Adjoint-aided inference of Gaussian process driven differential equations

Paterne Gahungu[1], Christopher Lanyon[2], Mauricio Alvarez[2], Engineer Bainomugisha[1], Michael Smith[2]
Richard Wilkinson[3]

[1] Department of Computer Science, Makerere University
[2] Department of Computer Science, University of Sheffield
[3] School of Mathematical Sciences, University of Nottingham

February 2022

Project team

Paterne Engineer Mike Mauricio Chris







Funders:

airqo
Breathe Clean

GCRF
Global Challenges
Research Fund

EPSRC
Engineering and Physical Sciences
Research Council

Google
Impact Challenge

Outline

- Motivating example: Air pollution in Kampala
- Inference for linear systems:

$$\mathcal{L}u = f$$

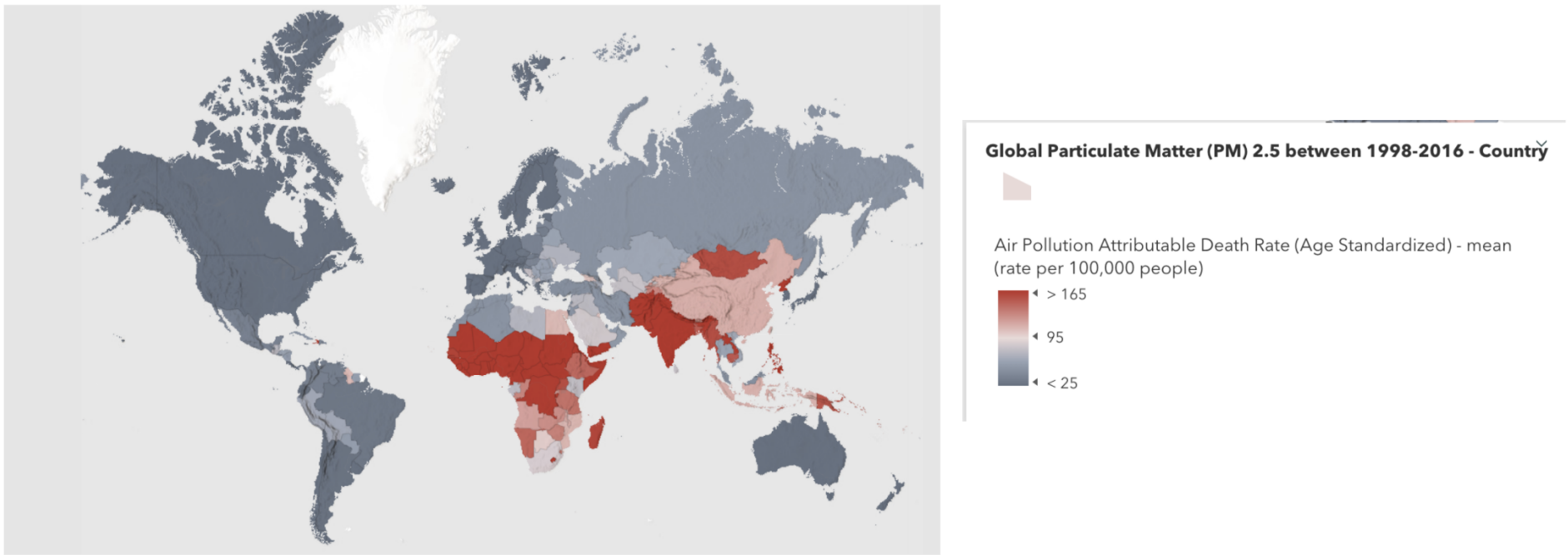
  Given noisy measurements of u can we infer f?
- Adjoints

$$\mathcal{L}^* v \text{ such that } \langle \mathcal{L}u, v \rangle = \langle u, \mathcal{L}^* v \rangle$$

- Examples

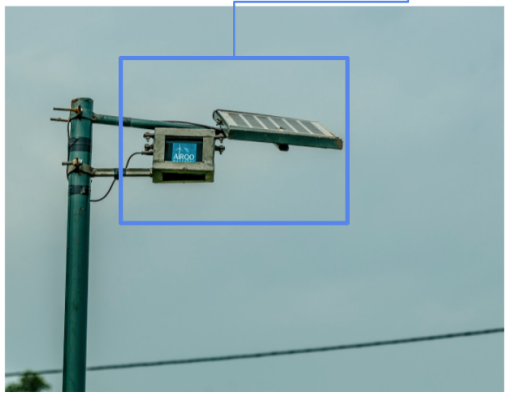
Air pollution

7 million people die every year from exposure to air pollution, the majority in LMICs.



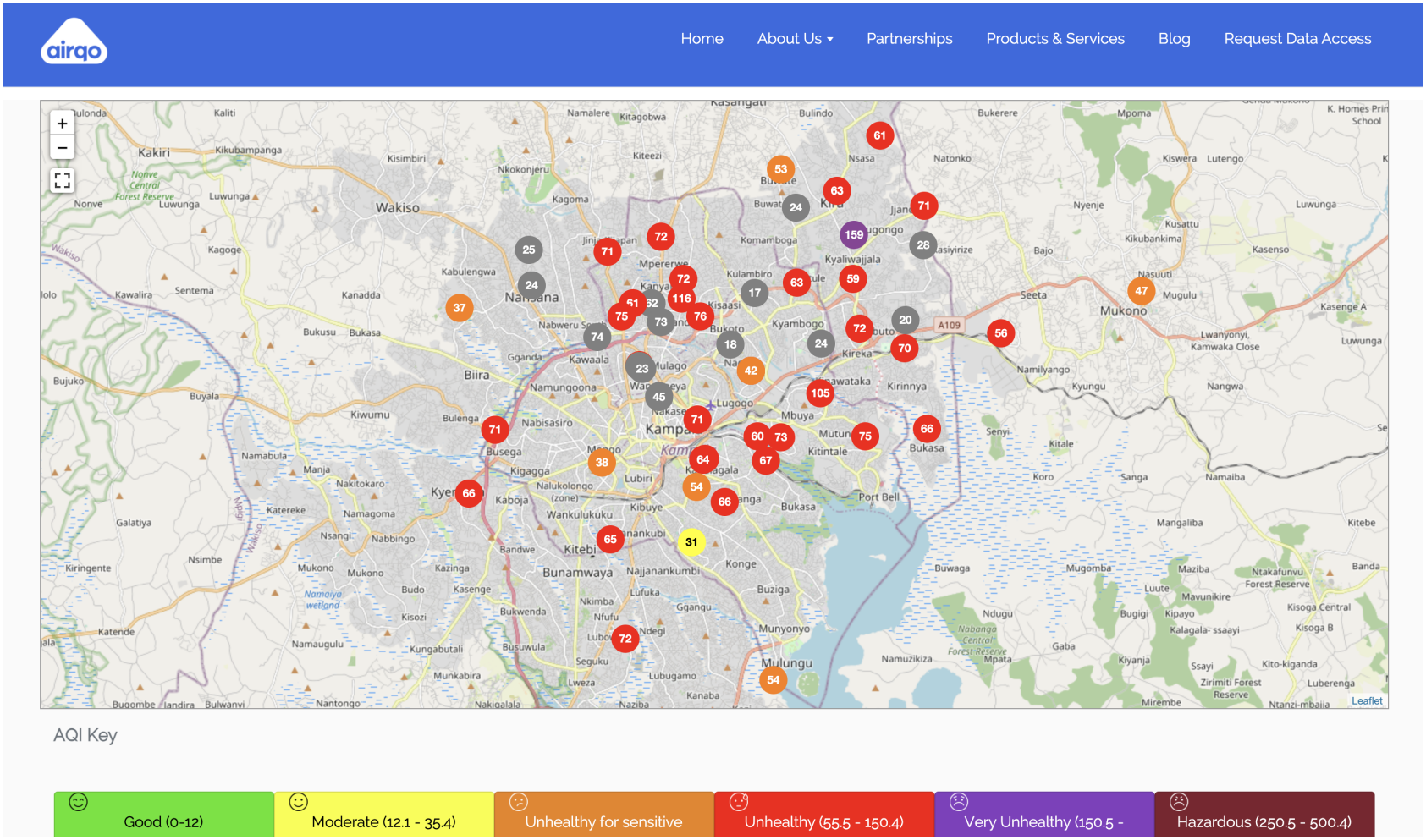
Kampala and AirQo

- AirQo, a portable air quality monitor
- Measures particulate matter
- Solar powered or other available power sources
- Cellular data transmission
- Weather proof for unique African settings



Accurate gravimetric sensors costs \$10,000s.
AirQo have developed cheap (but less accurate) sensors that cost $<$ \$100 and have deployed them around Kampala.
The sensors measure PM2.5 and PM10.

Kampala: PM2.5 levels at 12pm on 23 Feb 2022



Nottingham: 6 $\mu g/m^3$

20 year average for UK is 11 $\mu g/m^3$

Modelling air pollution

Model pollution concentration $u(x, t)$ at location x at time t.

We want to

- infer air pollution (and predict future pollution levels)
- infer pollution sources

Standard non-parametric models (e.g., Gaussian processes) unable to do this.

Modelling air pollution

Model pollution concentration $u(x, t)$ at location x at time t.

We want to

- infer air pollution (and predict future pollution levels)
- infer pollution sources

Standard non-parametric models (e.g., Gaussian processes) unable to do this.

Instead build data models that *know* some physics

$$\frac{\partial u}{\partial t} = \nabla.(\nu u) + \nabla.(D \nabla u) - ru + \sum_i S_i$$

Here

- $S_i(x, t)$ are different pollution sources,
- we may choose to model different pollution types (PM2.5, PM10 etc)
- ν is related to the wind speed, D is the diffusion tensor, and r the reaction rate.

Modelling air pollution

Model pollution concentration $u(x, t)$ at location x at time t.

We want to

- infer air pollution (and predict future pollution levels)
- infer pollution sources

Standard non-parametric models (e.g., Gaussian processes) unable to do this.

