Learning unknown forces in nonlinear models with Gaussian processes and autoregressive flows

Wil O C Ward
w.ward@sheffield.ac.uk

Department of Physics and Astronomy, The University of Sheffield

GPSS Workshop: Structurally Constrained Gaussian Processes
12 Sep 2019

Collaborative Work

Mauricio Alvarez

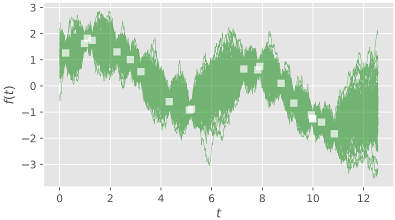
Tom Ryder

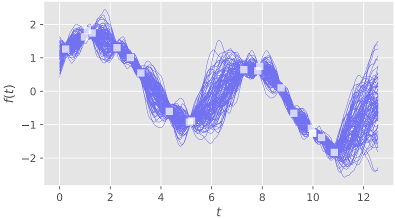
Dennis Prangle

Gaussian Processes

- GPs generalise Gaussian distribution
- Infinite dimension and non-parametric
- Defined in terms of mean and covariance function

$$f(t) \sim \mathrm{GP}(m(t), k(t, t'))$$





Motivating Example

Consider the model,

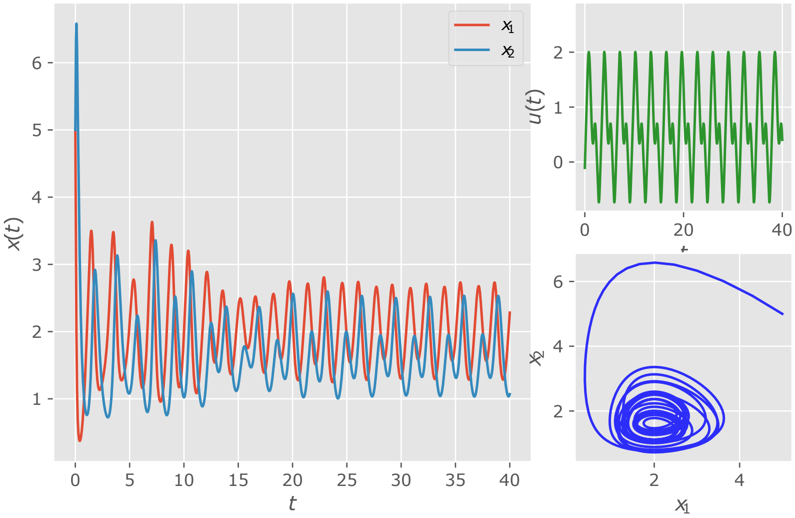
$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x} = \boldsymbol{\alpha}(\boldsymbol{x}(t), \boldsymbol{\theta}) + \begin{bmatrix} u(t) & 0 \end{bmatrix}^{\mathrm{T}}$$

Where $\boldsymbol{\alpha} : \mathbb{R}^2 \times \Theta \to \mathbb{R}^2$ are known dynamics:

$$\boldsymbol{\alpha}(\boldsymbol{x}, \boldsymbol{\theta}) = \begin{bmatrix} \theta_1 x_1 - \theta_2 x_1 x_2 \\ \theta_2 x_1 x_2 - \theta_3 x_2 \end{bmatrix}$$

…but $\boldsymbol{\theta}$ and $u(t)$ are unknown. How can we infer $\boldsymbol{x}(t)$ and $u(t)$ given some noisy observations $\mathbf{y} = [\boldsymbol{x}(\tau_j) + \varepsilon_j]_{j=0}^{N}$?

Motivating Example



Contents

Contents

Itô Processes

- Consider an ordinary differential equation describing the dynamics of some (vector-valued) function $\boldsymbol{x} : \mathbb{R} \to \mathbb{R}^d$
- The dynamics $\boldsymbol{\alpha_k} : \mathbb{R}^d \to \mathbb{R}^d$ are known but it is driven by a white-noise process with covariance as function of $\boldsymbol{x}$, $\boldsymbol{\Sigma} : \mathbb{R}^d \to \mathbb{R}^{d \times d}$

Ordinary Differential Equation with White Noise

$$\sum_{k=0}^{n} \alpha_k(\boldsymbol{x}, t; \boldsymbol{\theta}) \frac{\mathrm{d}^n}{\mathrm{d}t^n} \boldsymbol{x}(t) = \boldsymbol{\Sigma}^{1/2}(\boldsymbol{x}, t; \boldsymbol{\theta})\, \boldsymbol{w}(t)$$

Itô Processes

- Consider an ordinary differential equation describing the dynamics of some (vector-valued) function $\boldsymbol{x} : \mathbb{R} \to \mathbb{R}^d$
- The dynamics $\boldsymbol{\alpha_k} : \mathbb{R}^d \to \mathbb{R}^d$ are known but it is driven by a white-noise process with covariance as function of $\boldsymbol{x}$, $\boldsymbol{\Sigma} : \mathbb{R}^d \to \mathbb{R}^{d \times d}$

Stochastic Differential Equation

$$\sum_{k=0}^{n} \underbrace{\alpha_k(\boldsymbol{x}, t; \boldsymbol{\theta})}_{\text{drift terms}} \frac{\mathrm{d}^n}{\mathrm{d}t^n} \boldsymbol{x}(t) = \underbrace{\boldsymbol{\Sigma}^{1/2}(\boldsymbol{x}, t; \boldsymbol{\theta})}_{\text{diffusion}} \boldsymbol{w}(t)$$

Solutions to Itô Processes

- If system has linear dynamics, can solve exactly using Kalman filtering / Rauch-Tung-Streibel smoothing
- Assuming non-linearity, there are a number of approximation methods
- Stochastic extension to Euler method for iterative discrete-time estimation

Euler-Maruyama Discretisation

$$\boldsymbol{x}(t_{k+1}) - \boldsymbol{x}(t_k) \sim \mathrm{N}(\boldsymbol{\alpha}(\boldsymbol{x}(t_k))\Delta_t, \boldsymbol{\Sigma}\Delta_t)$$

Solutions to Itô Processes

- If system has linear dynamics, can solve exactly using Kalman filtering / Rauch-Tung-Streibel smoothing
- Assuming non-linearity, there are a number of approximation methods
- Stochastic extension to Euler method for iterative discrete-time estimation

Euler-Maruyama Discretisation as a Generative Prior

$$\boldsymbol{x}(t_{k+1}) \,|\, \boldsymbol{x}(t_k) \sim \mathrm{N}(\boldsymbol{x}(t_k) + \boldsymbol{\alpha}(\boldsymbol{x}(t_k))\Delta_t, \boldsymbol{\Sigma}\Delta_t)$$

