Rao–Blackwellized Particle MCMC for Parameter Estimation in Spatio-Temporal Gaussian Processes

Roland Hostettler, Simo Särkkä, and Simon J. Godsill

DOI:

10.1109/MLSP.2017.8168171

2017 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 25–28, 2017, TOKYO, JAPAN

RAO–BLACKWELLIZED PARTICLE MCMC FOR PARAMETER ESTIMATION IN SPATIO-TEMPORAL GAUSSIAN PROCESSES

Roland Hostettler[⋆], *Simo Särkkä*[⋆], *and Simon J. Godsill*[†]

[⋆]Department of Electrical Engineering and Automation, Aalto University, Espoo, Finland
E-Mail: {roland.hostettler, simo.sarkka}@aalto.fi
[†]Department of Engineering, University of Cambridge, Cambridge, UK
E-Mail: sjg@eng.cam.ac.uk

ABSTRACT

In this paper, we consider parameter estimation in latent, spatio-temporal Gaussian processes using particle Markov chain Monte Carlo methods. In particular, we use spectral decomposition of the covariance function to obtain a high-dimensional state-space representation of the Gaussian processes, which is assumed to be observed through a nonlinear non-Gaussian likelihood. We develop a Rao–Blackwellized particle Gibbs sampler to sample the state trajectory and show how to sample the hyperparameters and possible parameters in the likelihood. The proposed method is evaluated on a spatio-temporal population model and the predictive performance is evaluated using leave-one-out cross-validation.

***Index Terms*—** Gaussian processes, statistical learning, Monte Carlo methods, parameter estimation

1. INTRODUCTION

Gaussian processes (GP) are a versatile Bayesian non-parametric modeling approach [1]. They have found widespread applications, for example in time series modeling in finance [2], meteorology [3], medical applications [4], and target tracking [5], to name a few. One well-known disadvantage is that the batch formulation in GP regression scales cubically with the number of training points. However, it has recently been shown that stationary, temporal and spatio-temporal GPs can be transformed into equivalent (infinite dimensional) linear state-space systems by decomposing the GP's spectral density. This can subsequently be used together with Kalman filtering and Rauch–Tung–Striebel smoothing [6–8], which greatly alleviates the computational burden.

In this work, we consider learning (i.e., estimation of the model parameters) of models where a latent, spatio-temporal process, observed indirectly through some proxy variable, is modeled using a GP. In particular, we consider models of the form

$$f(\mathsf{x}, t) \sim \mathcal{GP}(m(\mathsf{x}, t; \theta_f), k(\mathsf{x}, t, \mathsf{x}', t'; \theta_f)), \tag{1a}$$
$$y(t) \sim p(y(t) \mid f(\mathsf{x}, t), \theta_y), \tag{1b}$$

where $m(\mathsf{x}, t)$ is the GP's mean function (without loss of generality assumed to be zero for the remainder of this paper) and $k(\mathsf{x}, t, \mathsf{x}', t') = k(\Delta\mathsf{x}, \tau)$ with $\tau = t - t'$ and $\Delta\mathsf{x} = \mathsf{x} - \mathsf{x}'$ its stationary covariance function, both parametrized by the hyperparameters θ_f. Furthermore, $p(y(t) \mid f(\mathsf{x}, t), \theta_y)$ is the measurement likelihood, which is parametrized by parameters θ_y, and t and x denote the temporal and spatial variables, respectively.

The objective is then to infer the function values $f_n \triangleq f(\mathsf{x}, t_n)$ at times t_n (for $n = 1, \dots, N$) as well as the parameters θ_f and θ_y based on a set of measurements $y_{1:N} = \{y_1, \dots, y_N\}$. The most common approach to do this in batch settings is by maximizing the marginal likelihood of the data $p(y_{1:N} \mid \theta_f, \theta_y)$ using, for example, gradient-based methods, or Markov chain Monte Carlo (MCMC) [1]. These approaches can also readily be employed when using the equivalent state-space formulations and Kalman filtering [7]. When facing non-Gaussian likelihoods $p(y_n \mid f_n, \theta_y)$, one commonly has to resort to approximations such as expectation propagation or Laplace approximations [9–11]. These may overcome some of the difficulties involved with nonlinear non-Gaussian likelihoods, but introduce approximations to the posterior and may still be unfeasible for a large number of data points and require further approximations such as inducing points [12].

In this paper, we propose to use a fully Bayesian approach based on particle MCMC methods (particle Gibbs with ancestor sampling, PGAS [13, 14]), which can readily handle arbitrary likelihoods $p(y_n \mid f_n, \theta_y)$, together with the state-space representation of spatio-temporal GPs. This has the advantage of not approximating the posterior by an (implicitly) assumed density but rather approximating the true posterior using samples from it, while retaining the beneficial properties of the state-space formulation. Since the conversion procedure can yield high-dimensional state-space systems, Rao–Blackwellization of the conditionally linear substructure is used to alleviate the computational burden and increase scalability. The proposed method is conceptually similar to the methods proposed in [15] and [16]; however, it solves a different problem. The latter methods aim at estimating the parameters in state-space systems where the dynamic model and observation model are modeled using GPs, while the method proposed in this work aims at inferring the parameters in GPs on state-space form, observed through a known likelihood (up to some parametrization θ_y).

The remainder of this paper is organized as follows. Section 2 briefly reviews the procedure for converting spatio-temporal Gaussian processes to state-space models. The proposed method is introduced in Section 3, followed by numerical illustrations in Section 4. Some concluding remarks follow in Section 5.

