Neurocomputing 000 (2017) 1–14

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Bayesian inference for time-varying applications: Particle-based Gaussian process approaches

Yali Wang[a,*], Brahim Chaib-draa[b]

[a] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
[b] Department of Computer Science and Software Engineering, Laval University, Canada

ARTICLE INFO

Article history:
Received 24 June 2016
Revised 1 December 2016
Accepted 31 January 2017
Available online xxx

Communicated by Shiliang Sun

Keywords:
Gaussian process
Time-varying applications
Bayesian inference
Sequential Monte Carlo
Non-stationarity
Heteroscedasticity

ABSTRACT

Gaussian process (GP) is a popular non-parametric model for Bayesian inference. However, the performance of GP is often limited in temporal applications, where the input–output pairs are sequentially-ordered, and often exhibit time-varying non-stationarity and heteroscedasticity. In this work, we propose two particle-based GP approaches to capture these distinct temporal characteristics. Firstly, we make use of GP to design two novel state space models which take the temporal order of input–output pairs into account. Secondly, we develop two sequential-Monte-Carlo-inspired particle mechanisms to learn the latent function values and model parameters in a recursive Bayesian framework. Since the model parameters are time-varying, our approaches can model non-stationarity and heteroscedasticity of temporal data. Finally, we evaluate our proposed approaches on a number of challenging time-varying data sets to show effectiveness. By comparing with several related GP approaches, we show that our particle-based GP approaches can efficiently and accurately capture temporal characteristics in time-varying applications.

1. Introduction

Gaussian process (GP) is a popular Bayesian nonparametric model due to its elegant inference framework [1]. However, the performance of GP is often limited in temporal applications [2–4], mainly because of two following reasons. *First*, GP is a batch modeling approach which may be not efficient to make online prediction for the sequentially-ordered temporal data sets [2,4]. *Second*, it is difficult for GP to capture distinct characteristics such as non-stationarity and heteroscedasticity which often exist in the temporal applications [1,5].

To model temporal input–output data pairs sequentially, several online variants of GP have been investigated by designing autoregressive models [6]; local online GP approaches [2,7]; Bayesian online learning with sparsification [4,8–10]; GP-based state space models [11–14] with different Bayesian approximation techniques such as Kalman filter [15,16], assumed density filter [17], Monte Carlo sampling [18–20]. However, the model parameters in these GP approaches are often assumed to be time-invariant. As a result, it may be restricted for these approaches to model time-varying non-stationarity and heteroscedasticity.

In general, non-stationarity refers to the input-dependent smoothness, where the correlation between any two latent function values does not only depend on the similarity between two corresponding input vectors, but it is also related to these two input vectors themselves [1]. Additionally, heteroscedasticity refers to the input-dependent noise, where the output noise is changed along with the location of the corresponding input vector [1]. To capture these distinct data characteristics, a number of GP extensions have been investigated by designing non-stationary covariance functions in GP [1,21,22], adding another GP on the output noise [22–25], warping GP with different nonlinear functions [26–29], developing mixtures of GP experts [30–32]. However, these batch GP approaches are often inefficient to make online prediction for time-varying applications.

To address the difficulties above, we propose two novel particle-based GP approaches in this paper, where one can make online prediction as well as model time-varying non-stationarity and heteroscedasticity in two efficient and accurate recursive Bayesian frameworks. *Firstly*, we take advantage of GP to develop two novel state space models (SSMs) in which the sequential order of temporal data pairs is modeled to make efficient online prediction. *Furthermore*, the parameters in our two SSMs are time-varying to capture non-stationarity and heteroscedasticity in the temporal data sets. Note that, the differences between our two SSMs are the different time-varying assumptions of these model parameters. This mainly accounts the trade-off between efficiency and accuracy

* Corresponding author.
E-mail addresses: yl.wang@siat.ac.cn (Y. Wang), chaib@ift.ulaval.ca (B. Chaib-draa).

http://dx.doi.org/10.1016/j.neucom.2017.01.072
0925-2312/

when performing online inference. *Finally*, based on our two time-varying SSMs, we respectively design two effective particle mechanisms to infer the latent function values and model parameters over time, in order to learn the distinct temporal characteristics in time-varying applications.