Instead build data models that *know* some physics

$$\frac{\partial u}{\partial t} = \nabla.(\nu u) + \nabla.(D \nabla u) - ru + \sum_i S_i$$

Here

- $S_i(x, t)$ are different pollution sources,
- we may choose to model different pollution types (PM2.5, PM10 etc)
- ν is related to the wind speed, D is the diffusion tensor, and r the reaction rate.

Hypothesis: The inclusion of mechanistic behaviour will allow us to infer sources, plan interventions, and predict better.

Computational challenge

Given noisy measurements of pollution levels $z_i = h_i(u) + e_i$.
Can we infer

- the concentration field $u(x, t)$?
- the unknown source terms $S_i(x, t)$?
- the diffusion, advection and reaction parameters? Hyperparameters etc?

Computational challenge

Given noisy measurements of pollution levels $z_i = h_i(u) + e_i$.
Can we infer

- the concentration field $u(x, t)$?
- the unknown source terms $S_i(x, t)$?
- the diffusion, advection and reaction parameters? Hyperparameters etc?

We will use Gaussian process priors for $S_i(x, t)$

$$S_i \sim GP(m_i(\cdot), k_i(\cdot, \cdot))$$

where we carefully choose each prior mean and covariance function:

- Industrial regions
- Major roads and power stations
- Varying affluence levels between regions (related to paving of roads, burning of garbage, cooking on solid fuel stoves etc).

General linear systems

$$\mathcal{L}u = f$$

Linear systems with unknown parameters

Consider

$$\mathcal{L}_p u = f$$

where

- $\mathcal{L}_p$ = linear operator with non-linear dependence upon parameters p.
- f = forcing function.
- u is the quantity being modelled, e.g. pollution concentration.

Finding u given p and f is the **forward problem**.

Linear systems with unknown parameters

Consider

$$\mathcal{L}_p u = f$$

where

- $\mathcal{L}_p$ = linear operator with non-linear dependence upon parameters p.
- f = forcing function.
- u is the quantity being modelled, e.g. pollution concentration.

Finding u given p and f is the **forward problem**.

Inverse problem: infer u, f, p given noisy observations of u

$$z = h(u) + N(0, \Sigma).$$

Note: MCMC likely to be prohibitively expensive: each iteration requires a solution of the forward problem.

Linear systems with unknown parameters

Least squares/maximum-likelihood estimation:

$$\begin{aligned} \min_{p,f} \quad & (z - h(u))^\top (z - h(u)) \\ \text{subject to} \quad & \mathcal{L}_p u = f. \end{aligned}$$

Bayes: find

$$\pi(p, f|z).$$

Linear systems with unknown parameters

Least squares/maximum-likelihood estimation:

$$\begin{aligned} \min_{p,f} \quad & (z - h(u))^\top (z - h(u)) \\ \text{subject to} \quad & \mathcal{L}_p u = f. \end{aligned}$$

Bayes: find

$$\pi(p, f|z).$$

In both cases it would be useful to marginalize parameters, and compute derivatives with respect to the parameters.

Linear systems with unknown parameters

Least squares/maximum-likelihood estimation:

$$\begin{aligned} \min_{p,f} \quad & (z - h(u))^\top (z - h(u)) \\ \text{subject to} \quad & \mathcal{L}_p u = f. \end{aligned}$$

Bayes: find

$$\pi(p, f|z).$$

In both cases it would be useful to marginalize parameters, and compute derivatives with respect to the parameters.

- Adjoints can help!

What is an adjoint?

See Estep 2004

Let $\mathcal{L} : \mathcal{U} \mapsto \mathcal{V}$ be a linear operator between Banach spaces, and let $\mathcal{U}^*$ be the dual space of $\mathcal{U}$: the space of bounded linear functionals on $\mathcal{U}$.

Consider $v^* \in \mathcal{V}^*$ and define $F : \mathcal{U} \to \mathbb{R}$ by

$$F : u \mapsto v^*(\mathcal{L}(u)).$$

What is an adjoint?

See Estep 2004

Let $\mathcal{L} : \mathcal{U} \mapsto \mathcal{V}$ be a linear operator between Banach spaces, and let $\mathcal{U}^*$ be the dual space of $\mathcal{U}$: the space of bounded linear functionals on $\mathcal{U}$.

Consider $v^* \in \mathcal{V}^*$ and define $F : \mathcal{U} \to \mathbb{R}$ by

$$F : u \mapsto v^*(\mathcal{L}(u)).$$

Then F is a bounded linear functional on $\mathcal{U}$, i.e. $F = u^*$ for some $u^* \in \mathcal{U}^*$.

Thus for all $v^* \in \mathcal{V}^*$ we've associated a unique $u^* \in \mathcal{U}^*$.

What is an adjoint?

See Estep 2004

Let $\mathcal{L} : \mathcal{U} \mapsto \mathcal{V}$ be a linear operator between Banach spaces, and let $\mathcal{U}^*$ be the dual space of $\mathcal{U}$: the space of bounded linear functionals on $\mathcal{U}$.

Consider $v^* \in \mathcal{V}^*$ and define $F : \mathcal{U} \to \mathbb{R}$ by

$$F : u \mapsto v^*(\mathcal{L}(u)).$$

Then F is a bounded linear functional on $\mathcal{U}$, i.e. $F = u^*$ for some $u^* \in \mathcal{U}^*$.

Thus for all $v^* \in \mathcal{V}^*$ we've associated a unique $u^* \in \mathcal{U}^*$.

$$\mathcal{L}^* : v^* \mapsto u^*.$$

$\mathcal{L}^*$ is the **adjoint** of $\mathcal{L}$, and is itself a bounded linear operator.

What is an adjoint?

See Estep 2004

Let $\mathcal{L} : \mathcal{U} \mapsto \mathcal{V}$ be a linear operator between Banach spaces, and let $\mathcal{U}^*$ be the dual space of $\mathcal{U}$: the space of bounded linear functionals on $\mathcal{U}$.

Consider $v^* \in \mathcal{V}^*$ and define $F : \mathcal{U} \to \mathbb{R}$ by

$$F : u \mapsto v^*(\mathcal{L}(u)).$$

Then F is a bounded linear functional on $\mathcal{U}$, i.e. $F = u^*$ for some $u^* \in \mathcal{U}^*$.

Thus for all $v^* \in \mathcal{V}^*$ we've associated a unique $u^* \in \mathcal{U}^*$.

$$\mathcal{L}^* : v^* \mapsto u^*.$$

$\mathcal{L}^*$ is the **adjoint** of $\mathcal{L}$, and is itself a bounded linear operator.
By definition

$$v^*(\mathcal{L}(u)) = \mathcal{L}^* v^*(u)$$

which is known as the **bilinear identity**.

Adjoints in Hilbert space

See Estep 2004

$$\mathcal{L}^* : v^* \mapsto u^*.$$
$$v^*(\mathcal{L}(u)) = \mathcal{L}^* v^*(u)$$

When $\mathcal{U}$ and $\mathcal{V}$ are Hilbert spaces, then we can identify them with their dual space:

- by the Riesz representation theorem if $v^* \in \mathcal{V}^*$ there exists $v \in \mathcal{V}$ such that $v^* = \langle \cdot, v \rangle_{\mathcal{V}}$ (and vice versa...).

Adjoints in Hilbert space

See Estep 2004

$$\mathcal{L}^* : v^* \mapsto u^*.$$
$$v^*(\mathcal{L}(u)) = \mathcal{L}^* v^*(u)$$

When $\mathcal{U}$ and $\mathcal{V}$ are Hilbert spaces, then we can identify them with their dual space:

- by the Riesz representation theorem if $v^* \in \mathcal{V}^*$ there exists $v \in \mathcal{V}$ such that $v^* = \langle \cdot, v \rangle_{\mathcal{V}}$ (and vice versa...).

In this case, the **bilinear identity** reduces to

$$\langle \mathcal{L}u, v \rangle_{\mathcal{V}} = v^*(\mathcal{L}(u)) = \mathcal{L}^* v^*(u) = \langle u, \mathcal{L}^* v \rangle_{\mathcal{U}}.$$

where we now consider $\mathcal{L}^* : \mathcal{V} \to \mathcal{U}$.

Example 1: Ordinary differential equation

Consider the ordinary differential equation

$$-D\ddot{u} + \nu\dot{u} + u = f(t) \quad \text{with } u(0) = \dot{u}(0) = 0.$$

Example 1: Ordinary differential equation

Consider the ordinary differential equation

$$-D\ddot{u} + \nu\dot{u} + u = f(t) \quad \text{with } u(0) = \dot{u}(0) = 0.$$

Use the bilinear identity to find the adjoint of

$$\mathcal{L}u = (-D\frac{\mathrm{d}^2}{\mathrm{d}t^2} + \nu\frac{\mathrm{d}}{\mathrm{d}t} + 1)u \quad \text{with } u(0) = \dot{u}(0) = 0$$

Example 1: Ordinary differential equation

Consider the ordinary differential equation

$$-D\ddot{u} + \nu\dot{u} + u = f(t) \quad \text{with } u(0) = \dot{u}(0) = 0.$$

Use the bilinear identity to find the adjoint of

$$\mathcal{L}u = (-D\frac{\mathrm{d}^2}{\mathrm{d}t^2} + \nu\frac{\mathrm{d}}{\mathrm{d}t} + 1)u \quad \text{with } u(0) = \dot{u}(0) = 0$$

$$\begin{aligned}\langle \mathcal{L}u, v\rangle &= \int_0^T \mathcal{L}u(t)v(t)\mathrm{d}t = \int_0^T (-D\ddot{u} + \nu\dot{u} + u)v\mathrm{d}t \\ &= [-D\dot{u}v]_0^T + \int_0^T D\dot{u}\dot{v}\mathrm{d}t + [\nu uv]_0^T - \int_0^T \nu u\dot{v}dt + \int_0^T uv\mathrm{d}t\end{aligned}$$

