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

January 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 (Cf. Niklas Wahlström's talk)
- Adjoints
- 3 examples

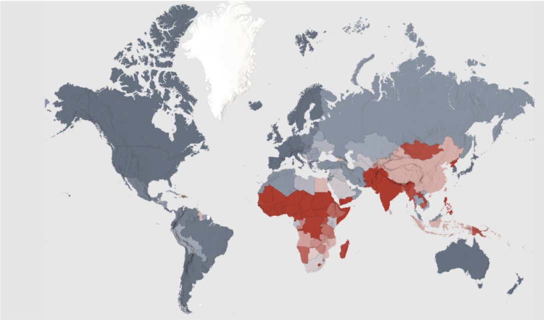
Outline

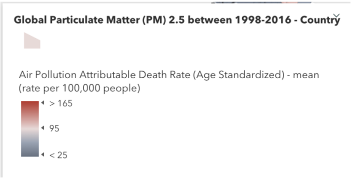
- Motivating example: Air pollution in Kampala
- Inference for linear systems (Cf. Niklas Wahlström's talk)
- Adjoints
- 3 examples

First time for this talk...

Air pollution

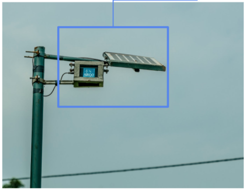
7 million people die every year from exposure to air pollution, the majority in LMICs.





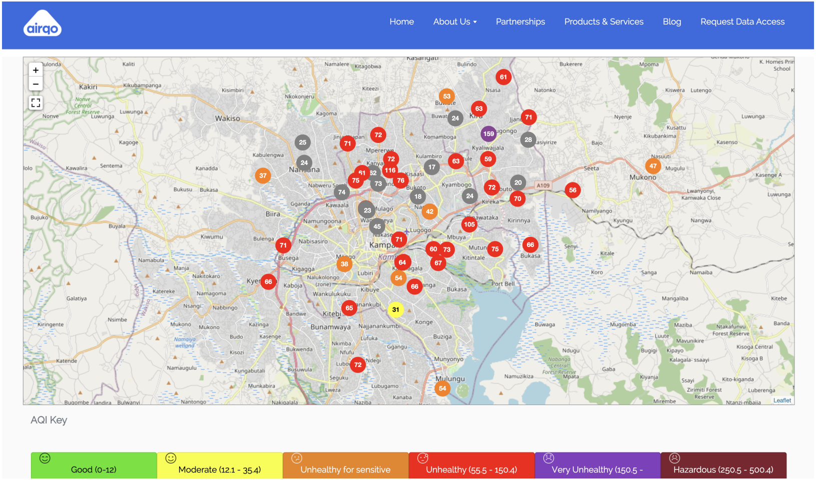
Kampala and AirQo

- 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 4 Jan 2022



Bern: 7 $\mu g/m^3$

Sheffield: 3 $\mu g/m^3$

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

Modelling air pollution

In order to take action, we need to be able to

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

Model pollution concentration $c(x,t)$ as a GP

- with standard kernels we cannot infer the pollution sources.

Modelling air pollution

In order to take action, we need to be able to

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

Model pollution concentration $c(x, t)$ as a GP

- with standard kernels we cannot infer the pollution sources.

Instead build data models that *know* some physics

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

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 and D is the diffusion tensor.

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

Statistical problem

Given noisy measurements of pollution levels $z_i = \int_{t_i^-}^{t_i^+} c(x_i, t)\mathrm{d}t + e_i$.

Can we infer

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

Statistical problem

Given noisy measurements of pollution levels $z_i = \int_{t_i^-}^{t_i^+} c(x_i, t)\mathrm{d}t + e_i$.

Can we infer

- the concentration field $c(x, t)$?
- the unknown source terms $S_i(x, t)$?
- the diffusion and advection 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

$$\mathcal{L}_p x = f_q$$

Linear systems with unknown parameters

Cf. Niklas Wahlström's talk

Consider

$$\mathcal{L}_p x = f_{q,p}$$

where

- $\mathcal{L}_p$ = linear operator with non-linear dependence upon parameters p.
- $f_{q,p}$ = forcing function, which depends **linearly** on parameters q.
- x is the quantity being modelled, e.g. pollution concentration, observed with noise
$$z = g(x) + N(0, \Sigma).$$

Finding x given p and q is the **forward problem**.

