Adjoint-aided inference of Gaussian process driven differential equations

Paterne Gahungu[1], **Christopher Lanyon**[5], Mauricio Alvarez[3], Engineer Bainomugisha[4], Michael Smith[2]
Richard Wilkinson[5]

[1] Department of Mathematics, Imperial College London
[2] Department of Computer Science, University of Sheffield
[3] Department of Computer Science, University of Manchester
[4] Department of Computer Science, Makerere University
[5] School of Mathematical Sciences, University of Nottingham

GPSS 2024

Project team

Paterne Engineer Mike Mauricio Richard

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:

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

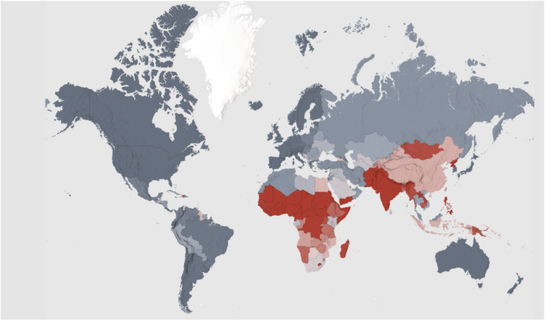
  Given noisy measurements of u can we infer f?
- Adjoints

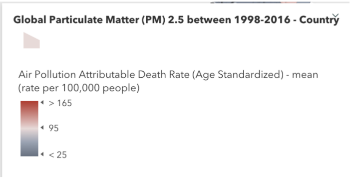
$$\mathcal{L}^* v \text{ such that } \langle \mathcal{L}u, v \rangle = \langle u, \mathcal{L}^* v \rangle$$

- Examples

Air pollution

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



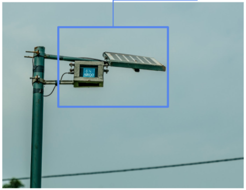


The UK government estimates the annual mortality of human-made air pollution to be 28,000 to 36,000 deaths, and costs UK $\sim$£10^{10}

Kampala and AirQo

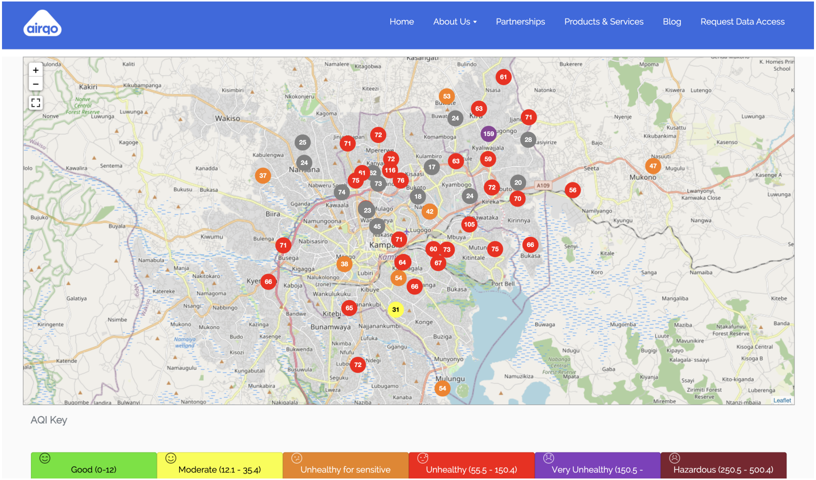
Smith et al. to appear JRSS C

- 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 snapshot from 2023 (midday)



London (2022 average): 9.6 $\mu g/m^3$
20 year average for UK: 11 $\mu g/m^3$
WHO guideline: $5\mu g/m^3$

Air pollution digital twin

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

Air pollution digital twin

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

$$\frac{\partial u}{\partial t} = \nabla.(\mathbf{p}_1 u) + \nabla.(p_2 \nabla u) - p_3 u + \sum_i f_i$$

- $f_i(x, t)$ are different pollution sources,
- we may choose to model different pollution types (PM2.5, PM10 etc)

Air pollution digital twin

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

$$\frac{\partial u}{\partial t} = \nabla.(\mathbf{p}_1 u) + \nabla.(p_2 \nabla u) - p_3 u + \sum_i f_i$$

- $f_i(x, t)$ are different pollution sources,
- we may choose to model different pollution types (PM2.5, PM10 etc)

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

Air pollution digital twin

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

$$\frac{\partial u}{\partial t} = \nabla.(\mathbf{p}_1 u) + \nabla.(p_2 \nabla u) - p_3 u + \sum_i f_i$$

- $f_i(x,t)$ are different pollution sources,
- we may choose to model different pollution types (PM2.5, PM10 etc)

Hypothesis: The inclusion of mechanistic behaviour will allow us to infer sources, plan interventions, and predict better.
NB: can also extend the model with a GP to capture missing physics

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 $f_i(x, t)$?
- 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 $f_i(x, t)$?
- the diffusion, advection and reaction parameters? Hyperparameters etc?