2. CONVERSION OF GAUSSIAN PROCESSES

In this section, we briefly review the procedure for converting stationary GPs to state-space models. For a more detailed treatment, the reader is referred to [6–8].

The main idea of the conversion approach is to decompose the spectral density of the stationary Gaussian process which is given by the Fourier transform of the covariance function

$$k(\Delta \mathsf{x}, \tau) \Leftrightarrow S_f(\omega_{\mathsf{x}}, \omega_t)$$

into a white noise process with spectral density S_w and a linear system with frequency response $H(\mathrm{i}\,\omega_{\mathsf{x}}, \mathrm{i}\,\omega_t)$ such that [17]

$$S_f(\omega_{\mathsf{x}}, \omega_t) = S_w H(\mathrm{i}\,\omega_{\mathsf{x}}, \mathrm{i}\,\omega_t) H(\mathrm{i}\,\omega_{\mathsf{x}}, \mathrm{i}\,\omega_t)^*, \tag{2}$$

where the superscript $*$ denotes the complex conjugate. From the transfer function $H(\mathrm{i}\,\omega_{\mathsf{x}}, \mathrm{i}\,\omega_t)$ of the linear system, the corresponding infinite dimensional state-space representation can then readily be found using well known conversion techniques to obtain [7]

$$\begin{aligned}
\dot{x}(\mathsf{x}, t) &= Ax(\mathsf{x}, t) + Bw(\mathsf{x}, t), && \text{(3a)}\\
f(\mathsf{x}, t) &= Cx(\mathsf{x}, t), && \text{(3b)}\\
x(\mathsf{x}, 0) &\sim \mathcal{N}(0, P_0), && \text{(3c)}
\end{aligned}$$

where $w(\mathsf{x}, t)$ is the white random process with spectral density S_w as given by the decomposition in (2). Furthermore, A, B, and C are, in general, a matrix and vectors of linear operators, respectively, and are given through $H(\mathrm{i}\,\omega_{\mathsf{x}}, \mathrm{i}\,\omega_t)$. The covariance P_0 of the initial state $x(\mathsf{x}, 0)$ is given by the solution of the following continuous time Lyapunov equation

$$AP_0 + P_0A^\mathsf{T} + BS_wB^\mathsf{T} = 0. \tag{4}$$

Finally, discretizing (3) with respect to time yields

$$\begin{aligned}
x_n &= F_n x_{n-1} + q_n, && \text{(5a)}\\
f_n &= Cx_n, && \text{(5b)}\\
x_0 &\sim \mathcal{N}(0, P_0), && \text{(5c)}
\end{aligned}$$

where $F_n = \exp(A\Delta t)$ is the linear operator exponential with $\Delta t = t_n - t_{n-1}$, and $q_n \sim \mathcal{N}(0, Q_n)$ is white, Gaussian noise with covariance operator

$$Q_n = \int_0^{\Delta t} \exp(A(\Delta t - \tau)) BS_wB^\mathsf{T} \exp(A(\Delta t - \tau))^\mathsf{T} \mathrm{d}\tau. \tag{6}$$

As an example, consider a covariance function composed of the product of two Matérn kernels $k_\mathrm{M}(\cdot)$

$$k(\Delta \mathsf{x}, \tau) = k_\mathrm{M}(\Delta \mathsf{x}) k_\mathrm{M}(\tau), \tag{7}$$

with

$$k_\mathrm{M}(r) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}|r|}{l}\right)^\nu K_\nu\left(\frac{\sqrt{2\nu}|r|}{l}\right), \tag{8}$$

where σ^2, l, and ν are the kernel's hyperparameters, and $K_\nu(\cdot)$ is the modified Bessel function of the second kind of order ν.

Using the Fourier transform with respect to τ on (7)-(8) yields

$$\begin{aligned}
S(\Delta \mathsf{x}, \omega_t) &= k_\mathrm{M}(\Delta \mathsf{x}) \frac{2\sigma^2\sqrt{\pi}\lambda_t^{2\nu_t}\Gamma(\nu_t + 1/2)}{\Gamma(\nu_t)} \\
&\quad \times \left(\lambda_t^2 + \omega^2\right)^{-\nu_t + 1/2} \\
&= k_\mathrm{M}(\Delta \mathsf{x}) \frac{2\sigma^2\sqrt{\pi}\lambda_t^{2\nu_t}\Gamma(\nu_t + 1/2)}{\Gamma(\nu_t)} \\
&\quad \times (\lambda_t + \mathrm{i}\,\omega)^{-\nu_t + 1/2} (\lambda_t - \mathrm{i}\,\omega)^{-\nu_t + 1/2}
\end{aligned} \tag{9}$$

where we have used $\lambda_t = \sqrt{2\nu_t}/l_t$. Since (7) is a product, (9) can be converted without involving the spectral density with respect to $\Delta \mathsf{x}$ [6]. Furthermore, from (9) it is clear that the order of the linear system directly depends on the hyperparameter ν_t: Whenever $p = \nu_t - 1/2$ is a positive integer, the following companion form state-space representation can be obtained

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 1 \\ -\lambda_t^{p+1} & -a_p\lambda_t^p & \dots & -a_2\lambda_t^2 & -a_1\lambda_t \end{bmatrix}, \tag{10a}$$

$$B = \begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix}^\mathsf{T}, \tag{10b}$$

$$C = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}, \tag{10c}$$

where a_i is the ith binomial coefficient of $(\lambda_t + 1)^{p+1}$, and

$$S_w = k_\mathrm{M}(\Delta \mathsf{x}) \frac{2\sigma^2\lambda_t^{2\nu_t}\Gamma(\nu_t + 1/2)}{\Gamma(\nu_t)}, \tag{10d}$$

see [6] for details. Note that in this case, A, B, and C do not contain derivatives with respect to x, which is due to the covariance function being a product of a spatial and temporal kernel. In general, however, this is not the case and derivatives may appear for non-product covariance functions.

