Distributed Gaussian Process: New Paradigm and Application to Wireless Traffic Prediction

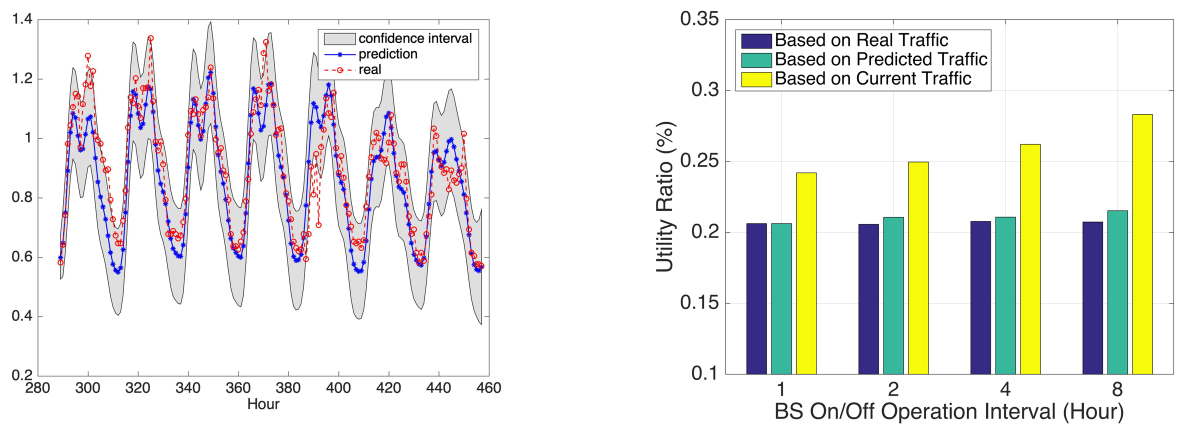
Yue Xu†*, Feng Yin*, Wenjun Xu†, Jiaru Lin†, Shuguang Cui‡*
†Key Lab of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications
*The Chinese University of Hong Kong, Shenzhen and SRIBD
‡Department of Electrical and Computer Engineering, University of California, Davis

Background

- **What is GP?**

Gaussian process (GP) is a class of important Bayesian non–parametric models for machine learning.

- **Applications**



(a) Predicted Traffic by GP (b) Energy Saving Result

Figure 1. Wireless traffic prediction based energy saving.

- **Challenge**

Standard GP suffers from the high complexity of hyper–parameter optimization, which scales as $\mathcal{O}(n^3)$ with n the number of training samples.

Main Result

A principled and elegant scalable GP framework for big data applications, specifically:

- **Training phase:** the first to bring in the alternating direction method of multipliers (ADMM) algorithm, which reduces the complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^3/k^3)$ with k the number of parallel computing units.
- **Prediction phase:** the first to fuse local prediction results via optimizing the fusion weights based on cross-validation, which has a complexity of $\mathcal{O}(\sqrt{\log K})$.

Regression Model

- **GP Definition**

A GP is a collection of random variables, any finite number of which follows a Gaussian distribution.

- **GP-based Regression Model**

$$y = f(\boldsymbol{x}) + e, \quad f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta}_h))$$

where
- $m(\boldsymbol{x})$: mean function
- $k(\boldsymbol{x}, \boldsymbol{x}'; \boldsymbol{\theta}_h)$: kernel function

- **Kernel Function for Wireless Traffic Prediction**

① Weekly periodic pattern:

$$k_1(t_i, t_j) = \sigma_{p_1}^2 \exp\left[-\frac{\sin^2\left(\frac{\pi(t_i - t_j)}{\lambda_1}\right)}{l_{p_1}^2}\right]$$

② Daily periodic pattern:

$$k_2(t_i, t_j) = \sigma_{p_2}^2 \exp\left[-\frac{\sin^2\left(\frac{\pi(t_i - t_j)}{\lambda_2}\right)}{l_{p_2}^2}\right]$$

③ Dynamic deviations:

$$k_3(t_i, t_j) = \sigma_{l_t}^2 \exp\left[-\frac{(t_i - t_j)^2}{2l_{l_t}^2}\right]$$

