Online Local Gaussian Process for Tensor-variate Regression: Application to Fast Reconstruction of Limb Movements From Brain Signal

Ming Hou Yali Wang Brahim Chaib-draa

Laval University, Quebec, Canada

Introduction

- Tensor Gaussian process (tensor GP) is a powerful Bayesian nonparametric regression method that flexibly models the nonlinearity of the tensorial data
- Computation load $O(N^3)$ of GP training restricts the applicability of tensor GP to large-scale problem in practice
- Tensor online local Gaussian process (tensor OLGP) is a computationally efficient framework for tensor-variate regression

Tensor GP Regression

- The training set $\mathcal{D} = \{(\mathcal{X}_n, y_n)\}_{n=1}^N$ where the scalar output $y_n \in \mathbb{R}$ is generated by a nonlinear function $f(\mathcal{X}_n)$ of the D-order tensor input $\mathcal{X}_n \in \mathbb{R}^{I_1 \times \cdots \times I_D}$ with an additive Gaussian noise $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$

$$y_n = f(\mathcal{X}_n) + \epsilon_n$$

- The latent function of tensor GP approach can be modeled by a GP

$$f(\mathcal{X}) \sim \mathcal{GP}(m(\mathcal{X}), k(\mathcal{X}, \mathcal{X}')|\theta)$$

- The covariance function $k(\mathcal{X}, \mathcal{X}')$ is product probabilistic kernel

$$k(\mathcal{X}, \mathcal{X}') = \alpha^2 \prod_{d=1}^{D} \exp\left(\frac{KL(p(\mathbf{x}|\Omega_d^{\mathcal{X}}) \parallel q(\mathbf{x}'|\Omega_d^{\mathcal{X}'}))}{-2\beta_d^2}\right)$$

- The joint predictive distribution of the latent function is Guassian

$$p(f_*, \mathbf{y}|\mathcal{X}_*, \boldsymbol{\mathcal{X}}, \theta, \sigma^2)$$

- The conditional predictive distribution is also Gaussian

$$p(f_*|\mathcal{X}_*, \boldsymbol{\mathcal{X}}, \mathbf{y}, \theta, \sigma^2) = \mathcal{N}(m_*, \sigma_*^2)$$

$$m_* = \mathbf{k}_{\mathcal{X}}^T(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y}$$

$$\sigma_*^2 = k_* - \mathbf{k}_{\mathcal{X}}^T(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{k}_{\mathcal{X}}$$

Tensor OLGP Regression

- **Stage 1 GP Experts Construction**: Use the covariance function of tensor GP as a similarity measurement to sequentially partition the training data points into a number of small-sized experts
- **Stage 2 Local Prediction**: Find a fixed-number of local GP experts to make predictions (for given test tensorial inputs) with a Gaussian mixture

Stage 1 GP Experts Construction

Algorithm 1 GP Experts Construction

```
Input: new tensor data pair {X_new, y_new}
for k = 1 to number of local experts R do
    Compute the similarity to the kth expert using probabilistic tensor kernel function (3):
    w_k = k(X_new, C_k)
end for
Choose the nearest local expert t: sim_t = max(w_k)
if sim_t > w_gen then
    Insert {X_new, y_new} to the nearest local expert t:
    𝓧_t = [𝓧_t, X_new], y_t = [y_t, y_new]
    if maximum number of data points is reached then
        delete another point by permutation
    end if
    Update the corresponding kernel matrix K_t by computing the kernel vector k_t(X_new, 𝓧_t) for X_new
else
    Create a new expert:
    C_{R+1} ≐ X_new, 𝓧_{R+1} = [X_new], y_{R+1} = [y_new]
    Initialize the new kernel matrix K_{R+1}
end if
```

Stage 2 Local Prediction

- **Input-based Searching Strategy**:
  - Find M local experts having the highest similarities with $\mathcal{X}_*$ among all the local experts according to tensor kernel function $w_k = k(\mathcal{X}_*, \mathcal{C}_k)$
- **Input-output-based Searching Strategy**:
  - Find its nearest local expert $\mathcal{C}_k \doteq \{\mathcal{X}_{\mathcal{C}_k}, y_{\mathcal{C}_k}\}$ from the input $\mathcal{X}$-space using tensor kernel function
  - Find M local experts $\{\mathcal{C}_m \doteq \{\mathcal{X}_{\mathcal{C}_m}, y_{\mathcal{C}_m}\}\}_{m=1}^M$ that are being closest to $y_{\mathcal{C}_k}$ in y-space among all the local expert centers as the candidates
- **Weighted Local Prediction**:
  - Use $w_k = k(\mathcal{X}_*, \mathcal{C}_k)$ as the weight of local expert k, and the prediction $\hat{y}_*$ is the weighted combination from each local prediction $\bar{y}_k = \mathbf{k}_k(\mathcal{X}_*, \boldsymbol{\mathcal{X}}_k)^T(\mathbf{K}_k + \sigma^2\mathbf{I})^{-1}\mathbf{y}_k$ as follows

