The Use of Gaussian Processes as Particles for Sequential Monte Carlo Estimation of Time-Varying Functions

Yuhao Liu
Department of Applied Mathematics and Statistics
Stony Brook University
Stony Brook, NY
yuhao.liu.1@stonybrook.edu

Petar M. Djurić
Department of Electrical and Computer Engineering
Stony Brook University
Stony Brook, NY
petar.djuric@stonybrook.edu

***Abstract*—We propose modeling of time-varying functions by Gaussian processes based on random features and relying on the sequential Monte Carlo methodology, also known as particle filtering. The models make use of time-varying random features and parameter variables to adapt to changes of the modeled functions with time. The Gaussian processes are treated as latent states and are estimated by using particle filtering, which altogether allows for learning functions at each time instant. The proposed models have the ability to search for optimal functions in the dynamic space over time. The experimental results show that the approach has better performance than existing state-of-the-art methods based on ensemble of Gaussian processes both in accuracy and stability.**

***Index Terms*—Gaussian process, particle filtering, random features, sequential learning**

I. Introduction

Gaussian processes (GPs) have become a useful tool in machine learning (ML), especially in measuring uncertainties and improving robustness during the learning process [13]. The main limitation of GPs is their high requirement for computations with scale. A number of approximations to GPs have been put forth that reduce their computational complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(NM^2)$, where N is the number of training samples, M is the dimension of the summary of the training set, and $M << N$ [10, 11]. One family of approximations is based on a random feature (RF) space, where the RF space is determined from spectral frequencies [2, 7]. Examples include the sparse trigonometric expansions for the Radial Basis Function (RBF) covariance and the Rectified Linear Unit (ReLU) functions for the ARC-COSINE covariance. Two major components of the RF-based functions are their random features and parameter vectors. We can view the random features as being equivalent to basis functions and the parameter vectors to be weights associated with the basis functions.

Most of the existing literature, however, focuses on learning parameter vectors, but leave the random features aside by pre-selecting only one set of random features (i.e., one set of basis functions) [1, 9]. By contrast, in this paper we aim at finding the optimal set of random features dynamically. Some ensemble methods have been proposed to leverage different candidate RFs but they suffer degeneration with outcomes that only one candidate is left after several iterations [1, 8]. Further, learners are expected to perform sequentially when the data are received successively as is the case in many important applications [3]. This motivates a sequential learning approach dealing with cases where the processing of the received data have to be completed often before the next data are received. Also, in many practical settings, the optimal function is not deterministic but changes over time with unknown dynamics. In a number of fields like in robotics and computational finance, the cumulative changes of the unknown functions over time are significant, even though they may slowly change from one instant to the next [12, 14]. Thus, it is clear that such scenarios would require a methodology that would be capable of tracking functions as they change with time.

To overcome the limitations of the existing methods, in this paper, we propose a sequential time-varying RF-based GPs. The methodology is motivated by particle filtering that treats the functions as particles. From the theory of particle filtering (PF) [4], we know that the time-varying random features and parameters can be learnt jointly and sequentially. In this paper, we use PF on a model where the states of the model are GPs. Compared with benchmark methods, our approach searches for the optimal set of spectral frequencies over time and avoids degeneration and thereby it stabilizes the learning process.

The main contributions of this paper are as follows:

- proposal of a novel sequential scheme for estimation of time-varying random features (rather than time-invariant ones as in the existing literature), and
- relaxing the Gaussian likelihood assumption and using the Monte Carlo approach to enable learning with general likelihoods such as in problems of classification.

II. Background

In this section, we provide a brief review of PF and random Fourier feature-based GPs.

The authors thank the support of NSF under Award 2021002.

A. Particle Filtering

In PF, we aim at tracking a hidden process $\mathbf{x}_t \in \mathbb{R}^{d_x}$ of a state-space model given by

$$\text{transition probability}: \quad p(\mathbf{x}_t|\mathbf{x}_{t-1}), \tag{1}$$

$$\text{likelihood of } \mathbf{x}_t: \quad p(y_t|\mathbf{x}_t), \tag{2}$$

where t is a discrete time index, and $y_t \in \mathbb{R}$ is an observation process. The main objective is to obtain the filtering probability density function (pdf) $p(\mathbf{x}_t|y_{1:t})$ from $p(\mathbf{x}_{t-1}|y_{1:t-1})$.

