \ldots, R.$$

This can be solved by a kernelized version of the eigenvalue problem, i.e., the optimal $\mathbf{t}_r, \mathbf{u}_r$ can be obtained by solving $\mathbf{\Phi}_{(1)}\mathbf{\Phi}_{(1)}^T\mathbf{\Psi}_{(1)}\mathbf{\Psi}_{(1)}^T\mathbf{t}_r = \lambda\mathbf{t}_r$ and $\mathbf{u}_r = \mathbf{\Psi}_{(1)}\mathbf{\Psi}_{(1)}^T\mathbf{t}_r$. Note that $\mathbf{\Phi}_{(1)}\mathbf{\Phi}_{(1)}^T$ contains only the inner products between vectorized input data tensors, which can be regarded as an $N \times N$ Gram matrix $\mathbf{K}_{\mathcal{X}}$. Similarly, $\mathbf{\Psi}_{(1)}\mathbf{\Psi}_{(1)}^T$ is represented as $\mathbf{K}_{\mathcal{Y}}$. Thus, we have $\mathbf{K}_{\mathcal{X}}\mathbf{K}_{\mathcal{Y}}\mathbf{t}_r = \lambda\mathbf{t}_r$ and $\mathbf{u}_r = \mathbf{K}_{\mathcal{Y}}\mathbf{t}_r$. In order to take the structure information into account, the kernel matrices should be computed using the specially defined kernel functions for tensorial data, which will be discussed in the next section, i.e., $(\mathbf{K}_{\mathcal{X}})_{nn'} = k\left(\mathcal{X}^{(n)}, \mathcal{X}^{(n')}\right)$ and $(\mathbf{K}_{\mathcal{Y}})_{nn'} = k\left(\mathcal{Y}^{(n)}, \mathcal{Y}^{(n')}\right)$. Finally, the prediction of new data point $\mathcal{X}^*$ can be achieved by

$$\mathbf{y}^{*T} = \mathbf{k}^{*T}\mathbf{U}(\mathbf{T}^T\mathbf{K}_{\mathcal{X}}\mathbf{U})^{-1}\mathbf{T}^T\mathbf{Y}_{(1)}, \tag{7}$$

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

Let us examine the predictive function given in (7). Observe that the prediction is a linear combination of N observations $\{\mathcal{Y}^{(n)}\}$ with the coefficients $\mathbf{k}^{*T}\mathbf{U}(\mathbf{T}^T\mathbf{K}_{\mathcal{X}}\mathbf{U})^{-1}\mathbf{T}^T$. 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.,

$$y_j^* = \sum_{n=1}^{N} \alpha_n k\left(\mathcal{X}^*, \mathcal{X}^{(n)}\right), \tag{8}$$

where $\alpha_n = (\mathbf{U}(\mathbf{T}^T\mathbf{K}_{\mathcal{X}}\mathbf{U})^{-1}\mathbf{T}^T\mathbf{Y}_{(1)})_{nj}$. The third interpretation is that the prediction can be represented as a linear regression against low dimensional latent variables $\mathbf{t}^* \in \mathbb{R}^R$, obtained by projecting $\mathcal{X}^*$ onto the latent space, with regression coefficient $\mathbf{C} = \mathbf{T}^T\mathbf{Y}_{(1)}$, i.e., $y_j^* = \mathbf{t}^{*T}\mathbf{c}_j$. Note that an interesting property is that although we apply nonlinear mapping $\phi(\cdot)$ for data $\boldsymbol{\mathcal{Y}}$, it is still possible to predict $\boldsymbol{\mathcal{Y}}$ in the original input space, without solving the preimage problem [19] due to the fact that two kernel matrices of $\mathbf{K}_{\mathcal{X}}, \mathbf{K}_{\mathcal{Y}}$ only affect the solution of the latent variables.

