The numerical challenges of moving beyond "Uncertainty Quantification" and towards "Statistics"

Daniel Simpson

Department of Mathematical Sciences
University of Bath

Outline

Me. I am Mariah... The Elusive Chanteuse

Who am I?

- My PhD was on Krylov methods computing matrix functions
- One of the main applications was to statistics
- (Where you sometimes need to compute matrix functions!)
- It all went horribly wrong and I went native
- Now I'm a statistician...
- The types of problems I'm interested in solving "happen to be" challenging numerically as well as statistically.

Me. I am Mariah... The Elusive Chanteuse

Who am I?

- My PhD was on Krylov methods computing matrix functions
- One of the main applications was to statistics
- (Where you sometimes need to compute matrix functions!)
- It all went horribly wrong and I went native
- Now I'm a statistician...
- The types of problems I'm interested in solving "happen to be" challenging numerically as well as statistically.

Me. I am Mariah... The Elusive Chanteuse

Who am I?

- My PhD was on Krylov methods computing matrix functions
- One of the main applications was to statistics
- (Where you sometimes need to compute matrix functions!)
- It all went horribly wrong and I went native
- Now I'm a statistician...
- The types of problems I'm interested in solving "happen to be" challenging numerically as well as statistically.

Me. I am Mariah... The Elusive Chanteuse

Who am I?

- My PhD was on Krylov methods computing matrix functions
- One of the main applications was to statistics
- (Where you sometimes need to compute matrix functions!)
- It all went horribly wrong and I went native
- Now I'm a statistician...
- The types of problems I'm interested in solving "happen to be" challenging numerically as well as statistically.

Me. I am Mariah. . . The Elusive Chanteuse

Who am I?

- My PhD was on Krylov methods computing matrix functions
- One of the main applications was to statistics
- (Where you sometimes need to compute matrix functions!)
- It all went horribly wrong and I went native
- Now I'm a statistician...
- The types of problems I'm interested in solving "happen to be" challenging numerically as well as statistically.

Me. I am Mariah... The Elusive Chanteuse

Who am I?

- My PhD was on Krylov methods computing matrix functions
- One of the main applications was to statistics
- (Where you sometimes need to compute matrix functions!)
- It all went horribly wrong and I went native
- Now I'm a statistician...
- The types of problems I'm interested in solving "happen to be" challenging numerically as well as statistically.

(with apologies to Jim Jones)

So why am I here?

- Over the last decade or so, applied mathematicians have realised that there is more than matching their models to data than `fminsearch`
- The field of "uncertainty quantification" has bloomed
- Uncertainty quantification is statistics done by other people
- (for the purpose of this talk, Forward UQ doesn't exist)
- The problems that I'm talking about in this talk haven't been really hit by mainstream UQ, but they're coming...

(with apologies to Jim Jones)

So why am I here?

- Over the last decade or so, applied mathematicians have realised that there is more than matching their models to data than `fminsearch`
- The field of "uncertainty quantification" has bloomed
- Uncertainty quantification is statistics done by other people
- (for the purpose of this talk, Forward UQ doesn't exist)
- The problems that I'm talking about in this talk haven't been really hit by mainstream UQ, but they're coming...

(with apologies to Jim Jones)

So why am I here?

- Over the last decade or so, applied mathematicians have realised that there is more than matching their models to data than `fminsearch`
- The field of "uncertainty quantification" has bloomed
- Uncertainty quantification is statistics done by other people
- (for the purpose of this talk, Forward UQ doesn't exist)
- The problems that I'm talking about in this talk haven't been really hit by mainstream UQ, but they're coming...

(with apologies to Jim Jones)

So why am I here?

- Over the last decade or so, applied mathematicians have realised that there is more than matching their models to data than `fminsearch`
- The field of "uncertainty quantification" has bloomed
- Uncertainty quantification is statistics done by other people
- (for the purpose of this talk, Forward UQ doesn't exist)
- The problems that I'm talking about in this talk haven't been really hit by mainstream UQ, but they're coming...

(with apologies to Jim Jones)

So why am I here?

- Over the last decade or so, applied mathematicians have realised that there is more than matching their models to data than `fminsearch`
- The field of "uncertainty quantification" has bloomed
- Uncertainty quantification is statistics done by other people
- (for the purpose of this talk, Forward UQ doesn't exist)
- The problems that I'm talking about in this talk haven't been really hit by mainstream UQ, but they're coming...