The standard PF is implemented as follows. Suppose that at time $t-1$ the filtering density $p(\mathbf{x}_{t-1}|y_{1:t-1})$ is approximated by

$$p^M(\mathbf{x}_{t-1}|y_{1:t-1}) = \frac{1}{M}\sum_{m=1}^{M}\delta(\mathbf{x}_{t-1} - \mathbf{x}_{t-1}^{(m)}), \tag{3}$$

where the $\mathbf{x}_{t-1}^{(m)}$s are particles (samples) of $\mathbf{x}_{t-1}$ and $\delta(\cdot)$ is the Dirac delta function, and M is the number of particles. Then we obtain $p(\mathbf{x}_t|y_{1:t})$ from $p^M(\mathbf{x}_{t-1}|y_{1:t-1})$ as follows:

1) Generate the particles $\mathbf{x}_t^{(m)}$ by

$$\mathbf{x}_t^{(m)} \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(m)}), \tag{4}$$

2) Compute the weights of the particles $\mathbf{x}_t^{(m)}$ according to

$$w_t^{(m)} \propto p(y_t|\mathbf{x}_t^{(m)}), \tag{5}$$

The approximation of $p(\mathbf{x}_t|y_{1:t})$ is then given by

$$p^M(\mathbf{x}_t|y_{1:t}) = \sum_{m=1}^{M} w_t^{(m)}\delta(\mathbf{x}_t - \mathbf{x}_t^{(m)}). \tag{6}$$

3) Resample the particles using their weights $w_t^{(m)}$.

B. Random Fourier Feature-based Gaussian Processes

Suppose we intend to estimate a function f having a GP prior $\mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}'))$, with $k(\mathbf{x}, \mathbf{x}')$ being a kernel defining the similarity of d_x dimensional $\mathbf{x}$ and $\mathbf{x}'$. For any set of inputs $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\top$ in the domain, the function values $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)]^\top$ are Gaussian distributed, i.e.,

$$p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}), \tag{7}$$

where the elements $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ are the covariances over all pairs in $\mathbf{X}$. Specifically, a shift-invariant kernel $k(\cdot,\cdot)$ has the form $\sigma_k^2 k_0(\cdot,\cdot)$, where $k_0(\cdot,\cdot)$ is a standardized kernel, and σ_k^2 is the magnitude. The inverse Fourier transform of the standardized kernel is

$$k_0(\mathbf{x}, \mathbf{x}') = \int \pi(\mathbf{v}) e^{i\mathbf{v}^\top(\mathbf{x}-\mathbf{x}')} d\mathbf{v}, \tag{8}$$

where $\pi(\cdot)$ is the power spectral density (PSD) of the kernel. Next we define a real $2J \times 1$ random feature (RF) vector as

$$\boldsymbol{\phi}(\mathbf{x}) = \frac{1}{\sqrt{J}}[\sin(\mathbf{x}^\top\mathbf{v}^1), \cos(\mathbf{x}^\top\mathbf{v}^1), ..., \sin(\mathbf{x}^\top\mathbf{v}^J), \cos(\mathbf{x}^\top\mathbf{v}^J)]^\top, \tag{9}$$

where $\{\mathbf{v}^j\}_{j=1}^J$ are vectors sampled from the PSD of the kernel [7]. This enables us to approximate $k_0(\mathbf{x}, \mathbf{x}')$ by

$$\widehat{k}_0(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}^\top(\mathbf{x})\boldsymbol{\phi}(\mathbf{x}'). \tag{10}$$

