Wilkinson, William; Chang, Paul; Riis Andersen, Michael; Solin, Arno

**State Space Expectation Propagation: Efficient Inference Schemes for Temporal Gaussian Processes**

*Published in:*  
Proceedings of the 37th International Conference on Machine Learning

Published: 13/07/2020

*Document Version*  
Publisher's PDF, also known as Version of record

*Please cite the original version:*  
State Space Expectation Propagation: Efficient Inference Schemes for Temporal Gaussian Processes

William J. Wilkinson 1  Paul E. Chang 1  Michael Riis Andersen 2  Arno Solin 1

Abstract

We formulate approximate Bayesian inference in non-conjugate temporal and spatio-temporal Gaussian process models as a simple parameter update rule applied during Kalman smoothing. This viewpoint encompasses most inference schemes, including expectation propagation (EP), the classical (Extended, Unscented, etc.) Kalman smoothers, and variational inference. We provide a unifying perspective on these algorithms, showing how replacing the power EP moment matching step with linearisation recovers the classical smoothers. EP provides some benefits over the traditional methods via introduction of the so-called cavity distribution, and we combine these benefits with the computational efficiency of linearisation, providing extensive empirical analysis demonstrating the efficacy of various algorithms under this unifying framework. We provide a fast implementation of all methods in JAX.

1. Introduction

Gaussian processes (GPs, Rasmussen & Williams, 2006) are a nonlinear probabilistic modelling tool that combine well calibrated uncertainty estimates with the ability to encode prior information, and as such they are an increasingly effective method for many difficult machine learning tasks. The well known limitations of GPs are (i) their cubic scaling in the number of data, and (ii) their intractability when the observation model is non-Gaussian.

For (i), a wide variety of methods have been proposed (e.g. Hensman et al., 2013; Salimbeni & Deisenroth, 2017; Wang et al., 2019), with perhaps the most common being the sparse-GP approach (Quiñonero-Candela & Rasmussen, 2005) which summarises the GP posterior through a subset of ‘inducing’ points. However, when the data exhibits a natural ordering—as in temporal or spatio-temporal tasks—many GP priors can be rewritten in closed-form in terms of stochastic differential equations (SDEs, Särkkä & Solin, 2019), allowing for linear-time exact inference via Kalman filtering (Hartikainen & Särkkä, 2010; Reece & Roberts, 2010). This link is beneficial in scenarios such as climate modelling, or audio signal analysis, which exhibit both high and low-frequency behaviour. A sparse-GP analogy to the Shannon–Nyquist theorem (Tobar, 2019) tells us that audio signals, for example, necessarily require tens of thousands of inducing points per second of data, rendering sparse ap-
projections infeasible for all but the shortest of time series. This strongly motivates our reformulation of temporal GPs as SDEs for efficient inference.

For limitations (ii), a wide variety of approximative inference methods have been considered, with the current gold-standard being various sampling schemes (see Gelman et al., 2013, for an overview), variational methods (Opper & Archambeau, 2009; Titsias, 2009; Wainwright & Jordan, 2008), and expectation propagation (EP, Bui et al., 2017; Jylänki et al., 2011; Kuss & Rasmussen, 2006; Minka, 2001). Despite Minka’s original work having its foundations in filtering and smoothing, all the special characteristics of temporal models have not been thoroughly leveraged in the machine learning community. We extend recent work on approximate inference under the state space paradigm, and provide a framework that unifies EP and traditional methods such as the Extended (EKF, Bar-Shalom et al., 2001) and Unscented Kalman filters (UKF, Julier et al., 1995; 2000). Our framework provides ways to trade off accuracy and computation, and we show that an iterated version of the EKF with EP-style updates can be efficient and easy to implement, whilst providing good performance in cases where the likelihood model is not locally highly nonlinear. For completeness, we also formulate variational inference in the same setting.

We show that such tools are not limited to one-dimensional-input models; instead they only require us to treat a single dimension of the data sequentially (regardless of whether it is actually ordered, or represents time). We apply our methods to multi-dimensional problems such as 2D classification (see Fig. 1) and 2D log-Gaussian Cox processes.

Our main contributions are: (i) We formulate EP as a Kalman smoother, showing how it unifies many classical smoothing methods, providing an efficient framework for inference in temporal GPs. (ii) We show how to rewrite common machine learning tasks (likelihoods) into canonical state space form, and provide extensive analysis demonstrating performance in many modelling scenarios. (iii) We show how the state space framework can be extended beyond the one-dimensional case, applying it to multidimensional classification and regression tasks, where we still enjoy linear-time inference over the sequential dimension. (iv) We provide fast JAX code for inference and learning with all the methods described in this paper, available at https://github.com/AaltoML/kalman-jax.

2. Background

Gaussian processes (GPs, Rasmussen & Williams, 2006) form a non-parametric family of probability distributions on function spaces, and are completely characterized by a mean function \( \mu(t) : \mathbb{R} \to \mathbb{R} \) and a covariance function \( \kappa(t, t') : \mathbb{R} \times \mathbb{R} \to \mathbb{R} \). Let \( \{(t_k, y_k)\}_{k=1}^{n} \) denote a set of \( n \) input–output pairs for a time series (we first consider the 1D input case), then GP models typically take the form

\[
 f(t) \sim \mathcal{GP}(\mu(t), \kappa(t, t')) , \quad y \sim \prod_{k=1}^{n} p(y_k \mid f(t_k)),
\]

which defines the prior for the latent function \( f : \mathbb{R} \to \mathbb{R} \) and the observation model for \( y_k \). For Gaussian observation models, the posterior distribution \( p(f \mid y) \) is also Gaussian and can be obtained analytically, but non-Gaussian likelihoods render the posterior intractable and approximate inference methods must be applied.

2.1. State Space Models for Gaussian Processes

In signal processing, the canonical (discrete-time) state space model formulation is (e.g., Bar-Shalom et al., 2001):

\[
 x_k = g(x_{k-1}, q_k) ,
\]

\[
 y_k = h(x_k, \sigma_k) ,
\]

for time instances \( t_k \), where \( x_k \in \mathbb{R}^s \) is the discrete-time state sequence, \( y_k \in \mathbb{R}^d \) is a measurement sequence, \( q_k \sim \mathcal{N}(0, Q_k) \) is the process noise, and \( \sigma_k \sim \mathcal{N}(0, \Sigma_k) \) is Gaussian measurement noise. The model dynamics (prior) are defined by the nonlinear mapping \( g(\cdot, \cdot) \), while the observation/measurement model (likelihood) is given in terms of the mapping \( h(\cdot, \cdot) \). We restrict the model dynamics \( g(\cdot, \cdot) \) to be linear-Gaussian—defining an \( s \)-dimensional Gaussian process. The dynamical model Eq. (2a) becomes

\[
 x_k = A_k x_{k-1} + q_k , \quad q_k \sim \mathcal{N}(0, Q_k) ,
\]

which is characterized by the transition matrix \( A_k \) and process noise covariance \( Q_k \). Whilst we restrict our interest to latent Gaussian dynamics, the inference methods presented later apply to more general nonlinear state estimation settings, where the prior is not necessarily a GP (see Sec. 5).