What's the opposite of hagiography?

Today's question

Is there a role for iterative linear algebra methods in Bayesian statistics?

- We will see that Cholesky factorisations are invaluable
- Can we replace them with iterative methods?
- Currently, no.
- But today's talk is about the problems we need to solve and a catalogue of failed attempts to replace direct numerical methods.

What's the opposite of hagiography?

Today's question

Is there a role for iterative linear algebra methods in Bayesian statistics?

- We will see that Cholesky factorisations are invaluable
- Can we replace them with iterative methods?
- Currently, no.
- But today's talk is about the problems we need to solve and a catalogue of failed attempts to replace direct numerical methods.

What's the opposite of hagiography?

Today's question

Is there a role for iterative linear algebra methods in Bayesian statistics?

- We will see that Cholesky factorisations are invaluable
- Can we replace them with iterative methods?
- Currently, no.
- But today's talk is about the problems we need to solve and a catalogue of failed attempts to replace direct numerical methods.

What's the opposite of hagiography?

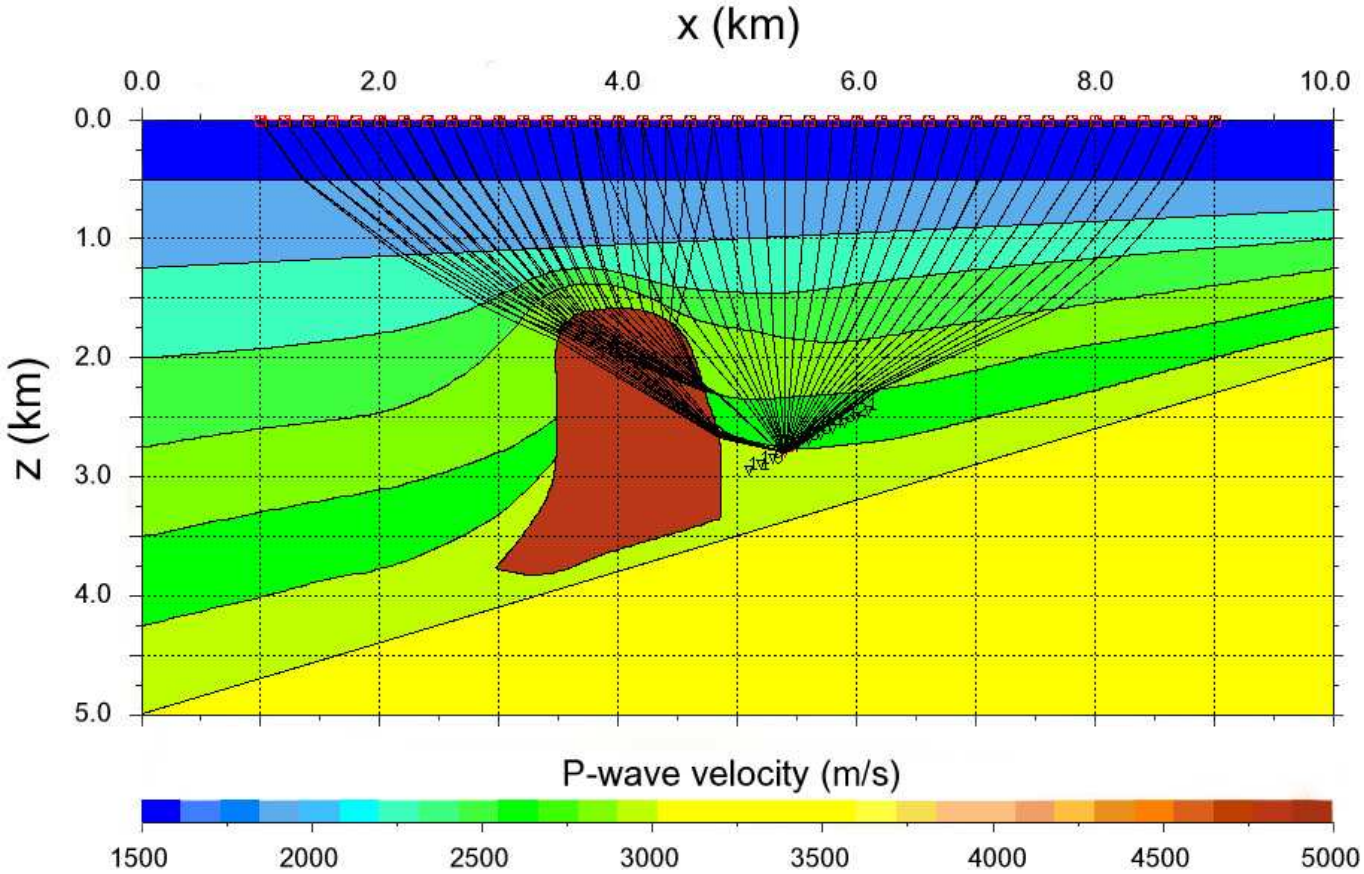
Today's question

Is there a role for iterative linear algebra methods in Bayesian statistics?

- We will see that Cholesky factorisations are invaluable
- Can we replace them with iterative methods?
- Currently, no.
- But today's talk is about the problems we need to solve and a catalogue of failed attempts to replace direct numerical methods.

Outline

Uncertainty quantification



Uncertainty quantification

- Data: y_i, $i = 1, \ldots, N$
- Assume $y_i \sim N(\mu_i, \sigma^2)$
- (probably assume σ is "known")
- Assume the mean is of the form $\mu_i = F(u(s_i; \log(kappa)))$ for some functional F
- Assume $u(\cdot; \kappa)$ is the solution to (e.g.) a PDE, e.g.

$$\nabla \cdot (\kappa(s) \nabla x(s)) = 0, \qquad + \text{B.C.s}$$

