Hostettler, Roland; Särkkä, Simo; Godsill, Simon J.

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

Published in:

DOI:
10.1109/MLSP.2017.8168171

Published: 01/01/2017

Document Version
Peer reviewed version

Please cite the original version:
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

\[ f(x, t) \sim GPM(x(t); \theta_f), k(x(t), x'(t'); \theta_k), \]

\[ y(t) \sim p(y(t) | f(x(t), \theta_y)), \]

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

The objective is then to infer the function values \( f_n \equiv f(x, t_n) \) at times \( t_n \) (for \( n = 1, \ldots, N \)) as well as the parameters \( \theta_f \) and \( \theta_y \) based on a set of measurements \( y_{1:N} = \{y_1, \ldots, y_N\} \). The most common approach to do this in batch settings is by maximizing the marginal likelihood of the data \( p(y_{1:N} | \theta_f, \theta_y) \) 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 \( p(y_{n} | f_n, \theta_y) \), 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 \( p(y_n | f_n, \theta_y) \), 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 \( \theta_y \)).

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
\[ k(\Delta x, \tau) = S_f(\omega_x, \omega_t) \]
into a white noise process with spectral density \( S_w \) and a linear system with frequency response \( H(\omega_x, \omega_t) \), such that [17]
\[ S_f(\omega_x, \omega_t) = S_w H(\omega_x, \omega_t) H(\omega_x, \omega_t)^* \]  \( , \) (2)
where the superscript * denotes the complex conjugate. From the transfer function \( H(\omega_x, \omega_t) \) of the linear system, the corresponding infinite dimensional state-space representation can then readily be found using well known conversion techniques to obtain [7]
\[ \dot{x}(x, t) = Ax(x, t) + Bu(x, t), \]  \( \) (3a)
\[ f(x, t) = Cx(x, t), \]  \( \) (3b)
\[ x(x, 0) \sim \mathcal{N}(0, P_0), \]  \( \) (3c)
where \( w(x, t) \) 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 \( H(\omega_x, \omega_t) \). The covariance \( P_0 \) of the initial state \( x(x, 0) \) is given by the solution of the following continuous time Lyapunov equation
\[ A P_0 + P_0 A^T + B S_w B^T = 0. \]  \( \) (4)
Finally, discretizing (3) with respect to time yields
\[ x_n = F_n x_{n-1} + q_n, \]  \( \) (5a)
\[ f_n = C x_n, \]  \( \) (5b)
\[ x_0 \sim \mathcal{N}(0, P_0), \]  \( \) (5c)
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
\[ Q_n = \int_0^{\Delta t} \exp(A(\Delta t - \tau)) B S_w B^T \exp(A(\Delta t - \tau))^T d\tau. \]  \( \) (6)
As an example, consider a covariance function composed of the product of two Matérn kernels \( k_M(\cdot) \)
\[ k(\Delta x, \tau) = k_M(\Delta x) k_M(\tau), \]  \( \) (7)
with
\[ k_M(r) = \sigma^2 2^{1-v} \frac{\Gamma(v)}{\Gamma(v)} \left( \frac{\sqrt{2\nu|\tau|}}{l} \right)^v K_v \left( \frac{\sqrt{2\nu|\tau|}}{l} \right), \]  \( \) (8)
where \( \sigma^2, l, \) and \( v \) are the kernel’s hyperparameters, and \( K_v(\cdot) \) is the modified Bessel function of the second kind of order \( v \).
Using the Fourier transform with respect to \( \tau \) on (7)-(8) yields
\[ S(\Delta x, \omega_t) = k_M(\Delta x) \frac{2\sigma^2 \sqrt{2\nu l^2}}{\Gamma(v) \Gamma(v)} \left[ \lambda_t^2 + \omega^2 \right]^{-v+1/2} \]
\[ \times \left( \lambda_t^2 + \omega^2 \right)^{-v+1/2} \]  \( \) (9)
where we have used \( \lambda_t = \sqrt{2\nu}/l_t \). Since (7) is a product, (9) can be converted without involving the spectral density with respect to \( \Delta x \) [6]. Furthermore, from (9) it is clear that the order of the linear system directly depends on the hyperparameter \( \nu_t \). Whenever \( \nu_t = 1/2 \) is a positive integer, the following companion form state-space representation can be obtained
\[ A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\lambda_t^{v+1} & -a_2 \lambda_t^v & \cdots & -a_2 \lambda_t^v & -a_1 \lambda_t \end{bmatrix}, \]  \( \) (10a)
\[ B = [0 \; \cdots \; 0 \; 1]^T, \]  \( \) (10b)
\[ C = [1 \; 0 \; \cdots \; 0]^T, \]  \( \) (10c)
where \( a_i \) is the \( i \)th binomial coefficient of \( \lambda_t + 1 \) \( \nu+1 \), and
\[ S_w = k_M(\Delta x) \frac{2\sigma^2 \lambda_t^{2v} \Gamma(\nu_t + 1/2)}{\Gamma(\nu_t)}, \]  \( \) (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 \( H(\omega_x, \omega_t) \). 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].

