Sequential sampling of Gaussian process latent variable models

Martin Tegnér [1] **Benjamin Bloem-Reddy** [2] **Stephen Roberts** [1]

Abstract

We consider the problem of inferring a latent function in a probabilistic model of data. When dependencies of the latent function are specified by a Gaussian process and the data likelihood is complex, efficient computation often involve Markov chain Monte Carlo sampling with limited applicability to large data sets. We extend some of these techniques to scale efficiently when the problem exhibits a sequential structure. We propose an approximation that enables sequential sampling of both latent variables and associated parameters. We demonstrate strong performance in growing-data settings that would otherwise be unfeasible with naive, non-sequential sampling.

1. Introduction

Gaussian processes (GP) are extensively used by the machine learning community as a flexible framework for non-parametric modelling (Rasmussen & Williams, 2006). They offer a probabilistic approach to infer and predict dependencies in data. Here we concentrate on their use in latent variable models, where an unobservable function is generative of data through a possibly complex and non-linear likelihood.

These models give rise to inversion problems. The reader can think of the latent quantity as a non-observable input-function to a physical system, from which one have noisy measurements of the output. The measurements relate to each other in a way prescribed by the latent function and the system, and the problem is to "inversely" find a function that best explains observed data.

In a probabilistic framework, the aim is to infer a distribution over the latent function, and not just a single "best" point-estimate. This acknowledges a notion of uncertainty attached to the estimate, which stems from availability and noisiness of data, and the interplay of the latent function with the system itself. Here the GP acts as a nonparametric model which encodes prior beliefs and domain knowledge about the latent function. The system's input-to-output mapping is formalised in a likelihood model, which also specifies a noise distribution. Together with a prior over the model's hyperparameters, the GP and likelihood of observed data then induces a posterior distribution over the latent function. These type of models are considered within system biology (Barenco et al., 2006), geostatistics (Rue et al., 2009), and robot kinematics (Williams et al., 2009), to mention just but a few examples.

Inference on the posterior of latent variable models is, however, generally intractable. The case of Gaussian likelihoods under known hyperparameters is a rare exception. For this purpose, we build on Markov chain Monte Carlo (MCMC), which approach inference with samples from the posterior. Elliptical slice sampling (ESS) (Murray et al., 2010) provides a versatile method proved to be fast and efficient for Gaussian latent variable models. While ESS updates latent variables for a known (or fixed) covariance, the related and equally applicable surrogate data slice sampling (SDSS) (Murray & Adams, 2010) updates hyperparameters. In cooperation they provide a MCMC strategy for full Bayesian sampling inference.

These methods are limited by their scaling with data size N. Exploitable structure and approximations aside (e.g. (Saatchi, 2012) and (Quiñonero-Candela & Rasmussen, 2005)), the computational cost of decompositions for covariance proposals is $\mathcal{O}(N^3)$, which becomes problematic already when N approaches a few thousand.

To the best of our knowledge, there is not much previous work that addresses the issue of computational complexity within MCMC samplers for inference in latent GP models. In this paper, we provide a strategy that aims to solve this issue when the data generating process has a sequential structure. We look at problems of data size TN, where N is the observation size at each of T steps. We propose a simple algorithm that makes sequential approximations of the posterior with expectation propagation (Minka, 2001) and use this as target distribution in a MCMC sampler. This cuts the cost from $\mathcal{O}(T^3N^3)$ for updating the full model to $\mathcal{O}(T\tau^3N^3)$, for a tunable or user-specified constant τ.

[1]Department of Engineering Science, University of Oxford, Oxford, United Kingdom [2]Department of Statistics, University of Oxford, Oxford, United Kingdom. Correspondence to: Martin Tegnér <martin.tegner@eng.ox.ac.uk>.

Working Notes of a ICML/FAIM 2018 Workshop, Stockholm, Sweden, 2018.

We then exploit the strengths of ESS and SDSS to obtain a practical and fast MCMC sampler. We demonstrate the benefits with empirical experiments based on two cases: one standard benchmark and one complex likelihood model.

2. Sequential latent Gaussian process models

We consider Bayesian inference for probabilistic models of observable data $\mathbf{y} \in \mathbb{R}^n$. We assume a given conditional likelihood which represents the data-generating process, $\mathbf{y} \sim p(\mathbf{y}|\mathbf{f}, \alpha)$, where a latent variable $\mathbf{f} \in \mathbb{R}^N$ is the main object of interest for inference. As an example, one can have in mind a system where $\mathbf{f}$ is an unobservable input and $\mathbf{y}$ noisy measurements of the output $\mathbf{y} = \Phi(\mathbf{f}) + \boldsymbol{\epsilon}$. Parameters associated with the likelihood—with the system Φ and noise distribution $p(\boldsymbol{\epsilon})$—are collected in α.

In this work, the latent variable is assumed to have a functional prior $f(x) \sim \mathcal{GP}(0, k_\kappa(x, x'))$ with input space $x \in \mathbb{R}^D$. Let $\mathcal{N}(\mathbf{z}|\mathbf{m}, \mathbf{K})$ denote a Gaussian density with mean $\mathbf{m}$ and covariance matrix $\mathbf{K}$. By definition, the GP induces a prior $\mathbf{f} \sim p(\mathbf{f}|\kappa) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}_\kappa)$. $\mathbf{K}_\kappa$ is the Gram matrix constructed by the covariance kernel with parameters κ: $[\mathbf{K}_\kappa]_{i,j} = k_\kappa(x_i, x_j)$ for all pairs x_i, x_j of the input set $\mathbf{x} \in \mathbb{R}^{N \times D}$ corresponding to $\mathbf{f}$. A mean vector $\mathbf{m}_m$ can be absorbed into the likelihood function $p(\mathbf{y}|\mathbf{f}, \alpha) \equiv p(\mathbf{y}|\mathbf{f} + \mathbf{m}_m, \alpha)$, where parameters m are included in α for convenience.

Sequential data model Consider now the data to be a sequence of observations $\mathbf{y} = \{\mathbf{y}_t\}_{t=1}^T$ with corresponding $\mathbf{f} = \{\mathbf{f}_t\}_{t=1}^T$, where the $\mathbf{y}_t \in \mathbb{R}^n$ are assumed conditionally independent given $\mathbf{f}_t \in \mathbb{R}^N$. The likelihood then factorises over t (but not necessarily over the elements of $\mathbf{y}_t$): $p(\mathbf{y}_{1:T}|\mathbf{f}_{1:T}, \alpha) = \prod_{t=1}^T p(\mathbf{y}_t|\mathbf{f}_t, \alpha)$ where we use a shorthand notation $\mathbf{y}_{1:T} \equiv \{\mathbf{y}_t\}_{t=1}^T$. Similarly for the system example, the noisy output $\mathbf{y}_t$ observed at t would depend on $\mathbf{f}_t$ only, with the noise being independent across t as well.