Example 1: Ordinary differential equation

Consider the ordinary differential equation

$$-D\ddot{u} + \nu\dot{u} + u = f(t) \quad \text{with } u(0) = \dot{u}(0) = 0.$$

Use the bilinear identity to find the adjoint of

$$\mathcal{L}u = (-D\frac{\mathrm{d}^2}{\mathrm{d}t^2} + \nu\frac{\mathrm{d}}{\mathrm{d}t} + 1)u \quad \text{with } u(0) = \dot{u}(0) = 0$$

$$\begin{aligned}
\langle \mathcal{L}u, v\rangle &= \int_0^T \mathcal{L}u(t)v(t)\mathrm{d}t = \int_0^T (-D\ddot{u} + \nu\dot{u} + u)v\mathrm{d}t \\
&= [-D\dot{u}v]_0^T + \int_0^T D\dot{u}\dot{v}\mathrm{d}t + [\nu uv]_0^T - \int_0^T \nu u\dot{v}dt + \int_0^T uv\mathrm{d}t \\
&= [-Du\dot{v}]_0^T - \int_0^T Du\ddot{v}\mathrm{d}t - \int_0^T \nu u\dot{v}dt + \int_0^T uv\mathrm{d}t \\
&= \int_0^T (-D\ddot{v} - \nu\dot{v} + v)u\mathrm{d}t \quad \text{when } v(T) = \dot{v}(T) = 0 \\
&= \langle u, \mathcal{L}^* v\rangle
\end{aligned}$$

So the linear operator

$$\mathcal{L}u = (-D\frac{\mathrm{d}^2}{\mathrm{d}t^2} + \nu\frac{\mathrm{d}}{\mathrm{d}t} + 1)u \quad \text{with } u(0) = \dot{u}(0) = 0$$

has adjoint operator

$$\mathcal{L}^* v = (-D\frac{\mathrm{d}^2}{\mathrm{d}t^2} - \nu\frac{\mathrm{d}}{\mathrm{d}t} + 1)v \quad \text{with } v(T) = \dot{v}(T) = 0$$

Note that initial conditions on the original system translated to final conditions on the adjoint system.

Benefits of adjoints

$$\begin{aligned} \min_{p,f} S(p,f) = \quad & (z - h(u))^\top (z - h(u)) \\ \text{subject to} \quad & \mathcal{L}_p u = f\,. \end{aligned}$$

1. If $f \equiv f_q$ depends linearly on some parameters q we can easily compute the least squares estimator

$$\hat{q}(p) = \arg\min_q S(p, f_q)$$

Benefits of adjoints

$$\begin{aligned} \min_{p,f} S(p,f) = \quad & (z - h(u))^\top (z - h(u)) \\ \text{subject to} \quad & \mathcal{L}_p u = f. \end{aligned}$$

1. If $f \equiv f_q$ depends linearly on some parameters q we can easily compute the least squares estimator

$$\hat{q}(p) = \arg\min_q S(p, f_q)$$

   - If $z = h(u) + N(0, \Sigma)$, and $q \sim N(m, C)$ a priori, then

$$q \mid z, p = N(m^*, C^*)$$

Benefits of adjoints

$$\begin{aligned} \min_{p,f} S(p,f) = \quad & (z - h(u))^\top (z - h(u)) \\ \text{subject to} \quad & \mathcal{L}_p u = f. \end{aligned}$$

1. If $f \equiv f_q$ depends linearly on some parameters q we can easily compute the least squares estimator

$$\hat{q}(p) = \arg\min_q S(p, f_q)$$

   - If $z = h(u) + N(0, \Sigma)$, and $q \sim N(m, C)$ a priori, then

$$q \mid z, p = N(m^*, C^*)$$

2. We can compute $\frac{\mathrm{d}S}{\mathrm{d}p}(p, f_q)$ and approximate $\frac{\mathrm{d}S}{\mathrm{d}p}(p, f_{\hat{q}(p)})$

Benefits of adjoints

$$\begin{aligned} \min_{p,f} S(p,f) = \quad & (z - h(u))^\top (z - h(u)) \\ \text{subject to} \quad & \mathcal{L}_p u = f. \end{aligned}$$

1. If $f \equiv f_q$ depends linearly on some parameters q we can easily compute the least squares estimator

$$\hat{q}(p) = \arg\min_q S(p, f_q)$$

   - If $z = h(u) + N(0, \Sigma)$, and $q \sim N(m, C)$ a priori, then

$$q \mid z, p = N(m^*, C^*)$$

2. We can compute $\frac{\mathrm{d}S}{\mathrm{d}p}(p, f_q)$ and approximate $\frac{\mathrm{d}S}{\mathrm{d}p}(p, f_{\hat{q}(p)})$

This may allow for efficient inference of p and f

Efficient inference for q

Suppose

$$f(\cdot) = \sum_{m=1}^{M} q_m \phi_m(\cdot). \tag{1}$$

When $\mathcal{U}$ and $\mathcal{V}$ are spaces of functions on $\mathcal{X}$, the ϕ_m will also be functions on $\mathcal{X}$. In the finite-dimensional case, the ϕ_m will be vectors of length n.

Efficient inference for q

Suppose

$$f(\cdot) = \sum_{m=1}^{M} q_m \phi_m(\cdot). \tag{1}$$

When $\mathcal{U}$ and $\mathcal{V}$ are spaces of functions on $\mathcal{X}$, the ϕ_m will also be functions on $\mathcal{X}$. In the finite-dimensional case, the ϕ_m will be vectors of length n. If the observation operator is linear

$$h_i(u) = \langle h_i, u \rangle$$

we can consider the n adjoint systems

$$\mathcal{L}_p^* v_i = h_i \text{ for } i = 1, \ldots, n.$$

Efficient inference for q

Suppose

$$f(\cdot) = \sum_{m=1}^{M} q_m \phi_m(\cdot). \tag{1}$$

When $\mathcal{U}$ and $\mathcal{V}$ are spaces of functions on $\mathcal{X}$, the ϕ_m will also be functions on $\mathcal{X}$. In the finite-dimensional case, the ϕ_m will be vectors of length n. If the observation operator is linear

$$h_i(u) = \langle h_i, u \rangle$$

we can consider the n adjoint systems

$$\mathcal{L}_p^* v_i = h_i \text{ for } i = 1, \ldots, n.$$

Then

$$\begin{aligned} \langle h_i, u \rangle &= \langle \mathcal{L}_p^* v_i, u \rangle = \langle v_i, \mathcal{L}_p u \rangle \\ &= \langle v_i, f \rangle, \end{aligned}$$

by the bilinear identity.

The i^{th} observation is the inner product between the unknown forcing function f and the solution of the i^{th} adjoint system.

$$z_i = h_i(u) + e_i = \langle v_i, f \rangle + e_i \quad \text{where} \quad \mathcal{L}_p^* v_i = h_i$$

The i^{th} observation is the inner product between the unknown forcing function f and the solution of the i^{th} adjoint system.

$$z_i = h_i(u) + e_i = \langle v_i, f \rangle + e_i \quad \text{where} \quad \mathcal{L}_p^* v_i = h_i$$

This doesn't appear to have helped.

- To evaluate the likelihood (or sum of squares) we have gone from needing a single forward solve, to n adjoint solves: an n-fold increase in computational cost!

The i^{th} observation is the inner product between the unknown forcing function f and the solution of the i^{th} adjoint system.

$$z_i = h_i(u) + e_i = \langle v_i, f \rangle + e_i \quad \text{where} \quad \mathcal{L}_p^* v_i = h_i$$

This doesn't appear to have helped.

- To evaluate the likelihood (or sum of squares) we have gone from needing a single forward solve, to n adjoint solves: an n-fold increase in computational cost!

The benefit arises if there is a linear dependence upon the parameters:

$$h_i(u) = \langle v_i, \sum_{m=1}^{M} q_m \phi_m \rangle = \sum_{m=1}^{M} q_m \langle v_i, \phi_m \rangle.$$

This is a linear model!

The complete observation vector z can then be written as

$$
\begin{aligned}
z &= \begin{pmatrix} \langle v_1, \phi_1 \rangle & \dots & \langle v_1, \phi_M \rangle \\ \vdots & & \vdots \\ \langle v_n, \phi_1 \rangle & \dots & \langle v_n, \phi_M \rangle \end{pmatrix} \begin{pmatrix} q_1 \\ \\ q_M \end{pmatrix} + e \\
&= \Phi q + e
\end{aligned} \tag{2}
$$

The complete observation vector z can then be written as

$$
\begin{aligned}
z &= \begin{pmatrix} \langle v_1, \phi_1 \rangle & \dots & \langle v_1, \phi_M \rangle \\ \vdots & & \vdots \\ \langle v_n, \phi_1 \rangle & \dots & \langle v_n, \phi_M \rangle \end{pmatrix} \begin{pmatrix} q_1 \\ \\ q_M \end{pmatrix} + e \\
&= \Phi q + e
\end{aligned} \tag{2}
$$

Thus, the solution of

$$
\begin{aligned}
\min_{q} \quad & S(q) = (z - h(u))^\top (z - h(u)) \\
\text{subject to} \quad & \mathcal{L}_p u = f_q
\end{aligned}
$$

is obtained at

$$
\hat{q} = (\Phi^\top \Phi)^{-1} \Phi^\top z
$$

with $\mathbb{V}\text{ar}(\hat{q}) = \sigma^2 (\Phi^\top \Phi)^{-1}$ when e_i are uncorrelated and homoscedastic with variance σ^2.

In a Bayesian setting, if we assume *a priori* that $q \sim \mathcal{N}_M(\mu_0, \Sigma_0)$, then the posterior for q given z (and other parameters) is

$$q \mid z \sim \mathcal{N}_M(\mu_n, \Sigma_n) \tag{3}$$

where

$$\mu_n = \Sigma_n(\frac{1}{\sigma^2}\Phi^\top z + \Sigma_0^{-1}\mu_0), \quad \Sigma_n = \left(\frac{1}{\sigma^2}\Phi^\top\Phi + \Sigma_0^{-1}\right)^{-1}. \tag{4}$$

Quick intro to Gaussian Processes

Suppose $f = \{f(x) : x \in \mathcal{X}\}$ is an unknown function.

- use a stochastic process to model our uncertainty about f

Quick intro to Gaussian Processes

Suppose $f = \{f(x) : x \in \mathcal{X}\}$ is an unknown function.

- use a stochastic process to model our uncertainty about f

If the joint distribution of $f(x_1), \ldots, f(x_n)$ is Gaussian, we say f is a Gaussian process (GP).