- Assume a prior model on $\kappa(\cdot)$, e.g.

$$x(\cdot) := \log(\kappa(\cdot)) \sim GP(0, c(\cdot, \cdot)).$$

Aim: Try to update the information about $\kappa(\cdot)$ in light of the observed measurements $\boldsymbol{y}$.

Uncertainty quantification

- Data: y_i, $i = 1, \ldots, N$
- Assume $y_i \sim N(\mu_i, \sigma^2)$
- (probably assume σ is "known")
- Assume the mean is of the form $\mu_i = F(u(s_i; \log(kappa)))$ for some functional F
- Assume $u(\cdot; \kappa)$ is the solution to (e.g.) a PDE, e.g.

$$\nabla \cdot (\kappa(s) \nabla x(s)) = 0, \qquad + \text{B.C.s}$$

- Assume a prior model on $\kappa(\cdot)$, e.g.

$$x(\cdot) := \log(\kappa(\cdot)) \sim GP(0, c(\cdot, \cdot)).$$

Aim: Try to update the information about $\kappa(\cdot)$ in light of the observed measurements $\boldsymbol{y}$.

Uncertainty quantification

- Data: y_i, $i = 1, \ldots, N$
- Assume $y_i \sim N(\mu_i, \sigma^2)$
- (probably assume σ is "known")
- Assume the mean is of the form $\mu_i = F(u(s_i; \log(kappa)))$ for some functional F
- Assume $u(\cdot; \kappa)$ is the solution to (e.g.) a PDE, e.g.

$$\nabla \cdot (\kappa(s) \nabla x(s)) = 0, \qquad + \text{B.C.s}$$

- Assume a prior model on $\kappa(\cdot)$, e.g.

$$x(\cdot) := \log(\kappa(\cdot)) \sim GP(0, c(\cdot, \cdot)).$$

Aim: Try to update the information about $\kappa(\cdot)$ in light of the observed measurements $\boldsymbol{y}$.

Uncertainty quantification

- Data: y_i, $i = 1, \ldots, N$
- Assume $y_i \sim N(\mu_i, \sigma^2)$
- (probably assume σ is "known")
- Assume the mean is of the form $\mu_i = F(u(s_i; \log(kappa)))$ for some functional F
- Assume $u(\cdot; \kappa)$ is the solution to (e.g.) a PDE, e.g.

$$\nabla \cdot (\kappa(s) \nabla x(s)) = 0, \qquad + \text{B.C.s}$$

- Assume a prior model on $\kappa(\cdot)$, e.g.

$$x(\cdot) := \log(\kappa(\cdot)) \sim GP(0, c(\cdot, \cdot)).$$

Aim: Try to update the information about $\kappa(\cdot)$ in light of the observed measurements $\boldsymbol{y}$.