Thus, the parametric approximation $\widehat{f}$ of f is defined by

$$\widehat{f}(\cdot) = \boldsymbol{\phi}^\top(\cdot)\boldsymbol{\theta}, \tag{11}$$

where the $2J$ dimensional parameter vector $\boldsymbol{\theta}$ has a Gaussian prior $\mathcal{N}(\boldsymbol{\theta}|\mathbf{0}, \sigma_\theta^2\mathbf{I}_{2J})$. As a result, the approximated GP prior of a realization $\widehat{\mathbf{f}}$ on any input $\mathbf{X}$ is given by

$$p(\widehat{\mathbf{f}}|\mathbf{X}) = \mathcal{N}(\widehat{\mathbf{f}}|\mathbf{0}, \widehat{\mathbf{K}}) \tag{12}$$

where $\widehat{\mathbf{K}} = \sigma_\theta^2\boldsymbol{\Phi}\boldsymbol{\Phi}^\top$, and $\boldsymbol{\Phi} = [\boldsymbol{\phi}(\mathbf{x}_1), \ldots, \boldsymbol{\phi}(\mathbf{x}_n)]^\top$. The posterior of $\widehat{f}$ is determined by the posterior of $\boldsymbol{\theta}$. Section III gives the approach to learn the posterior of $\boldsymbol{\theta}$ based on PF.

III. Sequential Time-varying Gaussian Processes

A. Problem Formulation

Suppose the observations y_t are expressed by

$$y_t = f_t(\mathbf{x}_t) + \epsilon_t \tag{13}$$

where the output $y_t \in \mathbb{R}$, the input $\mathbf{x}_t \in \mathbb{R}^{d_x}$, and where $\epsilon_t \in \mathbb{R}$ is an error (noise) that does not have to be Gaussian. The unknown function f_t has a GP prior $\mathcal{GP}(0, k_t(\mathbf{x}, \mathbf{x}'))$, with $k_t(\cdot,\cdot)$ being a shift-invariant kernel. Given a prior or a starting point of hyper-parameters, we determine the random feature vector $\boldsymbol{\phi}(\cdot)$ by (8) and (9). To handle a time-variant f_t, we propose that the parameter variable obeys a random walk. According to (11), the system is approximated by

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} \circ (\mathbf{1} + \boldsymbol{\eta}_t), \tag{14}$$

$$y_t = \widehat{f}_t(\mathbf{x}_t) + \epsilon_t = \boldsymbol{\phi}(\mathbf{x}_t)^\top\boldsymbol{\theta}_t + \epsilon_t \tag{15}$$

where $\boldsymbol{\eta}_t \in \mathbb{R}^{2J}$ is white noise, $\mathbf{1}$ is a $2J \times 1$ vector with all of its elements equal to one, and $a \circ b$ represents the Hadamard product. In other words, the transition process of $\widehat{f}_t$ is given by

$$\widehat{f}_t = \widehat{f}_{t-1} + \boldsymbol{\phi}^\top(\boldsymbol{\theta} \circ \boldsymbol{\eta}_t). \tag{16}$$

Compared with the standard random walk $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} + \boldsymbol{\eta}_t$, we propose the proportional random walk in (14). If an entry of $\boldsymbol{\theta}_t$ has optimal value close to 0 (or less than the standard deviation of $\boldsymbol{\eta}_t$), the standard random walk would not stay around the optimum. We wish to estimate the posterior of the parameter vector $\boldsymbol{\theta}_t$ sequentially. The prior of $\boldsymbol{\theta}_0$ is $\mathcal{N}(\mathbf{0}, \sigma_\theta^2\mathbf{I}_{2J})$ or is obtained from pre-training. Upon receiving y_t and $\mathbf{x}_t$, the posterior pdf of $\boldsymbol{\theta}_t$ is computed by the PF approach. First we sample M parameter vectors $\boldsymbol{\theta}_0^{(m)}$ from the prior at time 0, where $m = 1, ..., M$. Then we proceed as explained in the following two subsections.