Inverse problem: infer x, q, p given z.

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:

$$\min_{p,q} \quad (z - h(x))^\top (z - h(x))$$
$$\text{subject to} \quad \mathcal{L}_p x = f_q.$$

Bayes: find

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

Linear systems with unknown parameters

Least squares/maximum-likelihood estimation:

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

Bayes: find

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

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:

$$\min_{p,q} \quad (z - h(x))^\top (z - h(x))$$
$$\text{subject to} \quad \mathcal{L}_p x = f_q.$$

Bayes: find

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

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

Consider $y^* \in Y^*$ and define $F : X \to \mathbb{R}$ by

$$F : x \mapsto y^*(\mathcal{L}(x)).$$

What is an adjoint?

See Estep 2004

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

Consider $y^* \in Y^*$ and define $F : X \to \mathbb{R}$ by

$$F : x \mapsto y^*(\mathcal{L}(x)).$$

Then F is a bounded linear functional on X, i.e. $F = x^*$ for some $x^* \in X^*$.

Thus for all $y^* \in Y^*$ we've associated a unique $x^* \in X^*$.

What is an adjoint?

See Estep 2004

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

Consider $y^* \in Y^*$ and define $F : X \to \mathbb{R}$ by

$$F : x \mapsto y^*(\mathcal{L}(x)).$$

Then F is a bounded linear functional on X, i.e. $F = x^*$ for some $x^* \in X^*$.

Thus for all $y^* \in Y^*$ we've associated a unique $x^* \in X^*$.

$$\mathcal{L}^* : y^* \mapsto x^*.$$

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

What is an adjoint?

See Estep 2004

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

Consider $y^* \in Y^*$ and define $F : X \to \mathbb{R}$ by

$$F : x \mapsto y^*(\mathcal{L}(x)).$$

Then F is a bounded linear functional on X, i.e. $F = x^*$ for some $x^* \in X^*$.

Thus for all $y^* \in Y^*$ we've associated a unique $x^* \in X^*$.

$$\mathcal{L}^* : y^* \mapsto x^*.$$

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

$$y^*(\mathcal{L}(x)) = \mathcal{L}^* y^*(x)$$

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

Adjoints in Hilbert space

See Estep 2004

$$\mathcal{L}^* : y^* \mapsto x^*.$$
$$y^*(\mathcal{L}(x)) = \mathcal{L}^* y^*(x)$$

When X and Y are Hilbert spaces, then we can identify them with their dual space:

- by the Riesz representation theorem if $y^* \in Y^*$ there exists $y \in Y$ such that $y^* = \langle \cdot, y \rangle_Y$ (and vice versa...).

Adjoints in Hilbert space
See Estep 2004

$$\mathcal{L}^*: y^* \mapsto x^*.$$
$$y^*(\mathcal{L}(x)) = \mathcal{L}^* y^*(x)$$

When X and Y are Hilbert spaces, then we can identify them with their dual space:

- by the Riesz representation theorem if $y^* \in Y^*$ there exists $y \in Y$ such that $y^* = \langle \cdot, y \rangle_Y$ (and vice versa...).

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

$$\langle \mathcal{L}x, y \rangle_Y = y^*(\mathcal{L}(x)) = \mathcal{L}^* y^*(x) = \langle x, \mathcal{L}^* y \rangle_X.$$

where we now consider $\mathcal{L}^*: Y \to X$.

Benefits of adjoints

$$\min_{p,q} S(p,q) = (z - g(x))^\top (z - g(x))$$
$$\text{subject to} \quad \mathcal{L}_p x = f_q.$$

1. If f_q depends linearly on q we can easily compute the least squares estimator
$$\hat{q}(p) = \arg\min_q S(p,q)$$

Benefits of adjoints

$$\min_{p,q} S(p,q) = (z - g(x))^\top (z - g(x))$$
$$\text{subject to} \quad \mathcal{L}_p x = f_q.$$