Examples

- White noise process

$$w(t) \sim \mathrm{GP}\big(0, \varsigma^2 \delta(t - t')\big)$$

- Half-integer ($\nu = p + 1/2$) Matérn models

$$f_\nu(t) \sim \mathrm{GP}\left(0, \sigma^2 \exp\left(-\lambda |t - t'|\right) \frac{p!}{(2p)!} \sum_{i=0}^{p} \frac{(p+i)!}{i!(p-i)!} \left(2\lambda |t - t'|\right)^{p-i}\right)$$

- Gaussian Radial Basis / Exponentiated Quadratic ($\nu \to \infty$)

$$f(t) \sim \mathrm{GP}\big(0, \sigma^2 \exp(-\lambda |t - t'|^2)\big)$$

Gaussian Processes as SDEs

Examples

- White noise process

$$\mathrm{d}w(t) = \varsigma \mathrm{d}\beta$$

- Half-integer ($\nu = p + 1/2$) Matérn models

$$\sum_{i=1}^{p} \begin{pmatrix} p \\ i-1 \end{pmatrix} \lambda^{p+1-i} \frac{\mathrm{d}^i}{\mathrm{d}t^i} f(t) = -\lambda^{p+1} f(t) + w(t)$$

- Gaussian Radial Basis / Exponentiated Quadratic ($\nu \to \infty$)
  infinitely differentiable so cannot represent as Itô process exactly

Gaussian Processes as SDEs

Examples

- White noise process

$$\mathrm{d}w(t) = \varsigma \mathrm{d}\beta$$

- Half-integer ($\nu = p + 1/2$) Matérn models

$$\mathrm{d}\boldsymbol{f}(t) = \underbrace{\begin{bmatrix} 0 & 1 & & \\ \ddots & \ddots & \ddots & \\ & & 0 & 1 \\ -a_1\lambda^{p+1} & -a_2\lambda^{p} & \cdots & -a_p\lambda \end{bmatrix}}_{\mathbf{G}} \underbrace{\begin{bmatrix} f(t) \\ \mathrm{d}f/\mathrm{d}t \\ \vdots \\ \mathrm{d}^{p-1}f/\mathrm{d}t^{p-1} \end{bmatrix}}_{\boldsymbol{f}(t)} \mathrm{d}t + \underbrace{\varsigma_\nu \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \mathrm{d}\boldsymbol{\beta}}_{w(t)\mathrm{d}t}$$

Stochastic Latent Force Models

- Recall our motivating example, a mixture of known dynamics with some hidden input function
- General form:

$$\alpha_0(\boldsymbol{x}, t; \boldsymbol{\theta})\boldsymbol{x}(t) + \alpha_1(\boldsymbol{x}, t; \boldsymbol{\theta})\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t) + \ldots = \boldsymbol{u}(t)$$

- Placing a GP prior over $\boldsymbol{u}(t)$
- Termed *latent force models*

M. A. Alvarez, D. Luengo, and N. D. Lawrence. Linear latent force models using Gaussian processes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(11):2693–2705, 2013

Companion Form LFMs

- Easy enough to reframe n^{th}-order differential equation as first-order $\mathrm{d}\boldsymbol{f}/\mathrm{d}t = \mathbf{D}(\boldsymbol{f}(t), \boldsymbol{\theta}) + \boldsymbol{L}w(t)$

Companion Form LFMs

- Easy enough to reframe n^{th}-order differential equation as first-order $\mathrm{d}\boldsymbol{f}/\mathrm{d}t = \mathbf{D}(\boldsymbol{f}(t), \boldsymbol{\theta}) + \boldsymbol{L}w(t)$

Companion Form

$$\boldsymbol{f}(\tau) = \begin{bmatrix} x(\tau) & \frac{\mathrm{d}x}{\mathrm{d}t}\Big|_{t=\tau} & \cdots & \frac{\mathrm{d}^{n-1}x}{\mathrm{d}t^{n-1}}\Big|_{t=\tau} & u(\tau) & \frac{\mathrm{d}u}{\mathrm{d}t}\Big|_{t=\tau} & \cdots & \frac{\mathrm{d}^{m-1}u}{\mathrm{d}t^{m-1}}\Big|_{t=\tau} \end{bmatrix}^\top$$

$$\boldsymbol{D}(\boldsymbol{f}(t), \boldsymbol{\theta}) = \begin{bmatrix} f_2 \\ f_3 \\ \vdots \\ \breve{\alpha}_0 f_1 + \sum_{i=1}^{n-1} \breve{\alpha}_i f_{i+1} + f_{n+1} \\ f_{n+2} \\ f_{n+3} \\ \vdots \\ a_0 f_{n+1} + \sum_{i=1}^{m-1} a_i f_{n+i+1} \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

Contents

Inferring the Joint Posterior of a Non-Linear LFM

Problem: Infer $\boldsymbol{f}$ and $\boldsymbol{\theta}$

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{f}(t) = \boldsymbol{D}(\boldsymbol{f}(t), \boldsymbol{\theta}) + \boldsymbol{L}w(t)$$

- We cannot infer $\boldsymbol{f}$ exactly if $\boldsymbol{D}$ is non-linear since the joint posterior is intractible
- Pseudo-chaos under some systems
- Non-linear versions of filters/smoothers, e.g. E/UKF, ADF, SMC
- Difficult to do joint parameter estimation, difficult to use autodifferentiation

J. Hartikainen, M. Seppänen, and S. Särkkä. State-space inference for non-linear latent force models with application to satellite orbit prediction. In *ICML*, pages 723–730, 2012.

Variational Bridge Constructs

We want to build variational approximation of conditional posterior: $p(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta} \mid \boldsymbol{y})$.

Variational Bayes

Find $q^* \in \mathcal{Q}$, such that

$$q^* = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}[q(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}) \parallel p(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta} \mid \boldsymbol{y})]$$

where $\mathcal{Q}$ is a family of distributions parameterised by $\boldsymbol{\phi}$

Variational Bridge Constructs

We want to build variational approximation of conditional posterior: $p(\boldsymbol{f}, \boldsymbol{\theta} \mid \boldsymbol{y})$.