3. STATE AND PARAMETER ESTIMATION

In this section, we develop the proposed approach for estimating the posterior distribution of the parameters and the state, \( p(\theta, \phi, x_0:N \mid y_{1:N}) \). The approach is based on Gibbs sampling and particle MCMC methods (see [13, 19]) where the state trajectory \( x_{0:N} \), GP hyperparameters \( \theta \), and likelihood parameters \( \phi \) (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 \( p(y_{1:N} \mid \theta, \phi) \), 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 \( \theta_f \) and \( \theta_y \) 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 \( \tilde{x}_{0:N} \), 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 structure, which can be exploited. In particular, the states can be written as points as well as the prediction points). Thus, the dynamic model (12) allows \([20, 21]\). Starting from the joint filtering density, we obtain 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 can factorize as follows

\[
\mathbb{P}(s_{n}, s_{0:n} \mid y_{1:n}) = \mathbb{P}(z_{n} \mid s_{0:n}, y_{1:n})\mathbb{P}(s_{0:n} \mid y_{1:n}).
\]  

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

\[
\mathbb{P}(z_{n}, s_{0:n} \mid y_{1:n}) = \mathbb{P}(z_{n} \mid s_{0:n}, y_{1:n})\mathbb{P}(s_{0:n} \mid y_{1:n}),
\]

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

\[
\mathbb{P}(z_{n}, s_{0:n} \mid y_{1:n}) \approx \sum_{m=1}^{M} \mathbb{P}(z_{n} \mid s_{0:n}^{m}, P_{z_{n}}^{m})\mathbb{P}(s_{0:n} \mid s_{0:n}^{m}, P_{s_{0:n}}^{m})
\]

(13) where \( \mathbb{P}(z_{n} \mid s_{0:n}^{m}, P_{z_{n}}^{m})\mathbb{P}(s_{0:n} \mid s_{0:n}^{m}, P_{s_{0:n}}^{m}) \) denotes the conditional mean of the linear states, and \( P_{s_{0:n}}^{m} \) their covariance. Note that since neither \( Q_{n} \) nor \( H \) depend on \( s_{n-1} \), the covariance \( P_{s_{0:n}}^{m} \) does not depend on the state trajectory \( s_{0:n} \).

### 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

\[
p(z_{n-1} \mid s_{0:n-1}, y_{1:n-1}) = \mathcal{N}(z_{n-1} \mid \hat{z}_{n-1}, P_{z_{n-1}}^{m})
\]

(14) and that \( s_{n} \) has been sampled. Then, the update for \( z_{n} \) reduces to a Kalman filter prediction

\[
p(z_{n} \mid s_{0:n}, y_{1:n}) \propto p(y_{n} \mid s_{0:n}, z_{n})\mathbb{P}(z_{n} \mid s_{0:n}, y_{1:n-1})
\]

(15) since the likelihood is independent of \( z_{n} \). Considering the joint density \( p(z_{0:n}, s_{n}) \) yields

\[
p(z_{0:n}, s_{n} \mid s_{0:n-1}, y_{1:n-1}) = \int p(z_{n}, s_{n} \mid s_{0:n-1}, y_{1:n-1})\mathbb{P}(z_{n} \mid s_{0:n-1}, y_{1:n-1})\mathbb{P}(s_{n} \mid s_{0:n-1}, y_{1:n-1})
\]

(16) with

\[
\hat{z}_{n} = h_{n-1} + H\hat{z}_{n-1} + K(s_{n} - g_{n-1} - G\hat{z}_{n-1}),
\]

(18a)

\[
P_{n}^{x} = Q_{n}^{x} + H P_{n-1}^{z} H^{T} - KSK^{T},
\]

(18b)

\[
S = Q_{n}^{z} + GP_{n-1}^{z} G^{T},
\]

(18c)

\[
K = (Q_{n}^{x} + H P_{n-1}^{z} G)^{-1}. 
\]