In this setting, the temporal dependence in data is induced by the dependency structure of the latent GP. This is modelled by augmenting the input space with $t \in \mathbb{N}$ and we assume a covariance kernel that is separable and isotropic in t

$$k_\kappa(t, x; t', x') = k_\kappa^{(t)}(|t - t'|)\, k_\kappa^{(x)}(x, x'). \tag{1}$$

Thus, the covariance matrix $\mathbf{K}_\kappa$ of $\mathbf{f}_{1:T}$ is the Gram matrix over all (t, x)-inputs in $\mathbf{x}_{1:T} \equiv \{\mathbf{x}_t\}_{t=1}^T$, where $\mathbf{x}_t$ denotes the set of spatial inputs belonging to $\mathbf{f}_t$.

Remark The sampling algorithms we propose in this work are not restricted to Gaussian process models. The necessary assumption is $\mathbf{f} \sim \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}_\kappa)$ with a parametrised covariance. Similarly, the isotropic form (1) is not strictly necessary, although we consider it a natural assumption for sequential/temporal data. Alternatively, one may consider non-stationary covariances, such as periodic kernels. The implication for our work would be that one has to make more informed choices when dropping data in (5). Without loss of generality, we therefore take as assumed a GP with kernel (1) when developing our sampling strategy and, accordingly, when designing numerical experiments.

2.1. Naive sequential sampling

Posterior inference on $\mathbf{f}$ and hyperparameters (κ, α) is generally difficult. The likelihood of $\mathbf{y}$ may depend on $\mathbf{f}$ in a nonlinear way. In such cases, marginalisation of $\mathbf{f}$ is intractable, and inference relies on approximating the posterior distribution $p(\mathbf{f}, \kappa, \alpha|\mathbf{y}) = \frac{1}{Z} p(\mathbf{y}|\mathbf{f}, \alpha) p(\mathbf{f}|\kappa) p_h(\kappa, \alpha)$ where $p_h(\kappa, \alpha)$ is a hyperprior and Z a normalising constant.

Efficient MCMC methods for sampling from the posterior for known or fixed (κ, α) are proposed in (Murray et al., 2010). The hyperparameters may be inferred using techniques proposed in (Murray & Adams, 2010). We will leverage these methods—ESS and SDSS—but consider a situation where sampling the full $\mathbf{f}$ is better avoided; either because of computational complexity or because the data arrives in a sequential manner. We use blocked Gibbs sampling (Geman & Geman, 1984) and alternate between updating latent variables conditional on hyperparameters, and hyperparameters conditional on latent values.

To this end, assume we have observed data $\mathbf{y}_1$ and generated an initial sample $\mathcal{S}_1 = \{(\mathbf{f}_1^{(1)}, \kappa^{(1)}, \alpha^{(1)}), \ldots, (\mathbf{f}_1^{(M)}, \kappa^{(M)}, \alpha^{(M)})\}$ where $(\mathbf{f}_1^{(i)}, \kappa^{(i)}, \alpha^{(i)})$ is the i^{th} generated state of the Markov chain targeting the posterior $p(\mathbf{f}_1, \kappa, \alpha|\mathbf{y}_1)$—see end of Section 2.4. For the subsequent data $\mathbf{y}_2$, a standard approach is to target the joint posterior

$$p(\mathbf{f}_{1:2}, \kappa, \alpha|\mathbf{y}_{1:2}) = \frac{1}{Z} p(\mathbf{y}_2|\mathbf{f}_2, \alpha) p(\mathbf{f}_2|\mathbf{f}_1, \kappa) p(\mathbf{f}_1, \kappa, \alpha|\mathbf{y}_1) \tag{2}$$

by blocked Gibbs sampling: When iterating over $i \in \{1, \ldots, M\}$, the state $\mathbf{f}_2^{(i)}$ is first generated conditioned on $(\mathbf{f}_1^{(i)}, \kappa^{(i)}, \alpha^{(i)})$ by targeting its conditional[1]

$$\propto p(\mathbf{y}_2|\mathbf{f}_2, \alpha^{(i)}) p(\mathbf{f}_2|\mathbf{f}_1^{(i)}, \kappa^{(i)}).$$

Second, $(\kappa^{(i)}, \alpha^{(i)})$ is updated conditioned on $(\mathbf{f}_2^{(i)}, \mathbf{f}_1^{(i)})$ by targeting its corresponding conditional, which includes every factor of (2). Third, the state $\mathbf{f}_1$ should also be revisited in an update conditional on $(\mathbf{f}_2^{(i)}, \kappa^{(i)}, \alpha^{(i)})$. In proceeding steps $t = 3, \ldots, T$ we continue to "build up" and update the latent states while hyperparameter updates target the

[1] Here we mean $\mathbf{f}_2^{(i)} \sim p(\mathbf{f}_2|\mathbf{f}_1^{(i)}, \kappa^{(i)}, \alpha^{(i)}, \mathbf{y}_{1:2}) = \frac{1}{Z} p(\mathbf{y}_2|\mathbf{f}_2, \alpha^{(i)}) p(\mathbf{f}_2|\mathbf{f}_1^{(i)}, \kappa^{(i)}) p(\mathbf{f}_1^{(i)}, \kappa^{(i)}, \alpha^{(i)}|\mathbf{y}_1)$, i.e., we ignore writing out the normalising constant and all irrelevant terms not depending on $\mathbf{f}_2$.

posterior over the growing sequence $\mathbf{f}_{1:t}$ for each t

$$\propto p(\mathbf{y}_t|\mathbf{f}_t,\alpha)p(\mathbf{f}_t|\mathbf{f}_{1:t-1},\kappa)p(\mathbf{f}_{1:t-1},\kappa,\alpha|\mathbf{y}_{1:t-1}). \quad (3)$$

The overall complexity of this procedure is $\mathcal{O}(T^3N^3)$, which quickly becomes prohibitive.

2.2. Sequentially approximating the posterior

In sequential data settings where, for example, data is collected daily, our interest is often sequential as well. When this is the case, computation is saved by sampling only the most recent $\mathbf{f}_t$ from

$$\propto p(\mathbf{y}_t|\mathbf{f}_t,\alpha)p(\mathbf{f}_t|\mathbf{f}_{1:t-1},\kappa), \quad (4)$$

i.e., ignoring updating previous $\mathbf{f}_s$ for $s < t$. The computational cost of the predictive prior in (4) grows with t (cubed). If $\mathbf{f}_t$ has strongest prior dependency with its most recent neighbours (as for isotropic kernels (1) with respect to t), variables separated in time may be dropped in order to limit temporal dependency to the $\tau \geq 1$ most recent steps. The approximate predictive distribution is then

$$p(\mathbf{f}_t|\mathbf{f}_{1:t-1},\kappa) \approx p(\mathbf{f}_t|\mathbf{f}_{t-\tau:t-1},\kappa) \quad (5)$$

with a cost-cap of τ. This naive form of data selection has the purpose of limiting the size of $\mathbf{f}_{1:t-1}$ from growing with t when evaluating the prior. τ can be set using domain knowledge or by tuning trade-off between computation and accuracy. For sophisticated approaches to selecting *which* latent variables to include in the predictive distribution, e.g., in the case of non-isotropic kernels, see (Osborne, 2010).