Variational Bayes

Find $q^* \in \mathcal{Q}$, such that

$$q^* = \underset{q \in \mathcal{Q}}{\arg\min}\, \mathrm{KL}[q(\boldsymbol{f}, \boldsymbol{\theta}) \,\|\, p(\boldsymbol{f}, \boldsymbol{\theta} \mid \boldsymbol{y})]$$

where $\mathcal{Q}$ is a family of distributions parameterised by $\boldsymbol{\phi}$

Variational Bridge Constructs

Evidence Lower Bound (ELBO)

$$\mathcal{L}(\boldsymbol{\phi}) = \mathbb{E}_{\boldsymbol{f},\boldsymbol{\theta}\sim q}\left[\log p(\boldsymbol{f},\boldsymbol{\theta},\mathbf{y}) - \log q(\boldsymbol{f},\boldsymbol{\theta})\right]$$

Variational Bridge Constructs

Unbiased Evidence Lower Bound (ELBO)

$$\hat{\mathcal{L}}(\boldsymbol{\phi}) = \frac{1}{n_s} \sum_{i=1}^{n_s} \log \frac{p(\boldsymbol{\theta}^{(i)}) p(\boldsymbol{f}^{(i)} \mid \boldsymbol{\theta}^{(i)}) p(\mathbf{y} \mid \mathbf{f}^{(i)}, \boldsymbol{\theta}^{(i)})}{q(\boldsymbol{\theta}^{(i)}) q(\boldsymbol{f}^{(i)} \mid \boldsymbol{\theta}^{(i)})}$$

where $\boldsymbol{f}^{(i)} \sim q(\boldsymbol{f} \mid \boldsymbol{\theta}^{(i)})$ and $\boldsymbol{\theta}^{(i)} \sim q(\boldsymbol{\theta})$ $i = 1, \ldots, n_s$

Variational Bridge Constructs

Unbiased Evidence Lower Bound (ELBO)

$$\hat{\mathcal{L}}(\boldsymbol{\phi}) = \frac{1}{n_s} \sum_{i=1}^{n_s} \log \frac{p(\boldsymbol{\theta}^{(i)}) p(\boldsymbol{f}^{(i)} \mid \boldsymbol{\theta}^{(i)}) p(\mathbf{y} \mid \mathbf{f}^{(i)}, \boldsymbol{\theta}^{(i)})}{q(\boldsymbol{\theta}^{(i)}) q(\boldsymbol{f}^{(i)} \mid \boldsymbol{\theta}^{(i)})}$$

where $\boldsymbol{f}^{(i)} \sim q(\boldsymbol{f} \mid \boldsymbol{\theta}^{(i)})$ and $\boldsymbol{\theta}^{(i)} \sim q(\boldsymbol{\theta})\ i = 1, \ldots, n_s$

Likelihood Agnostic

Valid for any (differentiable?) observation model $p(\mathbf{y} \mid \boldsymbol{f}, \boldsymbol{\theta})$

Black-box Variational Inference

- Black-box variational inference (BBVI) is predicated on the fact that the gradient of ELBO can be written as an unbiased average
- Straightforward since we have $\hat{\mathcal{L}}(\phi)$ as an unbiased average

Black-box Variational Inference

- Black-box variational inference (BBVI) is predicated on the fact that the gradient of ELBO can be written as an unbiased average
- Straightforward since we have $\hat{\mathcal{L}}(\boldsymbol{\phi})$ as an unbiased average

Monte Carlo approximation of ELBO gradient

$$\nabla_\phi \mathcal{L}(\boldsymbol{\phi}) \approx \frac{1}{n_s} \sum_{i=1}^{n_s} \nabla_\phi \log q(\boldsymbol{f}^{(i)}, \boldsymbol{\theta}^{(i)}) \log \frac{p(\boldsymbol{f}^{(i)}, \boldsymbol{\theta}^{(i)}, \mathbf{y})}{q(\boldsymbol{f}^{(i)}, \boldsymbol{\theta}^{(i)})}$$

where $\boldsymbol{f}^{(i)} \sim q(\boldsymbol{f} \mid \boldsymbol{\theta}^{(i)})$ and $\boldsymbol{\theta}^{(i)} \sim q(\boldsymbol{\theta})$ $i = 1, \dots, n_s$

R. Ranganath, S. Gerrish, and D. Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, 2014.

D. Duvenaud and R. P. Adams. Black-box stochastic variational inference in five lines of Python. In *NIPS Workshop on Black-box Learning and Inference*, 2015.

Algorithm 1 BBVI with gradient ascent

Initialise $\boldsymbol{\phi}_0$ (randomly)
$j \leftarrow 0$
while *not converged* **do**
  Calculate $\nabla_{\phi}\mathcal{L}(\boldsymbol{\phi}_j)$
  Update $\boldsymbol{\phi}$ w.r.t. ELBO gradient, e.g.:
$$\boldsymbol{\phi}_{j+1} \leftarrow \boldsymbol{\phi}_j + h\nabla_{\phi}\mathcal{L}(\boldsymbol{\phi}_j)$$
  $j \leftarrow j + 1$
end while
Variational approximation $q(\boldsymbol{f}, \boldsymbol{\theta} \mid \boldsymbol{\phi}_j) \approx p(\boldsymbol{f}, \boldsymbol{\theta} \mid \mathbf{y})$

Parameter Estimation: $q(\boldsymbol{\theta})$

Commonly in system estimation, the model parameters, $\boldsymbol{\theta}$ are unknown.

We can also give these a Bayesian treatment by using variational representation of the posterior.

We can use any variational approach here, e.g. mean-field:

$$q(\boldsymbol{\theta}) = \prod \mathrm{N}(\theta_i \mid m_i, s_i)$$

Here, the free parameters are scalars, $\boldsymbol{\phi}_\theta = \{(m_i, s_i)\}_{\forall i}$

Filtering Density: $p(\boldsymbol{f} \mid \boldsymbol{\theta})$

Represent stochastic process, $\boldsymbol{f}$ as a filtering distribution with moments $\boldsymbol{m}(t)$ and $\boldsymbol{P}(t)$

Mean term: $$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{m}(t) = \boldsymbol{D}(\boldsymbol{m}, t; \boldsymbol{\theta})$$

Filtering Density: $p(\boldsymbol{f} \mid \boldsymbol{\theta})$

Represent stochastic process, $\boldsymbol{f}$ as a filtering distribution with moments $\boldsymbol{m}(t)$ and $\boldsymbol{P}(t)$

Mean term: $$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{m}(t) = \boldsymbol{D}(\boldsymbol{m}, t; \boldsymbol{\theta})$$

Covariance term: ??