B. Prediction

Suppose we have sampled M parameter vectors $\boldsymbol{\theta}_{t-1}^{(i)}$ at time $t-1$. The predictive pdf of $\widehat{f}(\mathbf{x}_t)$ can be obtained by

$$\begin{aligned} p(\widehat{f}_t(\mathbf{x}_t)|y_{1:t-1}, \mathbf{x}_{1:t-1}) &= \int p(\widehat{f}_t(\mathbf{x}_t)|\boldsymbol{\theta}_{t-1}) \\ &\quad \times p(\boldsymbol{\theta}_{t-1}|y_{1:t-1}, \mathbf{x}_{1:t-1}) d\boldsymbol{\theta}_{t-1} \\ &\approx \frac{1}{M} \sum_{m=1}^{M} \delta\left(\widehat{f}_t(\mathbf{x}_t) - \boldsymbol{\phi}(\mathbf{x}_t)^\top \boldsymbol{\theta}_{t-1}^{(m)}\right). \end{aligned}$$

Thus, the predictive pdf of y_t is given by

$$\begin{aligned} p(y_t|y_{1:t-1}) &= \int p(y_t|\widehat{f}_t(\mathbf{x}_t)) p(\widehat{f}_t(\mathbf{x}_t)|y_{1:t-1}) d\widehat{f}_t(\mathbf{x}_t) \\ &\approx \frac{1}{M} \sum_{m=1}^{M} p\left(y_t|\boldsymbol{\phi}(\mathbf{x}_t)^\top \boldsymbol{\theta}_{t-1}^{(m)}\right), \end{aligned} \tag{17}$$

C. Filtering

We generate particles of the parameter vectors $\boldsymbol{\theta}_t^{(m)}$ by drawing $\boldsymbol{\eta}_t^{(m)}$ and using (14). Upon receiving y_t, we assign the weights for each particle $\boldsymbol{\theta}_t^{(m)}$ by the formula

$$w_t^{(m)} \propto p(y_t|\widehat{f}_t^{(m)}) = p(y_t|\boldsymbol{\theta}_t^{(m)}, \mathbf{x}_t). \tag{18}$$

After normalizing the weights, the minimum mean square estimate (MMSE) of $\boldsymbol{\theta}_t$ is obtained by

$$\widehat{\boldsymbol{\theta}}_t = \sum_{i=1}^{N} w_t^{(i)} \boldsymbol{\theta}_t^{(i)}. \tag{19}$$

The approximation of the posterior of $p(\boldsymbol{\theta}_t|y_t)$ is

$$p^M(\boldsymbol{\theta}_t|y_t) \approx \sum_{m=1}^{M} w_t^{(m)} \delta(\boldsymbol{\theta}_t - \boldsymbol{\theta}_t^{(m)}). \tag{20}$$

Finally, we resample M new particles of $\boldsymbol{\theta}_t$ from (20) and use them for propagation to obtain $\boldsymbol{\theta}_{t+1}^{(m)}$. Our sequential PF algorithm has complexity $\mathcal{O}(TJM)$.

IV. Dynamic Random Features

In this section, we consider dynamic random feature variable $\boldsymbol{\phi}_t(\cdot)$ vectors rather than time-invariant ones. If the hyper-parameters of a GP are time-varying, then the power spectral density $\pi_t(\mathbf{v})$ of f_t has time-varying parameters. Accordingly, the corresponding random variables $\{\mathbf{v}_t^j\}_{j=1}^J$ of $\widehat{f}_t$ are dynamic. We model the random variables $\mathbf{v}_t^j$ following the random walk model

$$\mathbf{v}_t^j = \mathbf{v}_{t-1}^j \circ (\mathbf{1} + \boldsymbol{\zeta}_t^j), \tag{21}$$

where $\boldsymbol{\zeta}_t^j$ is white noise. As a result, the random feature $\boldsymbol{\phi}_t$ and hence the RF-based GP $\widehat{f}_t$ varies with time nonlinearly by

$$\begin{aligned} \boldsymbol{\phi}_t|\mathbf{v}_t &= g(\boldsymbol{\phi}_{t-1}|\mathbf{v}_{t-1}) = \boldsymbol{\phi}_{t-1}|\left(\mathbf{v}_{t-1} \circ (\mathbf{1} + \boldsymbol{\zeta}_t)\right), \\ \widehat{f}_t|\boldsymbol{\phi}_t &= h(\widehat{f}_{t-1}|\boldsymbol{\phi}_{t-1}) = g(\boldsymbol{\phi}_{t-1}|\mathbf{v}_{t-1})\boldsymbol{\theta}_t. \end{aligned} \tag{22}$$