Sampling $\mathbf{f}_t$ is straightforward in the sequential procedure; the computationally problematic term in (3) is the posterior from the previous time step. To circumvent the difficulties of parameter sampling, we approximate it by a factorised version

$$p(\mathbf{f}_{1:t-1},\kappa,\alpha|\mathbf{y}_{1:t-1}) \approx q_{t-1}(\mathbf{f}_{1:t-1})q_{t-1}(\kappa,\alpha) \quad (6)$$

which is substituted into (3). The sampling target for $\mathbf{f}_t$ is unchanged, but the approximation yields simple updates for (κ,α) from $\propto p(\mathbf{y}_t|\mathbf{f}_t,\alpha)p(\mathbf{f}_t|\mathbf{f}_{1:t-1},\kappa)q_{t-1}(\kappa,\alpha)$. Note that the approximate term $q_{t-1}(\mathbf{f}_{1:t-1})$ factors out for both updates. In effect, if $q_{t-1}(\kappa,\alpha)$ is a *tractable* approximation which can be updated in each t-step, the same basic sampling method can be applied sequentially by combining (5)-(6): (i) $(\mathbf{f}_t,\kappa,\alpha) \sim p(\mathbf{y}_t|\mathbf{f}_t,\alpha)p(\mathbf{f}_t|\mathbf{f}_{t-\tau:t-1},\kappa)q_{t-1}(\kappa,\alpha)$, (ii) $q_{t-1}(\kappa,\alpha) \rightarrow q_t(\kappa,\alpha)$, (iii) $p(\mathbf{f}_t|\mathbf{f}_{t-\tau:t-1},\kappa) \rightarrow p(\mathbf{f}_{t+1}|\mathbf{f}_{t+1-\tau:t},\kappa)$, using the priors $p_h(\kappa,\alpha)$ and $p(\mathbf{f}_1|\kappa)$ for the initial $t = 1$. To make this scheme operational, we next consider a specific choice of $q_{t-1}(\kappa,\alpha)$.

2.3. Sequential approximation and hyperparameter assumptions

Given the factorisation assumption (6), a number of methods for estimating q_{t-1} are possible. We propose an approach that seems natural given the problem constraints, based on ideas from expectation propagation.

Consider minimizing the Kullback–Leibler divergence $\mathrm{KL}(p(\mathbf{f}_{1:t},\kappa,\alpha|\mathbf{y}_{1:t})||q_t(\kappa,\alpha))$. The optimal solution is $\hat{q}_t(\kappa,\alpha) = \int p(\mathbf{f}_{1:t},\kappa,\alpha|\mathbf{y}_{1:t})d\mathbf{f}_{1:t} = p(\kappa,\alpha|\mathbf{y}_{1:t})$, which is intractable in general while for our purpose, we desire a tractable representation. One such arises when (κ,α) is specified as a deterministic transformations of a Gaussian vector $\mathbf{z}$ (the next section gives an example): Assuming a Gaussian approximating family, the optimal $\hat{q}_t(\mathbf{z})$ is given by $\mathcal{N}(\mathbf{m}_{\mathbf{z},t},\mathbf{K}_{\mathbf{z},t})$ with

$$\mathbf{m}_{\mathbf{z},t} = \mathbb{E}_{p(\mathbf{z}|\mathbf{y}_{1:t})}[\mathbf{z}] \quad \text{and} \quad \mathbf{K}_{\mathbf{z},t} = \mathbb{E}_{p(\mathbf{z}|\mathbf{y}_{1:t})}[\mathbf{z}\mathbf{z}'] \quad (7)$$

i.e., by moment matching. This form for the parameters allows for sequential updates to produce a sample $(\mathbf{z}^{(1)},\ldots,\mathbf{z}^{(M)})$, which can be used to re-estimate the moments (7). The approximation thus breaks the full temporal dependence into a sequential dependence similar to a Markov decomposition, and focuses computational resources via MCMC on parts of the model with more complex structure.

For the hyperprior, we consider a scaled sigmoid Gaussian (SSG)

$$\kappa = \kappa_{\min} + \frac{\kappa_{\max} - \kappa_{\min}}{1+\exp(-\mathbf{z})}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{m}_{\mathbf{z}},\mathbf{K}_{\mathbf{z}}) \ , \quad (8)$$

for covariance parameters, and similarly for α. We make this choice for a couple of reasons: (i) it is convenient for specifying the range of each parameter, $\kappa_i \in (\kappa_{\min,i},\kappa_{\max,i})$, as well as joint dependency and distributional shape with $\mathbf{m}_{\mathbf{z}}$, $\mathbf{K}_{\mathbf{z}}$, and (ii) as the parameter vector is effectively a transformed Gaussian, we can leverage ESS when updating hyperparameters in the sampling scheme. For a non-informative prior on may set $\mathbf{m}_{\mathbf{z}} = 0$ and $\mathbf{K}_{\mathbf{z}} = 1.5^2\mathbb{I}$ while domain knowledge can be used for setting $\kappa_{\max}$ and $\alpha_{\max}$. We can then take our posterior representation $q_{t-1}(\kappa,\alpha)$ to also be a SSG with $\mathbf{m}_{\mathbf{z}}$ and $\mathbf{K}_{\mathbf{z}}$ updated from the previous hyperparameter sample, i.e., moments estimated from the posterior sample $\{\alpha^{(1)},\kappa^{(1)},\ldots,\alpha^{(M)},\kappa^{(M)}\}$, generated at $t-1$.

2.4. Algorithm

At the backbone of our algorithm sits the elliptical slice sampler—see Figure 2 of (Murray et al., 2010). For a given t, we use it for sampling from our joint target (non-normalised, approximative posterior)

$$\pi(\mathbf{f}_t,\kappa,\alpha|\mathbf{y}_t) = p(\mathbf{y}_t|\mathbf{f}_t,\alpha)p(\mathbf{f}_t|\mathbf{f}_{t-\tau:t-1},\kappa)q_{t-1}(\kappa,\alpha) \quad (9)$$

in three steps:

Update 1: the latent variable $\mathbf{f}_t$ for fixed $(\kappa,\ \alpha,\mathbf{f}_{t-\tau:t-1})$ with ESS. Here we have to compute only once the covariance matrix $\mathbf{K}_{t,\kappa}$ (square-root decomposition $\mathbf{K}_{t,\kappa} =$

$\mathbf{L}_{t,\kappa}\mathbf{L}_{t,\kappa}^\top$) and mean $\mathbf{m}_{t,\kappa}$ of the conditional prior $\mathbf{f}_t|\mathbf{f}_{t-\tau:t-1} \sim \mathcal{N}(\mathbf{m}_{t,\kappa}, \mathbf{K}_{t,\kappa})$ using predictive equations for Gaussian processes (Rasmussen & Williams, 2006). Since κ is fixed, $\mathbf{K}_{t,\kappa}$ and $\mathbf{m}_{t,\kappa}$ stay unchanged when applying ESS so it is sensible to repeat this operation and update $\mathbf{f}_t$ several times.