Extended Filtering Density: $p(\boldsymbol{f} \mid \boldsymbol{\theta})$

Represent stochastic process, $\boldsymbol{f}$ as a filtering distribution with moments $\boldsymbol{m}(t)$ and $\boldsymbol{P}(t)$

Mean term: $$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{m}(t) = \boldsymbol{D}(\boldsymbol{m}, t; \boldsymbol{\theta})$$

Covariance term:

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{P}(t) = \boldsymbol{J}_D(\boldsymbol{m}, t; \boldsymbol{\theta})\boldsymbol{P}(t) + \boldsymbol{P}(t)\boldsymbol{J}_D(\boldsymbol{m}, t; \boldsymbol{\theta})^{\mathrm{T}} + \boldsymbol{L}\varsigma^2\boldsymbol{L}^{\mathrm{T}}$$

Extended Filtering Density: $p(\boldsymbol{f} \mid \boldsymbol{\theta})$

Represent stochastic process, $\boldsymbol{f}$ as a filtering distribution with moments $\boldsymbol{m}(t)$ and $\boldsymbol{P}(t)$

Assume steady state: $\mathrm{d}\boldsymbol{P}/\mathrm{d}t = 0$.

Extended Filtering Density: $p(\boldsymbol{f} \mid \boldsymbol{\theta})$

Represent stochastic process, $\boldsymbol{f}$ as a filtering distribution with moments $\boldsymbol{m}(t)$ and $\boldsymbol{P}(t)$

Assume steady state: $\mathrm{d}\boldsymbol{P}/\mathrm{d}t = 0$.

Denote covariance in steady state by $\tilde{\boldsymbol{\Sigma}}$ and solve

$$\boldsymbol{J}_D(\boldsymbol{m}, t; \boldsymbol{\theta})\tilde{\boldsymbol{\Sigma}} + \tilde{\boldsymbol{\Sigma}}\boldsymbol{J}_D(\boldsymbol{m}, t; \boldsymbol{\theta})^{\mathrm{T}} = -\boldsymbol{L}\varsigma^2\boldsymbol{L}^{\mathrm{T}}$$

Extended Filtering Density: $p(\boldsymbol{f} \mid \boldsymbol{\theta})$

Represent stochastic process, $\boldsymbol{f}$ as a filtering distribution with moments $\boldsymbol{m}(t)$ and $\boldsymbol{P}(t)$

Assume steady state: $\mathrm{d}\boldsymbol{P}/\mathrm{d}t = 0$.

Denote covariance in steady state by $\tilde{\boldsymbol{\Sigma}}$ and solve

$$\boldsymbol{J}_D(\boldsymbol{m}, t; \boldsymbol{\theta})\tilde{\boldsymbol{\Sigma}} + \tilde{\boldsymbol{\Sigma}}\boldsymbol{J}_D(\boldsymbol{m}, t; \boldsymbol{\theta})^{\mathrm{T}} = -\boldsymbol{L}\varsigma^2\boldsymbol{L}^{\mathrm{T}}$$

Example of continuous Lyapunov equation; easy to solve numerically, but need an differentiable form of $\tilde{\boldsymbol{\Sigma}}$

Extended Filtering Density: $p(\boldsymbol{f} \mid \boldsymbol{\theta})$

Represent stochastic process, $\boldsymbol{f}$ as a filtering distribution with moments $\boldsymbol{m}(t)$ and $\boldsymbol{P}(t)$

Assume steady state: $\mathrm{d}\boldsymbol{P}/\mathrm{d}t = 0$.

Denote covariance in steady state by $\tilde{\boldsymbol{\Sigma}}$ and solve

$$\boldsymbol{J}_D(\boldsymbol{m}, t; \boldsymbol{\theta})\tilde{\boldsymbol{\Sigma}} + \tilde{\boldsymbol{\Sigma}}\boldsymbol{J}_D(\boldsymbol{m}, t; \boldsymbol{\theta})^{\mathrm{T}} = -\boldsymbol{L}\varsigma^2\boldsymbol{L}^{\mathrm{T}}$$

Example of continuous Lyapunov equation; easy to solve numerically, but need an differentiable form of $\tilde{\boldsymbol{\Sigma}}$

$\tilde{\boldsymbol{\Sigma}}$ *may be* a function of $\boldsymbol{m}(t)$ and $\boldsymbol{\theta}$ so is stochastic too.

Extended Filtering Density: $p(\boldsymbol{f} \mid \boldsymbol{\theta})$

Represent stochastic process, $\boldsymbol{f}$ as a filtering distribution with moments $\boldsymbol{m}(t)$ and $\boldsymbol{P}(t)$

Assume steady state: $\mathrm{d}\boldsymbol{P}/\mathrm{d}t = 0$.

Extended Filtering Density: $p(\boldsymbol{f} \mid \boldsymbol{\theta})$

Represent stochastic process, $\boldsymbol{f}$ as a filtering distribution with moments $\boldsymbol{m}(t)$ and $\boldsymbol{P}(t)$

Assume steady state: $\mathrm{d}\boldsymbol{P}/\mathrm{d}t = 0$.

Denote covariance in steady state by $\tilde{\boldsymbol{\Sigma}}$ and solve

$$\boldsymbol{J}_D(\boldsymbol{m}, t; \boldsymbol{\theta})\tilde{\boldsymbol{\Sigma}} + \tilde{\boldsymbol{\Sigma}}\boldsymbol{J}_D(\boldsymbol{m}, t; \boldsymbol{\theta})^{\mathrm{T}} = -\boldsymbol{L}\varsigma^2\boldsymbol{L}^{\mathrm{T}}$$

Extended Filtering Density: $p(\boldsymbol{f} \mid \boldsymbol{\theta})$

Represent stochastic process, $\boldsymbol{f}$ as a filtering distribution with moments $\boldsymbol{m}(t)$ and $\boldsymbol{P}(t)$

Assume steady state: $\mathrm{d}\boldsymbol{P}/\mathrm{d}t = 0$.

Denote covariance in steady state by $\tilde{\boldsymbol{\Sigma}}$ and solve

$$\boldsymbol{J}_D(\boldsymbol{m}, t; \boldsymbol{\theta})\tilde{\boldsymbol{\Sigma}} + \tilde{\boldsymbol{\Sigma}}\boldsymbol{J}_D(\boldsymbol{m}, t; \boldsymbol{\theta})^{\mathrm{T}} = -\boldsymbol{L}\varsigma^2\boldsymbol{L}^{\mathrm{T}}$$

Example of continuous Lyapunov equation; easy to solve numerically, but need an differentiable form of $\tilde{\boldsymbol{\Sigma}}$

Extended Filtering Density: $p(\boldsymbol{f} \mid \boldsymbol{\theta})$

Represent stochastic process, $\boldsymbol{f}$ as a filtering distribution with moments $\boldsymbol{m}(t)$ and $\boldsymbol{P}(t)$

Assume steady state: $\mathrm{d}\boldsymbol{P}/\mathrm{d}t = 0$.