In the initialization step, we sample K different sets $\{\mathbf{v}_0^j(k)\}_{j=1}^J$ from the prior $\pi_0(\mathbf{v})$, where $k = 1, \ldots, K$. Therefore the particles $\boldsymbol{\phi}_0^{(k)}$ and $\widehat{f}_0^{(k)}$ are determined by $\{\mathbf{v}_0^j(k)\}_{j=1}^J$ and (22). For the kth particle $\widehat{f}_0^{(k)}$, we sample M parameter vectors $\boldsymbol{\theta}_0^{(k,m)}$ from the prior distribution of $\boldsymbol{\theta}_0^{(k)}$, where $m = 1, \ldots, M$.

A. Prediction

Suppose that at time $t-1$ we have sampled the GPs $\widehat{f}_{t-1}^{(k)}$, $k = 1, 2, \ldots, K$. The predictive pdf of $\widehat{f}_t^{(k)}(\mathbf{x}_t)$ by particle k is obtined from

$$\begin{aligned} p(\widehat{f}_t^{(k)}(\mathbf{x}_t)|y_{1:t-1}) &= \int p(\widehat{f}_t^{(k)}(\mathbf{x}_t)|\boldsymbol{\theta}_{t-1}^{(k)}) \\ &\quad \times p(\boldsymbol{\theta}_{t-1}^{(k)}|y_{1:t-1}) d\boldsymbol{\theta}_{t-1}^{(k)} \\ &\approx \frac{1}{M} \sum_{m=1}^{M} \delta\left(\widehat{f}_t^{(k)}(\mathbf{x}_t) - \boldsymbol{\phi}_{t-1}^{(k)}(\mathbf{x}_t)^\top \boldsymbol{\theta}_{t-1}^{(k,m)}\right). \end{aligned}$$

Then the predictive pdf of y_t from particle k is given by

$$\begin{aligned} p(y_t|y_{1:t-1}, \mathbf{x}_{1:t-1}, \mathbf{x}_t, k) &= \int p(y_t|\widehat{f}_t^{(k)}(\mathbf{x}_t)) \\ &\quad \times p(\widehat{f}_t^{(k)}(\mathbf{x}_t)|y_{1:t-1}) d\widehat{f}_t^{(k)}(\mathbf{x}_t) \\ &\approx \frac{1}{M} \sum_{m=1}^{M} p\left(y_t|\boldsymbol{\phi}_{t-1}^{(k)}(\mathbf{x}_t)^\top \boldsymbol{\theta}_{t-1}^{(k,m)}\right). \end{aligned}$$

Consequently, we have the predictive pdf of y_t by all the GPs

$$\begin{aligned} &p(y_t|y_{1:t-1}, \mathbf{x}_{1:t-1}, \mathbf{x}_t) \\ &= \sum_{k=1}^{K} p(k|y_{1:t-1}, \mathbf{x}_{1:t-1}) \times p(y_t|y_{1:t-1}, \mathbf{x}_{1:t-1}, \mathbf{x}_t, k) \\ &\approx \sum_{k=1}^{K} p(k|y_{1:t-1}, \mathbf{x}_{1:t-1}) \times \sum_{m=1}^{M} w_{t-1}^{(m)} p(y_t|\boldsymbol{\phi}_{t-1}^{(k)}(\mathbf{x}_t)^\top \boldsymbol{\theta}_{t-1}^{(k,m)}). \end{aligned} \tag{23}$$

B. Filtering

Before y_t arrives, we sample $\widehat{f}_t^{(k)}, \boldsymbol{\phi}_t^{(k)}, \boldsymbol{\theta}_t^{(k,m)}$ using $\widehat{f}_{t-1}^{(k)}, \boldsymbol{\phi}_{t-1}^{(k)}, \boldsymbol{\theta}_{t-1}^{(k,m)}$ via (14), (21) and (22). The PF approach assigns the weight for each approximated GP $\widehat{f}_t^{(k)}$ by the likelihood

$$w_t^{(k)} \propto p(y_t|\widehat{f}_t^{(k)}) = \frac{1}{M} \sum_{i=1}^{M} p\left(y_t|\boldsymbol{\phi}_t^{(k)}(\mathbf{x}_t)^\top \boldsymbol{\theta}_t^{(k,m)}\right). \tag{24}$$