Use Gaussian process priors for $f_i(x, t)$

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

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

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

What do we need?

We have a problem and a framework, what do we need to achieve our goal of inferring u and f

What do we need?

We have a problem and a framework, what do we need to achieve our goal of inferring u and f

- A method to perform efficient inference

What do we need?

We have a problem and a framework, what do we need to achieve our goal of inferring u and f

- A method to perform efficient inference
- A way to model f

The adjoint operator

See Estep 2004

Let $\mathcal{L} : \mathcal{U} \to \mathcal{V}$ be our linear operator where $\mathcal{U}$ and $\mathcal{V}$ are Hilbert spaces

- i.e. vector spaces with an inner product $\langle u, u' \rangle$,

then the adjoint of $\mathcal{L}$, $\mathcal{L}^* : \mathcal{V} \to \mathcal{U}$ satisfies

$$\begin{aligned} \langle \mathcal{L}u, v \rangle &= v^*(\mathcal{L}(u)) = \mathcal{L}^* v^*(u) \\ &= \langle u, \mathcal{L}^* v \rangle, \end{aligned}$$

known as the bilinear identity.

The adjoint operator

See Estep 2004

Let $\mathcal{L} : \mathcal{U} \to \mathcal{V}$ be our linear operator where $\mathcal{U}$ and $\mathcal{V}$ are Hilbert spaces

- i.e. vector spaces with an inner product $\langle u, u' \rangle$,

then the adjoint of $\mathcal{L}$, $\mathcal{L}^* : \mathcal{V} \to \mathcal{U}$ satisfies

$$\begin{aligned} \langle \mathcal{L}u, v \rangle &= v^*(\mathcal{L}(u)) = \mathcal{L}^* v^*(u) \\ &= \langle u, \mathcal{L}^* v \rangle, \end{aligned}$$

known as the bilinear identity.

NB: This formulation extends more generally to any Banach space, but for our purposes today, Hilbert spaces are enough.

Benefits of the adjoint

See Estep 2004

Adjoints have the additional properties of

- allowing us to easily calculate the derivative of some cost function between our inference and the observations
- can be used to easily compute the least squares estimate

Example 0

In the finite dimensional case, $\mathcal{U} = \mathbb{R}^n$, $\mathcal{V} = \mathbb{R}^m$, then $\langle u_1, u_2 \rangle = u_1^\top u_2$ etc and

$$\mathcal{L}u = Au \text{ for some m x n matrix } A.$$

Example 0

In the finite dimensional case, $\mathcal{U} = \mathbb{R}^n$, $\mathcal{V} = \mathbb{R}^m$, then $\langle u_1, u_2 \rangle = u_1^\top u_2$ etc and

$$\mathcal{L}u = Au \text{ for some m x n matrix } A.$$

Then

$$\mathcal{L}^* v = A^\top v$$

That is

$$\langle Au, v \rangle = \langle u, A^\top v \rangle$$

Efficient inference

$$\mathcal{L}u = f, \qquad z_i = h_i(u) + e$$

If the observation operator is linear

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

we can consider the n adjoint systems

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

Efficient inference

$$\mathcal{L}u = f, \qquad z_i = h_i(u) + e$$

If the observation operator is linear

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

we can consider the n adjoint systems

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

Then

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

by the bilinear identity.

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

Suppose f is a parametric model with a linear dependence upon some unknown parameters q:

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

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

Suppose f is a parametric model with a linear dependence upon some unknown parameters q:

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

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

A linear model!