On one hand, our approaches are different from those online GP variants, since our approaches can model non-stationarity and heteroscedasticity in the temporal applications. This is mainly because we learn the model parameters over time. On the other hand, our approaches are different from those non-stationary/heteroscedastic GP variants, since our approaches can make efficient online prediction for temporal data sets. This is mainly credited to our novel GP-constructed SSMs with effective particle inference mechanisms.

The rest of this paper is organized as follows. In Section 2, we review the basics of GP. In Section 3, we introduce our particle-based GP approaches in detail. In Section 4, we evaluate our proposed approaches on five challenging time-varying applications, by comparing them with several relevant GP approaches. Finally, we conclude our paper in Section 5.

2. Background

In this section, we first introduce the definition of Gaussian process (GP). Then we review the standard GP regression approach. Finally, we briefly discuss a conventional way to model temporal data with GP regression.

2.1. Definition of Gaussian process

Gaussian process (GP) is *a collection of random variables, any finite number of which have a joint Gaussian distribution* [1]. It has been widely used as a Bayesian prior over the latent function, where the function values at any finite number of inputs are Gaussian-distributed random variables [1,33–35]. Specifically, a GP prior over the latent function $f(\mathbf{x})$ is denoted by

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \tag{1}$$

with a mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x}, \mathbf{x}')$ [1],

$$m(\mathbf{x}) = E[f(\mathbf{x})], \tag{2}$$

$$k(\mathbf{x}, \mathbf{x}') = E[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))], \tag{3}$$

where $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^{d_x}$ are any two d_x-dimension input vectors.

According to the definition of GP, one can obtain that the prior over the latent function values $f(X) = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_T)]^T$ at any T input vectors $X = [\mathbf{x}_1, \ldots, \mathbf{x}_T]^T$ is a jointly Gaussian distribution, i.e.,

$$p(f(X)) = \mathcal{N}(\mathbf{m}(X), K(X, X)), \tag{4}$$

where the mean vector $\mathbf{m}(X)$ is computed from the mean function $m(\mathbf{x})$,

$$\mathbf{m}(X) = \begin{bmatrix} m(\mathbf{x}_1) \\ \cdots \\ m(\mathbf{x}_T) \end{bmatrix}, \tag{5}$$

and the covariance matrix $K(X, X)$ is computed from the covariance function $k(\mathbf{x}, \mathbf{x}')$,

$$K(X, X) = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k(\mathbf{x}_1, \mathbf{x}_T) \\ \cdots & \cdots & \cdots \\ k(\mathbf{x}_T, \mathbf{x}_1) & \cdots & k(\mathbf{x}_T, \mathbf{x}_T) \end{bmatrix}. \tag{6}$$

In this work, we follow [1,36] to choose a widely-used GP prior,

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}) = 0, k(\mathbf{x}, \mathbf{x}')), \tag{7}$$

where the mean function is zero for simplicity.[1] Furthermore, the covariance function is a popular squared exponential (SE) kernel [1,36], i.e.,

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 e^{-0.5(\mathbf{x}-\mathbf{x}')^T L^{-1}(\mathbf{x}-\mathbf{x}')} = \sigma_f^2 k_\ell(\mathbf{x}, \mathbf{x}'), \tag{8}$$

where σ_f^2 is the amplitude parameter, $k_\ell(\mathbf{x}, \mathbf{x}') = e^{-0.5(\mathbf{x}-\mathbf{x}')^T L^{-1}(\mathbf{x}-\mathbf{x}')}$ is the unscaled covariance function in which L is diagonal with the length-scale parameter vector $\ell = [l_1, \ldots, l_{d_x}]^T$. In the following, we illustrate how to use GP to address the standard nonlinear regression task from a Bayesian view.