Uncertainty quantification

- Data: y_i, $i = 1, \ldots, N$
- Assume $y_i \sim N(\mu_i, \sigma^2)$
- (probably assume σ is "known")
- Assume the mean is of the form $\mu_i = F(u(s_i; \log(kappa)))$ for some functional F
- Assume $u(\cdot; \kappa)$ is the solution to (e.g.) a PDE, e.g.

$$\nabla \cdot (\kappa(s) \nabla x(s)) = 0, \qquad + \text{B.C.s}$$

- Assume a prior model on $\kappa(\cdot)$, e.g.

$$x(\cdot) := \log(\kappa(\cdot)) \sim GP(0, c(\cdot, \cdot)).$$

Aim: Try to update the information about $\kappa(\cdot)$ in light of the observed measurements $\boldsymbol{y}$.

Uncertainty quantification

- Data: y_i, $i = 1, \ldots, N$
- Assume $y_i \sim N(\mu_i, \sigma^2)$
- (probably assume σ is "known")
- Assume the mean is of the form $\mu_i = F(u(s_i; \log(kappa)))$ for some functional F
- Assume $u(\cdot; \kappa)$ is the solution to (e.g.) a PDE, e.g.

$$\nabla \cdot (\kappa(s) \nabla x(s)) = 0, \qquad + \text{B.C.s}$$

- Assume a prior model on $\kappa(\cdot)$, e.g.

$$x(\cdot) := \log(\kappa(\cdot)) \sim GP(0, c(\cdot, \cdot)).$$

Aim: Try to update the information about $\kappa(\cdot)$ in light of the observed measurements $\boldsymbol{y}$.

So how isn't this statistics?

- By any reasonable definition, (backward) UQ is statistics
- In fact, most statistical problems look pretty much identical to this
- With the exception that it is traditional for the function $x(\cdot) \to F(u(s_i; x))$ to be local
- Here, as it involves the solution to a PDE, it is non-local.
- The real difference is that the mapping $x(\cdot) \to F(u(s_i; x)$ is really expensive to compute!

So how do we update $\log(\kappa(\cdot))$?

Classical approach: Tikhonov regularisation

$$\kappa(\cdot) = \arg\min \|y_i - F(u(s_i; x))\|_2^2 + \lambda \|x\|_H^2$$

- Not a bad idea
- λ balances the fidelity to the data (**mean square error**) with the complexity of the model
- Relatively straightforward to solve!
- The Hilbert space H is chosen for computational tractability
- This gives a single value of x. It doesn't directly show how uncertain we are about this value.
- This makes it of limited use when the object of inference is not the function x, but rather some decision.

A simpler problem

Consider the problem of recovering the mean of an p-dimensional multivariate normal distribution $N(\mu, \boldsymbol{I})$ from a sample $y \sim N(\mu, \boldsymbol{I})$.

- The natural way to measure the "goodness" of an estimate is the mean squared error $e^2(\hat{\mu}(y)) = \mathbb{E}_y \left(\|\mu - \hat{\mu}(y)\|^2 \right)$
- The natural estimator is $\hat{\mu}(y) = y$ (the sample mean)
- **Fact:** The estimator

$$\left(1 - \frac{(m-2)}{\|y\|^2}\right) y$$

always has lower mean-squared error.

- In fact, that estimator can also be uniformly beaten!

A simpler problem

Consider the problem of recovering the mean of an p-dimensional multivariate normal distribution $N(\mu, \boldsymbol{I})$ from a sample $y \sim N(\mu, \boldsymbol{I})$.

- The natural way to measure the "goodness" of an estimate is the mean squared error $e^2(\hat{\mu}(y)) = \mathbb{E}_y \left(\|\mu - \hat{\mu}(y)\|^2 \right)$
- The natural estimator is $\hat{\mu}(y) = y$ (the sample mean)
- **Fact:** The estimator

$$\left(1 - \frac{(m-2)}{\|y\|^2}\right) y$$

always has lower mean-squared error.

- In fact, that estimator can also be uniformly beaten!

A simpler problem

Consider the problem of recovering the mean of an p-dimensional multivariate normal distribution $N(\mu, \boldsymbol{I})$ from a sample $y \sim N(\mu, \boldsymbol{I})$.

