Adjoint-aided inference of Gaussian process driven differential equations

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

[1] Department of Computer Science, University of Burundi
[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

March 2023

Inference for complex models

> Nature. 2002 Apr 18;416(6882):726-9. doi: 10.1038/416726a.

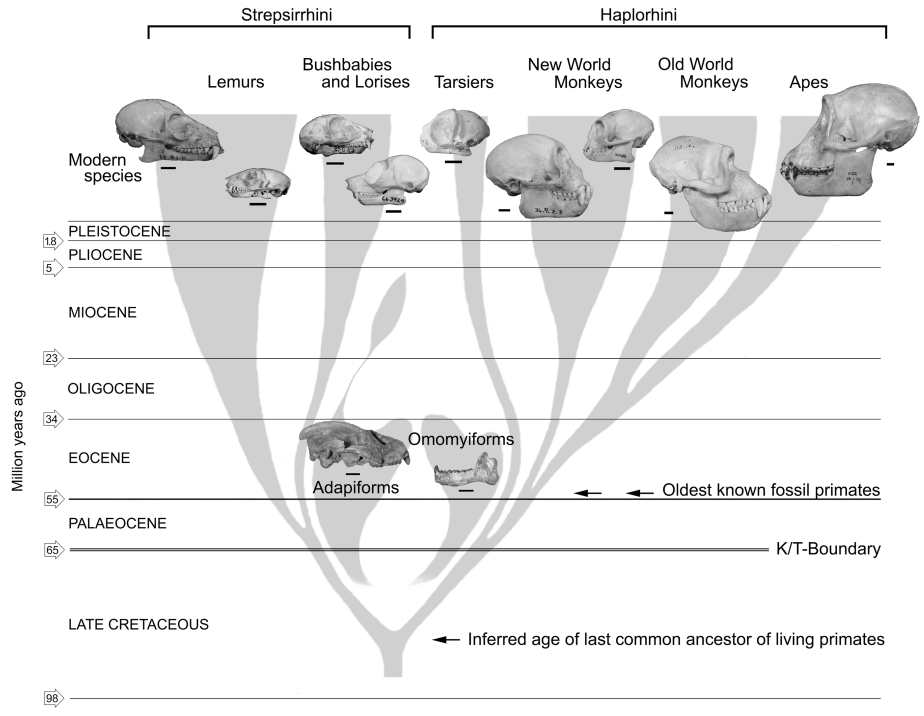
Using the fossil record to estimate the age of the last common ancestor of extant primates

Simon Tavaré [1], Charles R Marshall, Oliver Will, Christophe Soligo, Robert D Martin

Syst Biol. 2011 Jan; 60(1): 16–31. PMCID: PM(
Published online 2010 Nov 4. doi: 10.1093/sysbio/syq054 PMID:

Dating Primate Divergences through an Integrated Analysis of Palaeontological and Molecular Data

Richard D. Wilkinson,[1,*] Michael E. Steiper,[2,3,4,5] Christophe Soligo,[6] Robert D. Martin,[7] Ziheng Yang,[8] and Simon Tavaré[9]



Inference for complex models

> Nature. 2002 Apr 18;416(6882):726-9. doi: 10.1038/416726a.

Using the fossil record to estimate the age of the last common ancestor of extant primates

Simon Tavaré [1], Charles R Marshall, Oliver Will, Christophe Soligo, Robert D Martin

---

Syst Biol. 2011 Jan; 60(1): 16–31. PMCID: PM(
Published online 2010 Nov 4. doi: 10.1093/sysbio/syq054 PMID:

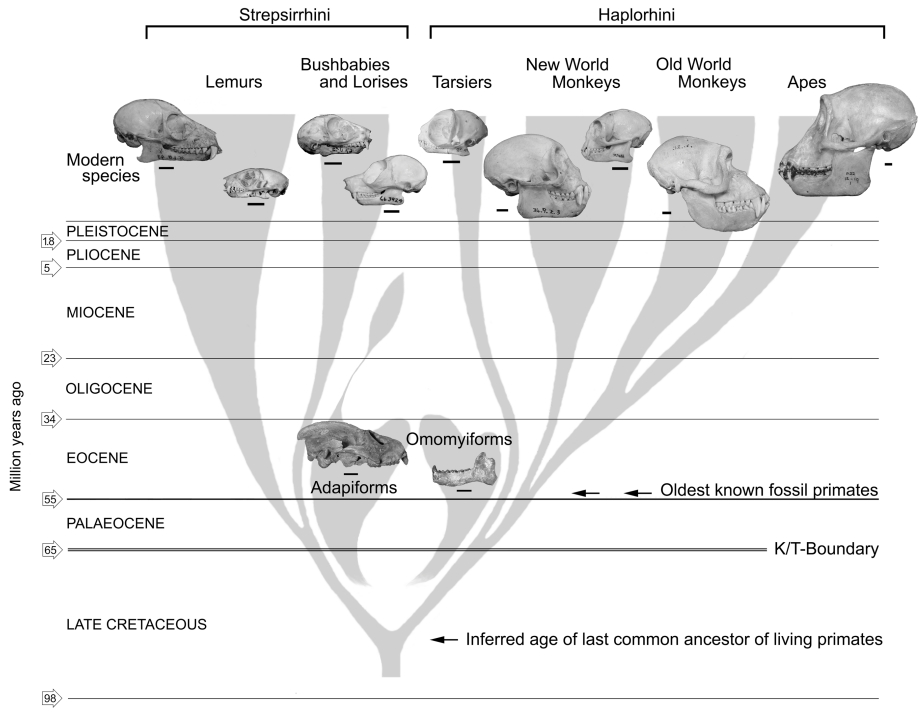
Dating Primate Divergences through an Integrated Analysis of Palaeontological and Molecular Data

Richard D. Wilkinson,[1,*] Michael E. Steiper,[2,3,4,5] Christophe Soligo,[6] Robert D. Martin,[7] Ziheng Yang,[8] and Simon Tavaré[9]

Genetics. 1997 Feb; 145(2): 505–518.
doi: 10.1093/genetics/145.2.505

Inferring Coalescence Times from DNA Sequence Data

S. Tavare, D. J. Balding, R. C. Griffiths, and P. Donnelly



ABC: given data $D = f(\theta) + e$, find $\pi(\theta|D)$

- Draw θ from $\pi(\theta)$
- Simulate $X \sim f(\theta)$
- Accept θ if $\rho(D, X) \leq \epsilon$

