Tensor-Based Gaussian Processes Regression Using a Probabilistic Kernel with Information Divergence

Lingbing Tang¹,², Pin Peng⁵, Qibin Zhao³*, Lihua Gui⁴, Luochang Qing⁵

¹Mobile E-business Collaborative Innovation Center of Hunan Province, Hunan University of Commerce, Changsha 410205, Hunan, China
²Key Laboratory of Hunan Province for Mobile Business Intelligence, Changsha 410205, Hunan, China
³Lab.for Advanced Brain Signal Processing, Brain Science Institute, RIKEN.2-1, Hirosawa, Wakoshi, Saitama, Japan
⁴Graduate School of Engineering, Saitama Institute of Technology, Japan
⁵School of Finance and Banking, Hunan University of Commerce, Changsha 410205, Hunan, China

Abstract — We present a Gaussian processes regression for tensor-valued inputs, which is based on a coherent treatment of a prior for the latent function with a covariance function defined on a tensor representation of multivariate data. Such a rigorous account of the underlying data dependencies within the model is also the basis for a new and powerful framework for structural data analysis. This is achieved under the assumption of generative models, and by introducing a family of probabilistic kernels with tensor arguments. By equipping these product kernels with the Jensen-Shannon divergence measure for tensorial data, the proposed model can be considered as both a generative and discriminative model. Further, a fully Bayesian treatment is used to estimate the hyperparameters and infer the predictive distributions. Experimental results on a real world application of decoding 3D movement trajectory based on electrocorticography (ECoG) signals recorded from a monkey brain demonstrate the effectiveness and performance advantage of the proposed framework.

Keywords - tensor; gaussian processes; probabilistic kernel; information divergence

I. INTRODUCTION

Gaussian processes (GP) models are powerful tools for Bayesian nonlinear and non-parametric regression, in which the prior distributions over latent function can be defined implicitly by the mean and covariance function [1, 2]. The hierarchical Bayesian modeling based on GPs and inference conducted in the function space by evaluating the posterior process are proposed in [3, 4]. However, the most existing aspects of GPs can only be achieved in multivariate input data spaces.

Tensor decomposition or factorization techniques can be regarded as the multilinear generalization of constrained matrix factorizations[5, 6] and have been successfully used in exploratory data analysis or unsupervised learning. The structure information of the data can be captured by multi-way structures, which is a priori information about original data nature. This promises advantages over matrix factorizations, due to a more effective use of the underlying properties of the structured data.

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 [7, 8] and have emphasized the need for the structural information of original data to be taken into account. Tensor representation is natural and efficient for such multiway structural data, while 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. Recent research has addressed extensions of the kernel concept into tensor decompositions [9, 10], bringing together the desirable properties of kernel methods and tensor decompositions for significant performance gain when the data are structured and exhibit nonlinear dependencies.

In order to combine the powerful GP model and Bayesian inference with tensor representation for structured data, in this study, we investigate the GP regression model based on tensor-variate inputs. A new family of kernels for tensorial data based on probabilistic generative models, called probabilistic product kernels for tensorial data, is introduced. Based on this, we propose a tensor-based GP regression model and algorithm with a fully Bayesian treatment. Unlike the standard GP, the input data X are also modeled with multiple generative models, which allow us to bring together the advantages of generative and discriminative models.

The paper is organized as follows. In Sec. II, some preliminary multilinear algebra is presented. In Sec.III, we introduce a hierarchical Bayesian models for tensor based GP and the fundamental inference of predictive distribution. The crucial issue is the covariance function for tensorial
inputs, implicitly a priori distribution over the latent function is discussed in Sec. IV. Subsequently, and the hyperparameter learning procedure is described in Sec. V. The effectiveness of the proposed model and the corresponding inference is demonstrated by both simulations on synthetic data and a real-world application of reconstruction of 3D movement trajectory based on ECoG signals recorded from monkey brain in Sec. VI. Finally, Sec. VII concludes the study.