3.1. Kernel function for tensorial data

The kernels are considered as defining a topology implying the *a priori* knowledge about invariance in the input space. Although many kernels have been designed for a number of structured objects, few approaches exploit the structure of tensorial representations. Recently, M. Signoretto et al. proposed a tensorial kernel exploiting algebraic geometry of spaces of tensors and a similarity measure between the different subspaces spanned by higher-order tensors [20]. There are a number of valid reproducing kernels toward a straightforward generalization to Mth-order tensors, such as the kernel functions $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ given as

$$\text{Linear kernel:} \quad k(\mathcal{X}, \mathcal{X}') = \langle \mathrm{vec}(\mathcal{X}), \mathrm{vec}(\mathcal{X}') \rangle,$$

$$\text{Gaussian kernel:} \quad k(\mathcal{X}, \mathcal{X}') = \exp\left(-\frac{1}{2\sigma^2}\|\mathcal{X} - \mathcal{X}'\|_F^2\right).$$

In order to define the similarity measure that takes advantage of the multilinear algebraic structure of input tensors, the general product kernels can be defined by M factor kernels, which is valid if the factor kernels are positive semi-definite, denoted by $k(\mathcal{X}, \mathcal{X}') = \prod_{m=1}^{M} k\left(\mathbf{X}_{(m)}, \mathbf{X'}_{(m)}\right)$, where each factor kernel represents a similarity measure between two matrices obtained by mode-m unfoldings of two tensor examples. One possibility of similarity measure between matrices is Chordal distance [20] (projection Frobenius norm) on the Grassmannian manifolds. Let $\mathcal{X}$ denote an Mth-order tensor example, SVD can be applied on mode-m unfoldings as $\mathbf{X}_{(m)} = \mathbf{U}_{\mathbf{X}}^{(m)}\mathbf{\Sigma}_{\mathbf{X}}^{(m)}\mathbf{V}_{\mathbf{X}}^{(m)T}$, then the Chordal distance can be computed based on the right singular vectors $\mathbf{V}_{\mathbf{X}}^{(m)}$, thus we have

$$k(\mathcal{X}, \mathcal{X}') = \prod_{m=1}^{M} \exp\left(-\frac{1}{2\sigma^2}\left\|\mathbf{V}_{\mathbf{X}}^{(m)}\mathbf{V}_{\mathbf{X}}^{(m)T} - \mathbf{V}_{\mathbf{X}'}^{(m)}\mathbf{V}_{\mathbf{X}'}^{(m)T}\right\|_F\right). \tag{9}$$

This kernel provides us rotation and reflection invariance for elements on the Grassmann manifold, which is effective for video classification and recognition.

As kernels can be interpreted as measures of similarity, it is also possible to define kernels based on information divergences that are measures of dissimilarity between probability distributions, such as Fisher kernel and *Kullback-Leibler* kernel [21]. We propose a new probabilistic kernel for tensorial data based on the assumption that an Mth-order tensor can be considered as M generative models, and each model corresponds to a set of observations obtained by matricization of the tensor in specific mode. We assume $\mathcal{X}^{(n)}$ is generated individually by M models governed by parameters $\left\{\boldsymbol{\theta}_m^{(n)}\right\}_{m=1}^{M}$. Once the model parameters $\boldsymbol{\theta}_m$ have been estimated from mode-m matricization $\mathbf{X}_{(m)}$, we can define the kernel distance based on the symmetric *Kullback-Leibler (sKL)* divergence, given by

$$\begin{aligned} D\left(p(\mathbf{x}|\boldsymbol{\theta})||q(\mathbf{x}|\boldsymbol{\theta}')\right) = &\int_{-\infty}^{+\infty} p(\mathbf{x}|\boldsymbol{\theta})\log\left(\frac{p(\mathbf{x}|\boldsymbol{\theta})}{q(\mathbf{x}|\boldsymbol{\theta}')}\right)d\mathbf{x} \\ &+ \int_{-\infty}^{+\infty} q(\mathbf{x}|\boldsymbol{\theta}')\log\left(\frac{q(\mathbf{x}|\boldsymbol{\theta}')}{p(\mathbf{x}|\boldsymbol{\theta})}\right)d\mathbf{x}. \end{aligned} \tag{10}$$

In order to ensure that the kernel matrix is a positive definite matrix, we use exponential kernel function with the sKL divergence. Finally, the product kernel based on M sKL kernels is represented by

$$k(\mathcal{X}, \mathcal{X}') = \prod_{m=1}^{M} \exp\left(-\frac{D\left(p(\mathbf{x}|\boldsymbol{\theta}_m)||q(\mathbf{x}|\boldsymbol{\theta}'_m)\right)}{2\sigma_m^2} \right), \quad (11)$$

where $\{\sigma_m\}_{m=1}^{M}$ are kernel parameters corresponding to mode-m sKL kernel.