Update 2: covariance parameters κ for fixed α. For this purpose we represent the conditional prior of $\mathbf{f}_t$ with a spherical Gaussian

$$\mathbf{f}_t = \mathbf{L}_{t,\kappa}\boldsymbol{\nu} + \mathbf{m}_{t,\kappa}, \quad \boldsymbol{\nu} \sim \mathcal{N}(0, \mathbb{I}) \tag{10}$$

and generate κ for fixed $\boldsymbol{\nu}$ with respect to $p(\mathbf{y}_t|\mathbf{f}_t = \mathbf{L}_{t,\kappa}\boldsymbol{\nu} + \mathbf{m}_{t,\kappa}, \alpha)q_{t-1}(\kappa, \alpha)$. Note that this will also update $\mathbf{f}_t$ as a bi-product. Due to the posterior representation of κ with a sigmoid Gaussian, we perform this in $\mathbf{z}$-space; again with ESS targeting $p(\mathbf{y}_t|\mathbf{f}_t(\mathbf{z}), \alpha)\mathcal{N}(\mathbf{z}; \mathbf{m}_\mathbf{z}, \mathbf{K}_\mathbf{z})$ where $\mathbf{f}_t(\mathbf{z}) = \mathbf{L}_{t,\kappa=\text{ssg}(\mathbf{z})}\boldsymbol{\nu} + \mathbf{m}_{t,\kappa=\text{ssg}(\mathbf{z})}$. Here $\kappa = \text{ssg}(\mathbf{z})$ is the transformation given in (8).

Update 3: likelihood parameters α for fixed $(\kappa, \mathbf{f}_t)$. As for κ, this is done in the underlying Gaussian with ESS targeting $p(\mathbf{y}_t|\mathbf{f}_t, \alpha = \text{ssg}(\mathbf{z}))\mathcal{N}(\mathbf{z}; \mathbf{m}_\mathbf{z}, \mathbf{K}_\mathbf{z})$. Note that this update does not involve the conditional prior of $\mathbf{f}_t$. Thus, it is relatively cheap since it only requires likelihood evaluations.

Before stepping to the next t, it make sense to perform some additional ESS-updates of $\mathbf{f}_t$, given updated parameters. This may also be done in an intermediate update between update 1 and update 2. Algorithm 1 summarises all steps: to sample the full $\mathbf{f}_{1:T}$ and hyperparameters, we apply the above three updates for $i = 1, \ldots, M$ in an inner loop to obtain the sample $\mathcal{S}_t$ given $\mathcal{S}_{t-1}$, and sequentially for $t = 2, \ldots, T$ in an outer loop.

Initial sample When generating $\mathcal{S}_1$ it is worth investing in a Markov chain with a large number of states to guarantee a good representation of the initial posterior. Generated variables and parameters are used for conditioning $\mathbf{f}_2$ and for re-estimating the hyperposterior. Updates of the subsequent $\mathcal{S}_2$ are therefore more efficient if the transient phase of the first chain has been discarded as burn-in. One also benefits from subsampling, since the size of $\mathcal{S}_1$ will fix M for all subsequent samples.

As the strength of the unconditional prior over $\mathbf{f}_1$ is often relatively weak to the likelihood, updating covariance parameters with a fixed-$\boldsymbol{\nu}$ representation (10) is likely to mix poorly. This is discussed in (Murray & Adams, 2010) who propose the more efficient SDSS. Thus, it is advantageous to use their method for sampling parameters at the first $t = 1$; especially since we can adopt an ESS update (in $\mathbf{z}$-space) within SDSS under our hyperprior/posterior assumptions. For subsequent t:s, however, we expect $\mathbf{f}_t$ to have strong ties with selected variables of $\mathbf{f}_{1:t-1}$. Therefore, we can indeed expect to have an informative (conditional) prior, such that a fixed-$\boldsymbol{\nu}$ update is likely to be efficient. In doubt, one may always apply SDSS for every time step.

Algorithm 1 Sequential sampling

Input: Initial sample $\mathcal{S}_1$ of size M. **Output:** Samples $\mathcal{S}_2, \ldots, \mathcal{S}_T$.

for $t = 2$ to T **do**
  Update: $\mathbf{m}_\mathbf{z} = \text{mean}(\mathbf{z}_{t-1})$, $\mathbf{K}_\mathbf{z} = \text{cov}(\mathbf{z}_{t-1}, \mathbf{z}_{t-1})$ with $\mathbf{z}_{t-1} = \text{ssg}^{-1}((\kappa^{(1:M)}; \alpha^{(1:M)}))$ from $\mathcal{S}_{t-1}$; likelihood function $p(\mathbf{y}_t|\cdot)$ based on new data $\mathbf{y}_t$.
  for $i = 1$ to M **do**
    Initiate: $\mathbf{f}_{t-1}^{(i)}$ from $\mathcal{S}_{t-1}$; κ, α from most recent updates.
    Initial draw: $\mathbf{f}_t^{(i)} \sim \mathcal{N}(\mathbf{m}_{t,\kappa}, \mathbf{K}_{t,\kappa})$ with $\mathbf{m}_{t,\kappa}$, $\mathbf{K}_{t,\kappa}$ calculated from κ and $\mathbf{f}_{t-1}^{(i)}$.
    Update 1: $\mathbf{f}_t^{(i)} \sim \text{ESS}(\mathbf{f}_t^{(i)}; \mathbf{m}_{t,\kappa}, \mathbf{K}_{t,\kappa})$ $\times$1–10.
    Update 2: $\kappa^{(i)}, \mathbf{f}_t^{(i)} \sim \text{ESS}(\kappa; \mathbf{m}_\mathbf{z}, \mathbf{K}_\mathbf{z})$ with fixed $\boldsymbol{\nu}$.
    Repeat update 1.
    Update 3: $\alpha^{(i)} \sim \text{ESS}(\alpha; \mathbf{m}_\mathbf{z}, \mathbf{K}_\mathbf{z})$.
    Repeat update 1.
  end for
  Save: $\mathcal{S}_t = \{\mathbf{f}_t^{(i)}, \kappa^{(i)}, \alpha^{(i)}\}_{i=1}^M$
end for

3. Conceptual analysis

Detailed theoretical analysis of the sequential approximation is beyond the scope of this paper. Rigorous theoretical theoretical treatment of MCMC sampling from an approximate posterior can be found in, e.g., Pillai & Smith (2014); Johndrow et al. (2015); Johndrow & Mattingly (2017). Here we examine our method from a high-level conceptual perspective.

The proposed method introduces an approximation of the true posterior at each time step. By limiting the temporal dependence between the current time step and previous time steps, the method induces a smaller effective sample size for performing inference at the current time step. As a result, the variance of the approximate posterior will in general be larger than that of the true posterior. Such an effect is seen in the experiments, particularly in Figures 1 and 3. In practice, restricting the influence of earlier time steps may have beneficial effects, particularly if there is model mis-specification, e.g., if the data are not stationary. Related ideas are explored in broader context in Jacob et al. (2017). As we show below, if the ratio of the approximate posterior variance to the true posterior variance grows sub-quadratically in the number of time steps, M, then the sequential approximation may reduce overall error.