The complete observation vector z can then be written as

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

$$= \Phi q + e$$

The complete observation vector z can then be written as

$$z = \begin{pmatrix} \langle v_1, \phi_1 \rangle & \dots & \langle v_1, \phi_M \rangle \\ \vdots & & \vdots \\ \langle v_n, \phi_1 \rangle & \dots & \langle v_n, \phi_M \rangle \end{pmatrix} \begin{pmatrix} q_1 \\ \\ q_M \end{pmatrix} + e \qquad (2)$$

$$= \Phi q + e$$

Thus

$$\min_f \quad S(f) = (z - h(u))^\top (z - h(u))$$
$$\text{subject to} \quad \mathcal{L}u = f$$

is equivalent to

$$\min_q \quad S(q) = (z - \Phi q)^\top (z - \Phi q)$$

The complete observation vector z can then be written as

$$z = \begin{pmatrix} \langle v_1, \phi_1 \rangle & \dots & \langle v_1, \phi_M \rangle \\ \vdots & & \vdots \\ \langle v_n, \phi_1 \rangle & \dots & \langle v_n, \phi_M \rangle \end{pmatrix} \begin{pmatrix} q_1 \\ \\ q_M \end{pmatrix} + e \qquad (2)$$

$$= \Phi q + e$$

Thus

$$\min_f \quad S(f) = (z - h(u))^\top (z - h(u))$$

$$\text{subject to} \quad \mathcal{L} u = f$$

is equivalent to

$$\min_q \quad S(q) = (z - \Phi q)^\top (z - \Phi q)$$

The solution is

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

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 we model unknown function $f = \{f(x) : x \in \mathcal{X}\}$ as a Gaussian process (GP)

- i.e. joint distribution of $f(x_1), \ldots, f(x_n)$ is Gaussian.

Quick intro to Gaussian Processes

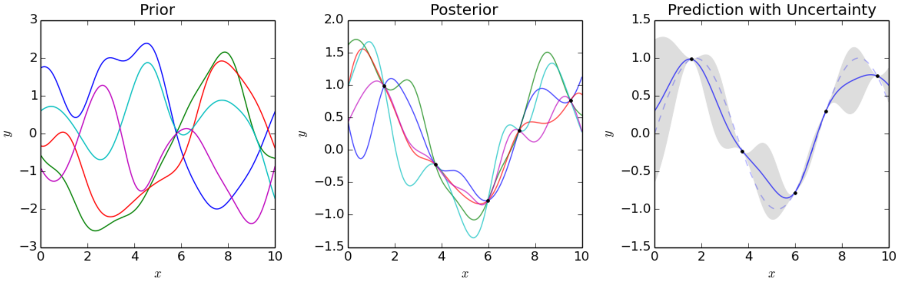
Suppose we model unknown function $f = \{f(x) : x \in \mathcal{X}\}$ as a Gaussian process (GP)

- i.e. joint distribution of $f(x_1), \ldots, f(x_n)$ is Gaussian.

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

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

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



Why use GPs?

- Mathematically attractive family
  - Closed under addition

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

Why use GPs?

- Mathematically attractive family
  - 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 family
  - 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 family
  - 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 family
  - 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

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?

- Let $\mathcal{F}$ be the RKHS (function space) associated with kernel k, i.e., $f \in \mathcal{F}$
- Consider $\{\phi_1(x), \phi_2(x), \ldots\}$ an orthonormal basis for $\mathcal{F}$.

Parameterizing GPs

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

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

- Let $\mathcal{F}$ be the RKHS (function space) associated with kernel k, i.e., $f \in \mathcal{F}$
- Consider $\{\phi_1(x), \phi_2(x), \ldots\}$ 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 approximated the GP by a linear model.

Choice of basis in $f_q(\cdot) = \sum^M q_i \lambda_i \phi_i(\cdot)$

- **Mercer basis**: $\phi_i(x) = \lambda_i \psi(x)$ where $\lambda_i, \phi_i(\cdot)$ are eigenpairs of

$$T_k(f)(\cdot) = \int_{\mathcal{X}} k(x, \cdot) f(x) \mathrm{d}x.$$

Karhunen-Loève theorem says this choice is mean square optimal

Choice of basis in $f_q(\cdot) = \sum^M q_i \lambda_i \phi_i(\cdot)$

- **Mercer basis**: $\phi_i(x) = \lambda_i \psi(x)$ where $\lambda_i, \phi_i(\cdot)$ are eigenpairs of

$$T_k(f)(\cdot) = \int_{\mathcal{X}} k(x, \cdot) f(x) \mathrm{d}x.$$

  Karhunen-Loève theorem says this choice is mean square optimal

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

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

  Thus we can use $\phi_i(x) = \cos(w_i^\top x + b_i)$ where $w_i \sim p(\cdot)$ and $b_i \sim U[0, 2\pi]$