Denote covariance in steady state by $\tilde{\boldsymbol{\Sigma}}$ and solve

$$\boldsymbol{J}_D(\boldsymbol{m}, t; \boldsymbol{\theta})\tilde{\boldsymbol{\Sigma}} + \tilde{\boldsymbol{\Sigma}}\boldsymbol{J}_D(\boldsymbol{m}, t; \boldsymbol{\theta})^{\mathrm{T}} = -\boldsymbol{L}\varsigma^2\boldsymbol{L}^{\mathrm{T}}$$

Example of continuous Lyapunov equation; easy to solve numerically, but need an differentiable form of $\tilde{\boldsymbol{\Sigma}}$

$\tilde{\boldsymbol{\Sigma}}$ *may be* a function of $\boldsymbol{m}(t)$ and $\boldsymbol{\theta}$ so is stochastic too.

Transition density: $p(f_k \mid f_{k-1}\boldsymbol{\theta})$

We construct a discrete-time transition density using Euler-Maruyama:

$$p(f_k \mid f_{k-1}, \boldsymbol{\theta}) = \mathrm{N}\left(f_k \mid \mu_\Delta, \Sigma_\Delta\right),$$

where

$$\begin{aligned}
\mu_\Delta &= f_{k-1} + \boldsymbol{D}(f_{k-1}, t_k; \boldsymbol{\theta})\Delta_t \\
\Sigma_\Delta &= \tilde{\boldsymbol{\Sigma}}(f_{k-1}, t_k; \boldsymbol{\theta}) - \exp\left(\Delta_t \boldsymbol{J}_D\right) \tilde{\boldsymbol{\Sigma}}(\boldsymbol{f}_{k-1}, t_k; \boldsymbol{\theta}) \exp\left(\Delta_t \boldsymbol{J}_D\right)^{\mathrm{T}}
\end{aligned}$$

Generative model: $p(\boldsymbol{f} \mid \boldsymbol{\theta})$

Marginal

$$p(\boldsymbol{f} \mid \boldsymbol{\theta}) = p(f_0 \mid \boldsymbol{\theta}) \prod_{k=1}^{T} p(f_k \mid f_{k-1}, \boldsymbol{\theta})$$

Generative model: $p(\boldsymbol{f} \mid \boldsymbol{\theta})$

Marginal

$$p(\boldsymbol{f} \mid \boldsymbol{\theta}) = p(f_0 \mid \boldsymbol{\theta}) \prod_{k=1}^{T} p(f_k \mid f_{k-1}, \boldsymbol{\theta})$$

Additional points

$$f_k = \boldsymbol{f}(t_k) \qquad f_{k+1} = \boldsymbol{f}(t_{k+1}) = \boldsymbol{f}(t_k + \Delta_t)$$

$$p(f_k \mid f_{k-1}) \equiv p(x_k \mid x_{k-1}, u_k) p(u_k \mid u_{k-1})$$

Variational Approximation: $q(\boldsymbol{f} \mid \boldsymbol{\theta})$

- Family of distributions parameterised by $\boldsymbol{\phi}$
- Needs to be flexible, sampleable and invertible (for autodifferentiation)

Variational Approximation: $q(\boldsymbol{f} \mid \boldsymbol{\theta})$

- Family of distributions parameterised by $\boldsymbol{\phi}$
- Needs to be flexible, sampleable and invertible (for autodifferentiation)
- Look to (Bayesian) neural networks and other deep models

Variational Approximation: $q(\boldsymbol{f} \mid \boldsymbol{\theta})$

- Family of distributions parameterised by $\boldsymbol{\phi}$
- Needs to be flexible, sampleable and invertible (for autodifferentiation)
- Look to (Bayesian) neural networks and other deep models
- Need to encode temporal (recurrent) structure

Variational Approximation: $q(\boldsymbol{f} \mid \boldsymbol{\theta})$

- Family of distributions parameterised by $\boldsymbol{\phi}$
- Needs to be flexible, sampleable and invertible (for autodifferentiation)
- Look to (Bayesian) neural networks and other deep models
- Need to encode temporal (recurrent) structure
    - RNNs with priors on the weights

Variational Approximation: $q(\boldsymbol{f} \mid \boldsymbol{\theta})$

- Family of distributions parameterised by $\boldsymbol{\phi}$
- Needs to be flexible, sampleable and invertible (for autodifferentiation)
- Look to (Bayesian) neural networks and other deep models
- Need to encode temporal (recurrent) structure
  - RNNs with priors on the weights
  - Normalising flows

Parametrising q with an RNN

Pros

- Can represent high dimensional recurrent structure
- Bi-directional RNN can represent first-order Markov properties of model
- Priors over weights and optimise in weight-space

Parametrising q with an RNN

Pros

- Can represent high dimensional recurrent structure
- Bi-directional RNN can represent first-order Markov properties of model
- Priors over weights and optimise in weight-space

Cons

- Need to sample sequentially
- BPTT inefficient for propagation of gradients
- Doesn't handle latent dimensions well

Inverse Autoregressive Flows

- Want to define a distribution for $\boldsymbol{f}$ that is invertible and expressive
- Inverse autoregressive flows (IAFs) introduce a base random vector $\boldsymbol{z}_0 \sim \mathrm{N}(\mathbf{0}, \mathbf{I})$
- Layers of this random variable are shifted and scaled through 1-D convolutions to create autoregressive model
- Very flexible, and can sample in parallel

Inverse Autoregressive Flows

- Want to define a distribution for $\boldsymbol{f}$ that is invertible and expressive
- Inverse autoregressive flows (IAFs) introduce a base random vector $\boldsymbol{z}_0 \sim \mathrm{N}(\mathbf{0}, \mathbf{I})$
- Layers of this random variable are shifted and scaled through 1-D convolutions to create autoregressive model
- Very flexible, and can sample in parallel