1. If f_q depends linearly on q we can easily compute the least squares estimator

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

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

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

Benefits of adjoints

$$\min_{p,q} S(p,q) = (z - g(x))^\top (z - g(x))$$
$$\text{subject to} \quad \mathcal{L}_p x = f_q.$$

1. If f_q depends linearly on q we can easily compute the least squares estimator
$$\hat{q}(p) = \arg\min_q S(p,q)$$
   - If $z = h(x) + N(0, \Sigma)$, and $q \sim N(m, C)$ a priori, then
$$q \mid z, p = N(m^*, C^*)$$
2. We can compute $\frac{\mathrm{d}S}{\mathrm{d}p}(p,q)$ and approximate $\frac{\mathrm{d}S}{\mathrm{d}p}(p, \hat{q}(p))$

Benefits of adjoints

$$\min_{p,q} S(p,q) = (z - g(x))^\top (z - g(x))$$
$$\text{subject to} \quad \mathcal{L}_p x = f_q.$$

1. If f_q depends linearly on q we can easily compute the least squares estimator
$$\hat{q}(p) = \arg\min_q S(p,q)$$
   - If $z = h(x) + N(0,\Sigma)$, and $q \sim N(m, C)$ a priori, then
$$q \mid z, p = N(m^*, C^*)$$
2. We can compute $\frac{\mathrm{d}S}{\mathrm{d}p}(p,q)$ and approximate $\frac{\mathrm{d}S}{\mathrm{d}p}(p,\hat{q}(p))$

This may allow for efficient inference of p and q

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)

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

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)

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

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

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

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

$$\begin{aligned} \frac{dL}{dp} = \frac{dg}{dp} &= y^\top \frac{dh}{dp} \\ &= y^\top (\frac{df}{dp} - \frac{dA}{dp} x) \\ \text{where } A^\top y = g & \end{aligned}$$

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

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

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

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

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

Least squares

Suppose we are given n noisy observations

$$z = G^\top x + e \text{ where } e \sim N(0, \sigma^2),$$

where

$$G = \begin{pmatrix} | & \dots & | \\ g_1 & \dots & g_n \\ | & \dots & | \end{pmatrix} \qquad \text{with } A_p x = f_q$$

so that $z_i = g_i^\top x + e_i$.

We can easily use z to infer parameters q.

Least squares

Suppose we are given n noisy observations

$$z = G^\top x + e \text{ where } e \sim N(0, \sigma^2),$$

where

$$G = \begin{pmatrix} | & \dots & | \\ g_1 & \dots & g_n \\ | & \dots & | \end{pmatrix} \qquad \text{with } A_p x = f_q$$

so that $z_i = g_i^\top x + e_i$.

We can easily use z to infer parameters q.
Consider least squares, where we want to choose q to minimize

$$S(q) = (z - G^\top x)^\top (z - G^\top x) \text{ s.t. } Ax_p = f_q$$

If we let $A^\top y = g$, then using the bilinear identity, we get

$$\langle g, x\rangle = \langle A^\top y, x\rangle = \langle y, Ax\rangle = \langle y, f_q\rangle$$

and so

$$G^\top x = \begin{pmatrix} \langle g_1, x\rangle \\ \vdots \\ \langle g_n, x\rangle \end{pmatrix} = \begin{pmatrix} \langle y_1, f_q\rangle \\ \vdots \\ \langle y_n, f_q\rangle \end{pmatrix}$$

where $y_i \in \mathbb{R}^d$ are the solutions to the n adjoint systems

$$A^\top y_i = g_i$$

or in matrix notation

$$A^\top Y = G$$

where

$$Y = \begin{pmatrix} | & \dots & | \\ y_1 & \dots & y_n \\ | & \dots & | \end{pmatrix} \in \mathbb{R}^{d\times n}.$$

Now if $f_q = \Phi q$, then

$$\langle y_i, f_q \rangle = \langle \Phi^\top y_i, q \rangle.$$

And so we have that

$$G^\top x = Y^\top \Phi q \text{ where } q \in \mathbb{R}^Q.$$

We can then rewrite the sum of squares as

$$S(\theta) = (z - G^\top x)^\top (z - G^\top x) = (z - Y^\top \Phi q)^\top (z - Y^\top \Phi q)$$

and thus we can see that the least squares estimator of q is

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

Now if $f_q = \Phi q$, then

$$\langle y_i, f_q \rangle = \langle \Phi^\top y_i, q \rangle.$$

And so we have that

$$G^\top x = Y^\top \Phi q \text{ where } q \in \mathbb{R}^Q.$$

We can then rewrite the sum of squares as

$$S(\theta) = (z - G^\top x)^\top (z - G^\top x) = (z - Y^\top \Phi q)^\top (z - Y^\top \Phi q)$$

and thus we can see that the least squares estimator of q is

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

The conjugate Bayesian result follows similarly.