Choice of basis in $f_q(\cdot) = \sum^M q_i \lambda_i \phi_i(\cdot)$

- **Mercer basis**: $\phi_i(x) = \lambda_i \psi(x)$ where $\lambda_i, \phi_i(\cdot)$ are eigenpairs of

$$T_k(f)(\cdot) = \int_{\mathcal{X}} k(x, \cdot) f(x) \mathrm{d}x.$$

  Karhunen-Loève theorem says this choice is mean square optimal

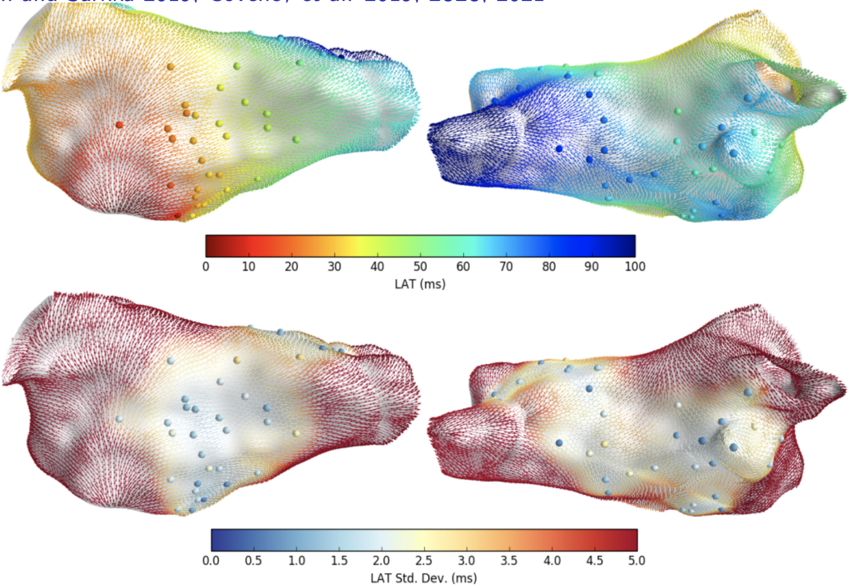
- **Random Fourier features**: If k stationary, Bochner's theorem:

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

  Thus we can use $\phi_i(x) = \cos(w_i^\top x + b_i)$ where $w_i \sim p(\cdot)$ and $b_i \sim U[0, 2\pi]$

Laplacian basis: useful for non-Euclidean domains

Solin and Sarkka 2019, Coveney et al. 2019, **2020**, 2021



Algorithm

For a given linear system $\mathcal{L}$ with unknown forcing function f, $\mathcal{L}u = f$ and observations $z_i = h_i(u) + \epsilon$

Algorithm

For a given linear system $\mathcal{L}$ with unknown forcing function f, $\mathcal{L}u = f$ and observations $z_i = h_i(u) + \epsilon$

- Choose a basis expression for f s.t.

$$f = \sum_{m=1}^{m} q_m \phi_m$$

Algorithm

For a given linear system $\mathcal{L}$ with unknown forcing function f, $\mathcal{L}u = f$ and observations $z_i = h_i(u) + \epsilon$

- Choose a basis expression for f s.t.

$$f = \sum_{m=1}^{m} q_m \phi_m$$

- Determine the adjoint operator, $\mathcal{L}^*$ (and boundary conditions if appropriate)

Algorithm

For a given linear system $\mathcal{L}$ with unknown forcing function f, $\mathcal{L}u = f$ and observations $z_i = h_i(u) + \epsilon$

- Choose a basis expression for f s.t.

$$f = \sum_{m=1}^{m} q_m \phi_m$$

- Determine the adjoint operator, $\mathcal{L}^*$ (and boundary conditions if appropriate)
- Solve the adjoint system $\mathcal{L}^* v_i = h_i$

Algorithm

For a given linear system $\mathcal{L}$ with unknown forcing function f, $\mathcal{L}u = f$ and observations $z_i = h_i(u) + \epsilon$

- Choose a basis expression for f s.t.

$$f = \sum_{m=1}^{m} q_m \phi_m$$

- Determine the adjoint operator, $\mathcal{L}^*$ (and boundary conditions if appropriate)
- Solve the adjoint system $\mathcal{L}^* v_i = h_i$
- Compute the regressor matrix Φ where $[\Phi]_{im} = \langle v_i, \phi_m \rangle$

Algorithm

For a given linear system $\mathcal{L}$ with unknown forcing function f, $\mathcal{L}u = f$ and observations $z_i = h_i(u) + \epsilon$