- The natural way to measure the "goodness" of an estimate is the mean squared error $e^2(\hat{\mu}(y)) = \mathbb{E}_y \left(\|\mu - \hat{\mu}(y)\|^2 \right)$
- The natural estimator is $\hat{\mu}(y) = y$ (the sample mean)
- **Fact:** The estimator

$$\left(1 - \frac{(m-2)}{\|y\|^2}\right) y$$

  always has lower mean-squared error.
- In fact, that estimator can also be uniformly beaten!

A simpler problem

Consider the problem of recovering the mean of an p-dimensional multivariate normal distribution $N(\mu, \boldsymbol{I})$ from a sample $y \sim N(\mu, \boldsymbol{I})$.

- The natural way to measure the "goodness" of an estimate is the mean squared error $e^2(\hat{\mu}(y)) = \mathbb{E}_y \left(\|\mu - \hat{\mu}(y)\|^2 \right)$
- The natural estimator is $\hat{\mu}(y) = y$ (the sample mean)
- **Fact:** The estimator

$$\left(1 - \frac{(m-2)}{\|y\|^2}\right) y$$

  always has lower mean-squared error.
- In fact, that estimator can also be uniformly beaten!

So how do you estimate a mean?

Consider the model

$$
\begin{aligned}
y \mid \mu &\sim N(\mu, \boldsymbol{I}) \\
\mu \mid \sigma &\sim N(0, \sigma^2 \boldsymbol{I}) \\
\sigma &\sim \pi(\sigma)
\end{aligned}
$$

- if σ^2 is fixed, then $\hat{\mu}_\sigma(y) = (1 - (1 + \sigma^2)^{-1})y$
- If we allow for an unknown, random σ^2, we get an estimator if the form $\hat{\mu}(y) = \left(1 - \mathbb{E}_{\sigma|y}(1 + \sigma^2)^{-1})\right) y$.
- Every admissible estimator is of this form!

So how do you estimate a mean?

Consider the model

$$
\begin{aligned}
y \mid \mu &\sim N(\mu, \boldsymbol{I}) \\
\mu \mid \sigma &\sim N(0, \sigma^2 \boldsymbol{I}) \\
\sigma &\sim \pi(\sigma)
\end{aligned}
$$

- if σ^2 is fixed, then $\hat{\mu}_\sigma(y) = (1 - (1 + \sigma^2)^{-1})y$
- If we allow for an unknown, random σ^2, we get an estimator if the form $\hat{\mu}(y) = \left(1 - \mathbb{E}_{\sigma|y}(1 + \sigma^2)^{-1})\right) y$.
- Every admissible estimator is of this form!

So how do you estimate a mean?

Consider the model

$$
\begin{aligned}
y \mid \mu &\sim N(\mu, \boldsymbol{I}) \\
\mu \mid \sigma &\sim N(0, \sigma^2 \boldsymbol{I}) \\
\sigma &\sim \pi(\sigma)
\end{aligned}
$$

- if σ^2 is fixed, then $\hat{\mu}_\sigma(y) = (1 - (1 + \sigma^2)^{-1})y$
- If we allow for an unknown, random σ^2, we get an estimator if the form $\hat{\mu}(y) = \left(1 - \mathbb{E}_{\sigma|y}(1 + \sigma^2)^{-1})\right) y$.
- Every admissible estimator is of this form!

What was the point of that?

Why Bayes?

Bayesian estimators can perform much better than their frequentist counterparts

- This is not always true!
- These improved estimators work by controlling the complexity of the model
- When there is a large signal, these estimators essentially leaves it alone
- When there is a small signal, these estimators strongly shrink it towards zero
- Just using a Gaussian prior on μ is not enough to do this! We need to use a hyperparameter.

What was the point of that?

Why Bayes?

Bayesian estimators can perform much better than their frequentist counterparts

- This is not always true!
- These improved estimators work by controlling the complexity of the model
- When there is a large signal, these estimators essentially leaves it alone
- When there is a small signal, these estimators strongly shrink it towards zero
- Just using a Gaussian prior on μ is not enough to do this! We need to use a hyperparameter.

What was the point of that?

Why Bayes?

Bayesian estimators can perform much better than their frequentist counterparts

- This is not always true!
- These improved estimators work by controlling the complexity of the model
- When there is a large signal, these estimators essentially leaves it alone
- When there is a small signal, these estimators strongly shrink it towards zero
- Just using a Gaussian prior on μ is not enough to do this! We need to use a hyperparameter.

What was the point of that?

Why Bayes?