Finally, if the decomposition (2) can not be done exactly, it can be approximated by, for example, using rational approximations of $H(\mathrm{i}\,\omega_{\mathsf{x}}, \mathrm{i}\,\omega_t)$. Examples for when this is necessary include when $\nu_t - 1/2$ in the Matérn covariance is not a positive integer or the popular squared exponential covariance function [6, 18].

Algorithm 1 Particle Gibbs Sampler

```
Input: $\theta_f^0$, $\theta_y^0$, $y_{1:N}$
Output: $x_{0:N}^{1:K}$, $\theta_f^{1:K}$, $\theta_y^{1:K}$
1: for $k = 1, \dots, K-1$ do
2:     Sample $x_{0:N}^k \sim p(x_{0:N} \mid y_{1:N}, \theta_f^{k-1}, \theta_y^{k-1})$
3:     Sample $\theta_f^k \sim p(\theta_f \mid y_{1:N}, x_{0:N}^k, \theta_y^{k-1})$
4:     Sample $\theta_y^k \sim p(\theta_y \mid y_{1:N}, x_{0:N}^k, \theta_f^k)$
5: end for
```

3. STATE AND PARAMETER ESTIMATION

In this section, we develop the proposed approach for estimating the posterior distribution of the parameters and the state, $p(\theta_f, \theta_y, x_{0:N} \mid y_{1:N})$. The approach is based on Gibbs sampling and particle MCMC methods (see [13, 19]) where the state trajectory $x_{0:N}$, GP hyperparameters θ_f, and likelihood parameters θ_y (if any) are sampled in turn, each conditioned on the other parameters and the data $y_{1:N}$. This approach has the advantage of being able to accommodate a broad class of possibly nonlinear non-Gaussian models and thus can overcome some of the limitations of approaches maximizing the marginal likelihood $p(y_{1:N} \mid \theta_f, \theta_y)$, at the expense of being computationally more demanding compared to Kalman filtering-based approaches. The basic sampler is shown in Algorithm 1 and each of its components is discussed below.

3.1. State Sampling

Sampling the states is achieved by using the particle Gibbs with ancestor sampling (PGAS) kernel introduced in [13]. The basic idea in

PGAS is to construct a particle filter that approximates the smoothing density $p(x_{0:N} \mid y_{1:N}, \theta_f, \theta_y)$ (for brevity, we will drop the conditioning on θ_f and θ_y throughout the remainder of this subsection) by generating a set of weighted trajectories $\{w_N^m, x_{0:N}^m\}_{m=1}^M$ from which a new trajectory is sampled with probability w_N^m. By introducing an a priori known, deterministic seed trajectory $\bar{x}_{0:N}$, it can be shown that the resulting algorithm indeed is an invariant sampler over $p(x_{0:N} \mid y_{1:N}, \theta_f, \theta_y)$, see [13] or [19] for details.

As noted in Section 2 the conversion of the GP to state-space form generally produces an infinite dimensional system. In practice, this has to be approximated by using a discrete set of training and prediction points in the spatial domain x. This turns the system from an infinite dimensional into a high-dimensional system instead. Thus, the dimensionality becomes challenging when sampling the states using sequential Monte Carlo methods.

Fortunately, the resulting system (5) exhibits a large linear substructure, which can be exploited. In particular, the states can be divided into nonlinear states s_n (the GP f_n at the training points) and linear states z_n (the GP's derivatives $\partial^i f_n / \partial t^i$ at the training points as well as the prediction points). Thus, the dynamic model can be written as

$$\begin{aligned} &p(s_n, z_n \mid s_{n-1}, z_{n-1}) \\ &\quad = \mathcal{N}\left(\begin{bmatrix} s_n \\ z_n \end{bmatrix}; \begin{bmatrix} g_{n-1} \\ h_{n-1} \end{bmatrix} + \begin{bmatrix} G \\ H \end{bmatrix} z_{n-1}, \begin{bmatrix} Q_n^s & Q_n^{sz} \\ Q_n^{zs} & Q_n^z \end{bmatrix}\right), \end{aligned} \tag{11}$$

where $g_{n-1} \triangleq g(s_{n-1})$ and $h_{n-1} \triangleq h(s_{n-1})$. Then, Rao–Blackwellization can be used to reduce the dimensionality of the state that has to be targeted using Monte Carlo sampling as follows [20, 21]. Starting from the joint filtering density, we obtain

$$p(z_n, s_{0:n} \mid y_{1:n}) = p(z_n \mid s_{0:n}, y_{1:n}) p(s_{0:n} \mid y_{1:n}), \tag{12}$$

where $p(z_n \mid s_{0:n}, y_{1:n})$ is analytically tractable using a Kalman filter and $p(s_{0:n} \mid y_{1:n})$ will be the target density for sequential Monte Carlo sampling. This yields

$$\begin{aligned} &p(z_n, s_{0:n} \mid y_{1:n}) \\ &\quad \approx \sum_{m=1}^{M} w_n^m \mathcal{N}(z_n; \hat{z}_n^m, P_n^z) \delta(s_{0:n} - s_{0:n}^m) \end{aligned} \tag{13}$$

where $\delta(\cdot)$ denotes Dirac's delta function, $s_{0:n}^m$ is the mth marginal trajectory, w_n^m the corresponding weight, $\hat{z}_n^m$ the conditional mean of the linear states, and P_n^z their covariance. Note that since neither Q_n nor G or H depend on s_{n-1}, the covariance P_n^z does not depend on the state trajectory $s_{0:n}$ either.