The motivation for linking the machine learning GP formalism with state space models comes from the special structure in temporal or spatio-temporal problems, where the data points have a natural ordering with respect to the temporal dimension. If the GP prior in Eq. (1) admits a Markovian structure, the model can be rewritten in the form of Eq. (3). We leverage the link between the kernel and state space forms of GPs (Särkkä & Solin, 2019; Särkkä et al., 2013), which comes through linear time-invariant SDEs:

\[
 \dot{x}(t) = F x(t) + L w(t) , \quad \text{such that } f(t) = H x(t) ,
\]

where \( w(t) \) is a white noise process, and \( F \in \mathbb{R}^{s \times s} \), \( L \in \mathbb{R}^{s \times v} \), \( H \in \mathbb{R}^{m \times s} \) are the feedback, noise effect, and measurement matrices, respectively. Many widely used covariance functions admit this form exactly or approximately (e.g., the Matérn class, polynomial, noise, constant, squared-exponential, rational quadratic, periodic, and sums/products.
thereof). Särkkä & Solin (2019) discuss methods for constructing the required matrices for many GP models. Key to this formulation is that linear time-invariant SDEs are guaranteed to have a closed-form discrete-time solution in the form of a linear Gaussian state space model as in Eq. (3). We leverage this link in order to apply sequential inference schemes to temporal and spatio-temporal GP models.

2.2. Extended and Unscented State Estimation

Many nonlinear variants of the Kalman filter have been developed to deal with the measurement model in Eq. (2b) (see Särkkä, 2013, for an overview). The most widely known are the EKF (e.g. Bar-Shalom et al., 2001) and UKF (Julier et al., 1995). The EKF linearises \( h(x_k, \sigma_k) \) via a first-order Taylor series expansion, which in turn results in linear Gaussian approximations to all the required Kalman update equations. We discuss the approach in detail in Sec. 3.2.

The UKF is a member of a wider class of Gaussian filtering methods (Ito & Xiong, 2000), which approximate the Kalman update equations via statistical linearisation rather than a Taylor expansion. Statistical linearisation is generally intractable, involving expectations that must be computed numerically (shown in App. B). Choosing the Unscented transform as the numerical integration method results in the UKF, but other sigma-point methods can also be used (see, e.g., Ito & Xiong, 2000; Kokkala et al., 2016; Šimandl & Duník, 2009; Wu et al., 2005; 2006)—e.g. using Gauss–Hermite cubature gives the Gauss–Hermite Kalman filter.

2.3. Expectation Propagation

Expectation propagation (EP) is a general framework for approximating probability distributions proposed by Minka (2001). EP and its extension Power-EP (PEP, Minka, 2004) have been extensively studied for Gaussian process models and shown to provide state-of-the-art results (Bui et al., 2017; Jylänki et al., 2011; Kuss & Rasmussen, 2006). EP approximates the target distribution \( p(f | y) \) with an approximation \( q(f) \) that factorises in the same way as the target,

\[
p(f | y) \propto \prod_{k=1}^n p(y_k | f_k)p(f) \approx q(f) \propto \prod_{k=1}^n q_{\text{site}}^k(f_k)p(f) \tag{5}
\]

The likelihood approximations \( q_{\text{site}}^k(f_k) \approx p(y_k | f_k) \) are usually referred to as sites. For GP models, the sites are chosen to be Gaussians and hence the global approximation \( q(f) \) is also Gaussian. The sites are updated in an iterative fashion by minimizing local Kullback–Leibler divergences between the so-called tilted distributions, \( \hat{q}_k(f_k) = \frac{1}{q_k^\text{site}}p(y_k | f_k)q_k^\text{cav}(f_k) \), and its approximation using the site,

\[
q_{\text{site}}^k = \arg \min_{q_k^\text{cav}} \text{KL} \left[ \hat{q}_k(f_k) || q_k^\text{site}(f_k)q_k^\text{cav}(f_k) \right] \tag{6}
\]

where \( q_k^\text{cav}(f_k) \) is the cavity distribution: \( q_k^\text{cav}(f_k) \propto q(f_k)/q_k^\text{site}(f_k) \). The KL-divergence in Eq. (6) is minimized using moment matching (Minka, 2001), i.e. \( q_k \) is chosen such that the approximation \( q_{\text{site}}^k q_k^\text{cav} \) matches the first two moments of the tilted distribution \( \hat{p}_k \). This process is iterated for all sites until convergence. Power EP is a generalization of EP, where the KL-divergence is generalized to the \( \alpha \)-divergence (Minka, 2005). Minka (2004) showed that PEP can be implemented using the EP algorithm, by raising the site terms in the tilted distributions to a power of \( \alpha \).

3. Methods

We consider non-conjugate (i.e. non-Gaussian likelihood) Gaussian process models with input \( t \), i.e. time, which have a dual kernel (left) and discrete state space (right) form for the prior (Särkkä et al., 2013),

\[
f(t) \sim GP(\mu(t), K_0(t, t')) \quad x_k = A_{\theta, k}x_{k-1} + q_k, \tag{7}
\]

where \( f(t) = (f^{(1)}(t), \ldots, f^{(m)}(t)) \in \mathbb{R}^m \) are GPs, \( x_k = (x_k^{(1)}, \ldots, x_k^{(m)}) \in \mathbb{R}^m \) is the latent state vector containing the GP dynamics, and \( y_k \in \mathbb{R}^d \) are observations. Each \( x_k^{(i)} \) contains the state dynamics for one GP. Using notation \( f_k = f(t_k) \), we define a time-varying linear map \( \mathbf{H}_k \in \mathbb{R}^{m \times s} \) from state space to function space, such that \( f_k = \mathbf{H}_k x_k \) (the time-varying mapping allows us to naturally incorporate spacial inducing points when considering multidimensional input models, see Sec. 3.6). The likelihood (left) / state observation model (right) are

\[
y_k \sim p(y_k | f_k) \quad \text{vs.} \quad y_k = h(f_k, \sigma_k). \tag{8}
\]