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).

Parameterizing GPs

In infinite dimensional problems, we model unknown functions as Gaussian processes (GPs).

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

Parameterizing GPs

In infinite dimensional problems, we model unknown functions as Gaussian processes (GPs).

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

$f \in \mathcal{F}_k$ the RKHS associated with kernel k.

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

We then approximate f using a truncated basis expansion

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

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 2: Ordinary differential equation

Consider the ordinary differential equation

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

with $x(0) = \dot{x}(0) = 0$.

Assume

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

Example 2: Ordinary differential equation

Consider the ordinary differential equation

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

with $x(0) = \dot{x}(0) = 0$.

Assume

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

Can we infer $f(t)$ given $z = g(x) + N(0, \Sigma^2)$?

Example 2: finding the adjoint

Use the bilinear identity to find the adjoint of

$$\mathcal{L}x = (-D\frac{\mathrm{d}^2}{\mathrm{d}t^2} + u\frac{\mathrm{d}}{\mathrm{d}t} + 1)x$$

$$\langle \mathcal{L}x, y\rangle = \int_0^T \mathcal{L}x(t)y(t)\mathrm{d}t = \int_0^T (-D\ddot{x} + u\dot{x} + x)y\mathrm{d}t$$

Example 2: finding the adjoint

Use the bilinear identity to find the adjoint of

$$\mathcal{L}x = (-D\frac{\mathrm{d}^2}{\mathrm{d}t^2} + u\frac{\mathrm{d}}{\mathrm{d}t} + 1)x$$

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

Example 2: finding the adjoint

Use the bilinear identity to find the adjoint of

$$\mathcal{L}x = (-D\frac{\mathrm{d}^2}{\mathrm{d}t^2} + u\frac{\mathrm{d}}{\mathrm{d}t} + 1)x$$

$$\begin{aligned}
\langle \mathcal{L}x, y\rangle &= \int_0^T \mathcal{L}x(t)y(t)\mathrm{d}t = \int_0^T (-D\ddot{x} + u\dot{x} + x)y\mathrm{d}t \\
&= [-D\dot{x}y]_0^T + \int_0^T D\dot{x}\dot{y}\mathrm{d}t + [uxy]_0^T - \int_0^T ux\dot{y}dt + \int_0^T xy\mathrm{d}t \\
&= [-Dx\dot{y}]_0^T - \int_0^T Dx\ddot{y}\mathrm{d}t - \int_0^T ux\dot{y}dt + \int_0^T xy\mathrm{d}t \\
&= \int_0^T (-D\ddot{y} - u\dot{y} + y)x\mathrm{d}t \quad \text{when } y(T) = \dot{y}(T) = 0 \\
&= \langle x, \mathcal{L}^* y\rangle
\end{aligned}$$

Example 2: finding the adjoint

Use the bilinear identity to find the adjoint of

$$\mathcal{L}x = (-D\frac{\mathrm{d}^2}{\mathrm{d}t^2} + u\frac{\mathrm{d}}{\mathrm{d}t} + 1)x$$

$$\begin{aligned}
\langle \mathcal{L}x, y\rangle &= \int_0^T \mathcal{L}x(t)y(t)\mathrm{d}t = \int_0^T (-D\ddot{x} + u\dot{x} + x)y\mathrm{d}t \\
&= [-D\dot{x}y]_0^T + \int_0^T D\dot{x}\dot{y}\mathrm{d}t + [uxy]_0^T - \int_0^T ux\dot{y}dt + \int_0^T xy\mathrm{d}t \\
&= [-Dx\dot{y}]_0^T - \int_0^T Dx\ddot{y}\mathrm{d}t - \int_0^T ux\dot{y}dt + \int_0^T xy\mathrm{d}t \\
&= \int_0^T (-D\ddot{y} - u\dot{y} + y)x\mathrm{d}t \quad \text{when } y(T) = \dot{y}(T) = 0 \\
&= \langle x, \mathcal{L}^* y\rangle
\end{aligned}$$