3.1.1. Linear States

Assume that at time t_{n-1}, the density $p(z_{n-1} \mid s_{0:n-1}, y_{1:n-1})$ is given by

$$p(z_{n-1} \mid s_{0:n-1}, y_{1:n-1}) = \mathcal{N}(z_{n-1}; \hat{z}_{n-1}, P_{n-1}^z) \tag{14}$$

and that s_n has been sampled. Then, the update for z_n reduces to a Kalman filter prediction

$$\begin{aligned} p(z_n \mid s_{0:n}, y_{1:n}) &\propto p(y_n \mid s_{0:n}, z_n) p(z_n \mid s_{0:n}, y_{1:n-1}) \\ &\propto p(z_n \mid s_{0:n}, y_{1:n-1}) \end{aligned} \tag{15}$$

since the likelihood is independent of z_n. Considering the joint density of z_n and s_n yields

$$\begin{aligned} &p(z_n, s_n \mid s_{0:n-1}, y_{1:n-1}) \\ &\quad = \int p(z_n, s_n \mid z_{n-1}, s_{n-1}) p(z_{n-1} \mid s_{0:n-1}, y_{1:n-1}) \mathrm{d}z_{n-1} \\ &\quad = \mathcal{N}\left(\begin{bmatrix} s_n \\ z_n \end{bmatrix}; \begin{bmatrix} g_{n-1} \\ h_{n-1} \end{bmatrix} + \begin{bmatrix} G \\ H \end{bmatrix} \hat{z}_{n-1}, Q_n + \begin{bmatrix} G \\ H \end{bmatrix} P_{n-1}^z \begin{bmatrix} G \\ H \end{bmatrix}^\mathsf{T}\right). \end{aligned} \tag{16}$$

Thus, by conditioning on s_n, it follows that

$$\begin{aligned} p(z_n \mid s_{0:n}, y_{1:n}) &= p(z_n \mid s_{0:n}, y_{1:n-1}) \\ &= \mathcal{N}(z_n; \hat{z}_n, P_n^z), \end{aligned} \tag{17}$$

with

$$\hat{z}_n = h_{n-1} + H\hat{z}_{n-1} + K(s_n - g_{n-1} - G\hat{z}_{n-1}), \tag{18a}$$

$$P_n^z = Q_n^z + HP_{n-1}^z H^\mathsf{T} - KSK^\mathsf{T}, \tag{18b}$$

$$S = Q_n^s + GP_{n-1}^z G^\mathsf{T}, \tag{18c}$$

$$K = (Q_n^{zs} + HP_{n-1}^z G^\mathsf{T}) S^{-1}. \tag{18d}$$

3.1.2. Nonlinear States

The marginal density $p(s_{0:n} \mid y_{1:n})$ in (12) is targeted using sequential Monte Carlo sampling. It can be factorized as follows

$$\begin{aligned} &p(s_{0:n} \mid y_{1:n}) \\ &\quad \propto p(y_n \mid s_{0:n}, y_{1:n-1}) p(s_{0:n} \mid y_{1:n-1}) \\ &\quad = p(y_n \mid s_n) p(s_n \mid s_{0:n-1}, y_{1:n-1}) p(s_{0:n-1} \mid y_{1:n-1}). \end{aligned} \tag{19}$$

The middle term is the marginalized dynamics for s_n and can be found from

$$\begin{aligned} &p(s_n \mid s_{0:n-1}, y_{1:n-1}) \\ &\quad = \int p(s_n \mid z_{n-1}, s_{n-1}) p(z_{n-1} \mid s_{0:n-1}, y_{1:n-1}) \mathrm{d}z_{n-1} \\ &\quad = \mathcal{N}(s_n; g_{n-1} + G\hat{z}_{n-1}, Q_n^s + GP_{n-1}^z G^\mathsf{T}). \end{aligned} \tag{20}$$

Note that $\hat{z}_{n-1}$ depends on the whole trajectory $s_{0:n-1}$ (see (17)) and hence, so does $p(s_n \mid s_{0:n-1}, y_{1:n-1})$, which yields a non-Markovian system.