### 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

\[
p(s_{0:n} \mid y_{1:n}) \propto p(y_{n} \mid s_{0:n}, y_{1:n-1})\mathbb{P}(s_{0:n} \mid y_{1:n-1})
\]

(19) The middle term is the marginalized dynamics for \( s_{n} \) and can be found from

\[
p(s_{n} \mid s_{0:n-1}, y_{1:n-1}) = \int p(s_{n} \mid z_{n-1}, s_{n-1}) p(z_{n-1} \mid s_{0:n-1}, y_{1:n-1})\mathbb{P}(z_{n-1} \mid s_{0:n-1}, y_{1:n-1})
\]

(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

\[
\tilde{w}_{n|N} = \prod_{j=1}^{N} p(\delta_{j} \mid \tilde{s}_{n,j-1}, s_{0:n-1}, y_{1:n-1}),
\]

(21) where \( \tilde{s}_{n-1} = \{} \) is the empty set (for \( j = n \)). The marginal future predictions \( p(\delta_{j} \mid \tilde{s}_{n,j-1}, s_{0:n-1}, y_{1:n-1}) \) 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

\[ p(s_j | s_{n+1-j}, y_{n+1-j}) = \int p(s_j | z_{j-1}, \hat{s}_{j-1}) \times p(z_{j-1} | s_{n+1-j}, y_{n+1-j}) dz_{j-1}. \]  

(22)

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

\[ p(z_{j-1}, \hat{s}_{j-1} | s_{n+1-j}, y_{n+1-j}) = \int p(z_{j-1}, \hat{s}_{j-1} | z_{j-2}, \hat{s}_{j-2}) \times p(z_{j-2} | s_{n+1-j}, y_{n+1-j}) dz_{j-2}, \]

(23)

where \( p(z_{j-2} | s_{n+1-j}, y_{n+1-j}) \) is the same density at \( j-2 \). Thus, assuming that

\[ p(z_{j-2} | s_{n+1-j}, y_{n+1-j}) = \mathcal{N}(z_{j-2}; \hat{z}_{j-2}, P_{j-2}^n), \]

(24)

it follows that \( p(z_{j-1}, \hat{s}_{j-1} | s_{n+1-j}, y_{n+1-j}) \) is as in (16) with \( \hat{z}_{j-1} \) and \( P_{j-1}^n \). Conditioning on \( s_{j-1} \) then yields the same update as in (17)-(18). Finally, this yields

\[ p(s_j | s_{n+1-j}, y_{n+1-j}) = \mathcal{N}(s_j; \hat{s}_{j-1} + G\hat{z}_{j-1}, Q_j + GP_{j-1}nG^T), \]

(25)

and the recursion is initialized with \( \hat{s}_{n+1} = \hat{z}_{n+1}, P_{n+1}^n = P_n^m \).

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 \( s_{n+1-N} \) for \( n+1-N \gg n \) only weakly correlates with \( s_{n-1-N} \). Hence, we can truncate (22) at some point \( n + N \leq N \) without introducing significant bias.

Finally, once a new trajectory \( s_{0:N} \) 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 \( \pi(a_n | s_{n-1}, y_{n+1}) \) and \( \pi(s_m | s_{m-1}, y_{m}, y_{m+1}) \) denote the proposal densities for the ancestor weights and nonlinear states, respectively.

3.2. GP Hyperparameter Sampling

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

\[ p(\theta_f | y_{1:N}, x_{0:N}, \theta_y) = p(\theta_f | x_{0:N}) \]

(26)

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

\[ p(\theta_f | x_{0:N}) \propto p(\theta_f) p(x_{0:N} | \theta_f) \]

\[ = p(\theta_f) p(x_0 | \theta_f) \prod_{n=1}^{N} p(x_n | x_{n-1}, \theta_f). \]

(27)

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

\[ p(x_n | x_{n-1}, \theta_f) = \mathcal{N}(x_n; F_n x_{n-1}, Q_n), \]

(28)