Works for any simulator f – no knowledge required

Project team

Paterne Engineer Mike Mauricio Chris







Funders:

airqo
Breathe Clean

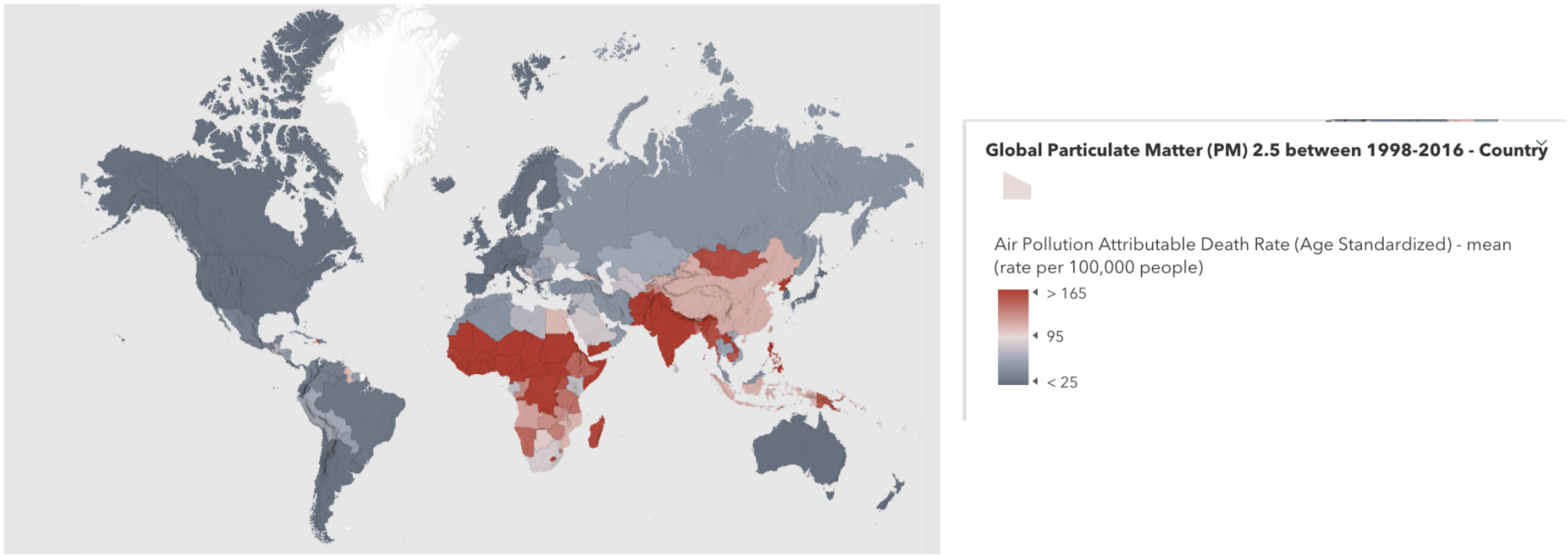
GCRF
Global Challenges
Research Fund

EPSRC
Engineering and Physical Sciences
Research Council

Google
Impact Challenge

Air pollution

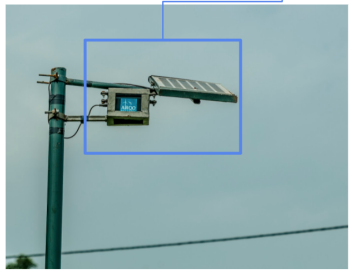
7 million people die every year from exposure to air pollution, the majority in LMICs.

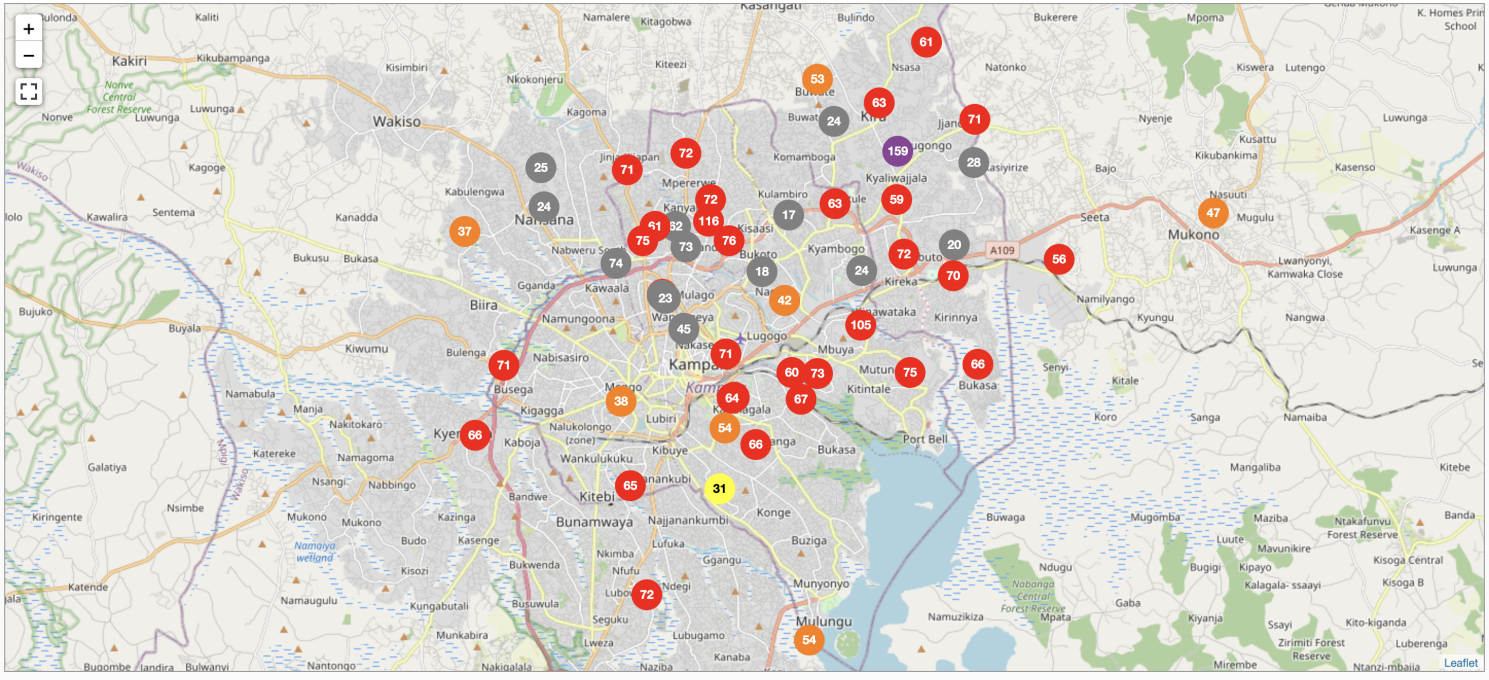


The UK government estimates the annual mortality of human-made air pollution to be 28,000 to 36,000 deaths, and costs UK $\sim$£10^{10}

Kampala and AirQo

- 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





AQI Key

Good (0-12)	Moderate (12.1 - 35.4)	Unhealthy for sensitive	Unhealthy (55.5 - 150.4)	Very Unhealthy (150.5 -	Hazardous (250.5 - 500.4)

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

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

Here $f(x, t)$ represents the pollution source.

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

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

Here $f(x,t)$ represents the pollution source.
Given noisy measurements of pollution levels $z_i = h_i(u) + e_i$ can we infer

- the concentration field $u(x,t)$?
- the source $f(x,t)$?
- ...

General linear systems

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

Linear systems with unknown parameters

Consider

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

where

- $\mathcal{L}$ = linear operator
- f = forcing function.
- u dependent quantity, e.g. pollution concentration.

Finding u given $\mathcal{L}$ and f is the **forward problem**.

Linear systems with unknown parameters

Consider

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

where

- $\mathcal{L}$ = linear operator
- f = forcing function.
- u dependent quantity, e.g. pollution concentration.

Finding u given $\mathcal{L}$ and f is the **forward problem**.

Inverse problem: infer u, f 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:

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

Bayes: find

$$\pi(f|z, \mathcal{L}).$$

Linear systems with unknown parameters

Least squares/maximum-likelihood estimation:

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

Bayes: find

$$\pi(f|z, \mathcal{L}).$$

Adjoints can help in both cases

- We can solve both problems with n simulator evaluations, where n =number of data points.

What is an adjoint?