- Choose a basis expression for f s.t.

$$f = \sum_{m=1}^{m} q_m \phi_m$$

- Determine the adjoint operator, $\mathcal{L}^*$ (and boundary conditions if appropriate)
- Solve the adjoint system $\mathcal{L}^* v_i = h_i$
- Compute the regressor matrix Φ where $[\Phi]_{im} = \langle v_i, \phi_m \rangle$
- Compute the posterior distribution for q

$$q \mid z \sim \mathcal{N}_M(\mu_n, \Sigma_n) \tag{5}$$

where

$$\mu_n = \Sigma_n(\frac{1}{\sigma^2}\Phi^\top z + \Sigma_0^{-1}\mu_0), \quad \Sigma_n = \left(\frac{1}{\sigma^2}\Phi^\top \Phi + \Sigma_0^{-1}\right)^{-1}. \tag{6}$$

Example 1: PDE

Advection-diffusion-reaction is a linear operator:

$$\mathcal{L}u = \frac{\partial u}{\partial t} - \nabla.(\mathbf{p}_1 u) - \nabla.(p_2 \nabla u) + p_3 u$$

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

$$\mathcal{L}u = f \text{ on } \mathcal{X} \times [0, T].$$

Example 1: PDE

Advection-diffusion-reaction is a linear operator:

$$\mathcal{L}u = \frac{\partial u}{\partial t} - \nabla.(\mathbf{p}_1 u) - \nabla.(p_2 \nabla u) + p_3 u$$

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

$$\mathcal{L}u = f \text{ on } \mathcal{X} \times [0, T].$$

Inverse problem: assume

$$f(x,t) \sim GP(m, k_\lambda((x,t),(x',t')))$$

and estimate f given $z_i = \langle h_i, u \rangle + N(0, \sigma)$.

Example 1: PDE

Advection-diffusion-reaction is a linear operator:

$$\mathcal{L}u = \frac{\partial u}{\partial t} - \nabla.(\mathbf{p}_1 u) - \nabla.(p_2 \nabla u) + p_3 u$$

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

$$\mathcal{L}u = f \text{ on } \mathcal{X} \times [0, T].$$

Inverse problem: assume

$$f(x,t) \sim GP(m, k_\lambda((x,t),(x',t')))$$

and estimate f given $z_i = \langle h_i, u \rangle + N(0, \sigma)$.
h_i are sensor functions that average the pollution at a specific location over a short window

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

Example 1: PDE adjoint

The adjoint system is derived by integrating by parts twice:

$$\mathcal{L}^* v = -\frac{\partial v}{\partial t} - \mathbf{p}_1 . \nabla v - \nabla \cdot (p_2 \nabla v) + p_3 u.$$

Example 1: PDE adjoint

The adjoint system is derived by integrating by parts twice:

$$\mathcal{L}^* v = -\frac{\partial v}{\partial t} - \mathbf{p}_1.\nabla v - \nabla \cdot (p_2 \nabla v) + p_3 u.$$

For n observations we need n adjoint equations!

$$\mathcal{L}^* v_i = h_i \text{ in } \mathcal{X} \times [0, T] \text{ for } i = 1, \ldots, n.$$

Example 1: PDE adjoint

The adjoint system is derived by integrating by parts twice:

$$\mathcal{L}^* v = -\frac{\partial v}{\partial t} - \mathbf{p}_1 . \nabla v - \nabla \cdot (p_2 \nabla v) + p_3 u.$$

For n observations we need n adjoint equations!

$$\mathcal{L}^* v_i = h_i \text{ in } \mathcal{X} \times [0, T] \text{ for } i = 1, \ldots, n.$$

If we use initial and boundary conditions

$$u(x, 0) = 0 \text{ for } x \in \mathcal{X} \text{ and } \nabla_n u = 0 \text{ for } x \in \partial\mathcal{X}$$

then the final and boundary conditions on the adjoint system are

$$v_i(x, T) = 0 \text{ for } x \in \mathcal{X}$$

$$\mathbf{p}_1 v_i(x, t) + p_2 \nabla v_i(x, t) = 0 \text{ for } x \in \partial\Omega \text{ and } t \in [0, T].$$

Example 1: PDE adjoint

The adjoint system is derived by integrating by parts twice:

$$\mathcal{L}^* v = -\frac{\partial v}{\partial t} - \mathbf{p}_1.\nabla v - \nabla \cdot (p_2 \nabla v) + p_3 u.$$