3.1.3. Ancestor Weights

The final step is to derive the ancestor weights used for sampling the seed particle's ancestor trajectory. These are given by [13]

$$\begin{aligned} \bar{w}_{n|N} &= w_{n-1} \frac{p(s_{0:n-1}, \bar{s}_{n:N} \mid y_{1:N})}{p(s_{0:n-1} \mid y_{1:n-1})} \\ &\propto w_{n-1} p(y_{n:N} \mid \bar{s}_{n:N}) p(\bar{s}_{n:N} \mid s_{0:n-1}, y_{1:n-1}) \\ &\propto w_{n-1} p(\bar{s}_{n:N} \mid s_{0:n-1}, y_{1:n-1}) \\ &= w_{n-1} \prod_{j=n}^{N} p(\bar{s}_j \mid \bar{s}_{n:j-1}, s_{0:n-1}, y_{1:n-1}), \end{aligned} \tag{21}$$

where $\bar{s}_{n:n-1} = \{\}$ is the empty set (for $j = n$). The marginal future predictions $p(\bar{s}_j \mid \bar{s}_{n:j-1}, s_{0:n-1}, y_{1:n-1})$ depend on the complete history of $s_{0:n-1}$ due to the non-Markovianity introduced by the marginalization. However, they can be calculated recursively as follows. First, note that

$$\begin{aligned}
&p(\bar{s}_j \mid \bar{s}_{n:j-1}, s_{0:n-1}, y_{1:n-1}) \\
&\quad = \int p(\bar{s}_j \mid z_{j-1}, \bar{s}_{j-1}) \\
&\qquad \times p(z_{j-1} \mid \bar{s}_{n:j-1}, s_{0:n-1}, y_{1:n-1}) \mathrm{d}z_{j-1}.
\end{aligned} \tag{22}$$

The second term of the integrand in (22) can be found from the joint density

$$\begin{aligned}
&p(z_{j-1}, \bar{s}_{j-1} \mid \bar{s}_{n:j-2}, s_{0:n-1}, y_{1:n-1}) \\
&\quad = \int p(z_{j-1}, \bar{s}_{j-1} \mid z_{j-2}, \bar{s}_{j-2}) \\
&\qquad \times p(z_{j-2} \mid \bar{s}_{n:j-2}, s_{0:n-1}, y_{1:n-1}) \mathrm{d}z_{j-2},
\end{aligned} \tag{23}$$

where $p(z_{j-2} \mid \bar{s}_{n:j-2}, s_{0:n-1}, y_{1:n-1})$ is the same density at $j-2$. Thus, assuming that

$$p(z_{j-2} \mid \bar{s}_{n:j-2}, s_{0:n-1}, y_{1:n-1}) = \mathcal{N}(z_{j-2}; \bar{z}_{j-2}, \bar{P}^z_{j-2}), \tag{24}$$

it follows that $p(z_{j-1}, \bar{s}_{j-1} \mid \bar{s}_{n:j-2}, s_{0:n-1}, y_{1:n-1})$ is as in (16) but with $\bar{z}_{j-1}$ and $\bar{P}^z_{j-1}$. Conditioning on $\bar{s}_{j-1}$ then yields the same update as in (17)–(18). Finally, this yields

$$\begin{aligned}
&p(\bar{s}_j \mid \bar{s}_{n:j-1}, s_{0:n-1}, y_{1:n-1}) \\
&\quad = \mathcal{N}(\bar{s}_j; g_{j-1} + G\bar{z}_{j-1}, Q^s_j + G\bar{P}^z_{j-1}G^\mathsf{T}),
\end{aligned} \tag{25}$$

and the recursion is initialized with $\bar{z}_{n-1} = \hat{z}_{n-1}$, $\bar{P}^z_{n-1} = P^z_{n-1}$.

In practice, evaluating (21) for the whole future time horizon from n to N and for each time step and particle is impractical, due to the computational complexity. Hence, we propose to make use of the approximation suggested in [13] based on the fact that the state $\bar{s}_{n+\bar{N}}$ for $n + \bar{N} \gg n$ only weakly correlates with s_{n-1}. Hence, we can truncate (22) at some point $n + \bar{N} \leq N$ without introducing significant bias.

Finally, once a new trajectory $s^k_{0:N}$ has been sampled, smoothing the linear states is achieved by using a Rauch–Tung–Striebel smoothing pass [22]. This results in the method summarized in Algorithm 2, where $\pi(a_n \mid s_{0:n-1}, y_{1:n})$ and $\pi(s_n \mid s^{a^m_n}_{0:n-1}, y_{1:n})$ denote the proposal densities for the ancestor weights and nonlinear states, respectively.

Algorithm 2 Rao–Blackwellized PGAS

```
Input: Trajectory $\bar{s}_{0:N}$
Output: Trajectory $x^k_{0:N}$
 1: Sample $s^m_0 \sim \mathcal{N}(0, P^s_0)$ for $m = 1, \dots, M-1$ and set $s^M_0 = \bar{s}_0$
 2: Set $\hat{z}^m_0 = P^{zs}_0 (P^s_0)^{-1} s^m_0$, $P^z_0 = P^z_0 - P^{zs}_0 (P^s_0)^{-1} P^{sz}_0$, and $w^m_0 = 1/M$ for $m = 1, \dots, M$
 3: for $n = 1, \dots, N$ do
 4:     Sample $a^m_n \sim \pi(a_n \mid s_{0:n-1}, y_{1:n})$ for $m = 1, \dots, M-1$
 5:     Sample $s^m_n \sim \pi(s_n \mid s^{a^m_n}_{0:n-1}, y_{1:n})$ for $m = 1, \dots, M-1$
 6:     Set $s^M_n = \bar{s}^M_n$
 7:     Calculate the ancestor weights $\bar{w}^m_{n|N}$ using (22)–(25)
 8:     Sample $a^M_n \sim \mathcal{C}(\{\bar{w}^i\}^M_{i=1})$
 9:     Calculate $\hat{z}^m_n$ and $P^z_n$ according to (18)
10:     Set $s^m_{0:n} = \{s^{a^m_n}_{0:n-1}, s^m_n\}$
11:     Calculate the particle weights
            $w^m_n \propto w^{a^m_n}_{n-1} \frac{p(y_n \mid s^m_n) p(s^m_n \mid s^{a^m_n}_{0:n-1}, y_{1:n-1})}{\pi(a^m_n \mid s_{0:n-1}, y_{1:n}) \pi(s^m_n \mid s^{a^m_n}_{0:n-1}, y_{1:n})}$
12: end for
13: Sample $s^k_{0:N} \sim \mathcal{C}(\{w^m_N\}^M_{m=1})$
14: Calculate $p(z^k_{0:N} \mid s^k_{0:N}, y_{1:N})$ using a Rauch–Tung–Striebel smoother
```