$$\hat{y}_* = \frac{\sum_{k=1}^M w_k \bar{y}_k}{\sum_{k=1}^M w_k}$$

Computational Complexity

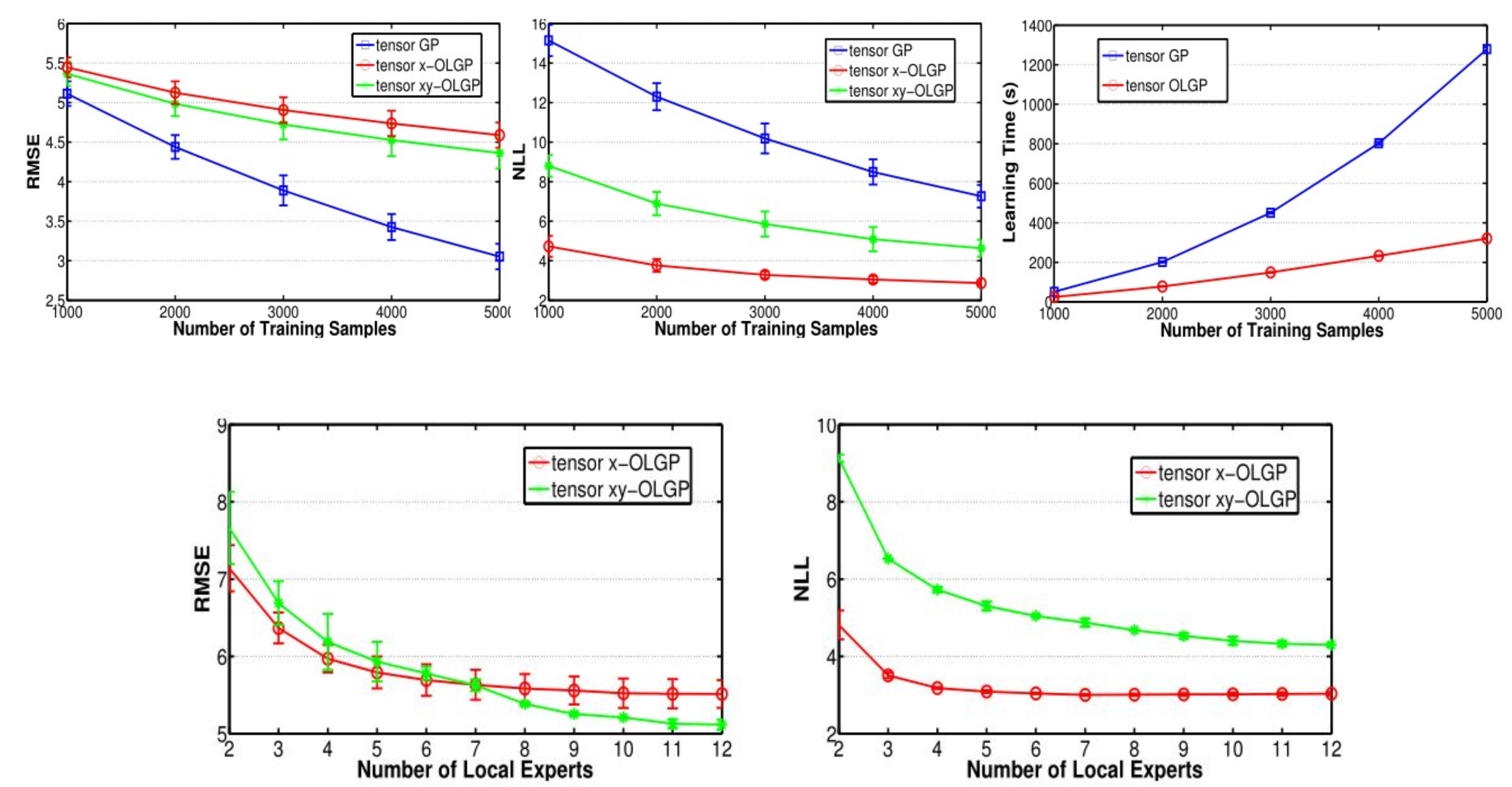
	Partation + Training	Prediction
tensor GP	$\mathcal{O}(N^2 I^{D+1} + N^3)$	$\mathcal{O}(N I^{D+1} + N^2)$
tensor OLGP	$\mathcal{O}(NRI^{D+1} + NSI^{D+1} + S^3)$	$\mathcal{O}(RI^{D+1} + M(SI^{D+1} + S^2))$

ECoG Food Tracking Task Dataset

- **Comparison of prediction of movement on shoulder marker along x-axis**

Data size, w_{gen}	Method	RMSE	NLL	Running Time (s)	
				Training	Testing
10000, 0.5	tensor GP	3.05 ± 0.16	7.26 ± 0.57	1279.1 ± 9.2	2480.6 ± 16.7
	tensor x-OLGP	4.71 ± 0.15	2.86 ± 0.10	321.0 ± 3.9	503.5 ± 4.7
	tensor xy-OLGP	4.39 ± 0.18	4.53 ± 0.43	321.0 ± 3.9	492.4 ± 8.3
10000, 0.6	tensor x-OLGP	4.56 ± 0.14	2.66 ± 0.07	511.1 ± 3.2	829.9 ± 6.4
	tensor xy-OLGP	3.82 ± 0.15	4.03 ± 0.41	511.1 ± 3.2	822.0 ± 6.8
36000, 0.4	tensor GP	3.40 ± 0.19	10.15 ± 0.81	19141.9 ± 163.5	39152.4 ± 230.9
	tensor x-OLGP	5.77 ± 0.19	3.18 ± 0.12	2819.9 ± 37.3	5135.2 ± 66.3
	tensor xy-OLGP	5.62 ± 0.24	4.67 ± 0.48	2819.9 ± 37.3	4503.0 ± 48.1

- **Comparison vs. number of training samples and vs. number of local experts**



Conclusion

- A new tensor-variate local GP regression framework has been introduced, it successfully adapts the local GP modeling to the tensor input space
- Large data is efficiently processed by several small-sized GP in an online way