Quick intro to Gaussian Processes

Suppose $f = \{f(x) : x \in \mathcal{X}\}$ is an unknown function.

- use a stochastic process to model our uncertainty about f

If the joint distribution of $f(x_1), \ldots, f(x_n)$ is Gaussian, we say f is a Gaussian process (GP).

All we need to do is specify the prior mean and covariance functions

$$\mathbb{E}f(x) = m(x), \qquad \mathbb{C}\text{ov}(f(x), f(x')) = k(x, x')$$

We write

$$f \sim GP(m, k).$$

Why use GPs?

- Mathematically attractive
  - Closed under addition

$$f_1, f_2 \sim GP \text{ then } f_1 + f_2 \sim GP$$

Why use GPs?

- Mathematically attractive
  - Closed under addition

$$f_1, f_2 \sim GP \text{ then } f_1 + f_2 \sim GP$$

  - Closed under Bayesian conditioning: if we observe $\mathbf{D} = (f(x_1), \ldots, f(x_n))$ then

$$f|D \sim GP$$

    but with updated mean and covariance functions.

Why use GPs?

- Mathematically attractive
  - Closed under addition

$$f_1, f_2 \sim GP \text{ then } f_1 + f_2 \sim GP$$

  - Closed under Bayesian conditioning: if we observe $\mathbf{D} = (f(x_1), \ldots, f(x_n))$ then

$$f|D \sim GP$$

    but with updated mean and covariance functions.
  - Closed under any linear operator. If $f \sim GP(m(\cdot), k(\cdot, \cdot))$, then $\mathcal{L}$ is a linear operator

$$\mathcal{L} \circ f \sim GP(\mathcal{L} \circ m, \mathcal{L}^2 \circ k)$$

    e.g. $\frac{df}{dx}$, $\int f(x)dx$, Af are all GPs

Why use GPs?

- Mathematically attractive
  - Closed under addition

$$f_1, f_2 \sim GP \text{ then } f_1 + f_2 \sim GP$$

  - Closed under Bayesian conditioning: if we observe $\mathbf{D} = (f(x_1), \ldots, f(x_n))$ then

$$f|D \sim GP$$

    but with updated mean and covariance functions.
  - Closed under any linear operator. If $f \sim GP(m(\cdot), k(\cdot, \cdot))$, then $\mathcal{L}$ is a linear operator

$$\mathcal{L} \circ f \sim GP(\mathcal{L} \circ m, \mathcal{L}^2 \circ k)$$

    e.g. $\frac{df}{dx}$, $\int f(x)dx$, Af are all GPs
- Natural - Best linear unbiased predictors etc

Why use GPs?

- Mathematically attractive
  - Closed under addition

$$f_1, f_2 \sim GP \text{ then } f_1 + f_2 \sim GP$$

  - Closed under Bayesian conditioning: if we observe $\mathbf{D} = (f(x_1), \ldots, f(x_n))$ then

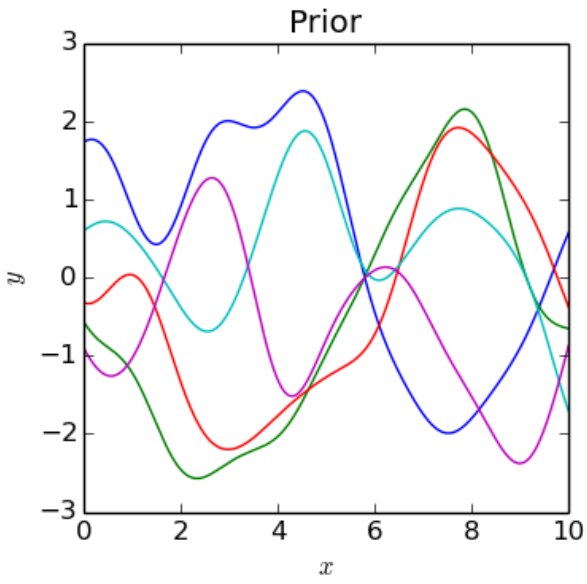
$$f|D \sim GP$$

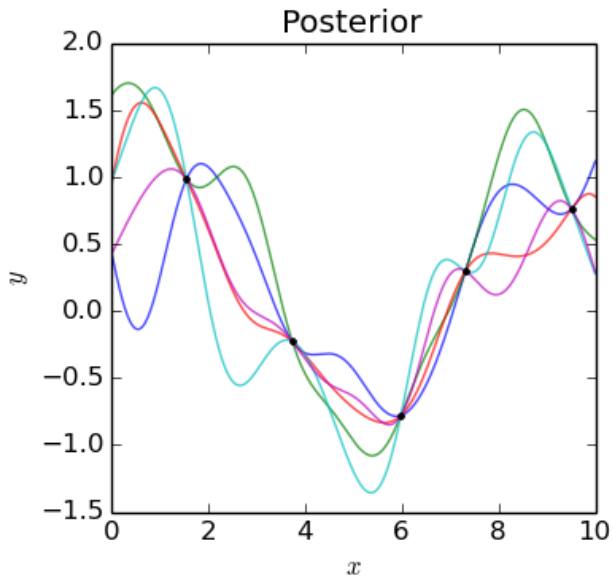
    but with updated mean and covariance functions.
  - Closed under any linear operator. If $f \sim GP(m(\cdot), k(\cdot, \cdot))$, then $\mathcal{L}$ is a linear operator

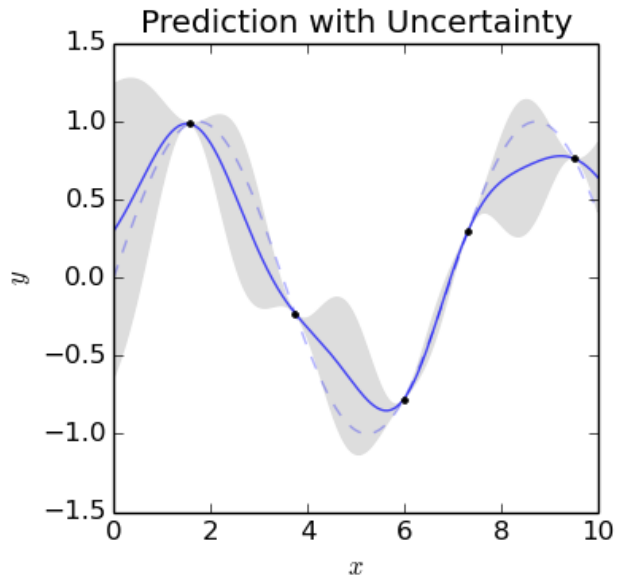
$$\mathcal{L} \circ f \sim GP(\mathcal{L} \circ m, \mathcal{L}^2 \circ k)$$

    e.g. $\frac{df}{dx}$, $\int f(x)dx$, Af are all GPs
- Natural - Best linear unbiased predictors etc
- Relate to other methods such as kernel regression

GP illustration







Parameterizing GPs

$$f(x) \sim GP(m(x), k(x, x')).$$

How can we use GPs within the adjoint framework developed earlier?

Parameterizing GPs

$$f(x) \sim GP(m(x), k(x, x')).$$

How can we use GPs within the adjoint framework developed earlier?
$f \in \mathcal{F}_k$ the RKHS associated with kernel k.

- Let $\{\phi_1(x), \phi_2(x), \ldots\}$ be an orthonormal basis for $\mathcal{F}$.

We can then approximate f using a truncated basis expansion

$$\begin{aligned} f(x) \approx f_q(x) &= \sum_{j=1}^{M} q_i \phi_i(x) \text{ where } \textit{a priori } q_i \sim N(0, \lambda_i^2) \\ &= \Phi \mathbf{q} + e \end{aligned}$$

We've reduced the GP to a linear model.

Choice of basis

- **Mercer basis**: Consider $T_k(f)(\cdot) = \int_{\mathcal{X}} k(x, \cdot) f(x) \mathrm{d}x$. Mercer's theorem gives

$$k(x, x') = \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \phi_i(x')$$

where $\lambda_i, \phi_i(\cdot)$ are eigenpairs of T_k, i.e. $T_k(\phi)(\cdot) = \lambda \phi(\cdot)$
Karhunen-Loève theorem says optimal mean square approximation is

$$\hat{f}(x) = \sum_{i=1}^{M} q_i \sqrt{\lambda_i} \phi_i(x)$$

Choice of basis

- **Mercer basis**: Consider $T_k(f)(\cdot) = \int_{\mathcal{X}} k(x, \cdot) f(x) \mathrm{d}x$. Mercer's theorem gives

$$k(x, x') = \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \phi_i(x')$$

where $\lambda_i, \phi_i(\cdot)$ are eigenpairs of T_k, i.e. $T_k(\phi)(\cdot) = \lambda\phi(\cdot)$
Karhunen-Loève theorem says optimal mean square approximation is

$$\hat{f}(x) = \sum_{i=1}^{M} q_i \sqrt{\lambda_i} \phi_i(x)$$

- **Random Fourier features**: If k stationary, Bochner's theorem:

$$k(x - x') = \int \exp(iw^\top (x - x')) p(w) dw = \mathbb{E}_{w \sim p} \exp(iw^\top (x - x'))$$

$$\approx \frac{1}{M} \sum_{i=1}^{M} (\cos(w_i^\top x),\ \sin(w_i^\top x)) \begin{pmatrix} \cos(w_i^\top x) \\ \sin(w_i^\top x) \end{pmatrix} \text{ if } w_i \sim p(\cdot)$$

$$\hat{f}(x) = \sum_{i=1}^{M} q_i cos(w_i x + b_i)$$

Example 1: ODE continued

$$-D\ddot{u} + \nu\dot{u} + u = f(t)$$

with $u(0) = \dot{u}(0) = 0$ and $f \sim GP$.
The linear operator and adjoint were

$$\mathcal{L}u = (-D\frac{\mathrm{d}^2}{\mathrm{d}t^2} + \nu\frac{\mathrm{d}}{\mathrm{d}t} + 1)u \quad \text{with } u(0) = \dot{u}(0) = 0$$

$$\mathcal{L}^* v = (-D\frac{\mathrm{d}^2}{\mathrm{d}t^2} - \nu\frac{\mathrm{d}}{\mathrm{d}t} + 1)v \quad \text{with } v(T) = \dot{v}(T) = 0$$