3.2. GP Hyperparameter Sampling

The second step in Algorithm 1 is to draw new samples of the GP's hyperparameters from $p(\theta_f \mid y_{1:N}, x_{0:N}, \theta_y)$. First, note that

$$p(\theta_f \mid y_{1:N}, x_{0:N}, \theta_y) = p(\theta_f \mid x_{0:N}) \tag{26}$$

due to the fact that, conditionally on the complete state trajectory $x_{0:N}$, θ_f is independent of $y_{1:N}$ and θ_y. Rewriting the posterior $p(\theta_f \mid x_{0:N})$ then yields

$$\begin{aligned}
p(\theta_f \mid x_{0:N}) &\propto p(\theta_f) p(x_{0:N} \mid \theta_f) \\
&= p(\theta_f) p(x_0 \mid \theta_f) \prod_{n=1}^{N} p(x_n \mid x_{n-1}, \theta_f).
\end{aligned} \tag{27}$$

The densities $p(x_0 \mid \theta_f)$ and $p(x_n \mid x_{n-1}, \theta_f)$ are given by (5c) and

$$p(x_n \mid x_{n-1}, \theta_f) = \mathcal{N}(x_n; F_n x_{n-1}, Q_n), \tag{28}$$

respectively, where the latter follows from (5a).

Depending on how the hyperparameters θ_f enter (27), it may be possible to sample all or a subset of θ_f from $p(\theta_f \mid x_{0:N})$ directly (with an appropriate prior). If this is not possible, we note that (27) can readily be evaluated numerically (up to proportionality). Thus, a Metropolis-within-Gibbs approach can be used in this case to sample θ_f (or a subset thereof).

3.3. Likelihood Parameter Sampling

Finally, the third step is to sample the likelihood's parameters θ_y. Similar to the previous section, $p(\theta_y \mid y_{1:N}, x_{0:N}, \theta_f)$ can be written as

$$\begin{aligned}
p(\theta_y \mid y_{1:N}, x_{0:N}, \theta_f) &= p(\theta_y \mid y_{1:N}, x_{0:N}) \\
&\propto p(y_{1:N} \mid x_{0:N}, \theta_y) p(\theta_y) \\
&\propto p(\theta_y) \prod_{n=1}^{N} p(y_n \mid x_n, \theta_y)
\end{aligned} \tag{29}$$

where the first equality is due to the independence of θ_y on θ_f given $x_{0:N}$.

Depending on the likelihood $p(y_n \mid x_n, \theta_y)$, it is often possible to sample from (29) directly. For example, when the likelihood consists of additive Gaussian noise and the noise variance is unknown, the conjugate prior can be used for θ_y to obtain a closed form solution for the posterior, from which we can draw samples. If this is not possible, we can still resort to a Metropolis-within-Gibbs approach as discussed in the previous section.

4. NUMERICAL ILLUSTRATIONS

To illustrate the proposed method, we consider a spatio-temporal population regression problem.

4.1. Setup

We model the logarithm of a time-varying population r_n in an area as a Gaussian process, that is, $f_n \sim \mathcal{GP}(0, k(\Delta\mathsf{x}, \tau))$ where $f_n = \log(r_n)$. As for the covariance function, we use the covariance function $k(\Delta\mathsf{x}, \tau) = k_M(\Delta\mathsf{x})k_M(\tau)$ discussed in Section 2 and assume known orders $\nu_\mathsf{x} = 3.5$ and $\nu_t = 4.5$. The unknown GP parameters are thus the length scales l_x and l_t, and the variance σ^2.

The population size measurements for the jth location x_j ($j = 1, \dots, J$) at time t_n are assumed to be Poisson distributed according to the likelihood

$$y_{j,n} \sim \mathcal{P}(\exp(f_{j,n})),$$

where $\mathcal{P}(\cdot)$ denotes the Poisson distribution and $f_{j,n} \triangleq f_n(\mathsf{x}_j)$ is the population's logarithm at the jth location x_j.

The simulation data at each location is generated from a Ricker model [23] given by

$$r_{j,n} = r_{j,n-1} \exp\left(\rho\left(1 - \frac{r_{j,n-1}}{\kappa} + e_{j,n}\right)\right),$$

where $e_{j,n}$ is the process noise with $\mathrm{Cov}\{e_{i,n}e_{j,n}\} = \sigma_{ij}^2$, $\rho = 0.1$ is the intrinsic growth rate, and $\kappa = 100$ the environmental carrying capacity.

In total, $N = 1000$ time samples in an area of 40 km by 20 km at 8 locations (separated by 10 km) are simulated. The performance of the proposed algorithm is evaluated by leave-one-out cross-validation and calculating the time-averaged root mean squared error for the prediction at the test location. This yields a total of 7000 training points and 1000 test points.