See Estep 2004

Suppose $\mathcal{U}$ and $\mathcal{V}$ are Hilbert spaces

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

and $\mathcal{L} : \mathcal{U} \mapsto \mathcal{V}$ a linear operator between spaces.

What is an adjoint?

See Estep 2004

Suppose $\mathcal{U}$ and $\mathcal{V}$ are Hilbert spaces

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

and $\mathcal{L} : \mathcal{U} \mapsto \mathcal{V}$ a linear operator between spaces.

Reisz representation theorem: any bounded linear functional on $\mathcal{V}$, v^* say, can be written as

$$v^*(\cdot) = \langle \cdot, v \rangle \quad \text{for some } v \in \mathcal{V}$$

What is an adjoint?

See Estep 2004

Suppose $\mathcal{U}$ and $\mathcal{V}$ are Hilbert spaces

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

and $\mathcal{L} : \mathcal{U} \mapsto \mathcal{V}$ a linear operator between spaces.

Reisz representation theorem: any bounded linear functional on $\mathcal{V}$, v^* say, can be written as

$$v^*(\cdot) = \langle \cdot, v \rangle \quad \text{for some } v \in \mathcal{V}$$

Define $F : \mathcal{U} \to \mathbb{R}$ by

$$F : u \mapsto \langle \mathcal{L}u, v \rangle_{\mathcal{V}}.$$

What is an adjoint?

See Estep 2004

Suppose $\mathcal{U}$ and $\mathcal{V}$ are Hilbert spaces

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

and $\mathcal{L} : \mathcal{U} \mapsto \mathcal{V}$ a linear operator between spaces.

Reisz representation theorem: any bounded linear functional on $\mathcal{V}$, v^* say, can be written as

$$v^*(\cdot) = \langle \cdot, v \rangle \quad \text{for some } v \in \mathcal{V}$$

Define $F : \mathcal{U} \to \mathbb{R}$ by

$$F : u \mapsto \langle \mathcal{L}u, v \rangle_{\mathcal{V}}.$$

F is a bounded linear functional on $\mathcal{U}$, thus $F(\cdot) = \langle \cdot, u \rangle_{\mathcal{U}}$ for some $u \in \mathcal{U}$.

What is an adjoint?

See Estep 2004

Suppose $\mathcal{U}$ and $\mathcal{V}$ are Hilbert spaces

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

and $\mathcal{L} : \mathcal{U} \mapsto \mathcal{V}$ a linear operator between spaces.

Reisz representation theorem: any bounded linear functional on $\mathcal{V}$, v^* say, can be written as

$$v^*(\cdot) = \langle \cdot, v \rangle \quad \text{for some } v \in \mathcal{V}$$

Define $F : \mathcal{U} \to \mathbb{R}$ by

$$F : u \mapsto \langle \mathcal{L}u, v \rangle_{\mathcal{V}}.$$

F is a bounded linear functional on $\mathcal{U}$, thus $F(\cdot) = \langle \cdot, u \rangle_{\mathcal{U}}$ 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

Suppose $\mathcal{U}$ and $\mathcal{V}$ are Hilbert spaces

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

and $\mathcal{L} : \mathcal{U} \mapsto \mathcal{V}$ a linear operator between spaces.

Reisz representation theorem: any bounded linear functional on $\mathcal{V}$, v^* say, can be written as

$$v^*(\cdot) = \langle \cdot, v \rangle \quad \text{for some } v \in \mathcal{V}$$

Define $F : \mathcal{U} \to \mathbb{R}$ by

$$F : u \mapsto \langle \mathcal{L}u, v \rangle_{\mathcal{V}}.$$

F is a bounded linear functional on $\mathcal{U}$, thus $F(\cdot) = \langle \cdot, u \rangle_{\mathcal{U}}$ 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

$$\langle \mathcal{L}u, v \rangle = \langle u, \mathcal{L}^* v \rangle \text{ the '**bilinear identity**'}$$

Example 0

In the finite dimensional case,

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

Example 0

In the finite dimensional case,

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

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

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

The complete observation vector z can then be written as

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

Thus

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

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

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

Thus

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

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

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

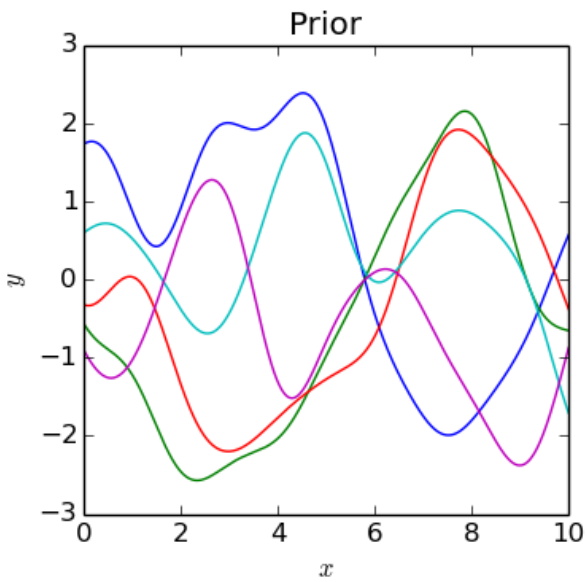
Gaussian Processes

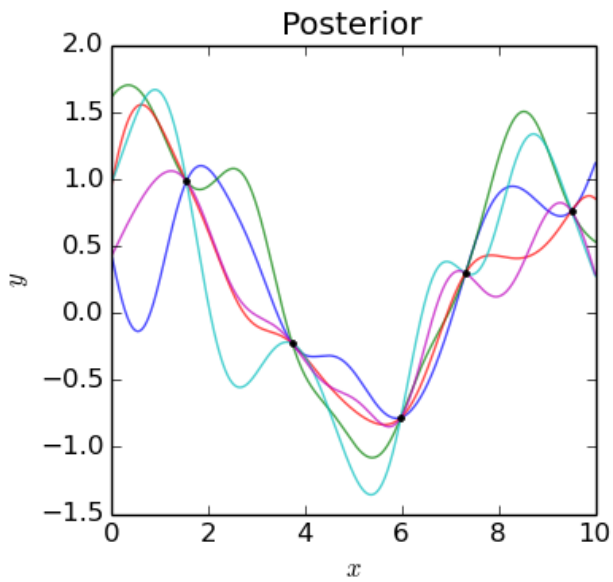
Suppose we model unknown function $f = \{f(x) : x \in \mathcal{X}\}$ as a Gaussian process (GP)

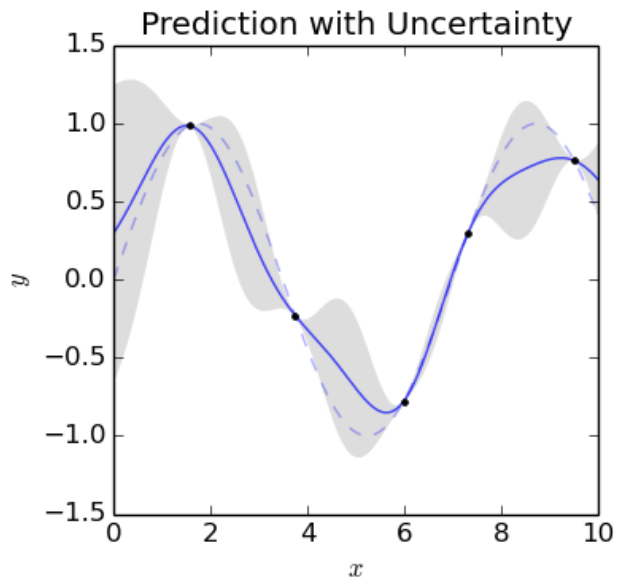
$$f \sim GP(m, k)$$

where we need to specify the prior mean and covariance functions

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







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 finite dimensional linear model.