II. NOTATION AND MULTILINEAR ALGEBRA

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 element $i_1, i_2, \ldots, i_N$ of an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{i_1 \times i_2 \times \cdots \times i_N}$ is denoted as $\mathcal{X}(i_1, i_2, \ldots, i_N)$. 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 $\mathbf{X}_{(n)} \in \mathbb{R}^{i_1 \times i_2 \times \cdots \times i_{n-1} \times i_{n+1} \times \cdots \times i_N}$, while the vectorization of a tensor is denoted as $\text{vec}(\mathcal{X})$. The n-mode product of a tensor $\mathcal{X} \in \mathbb{R}^{i_1 \times i_2 \times \cdots \times i_N}$ and matrix $\mathbf{A} \in \mathbb{R}^{i_1 \times k}$, denoted by $\mathcal{Y} = \mathcal{X} \otimes_n \mathbf{A}$, is defined as:

$$
\mathcal{Y}(i_1, i_2, \ldots, i_{n-1}, k, i_{n+1}, \ldots, i_N) = \sum_{i_n} \mathcal{X}(i_1, i_2, \ldots, i_N) A_{i_1, i_2, \ldots, i_{n-1}, i_{n+1}, \ldots, i_N}
$$

The inner product of two tensors $\mathcal{X}, \mathcal{X}' \in \mathbb{R}^{i_1 \times i_2 \times \cdots \times i_N}$ is defined by $\langle \mathcal{X}, \mathcal{X}' \rangle = \sum_{i_1, i_2, \ldots, i_N} \mathcal{X}(i_1, i_2, \ldots, i_N) \mathcal{X}'(i_1, i_2, \ldots, i_N)$ and is Frobenius norm by $\|\mathcal{X}\|_F = \langle \mathcal{X}, \mathcal{X} \rangle^{\frac{1}{2}}$.

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 N-order tensor, then Tucker model is defined as follows:

$$
\mathcal{X} = \mathbf{A} \times^{(1)} \mathbf{X} \times^{(2)} \cdots \times^{(N)} \mathbf{U}^{(N)}
$$

(2)

where $\mathbf{A} \in \mathbb{R}^{i_1 \times r}$ denotes the core tensor and $\mathbf{U}^{(n)} \in \mathbb{R}^{i_n \times r}$ denotes the mode-$n$ factor matrix. When all factor matrices are orthogonal and the core tensor is all-orthogonal this model is called HOSVD [11]. When all the factor matrices have the same number of components, and the core tensor is super-diagonal, Tucker model simplifies to CP model [6, 12] which is defined as a sum of rank-one tensors:

$$
\mathcal{X} = \sum_{r=1}^{R} \lambda_r \mathbf{u}^{(1)}_r \otimes \mathbf{u}^{(2)}_r \otimes \cdots \otimes \mathbf{u}^{(N)}_r
$$

(3)

Where the symbol $\otimes$ denotes the outer product of vectors and $\mathbf{R}$ is defined as tensor rank. In general, CP model is considered to be a multilinear low-rank approximation while Tucker model is regarded as a multilinear subspace approximation.

III. MODEL AND INFERENCE

We consider a supervised learning problem, where the input is represented as an Mth-order tensor $\mathcal{X}_n \in \mathbb{R}^{i_1 \times \cdots \times i_M}$ and a scalar output or target is denoted by $y_n \in \mathbb{R}$ Given a paired dataset of observations, $D = \{(\mathcal{X}_n, y_n) \mid n = 1, \ldots, N\}$, the tensor inputs for all $N$ cases are concatenated into an $(M + 1)$th-order design tensor $\mathcal{X} \in \mathbb{R}^{i_1 \times \cdots \times i_M \times i_0}$, and the targets are collected in a vector $y = \{y_1, \ldots, y_N\}^T$. After observing the training data $D = \{(\mathcal{X}_n, y_n)\}$, we are interested in making inferences about the relationship between inputs and targets, i.e. the conditional distribution of the targets given the inputs, and making predictions for a new input $\mathcal{X}$ that we have not seen in the training set.