④ Composite kernel function:

$$k(t_i, t_j) = k_1(t_i, t_j) + k_2(t_i, t_j) + k_3(t_i, t_j)$$

⑤ Hyper-parameters to learn:

$$\boldsymbol{\theta}_h = [\sigma_{p_1}^2, \sigma_{p_2}^2, \sigma_{l_t}^2, l_{p_1}^2, l_{p_2}^2, l_{l_t}^2]^T$$

Standard GP

- **Hyper-parameter Optimization**

$$\mathcal{P}_0: \quad \min_{\boldsymbol{\theta}} \quad l(\boldsymbol{\theta}) = \boldsymbol{y}^T \boldsymbol{C}^{-1}(\boldsymbol{\theta})\boldsymbol{y} + \log|\boldsymbol{C}(\boldsymbol{\theta})|$$
$$\text{s.t.} \quad \boldsymbol{\theta} \in \Theta, \ \Theta \subseteq \mathbb{R}^p$$

where
- $\boldsymbol{C}(\boldsymbol{\theta}) \triangleq \boldsymbol{K}(\boldsymbol{\theta}_h) + \sigma_e^2 \boldsymbol{I}_n$: covariance matrix
- $\boldsymbol{K}(\boldsymbol{\theta}_h)$: kernel matrix

- **Gradient Decent**

$$\theta_i^{r+1} = \theta_i^r - \eta \cdot \frac{\partial l(\boldsymbol{\theta})}{\partial \theta_i}\Big|_{\boldsymbol{\theta} = \boldsymbol{\theta}^r}$$

where

$$\frac{\partial l(\boldsymbol{\theta})}{\partial \theta_i} = \mathrm{Tr}\left(\left(\boldsymbol{C}^{-1}(\boldsymbol{\theta}) - \boldsymbol{\gamma}\boldsymbol{\gamma}^T\right)\frac{\partial \boldsymbol{C}(\boldsymbol{\theta})}{\partial \theta_i}\right)$$

with $\mathrm{Tr}(\cdot)$ the matrix trace and $\boldsymbol{\gamma} \triangleq \boldsymbol{C}^{-1}(\boldsymbol{\theta})\boldsymbol{y}$.

- **Posterior Inference**

$$p(\boldsymbol{y}_* | \mathcal{D}, \boldsymbol{X}_*; \boldsymbol{\theta}) \sim \mathcal{N}(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\sigma}})$$

where

$$\mathbb{E}[f(\boldsymbol{X}_*)] = \bar{\boldsymbol{\mu}} = \boldsymbol{k}_*^T \left(\boldsymbol{K} + \sigma_e^2 \boldsymbol{I}_n\right)^{-1} \boldsymbol{y}$$
$$\mathbb{V}[f(\boldsymbol{X}_*)] = \bar{\boldsymbol{\sigma}} = \boldsymbol{k}_{**} - \boldsymbol{k}_*^T \left(\boldsymbol{K} + \sigma_e^2 \boldsymbol{I}_n\right)^{-1} \boldsymbol{k}_*$$