Measurement model \( h(f_k, \sigma_k) \) is a (nonlinear) function of \( f_k \) and observation noise \( \sigma_k \sim N(0, \Sigma_k) \), and can generally be derived for continuous likelihoods or approximated for discrete ones by letting \( h(f_k, \sigma_k) \approx E[y_k | f_k] + \varepsilon_k \), \( \varepsilon_k \sim N(0, \text{Cov}[y_k | f_k]) \). See Sec. 4 and App. I for derivations of some common models. We aim to calculate the posterior over the states, \( p(x_k | y_1, \ldots, y_n) \), known as the smoothing solution, which can be obtained via application of a Gaussian filter (to obtain the filtering solution \( p(x_k | y_1, \ldots, y_{k-1}) \)) followed by a Gaussian smoother. If \( h(\cdot, \cdot) \) is linear, i.e. \( p(y_k | f_k) \) is Gaussian, then the Kalman filter and Rauch–Tung–Striebel (RTS, Rauch et al., 1965) smoother return the closed-form solution.

3.1. Power EP as a Gaussian Smoother

Our inference methods approximate the filtering distributions with Gaussians, \( p(x_k | y_{1:k}) \approx N(x_k | \mathbf{m}_{k|k}^{\text{filt}}, \mathbf{P}_{k|k}^{\text{filt}}) \). The prediction step remains the same as in the standard Kalman filter: \( \mathbf{m}_{k|k}^{\text{pred}} = A_{\theta, k}\mathbf{m}_{k-1}^{\text{filt}} \), and \( \mathbf{P}_{k|k}^{\text{pred}} = A_{\theta, k}\mathbf{P}_{k-1}^{\text{filt}}A_{\theta, k}^\top + Q_{\theta, k} \). The resulting distribution provides a means by which to calculate the EP cavity,
Comparison between EP, VI and iterated linearisation (EKS, GHKS). When measurement function \( h \) is approximately linear in the region of the prior (or the cavity / posterior in the full algorithm), (left), linearisation \( h \) provides a similar result to EP / VI. When \( h \) is highly nonlinear (right), the posterior approximations have different properties. 20-point Gauss–Hermite quadrature used for all methods except EKS. All methods are iterated 10 times except EP which does not require iteration for a single data point.

\[
g^\text{cav}(f_k) = N(f_k \mid \mu_k^\text{cav}, \Sigma_k^\text{cav}), \text{ on the first forward pass:} \\
\mu_k^\text{cav} = H_k m_k^\text{pred}, \quad \Sigma_k^\text{cav} = H_k P_k^\text{pred} H_k^\top. \tag{9}
\]

In this sense, we can view the first pass of the Kalman filter as an effective way to initialise the EP parameters. To account for the non-Gaussian likelihood in the update step we follow Nickisch et al. (2018), introducing an intermediary step in which the parameters of the sites, \( \hat{\theta}_k^\text{site}(f_k) = N(f_k \mid \mu_k^\text{site}, \Sigma_k^\text{site}) \), are set via moment matching and stored before continuing with the Kalman updates.

This PEP formulation, with power \( \alpha \), makes use of the fact that the required moments can be calculated via the derivatives of the log-normaliser, \( L_k \), of the tilted distribution (see Seeger, 2005). Letting \( \nabla L_k \in \mathbb{R}^m \) and \( \nabla^2 L_k \in \mathbb{R}^{m \times m} \) be the Jacobian and Hessian of \( L_k \) w.r.t. \( \mu_k^\text{cav} \) respectively, this gives the following site update rule,

**Power expectation propagation**

\[
L_k = \log \mathbb{E}_{N(f_k \mid \mu_k^\text{cav}, \Sigma_k^\text{cav})}[p^\alpha(y_k \mid f_k)], \\
\Sigma_k^\text{site} = -\alpha \left( \Sigma_k^\text{cav} + \left( \nabla^2 L_k \right)^{-1} \right), \tag{10} \\
\mu_k^\text{site} = \mu_k^\text{cav} - \left( \nabla^2 L_k \right)^{-1} \nabla L_k.
\]

After the mean and covariance of our new likelihood approximation have been calculated, we proceed with a modified set of linear Kalman updates,

\[
S_k = H_k P_k^\text{pred} H_k^\top + \Sigma_k^\text{site}, \\
K_k = P_k^\text{pred} H_k^\top S_k^{-1}, \\
m_k^\text{filt} = m_k^\text{pred} + K_k (\mu_k^\text{site} - H_k m_k^\text{pred}), \\
P_k^\text{filt} = P_k^\text{pred} - K_k S_k K_k^\top. \tag{11}
\]

As in Wilkinson et al. (2019), we augment the RTS smoother with another moment matching step where the cavity distribution is calculated by removing (a fraction \( \alpha \) of) the local site from the marginal smoothing distribution, i.e. the posterior, \( p(x_k \mid y_{1:n}) = N(x_k \mid m_k^\text{post}, P_k^\text{post}) \),

\[
\Sigma_k = \left[ \left( H_k P_k^\text{post} H_k^\top \right)^{-1} - \alpha \left( \Sigma_k^\text{site} \right)^{-1} \right]^{-1}, \\
\mu_k^\text{cav} = \Sigma_k^\text{cav} \left[ \left( H_k P_k^\text{post} H_k^\top \right)^{-1} H_k m_k^\text{post} - \alpha \left( \Sigma_k^\text{site} \right)^{-1} \mu_k^\text{site} \right]. \tag{12}
\]

Moment matching is again performed via Eq. (10) using this new cavity. The site parameters, \( \mu_k^\text{site}, \Sigma_k^\text{site} \), are stored to be used on the next forward (filtering) pass. App. F discusses methods for avoiding numerical issues that could occur due to the subtraction of covariance matrices in Eq. (12). Algorithm 1 summarises the full learning algorithm, and App. G describes how the marginal likelihood, \( p(y \mid \theta) \), is computed to enable hyperparameter learning.