So we have

$$\mathcal{L}^* y = (-D\frac{\mathrm{d}^2}{\mathrm{d}t^2} - u\frac{\mathrm{d}}{\mathrm{d}t} + 1)y$$

Example 2: finding the adjoint

Use the bilinear identity to find the adjoint of

$$\mathcal{L}x = (-D\frac{\mathrm{d}^2}{\mathrm{d}t^2} + u\frac{\mathrm{d}}{\mathrm{d}t} + 1)x$$

$$\begin{aligned}
\langle \mathcal{L}x, y\rangle &= \int_0^T \mathcal{L}x(t)y(t)\mathrm{d}t = \int_0^T (-D\ddot{x} + u\dot{x} + x)y\mathrm{d}t \\
&= [-D\dot{x}y]_0^T + \int_0^T D\dot{x}\dot{y}\mathrm{d}t + [uxy]_0^T - \int_0^T ux\dot{y}dt + \int_0^T xy\mathrm{d}t \\
&= [-Dx\dot{y}]_0^T - \int_0^T Dx\ddot{y}\mathrm{d}t - \int_0^T ux\dot{y}dt + \int_0^T xy\mathrm{d}t \\
&= \int_0^T (-D\ddot{y} - u\dot{y} + y)x\mathrm{d}t \quad \text{when } y(T) = \dot{y}(T) = 0 \\
&= \langle x, \mathcal{L}^* y\rangle \qquad \text{NB: the adjoint is solved backwards in time}
\end{aligned}$$

So we have

$$\mathcal{L}^* y = (-D\frac{\mathrm{d}^2}{\mathrm{d}t^2} - u\frac{\mathrm{d}}{\mathrm{d}t} + 1)y$$

Example 2: Bilinear identity

If the primal system is

$$\mathcal{L}x = f \text{ and we observe } z_i = \langle g_i, x \rangle + e_i$$

then by the bilinear identity

$$z_i = \langle y_i, f \rangle + e_i$$

where

$$\mathcal{L}^* y_i = g_i.$$

Think of $g_i(x) = \langle g_i, x \rangle$ as linear sensor functions. Typical choices

- Point value $g_i(x) = x(t_i)$
- Temporal average $g_i(x) = \int_{t_i - \delta}^{t_i + \delta} x(t) \mathrm{d}t$

Example 2: GP expansion

$f(\cdot) \sim GP$. If we write

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

then

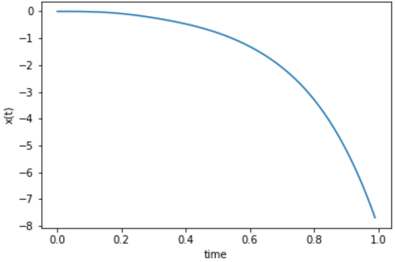
$$\begin{aligned} z_i &= \langle y_i, f \rangle + e_i \\ &= \sum_{j=1}^{M} \langle y_i, \phi_j \rangle q_j + e_i \\ &= y_i^\top \Phi \mathbf{q} + e_i \end{aligned}$$

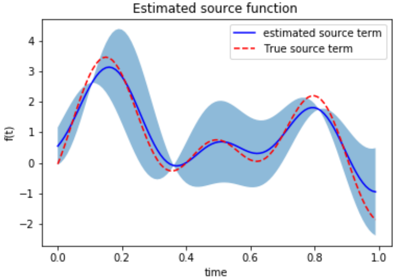
Thus we can estimate $\mathbf{q}$ by

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

Example 2: Results

20 observations, each a noisy average over 0.025s. 100 Fourier features

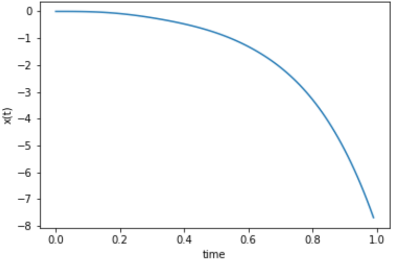


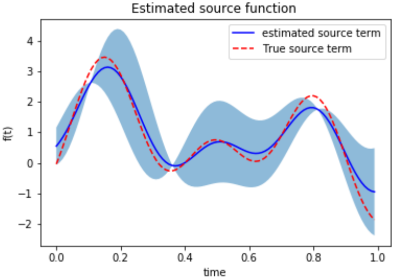


These results require 20 adjoint solves: < 1 second.