For simplicity, Gaussian model assumption can be employed with model parameters including a mean vector and a full covariance matrix, i.e., $\boldsymbol{\theta}_m = \{\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m\}$ that can be estimated by maximum likelihood from $\mathbf{X}_{(m)}$. The detailed algorithms of sKL divergences between two multivariate Gaussian distributions are given in[21, 22].

4. EXPERIMENTAL RESULTS

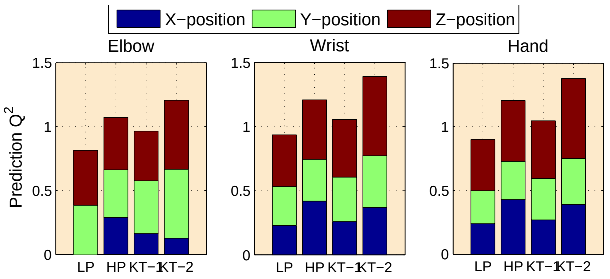

Fig. 2. The prediction performance for 3D movement trajectories recorded from elbow, wrist and hand using four regression models including linear PLS (LP), HOPLS (HP), KTPLS with Chordal distance-based kernel (KT-1) and KTPLS with sKL divergence-based kernel (KT-2). The performance of $Q^2 = 1 - \|\hat{\mathbf{y}} - \mathbf{y}\|^2 / \|\mathbf{y}\|^2$ indicates that TK-2 outperforms the other methods.

We apply KTPLS for decoding of 3D movement trajectories from ECoG signals recorded from monkey brain[1]. 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 dependent data was naturally represented as a 3rd-order tensor $\boldsymbol{\mathcal{Y}}$ (i.e., samples × 3D positions × markers). In order to represent the discriminative features, the ECoG signals are transformed to the time-frequency domains and are represented as a 4th-order tensor $\boldsymbol{\mathcal{X}}$ (i.e., epoch × channel × frequency × time) where each ECoG epoch $\mathcal{X}^{(n)}$ corresponds to one sample of movement data $\mathcal{Y}^{(n)}$. Since KTPLS enables us to create a regression model between two higher-order tensors, it is used to predict every sample of positions of monkey movements based on the most recent past 1-second ECoG epoch. The dataset is divided into training set (10 minutes) and test set (5 minutes) and the selection of tuning parameters, such as kernel parameters σ_m and number of latent vectors, is performed by cross-validation on the training data. The prediction performances for the test set are shown in Fig. 2, demonstrating the superiority of KTPLS over linear PLS and HOPLS. Fig. 3 illustrates the predicted 3D movement trajectories of hand using two different kernel functions defined in (11) and (9) for comparison, which demonstrates the advantages of the proposed generative probabilistic kernels based on sKL divergence with respect to predictive performance.

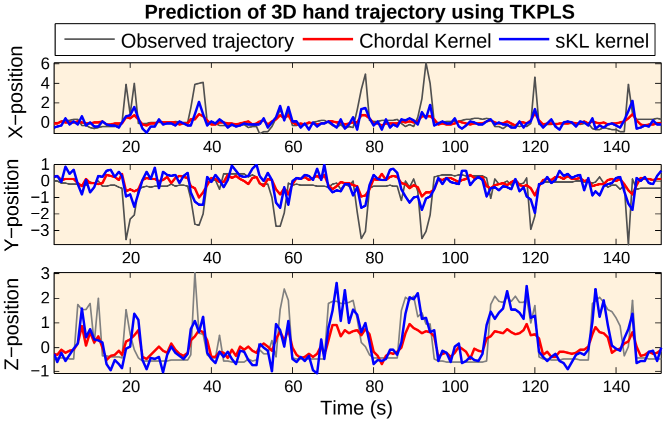

Fig. 3. The visualization of predicted and recorded trajectories of hand movements.