Bias-variance trade-off The sequential algorithm proposed in Section 2 introduces bias: At each t, the method generates exact samples from an approximate posterior distribution. Conversely, estimates made with those samples will exhibit smaller variance than an unbiased MCMC method due to the larger number of samples generated by the sequential approximation per unit of computation time. In particular, let φ be some function of $\mathbf{f}_T, \kappa, \alpha$, and suppose we wish to compute $I^\varphi = \mathbb{E}_{p(\mathbf{f}_T,\kappa,\alpha|\mathbf{y}_{1:T})}[\varphi]$. Let $\hat{I}^\varphi_{\text{SGP}}(s) := \frac{1}{s}\sum_{i=1}^s \varphi(\mathbf{f}_T^i, \kappa^i, \alpha^i)$ denote the estimate of I^φ based on s independent samples from the sequential GP approximation, and likewise for $\hat{I}^\varphi_{\text{MCMC}}(s)$. The mean square error (MSE) of $\hat{I}^\varphi_{\text{SGP}}(s)$ is

$$\begin{aligned}\text{MSE}(\hat{I}^\varphi_{\text{SGP}}(s)) &= \mathbb{E}[(\hat{I}^\varphi_{\text{SGP}}(s) - I^\varphi)^2] \\ &= \text{Var}[\hat{I}^\varphi_{\text{SGP}}(s)] + b^2_{\varphi,T} \ ,\end{aligned}$$

where $b_{\varphi,T}$ is the bias of the sequential approximation after T time steps. For an unbiased MCMC method, $\text{MSE}(\hat{I}^\varphi_{\text{MCMC}}(s)) = \text{Var}[\hat{I}^\varphi_{\text{MCMC}}(s)]$. Therefore, assuming that $\text{Var}_{p(\mathbf{f}_T,\kappa,\alpha|\mathbf{y}_{1:T})}[\varphi] = \sigma^2_T < \infty$ for the true posterior and likewise $\sigma^2_{\text{SGP},T} < \infty$ for the sequential approximation,

$$\begin{aligned}\Delta &:= \text{MSE}(\hat{I}^\varphi_{\text{MCMC}}(s)) - \text{MSE}(\hat{I}^\varphi_{\text{SGP}}(s)) \\ &= \frac{\sigma^2_T}{s} - \frac{\sigma^2_{\text{SGP},T}}{s} - b^2_{\varphi,T} \ .\end{aligned}$$

For a fixed amount of computation time, t_c, Δ may be approximated using the complexity of the sampling algorithms (with C and C' capturing the hidden constants for MCMC and sequential sampling, respectively),

$$\Delta \approx \tilde{\Delta} := \sigma^2_T \frac{CT^3N^3}{t_c} - \sigma^2_{\text{SGP},T}\frac{C'T\tau^3N^3}{t_c} - b^2_{\varphi,T} \ .$$

As discussed above, the approximate posterior will in general have higher variance than the true posterior; denote the ratio of the two as $R_T := \sigma^2_{\text{SGP},T}/\sigma^2_T \geq 1$. Therefore, the sequential approximation yields lower error than unbiased MCMC if $\tilde{\Delta} > 0$; equivalently,

$$b^2_{\varphi,T} < \frac{\sigma^2_T TN^3}{t_c}(CT^2 - C'R_T\tau^3) \ . \qquad (11)$$

In order for the r.h.s. to be positive, $R_T < \frac{CT^2}{C'\tau^3}$. Experiments indicate that R_T is relatively stable for modest T (see Figures 1 and 3); assuming positivity for large enough T indicates that the sequential approximation will reduce MSE if the bias grows at most as $\sqrt{T}$. The experiments in Section 4 provide evidence that the approximation bias does not grow quickly for the cases considered there.

4. Experiments

To demonstrate empirical performance of our approach, we apply sequential sampling to a GP regression model on synthetic data in the next section. For brevity, we omit further standard applications, such as GP classification and Cox-process inference. Instead we concentrate on a more complicated financial model with a nonlinear likelihood from an option pricing problem. Since our experiments are of much larger scale than what is typically considered by the literature (TN observations versus N), we do not perform a comparative study of methods, but concentrate on sample result and their representativeness of the ground truth.

4.1. Gaussian Process regression for 3D inputs

We consider the Gaussian regression problem from (Murray et al., 2010) as our starting point. For each t, data $\mathbf{y}_t$ are noisy observations of latent values $\mathbf{f}_t$ taken at input locations $\mathbf{x}_t$. We set up the data to have $N = 200$ observations for each t, with inputs $\mathbf{x}_t$ drawn uniformly over a unit square. Latent values are generated from a GP prior with squared-exponential kernel of parameters $\kappa = (l_{x_1}, l_{x_2}, l_t, \sigma_f)$,

$$k(t,x;t',x') = \sigma_f^2 \exp\left(-\sum_{d=1}^2 \frac{(x_{(d)} - x'_{(d)})^2}{2l^2_{x_d}} - \frac{(t-t')^2}{2l_t^2}\right).$$

We let $\sigma_f = 1$ and draw length-scales uniformly over $(0, \sqrt{10})$. We set $(\mu_f, \sigma_y) = (0.5, .3)$ for the likelihood $p(\mathbf{y}_t|\mathbf{f}_t, \alpha) = \mathcal{N}(\mathbf{y}_t|\mathbf{f}_t + \mu_f, \sigma_y^2\mathbb{I})$, and $\kappa_{\max} = (\sqrt{10}, \sqrt{10}, \sqrt{10}, 2)$ and $\alpha_{\max} = (1,1)$ for the hyperprior. Finally, we generate the full data set with $T = 20$ steps of t equally spaced over the unit interval. Each of $\mathbf{y}_{1:T}$, $\mathbf{f}_{1:T}$, $\mathbf{x}_{1:T}$, thus has $TN = 4000$ elements.

For $t = 1$, we generate 6000 states by the algorithm outlined in Section 2.4 for the initial sample, with three $\mathbf{f}_1 \sim$ ESS updates in the first update (running time < 4 minutes on a 2.8GHz quad-core Intel i7 processor). Note that we target the exact posterior and therefore l_t can not be inferred as we see observed data for a singe t only. We discard the first 1000 states as burn-in and keep every 5^{th} state to obtain a thinned sample $\mathcal{S}_1$ of size $M = 1000$. We then continue to generate samples $\mathcal{S}_2, \mathcal{S}_3, \ldots, \mathcal{S}_{20}$ sequentially from their approximative posteriors: For each t, we generate $\mathcal{S}_t$ from the target $\pi(\mathbf{f}_t, \kappa, \alpha|\mathbf{y}_t)$ in (9) based on the previous states of $\mathcal{S}_{t-1}$. We drop all but $\tau = 1$ variable. In each iteration, we repeat $\mathbf{f}_t \sim$ ESS $\times$ 5 in the first update, and another five repetitions between the second and third update of the parameters (total running time 35min).