For n observations we need n adjoint equations!

$$\mathcal{L}^* v_i = h_i \text{ in } \mathcal{X} \times [0, T] \text{ for } i = 1, \ldots, n.$$

If we use initial and boundary conditions

$$u(x, 0) = 0 \text{ for } x \in \mathcal{X} \text{ and } \nabla_n u = 0 \text{ for } x \in \partial\mathcal{X}$$

then the final and boundary conditions on the adjoint system are

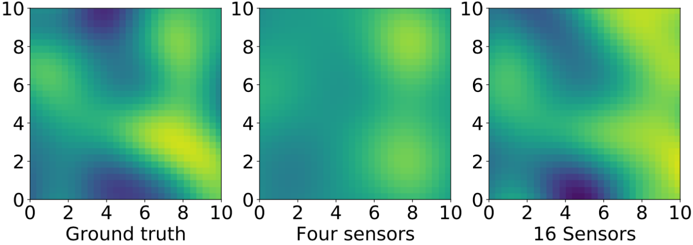
$$v_i(x, T) = 0 \text{ for } x \in \mathcal{X}$$

$$\mathbf{p}_1 v_i(x, t) + p_2 \nabla v_i(x, t) = 0 \text{ for } x \in \partial\Omega \text{ and } t \in [0, T].$$

- May find numerical issues: depends on the discretization, the sensor functions h_i, diffusion rate etc
- The cost of solving the adjoint is the same as solving the forward problem.

Results: $n = 20$ (4 sensors) and $n = 80$ (16), noise =10%

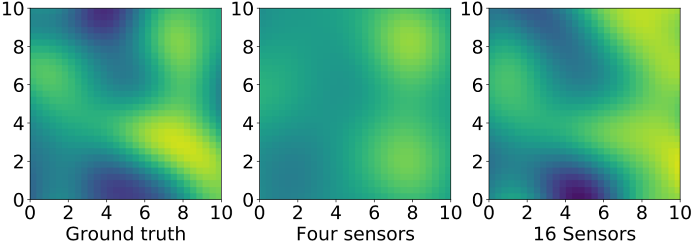
Posterior mean of time slice $u(x, 5)$ - more sensors, improved estimates!



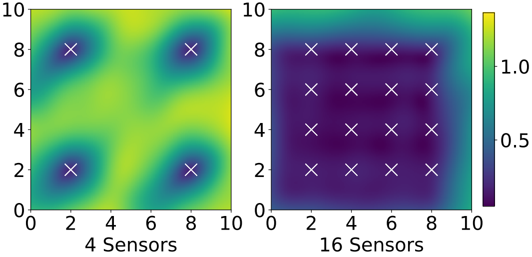
Variance of $u(x, 5)$: Wind from the south west.

Results: $n = 20$ (4 sensors) and $n = 80$ (16), noise =10%

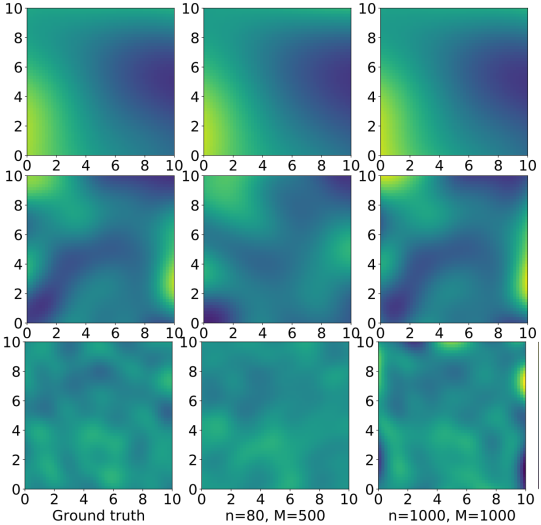
Posterior mean of time slice $u(x, 5)$ - more sensors, improved estimates!



Variance of $u(x, 5)$: Wind from the south west.