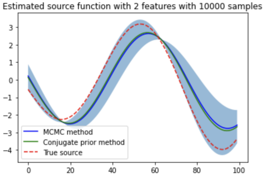
Example 2: Results

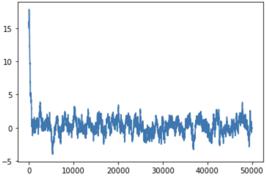
20 observations, each a noisy average over 0.025s. 100 Fourier features





These results require 20 adjoint solves: < 1 second.
MCMC works here for a small number of features. But even with 2 features, we need $\sim 1000s$ of ODE solves.





Example 3: PDE

Advection-diffusion is a linear operator:

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

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

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

Example 3: PDE

Advection-diffusion is a linear operator:

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

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

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

Inverse problem: assume

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

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

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

Example 3: PDE adjoint

For n observations we need n adjoint equations!

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

along with initial (final) and boundary conditions

Example 3: PDE adjoint

For n observations we need n adjoint equations!

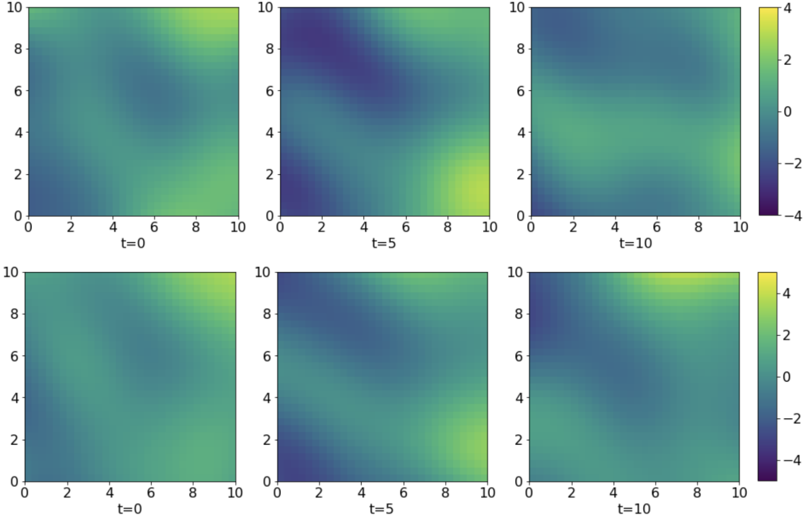
$$-\frac{\partial v}{\partial t} - \nu \nabla v - \nabla.(D \nabla v) = g_i \text{ in } \Omega \times (T, 0)$$

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 g_i vs diffusion rate etc
- The cost of solving the adjoint is the same as solving the forward problem.

Example 3: Results

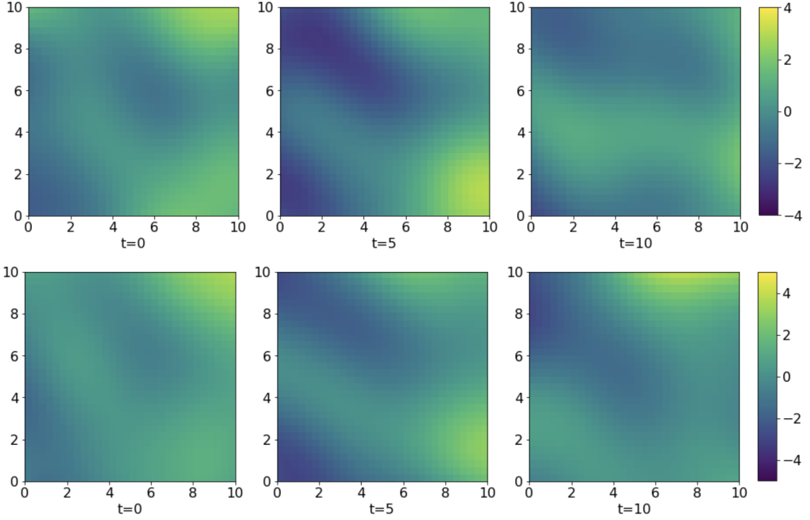
25 sensors, 3 observations from each, $\sigma = 0.05$, Domain $= [0, 10] \times [0, 10]^2$



These are best case results with known GP and PDE hyperparameters.