2.2. Gaussian process regression

Suppose that there is a training set $D = (X, \mathbf{y}) = \{(\mathbf{x}_t, y_t)\}_{t=1}^T$ with T input–output data pairs, where $\mathbf{x}_t \in \mathbb{R}^{d_x}$, $y_t \in \mathbb{R}$, $X = [\mathbf{x}_1, \ldots, \mathbf{x}_T]^T$, $\mathbf{y} = [y_1, \ldots, y_T]^T$. Each output is assumed to be generated from

$$y = f(\mathbf{x}) + \epsilon_y, \tag{9}$$

where the Gaussian noise is $\epsilon_y \sim \mathcal{N}(0, \sigma_y^2)$ with variance σ_y^2. Furthermore, the GP prior over the latent function is assumed to be $f(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}'))$ with the SE covariance function $k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 k_\ell(\mathbf{x}, \mathbf{x}')$ in Eq. (8). For convenience, we collect the parameters of the SE covariance function (σ_f^2, ℓ) and the noise variance σ_y^2 into a parameter vector $\Theta = [\sigma_f^2, \ell, \sigma_y^2]^T$.

Given the training set $D = (X, \mathbf{y})$ and M test inputs $X_\star = [\mathbf{x}_\star^1, \ldots, \mathbf{x}_\star^M]^T$, the regression task can be addressed by using GP in the following Bayesian manner. Specifically, predicting the test output vector $\mathbf{y}_\star$ and the model parameters Θ can be interpreted to learn the predictive distribution over $\mathbf{y}_\star$ and Θ,

$$p(\mathbf{y}_\star, \Theta | X_\star, X, \mathbf{y}) = p(\mathbf{y}_\star | X_\star, X, \mathbf{y}, \Theta) p(\Theta | X, \mathbf{y}). \tag{10}$$

Parameter learning by $p(\Theta|X, \mathbf{y})$: As $p(\Theta|X, \mathbf{y}) \propto p(\mathbf{y}|X, \Theta)$, a popular approach to infer Θ is to minimize the negative log likelihood with gradient optimization [1]

$$\begin{aligned} &-\log p(\mathbf{y}|X, \Theta) \\ &= \frac{1}{2}\mathbf{y}^T[K(X, X) + \sigma_y^2 I]^{-1}\mathbf{y} + \frac{1}{2}\log|K(X, X) + \sigma_y^2 I| + \frac{n}{2}\log 2\pi, \end{aligned} \tag{11}$$

where $K(X, X)$ is constructed by using Eq. (6). In practice, this approach works well [1] and thus we apply it for the standard GP regression in this paper.

Output inference by $p(\mathbf{y}_\star|X_\star, X, \mathbf{y}, \Theta)$: After Θ is learned, one can make prediction at test inputs $X_\star$ by using $p(\mathbf{y}_\star|X_\star, X, \mathbf{y}, \Theta)$. *Firstly*, due to the fact that $f(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}'))$ and the noise in Eq. (9) is Gaussian, the joint distribution over the training outputs $\mathbf{y}$ and latent function values at test inputs $f(X_\star) = [f(\mathbf{x}_\star^1), \ldots, f(\mathbf{x}_\star^M)]^T$ is Gaussian [1],

$$p(\mathbf{y}, f(X_\star)|X, X_\star, \Theta) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X, X) + \sigma_y^2 I & K(X, X_\star) \\ K(X_\star, X) & K(X_\star, X_\star) \end{bmatrix}\right), \tag{12}$$

where $K(X_\star, X)$ and $K(X_\star, X_\star)$ are constructed by using $X_\star$ and X in Eq. (6). *Secondly*, based on the conditional property of a joint multivariate Gaussian distribution (Appendix A of [1]), one can obtain that the conditional distribution $p(f(X_\star)|X_\star, X, \mathbf{y}, \Theta)$ is Gaussian,

$$p(f(X_\star)|X_\star, X, \mathbf{y}, \Theta) = \mathcal{N}(\mu_\star, \Sigma_\star), \tag{13}$$

with the following mean vector $\mu_\star$ and covariance matrix $\Sigma_\star$ which are computed from the corresponding joint distribution $p(\mathbf{y}, f(X_\star)|X, X_\star, \Theta)$ in Eq. (12) [1],

$$\mu_\star = K(X_\star, X)[K(X, X) + \sigma_y^2 I]^{-1}\mathbf{y}, \tag{14}$$

[1] Note that it is straightforward to choose other mean functions to do mathematical derivations without difficulties.

Please cite this article as: Y. Wang, B. Chaib-draa, Bayesian inference for time-varying applications: Particle-based Gaussian process approaches, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.01.072