Effect of length scale, $\lambda = 5, 2, 1$

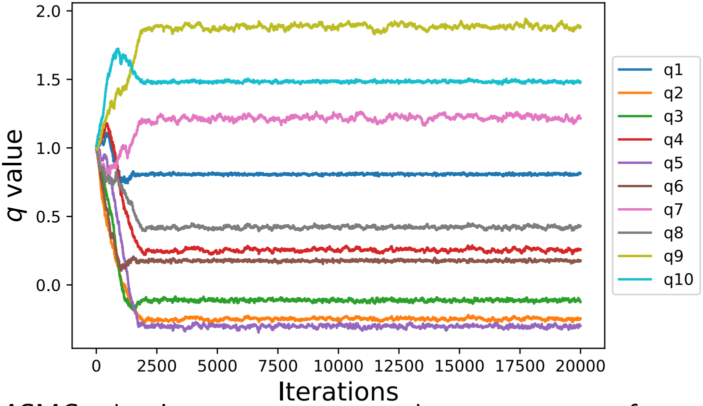


MSE 0.008 and 0.004

MSE 0.68 and 0.07

MSE 1.85 and 2.55

MCMC is fine as long as you have a small number of features.
But even with only 10 features, we need $\sim 1000s$ of ODE solves vs 10 ODE solves for the adjoint method.



MCMC takes longer to converge when we use more features.

Example 1: Results

Mean square error vs number of features and sensors

Median MSE as a function of number of sensors and basis vectors.

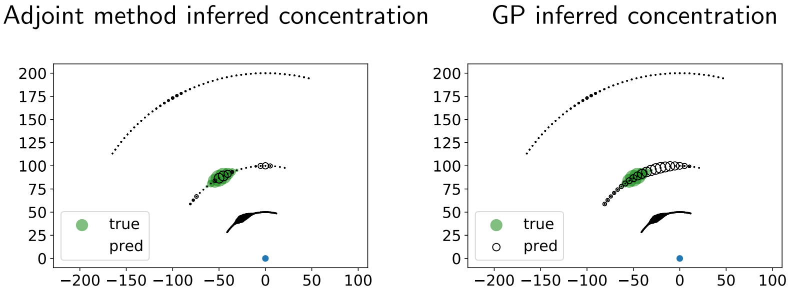
Sensors	# basis vectors					
	10	50	100	200	300	500
1	3.42 (2.82,4.39)	3.27 (3.13,3.38)	3.24 (3.10,3.37)	3.27 (3.17,3.44)	3.24 (3.12,3.36)	3.27 (3.17
4	7.12 (1.57,28.81)	2.39 (2.06,2.62)	2.41 (2.13,2.60)	2.45 (2.32,2.57)	2.50 (2.41,2.69)	2.53 (2.32
9	2.38 (1.41,4.40)	2.12 (1.48,3.98)	1.70 (1.49,2.07)	1.48 (1.40,1.72)	1.47 (1.32,1.61)	1.45 (1.40
16	1.73 (1.23,3.28)	3.99 (2.32,10.90)	2.18 (1.72,3.54)	1.3 (1.02,1.68)	1.12 (0.98,1.37)	1.12 (1.02
25	1.35 (1.19,3.09)	8.93 (4.92,39.86)	4.36 (2.53,8.20)	1.86 (1.43,2.75)	1.35 (1.07,1.81)	1.05 (0.89
25 (MH)	3.27 (1.73,6.12)	-	-	-	-	-

MH algorithm did not converge after 20,000 iterations for 50 or more RFFs.

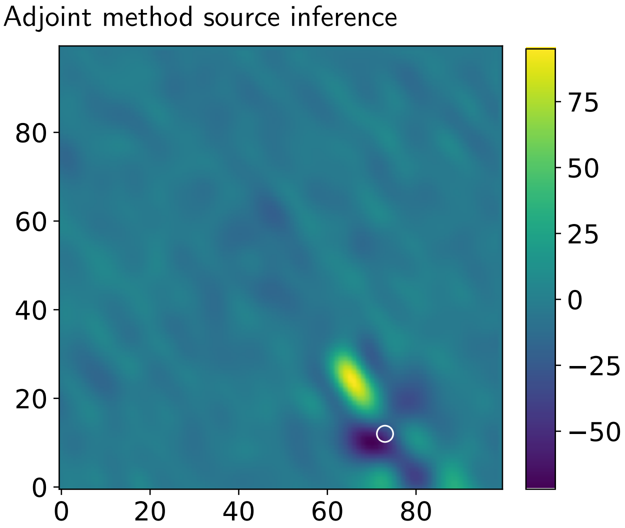
Example 2: Round Hill II Experiment

- We tested the approach using the Round Hill II advection-diffusion experiment (see Cramer & Record, 1957) using the advection-diffusion model.
- A constant source of sulphur dioxide was released over a ten minute period.
- 183 sensors were deployed in three partial concentric rings.