Bayesian estimators can perform much better than their frequentist counterparts

- This is not always true!
- These improved estimators work by controlling the complexity of the model
- When there is a large signal, these estimators essentially leaves it alone
- When there is a small signal, these estimators strongly shrink it towards zero
- Just using a Gaussian prior on μ is not enough to do this! We need to use a hyperparameter.

What was the point of that?

Why Bayes?

Bayesian estimators can perform much better than their frequentist counterparts

- This is not always true!
- These improved estimators work by controlling the complexity of the model
- When there is a large signal, these estimators essentially leaves it alone
- When there is a small signal, these estimators strongly shrink it towards zero
- Just using a Gaussian prior on μ is not enough to do this! We need to use a hyperparameter.

Back to the inverse problems

So how does this thinking help us?

- Let's abstract away the hard bit!

$$y_i \sim \pi(y \mid x(\cdot), \theta)$$

- The remaining part of the model is a Gaussian process

$$x(\cdot) \mid \theta \sim GP(0, c(\cdot, \cdot); \theta)$$

- That is, the function evaluated at n points is Gaussian, i.e.

$$(x(s_1), \cdots, x(s_n))^T \sim N(\mathbf{0}, \mathbf{\Sigma}(\theta)),$$

where the covariance matrix is given by $\Sigma_{ij} = c(s_i, s_j; \theta)$.

If we let θ be random, we can make good estimators!

Dealing with parameters

- UQ typically computes quantities related to $\pi(x \mid y)$
- If we have structural parameters, we replace that with $\pi(x \mid y, \theta)$
- So to get what we want, we need the posterior $\pi(\theta \mid y)$ because

$$\pi(x \mid y) = \int_{\Theta} \pi(x \mid y, \theta)\pi(\theta \mid y)\, d\theta$$

- This except for linear inverse problems, $\pi(\theta \mid y)$ is analytically intractable...

The Laplace approximation

We can use Bayes' theorem to show that

$$\pi(\theta \mid y) \propto \frac{\pi(y \mid x, \theta)\pi(x \mid \theta)\pi(\theta)}{\pi(x \mid y, \theta)}$$

- We have all of these things!
- Except for $\pi(x \mid y, \theta)$
- But asymptotically, $\pi(x \mid y, \theta)$ is Gaussian!
- **IDEA:** Replace $\pi(x \mid y, \theta)$ with a Gaussian process that matches the first two moments at the mode.
- $\pi_G(x \mid y, \theta) \sim GP(x^*(\theta), \tilde{c}(\cdot, \cdot)(\theta))$, where $x^*(\theta)$ is the solution of the Tykhonov regularisation problem and $\tilde{c}(\cdot, \cdot)(\theta)$ is the "Hessian" at the mode
- The Laplace approximation

$$\tilde{\pi}(\theta \mid y) \propto \frac{\pi(y \mid x^*(\theta), \theta)\pi(x^*(\theta) \mid \theta)\pi(\theta)}{\pi_G(x^*(\theta) \mid y, \theta)}$$

has relative error of $n^{-3/2}$.

The Laplace approximation

We can use Bayes' theorem to show that

$$\pi(\theta \mid y) \propto \frac{\pi(y \mid x, \theta)\pi(x \mid \theta)\pi(\theta)}{\pi(x \mid y, \theta)}$$

- We have all of these things!
- Except for $\pi(x \mid y, \theta)$
- But asymptotically, $\pi(x \mid y, \theta)$ is Gaussian!
- **IDEA:** Replace $\pi(x \mid y, \theta)$ with a Gaussian process that matches the first two moments at the mode.
- $\pi_G(x \mid y, \theta) \sim GP(x^*(\theta), \tilde{c}(\cdot, \cdot)(\theta))$, where $x^*(\theta)$ is the solution of the Tykhonov regularisation problem and $\tilde{c}(\cdot, \cdot)(\theta)$ is the "Hessian" at the mode
- The Laplace approximation

$$\tilde{\pi}(\theta \mid y) \propto \frac{\pi(y \mid x^*(\theta), \theta)\pi(x^*(\theta) \mid \theta)\pi(\theta)}{\pi_G(x^*(\theta) \mid y, \theta)}$$

has relative error of $n^{-3/2}$.