The estimated GP at time instant t is given by

$$\widehat{f}_t = \sum_{k=1}^{K} w_t^{(k)} \widehat{f}_t^{(k)}. \tag{25}$$

Next we resample K GPs $\widehat{f}^{(k)}$ from (25). After resampling the GPs, the parameter vector $\boldsymbol{\theta}_t^{(k,m)}$ is attached to the newly resampled GPs $\widehat{f}_t^{(k)}$, which are updated as shown in Section III. Our dynamic random feature has complexity of $\mathcal{O}(TJMK)$, and is presented by Algorithm 1.

Algorithm 1: Dynamic Random Feature

for $k = 1$ *to* K **do**
- Sample $\mathbf{v}_0^m(k)$ from $\pi_0(\mathbf{v})$;
- Construct random feature $\boldsymbol{\phi}_0^{(k)}(\cdot)$ via (9);
- Initialize the $\widehat{f}_0^{(k)}$ and their weights as $w_0^{(k)} = 1/K$;
- **for** $i = 1$ *to* N **do**
  - Sample $\boldsymbol{\theta}_0^{(k,i)} \sim p(\boldsymbol{\theta}_0)$;
  - Initialize the weight of $\boldsymbol{\theta}_0^{(k,i)}$ as $w_0^{(k,i)} = 1/N$;

for $t = 1$ *to* T **do**
- **Prediction:**
- Predict y_t via (23);
- **Transition:**
- Sample $\widehat{f}_t^{(k)}, \boldsymbol{\theta}_t^{(k,j)}$ from (14), (21) and (22);
- **GP Filtering:**
- Assign a weight to the candidate GP $\widehat{f}_t^{(k)}$ by (24);
- Estimate the GP $\widehat{f}_t$ by (25);
- Resample the GPs $\widehat{f}_t^{(k)}$ based on their weights.
- **Parameter Variable Filtering:**
- **for** $k = 1$ *to* K **do**
  - Assign weights to the particles $\boldsymbol{\theta}_t^{(k,i)}$ by (18);
  - Estimate the parameter vector $\widehat{\boldsymbol{\theta}}_t^{(k)}$ by (19);
  - Resample new particles $\boldsymbol{\theta}_t^{(k,i)}$ from (20).

V. Experiments

We compare the proposed sequential time-varying GP with dynamic random features and an online ensemble GP (O-EGP) [8] using the normalized mean square error (nMSE) defined as

$$nMSE_T := \frac{1}{T\sigma_y^2}\sum_{t=1}^{T}(y_t - \widehat{y}_t)^2, \tag{26}$$

where σ_y^2 is the sample variance of the output $y_{1:T}$, and $\widehat{y}_t$ is the predicted output at time t. O-EGP is an online method based on a Bayesian update, which preselects the sets of basis functions $\boldsymbol{\phi}$ and assumes that the observations have Gaussian likelihoods. Both time-invariant and time-varying functions are tested.

A. Synthetic test with a time-invariant function

The data were generated as in [6], where

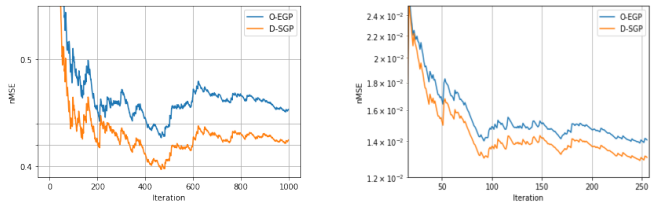
$$y_t = \frac{x_t}{25} + \frac{2x_t \cdot \cos(x_t)}{1 + x_t^2} + \epsilon_t, \tag{27}$$

where $\epsilon_t \sim \mathcal{N}(0, 0.1)$. In total we generated 2000 input-output pairs, where the inputs were randomly selected from the interval [-10,10]. We took the first 1000 pairs as a training set and the other half as a testing set. We employed the radial basis function (RBF) kernel, which has a power spectral density