The observations are assumed to satisfy

$$
y_n = f(\mathcal{X}_n) + \mathcal{E}, \text{ where } \mathcal{E} \sim \mathcal{N}(0, \sigma^2)
$$

(4)

The $\mathcal{E}$ is an i.i.d. Gaussian noise with zero mean and variance $\sigma^2$. $f$ denotes a nonlinear latent function, which we are interested in. The distribution of observations can be factored over cases in the training set by

$$
Y \sim \prod_{n=1}^{N} \mathcal{N}(y_n \mid f_n, \sigma^2)
$$

(5)

where $f_n$ denotes latent function $f(\mathcal{X}_n)$. A Gaussian process prior can be placed over the latent function, which

---

DOI 10.5013/IJSSST.a.16.05.13 13.2 ISSN: 1473-804x online, 1473-8031 print
implies that any finite subset of latent variables has a multivariate Gaussian distribution, denoted by

\[ f(\chi) \sim \Phi(m(\chi), k(\chi, \chi')) \theta \]  

where \( m(\chi) \) is mean function and usually is set to zero for notational simplicity, and \( k(\chi, \chi') \) is the covariance function for tensorial data with a set of hyperparameters \( \theta \). The hyperparameters from observation model (4) and GP prior (5) are collected in \( \Theta = \{\sigma, \theta\} \). The model is hierarchy extended to the third level by giving also prior for the hyperparameters of \( \Theta \). The probabilistic graphic model of tensor based GPs is illustrated in Fig. 1. Unlike standard GP, observations \( \{\chi_n\} \) are also modeled individually with parameters \( \lambda_n \), which will be used for defining probabilistic product kernels for tensorial data. The probabilistic product kernel is discussed in Section IV.

To incorporate the knowledge that the training data provides about the function, we use Bayes rule to infer the posterior of the latent function \( f = [f(\chi_1), \ldots, f(\chi_N)]^T \) by

\[ p(f|D, \Theta) = \frac{p(Y|f, \sigma) p(f|\chi, \theta)}{p(Y|f, \sigma) p(f|\chi, \theta) df} \]  

where the denominator in (7) is the marginal likelihood obtained by integration over \( f \), yielding

\[ Y|\chi, \theta, \sigma^2 \sim N(Y|0, K + \sigma^2 I) \]  

where \( (K)_{ij} = k(\chi_i, \chi_j) \) denotes the covariance matrix or kernel matrix. According to the conditioning joint Gaussian prior distribution on the observations, for a new data point \( \chi_* \), we have

\[ p(f_*|\chi_*, f, \Theta) \sim N(f_*, \text{cov}(f_*)) \]  

where

\[ f_* = k(\chi_*, \chi) k(\chi, \chi)^{-1} f \]

\[ \text{cov}(f) = k(\chi_*, \chi_*) - k(\chi_*, \chi) k(\chi, \chi)^{-1} k(\chi, \chi_*) \]

Based on (9) and (7), the predictive distribution of \( f \) is obtained by integrating out the latent \( f \),

\[ p(f_*|\chi_*, D, \Theta) = \int p(f_*|\chi_*, f, \Theta) p(f|D, \Theta) df \]  

and the predictive distribution for noisy observations \( y \), can be obtained by integrating out \( f_* \),

\[ p(y_*|\chi_*, D, \Theta) = \int p(y_*|f_*, \sigma^2) p(f_*|\chi_*, D, \Theta) df_* \]  

Since the Gaussian observation model is analytically tractable case, which avoids the approximation inference, the conditional posterior of latent function \( f \) is Gaussian, and posterior of \( f \) is also Gaussian together with the observation \( y \). Finally, the predictive distribution of \( y \) corresponding to \( \chi_* \) can be inferred as

\[ y_*|\chi_*, \chi, y, \Theta \sim N(y_*, \text{cov}(y_*)) \]

where

\[ y_* = k(\chi_*, \chi)(k(\chi, \chi) + \sigma^2 I)^{-1} y \]

\[ \text{cov}(y) = k(\chi_*, \chi_*) - k(\chi_*, \chi)(k(\chi, \chi) + \sigma^2 I)^{-1} k(\chi, \chi_*) \]

Note that mean prediction is considered as a linear combination of observations \( y \) and can also be considered as a linear combination of \( N \) kernels with coefficients \( (K + \sigma^2 I)^{-1} y \).