Posterior representation Input #100–200 of Figure 1 represents the posterior over the initial latent variable $\mathbf{f}_1$ (μ_f included) from $\mathcal{S}_1$. The sample captures the latent process to a good extend: 85% of true values (blue stars) fall within the $\pm$2SD region. A credible interval for $\mathbf{y}_1$ is also calculated from the sample of $\mathbf{f}_1$ and σ_y (light grey). Comparing to observed data (red dots) we see a good representation with a few data points lying outside the interval (8 out of 200). Sequential samples over $\mathbf{f}_t$ and credible intervals for

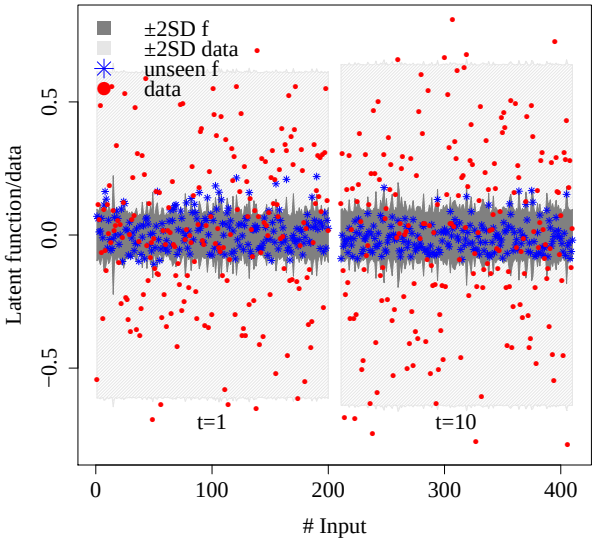

Figure 1. Inputs 1–200 show $\mathcal{S}_1$ from the GP regression problem. The posterior over $\mathbf{f}_1$ is represented by a $\pm$2SD region in dark grey. Light grey is the $\pm$2SD credible interval for corresponding data. For clarity, the sample mean of $\mathbf{f}_1$ is subtracted from all values, such that the posterior is centered around zero. Inputs 201–400 show corresponding results for $\mathcal{S}_{20}$.

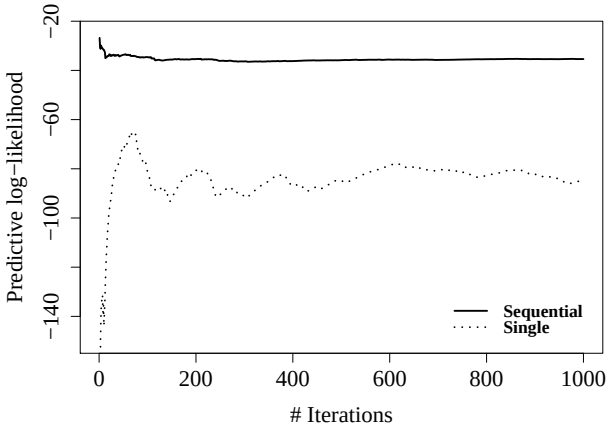

Figure 2. GP regression. The predictive distribution $\mathbf{f}_{20}^{\star}|\mathbf{f}_{20}^{o(i)}$ is based on $\mathbf{f}_{20}^{o(i)}$ sampled with the sequential approximation (solid line), and from a single MCMC with no dependence on previous time steps for $t = 20$ (dotted line), respectively. The higher predictive log-likelihood of the sequential approximation indicates that it captures statistical information from previous time steps.

$\mathbf{y}_t$ yield similar results for each t as seen for $\mathcal{S}_1$. Plots are not shown here for brevity, except for results from $\mathcal{S}_{20}$ shown as #201–400 in Figure 1. The samples exhibit a good representation of the ground truth: all latent values fall inside the $\pm$2SD posterior sample, and data observations fall inside their $\pm$2SD credible interval (9 out of 200).

Posterior prediction To further demonstrate the applicability of our approach, we do a prediction experiment as follows. Given samples up to and including $\mathcal{S}_{t-1}$, we divide the subsequent data into a training set $\mathbf{y}_t^o$ and a test set $\mathbf{y}_t^\star$ of equal size.[2] We sample $\mathbf{f}_t^o$ from the target (9) with likelihood given by $\mathbf{y}_t^o$. From each such sample $\mathbf{f}_t^{o(i)}$, we compute the predictive mean $\mathbf{m}_t^{\star(i)}$ and covariance $\mathbf{K}_t^{\star(i)}$ of the test variables $\mathbf{f}_t^\star|\mathbf{f}_t^{o(i)}$ and represent their predictive power by the predictive likelihood of $\mathbf{y}_t^\star$, based on $\mathbf{m}_t^{\star(i)}$ and $\mathbf{K}_t^{\star(i)}$. Denoting the corresponding log-likelihood value $ll^{\star(i)}$, we then look at how the accumulative mean of $ll^{\star(1:i)}$ varies with iteration i. This gives an idea of how predictive power changes as an increasing number posterior samples of $\mathbf{f}_t^o$ are used; Figure 2 shows the result for $t = 20$.

For comparison, we repeat the experiment but with no previous information from $\mathcal{S}_{t-1}$. That is, we use the "initial" sampling scheme based only on data $\mathbf{y}_t^o$, breaking all dependence with earlier time steps. The resulting predictive likelihood of $\mathbf{y}_t^\star$—shown in Figure 2 for $t = 20$—is now weaker, indicating that the sequential approximation is capturing information from previous time steps.

Comparison with full sampling For comparison, we use the full, non-approximate MCMC method on a GP regression model with $N = 100$ and $T = 10$. We sample the full model—non-sequentially, without posterior approximations—with the **initial sample** method. We generate 6000 MCMC samples of $\mathbf{f}_{1:t}$ for $t = 1, \ldots, 10$; computation time is 1 minute for $\mathbf{f}_1$ up to 11 hours for $\mathbf{f}_{10}$. Similarly, we generate 6000 samples with sequential sampling, which takes 3 minutes. The sub-sampled result is illustrated in Figure 3. In qualitative terms, the main difference is that the overall variance decreases in t for the full sampling method because it uses the full data set $\mathbf{y}_{1:t}$ (up to a certain point $\sim$ 300 inputs), while sequential sampling has a constant, higher sample variance due to the approximations, which limit temporal dependence. Note also that the seq. posterior does not deteriorate with t, indicating that sequential approximation does not introduce a bias which *accumulates*. Similarly it does not drift, such that the bias is *stable* across the sequence.

4.2. Option pricing problem

As a second application of our approach, we consider inferring a latent positive function $\sigma(T, K, t)$ of an option

[2] This is simply done by assigning all input points of $\mathbf{x}_t$ lying in the half plane $[0, 0.5] \times [0, 1]$ to the training set, and remaining points to the test set.