5. CONCLUSIONS

In this paper, we discussed PLS regression, extension to tensor (e.g., NPLS and HOPLS) and developed a kernel-based tensor regression approach based on specially defined kernel functions for tensorial data, which has been successfully applied for reconstruction of movement trajectories from brain signals. The experimental results demonstrats the effectiveness and advantages of the proposed method.

Acknowledgments

This work was supported by JSPS KAKENHI (Grant No. 24700154) and the National Natural Science Foundation of China (Grant No. 61202155, 90920014, 91120305).

[1] The datasets and more detailed description are freely available from `http://neurotycho.org`.

6. REFERENCES

[1] A. Cichocki, R. Zdunek, A. H. Phan, and S. I. Amari, *Nonnegative Matrix and Tensor Factorizations*, John Wiley & Sons, 2009.

[2] T.G. Kolda and B.W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[3] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004.

[4] M. Signoretto, L. De Lathauwer, and J.A.K. Suykens, "A kernel-based framework to tensorial data analysis," *Neural Networks*, 2011.

[5] 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.

[6] H. Abdi, "Partial least squares regression and projection on latent structure regression (PLS Regression)," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 1, pp. 97–106, 2010.

[7] R. Rosipal and N. Krämer, "Overview and recent advances in partial least squares," in *Subspace, Latent Structure and Feature Selection*, vol. 3940 of *Lecture Notes in Computer Science*, pp. 34–51. Springer, 2006.

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

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

[10] 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 (ICML 2000)*, 2000, vol. 1, pp. 288–293.

[11] R. Rosipal and L.J. Trejo, "Kernel partial least squares regression in reproducing kernel Hilbert space," *The Journal of Machine Learning Research*, vol. 2, pp. 97–123, 2002.

[12] E. Martinez-Montes, P.A. Valdés-Sosa, F. Miwakeichi, R.I. Goldman, and M.S. Cohen, "Concurrent EEG/fMRI analysis by multiway partial least squares," *NeuroImage*, vol. 22, no. 3, pp. 1023–1034, 2004.

[13] E. Acar, C.A. Bingol, H. Bingol, R. Bro, and B. Yener, "Seizure recognition on epilepsy feature tensor," in *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, 2007, pp. 4273–4276.

[14] Q. Zhao, C. F. Caiafa, D. P. Mandic, Z. Chao, Y. Nagasaka, N. Fujii, L. Zhang, and A. Cichocki, "Higher-order partial least squares (HOPLS): A generalized multilinear regression method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013, (to appear).

[15] L. De Lathauwer, "Decompositions of a higher-order tensor in block terms - Part II: Definitions and uniqueness," *SIAM J. Matrix Anal. Appl*, vol. 30, no. 3, pp. 1033–1066, 2008.

[16] Q. Zhao, G. Zhou, T. Adalı, and A. Cichocki, "Kernelization of tensor-based models for multimodal data analysis," *IEEE Signal Processing Magazine*, 2013, (to appear).

[17] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[18] Q. Zhao, C. F. Caiafa, D. P. Mandic, L. Zhang, T. Ball, A. Schulze-bonhage, and A. Cichocki, "Multilinear subspace regression: An orthogonal tensor decomposition approach," in *Advances in Neural Information Processing Systems 24 (NIPS)*, pp. 1269–1277. 2011.

[19] P. Honeine and C. Richard, "Preimage problem in kernel-based machine learning," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 77–88, 2011.

[20] M. Signoretto, L. De Lathauwer, and J. Suykens, "Kernel-based learning from infinite dimensional 2-way tensors," *Artificial Neural Networks–ICANN 2010*, pp. 59–69, 2010.

[21] P.J. 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.

[22] K.T. Abou-Moustafa and F.P. Ferrie, "A note on metric properties for some divergence measures: The Gaussian case," *The Journal of Machine Learning Research*, vol. 25, pp. 1–15, 2012.