Simulation Result

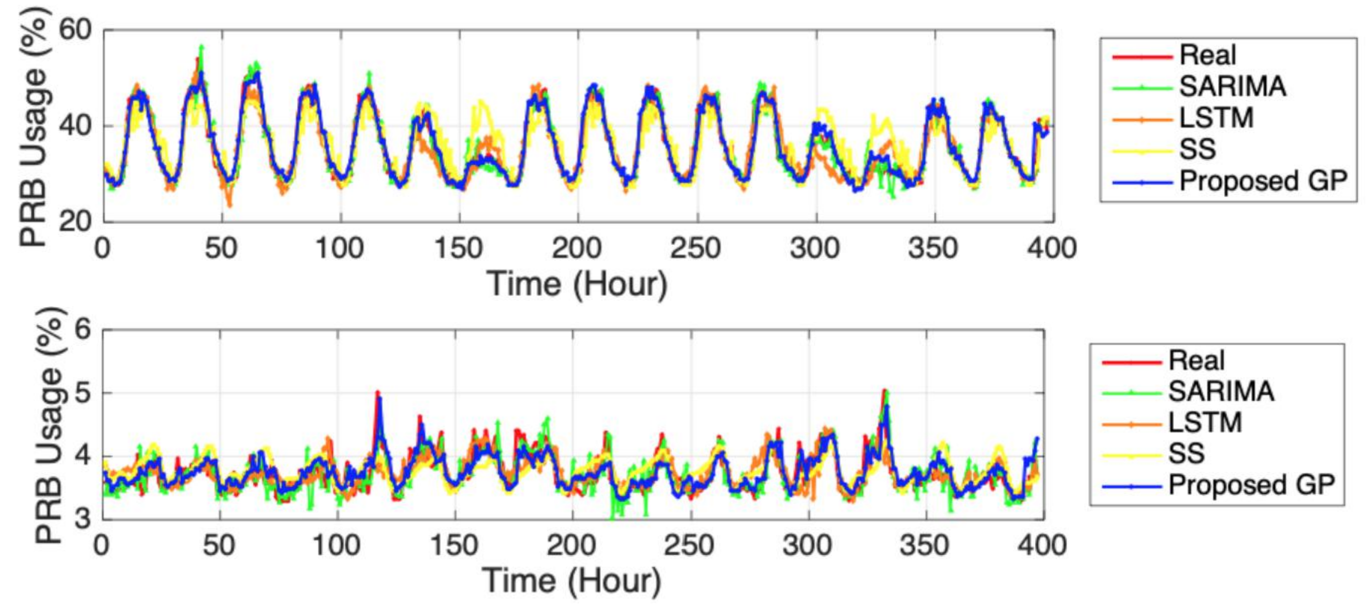


Figure 2. One-hour look-ahead prediction of three clusters.

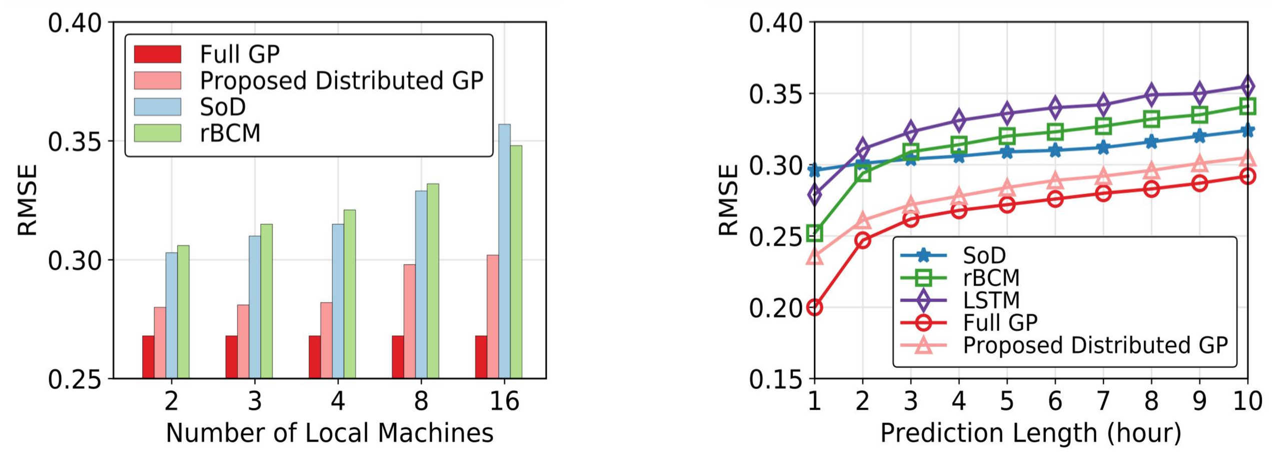


Figure 3. The wireless traffic prediction performance.

Model	1 BBU	2 BBUs	4 BBUs	8 BBUs	16 BBUs
STD	16.8s	3.5s	1.1s	0.4s	0.1s
TPLZ	6.9s	1.2s	0.4s	0.2s	0.1s
rBCM	16.8s	4.9s	2.4s	0.4s	0.2s

Table 1. Time consumption for training phase.

Weight Model	2 BBUs	4 BBUs	8 BBUs	16 BBUs
Mirror	0.07s	0.13s	0.21s	0.37s
Soft-max	0.06s	0.05s	0.03s	0.03s
rBCM	0.08s	0.06s	0.06s	0.05s

Table 2. Time consumption for prediction phase.