Example 1: GP expansion

If we write

$$f(t) = \sum_{j=1}^{M} q_j \phi_j(t) = \Phi \mathbf{q}$$

then given observations

$$\begin{aligned} z_i &= \langle h_i, u \rangle + e_i \\ &= \langle v_i, f \rangle + e_i \\ &= v_i^\top \Phi \mathbf{q} + e_i \end{aligned}$$

Example 1: GP expansion

If we write

$$f(t) = \sum_{j=1}^{M} q_j \phi_j(t) = \Phi \mathbf{q}$$

then given observations

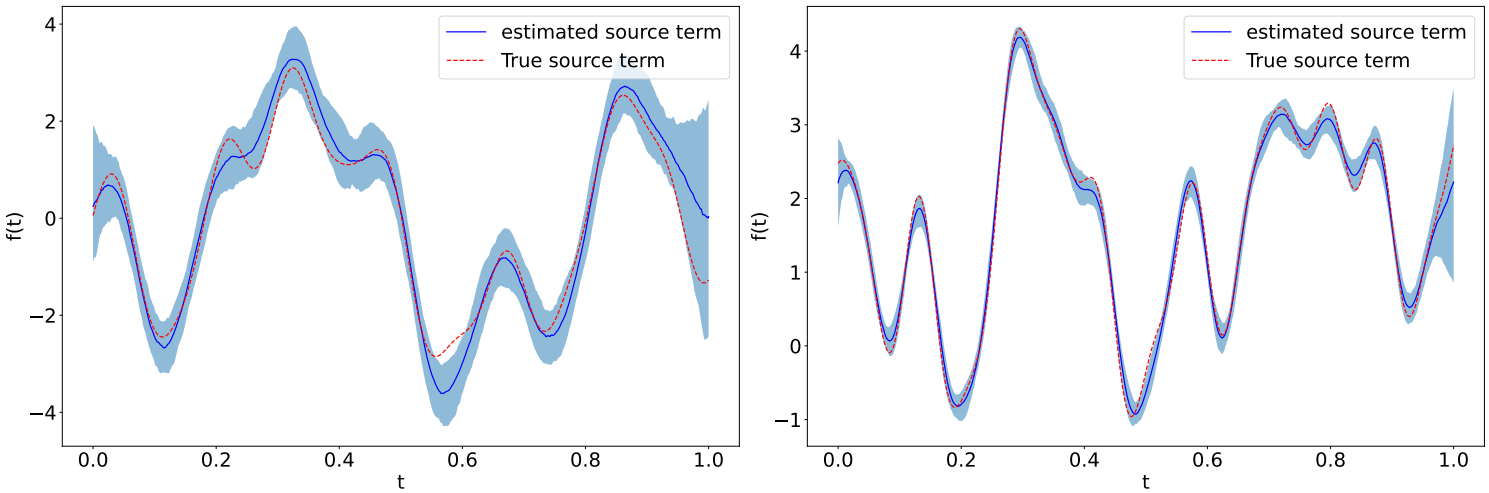
$$\begin{aligned} z_i &= \langle h_i, u \rangle + e_i \\ &= \langle v_i, f \rangle + e_i \\ &= v_i^\top \Phi \mathbf{q} + e_i \end{aligned}$$

Thus we can estimate $\mathbf{q}$ by

$$\hat{\mathbf{q}} = (\Phi^\top V^\top V \Phi)^{-1} \Phi^\top V \mathbf{z}$$

Example 1: Results

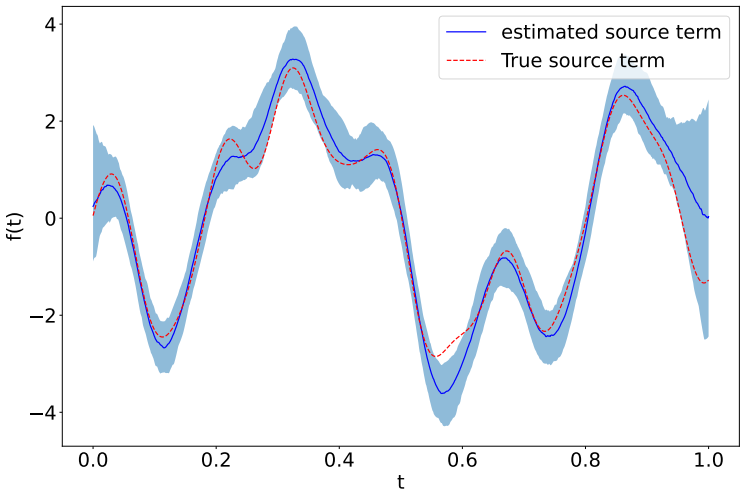
50 and 500 observations, each a noisy average over a short time window (100 Fourier features

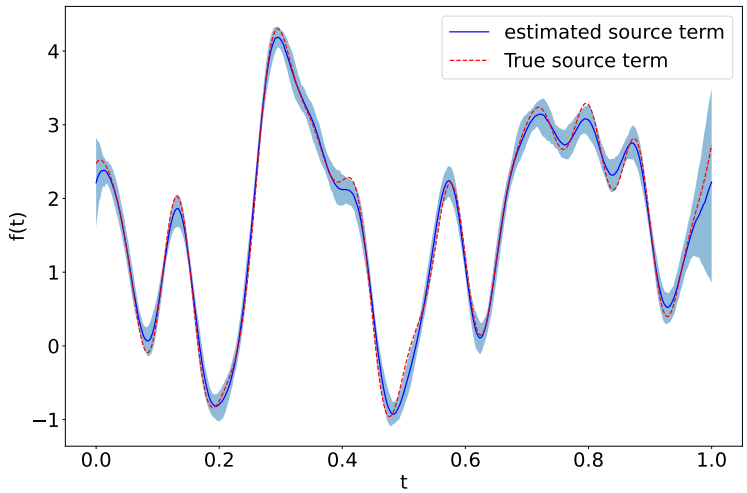


These results require 50 and 500 ODE solves respectively.

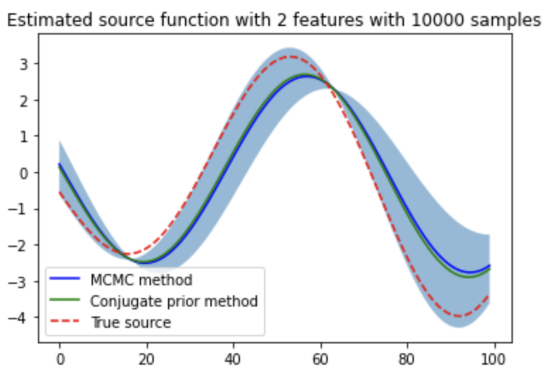
Example 1: Results

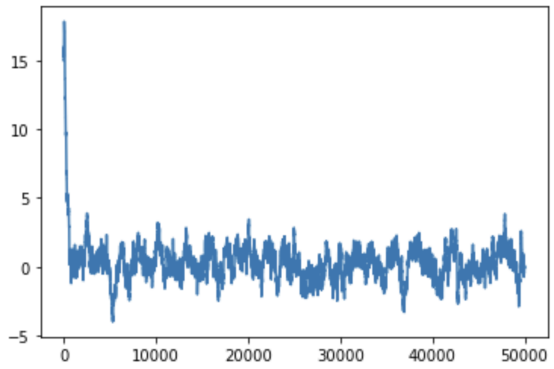
50 and 500 observations, each a noisy average over a short time window (100 Fourier features





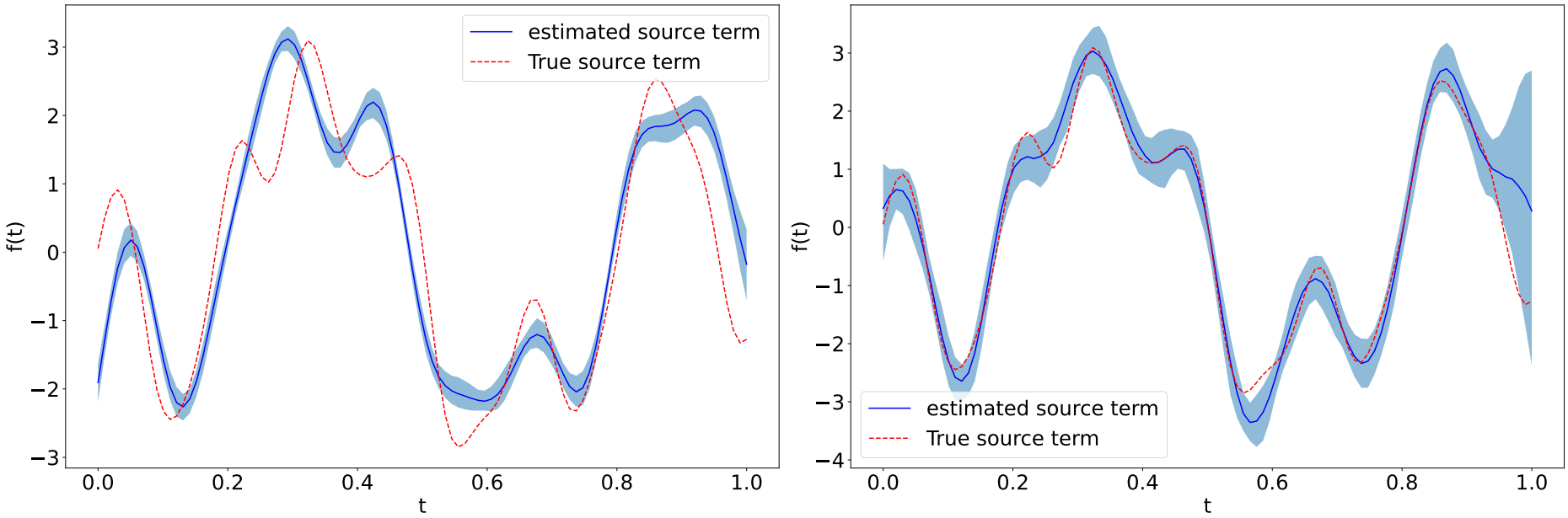
These results require 50 and 500 ODE solves respectively.
MCMC works here for a small number of features. But even with 2 features, we need $\sim 1000s$ of ODE solves.