3.2. Unifying PowerEP and Extended Kalman Filtering

In the above inference scheme, a computational saving can be gained by noticing that when \( h(\cdot, \cdot) \) is linear, \( L \) can be calculated in closed form. This fact has been exploited previously to aid inference in GP dynamical systems (Deisenroth & Mohamed, 2012). Fig. 2 demonstrates that such an approximation can be accurate when \( h(\cdot, \cdot) \) is locally linear, or when the cavity variance is small. Using a first-order Taylor series expansion about the mean \( \mu_k^\text{cav} \), we obtain

\[
h(f_k, \sigma_k) \approx J_{\sigma_k}(f_k - \mu_k^\text{cav}) + h(\mu_k^\text{cav}, 0) + J_{\sigma_k}, \sigma_k, \tag{13}
\]

a linear function of \( f_k \) and \( \sigma_k \sim N(0, \Sigma_k) \), such that \( p(y_k \mid f_k) \approx N(y_k \mid h(f_k), J_{\sigma_k} \Sigma_k J_{\sigma_k}^\top) \), where \( \hat{h}(f_k) = J_{\sigma_k}(f_k - \mu_k^\text{cav}) + h(\mu_k^\text{cav}, 0) \). Here \( J_{\sigma_k} = J_f |_{\mu_k^\text{cav}, 0} \in \mathbb{R}^{d \times m} \) and \( J_{\sigma_k} = J_{\sigma_k} |_{\mu_k^\text{cav}, 0} \in \mathbb{R}^{d \times d} \) are the Jacobians of \( h(\cdot, \cdot) \) w.r.t. \( f_k \) and \( \sigma_k \) evaluated at the mean, respectively.

In order to frame approximate inference in the same setting as EP, we seek the site update rule implied by this linearisation. If \( J_f \) is invertible, then writing down such a rule would be trivial, but since this is not generally the case we instead use the EP moment matching steps, Eq. (10), which give,

\[
L_k = \log \mathbb{E}_{N(f_k \mid \mu_k^\text{cav}, \Sigma_k^\text{cav})}[N^\alpha(y_k \mid \hat{h}(f_k), J_{\sigma_k} \Sigma_k J_{\sigma_k}^\top)] \\
= c + \log N(y_k \mid h(\mu_k^\text{cav}, 0), \alpha^{-1} \hat{\Sigma}_k), \tag{14}
\]
where $\Sigma_k = J\sigma_k, \Sigma_k J^T + \alpha J\kappa_k \Sigma_k^{\text{cav}} J^T$. Taking the derivatives of this log-Gaussian w.r.t. the cavity mean, we get

$$\nabla L_k = \frac{\partial L_k}{\partial \mu_k^{\text{cav}}} = \alpha J^T \Sigma_k^{-1} v_k,$$

$$\nabla^2 L_k = \frac{\partial^2 L_k}{\partial \mu_k^{\text{cav}} \partial (\mu_k^{\text{cav}})} = -\alpha J^T \Sigma_k^{-1} J_k,$$  \hfill (15)

where $v_k = y_k - h(\mu_k^{\text{cav}}, 0)$. It is important to note that we have assumed the derivative of $\Sigma_k$ to be zero, even though it depends on $\mu_k^{\text{cav}}$. This assumption is crucial in ensuring that the updates are consistent, since it reflects the knowledge that the model is now linear (see Deisenroth & Mohamed (2012) for detailed discussion). Now we update the site in closed form (App. C gives the derivation).

### Extended expectation propagation

$$\Sigma_k^{\text{site}} = \left( J_k^T (J_k, \Sigma_k, J_k^T)^{-1} J_k \right)^{-1},$$

$$\mu_k^{\text{site}} = \mu_k^{\text{cav}} + (\Sigma_k^{\text{site}} + \alpha \Sigma_k^{\text{cav}}) J_k \Sigma_k^{-1} v_k.$$ \hfill (16)

The result when we use Eq. (16) (with $\alpha = 1$) to modify the filter updates, Eq. (11), is exactly the EKF (see App. D for the proof). Additionally, since these updates are now available in closed form, taking the limit $\alpha \to 0$ is now possible and avoids the matrix subtractions and inversions in Eq. (12), which can be costly and unstable. This is not possible prior to linearisation because the intractable integrals also depend on $\alpha$. App. H describes our full algorithm.

### 3.3. Power EP and the Unscented/GH Kalman Filters

We now consider the relationship between EP and general Gaussian filters, which use the likelihood approximation

$$p(y_k | f_k) \approx N(y_k | \mu_k, C_k^{\text{cav}})^{-1}(f_k - \mu_k^{\text{cav}}),$$

$$S_k = C_k^{\text{cav}}^{-1} C_k,$$ \hfill (17)

where $\mu_k, S_k$ and $C_k$ are the Kalman mean, innovation and cross-covariance terms respectively, given in App. B. Eq. (17) amounts to statistical linear regression (Särkkä, 2013) of $h(f_k, \sigma_k)$. Letting $\mu_k^{\text{cav}} = H_k \mu_k^{\text{pred}}, \Sigma_k^{\text{cav}} = H_k P_k^{\text{pred}} H_k^T$ and using the Unscented transform / Gauss–Hermite to approximate $\mu_k, S_k$ and $C_k$ results in the UKF / GHKF. This approximation has a similar form to the EKF (which uses analytical linearisation, see Fig. 2 for comparison), and as in Sec. 3.2 we can insert the Gaussian likelihood approximation into Eq. (10) to derive an iterated algorithm that matches the Gaussian filters on the first forward pass, but then refines the linearisation using EP style updates. This provides the following site update rule (see App. E):

**Algorithm 1 Sequential inference & learning algorithm**

**Input:** $\{t_k, y_k\}_{k=1}^{n}$, $\theta_0$, $\alpha$, and learning rate $\rho$,

update_rule $\leftarrow$ Eq. (10), Eq. (16), Eq. (18) or Eq. (20)

for $i = 1$ to num_iters do

build model, Eq. (7), with $\theta_{i-1}, m_0^{\text{filt}}, P_0^{\text{filt}} \leftarrow 0, P_{\infty}$

for $k = 1$ to $n$ do

$m_k^{\text{site}}, P_k^{\text{site}} \leftarrow \text{predict}(m_k^{\text{filt}}, P_k^{\text{filt}})$

if $i = 1$ then

 initialise $\mu_k^{\text{site}}, \Sigma_k^{\text{site}}$ via update_rule using

$\alpha = 1$ and $m_k^{\text{site}}, P_k^{\text{site}}$ as cavity / posterior end if

$m_k^{\text{site}}, P_k^{\text{site}} \leftarrow \text{update}(m_k^{\text{site}}, P_k^{\text{site}}, \mu_k^{\text{site}}, \Sigma_k^{\text{site}})$

$e_k = -\log p(y_k | y_{1:k-1}, \theta_{i-1})$ see App. G

end for

for $k = n - 1 \to 1$ do

$m_k^{\text{post}}, P_k^{\text{post}} \leftarrow \text{smooth}(m_k^{\text{post}}, P_k^{\text{post}}, m_{k+1}^{\text{filt}}, P_{k+1}^{\text{filt}})$ update $\mu_k^{\text{site}}, \Sigma_k^{\text{site}}$ via update_rule

end for

$\theta_i = \theta_{i-1} + \rho \nabla \theta \sum_k e_k$ update hyper.