Autoregressive Flows

$$\boldsymbol{z}_j = \boldsymbol{\sigma}_j \odot \boldsymbol{z}_{j-1} + \boldsymbol{\mu}_j$$

where

$$[\boldsymbol{\mu}_j, \boldsymbol{s}_j] = \textsc{autoregressiveNN}(\boldsymbol{z}_{j-1}, \boldsymbol{y}, \boldsymbol{\theta}) \text{ and } \boldsymbol{\sigma}_j = \log(1 + \exp \boldsymbol{s}_j)$$

$$\boldsymbol{f} = \textsc{bijector}(\boldsymbol{z}_N)$$

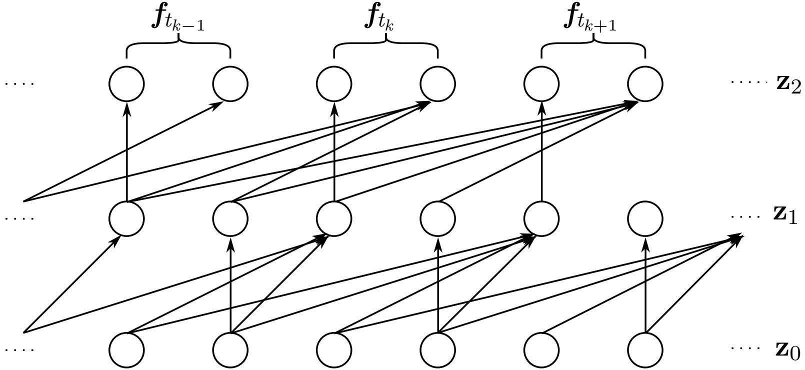
Algorithm 2 j^{th} Autoregressive Neural Network Layer

$\xi^{(0a)} \leftarrow \textsc{conv1d}(\boldsymbol{z}_{j-1}, \mathbf{y}, \mathbf{t})$
$\xi^{(0b)} \leftarrow \textsc{dense}(\boldsymbol{\theta})$
$\xi^{(1)} \leftarrow \textsc{elu}(\xi^{(0a)} + \xi^{(0b)})$
for $i = 2 \ldots n_\ell$ **do**
  $\xi^{(i)} \leftarrow \textsc{batchnorm}(\textsc{conv1d}(\textsc{elu}(\xi^{(i-1)})))$
end for
$[\boldsymbol{\mu}_j, \boldsymbol{s}_j] \leftarrow \textsc{conv1d}(\xi^{(n_\ell)})$
$\boldsymbol{\sigma}_j \leftarrow \textsc{softplus}(\boldsymbol{s}_j)$
$\boldsymbol{z}_j \leftarrow \boldsymbol{\sigma}_j \odot \boldsymbol{z}_{j-1} + \boldsymbol{\mu}_j$

Locally Masked Multivariate Inverse Autoregressive Flows

- Passing the entire flow vector, $\boldsymbol{z}_j$ can lead to complex (unrepresentative temporal dependencies)
- We use a local receptive field to update flow layers (similar to Wavenet)
- Rolled out multidimensional state (f_k) in sequence
- Hacks and tricks to approximate locally informed flow state

Locally Masked Multivariate Inverse Autoregressive Flows



Variational Log Density

$$\log q(\boldsymbol{f} \mid \boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{z}_0^{\mathrm{T}}\boldsymbol{z}_0 + \frac{T}{2}\log 2\pi + T\sum_{i=1}^{N}\log \boldsymbol{\sigma}_j + \log|\boldsymbol{J}_{-1}(\boldsymbol{f})|$$

Variational Log Density

$$\log q(\boldsymbol{f}^{(i)} \mid \boldsymbol{\theta}^{(i)}) = -\frac{1}{2}\boldsymbol{z}_0^{(i)\mathrm{T}}\boldsymbol{z}_0^{(i)} + \frac{T}{2}\log 2\pi + T\sum_{i=1}^{N}\log \boldsymbol{\sigma}_j^{(i)} + \log |\boldsymbol{J}_{-1}(\boldsymbol{f}^{(i)})|$$

$$\boldsymbol{z}_0^{(i)} \sim \mathrm{N}(\mathbf{0}, \mathbf{I})$$

Unbiased Evidence Lower Bound

ELBO

$$\hat{\mathcal{L}}(\boldsymbol{\phi}) = \frac{1}{n_s}\sum_{i=1}^{n_s} \log \frac{p(\boldsymbol{\theta}^{(i)})p(\boldsymbol{f}^{(i)} \mid \boldsymbol{\theta}^{(i)})p(\mathbf{y} \mid \mathbf{f}^{(i)}, \boldsymbol{\theta}^{(i)})}{q(\boldsymbol{\theta}^{(i)})q(\boldsymbol{f}^{(i)} \mid \boldsymbol{\theta}^{(i)})}$$

where $\boldsymbol{f}^{(i)} \sim q(\boldsymbol{f} \mid \boldsymbol{\theta}^{(i)})$ and $\boldsymbol{\theta}^{(i)} \sim q(\boldsymbol{\theta})\ i = 1, \ldots, n_s$

Contents

Exponential Gaussian Process

$$f(t) \sim \mathrm{GP}\big(0, \sigma_f^2 \exp(-\lambda|t - t'|)\big)$$

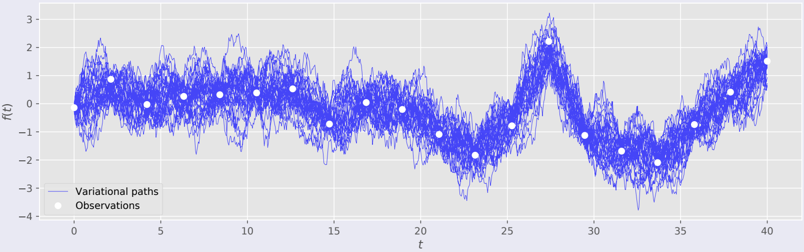
Exponential Gaussian Process

$$f(t) \sim \mathrm{GP}\big(0, \sigma_f^2 \exp(-\lambda|t - t'|)\big)$$

$$\mathrm{d}f(t) = -\lambda f(t) + 2\sigma_f^2 \lambda \mathrm{d}\beta(t)$$

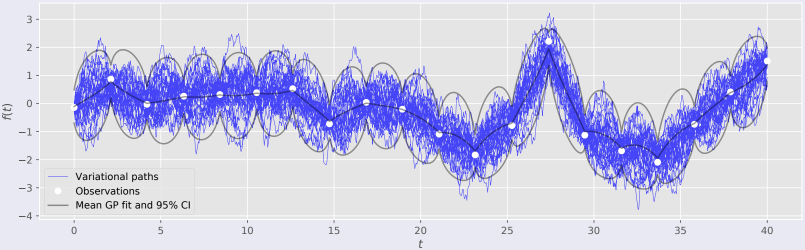
Exponential Gaussian Process

Samples from $q(f \mid \boldsymbol{\theta})$



Exponential Gaussian Process

Samples from $q(f \mid \boldsymbol{\theta})$ and mean and covariance of $p(f \mid \mathbf{y}, \boldsymbol{\theta})$



Model Criticism

- Visual confirmation fine, it *looks* like a good estimate
- Empirical evidence for reliability needed
- Map corresponding samples from p and q to RKHS
- Two-sample test with MMD to validate approximation

Maximum Mean Discrepancy (MMD)