Example 1: Results

We need to include enough features to have sufficient modelling flexibility.
Left is the posterior with 10 Fourier features, right uses 150 features.



Note the over-confidence in the misspecified model.
The number of Fourier features doesn't have any meaningful effect of the algorithmic complexity.

Example 2: PDE

Advection-diffusion-reaction is a linear operator:

$$\mathcal{L}_p u = \frac{\partial u}{\partial t} - \nabla.(\nu u) - \nabla.(D \nabla u) + ru$$

Forward problem: solve (for some initial and boundary conditions)

$$\mathcal{L}_p u = f_q.$$

Example 2: PDE

Advection-diffusion-reaction is a linear operator:

$$\mathcal{L}_p u = \frac{\partial u}{\partial t} - \nabla.(\nu u) - \nabla.(D \nabla u) + ru$$

Forward problem: solve (for some initial and boundary conditions)

$$\mathcal{L}_p u = f_q.$$

Inverse problem: assume

$$f_q(x,t) \sim GP(m, k_\lambda((x,t),(x',t'))) \approx \sum_{i=1}^{M} q_i \phi_i(x,t) \text{ where } q_i \sim N(0,1)$$

and estimate q, $p = (\nu, D, \lambda)$ given $z_i = \langle h_i, u \rangle + N(0, \sigma)$. Typically h_i will be a sensor function that might average the pollution at a specific location over a short window

$$\langle h_i, u \rangle = \frac{1}{|\mathcal{T}|} \int_{\mathcal{T}} u(x_i, t) dt$$

Example 2: PDE adjoint

For n observations we need n adjoint equations!

$$-\frac{\partial v}{\partial t} - \nu \nabla v - \nabla.(D \nabla v) + rv = h_i \text{ in } \Omega \times (T, 0)$$

along with initial (final) and boundary conditions

Example 2: PDE adjoint

For n observations we need n adjoint equations!

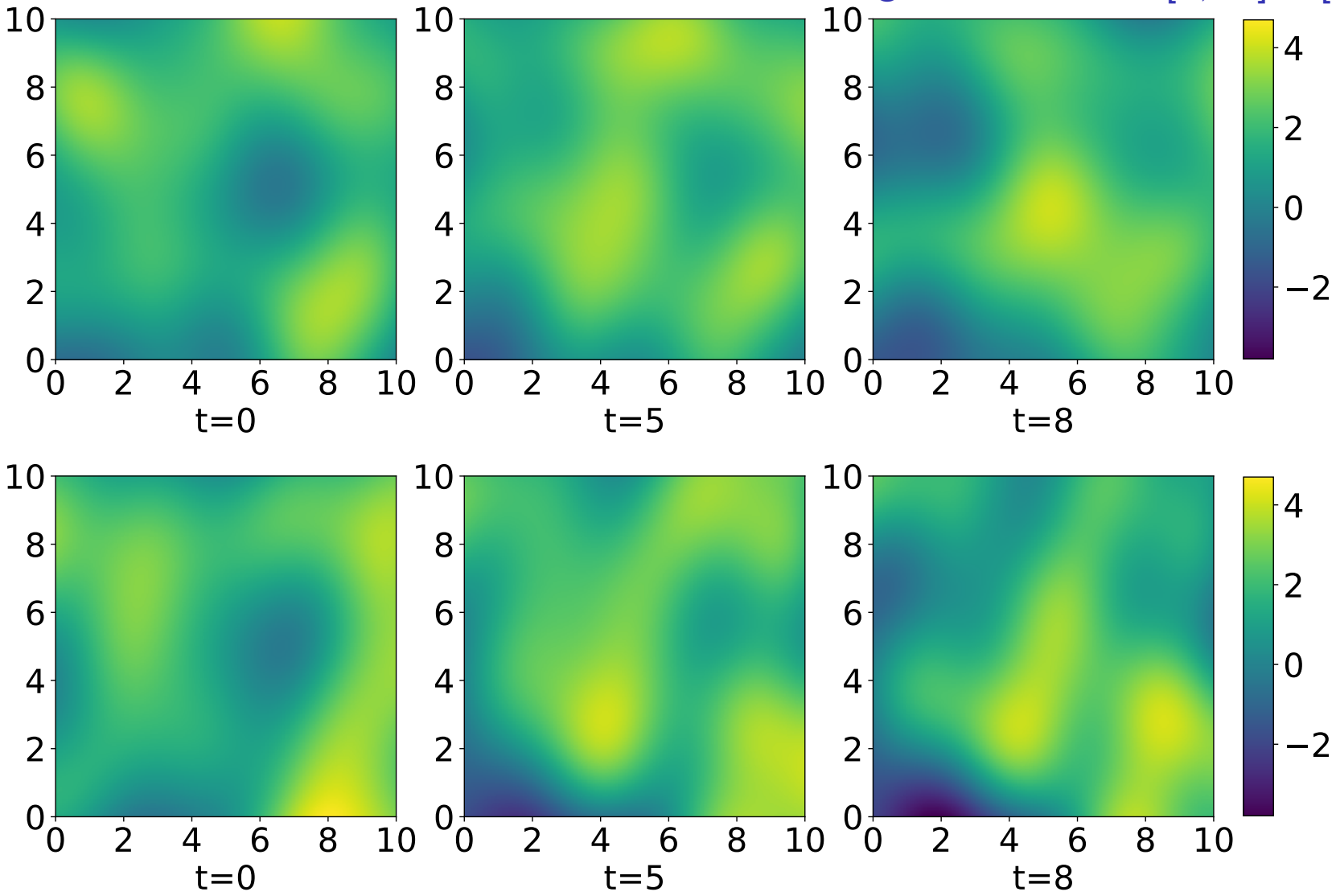
$$-\frac{\partial v}{\partial t} - \nu \nabla v - \nabla.(D \nabla v) + rv = h_i \text{ in } \Omega \times (T, 0)$$

along with initial (final) and boundary conditions

- Initial conditions and boundary conditions can be tricky to compute...
- Numerical issues can arise depending on the discretization vs the sensor function h_i vs diffusion rate etc
- The cost of solving the adjoint is the same as solving the forward problem.

Example 2: Results

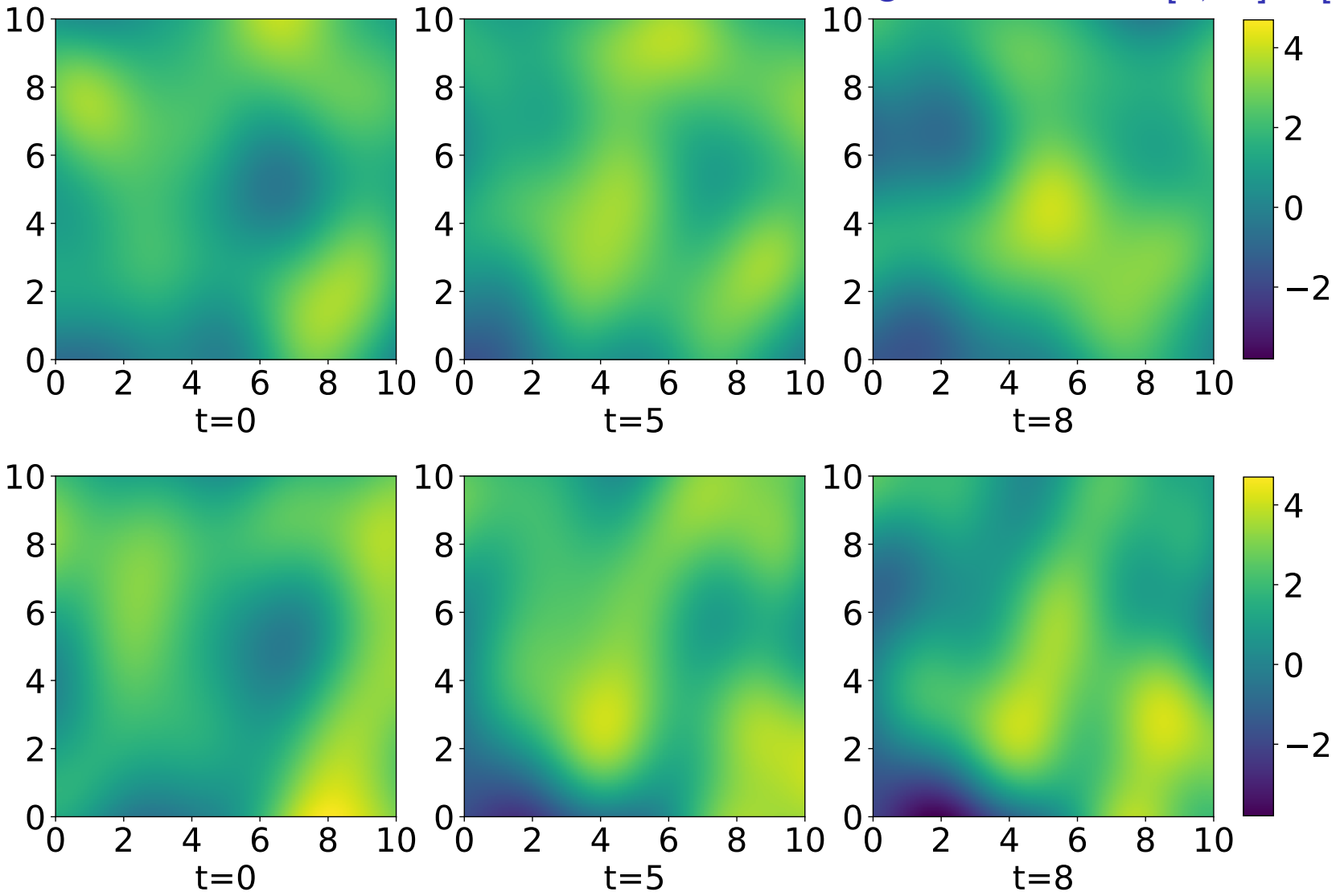
16 sensors, 5 observations from each, noise = 10% signal, Domain = $[0, 10] \times [0, 10]^2$