- Simulation Settings

– Real 4G traffic data, 3072 base stations, from Sep. 1st to Sep. 30th in 2015, grouped into 360 clusters

– 720 data points for each cluster, use 600 points to predict the next 10 points, repeated over 10000 times

Training Phase

- **Product-of-expert (PoE) model**

$$\log p(\boldsymbol{y}|\boldsymbol{X}; \boldsymbol{\theta}) \approx \sum_{i=1}^{K} \log p(\boldsymbol{y}^{(i)}|\boldsymbol{X}^{(i)}; \boldsymbol{\theta})$$

- **ADMM-based Hyper-parameter Optimization**

$$\mathcal{P}_2: \quad \min_{\boldsymbol{\theta}_i} \quad \sum_{i=1}^{K} l^{(i)}(\boldsymbol{\theta}_i)$$
$$\text{s.t.} \quad \boldsymbol{\theta}_i - \boldsymbol{z} = \boldsymbol{0}, \quad i = 1, 2, \ldots, K$$
$$\boldsymbol{\theta}_i \in \Theta, \quad i = 1, 2, \ldots, K$$

- **ADMM Iteration**

$$\boldsymbol{\theta}_i^{r+1} := \arg\min_{\boldsymbol{\theta}_i} \left(l^{(i)}(\boldsymbol{\theta}_i) + \boldsymbol{\zeta}_i^T(\boldsymbol{\theta}_i - \boldsymbol{z}) + \frac{\rho}{2}\|\ \boldsymbol{\theta}_i - \boldsymbol{z}\ \|_2^2\right)$$
$$\boldsymbol{z}^{r+1} := \frac{1}{K}\sum_{i=1}^{K}\left(\boldsymbol{\theta}_i^{r+1} + \frac{1}{\rho}\boldsymbol{\zeta}_i^r\right)$$
$$\boldsymbol{\zeta}_i^{r+1} := \boldsymbol{\zeta}_i^r + \rho(\boldsymbol{\theta}_i^{r+1} - \boldsymbol{z}^{r+1})$$

Prediction Phase

- **PoE-based Inference**

$$p(f_* | \boldsymbol{x}_*, \mathcal{D}) \approx \prod_{i=1}^{K} p_i^{\beta_i}(f_* | \boldsymbol{x}_* \mathcal{D}^{(i)})$$
$$\mu_* = (\sigma_*)^2 \sum_{i=1}^{K} \beta_i \sigma_i^{-2}(\boldsymbol{x}_*)\mu_k(\boldsymbol{x}_*), \quad \sigma_*^2 = \left(\sum_{i=1}^{K} \beta_i \sigma_i^{-2}(\boldsymbol{x}_*)\right)^{-1}$$

- **Cross-validation-based Fusion**

$$\mathcal{P}_3: \quad \min_{\boldsymbol{\beta}} \quad f(\boldsymbol{\beta}) = \sum_{m=1}^{M}\left(y_m - \frac{\sum_{i=1}^{K} a_i(x_m)\beta_i}{\sum_{i=1}^{K} b_i(x_m)\beta_i}\right)^2$$
$$\text{s.t.} \quad \boldsymbol{\beta} \in \Omega$$

- **Mirror Decents**

$$\beta_i^{r+\frac{1}{2}} = \beta_i^r \exp\left\{-\eta^r g_i^r\right\}$$
$$\beta_i^{r+1} = \frac{\beta_i^{r+\frac{1}{2}}}{\boldsymbol{e}^T \boldsymbol{\beta}^{r+\frac{1}{2}}}$$

- **Softmax-based Fusion**

$$\beta_k = \frac{\exp(-e_k)}{\sum_{k=1}^{K} \exp(-e_k)}$$

– Y. Xu, F. Yin, W. Xu, J. Lin and S. Cui, "Wireless Traffic Prediction with Scalable Gaussian Process: Framework, Algorithms, and Verification," in IEEE Journal on Selected Areas in Communications (JSAC), vol. 37, no. 6, pp. 1291-1306, June 2019.

– Y. Xu, F. Yin, W. Xu, J. Lin and S. Cui, " Scalable Gaussian Process Using Inexact ADMM for Big Data," in IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Brighton, UK, May 2019, to appear.