end for

Return: posterior mean and covariance: $m^{\text{post}}, P^{\text{post}}$

### Statistically linearised expectation propagation

$$\Sigma_k^{\text{site}} = -\alpha \Sigma_k^{\text{cav}} + \left( \Omega_k^T \Sigma_k^{\text{cav}} \Omega_k \right)^{-1},$$

$$\mu_k^{\text{site}} = \mu_k^{\text{cav}} + \left( \Omega_k^T \Sigma_k^{\text{cav}} \Omega_k \right)^{-1} \Omega_k^T \Sigma_k^{\text{cav}}^{-1} v_k.$$ \hfill (18)

where $v_k = y_k - \mu_k, \Sigma_k = S_k + (\alpha - 1) C_k^T (\Sigma_k^{\text{cav}})^{-1} C_k, \Omega_k = \frac{\partial \mu_k^{\text{cav}}}{\partial \mu_k^{\text{cav}}} = \int h(f_k, \sigma_k) (\Sigma_k^{\text{cav}})^{-1} (f_k - \mu_k^{\text{cav}}) \times N(f_k | \mu_k^{\text{cav}}, \Sigma_k^{\text{cav}}) N(\sigma_k | 0, \Sigma_k) \text{d} f_k \text{d} \sigma_k.$ \hfill (19)

### 3.4. Nonlinear Kalman Smoothers

Iterated versions of nonlinear filter-smoothers have been developed to address the fact that the forward prediction, $N(x_k^{\text{pred}} | m_k^{\text{pred}}, P_k^{\text{pred}})$, may not be the optimal distribution about which to perform linearisation. It is argued that the posterior, $N(x_k^{\text{post}} | m_k^{\text{post}}, P_k^{\text{post}})$, obtained via smoothing, provides a better estimate of the region in which the likelihood affects the posterior (García-Fernández et al., 2015).

These iterated smoothers (Bell, 1994) can be seen as special cases of the algorithms described in Sec. 3.2 and Sec. 3.3, where the posterior is used to perform the linearisation in place of the cavity, i.e. $\alpha = 0$. The classical smoothers seek a linear approximation to the likelihood $p(y_k | f_k) \approx N(y_k | B_k f_k + b_k, E_k)$ via a Taylor expansion, Eq. (13), or SLR, Eq. (17), and then store parameters $B_k, b_k, E_k$ to be used during the next forward pass. Instead, we
use the current posterior approximation to compute the site parameters via Eq. (16) or Eq. (18), which differs slightly from the standard presentation of these algorithms. We argue that framing the Kalman smoothers as site update rules is beneficial in that it allows for direct comparison with EP, but also that introduction of the cavity is beneficial. The cavity may be a better distribution about which to linearise than the posterior, since it does not already include the effect of the local data. However, Table 1 shows that setting $\alpha = 0$ typically provides the best performance.

### 3.5. Variational Inference with Natural Gradients

Variational inference (VI) is an alternative to EP, often favoured due to its convergence guarantees and ease of implementation. If VI is formulated such that the variational parameters of the approximate posterior $q(\mathbf{f})$ are the likelihood (i.e. site) mean and covariance, as in Eq. (5), then it can also be framed as a site update rule during Kalman smoothing (Chang et al., 2020). This parametrisation is in fact the optimal one, as discussed in Opper & Archambeau (2009), but is often avoided because the resulting optimisation problem is non-convex (instead it is common to declare a variational distribution over the full posterior, $q(\mathbf{f}) = \mathcal{N}(\mathbf{m}, \mathbf{K})$, and optimise $\mathbf{m}, \mathbf{K}$ with respect to the evidence lower bound. Adam et al. (2020) show how to perform natural gradient VI under this parametrisation).

We present here the VI site update rule, based on conjugate-computation variational inference (CVI, Khan & Lin, 2017), in order to show their similarity to EP, and to enable direct comparison between the algorithms. CVI sidesteps the issues with the optimal parametrisation by showing that natural gradient VI can be performed via local site parameter updates that avoid directly differentiating the evidence lower bound. The updates can be written,

$$
\tilde{L}_k = \mathbb{E}_{q(\mathbf{f}_k | \mathbf{\mu}_k, \mathbf{\Sigma}_k)} \left[ \log p(\mathbf{y}_k | \mathbf{f}_k) \right],
\Sigma^\text{site} = -\left( \nabla^2 \tilde{L}_k \right)^{-1},
\mu^\text{site} = \mathbf{\mu}_k \text{post} - \left( \nabla^2 \tilde{L}_k \right)^{-1} \nabla \tilde{L}_k,
$$

(20)

where $\nabla \tilde{L}_k \in \mathbb{R}^m$ and $\nabla^2 \tilde{L}_k \in \mathbb{R}^{m \times m}$ are the Jacobian and Hessian of $\tilde{L}_k$ w.r.t. $\mu^\text{post}_k$ respectively.

### 3.6. Spatio-Temporal Filtering and Smoothing

The methodology presented in the previous sections for temporal models directly lends itself to generalisations in spatio-temporal modelling. We consider a GP prior which is separable in the sequential (temporal) input $t$ and the remaining (spatial) input(s) $r$: $\kappa(\mathbf{r}, t; \mathbf{r}', t') = \kappa_t(\mathbf{r}, \mathbf{r}') \kappa_s(t, t')$.

Following Särkkä et al. (2013), we extend the state $x(t)$ of the system via $m$ coupled temporal processes. These processes are associated with inducing points $\{\mathbf{r}_{u,j}\}_{j=1}^m$ in the spatial domain. The measurement model matrix now projects the latent state at time $t_k$ from the inducing processes in the state to function space by,

$$
H_k = [\mathbf{K}_{f_k,u} \mathbf{K}_{u,u}^{-1}] \otimes H_t,
$$

(21)

with Gram matrices $\mathbf{K}_{f_k,u} = \kappa_t(\mathbf{r}_k, \mathbf{r}_{u,j})$ and $\mathbf{K}_{u,u} = \kappa_t(\mathbf{r}_{u,j}, \mathbf{r}_{u,j})$ for $j = 1, 2, \ldots, m$, where $H_t$ is the measurement model matrix for the GP prior. If the data lies on a fixed set of spatial points $\{\mathbf{r}_j\}_{j=1}^m$ (an irregular grid of $m$ points), the above expression simplifies to $H_k = \mathbf{I}_m \otimes H_t$ and the models becomes exact (see, Hartikainen, 2013; Solin, 2016, for further details and discussion, also covering non-separable models).