Top row: truth; bottom: posterior mode

Example 2: Results

16 sensors, 5 observations from each, noise = 10% signal, Domain = $[0, 10] \times [0, 10]^2$

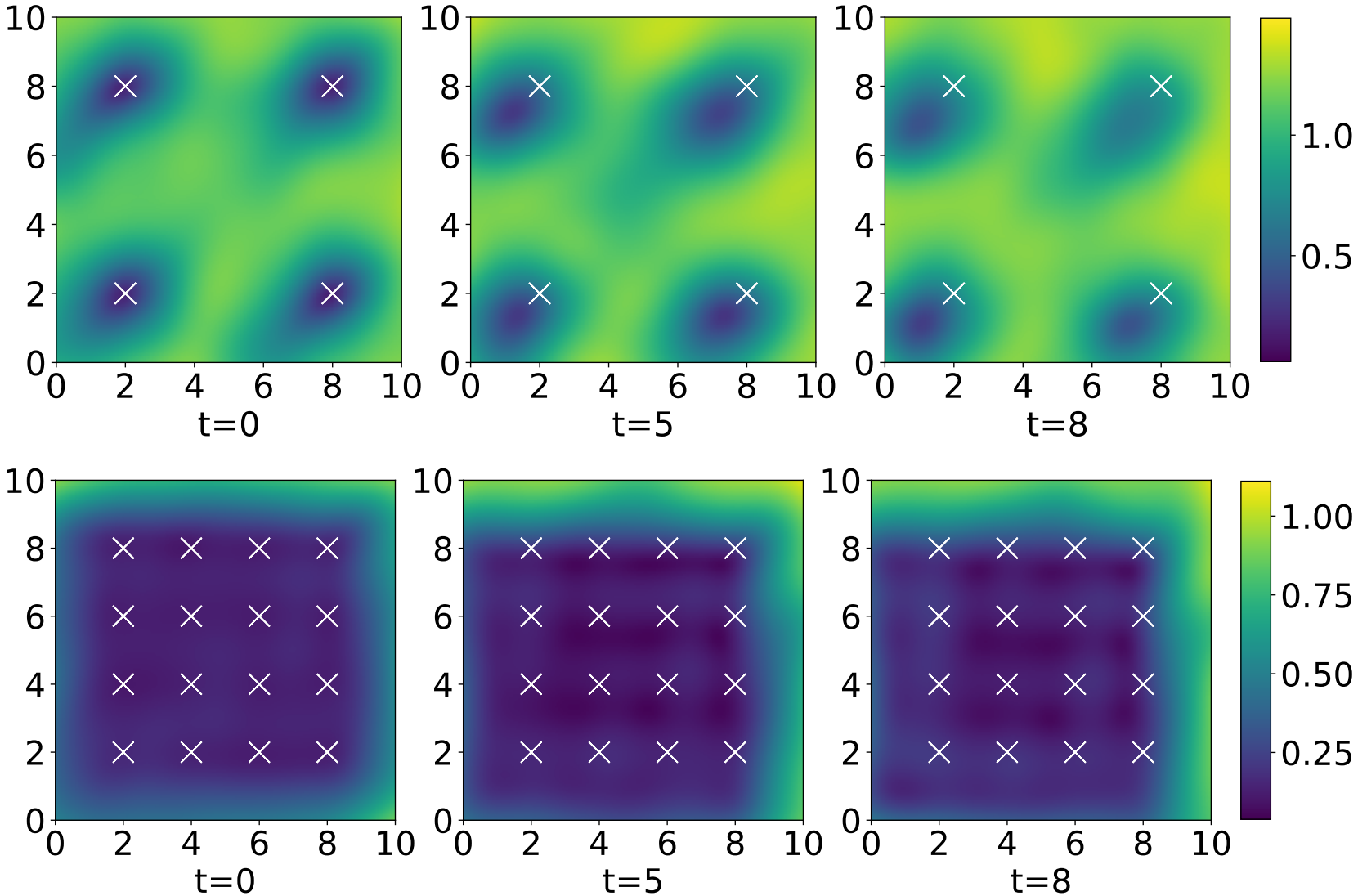


Top row: truth; bottom: posterior mode

Note the negative values....