Choice of basis

$$f(x) = \sum_{j=1}^{\infty} q_i \phi_i(x)$$

- **Random Fourier features**: $\phi_i(x) = cos(w_i x + b_i)$ where $w_i, b_i \sim p(\cdot)$

Choice of basis

$$f(x) = \sum_{j=1}^{\infty} q_i \phi_i(x)$$

- **Random Fourier features**: $\phi_i(x) = cos(w_i x + b_i)$ where $w_i, b_i \sim p(\cdot)$
- **Mercer basis**: $\phi_i(x) = \lambda_i \psi(x)$ where $\lambda_i, \phi_i(\cdot)$ are eigenpairs of

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

- **Laplacian basis**: useful for non-Euclidean domains...

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

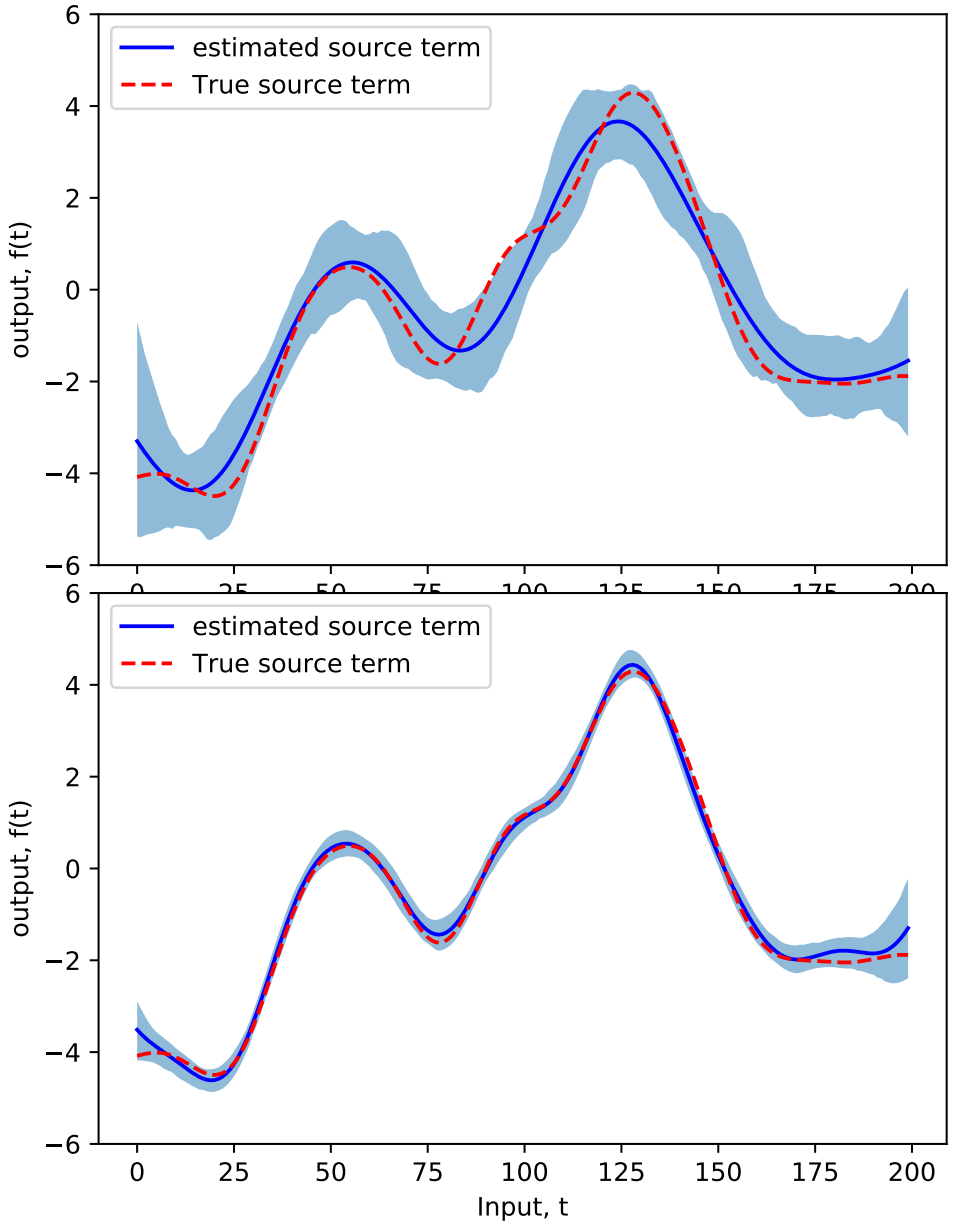
$$-D\ddot{u} + \nu\dot{u} + u = f(t) \quad \text{with } u(0) = \dot{u}(0) = 0.$$

Use the bilinear identity to find the adjoint of

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

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

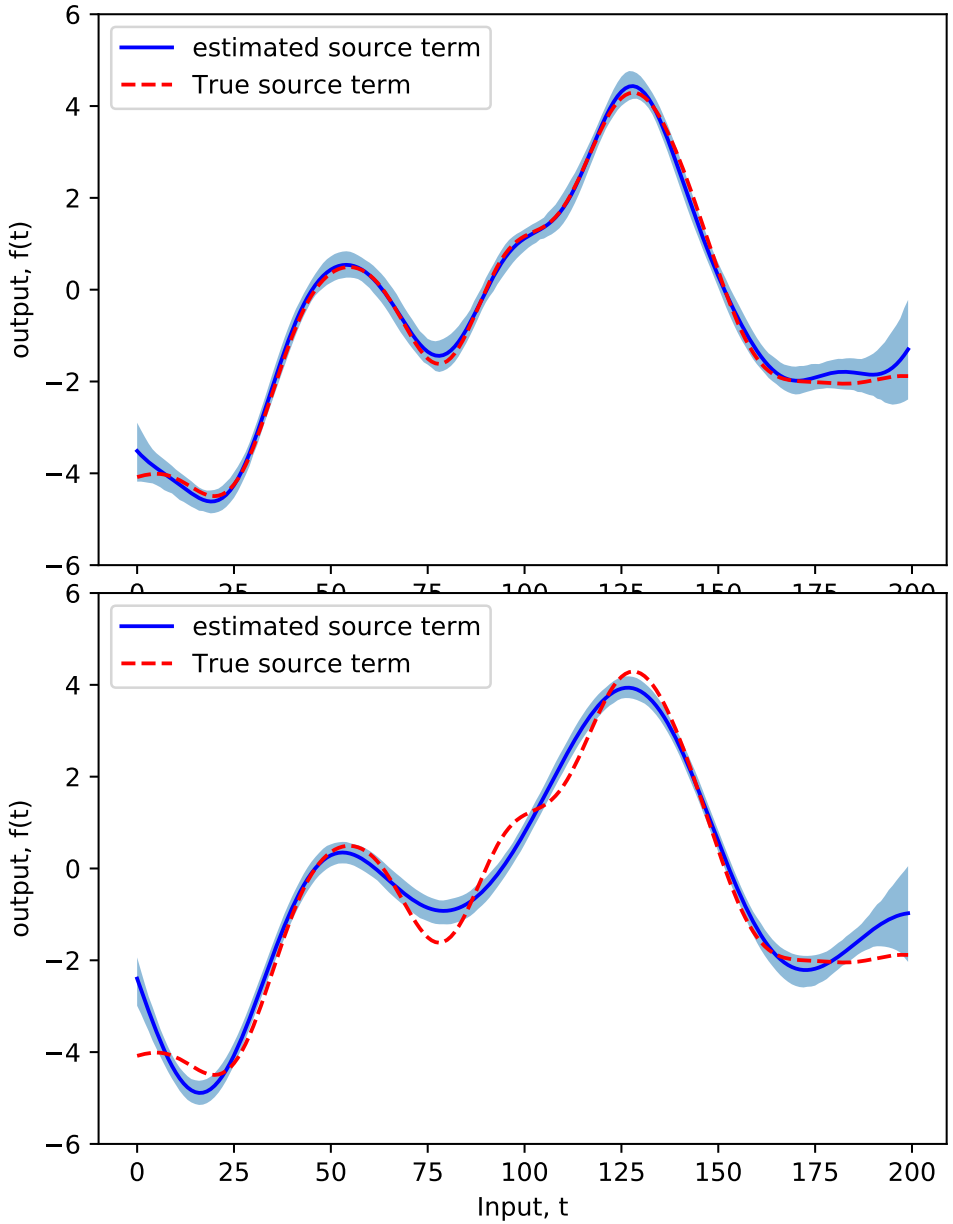
Example 1: Posterior mean and 95% CI (blue), true (red)