### 3.7. Fast Implementation Using JAX

Sequential inference in GPs is extremely efficient, however optimising the model hyperparameters involves differentiating functions with large loops. When using automatic differentiation this typically results in a massive computational graph with large compilation overheads, memory usage and runtime. Previous approaches have avoided this issue either by using finite differences (Nickisch et al., 2018), which are slow when the number of parameters is large, or by reformulating the model to exploit linear algebra methods applicable to sparse precision matrices (Durrande et al., 2019).

We utilise the following features of the differential programming Python framework, JAX (Bradbury et al., 2018): (i) we avoid ‘unrolling’ of for-loops, i.e. instead of building a large graph of repeated operations, a smaller graph is recursively called, reducing the compilation overhead and memory, (ii) we just-in-time (JIT) compile the loops, to avoid the cost of graph retraceing, (iii) we use accelerated linear algebra (XLA) to speed up the underlying filtering/smoothing operations. Combined, these features result in an extremely fast implementation, and based on this we provide a fully featured temporal GP framework with all models and inference methods, available at https://github.com/AaltoML/kalman-jax.

### 4. Empirical Analysis

Table 1 examines the performance of 17 methods from our GP framework on 7 benchmark tasks of varying data size and model complexity. Blanks (—) in the table represent scenarios where the method does not scale practically to the size of the task. First we demonstrate that state space approximate inference schemes are competitive with two state-of-the-art baseline methods on three small data sets.
We optimise the model hyperparameters by maximising the marginal likelihood $p(\theta)$ separately for each method (see App. G), hence the results in Table 1 are affected by both training and inference, demonstrating their practical applicability. The baseline methods scale as $O(n^3)$, while all the sequential schemes scale as $O(ns^3)$.

### Results

Table 1 confirms it is possible to achieve state-of-the-art performance with sequential inference. The log-Gaussian Cox process and classification experiments return consistent results across all methods. However, Audio and Rainforest involve multidimensional sites, making them difficult tasks. In such cases, EEP performs well because it is the only method capable of maintaining full site cross-covariance terms whilst remaining stable. Statistical linearisation suffers less from a reduction of cubature points than EP moment matching or VI updates, as shown by the performance of UEP on Audio. EP generally performed well, but EEP matches its performance sometimes whilst being the only method applicable to the Rainforest task. In other cases, cubature methods outperform linearisation, particularly on difficult tasks. In such cases, EEP performs well because it is the only method capable of maintaining full site cross-covariance terms whilst remaining stable. Statistical linearisation suffers less from a reduction of cubature points than EP moment matching or VI updates, as shown by the performance of UEP on Audio. EP generally performed well, but EEP matches its performance sometimes whilst being the only method applicable to the Rainforest task. In other cases, cubature methods outperform linearisation, particularly on the Motorcycle task.

### Motorcycle (heteroscedastic noise)

The motorcycle crash data set (Silverman, 1985) contains 131 non-uniformly spaced measurements from an accelerometer placed on a motorcycle helmet during impact, over a period of 60 ms. It is a challenging benchmark (e.g., Tolvanen et al., 2014), due

### Table 1

Normalised negative log predictive density (NLPD) results with 10-fold cross-validation. Smaller is better. Blank entries (—) represent scenarios where the method does not scale to the size of the task. EEP, UEP, and GHEP are the iterated smoothers with linearisation. EP(U) and EP(GH) are state space EP, where the intractable moment matching is performed via the Unscented transform or Gauss–Hermite, respectively. Linearisation performs poorly on the heteroscedastic noise task, however EEP performs well on the audio task since it is the only method capable of maintaining full site cross-covariance terms without compromising stability. The state space methods are able to match the performance of the non-sequential (batch) EP and VGP baselines.