IV. COVARIANCE FUNCTION IN TENSOR INPUT SPACE

A. Kernels for Tensorial Data

The covariance function is the crucial ingredient in a Gaussian process predictor, as it encodes our assumption about the function which we wish to learn [1]. In supervised learning the notion of similarity between data points is the key issue. A basic assumption that points with inputs \( \chi_n \) which are close are likely to have similar target value \( y_n \), and thus training points that are near to a test point should be informative about the prediction at that point. For Gaussian process, it is the covariance function that defines nearness or similarity, which is same to the kernel function. The kernels are considered as defining a topology implying the apriori knowledge about invariance in the input space. The complex kernels can be created by simple operations that combine simpler kernels. Note that the kernel function is valid only if it satisfies the finitely positive semi-definite property [13].

It is important to take into account structure in the input domain. There are many kernels designed for a number of structured objects, but few methods exploit the structure of tensorial representations until that a tensorial kernel exploiting algebraic geometry of spaces of tensors and a similarity measure between the different subspaces spanned by higher-order tensors were proposed [14]. There are some valid reproducing kernels toward a straightforward generalization to Mth-order tensors, such as the kernel functions \( k: \chi \times \chi \rightarrow R \) given as

Linear kernel: \( k(\chi, \chi') = \langle \text{vec}(\chi), \text{vec}(\chi') \rangle \)

Gaussian RBF: \( k(\chi, \chi') = \exp\left(-\frac{1}{2\sigma^2}||\chi - \chi'||^2\right) \)

The general product kernels can be defined by M factor kernels to describe the similarity measure using the multilinear algebraic structure of input tensors, denoted by

\[ k(\chi, \chi') = \prod_{m=1}^{M} k(X_{(m)}, X'_{(m)}) \]
where each factor kernel represents a similarity measure between two matrices gained by mode-m unfolding of two tensor examples. One possibility of similarity measure between matrices is Chordal distance (projection Frobenius norm) on the Grassmannian manifolds [14]. The Chordal distance can be computed based on the right singular vectors $V_X^{(n)}$, which provides us rotation and reflection invariance for elements on the Grassmann manifold.

**B. Probabilistic Kernels Based on Information Divergence**

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 [15] and Kullback-Leibler Kernel [16]. The Fisher kernel assumes a generative model that explains well all possible data and maps each individual sample into a gradient log likelihood distribution, or equivalently as the diversity of two probability distributions, such as Fisher kernel [15] and divergences that are measures of dissimilarity between moment of the posterior (MAP) estimate, given by

$$k(\chi, \chi') = \alpha^2 \prod_{m=1}^{M} \exp\left(-\frac{\sqrt{D(p(\chi|\alpha^m)^{\text{cond}})(p(\chi'|\alpha^m))}}{2\beta^2_m}\right) \tag{16}$$

where $\alpha$ denotes the magnitude hyperparameters and $[\beta_1,\ldots,\beta_M]$ play the role of characteristic length-scales which implement automatic relevance determination (ARD) [19]. Since the inverse of the length-scale determines how relevant an input is: if the length-scale has a very large value, the covariance will become almost independent of that input, effectively removing it from inference $\theta = [\alpha, \beta_m, m = 1,\ldots,M]$ is a vector containing all the hyperparameters of the tensor product kernel. Since the distance of $\sqrt{\text{JS}(p||q)}$ has been proved to be a metric, the tensor kernel defined in (16) is a metric kernel and positive definite.