Example 3: Roundhill Results

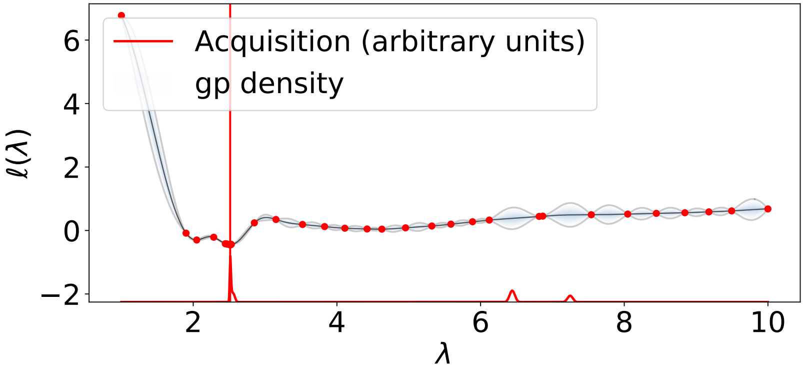


Example 2: Roundhill Results



Parameter estimation

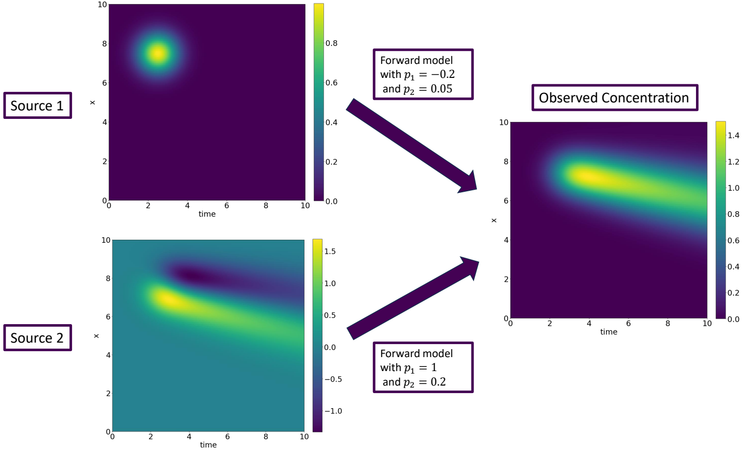
A naive way to estimate the non-linear parameters is via Bayesian optimization



- use the adjoint sensitivity to estimate derivative information
- estimate posterior using a variational approach

Parameter estimation: Identifiability

If we allow both the system parameters, p and the forcing function f to vary with no constraints, we have too many degrees of freedom and have non-unique solutions.



Parameter estimation: Identifiability

Possible solutions:

- Constrain our parameter posterior distribution
- Assume some parameters known
- Constrain our source posterior distribution

Sequential data

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

Adding features, or incorporating new data is easy

- New features/basis vectors require new columns in Φ - no new simulation is required
- New data adds rows to Φ - each new data point necessitates one additional simulation.

Costs

Adjoint method:

- require n solves of the adjoint system to infer f.
- (essentially) insensitive to the number of basis functions used.
- The non-linear parameters (GP hyperparameters, PDE parameters) can be inferred in an outer-loop

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 may be unavailable (autodiff often unstable for PDE solvers)

Conclusions

- Developed a method to infer forcing functions of linear systems given noisy observations
- requires n adjoint solves to infer the posterior
  - essentially insensitive to the number of basis functions used
- Adjoint gives numerically stable derivatives of the cost function with respect to other parameters, $\frac{\mathrm{d}S}{\mathrm{d}\boldsymbol{p}}$ etc.
- Opportunities for additional efficiencies...
  - Efficient use of adjoint simulations
  - Gradient based optimization
  - Sequential data

Ref: Gahungu et al. NeurIPS 2022, Smith et al. 2024, (forthcoming pre-prints).

Conclusions

- Developed a method to infer forcing functions of linear systems given noisy observations
- requires n adjoint solves to infer the posterior
  - essentially insensitive to the number of basis functions used
- Adjoint gives numerically stable derivatives of the cost function with respect to other parameters, $\frac{\mathrm{d}S}{\mathrm{d}p}$ etc.
- Opportunities for additional efficiencies...
  - Efficient use of adjoint simulations
  - Gradient based optimization
  - Sequential data

Ref: Gahungu et al. NeurIPS 2022, Smith et al. 2024, (forthcoming pre-prints).

Thank you for listening!