<table>
<thead>
<tr>
<th>DATA POINTS</th>
<th>MOTORCYCLE</th>
<th>COAL</th>
<th>BANANA</th>
<th>BINARY</th>
<th>AUDIO</th>
<th>AIRLINE</th>
<th>RAINFOREST</th>
</tr>
</thead>
<tbody>
<tr>
<td># INPUT DIMENSION</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>LIKELIHOOD</td>
<td>HETEROSCEDASTIC</td>
<td>POISSON</td>
<td>BERNOUlli</td>
<td>BERNOUlli</td>
<td>PRODUCT</td>
<td>POISSON</td>
<td>POISSON</td>
</tr>
<tr>
<td>EEP ($\alpha = 1$)</td>
<td>0.855±0.25</td>
<td>0.922±0.11</td>
<td>0.228±0.07</td>
<td>0.536±0.01</td>
<td>-0.433±0.04</td>
<td>0.142±0.01</td>
<td>0.325±0.01</td>
</tr>
<tr>
<td>EEP ($\alpha = 0.5$)</td>
<td>0.855±0.25</td>
<td>0.922±0.11</td>
<td>0.228±0.07</td>
<td>0.536±0.01</td>
<td>-0.499±0.03</td>
<td>0.142±0.01</td>
<td>0.321±0.01</td>
</tr>
<tr>
<td>EEP ($\alpha = 0$) / EKS</td>
<td>0.855±0.25</td>
<td>0.922±0.11</td>
<td>0.229±0.07</td>
<td>0.537±0.01</td>
<td>-0.570±0.04</td>
<td>0.142±0.01</td>
<td>0.309±0.01</td>
</tr>
<tr>
<td>UEP ($\alpha = 1$)</td>
<td>0.745±0.28</td>
<td>0.922±0.11</td>
<td>0.217±0.08</td>
<td>0.536±0.01</td>
<td>-0.471±0.02</td>
<td>0.142±0.01</td>
<td>—</td>
</tr>
<tr>
<td>UEP ($\alpha = 0.5$)</td>
<td>0.745±0.28</td>
<td>0.922±0.11</td>
<td>0.217±0.08</td>
<td>0.536±0.01</td>
<td>-0.474±0.02</td>
<td>0.142±0.01</td>
<td>—</td>
</tr>
<tr>
<td>UEP ($\alpha = 0$) / UKS</td>
<td>0.745±0.28</td>
<td>0.922±0.11</td>
<td>0.217±0.08</td>
<td>0.536±0.01</td>
<td>-0.484±0.02</td>
<td>0.142±0.01</td>
<td>—</td>
</tr>
<tr>
<td>GHEP ($\alpha = 1$)</td>
<td>0.750±0.26</td>
<td>0.922±0.11</td>
<td>0.217±0.08</td>
<td>0.536±0.01</td>
<td>—</td>
<td>0.142±0.01</td>
<td>—</td>
</tr>
<tr>
<td>GHEP ($\alpha = 0.5$)</td>
<td>0.747±0.27</td>
<td>0.922±0.11</td>
<td>0.217±0.08</td>
<td>0.536±0.01</td>
<td>—</td>
<td>0.142±0.01</td>
<td>—</td>
</tr>
<tr>
<td>GHEP ($\alpha = 0$) / GHKS</td>
<td>0.746±0.26</td>
<td>0.922±0.11</td>
<td>0.217±0.08</td>
<td>0.536±0.01</td>
<td>—</td>
<td>0.142±0.01</td>
<td>—</td>
</tr>
<tr>
<td>VI (U) ($\alpha = 1$)</td>
<td>0.696±0.59</td>
<td>0.922±0.11</td>
<td>0.217±0.08</td>
<td>0.536±0.01</td>
<td>-0.321±0.15</td>
<td>0.143±0.01</td>
<td>—</td>
</tr>
<tr>
<td>VI (U) ($\alpha = 0.5$)</td>
<td>0.479±0.30</td>
<td>0.922±0.11</td>
<td>0.217±0.08</td>
<td>0.536±0.01</td>
<td>-0.327±0.20</td>
<td>0.143±0.01</td>
<td>—</td>
</tr>
<tr>
<td>VI (U) ($\alpha = 0$)</td>
<td>0.465±0.29</td>
<td>0.924±0.11</td>
<td>0.222±0.08</td>
<td>0.536±0.01</td>
<td>—</td>
<td>0.143±0.01</td>
<td>—</td>
</tr>
<tr>
<td>VI (GH)</td>
<td>0.569±0.41</td>
<td>0.922±0.11</td>
<td>0.217±0.08</td>
<td>0.536±0.01</td>
<td>—</td>
<td>0.142±0.01</td>
<td>—</td>
</tr>
<tr>
<td>VI (GH) ($\alpha = 0.5$)</td>
<td>0.531±0.38</td>
<td>0.922±0.11</td>
<td>0.217±0.08</td>
<td>0.536±0.01</td>
<td>—</td>
<td>0.142±0.01</td>
<td>—</td>
</tr>
<tr>
<td>VI (GH) ($\alpha = 0$)</td>
<td>0.444±0.32</td>
<td>0.922±0.11</td>
<td>0.217±0.08</td>
<td>0.536±0.01</td>
<td>—</td>
<td>0.142±0.01</td>
<td>—</td>
</tr>
<tr>
<td>VI (GH) ($\alpha = 0$)</td>
<td>0.444±0.31</td>
<td>0.922±0.11</td>
<td>0.217±0.08</td>
<td>0.536±0.01</td>
<td>-0.204±0.02</td>
<td>0.142±0.01</td>
<td>—</td>
</tr>
<tr>
<td>VI (GH) ($\alpha = 0$)</td>
<td>0.495±0.34</td>
<td>0.922±0.11</td>
<td>0.217±0.08</td>
<td>0.536±0.01</td>
<td>—</td>
<td>0.142±0.01</td>
<td>—</td>
</tr>
<tr>
<td>VI (GH) ($\alpha = 0$)</td>
<td>0.444±0.30</td>
<td>0.922±0.11</td>
<td>0.216±0.10</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>VI (GH) ($\alpha = 0$)</td>
<td>0.444±0.30</td>
<td>0.922±0.11</td>
<td>0.219±0.09</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>
We use a log-Gaussian Cox process, an inhomogeneous Poisson process, with likelihood for \( n \) equal time interval bins:

\[
\begin{align*}
\lambda(t) &= \exp(f(t)) \\
y_k | f_k \sim \text{Poisson}(\lambda_k), \quad \lambda_k = \exp(f(t_k)), \quad k \in 1, \ldots, n.
\end{align*}
\]

The nonlinear inter-domain function \( \phi(f) = \log(1 + e^f) \) to ensure positive noise scale. The full EP and VGP baselines were implemented and hand-tailored for this task. For VGP, we used GPflow 2 (Matthews et al., 2017) with a custom model. VI and EP (\( \alpha \approx 0 \)) performed well, however the linearisation-based methods failed to capture the time-varying noise (see App. I for discussion).

### Binary (1D classification)

As a 1D classification task, we create a long binary time series, \( n = 10,000 \), using the generating function \( y(t) = \text{sign}(\frac{12\sin(4\pi t)}{0.25\pi^2 t^2 + 1} + \sigma_t) \), with \( \sigma_t \approx N(0, 0.25^2) \). Our GP prior has a Matérn-\( 3/2 \) kernel, \( s = 4 \), and the sigmoid function \( \psi(f) = (1 + e^{-f})^{-1} \) maps \( \mathbb{R} \rightarrow [0, 1] \) (logit classification). See App. I for derivation of approximate continuous state observation model, \( h(f_k, \sigma_k) = \psi(f_k) + \sqrt{\psi(f_k)(1 - \psi(f_k))}\sigma_k \).

### Audio (product of GPs)

We apply a simplified version of the Gaussian Time-Frequency model from Wilkinson et al. (2019) to half a second of human speech, sampled at 44.1 kHz, \( n = 22,050 \). The prior consists of 3 quasi-periodic \( \kappa_{\text{exp}}(t, t') \kappa_{\text{cos}}(t, t') \) 'subband' GPs, and 3 smooth \( \kappa_{\text{Mat.}}(t, t') \) 'amplitude' GPs. The likelihood consists of a sum of the product of these processes with additive noise and a softplus mapping \( \phi(f) \) for the positive amplitudes:

\[
y_k | f_k \sim \text{N}(\sum_{i=1}^{3} f_{i,k} \phi(f_{i,k}^\text{amp}), \sigma_k^2).
\]

The nonlinear inter-domain function becomes

\[
\phi(f) = \log(1 + e^f).
\]

We use a Matérn-\( 3/2 \) GP prior with likelihood \( p(y | f) \approx \prod_{k=1}^{n} \text{Poisson}(y_k | \exp(f(t_k))) \), where \( f_k \) is the bin coordinate and \( y_k \) the number of disasters in the bin. This model reaches posterior consistency in the limit of bin width going to zero (Tokdar & Ghosh, 2007). Since linearisation requires a continuous likelihood, we approximate the discrete Poisson with a Gaussian by noticing that its first two moments are equal to the intensity \( \lambda(t) = \exp(f(t)) \), giving

\[
y_k | f_k \approx N(y_k | \lambda(t_k), \lambda(t_k)).
\]

### Airline (log-Gaussian Cox process)

The airline accidents data (Nickisch et al., 2018) consists of 1210 dates of commercial airline accidents between 1919–2017. We use a log-Gaussian Cox process with bin width of one day, leading to \( n = 35,959 \) observations. The prior has multiple components, \( \kappa(t, t') = \kappa(t, t')^{\text{Mat.}} + \kappa_{\text{year}}^{t,t'} + \kappa_{\text{week}}^{t,t'} + \kappa_{\text{per.}}^{t,t'} \) capturing a long-term trend, time-of-year variation (with decay), and day-of-week variation (with decay). The state dimension is \( s = 59 \).