**V. MAP ESTIMATION OF HYPERPARAMETERS**

The problem of learning in Gaussian processes is exactly the problem of finding suitable properties for the covariance function. Many covariance functions have adjustable parameters that can be inferred or learned from the data based on Bayesian approach. In the tensor-variate covariance function defined in (16), a number of free hyperparameters whose values also need to be inferred. In full Bayesian approach we should integrate over all unobserved variables. Given we have integrated over the latent function, it often happens that the posterior of the hyperparameters is peaked or predictions are insensitive to small changes in parameter values. Hence, we can approximate the integral over $p(\theta, \sigma|D)$ with maximum a posterior (MAP) estimate, given by

$$\hat{\theta}, \hat{\sigma} = \arg\min_{\theta, \sigma} [-\log p(D|\theta, \sigma) - \log p(\theta)\sigma - p(\sigma|y)] \quad \tag{17}$$

$\theta = [\alpha, \beta_1,\ldots,\beta_M]$ is hyperparameters for covariance function, while $\sigma$ represents a hyperparameter of noise for the observation model. In this approximation, the parameter values are given a point mass one at the posterior mode, and the latent function marginal is approximated as $p(f|D) \approx p(f|D, \hat{\theta}, \hat{\sigma})$ .The log marginal likelihood is differentiable with respect to the parameters, and thus differentiable also for the log posterior, which allows gradient based optimization. In (17), the first item is marginal likelihood and its partial derivatives w.r.t. the hyperparameters can be obtained by

$$\frac{\partial}{\partial \theta_j} \log p(y|\chi, \theta, \sigma) = \frac{1}{2} y_j K^{-1}_r \frac{\partial K}{\partial \theta_j} K^{-1}_r y - \frac{1}{2} r(K^{-1}_r \frac{\partial K}{\partial \theta_j} K^{-1}_r) \tag{18}$$

$$$$
where $K_f = K_f + \sigma_f^2 I$ denotes the covariance matrix for the noisy targets $y$. The inference on the parameters of covariance functions is conducted mainly transformed space, e.g. log-transformation, which has the advantage that the parameter space is transformed into $(-\infty, +\infty)$. For example, $p(\theta)$ is transformed into the parameter $w = g(\theta)$, the posterior of $w$ can be written as

$$p(w|\theta) \propto p(y|x,\theta) p(\theta|\theta_0) \theta$$

which leads to energy function $E(w) = E(\theta) - \log(\theta)$ and the gradient expressed by $\partial E(w) / \partial w = \partial E(\theta) / \partial \theta - 1$. The partial derivatives of covariance matrix $K_f$ obtained by covariance function defined in (16) w.r.t. the hyperparameters is expressed by

$$\frac{\partial K_f}{\partial \log(\beta_m^n)} = \frac{1}{\beta_m^n} \sqrt{D(p(\gamma|x|\beta_m^n)q(\gamma'|x'|\beta_m^n))} K_f,$$

$$\frac{\partial K_f}{\partial \log(\alpha^2)} = K_f$$

(20)

For hyperprior of hyperparameters $\theta$, we placed an inverse-gamma distribution for length-scale parameters $[\beta_1, ..., \beta_d]$ which is given by

$$p(\beta) = \frac{b^a}{\Gamma(a)} \beta^{-a-b} \exp(-b/\beta)$$

(21)

where shape parameter $a = 4$ and scale parameter $b = 1$. The magnitude parameter $\alpha^2$ is simply placed a hyperprior of square root uniform distribution. For hyperprior of $\sigma^2$, we use a log-uniform distribution.

VI. RESULTS

A brain-computer interface (BCI) is a direct communication pathway between brain and an external device, which gained a great deal of attention in the past decades [20]. A common BCI research strategy begins by decoding kinematic parameters from neural activities recorded from the surface of the cortex, e.g. electrocorticography (ECoG), or from the scalp, e.g. electroencephalography (EEG). Decoding 2D/3D limb movement trajectories from ECoG has been successfully investigated in [21, 22, 23] using linear or multilinear regression methods such as partial least squares (PLS) and higher-order partial least squares (HOPLS). In order to demonstrate the effectiveness of the proposed model, in this study, we apply the tensor-based Gaussian process (tensorGP) regression for decoding of 3D hand movement trajectories from ECoG signals recorded from a monkey based on an open database (http://neurotycho.org).

