Sparse Algorithms for Markovian Gaussian Processes

William Wilkinson[1,2]
Arno Solin[1]
Vincent Adam[1,3]

Aalto University
Secondmind

[1]Aalto University, [2]The Alan Turing Institute, [3]Secondmind.ai

Artificial Intelligence and Statisitcs (AISTATS) 2021

Motivation

- GPs for time series can be reformulated as state-space models (Markov GPs) with efficient inference via filtering and smoothing.
- Many operations during inference can be parallelised, but GPUs do not handle sequential filtering efficiently.

Motivation

- GPs for time series can be reformulated as state-space models (Markov GPs) with efficient inference via filtering and smoothing.
- Many operations during inference can be parallelised, but GPUs do not handle sequential filtering efficiently. **How can we speed things up?**

Markovian Gaussian Processes on 1D inputs, x,

$$f \sim GP(0, \kappa(x, x')) \longleftrightarrow \begin{cases} \frac{d\mathbf{s}(x)}{dx} = \mathbf{F}\mathbf{s}(x) + \mathbf{L}w(x) \\ f(x) = \mathbf{H}\mathbf{s}(x) \end{cases}$$

Sparse Markovian Gaussian processes (Adam et. al., 2020)

Markovian Gaussian Processes on 1D inputs, x,

$$f \sim GP(0, \kappa(x, x')) \longleftrightarrow \begin{cases} \frac{d\mathbf{s}(x)}{dx} = \mathbf{F}\mathbf{s}(x) + \mathbf{L}w(x) \\ f(x) = \mathbf{H}\mathbf{s}(x) \end{cases}$$

Sparse Markov GPs define inducing states $\mathbf{u} = \mathbf{s}(\mathbf{z})$: evaluations of the SDE at a reduced set of inputs $\mathbf{z} = \mathbf{z}_1, \ldots, \mathbf{z}_M$.

Markovian Gaussian Processes on 1D inputs, x,

$$f \sim GP(0, \kappa(x, x')) \longleftrightarrow \begin{cases} \frac{d\mathbf{s}(x)}{dx} = \mathbf{F}\mathbf{s}(x) + \mathbf{L}w(x) \\ f(x) = \mathbf{H}\mathbf{s}(x) \end{cases}$$

Sparse Markov GPs define inducing states $\mathbf{u} = \mathbf{s}(\mathbf{z})$: evaluations of the SDE at a reduced set of inputs $\mathbf{z} = \mathbf{z}_1, \ldots, \mathbf{z}_M$.

The number of sequential steps is reduced from N to M — good for GPU processing.

Markovian Gaussian Processes on 1D inputs, x,

$$f \sim GP(0, \kappa(x, x')) \longleftrightarrow \begin{cases} \frac{d\mathbf{s}(x)}{dx} = \mathbf{F}\mathbf{s}(x) + \mathbf{L}w(x) \\ f(x) = \mathbf{H}\mathbf{s}(x) \end{cases}$$

Sparse Markov GPs define inducing states $\mathbf{u} = \mathbf{s}(\mathbf{z})$: evaluations of the SDE at a reduced set of inputs $\mathbf{z} = \mathbf{z}_1, \ldots, \mathbf{z}_M$.

The number of sequential steps is reduced from N to M — good for GPU processing.

Compute: $\mathcal{O}((N_* + M)d^3)$ Memory: $\mathcal{O}(Md^2)$

Site-Based Inference I

Our main contribution is to develop site-based inference for sparse Markov GPs.

Approximate the GP posterior as:

$$p(\mathbf{f} \mid \mathbf{y}) \propto p(\mathbf{f}) \prod_n p(y_n \mid f_n) \approx \int p(\mathbf{u}) p(\mathbf{f} \mid \mathbf{u}) \prod_m t_m(\mathbf{u}_m, \mathbf{u}_{m+1}) \, \mathrm{d}\mathbf{u}$$

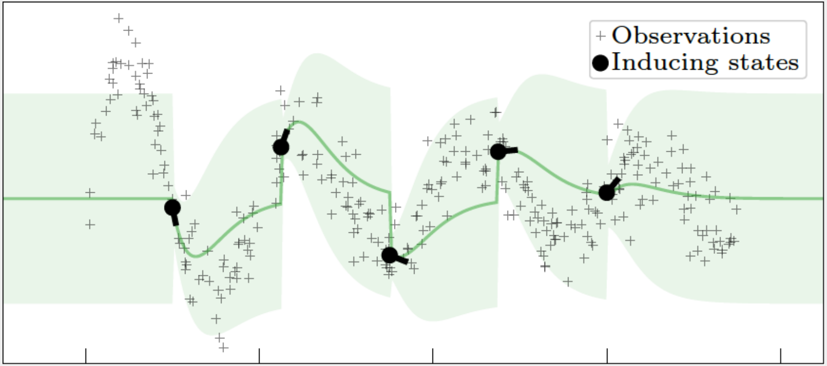
where the approximate likelihoods (sites), $t_m(\mathbf{u}_m, \mathbf{u}_{m+1})$, only depend on the local inducing states due to the Markov property.

Site-Based Inference II

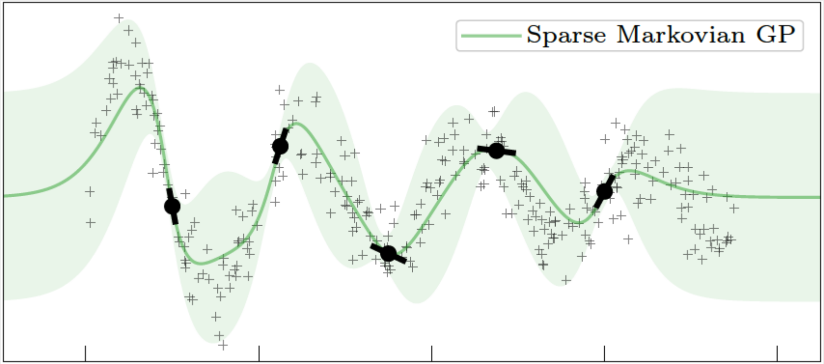
All inference methods amount to different ways of updating $t_m(\cdot, \cdot)$.

- Natural gradient variational inference (VI)
- Power expectation propagation (PEP)
- Classical Kalman Smoothers

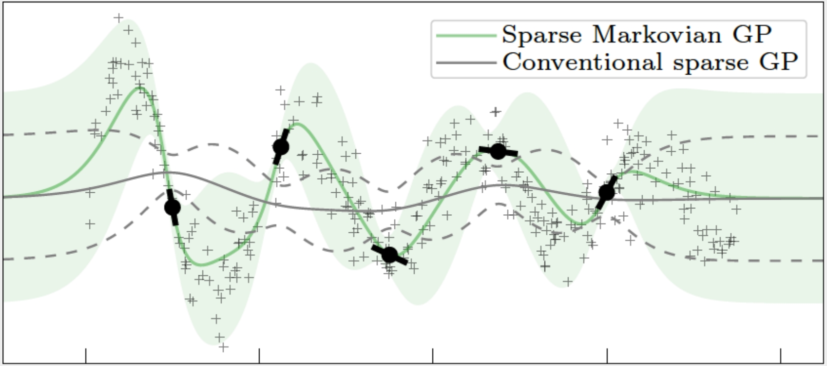
Site-based inference unifies these methods, but also has computational and stability benefits over the previous VI approach.



⟶ Forward FILTERING pass



⟵ Backward SMOOTHING pass



⟵ Backward SMOOTHING pass

Thanks for listening

william.wilkinson@aalto.fi

`https://github.com/AaltoML/Newt`