- MMD is a measure of distance between two probabilities
- Samples are embedded in an RKHS
- Metric describes distance as some norm in the RKHS
- Two-sample testing for $H_0 : \widehat{\mathrm{MMD}}^2(\mu_p, \mu_q) = 0$
- Gretton, et al. A kernel two-sample test. *JMLR*, 2012

Model Criticism

MMD^2 values comparing samples from $q(f \mid \boldsymbol{\theta})$ and $p(f \mid \mathbf{y}, \boldsymbol{\theta})$

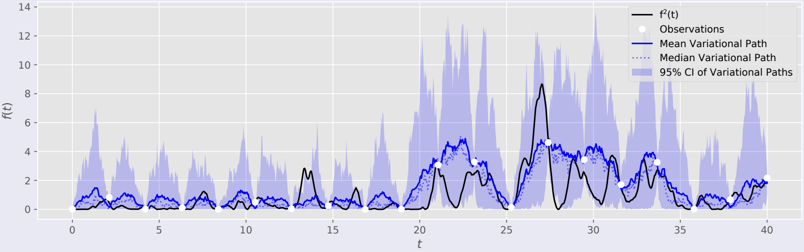
Fit on different number of observations, N

Epoch	10	100	500	1 000	2 500	25 000
$N = 6$	0.1111	0.1267	0.0596	0.0484	0.0556	–
$N = 20$	0.2731	0.1147	0.0654	0.0696	0.0471	**0.0316**

Thresholds for rejection at 95% confidence: 0.0371 ($N = 6$) and 0.0337 ($N = 20$)

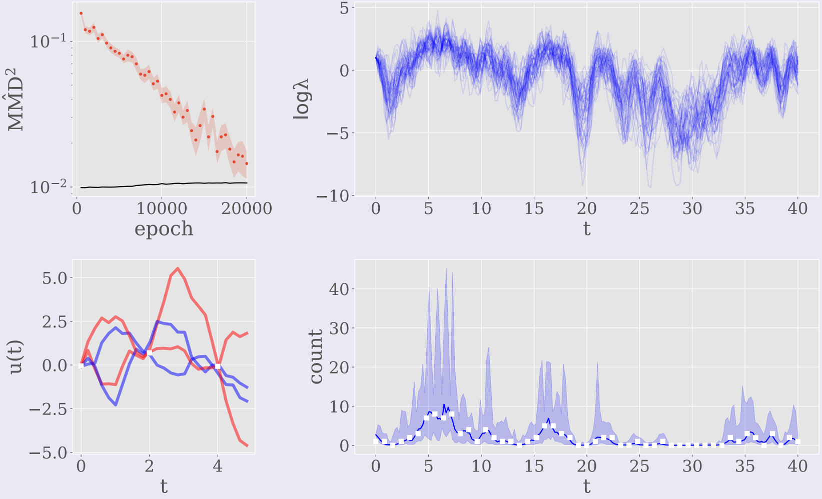
Matérn Covariances and Non-Gaussian Likelihoods

Summary statistics for $p(\mathbf{y} \mid f^{(i)}, \boldsymbol{\theta})$, $f^{(i)} \sim q(f \mid \boldsymbol{\theta})$, plotted against true latent function, $f^2(t)$



Matérn Covariances and Non-Gaussian Likelihoods

Approximating Matérn 3/2 GPs



Contents

Toy Non-Linear ODE

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = -\frac{2}{3}\sin(\omega x(t)) + u(t)$$

Toy Non-Linear ODE

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = -\frac{2}{3}\sin(\omega x(t)) + u(t)$$

$$u(t) \sim \mathrm{GP}(0, k_{\nu=1/2}(t, t'))$$

Toy Non-Linear ODE

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = -\frac{2}{3}\sin(\omega x(t)) + u(t)$$

$$u(t) \sim \mathrm{GP}(0, k_{\nu=1/2}(t, t'))$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\underbrace{\begin{bmatrix} x(t) \\ u(t) \end{bmatrix}}_{\boldsymbol{f}(t)} = \underbrace{\begin{bmatrix} -2\cos(\omega f_1)/3 + f_2 \\ -\lambda f_2 \end{bmatrix}}_{\boldsymbol{D}(\boldsymbol{f}(t), \boldsymbol{\theta})} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\mathbf{L}} w(t),$$

Toy Non-Linear ODE

The Jacobian of $\boldsymbol{D}(\boldsymbol{f}(t), \boldsymbol{\theta})$ w.r.t $\boldsymbol{f}$ is defined:

$$\mathbf{J}_D(\boldsymbol{f}(t)) = \begin{bmatrix} 2\omega \sin(\omega f_1)/3 & 1 \\ 0 & -\lambda \end{bmatrix},$$

Toy Non-Linear ODE

The Jacobian of $\boldsymbol{D}(\boldsymbol{f}(t), \boldsymbol{\theta})$ w.r.t $\boldsymbol{f}$ is defined:

$$\mathbf{J}_D(\boldsymbol{f}(t)) = \begin{bmatrix} 2\omega\sin(\omega f_1)/3 & 1 \\ 0 & -\lambda \end{bmatrix},$$

and steady state covariance $\tilde{\boldsymbol{\Sigma}}$ such that

$$\mathbf{J}_D(\boldsymbol{f}(t))\tilde{\boldsymbol{\Sigma}} + \tilde{\boldsymbol{\Sigma}}[\mathbf{J}_D(\boldsymbol{f}(t))]^\top + 2\lambda\sigma^2\mathbf{L}\mathbf{L}^\mathrm{T} = 0$$

Toy Non-Linear ODE

The Jacobian of $\boldsymbol{D}(\boldsymbol{f}(t), \boldsymbol{\theta})$ w.r.t $\boldsymbol{f}$ is defined:

$$\mathbf{J}_D(\boldsymbol{f}(t)) = \begin{bmatrix} 2\omega \sin(\omega f_1)/3 & 1 \\ 0 & -\lambda \end{bmatrix},$$