- top: $n = 10$ data points, $M = 100$ basis vectors
- bottom: $n = 100$ and $M = 100$

Results required 10 and 100 ODE solves respectively.

Example 1: Too few features

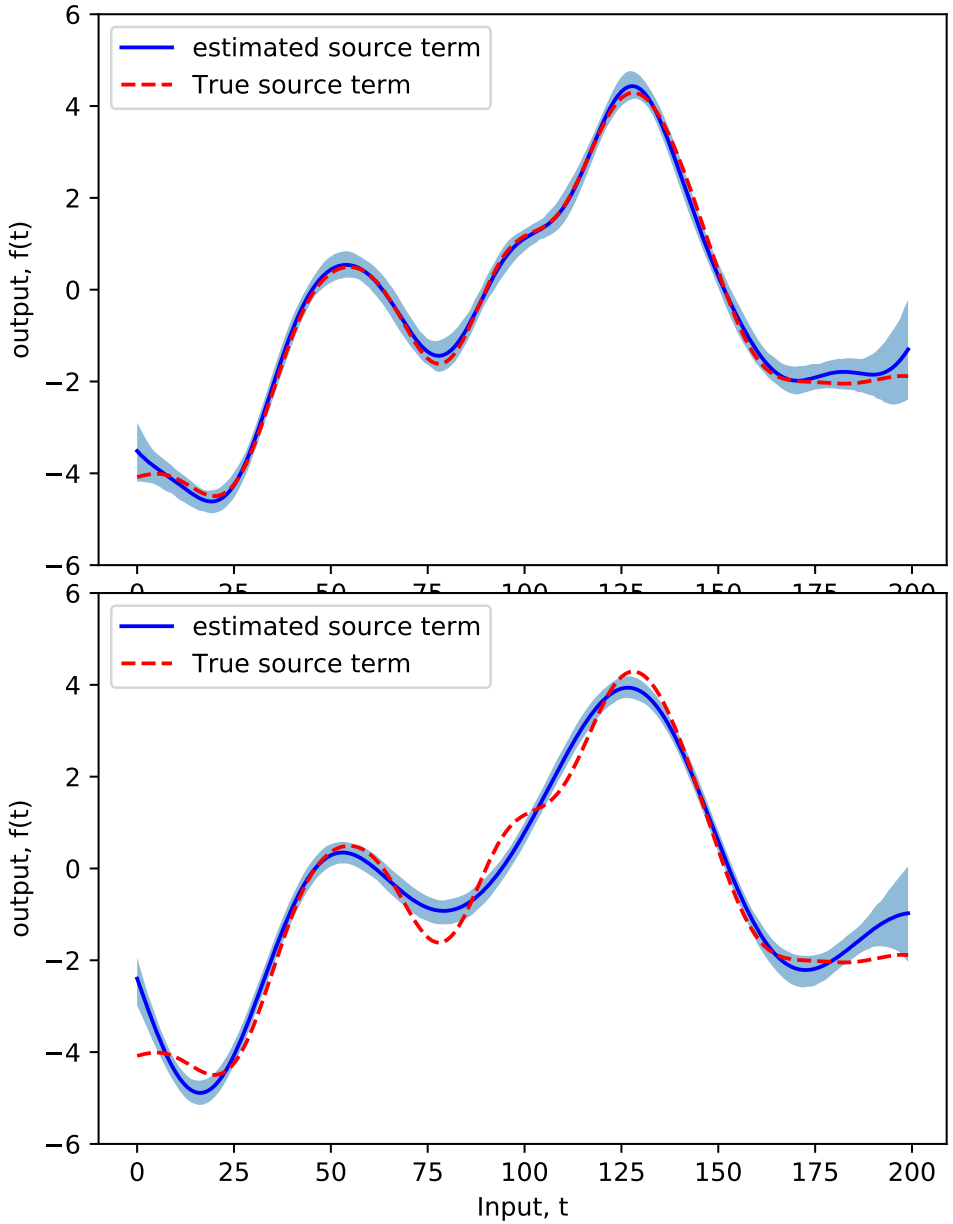


$n = 100$ data points

- top: $M = 100$ basis vectors
- bottom: $M = 10$

NB: overconfident and wrong when $M = 10$ - misspecified model!