Example 3: Results

25 sensors, 3 observations from each, $\sigma = 0.05$, Domain $= [0, 10] \times [0, 10]^2$

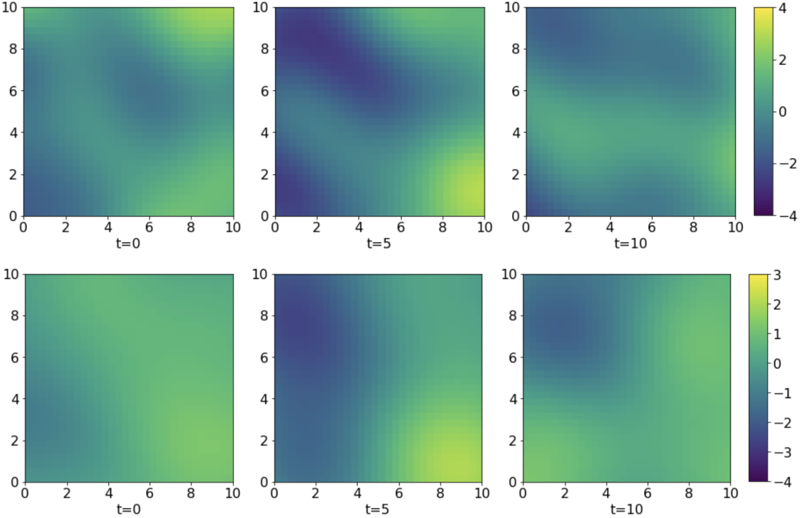


These are best case results with known GP and PDE hyperparameters.

Note the negative values....

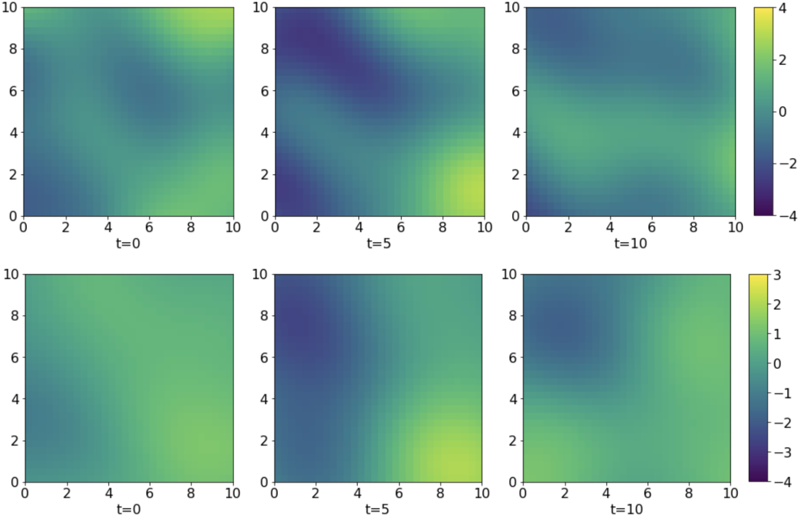
Example 3: Results - posterior mean

4 sensors, 3 observations from each, $\sigma = 0.05$, Domain $= [0, 10] \times [0, 10]^2$



Example 3: Results - posterior mean

4 sensors, 3 observations from each, $\sigma = 0.05$, Domain $= [0, 10] \times [0, 10]^2$



We're currently working on using the adjoint to estimate the non-linear parameters.

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 mathematics...
- 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...
- First paper to appear on arXiv soon.

Conclusions

Adjoints of linear systems

- an intrusive method; development does require some mathematics...
- 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...
- First paper to appear on arXiv soon.

Thank you for listening!