and steady state covariance $\tilde{\boldsymbol{\Sigma}}$ such that

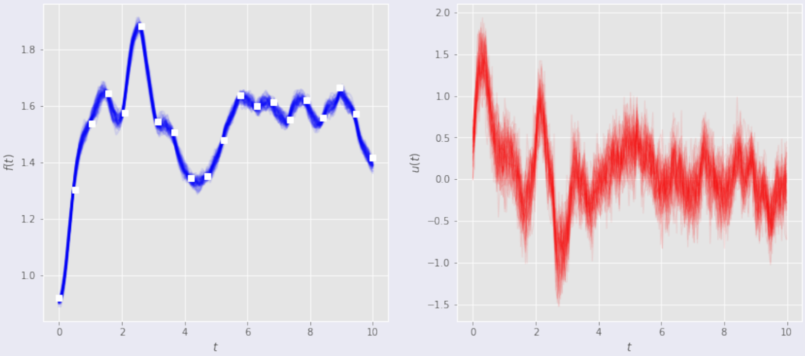
$$\mathbf{J}_D(\boldsymbol{f}(t))\tilde{\boldsymbol{\Sigma}} + \tilde{\boldsymbol{\Sigma}}[\mathbf{J}_D(\boldsymbol{f}(t))]^\top + 2\lambda\sigma^2\mathbf{L}\mathbf{L}^\mathrm{T} = 0$$

is

$$\tilde{\boldsymbol{\Sigma}} = \begin{bmatrix} \frac{\sigma^2\lambda}{2\lambda\omega\sin(\omega f_1)(2\omega\sin(\omega f_1)/3-\lambda)/3} & \frac{\sigma^2\lambda}{\lambda^2-2\lambda\omega\sin(\omega f_1)/3} \\ \frac{\sigma^2\lambda}{\lambda^2-2\lambda\omega\sin(\omega f_1)/3} & \sigma^2 \end{bmatrix}$$

Toy Non-Linear ODE

Samples of joint posterior of $\boldsymbol{x}$, $\boldsymbol{u}$ and $\boldsymbol{\theta}$



Real World: Gene Expression Data

- Multi-output system with non-linear dependency on input

$$\frac{\mathrm{d}}{\mathrm{d}t}x_d(t) = a_d - b_d x_d(t) + s_d \frac{u(t)}{\gamma_d + u(t)}$$

- x_d is a model of gene expression, that's noisily observable
- u models the concentration of the transcription factor regulating the observed genes

Real World: Gene Expression Data

- Multi-output system with non-linear dependency on input

$$\frac{\mathrm{d}}{\mathrm{d}t}x_d(t) = a_d - b_d x_d(t) + s_d \frac{u(t)}{\gamma_d + u(t)}$$

- $x_d(t), u(t) > 0$, $\theta_d = \{a_d, b_d, s_d, \gamma_d\}$
- $d = 1, ..., ?$

Real World: Gene Expression Data

- Multi-output system with non-linear dependency on input

$$\frac{\mathrm{d}}{\mathrm{d}t} x_d(t) = a_d - b_d x_d(t) + s_d \frac{u(t)}{\gamma_d + u(t)}$$

- $x_d(t), u(t) > 0$, $\theta_d = \{a_d, b_d, s_d, \gamma_d\}$
- $d = 1, ..., ?$
- Place GP prior over $\exp u(t)$

Real World: Gene Expression Data

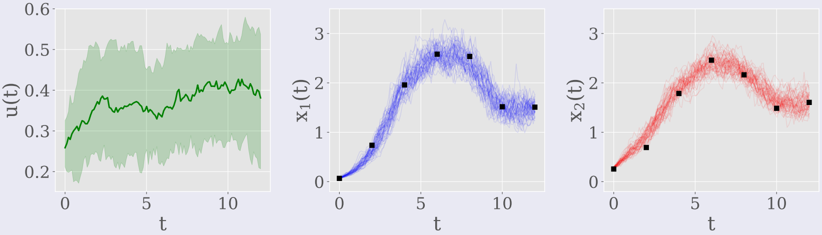
- Multi-output system with non-linear dependency on input

$$\frac{\mathrm{d}}{\mathrm{d}t} x_d(t) = a_d - b_d x_d(t) + s_d \frac{u(t)}{\gamma_d + u(t)}$$

- $x_d(t), u(t) > 0$, $\theta_d = \{a_d, b_d, s_d, \gamma_d\}$
- $d = 1, ..., ?$
- Place GP prior over $\exp u(t)$
- Infer θ_d simultaneously

Real World: Gene Expression Data

Inferred TF concentration and predicted gene expressions for TNFRSF10b (blue) and p26 sesn1 (red)



Contents

Overview

- GP priors on non-linear forced models are non-linear SDEs

Overview

- GP priors on non-linear forced models are non-linear SDEs
- Filtering approaches struggle with joint parameter estimation
- Sequential inference slow for propagating gradients

Overview

- GP priors on non-linear forced models are non-linear SDEs
- Filtering approaches struggle with joint parameter estimation
- Sequential inference slow for propagating gradients
- With inverse autoregressive flows we can batch sample time-series

Overview

- GP priors on non-linear forced models are non-linear SDEs
- Filtering approaches struggle with joint parameter estimation
- Sequential inference slow for propagating gradients
- With inverse autoregressive flows we can batch sample time-series
- Can construct approximate model for joint posterior and infer state, input and parameters by optimising NN weights

Overview

- GP priors on non-linear forced models are non-linear SDEs
- Filtering approaches struggle with joint parameter estimation
- Sequential inference slow for propagating gradients
- With inverse autoregressive flows we can batch sample time-series
- Can construct approximate model for joint posterior and infer state, input and parameters by optimising NN weights Approximation of GPs quantifiably good

Contents

Open Issue: Calculating Steady State Covariance

Solving continuous Lyapunov equation

$$\boldsymbol{J}_D(\boldsymbol{f}, t; \boldsymbol{\theta})\tilde{\boldsymbol{\Sigma}} + \tilde{\boldsymbol{\Sigma}}\boldsymbol{J}_D(\boldsymbol{f}, t; \boldsymbol{\theta})^{\mathrm{T}} = -\boldsymbol{L}\varsigma^2\boldsymbol{L}^{\mathrm{T}},$$

is possible to do for fixed values of $\boldsymbol{f}$, t, and θ using numerical solvers, but hard to do online, so no gradients !

Manually solving is increasingly difficult with increase in dimension: solution is system of $d(d-1)/2$ equations

Open Issue: Dimensionality

- Smoother GPs have more orders of differentiation
- Approximations of infinitely-differentiable covariance functions, e.g. periodic, Gaussian RBF, require series approximation
- State dimension proportional to series threshold

References

M. A. Alvarez, D. Luengo, and N. D. Lawrence. Linear latent force models using Gaussian processes. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2693–2705, 2013.

D. Duvenaud and R. P. Adams. Black-box stochastic variational inference in five lines of Python. In *NIPS Workshop on Black-box Learning and Inference*, 2015.

A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar): 723–773, 2012.

J. Hartikainen, M. Seppänen, and S. Särkkä. State-space inference for non-linear latent force models with application to satellite orbit prediction. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 723–730, 2012.

R. Ranganath, S. Gerrish, and D. Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822, 2014.