In total, 200 MCMC samples are drawn, of which 100 are discarded as burn-in. For the particle filter, $M = 2000$ particles are used. The ancestor indices are sampled according to the particle weight at time $n-1$ (i.e. $a_n \sim \mathcal{C}\{w_{n-1}^m\}_{m=1}^M$) while the bootstrap proposal $p(s_n \mid s_{0:n-1}, y_{1:n-1})$ (20) is used for s_n. Finally, $\bar{N} = 1$ was chosen (without noticeable performance degradation).

4.2. Parameter Sampling

In this scenario, the unknown parameters are $\theta_f = \begin{bmatrix} l_\mathsf{x} & l_t & \sigma^2 \end{bmatrix}^\mathsf{T}$ with no unknowns in the likelihood. First, note that the Lyapunov equation (4) can be solved analytically in this case and P_0 can be written as

$$P_0 = \sigma^2 \tilde{P}_0$$

where $\tilde{P}_0$ is independent of σ^2. Similarly, from (6) it can be seen that Q_n can be written as

$$Q_n = \sigma^2 \tilde{Q}_n.$$

Thus, it follows from (27)–(28) that

$$\begin{aligned} &p(\sigma^2 \mid l_t, l_\mathsf{x}, x_{0:N}) \\ &\quad \propto p(\sigma^2)\mathcal{N}(x_0; 0, \sigma^2\tilde{P}_0) \prod_{n=1}^N \mathcal{N}(x_n; F_n x_{n-1}, \sigma^2 \tilde{Q}_n). \end{aligned}$$

Using the inverse Gamma prior $p(\sigma^2) = \mathcal{IG}(\sigma^2; \alpha, \beta)$ with prior parameters α and β yields the posterior

$$p(\sigma^2 \mid l_t, l_\mathsf{x}, x_{0:N}) = \mathcal{IG}(\sigma^2; \tilde{\alpha}, \tilde{\beta})$$

with posterior parameters

$$\tilde{\alpha} = \alpha + \frac{NN_x}{2},$$

$$\tilde{\beta} = \beta + \frac{1}{2}\|x_0\|_{\tilde{P}_0^{-1}}^2 + \frac{1}{2}\sum_{n=1}^N \|x_n - F_n x_{n-1}\|_{\tilde{Q}^{-1}}^2,$$

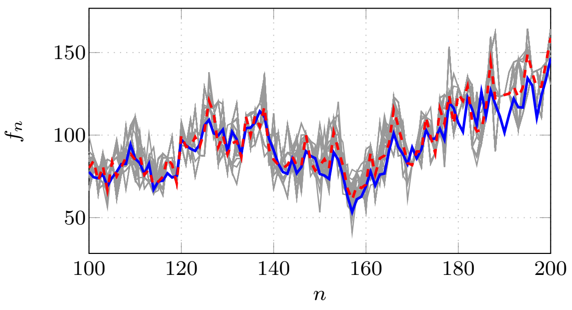

Fig. 1. Prediction example with true trajectory (──) simulated from a Ricker model, together with the MMSE estimate (- - -), and the sampled trajectories (──) for the time period $n = 100, \dots, 200$.

where $\|x\|_C^2 = x^\mathsf{T} C x$ and N_x is the state dimension. Thus, sampling σ^2 in this example can be done efficiently by sampling from the posterior.

Sampling the length scales l_x and l_t can not be done directly as they enter the model in a more complicated way (see Section 2). Hence, we use the Metropolis-within-Gibbs approach as discussed in Section 3.

4.3. Results and Discussion

The mean of the time-averaged RMSEs of the leave-one-out cross-validation for the presented scenario is 8.86 ($\pm$2.47). An example for the predicted population for the time between $n = 100$ and $n = 200$ is shown in Fig. 1. The illustration depicts the true population over time (blue, solid), the sampled trajectories (grey, solid), as well as the minimum mean squared error (MMSE) estimate (red, dashed). As it can be seen, the sampled trajectories are distributed around the true state and the MMSE estimate matches the true trajectory well. The RMSE is 6.21 in this case, which is somewhat lower than the mean RMSE. Furthermore, Fig. 2 shows the histograms of the corresponding parameter samples.

The example shows that the learned model exhibits good predictive performance based on the estimated parameters and trajectories. A major challenge that remains is the scalability of the proposed method with respect to the number of spatial training points: The dimension of the nonlinear state s_n grows linearly with the number of spatial training points, which in turn increases the number of particles required for sampling the state trajectories $s_{0:N}^k$. Experiments showed that for a low number of points (e.g. 2–3), as few as 100 particles can be enough to accurately learn the system and generate predictions, while, at around 10 to 15 spatial training points, several thousand particles are required, slowing down the learning. Note, however, that increasing the number of temporal training points does not affect the state dimension, and only increases the number of time samples to be processed.