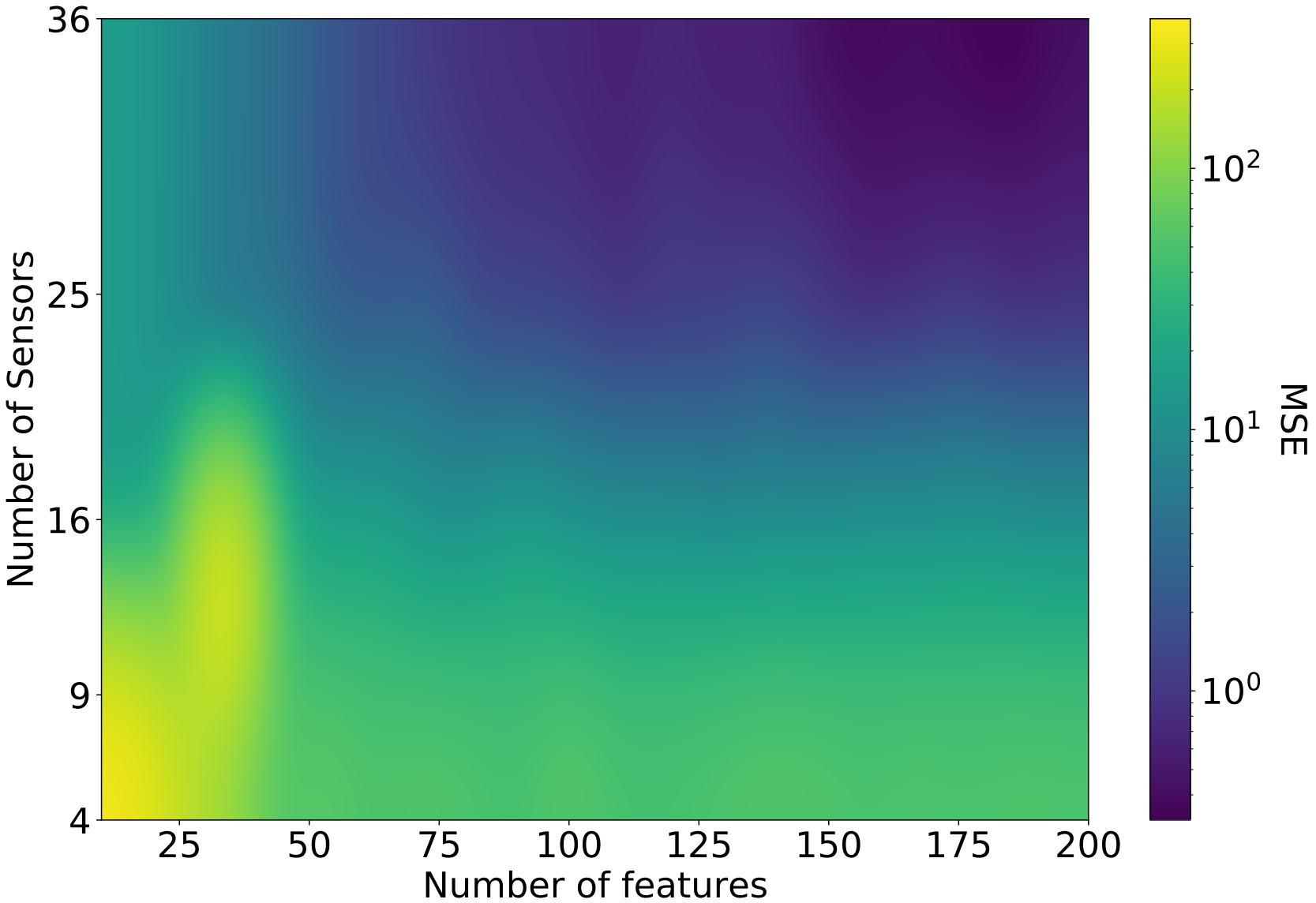
Example 2: Results

Posterior variances, wind from bottom left



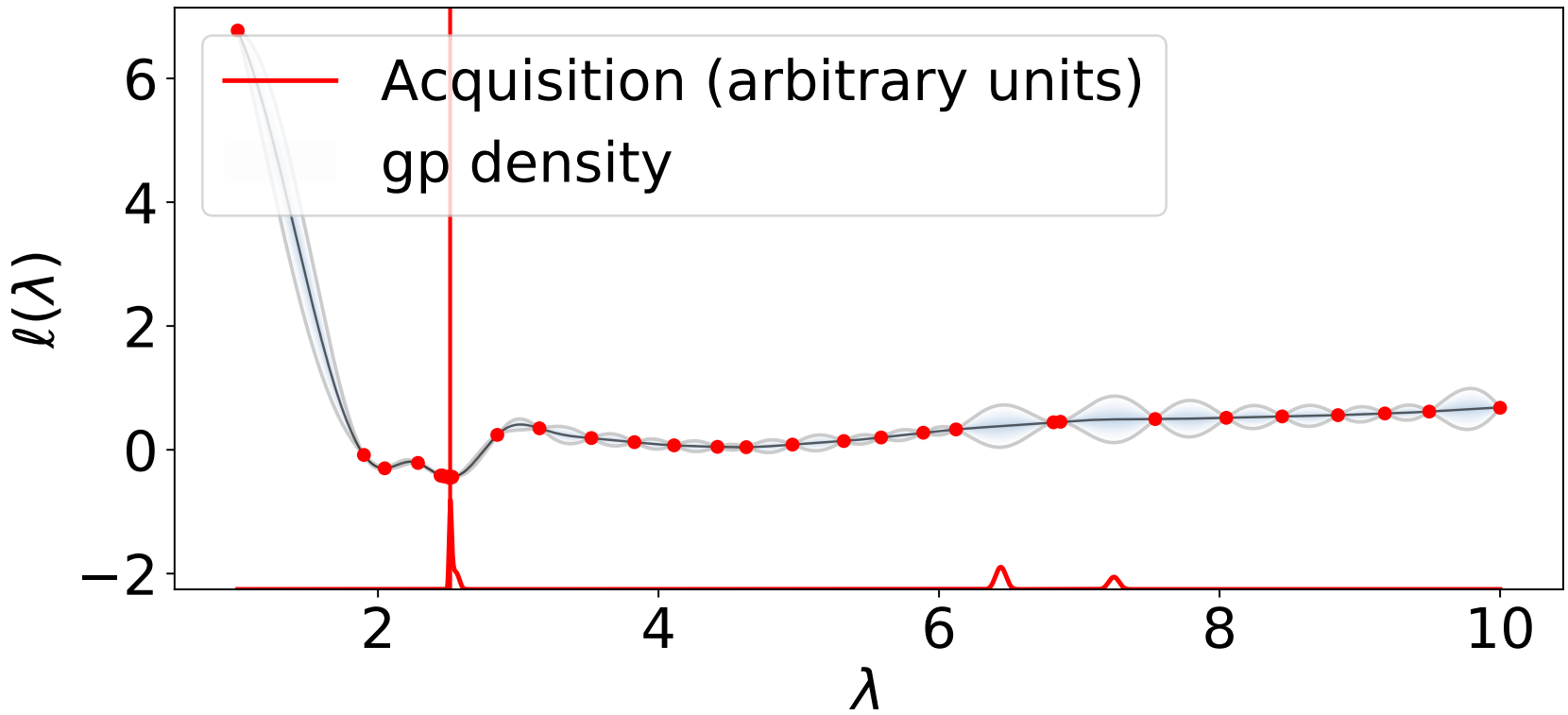
Example 2: Results

Mean square error vs number of features and sensors



Non-linear parameter estimation

A naive way to estimate the non-linear parameters is via Bayesian optimization iteration



Working on ways of using the adjoint sensitivity....

Costs

Adjoint method:

- For the linear forcing/source parameter, we require n solves of the adjoint system to infer the posterior.
- The method is essentially independent of the number of basis functions used.
- The non-linear parameters (GP hyperparameters, PDE parameters) can be inferred in an outer-loop - each step requires a further n adjoint solves (and another n forward solves if we want gradient information).

MCMC:

- All parameters inferred together.
- Hard to say how many iterations will be required, but likely to grow with the the number of parameters (and hence number of GP features).
- Number of iterations required largely independent of n.
- Derivative information generally helps, but this is likely to be unavailable.

Conclusions

Adjoints of linear systems

- an intrusive method; development does require some work...
- Gives numerically stable derivatives
- For linear parametric forcing models, leads to cheap inference
  - May or may not be faster than MCMC depending on the number of data points, and the dimension of the parameter.

GP models that know some physics can improve predictions over vanilla GPs.

- Lots of opportunities for finding efficiencies...
  - Efficient usage of adjoint simulations
  - Multi-level approaches
  - Gradient based optimization
- Preprint at https://arxiv.org/abs/2202.04589

Conclusions

Adjoints of linear systems

- an intrusive method; development does require some work...
- Gives numerically stable derivatives
- For linear parametric forcing models, leads to cheap inference
  - May or may not be faster than MCMC depending on the number of data points, and the dimension of the parameter.

GP models that know some physics can improve predictions over vanilla GPs.

- Lots of opportunities for finding efficiencies...
  - Efficient usage of adjoint simulations
  - Multi-level approaches
  - Gradient based optimization
- Preprint at https://arxiv.org/abs/2202.04589

Thank you for listening!

Example 1: Matrix system

Suppose $X = Y = \mathbb{R}^d$. A linear operator $\mathcal{L}_p : X \to Y$ can be written as

$$\mathcal{L}_p x = A_p x \text{ where } A_p \in \mathbb{R}^d$$

where A_p depends on unknown parameters p.

The **forward problem** is solving the square linear system $A_p x = f$, i.e., $x_{p,q} = A_p^{-1} f$.

Example 1: Matrix system

Suppose $X = Y = \mathbb{R}^d$. A linear operator $\mathcal{L}_p : X \to Y$ can be written as

$$\mathcal{L}_p x = A_p x \text{ where } A_p \in \mathbb{R}^d$$

where A_p depends on unknown parameters p.

The **forward problem** is solving the square linear system $A_p x = f$, i.e., $x_{p,q} = A_p^{-1} f$.
The **adjoint operator** is

$$\mathcal{L}_p^* y = A_p^\top y$$

as we can see that

$$\begin{aligned} \langle A_p x, y \rangle &= (A_p x)^\top y \\ &= x^\top (A^\top y) \\ &= \langle x, A_p^\top y \rangle \end{aligned}$$

Sensitivity

Consider the quantity of interest (QoI)

$$h(x) \equiv \langle g, x \rangle = g^\top x$$

for some $g \in \mathbb{R}^d$, where x is the solution to $h(x, p) := f - Ax = 0$.

We want to compute $\frac{\mathrm{d}g}{\mathrm{d}p}$ (as then we can compute $\frac{\mathrm{d}S}{\mathrm{d}p}(p, q)$)

Sensitivity

Consider the quantity of interest (QoI)

$$h(x) \equiv \langle g, x \rangle = g^\top x$$

for some $g \in \mathbb{R}^d$, where x is the solution to $h(x, p) := f - Ax = 0$.

We want to compute $\frac{\mathrm{d}g}{\mathrm{d}p}$ (as then we can compute $\frac{\mathrm{d}S}{\mathrm{d}p}(p, q)$)

Define Lagrangian the

$$L = g^\top x + y^\top h(x, p)$$

Think of $y \in \mathbb{R}^d$ as Lagrange multipliers.

$$L = g^\top x + y^\top h(x, p)$$

Differentiating with respect to p gives

$$\frac{dL}{dp} = g^\top \frac{dx}{dp} + y^\top \left(\frac{dh}{dx}\frac{dx}{dp} + \frac{dh}{dp}\right)$$

This is true for all y, so if we set $g^\top + y^\top \frac{dh}{dx} = 0$ then we get

$$\begin{aligned}\frac{dL}{dp} = \frac{dg}{dp} &= y^\top \frac{dh}{dp} \\ &= y^\top \left(\frac{df}{dp} - \frac{dA}{dp}x\right)\end{aligned}$$

where $A^\top y = g$

$$L = g^\top x + y^\top h(x, p)$$

Differentiating with respect to p gives

$$\frac{dL}{dp} = g^\top \frac{dx}{dp} + y^\top \left(\frac{dh}{dx}\frac{dx}{dp} + \frac{dh}{dp}\right)$$

This is true for all y, so if we set $g^\top + y^\top \frac{dh}{dx} = 0$ then we get

$$\begin{aligned}\frac{dL}{dp} = \frac{dg}{dp} &= y^\top \frac{dh}{dp} \\ &= y^\top \left(\frac{df}{dp} - \frac{dA}{dp}x\right)\end{aligned}$$

where $A^\top y = g$

This doesn't require $\frac{\mathrm{dx}}{\mathrm{d}p}$, but does need solutions to the forward $Ax = f$ **and** adjoint systems $A^\top y = g$.

$$L = g^\top x + y^\top h(x, p)$$

Differentiating with respect to p gives

$$\frac{dL}{dp} = g^\top \frac{dx}{dp} + y^\top \left(\frac{dh}{dx}\frac{dx}{dp} + \frac{dh}{dp}\right)$$

This is true for all y, so if we set $g^\top + y^\top \frac{dh}{dx} = 0$ then we get

$$\begin{aligned} \frac{dL}{dp} = \frac{dg}{dp} &= y^\top \frac{dh}{dp} \\ &= y^\top \left(\frac{df}{dp} - \frac{dA}{dp}x\right) \end{aligned}$$

where $A^\top y = g$

This doesn't require $\frac{\mathrm{d}x}{\mathrm{d}p}$, but does need solutions to the forward $Ax = f$ **and** adjoint systems $A^\top y = g$.

- Autodiff software (eg TensorFlow, JAX etc) will give us this, but can be unreliable for differential equations with long iterative loops

Non-identifiable linear model

Let

$$A_p = \begin{pmatrix} 2 + p_2^2 & -1 \\ 1 & 1 + p_1^2 \end{pmatrix} \text{ and } f_q = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = q_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + q_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

and suppose we're given 4 observations with

$$G = \begin{pmatrix} 2 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 \end{pmatrix}$$

Non-identifiable linear model

Let

$$A_p = \begin{pmatrix} 2 + p_2^2 & -1 \\ 1 & 1 + p_1^2 \end{pmatrix} \text{ and } f_q = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = q_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + q_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

and suppose we're given 4 observations with

$$G = \begin{pmatrix} 2 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 \end{pmatrix}$$

Given any dataset we can learn q (given p) with a single adjoint solve.

Non-identifiable linear model

Let

$$A_p = \begin{pmatrix} 2 + p_2^2 & -1 \\ 1 & 1 + p_1^2 \end{pmatrix} \text{ and } f_q = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = q_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + q_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

and suppose we're given 4 observations with

$$G = \begin{pmatrix} 2 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 \end{pmatrix}$$

Given any dataset we can learn q (given p) with a single adjoint solve. We can also compute the gradient of $S(p, \hat{q})$ wrt p, but in this case

$$\frac{\mathrm{d}S}{\mathrm{d}p} = 0 \; \forall \; p.$$

and so p is unidentifiable.

Non-identifiable linear model

Let

$$A_p = \begin{pmatrix} 2 + p_2^2 & -1 \\ 1 & 1 + p_1^2 \end{pmatrix} \text{ and } f_q = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = q_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + q_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

and suppose we're given 4 observations with

$$G = \begin{pmatrix} 2 & 1 & 0 & 1 \\ 1 & 2 & 1 & 0 \end{pmatrix}$$

Given any dataset we can learn q (given p) with a single adjoint solve.
We can also compute the gradient of $S(p, \hat{q})$ wrt p, but in this case

$$\frac{\mathrm{d}S}{\mathrm{d}p} = 0 \; \forall \; p.$$

and so p is unidentifiable.
Consider the solution to the unconstrained optimization problem.

$$x^* = \arg\min_x (z - G^\top x)^\top (z - G^\top x)$$

The basis functions used for f form a complete basis for $\mathbb{R}^2$, and we can always find a q so that $A_p x^* = f_q$ (for all p as A_p is invertible).