### Coal (log-Gaussian Cox process)

The coal mining disaster data set (Vanhatalo et al., 2013) contains 191 explosions that killed ten or more men in Britain between 1851–1962. We use a log-Gaussian Cox process, i.e. an inhomogeneous Poisson process (approximated with a Poisson likelihood for \( n = 333 \) equal time interval bins).
action of 6 GPs ($s = 15$) in the likelihood makes this a challenging task. EEP performs best since it is capable of maintaining full site covariance terms without compromising stability. UEP outperforms EP and VI since statistical linearisation is still accurate when using few cubature points.

4.1. Spatio-Temporal Models

As presented in Sec. 3.6, the sequential inference schemes are also applicable to spatio-temporal problems. We illustrate this via two spatial problems, treating one spatial input as the sequential dimension (‘time’) and the other as ‘space’.

**Banana (2D classification)** The banana data set, $n = 400$, is a common 2D classification benchmark (e.g., Hensman et al., 2015). We use the logit likelihood with a separable space-time kernel: $\kappa(r, t; r', t') = \kappa(t)\kappa(t')^{\nu=5/2} = \kappa(r)\kappa(r')^{\nu=5/2} = \kappa(r, r')^{\nu=5/2}$. The vertical dimension is treated as space $r$ and the horizontal as the sequential (‘temporal’) dimension $t$. We use $m = 15$ inducing points in $r$ (see Sec. 3.6), visualised by lines in Fig. 4(c)–(e). The state dimension is $s = 3m = 45$. Fig. 4 shows that the EKF provides a similar solution to the VGP baseline of Hensman et al. (2015), and an even closer match is obtained by EEP (3 iterations). The forward and backward passes are visualised in Fig. 1. The method is also applicable to the larger ($n = 5400$) version of the data set (Fig. 4e).

**Rainforest (2D log-Gaussian Cox process)** We study the density of a single tree species, Trichilia tuberculata, from a 1000 m $\times$ 500 m region of a rainforest in Panama (Condit, 1998; Hubbell et al., 1999; 2005). We segment the space into 4 m$^2$ bins, giving a $500 \times 250$ grid with 125,000 data points ($n = 500$ time steps), and use a log-Gaussian Cox process (Fig. 3d). The space-time GP has a separable Matérn-$3/2$ kernel. We do not use a sparse approximation in $r$; instead we have $m = 250$ temporal processes, so $s = 2m = 500$.

**Run Times** Table 2 compares time taken for all methods to make a single training step on a MacBook Pro with 2.3 GHz Intel Core i5 and 16 GB RAM using JAX. For tasks with one-dimensional sites all methods are similar, however EEP is faster than the cubature methods for the Audio task which involves 6-dimensional sites. The gridded data of the Rainforest task requires a 250-dimensional site parameter update which is impractical for most methods. Conversely, in the Banana task data points are handled one by one, such that only one-dimensional updates are required.

5. Discussion and Conclusions

We argue that development of methods capable of naturally handling sequential data is crucial to extend the applicability of GPs beyond short time series. EP was originally inspired by, and derived from, Kalman filtering and here we make the case that a return to sequential methods is desirable for large spatio-temporal problems. We present a flexible and efficient framework for sequential learning that encompasses many state-of-the-art approximate inference schemes, whilst also illuminating the connections between modern day inference methods and traditional filtering approaches.

Our theoretical contributions confirm that using linearisation in place of EP moment matching results in iterated algorithms that exactly match the classical nonlinear Kalman filters on the first pass, and also generalise the classical smoothers by refining the linearisations via multiple passes through the data. These algorithms are fast and scale to high-dimensional spatio-temporal problems more effectively than EP and VI. The methods based on Taylor series approximations only require one evaluation of the likelihood (and its Jacobian) for each data point, as opposed to cubature methods, and these algorithms make it particularly straightforward to prototype and implement new likelihood models.

We provide a detailed examination of the different properties of all these methods on five time series and two spatial tasks, showing that the state space framework for GPs is applicable beyond one-dimensional problems. We have also highlighted the scenarios in which such methods might fail: linearisation is a poor approximation when the cavities are diffuse (high variance) and the likelihood is highly nonlinear, but cubature methods do not scale well to high dimensions.
Acknowledgements

We acknowledge funding from the Academy of Finland (grant numbers 308640 and 324345) and from Innovation Fund Denmark (grant number 8057-00036A). Our results were obtained using computational resources provided by the Aalto Science-IT project. Thanks to S. T. John for help implementing the VGP baseline for the heteroscedastic noise model.

References


Hubbell, S., Condit, R., and Foster, R. Barro Colorado forest census plot data. URL: http://ctfs.si.edu/webatlas/datasets/bci/, 2005.


State Space Expectation Propagation


Vanhatalo, J., Riihimäki, J., Hartikainen, J., Jylänki, P.,
Tolvanen, V., and Vehtari, A. GPstuff: Bayesian modeling
with Gaussian processes. Journal of Machine Learning

Wainwright, M. J. and Jordan, M. I. Graphical models, ex-
ponential families, and variational inference. Foundations

Wang, K., Pleiss, G., Gardner, J., Tyree, S., Weinberger,
K. Q., and Wilson, A. G. Exact Gaussian processes on a
million data points. In Advances in Neural Information
Processing Systems (NeurIPS), volume 32, pp. 14622–

Wilkinson, W. J., Andersen, M. R., Reiss, J. D., Stowell,
D., and Solin, A. End-to-end probabilistic inference for
nonstationary audio analysis. In Proceedings of the 36th
International Conference on Machine Learning (ICML),
volume 97 of Proceedings of Machine Learning Research,

filtering for additive noise case: Augmented vs. non-
augmented. In Proceedings of the American Control

perspective on Gaussian filters. IEEE Transactions on