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

Algorithm 2 Rao–Blackwellized PGAS

**Input:** Trajectory \( s_{0:N} \)

**Output:** Trajectory \( s_{0:N}^m \)

1. Sample \( s_{m}^{n} \sim \mathcal{N}(0, P_m^n) \) for \( m = 1, \ldots, M-1 \) and set \( s_{M}^m = s_0^m \)
2. Set \( \hat{s}_{0:n}^m = P_m^n (P_0^0)^{-1}s_{0:n}^m, P_0^m = P_0^0 - P_0^0 (P_0^0)^{-1}P_m^n, \) and \( w_0^m = 1/M \) for \( m = 1, \ldots, M \)
3. for \( n = 1, \ldots, N \) do
4. Sample \( a_n^m \sim \pi(a_n | s_{0:n-1}, y_{n+1}) \) for \( m = 1, \ldots, M-1 \)
5. Sample \( s_{n}^m \sim \pi(s_n | s_{0:n-1}, y_{n}) \) for \( m = 1, \ldots, M-1 \)
6. Set \( s_{n}^m = \hat{s}_{n}^m \)
7. Calculate the ancestor weights \( w_n^m \) using (22)–(25)
8. Sample \( a_n^m \sim \mathcal{C}(\{w_n^m\}_{m=1}^{M}) \)
9. Calculate \( s_{n}^m = \hat{s}_{n}^m \) and \( P_{n}^m \) according to (18)
10. Set \( s_{0:n}^m = \{s_{0:n}^m, s_{n}^m\} \)
11. Calculate the particle weights
\[ w_n^m \propto w_{n-1}^m \times p(y_n | s_n^m) p(s_n^m | s_{0:n-1}^m, y_{n+1}) \]
\[ \times \pi(s_n^m | s_{0:n-1}^m, y_{n+1}) \]

12. end for
13. Sample \( s_{0:N}^m \sim \mathcal{C}(\{w_n^m\}_{n=1}^{N}) \)
14. Calculate \( p(s_{0:N}^m | s_{0:N}^m, y_{N+1}) \) using a Rauch–Tung–Striebel smoother

Depending on how the hyperparameters \( \theta_f \) enter (27), it may be possible to sample all or a subset of the \( \theta_f \) from \( p(\theta_f | x_{0:N}) \) 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 \( \theta_f \) (or a subset thereof).

3.3. Likelihood Parameter Sampling

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

\[ p(\theta_y | y_{1:N}, x_{0:N}, \theta_f) = p(\theta_y | y_{1:N}, x_{0:N}) \]
\[ \times \frac{p(y_{1:N} | x_{0:N}, \theta_y) p(\theta_y)}{p(y_{1:N} | x_{0:N}, \theta_f) p(\theta_f)} \]

(29)

where the first equality is due to the independence of \( \theta_y \) on \( \theta_f \) given \( x_{0:N} \).

Depending on the likelihood \( p(y_n | x_n, \theta_y) \), 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 \( \theta_y \) 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, $f_n = \log(r_n)$. As for the covariance function, we use the covariance function $k(\Delta x, \tau) = k_M(\Delta x)k_M(\tau)$ discussed in Section 2 and assume known orders $\nu_x = 3.5$ and $\nu_t = 4.5$. The unknown GP parameters are thus the length scales $l_x$ and $l_t$, and the variance $\sigma^2$.

The population size measurements for the $j$th location $x_j$ ($j = 1, \ldots, J$) 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 $f_{j,n} \triangleq f_n(x_j)$ is the population’s logarithm at the $j$th 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 $\text{Cov}(e_{i,n}, e_{j,n}) = \sigma^2 j$, $\rho = 0.1$ is the intrinsic growth rate, and $\kappa = 100$ is 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. $a_n \sim \mathcal{C}(w_{n-1})_{M=1}^N$) while the bootstrap proposal $p(s_n | s_{n-1}, y_{1:n-1})$ (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 $\theta_f = [l_x \quad l_t \quad \sigma^2]^{T}$ with no unknowns in the likelihood. First, note that the Lyapunov equation (4) can be solved analytically and $P_0$ can be written as

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

where $\tilde{P}_0$ is independent of $\sigma^2$. 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

$$p(\sigma^2 | l_t, l_x, x_{0:N}) \propto p(\sigma^2)N(x_0; 0, \sigma^2 \tilde{P}_0) \prod_{n=1}^{N} N(x_n; F_n x_{n-1}, \sigma^2 \tilde{Q}_n).$$

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

$$p(\sigma^2 | l_t, l_x, x_{0:N}) = IG(\sigma^2; \tilde{\alpha}, \tilde{\beta})$$

with posterior parameters

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

$$\tilde{\beta} = \beta + \frac{1}{2} ||x_0||_F^2 + \frac{1}{2} \sum_{n=1}^{N} ||x_n - F_n x_{n-1}||_F^2.$$
scaling can become an issue as the number of required particles in such scenarios increases quickly.

6. REFERENCES