$$\pi(v) = \sqrt{2\pi l^2}\exp^{-2\pi^2 l^2 v^2}, \tag{28}$$

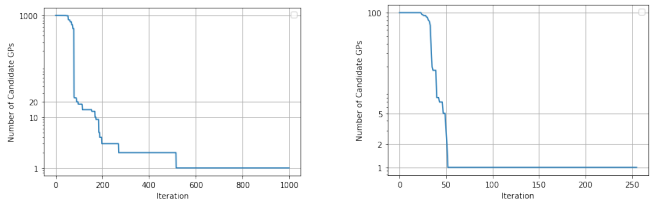
where the length scale l was learnt during training. Our model was set to have $K = 1000$ particle GPs $\widehat{f}_t^{(k)}$, for each particle GP $\widehat{f}_t^{(k)}$ $M = 1000$ particles of parameter variables $\boldsymbol{\theta}_t^{(k,m)}$, and $J = 50$ spectral frequencies.

The performance in terms of nMSE is plotted in Fig. 1(a). The proposed dynamic sequential GP model (D-SGP) outperformed the competing O-EGP by around 6% reduction of nMSE after convergence. Moreover, in our experiments the O-EGP always encountered degeneration in the number of candidate GPs (i.e., the sets of $\boldsymbol{\phi}$). At the end, there was always one GP that had a weight practically equal to one. (cf. Fig. 2(a)). By contrast, our approach did not experience degeneration because of the constant number of GP particles via transition provided by (22).



(a) (b)

Fig. 1. Log scale nMSE plots on syntehtic data generated by (a) a time-invariant function, and (b) a time-varying function.



(a) (b)

Fig. 2. Number of candidate GPs of O-EGP under (a) a time-invariant function, and (b) a time-varying function.

B. Synthetic test with a time-varying function

In this experiment, we generated synthetic signals that were used in [5] and that represented a superposition of a sinusoid and chirp signals embedded in zero mean white Gaussian noise. More specifically, the synthesized observations were obtained by

$$y_t = \cos(\omega_1 t) + \cos(\omega_2(t)t) + \cos(\omega_3(t)t) + \epsilon_t, \tag{29}$$

where $\omega_1 = \frac{2\pi}{4\cdot 512}$, $\omega_2(t) = \frac{2\pi t}{2\cdot 512^2}$, $\omega_3(t) = \frac{2\pi(512-t)}{2\cdot 512^2}$, $\epsilon_t \sim \mathcal{N}(0, 0.01)$, and where $t = 0, \ldots, 511$. We used the first 256 signal samples for training, and the remaining samples for testing. In light of the data size, our model was set to have $K = 100$ GP particles $\widehat{f}_t^{(k)}$, $M = 1000$ particles of the parameter vector $\boldsymbol{\theta}_t^{(k,m)}$, and $J = 10$ spectral frequencies.

The performance is presented in Fig. 1(b). Our model achieved over 8% reduction in nMSE compared with the

benchmark O-EGP model. The degeneration problem of O-EGP persisted in this experiment as well (cf. Fig. 2(b)).

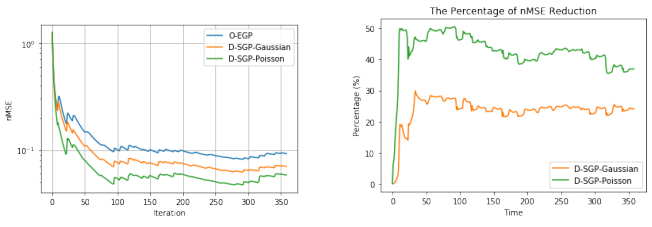
C. Real data with non-Gaussian likelihood

The bike sharing data set comes from the UCI machine learning repository,[1] which is time-varying and with non-Gaussian likelihoods. The outputs are the counts of casual users (hourly) in a city and are, thus, integers. The inputs are five-dimensional, including temperature in Celsius, feeling temperature in Celsius, humidity, wind speed, and hour. The former four variables are normalized while the hour variable is an integer that takes values from 0 to 23. A natural model of the conditional distribution of outputs is the Poisson distribution. Consequently, we assign the conditional probability mass function $p(y_t|\widehat{\alpha}_t)$ as a Poisson distribution,

$$P(y_t|\widehat{\alpha}_t) = \frac{\widehat{\alpha}_t^{y_t} e^{-\widehat{\alpha}_t}}{y_t!}. \tag{30}$$