Example 1: Too few features



$n = 100$ data points

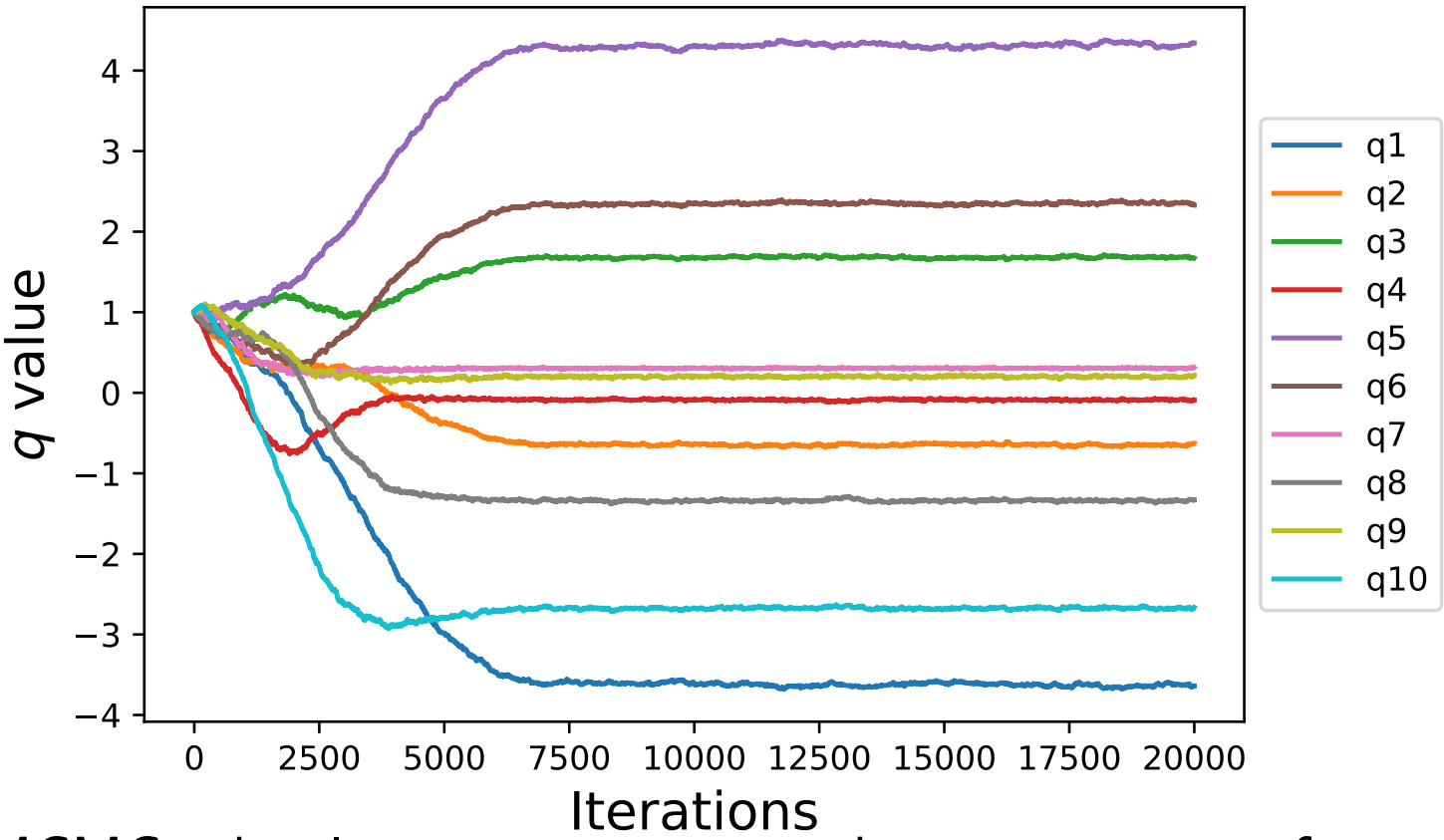
- top: $M = 100$ basis vectors
- bottom: $M = 10$

NB: overconfident and wrong when $M = 10$ - misspecified model!

We need to include enough features to have sufficient modelling flexibility.

Using additional features doesn't require additional ODE solves.

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 2: 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 2: 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 q given $z_i = \langle h_i, u \rangle + N(0, \sigma)$.

Example 2: 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 q 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

The adjoint system is again 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 2: PDE adjoint

The adjoint system is again 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 2: PDE adjoint

The adjoint system is again 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

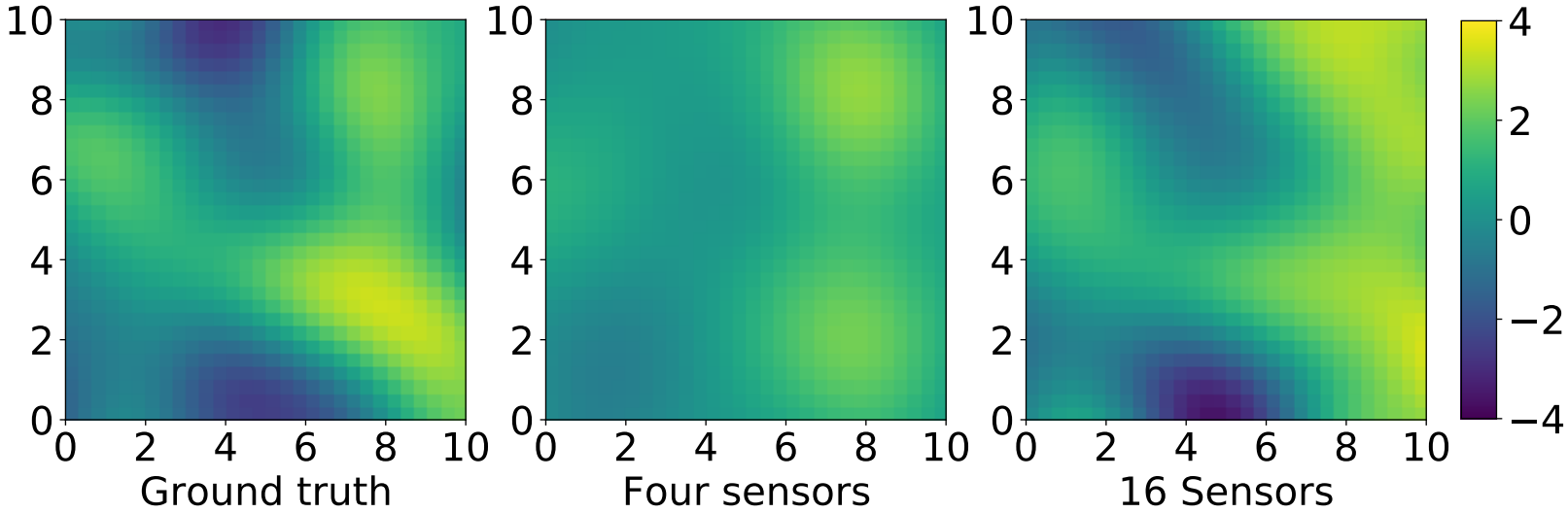
$$u(x, 0) = 0 \text{ for } x \in \mathcal{X} \text{ and } \nabla_n u = 0 \text{ for } x \in \partial\mathcal{X}$$

then the final and boundary conditions on the adjoint system are

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

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

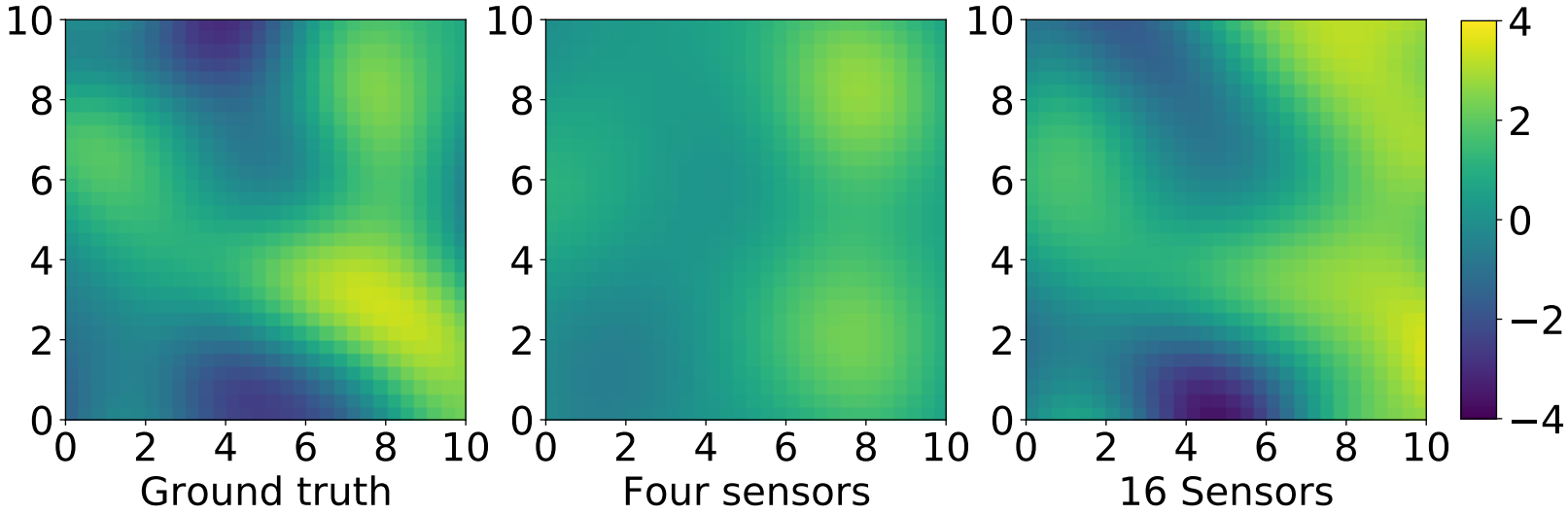
Posterior mean of time slice $u(x, 5)$ - more sensors, improved estimates!