5. CONCLUSIONS

In this paper, we proposed a particle MCMC method for estimating the posterior distribution of the state and parameters in latent, spatio-temporal Gaussian process regression problems. As shown in the illustrations, the proposed method can handle nonlinear non-Gaussian likelihoods, which can be a limiting factor for other methods. For problems with a moderate to large number of spatial training points,

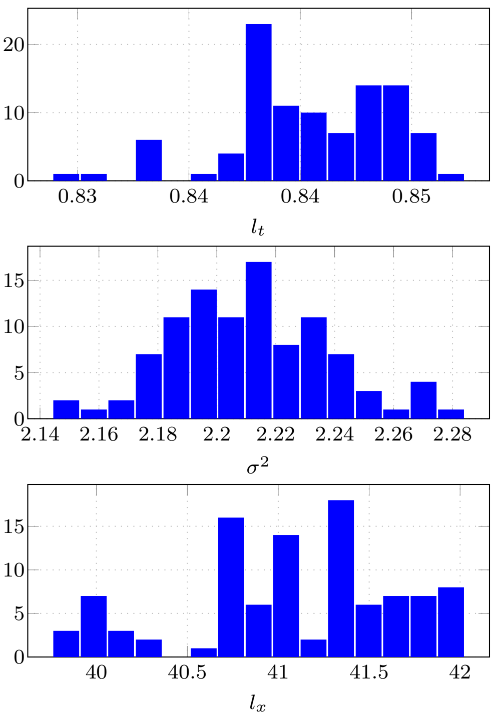

Fig. 2. Histograms of the hyperparameter samples: The temporal length scale l_t (top), variance σ^2 (middle), and spatial length scale l_x (bottom).

scaling can become an issue as the number of required particles in such scenarios increases quickly.

6. REFERENCES

[1] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[2] J. Han, X.-P. Zhang, and F. Wang, "Gaussian process regression stochastic volatility model for financial time series," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 6, pp. 1015–1028, September 2016.

[3] N. Chen, Z. Qian, I. T. Nabney, and X. Meng, "Wind power forecasts using Gaussian processes and numerical weather prediction," *IEEE Transactions on Power Systems*, vol. 29, no. 2, pp. 656–665, March 2014.

[4] J. Vanhatalo, V. Pietiläinen, and A. Vehtari, "Approximate inference for disease mapping with sparse Gaussian processes," *Statistics in Medicine*, vol. 29, no. 15, pp. 1580–1607, 2010.

[5] N. Wahlström and E. Özkan, "Extended target tracking using Gaussian processes," *IEEE Transactions on Signal Processing*, vol. 63, no. 16, pp. 4165–4178, August 2015.

[6] J. Hartikainen and S. Särkkä, "Kalman filtering and smoothing solutions to temporal Gaussian process regression models," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, August 2010, pp. 379–384.

[7] S. Särkkä, A. Solin, and J. Hartikainen, "Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing: A look at Gaussian process regression through Kalman filtering," *IEEE Signal Processing Magazine*, vol. 30, no. 4, pp. 51–61, July 2013.

[8] A. Carron, M. Todescato, R. Carli, L. Schenato, and G. Pillonetto, "Machine learning meets Kalman filtering," in *55th IEEE Conference on Decision and Control (CDC)*, December 2016, pp. 4594–4599.

[9] J. Hartikainen, J. Riihimäki, and S. Särkkä, "Sparse spatio-temporal Gaussian processes with general likelihoods," in *21st International Conference on Artificial Neural Networks (ICANN)*, 2011, pp. 193–200.

[10] T. Heskes and O. Zoeter, "Expectation propagation for approximate inference in dynamic Bayesian networks," in *18th Conference on Uncertainty in Artificial Intelligence*, Alberta, Canada, 2002, pp. 216–223.

[11] H. Rue, S. Martino, and N. Chopin, "Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 71, no. 2, pp. 319–392, 2009.

[12] J. Quiñonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *Journal of Machine Learning Research*, vol. 6, pp. 1939–1959, 2005.

[13] F. Lindsten, M. I. Jordan, and T. B. Schön, "Particle Gibbs with ancestor sampling," *Journal of Machine Learning Research*, vol. 15, pp. 2145–2184, 2014.

[14] N. Kantas, A. Doucet, S. S. Singh, J. Maciejowski, and N. Chopin, "On particle methods for parameter estimation in state-space models," *Statistical Science*, vol. 30, no. 3, pp. 328–351, August 2015.

[15] R. Frigola, F. Lindsten, T. B. Schön, and C. E. Rasmussen, "Bayesian inference and learning in Gaussian process state-space models with particle MCMC," in *Advances in Neural Information Processing Systems 26*, 2013, pp. 3156–3164.

[16] A. Svensson, A. Solin, S. Särkkä, and T. Schön, "Computationally efficient Bayesian learning of Gaussian process state space models," in *19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 51, Cadiz, Spain, May 2016, pp. 213–221.

[17] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1984.

[18] T. Karvonen and S. Särkkä, "Approximate state-space Gaussian processes via spectral transformation," in *26th IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, September 2016.

[19] C. Andrieu, A. Doucet, and R. Holenstein, "Particle Markov chain Monte Carlo methods," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 72, no. 3, pp. 269–342, 2010.

[20] A. Doucet, N. De Freitas, K. Murphy, and S. Russell, "Rao–Blackwellised particle filtering for dynamic Bayesian networks," in *16th Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA, USA, 2000, pp. 176–183.

[21] T. Schön, F. Gustafsson, and P.-J. Nordlund, "Marginalized particle filters for mixed linear/nonlinear state-space models," *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2279–2289, July 2005.

[22] H. E. Rauch, C. T. Striebel, and F. Tung, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, vol. 3, no. 8, pp. 1445–1450, August 1965.

[23] W. E. Ricker, "Stock and recruitment," *Journal of the Fisheries Research Board of Canada*, vol. 11, no. 5, pp. 559–623, 1954.