In order to compute the particle weights, we estimated $\widehat{\alpha}_t$ with $\widehat{f}_t^{(m)}(\mathbf{x}_t)$ or $\widehat{f}_t^{(k)}(\mathbf{x}_t)$ in Eq. (18) or (24), respectively. As benchmarks, O-EGP and D-SGP with Gaussian likelihoods were both applied. We chose the first 120 samples as a training set and the following 360 samples as a testing set, since the period was 120 hours. The setting of this Poisson model was set to be $K = 100, M = 100$, and $J = 10$ for O-EGP and D-SGP with Gaussian and Poisson likelihoods.

The nMSE plot for all three models is shown in Fig. 3(a). D-SGP with a Poisson likelihood (D-SGP-Poisson) outperformed both D-SGP with a Gaussian likelihood (D-SGP-Gaussian) and O-EGP. In Fig. 3(b), we show the percentage of nMSE reduction. Our proposed method achieved over 40% and 15% reduction compared to O-EGP and D-SGP-Gaussian, respectively. The 15% gap between D-SGP-Poisson and D-SGP-Gaussian benefits from the flexibility of the likelihood function in the proposed model.



(a) (b)

Fig. 3. (a) Log scale nMSE plots on the bike sharing data, and (b) Percentage of nMse reduction.

VI. Conclusions

In this paper, we proposed sequential RF-based GP learners that can infer both time-varying random features and parameter variables with any computable likelihood. The learners are based on GPs and PF, where the GPs are treated as latent states. The experimental results illustrate good performance with synthesized and real data for both time-invariant and time-varying functions.

[1]https://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset.

References

[1] C.-A. Cheng and B. Boots. Incremental variational sparse Guassian process regression. In *Advances in Neural Information Processing Systems*, pages 4410–4418, 2016.

[2] K. Cutajar, E. V. Bonilla, P. Michiardi, and M. Filippone. Random feature expansions for deep Guassian processes. In *International Conference on Machine Learning*, pages 884–893. PMLR, 2017.

[3] T. G. Dietterich. Machine learning for sequential data: A review. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 15–30. Springer, 2002.

[4] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez. Particle filtering. *IEEE Signal Processing magazine*, 20(5):19–38, 2003.

[5] K. B. Eom. Analysis of acoustic signatures from moving vehicles using time-varying autoregressive models. *Multidimensional Systems and Signal Processing*, 10(4):357–378, 1999.

[6] M. F. Huber. Recursive Guassian process: On-line regression and learning. *Pattern Recognition Letters*, 45:85–91, 2014.

[7] M. Lázaro-Gredilla, J. Quiñonero-Candela, C. E. Rasmussen, and A. R. Figueiras-Vidal. Sparse spectrum Guassian process regression. *The Journal of Machine Learning Research*, 11:1865–1881, 2010.

[8] Q. Lu, G. Karanikolas, Y. Shen, and G. B. Giannakis. Ensemble Gaussian processes with spectral features for online interactive learning with scalability. In *International Conference on Artificial Intelligence and Statistics*, pages 1910–1920, 2020.

[9] Y. Shen, T. Chen, and G. B. Giannakis. Random feature-based online multi-kernel learning in environments with unknown dynamics. *The Journal of Machine Learning Research*, 20(1):773–808, 2019.

[10] E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems*, pages 1257–1264, 2006.

[11] M. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574, 2009.

[12] J. M. Wang, D. J. Fleet, and A. Hertzmann. Guassian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):283–298, 2007.

[13] C. K. Williams and C. E. Rasmussen. *Guassian Processes for Machine Learning*, volume 2. MIT Press Cambridge, MA, 2006.

[14] Y. Wu, J. M. Hernández-Lobato, and Z. Ghahramani. Guassian process volatility model. In *Advances in Neural Information Processing Systems*, pages 1044–1052, 2014.