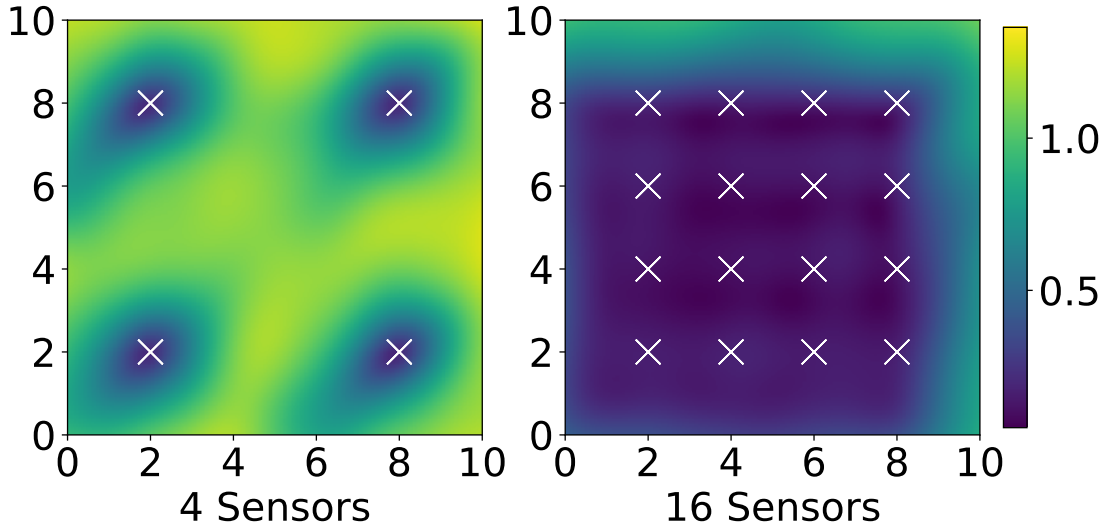
Variance of $u(x, 5)$: Wind from the south west.

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.



Conclusions

Adjoints of linear systems

- an intrusive method; development does require some work but can be automated
- Requires n adjoint solves to infer the posterior
  - essentially insensitive to the number of basis functions used
  - In contrast, MCMC requires a typically an *a priori* unknown number of simulations (but is largely independent of n).
- 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
  - Multi-level approaches
  - Gradient based optimization
  - Sequential data

Ref: Gahungu et al. NeurIPS 2022, plus forthcoming pre-prints.

Conclusions

Adjoints of linear systems

- an intrusive method; development does require some work but can be automated
- Requires n adjoint solves to infer the posterior
  - essentially insensitive to the number of basis functions used
  - In contrast, MCMC requires a typically an *a priori* unknown number of simulations (but is largely independent of n).
- 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
  - Multi-level approaches
  - Gradient based optimization
  - Sequential data

Ref: Gahungu et al. NeurIPS 2022, plus forthcoming pre-prints.

Thank you for listening!

Link to Green's function approach

Consider the linear system

$$\mathcal{L}u = f \qquad \text{for } x \in \Omega$$

The Green's function for this system, $G_y(x)$, satisfies

$$\mathcal{L}^* G_y(x) = \delta_y(x) \qquad \text{for } x \in \Omega$$

Link to Green's function approach

Consider the linear system

$$\mathcal{L}u = f \qquad \text{for } x \in \Omega$$

The Green's function for this system, $G_y(x)$, satisfies

$$\mathcal{L}^* G_y(x) = \delta_y(x) \qquad \text{for } x \in \Omega$$

Solution of the original problem is found by computing the convolution of G with f:

$$\begin{aligned} u(y) &= \langle \delta_y, u \rangle = \langle \mathcal{L}^* G_y,\ u \rangle \\ &= \langle G_y,\ \mathcal{L}u \rangle = \langle G_y,\ f \rangle = \int G_y(x) f(x) \mathrm{d}x. \end{aligned}$$

Link to Green's function approach

Consider the linear system

$$\mathcal{L}u = f \qquad \text{for } x \in \Omega$$

The Green's function for this system, $G_y(x)$, satisfies

$$\mathcal{L}^* G_y(x) = \delta_y(x) \qquad \text{for } x \in \Omega$$

Solution of the original problem is found by computing the convolution of G with f:

$$\begin{aligned} u(y) &= \langle \delta_y, u \rangle = \langle \mathcal{L}^* G_y,\ u \rangle \\ &= \langle G_y,\ \mathcal{L}u \rangle = \langle G_y,\ f \rangle = \int G_y(x) f(x) \mathrm{d}x. \end{aligned}$$

If $f \sim GP(0, k)$, then u is also distributed as a Gaussian process,

$$u \sim GP(0, k_u)$$

with covariance function

$$k_u(y, y') = \int G_y(x) \int G_{y'}(x') k(x, x') \mathrm{d}x' \mathrm{d}x.$$

$$k_u(y, y') = \int G_y(x) \int G_{y'}(x') k(x, x') \mathrm{d}x' \mathrm{d}x.$$

If G is known then sometimes it is possible to compute this analytically. Otherwise numerical methods must be used.

- Likely to be cheaper than the adjoint approach

$$k_u(y, y') = \int G_y(x) \int G_{y'}(x') k(x, x') \mathrm{d}x' \mathrm{d}x.$$

If G is known then sometimes it is possible to compute this analytically. Otherwise numerical methods must be used.

- Likely to be cheaper than the adjoint approach

If G is unknown, then need to approximate G before approximating the integral....

- Expensive, unstable...
- Poorly developed

$$k_u(y, y') = \int G_y(x) \int G_{y'}(x') k(x, x') \mathrm{d}x' \mathrm{d}x.$$

If G is known then sometimes it is possible to compute this analytically. Otherwise numerical methods must be used.

- Likely to be cheaper than the adjoint approach

If G is unknown, then need to approximate G before approximating the integral....