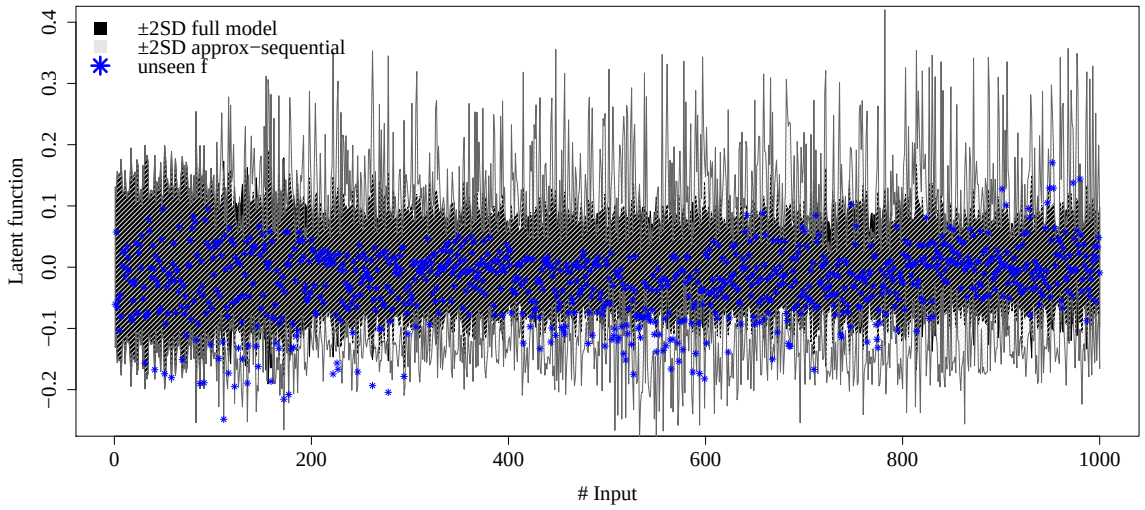

Figure 3. Comparison of approximative-sequential sampling vs. sampling the full model.

pricing model (Dupire, 1994). For fixed t, the model is represented by a mapping $\sigma(\cdot, t) \mapsto C(\cdot)$ where the price function $C(T, K)$ solves the PDE

$$\frac{\partial C}{\partial T} + rK\frac{\partial C}{\partial K} - \frac{K^2\sigma^2(T,K)}{2}\frac{\partial^2 C}{\partial K^2} = 0 \qquad (12)$$

with boundary condition $C(0, K) = (S_t - K)^+$. The function $C(T, K)$ yields the time-t price of a call option with maturity T and strike K on an underlying asset with current price S_t.

The construction and calibration of σ from a set of observed market prices at a single date t is a problem commonly encountered in quantitative finance—see for instance the discussion by (Luo & Liu, 2010). We consider it here as an example of a challenging likelihood and inference problem naturally placed in a sequential context. Given $\boldsymbol{\sigma}_t = \sigma(\mathbf{x}_t)$, where $\mathbf{x}_t$ is the set of (T, K)-inputs observed at t, we take observed call prices $\mathbf{c}_t$ to be generated with Gaussian noise

$$p(\mathbf{c}_t|\boldsymbol{\sigma}_t) = \mathcal{N}(\mathbf{c}_t; C(\mathbf{x}_t; \boldsymbol{\sigma}_t), \sigma_c^2\mathbb{I}).$$

Here, $C(\mathbf{x}_t; \boldsymbol{\sigma}_t)$ are model prices for each strike-maturity in $\mathbf{x}_t$, calculated with $\boldsymbol{\sigma}_t$. Further, we place a zero-mean GP prior on $\mathbf{f} = \{\mathbf{f}_t\}_{t=1}^T$ and use the "softplus" function $\zeta(f) = \log(1 + \exp(f))$ to impose positiveness; $\boldsymbol{\sigma}_t = \zeta(\mathbf{f}_t + \mu_f)$. In effect, the likelihood factorises over time components $p(\mathbf{c}_t|\zeta(\mathbf{f}_t + \mu_f))$. However, it does not factorise over the components of $\mathbf{f}_t$, it is highly nonlinear (thus non-Gaussian) and intractable—it can not even be evaluated with a closed-form expression.[3]

[3] Since no closed-form solution of (12) is known for a general function σ, we follow standard procedure and use a numerical Crank-Nicolson solver—see e.g. (Hirsa, 2012).

We generate data with $T = 12$ equidistant time steps and $N = 75$ observations for each t from $\mathbf{x}_t$ placed at a grid of 15 strikes $\times$ 5 maturities. Each $C(\mathbf{x}_t; \boldsymbol{\sigma}_t)$ is computed with $\boldsymbol{\sigma}_t$ from a draw $\mathbf{f}_{1:T}$ of a GP with squared-exponential kerned. We use $(l_T, l_K, l_t, \sigma_f) = (0.5, 0.3, 0.5, 0.75)$ and $(\mu_f, \sigma_c) = (-1.5, 0.05)$ for the likelihood. For the prior, $\kappa_{\text{max}} = (1, 1, 1, 1)$ and $\alpha_{\text{max}} = (0.5, 0.5)$. We set an interest rate $r = 0$ while the underlying price $\{S_t\}_{t=1}^M$ is simulated from a geometric Brownian motion with $(S_1, \mu, \sigma) = (1000, 0.04, 0.2)$.

Sequential sampling from the posterior is carried out in the same manner as for the regression problem of Section 4.1. We generate 20,000 states for $t = 1$ (running time 18min), discard 10% as burn-in and subsample to obtain $\mathcal{S}_1$ of size 1000 before continuing with $\mathcal{S}_2, \ldots, \mathcal{S}_{12}$ (total time 24min).

Posterior representation For the last sequential step, left Figure 4 shows the posterior sample of latent variables $\boldsymbol{\sigma}_{12} = \zeta(\mathbf{f}_{12} + \mu_f)$. The posterior sample covers true latent values to a satisfactory extent. We also see that the MAP estimate of $\boldsymbol{\sigma}_{12}$ is close to true values. More interesting is that the *uncertainty* in the posterior clearly varies over the input space. This is a consequence of the non-linearity of the likelihood, since the sensitivity of C with respect to σ varies with maturity and strike (c.f. variability over latent variables in Figure 1). In left (and right) Figure 4, inputs are ordered in groups of five with common strike. Taking one such group from the right hand side of the (left) plot where **strikes are low**, the variability over $\boldsymbol{\sigma}$ is large. The reason is that a low strikes gives a call option deep *'in-the-money'*, such that its pay-off is relatively certain, and thereby its price insensitive to σ. In effect, the likelihood is relatively uninformative about $\boldsymbol{\sigma}$ in low-strike regions.

The corresponding posterior sample over option prices $\mathbf{c}_{12}$

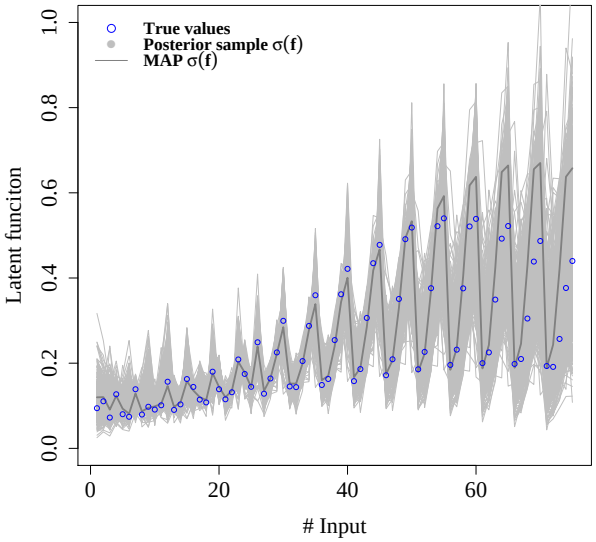

Figure 4. The inferred latent function $\boldsymbol{\sigma}_{12} = \zeta(\mathbf{f}_{12} + \mu_f)$ from $\mathcal{S}_{12}$ for the option pricing problem. The posterior sample is shown in grey, its MAP estimate with a solid grey line and true latent values with blue circles.