Figure 2. The comparisons of predictive performance among PLS, HOPLS and tensorGP. (a) shows correlation coefficients $r^2$ between predicted $\hat{y}$ and real trajectories $y$ while (b) shows performance evaluated by $Q^2 = 1 - \|\hat{y} - y\|^2 / \|y - \bar{y}\|^2$. Observe that tensorGP outperforms both PLS based on vectorization of tensor $X_e$ and HOPLS based on tensor representation and multilinear regression.

More specifically, ECoG signals were preprocessed by a band-pass filter with cutoff frequencies at 0.1 and 600Hz and a spatial filter with a common average reference. The feature extraction using a Morlet wavelet transformation at 10 different center frequencies (10-150Hz, arranged in a logarithmic scale) was performed to obtain the time-frequency representation. The hand movement of a monkey was captured by a motion capture system and down-sampled to 1 Hz, thus the output data was represented as a matrix $Y \in \mathbb{R}^{N \times 3}$ consisting of multiple variables (i.e., time 3D positions). For every time point $y$, a corresponding ECoG epoch (one second before the current time point) was extracted to construct the predictors. Hence, the whole independent data is naturally represented as a 4th-order tensor (i.e., epoch $\times$ channel $\times$ time $\times$ frequency), denoted by

$\begin{bmatrix}
PLS & HOPLS & TensorGP \\
\hline
X-position & 0.5 & 0.6 & 0.7 \\
Y-position & 0.4 & 0.5 & 0.6 \\
Z-position & 0.3 & 0.4 & 0.5 \\
\end{bmatrix}$

$\begin{bmatrix}
PLS & HOPLS & TensorGP \\
\hline
Prediction Q^2 & 0.7 & 0.8 & 0.9 \\
\end{bmatrix}$
The dataset is divided into training set (10 minutes) and test set (5 minutes).

Three regression models are used for performance evaluation and comparison, including linear PLS, multilinear HOPLS and nonlinear tensor-based Gaussian process. The hyperparameters of tensorGP are learned from data using maximum a posterior (MAP) estimation while the selection of parameters in PLS and HOPLS is obtained by cross-validation on the training data. The prediction performances for the test set are shown in Fig.2, demonstrating the superiority of tensorGP over both PLS and HOPLS. Fig. 3 illustrates the reconstructed 3D hand movement trajectories with uncertainty obtained by tensorGP model. Observe that the uncertainty is always small during movements and large during nonmovements, implying the discriminative patterns of ECoG corresponding to movements were successfully captured by tensorGP model. These results are meaningful from neuroscience perspective since the exact positions of any limbs without movements are more difficult to be reconstructed from dynamic neural activities than movements.

Table I shows MAP estimations of hyperparameters. Since the length-scales in $\theta$ play a key role for automatic relevance determination (ARD), i.e., a very large value indicates an irrelevant input, $\theta_{\text{MAP}}$ provides us meaningful explanations that the spatio-temporal features are more discriminative than spectral features for reconstruction of X- and Y-positions, while temporal features are more discriminative than spatial and spectral features for reconstruction of Z-positions.

VII. CONCLUSION

We have present a Gaussian process regression framework based on tensorial data space that brings together the advantages of GP model and tensor representation, such that tensor-based GP regression with Bayesian inference can be performed directly on structured data represented by higher-order tensors. The presented probabilistic kernel function for tensorial data has been demonstrated as an effective similarity measure with respect to predictive performance, thus providing a new perspective for the development of a range of machine learning methods that admit the underlying multilinear structure. The advantages of the proposed approach have been demonstrated by a practical application for reconstruction of movement trajectories from brain signals.

ACKNOWLEDGMENT

This work was supported by National Social Science Fund Project (No.12GBL028); The humanities and social science Fund project of the Ministry of Education
REFERENCES