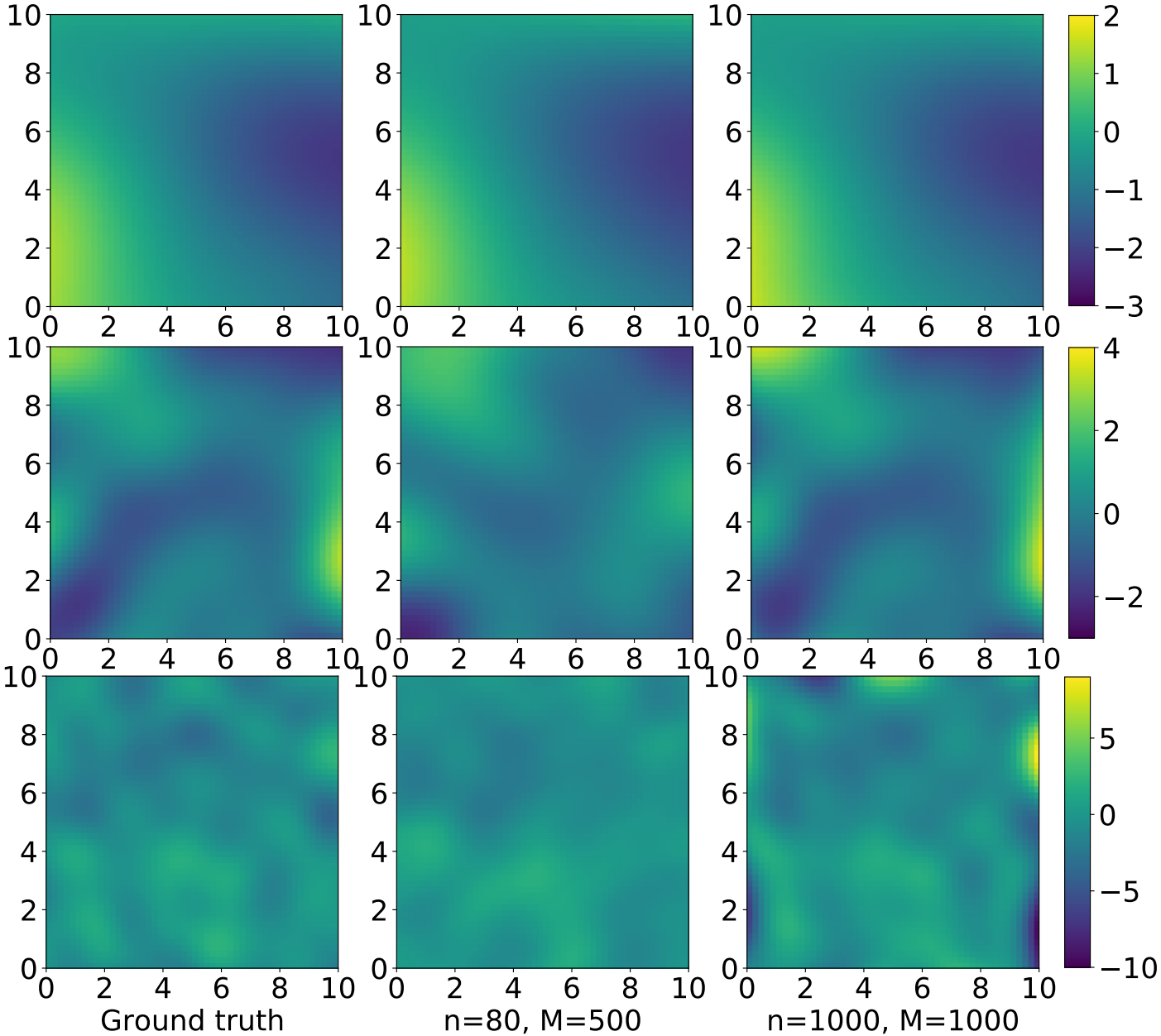
- Expensive, unstable...
- Poorly developed

In contrast, our approach relies on

- existence of the adjoint operator $\mathcal{L}^*$
- ability to solve adjoint systems numerically - deploy modern finite element solvers (efficient, stable, and offer good error-control).

Recommendation: Use Green's function approach only when G known and covariance integral tractable.

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

Example 2: Results

Mean square error vs number of features and sensors

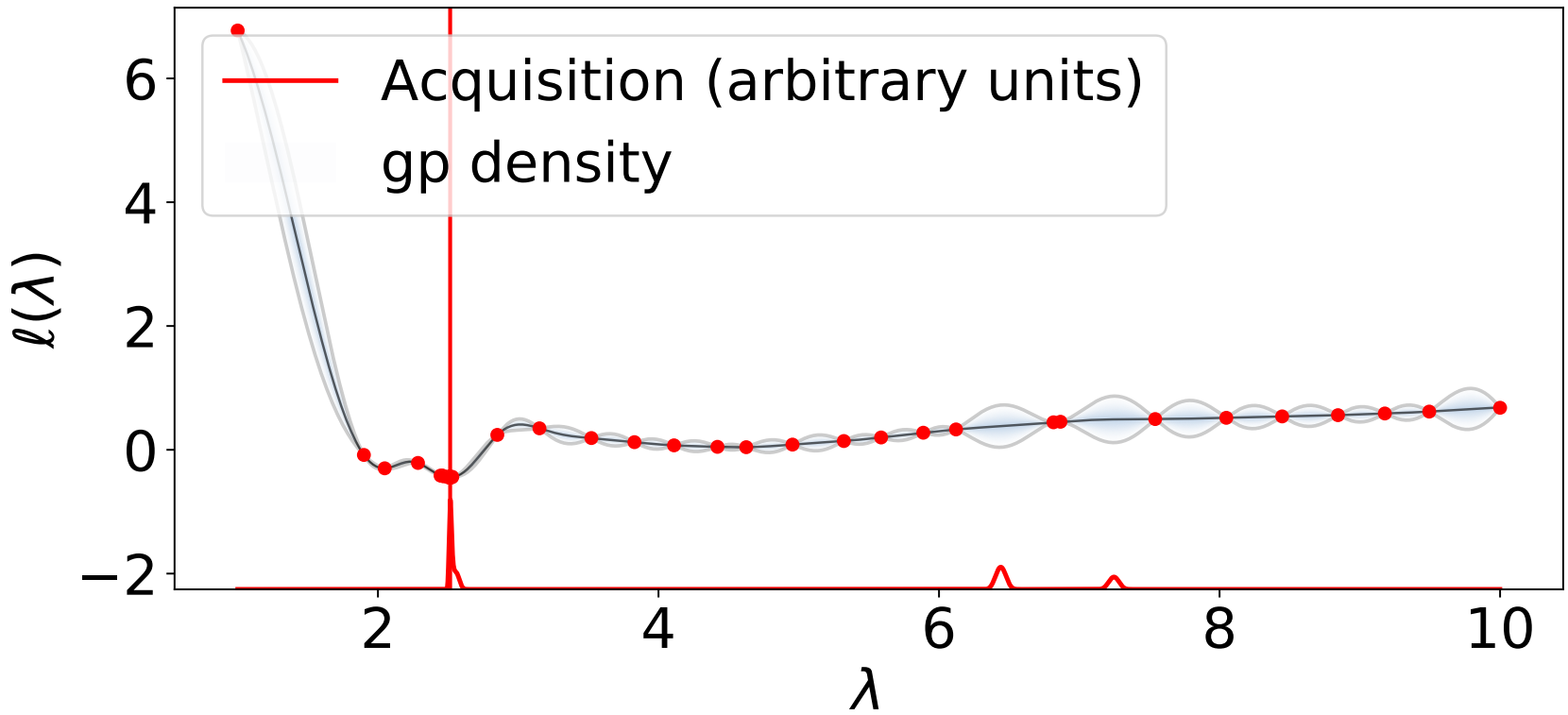
Median MSE as a function of number of sensors and RFFs.

Sensors	Features				
	10	50	100	200	300
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
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
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
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
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
25 (MH)	3.27 (1.73,6.12)	-	-	-	-

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

Non-linear parameter estimation

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



Preprint showing how to use the adjoint sensitivity soon....

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).