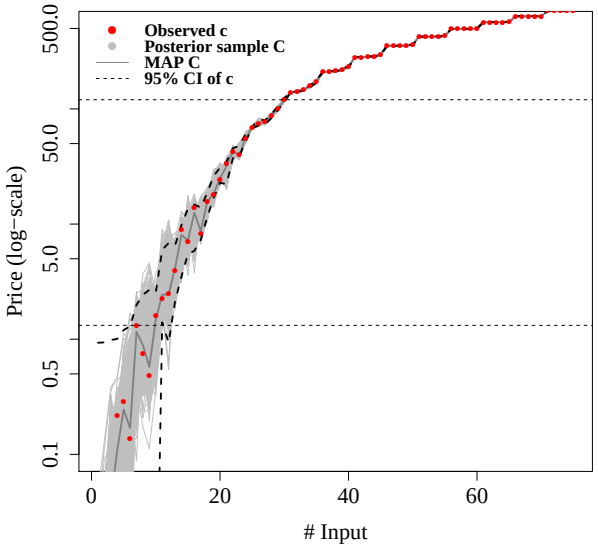

Figure 5. Option pricing problem; $t = 12$ sample. The estimated posterior of the model price is shown in logarithmic scale (grey area), corresponding to the latent function shown in Figure 4. Prices are plotted in groups of five maturities with common strike (strike-price in descending order with # Input). Thick dashed lines represents a posterior 95% credible interval for observable prices (model price + noise) while red dots shows observed values used for inference. The two horizontal lines indicates which options are close to at-the-money.

is shown in right Figure 4 in log-scale. The non-linearity effect of the pricing function is clearly manifested: even if there is large uncertainty about $\boldsymbol{\sigma}$ for low strikes (left Figure 4, # Input > 40) there is little uncertainty over corresponding prices (right Figure 4, # Input > 30, albeit the log-scale). More interesting is the options around *at-the-money*, with prices between the two horizontal lines (inputs $\sim$ 10–30). These options are the most actively traded, and hence a good fit of model to market prices is desirable. Taking the MAP-estimate (the sample surface $\boldsymbol{\sigma}_{12}^{(i)}$ which achieves largest posterior likelihood) this is indeed the case: observed data falls close to MAP prices in right Figure 4. Further, the full posterior sample demonstrates how *parameter uncertainty*—inherent in a model when estimated form data—is distributed across strikes and maturities. This representation of uncertainty is important, not the least as it should be taken into account when the model is used for prediction. Finally, we note that the results for $t \in \{1, \ldots, 11\}$ are very similar in quality to those discussed in the above, but not shown for brevity: the efficiency of our sequential sampling procedure is consistent across t.

5. Conclusion and discussion

In this paper we have proposed a computationally efficient sampling strategy that applies to Bayesian inference for GP-based latent variable models with sequentially increasing data. We proposed a practical approach based on a tactical approximation that (i) breaks the joint posterior from the previously sampled step into its marginals over latent variables and parameters, (ii) represents this parameter marginal with a transformed Gaussian to enable it being updated from its the recent sample, and (iii) drops variable in the conditional prior over latent variables. We demonstrated its benefits for a standard GP regression model on synthetic data of size that would be unpractical with standard sampling, and for a complicated option-pricing model with highly nonlinear likelihood. Both exampled showed strong performance of our method with good posterior representation of the ground truth. For the regression problem, we also shows that it is comparable with full sampling.

Our sampling scheme will not produce outcomes from the true posterior distribution as it targets the approximation (9). This is the price we have to pay for computational efficiency. The approximation will perform best when the data are highly informative: $p(\mathbf{f}_{1:t}|\mathbf{y}_{1:t}, \kappa, \alpha) \approx p(\mathbf{f}_{1:t}|\mathbf{y}_{1:t})$ and when there is an isotropic dependency structure over time. In our examples, this is the case. The savings in computation can be substantial: If the conditional prior of $\mathbf{f}_t \in \mathbb{R}^N$ is capped to τ previous labels, we have a reduced overall complexity from $\mathcal{O}(T^3N^3)$ to $\mathcal{O}(T\tau^3N^3)$ for T sequential updates.

Acknowledgments

The research leading to these results has received funding from the Oxford-Man Institute of Quantitative Finance (MT and SR), and European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) ERC grant agreement no. 617071 (BBR).

References

Barenco, M., Tomescu, D., Brewer, D., Callard, R., Stark, J., and Hubank, M. Ranked prediction of p53 targets using hidden variable dynamic modeling. *Genome biology*, 7 (3):R25, 2006.

Dupire, B. Pricing with a smile. *Risk*, 7(1):18–20, 1994.

Geman, S. and Geman, D. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.

Hirsa, A. *Computational methods in finance*. CRC Press, 2012.

Jacob, P. E., Murray, L. M., Holmes, C. C., and Robert, C. P. Better together? statistical learning in models made of modules. 08 2017. URL https://arxiv.org/pdf/1708.08719.

Johndrow, J. E. and Mattingly, J. C. Error bounds for approximations of markov chains. 11 2017. URL https://arxiv.org/pdf/1711.05382.

Johndrow, J. E., Mattingly, J. C., Mukherjee, S., and Dunson, D. Optimal approximating markov chains for bayesian inference. 08 2015. URL https://arxiv.org/pdf/1508.03387.

Luo, C. and Liu, X. Local volatility model. *Encyclopedia of Quantitative Finance*, 2010.

Minka, T. P. Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pp. 362–369. Morgan Kaufmann Publishers Inc., 2001.

Murray, I. and Adams, R. P. Slice sampling covariance hyperparameters of latent gaussian models. In *Advances in Neural Information Processing Systems*, pp. 1732–1740, 2010.

Murray, I., P Adams, R., and MacKay, D. J. Elliptical slice sampling. In *AISTATS*, 2010.

Osborne, M. *Bayesian Gaussian Processes for Sequential Prediction, Optimisation and Quadrature*. PhD thesis, University of Oxford, 2010.

Pillai, N. S. and Smith, A. Ergodicity of approximate mcmc chains with applications to large data sets. 05 2014. URL https://arxiv.org/pdf/1405.0182.

Quiñonero-Candela, J. and Rasmussen, C. E. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959, 2005.

Rasmussen, C. E. and Williams, C. K. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, 2006.

Rue, H., Martino, S., and Chopin, N. Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations. *Journal of the royal statistical society: Series b (statistical methodology)*, 71(2): 319–392, 2009.

Saatchi, Y. *Scalable inference for structured Gaussian process models*. PhD thesis, University of Cambridge, 2012.

Williams, C., Klanke, S., Vijayakumar, S., and Chai, K. M. Multi-task gaussian process learning of robot inverse dynamics. In *Advances in Neural Information Processing Systems*, pp. 265–272, 2009.