The Laplace approximation

We can use Bayes' theorem to show that

$$\pi(\theta \mid y) \propto \frac{\pi(y \mid x, \theta)\pi(x \mid \theta)\pi(\theta)}{\pi(x \mid y, \theta)}$$

- We have all of these things!
- Except for $\pi(x \mid y, \theta)$
- But asymptotically, $\pi(x \mid y, \theta)$ is Gaussian!
- **IDEA:** Replace $\pi(x \mid y, \theta)$ with a Gaussian process that matches the first two moments at the mode.
- $\pi_G(x \mid y, \theta) \sim GP(x^*(\theta), \tilde{c}(\cdot, \cdot)(\theta))$, where $x^*(\theta)$ is the solution of the Tykhonov regularisation problem and $\tilde{c}(\cdot, \cdot)(\theta)$ is the "Hessian" at the mode
- The Laplace approximation

$$\tilde{\pi}(\theta \mid y) \propto \frac{\pi(y \mid x^*(\theta), \theta)\pi(x^*(\theta) \mid \theta)\pi(\theta)}{\pi_G(x^*(\theta) \mid y, \theta)}$$

has relative error of $n^{-3/2}$.

The Laplace approximation

We can use Bayes' theorem to show that

$$\pi(\theta \mid y) \propto \frac{\pi(y \mid x, \theta)\pi(x \mid \theta)\pi(\theta)}{\pi(x \mid y, \theta)}$$

- We have all of these things!
- Except for $\pi(x \mid y, \theta)$
- But asymptotically, $\pi(x \mid y, \theta)$ is Gaussian!
- **IDEA:** Replace $\pi(x \mid y, \theta)$ with a Gaussian process that matches the first two moments at the mode.
- $\pi_G(x \mid y, \theta) \sim GP(x^*(\theta), \tilde{c}(\cdot, \cdot)(\theta))$, where $x^*(\theta)$ is the solution of the Tykhonov regularisation problem and $\tilde{c}(\cdot, \cdot)(\theta)$ is the "Hessian" at the mode
- The Laplace approximation

$$\tilde{\pi}(\theta \mid y) \propto \frac{\pi(y \mid x^*(\theta), \theta)\pi(x^*(\theta) \mid \theta)\pi(\theta)}{\pi_G(x^*(\theta) \mid y, \theta)}$$

has relative error of $n^{-3/2}$.

The Laplace approximation

We can use Bayes' theorem to show that

$$\pi(\theta \mid y) \propto \frac{\pi(y \mid x, \theta)\pi(x \mid \theta)\pi(\theta)}{\pi(x \mid y, \theta)}$$

- We have all of these things!
- Except for $\pi(x \mid y, \theta)$
- But asymptotically, $\pi(x \mid y, \theta)$ is Gaussian!
- **IDEA:** Replace $\pi(x \mid y, \theta)$ with a Gaussian process that matches the first two moments at the mode.
- $\pi_G(x \mid y, \theta) \sim GP(x^*(\theta), \tilde{c}(\cdot, \cdot)(\theta))$, where $x^*(\theta)$ is the solution of the Tykhonov regularisation problem and $\tilde{c}(\cdot, \cdot)(\theta)$ is the "Hessian" at the mode
- The Laplace approximation

$$\tilde{\pi}(\theta \mid y) \propto \frac{\pi(y \mid x^*(\theta), \theta)\pi(x^*(\theta) \mid \theta)\pi(\theta)}{\pi_G(x^*(\theta) \mid y, \theta)}$$

has relative error of $n^{-3/2}$.

The Laplace approximation

We can use Bayes' theorem to show that

$$\pi(\theta \mid y) \propto \frac{\pi(y \mid x, \theta)\pi(x \mid \theta)\pi(\theta)}{\pi(x \mid y, \theta)}$$

- We have all of these things!
- Except for $\pi(x \mid y, \theta)$
- But asymptotically, $\pi(x \mid y, \theta)$ is Gaussian!
- **IDEA:** Replace $\pi(x \mid y, \theta)$ with a Gaussian process that matches the first two moments at the mode.
- $\pi_G(x \mid y, \theta) \sim GP(x^*(\theta), \tilde{c}(\cdot, \cdot)(\theta))$, where $x^*(\theta)$ is the solution of the Tykhonov regularisation problem and $\tilde{c}(\cdot, \cdot)(\theta)$ is the "Hessian" at the mode
- The Laplace approximation

$$\tilde{\pi}(\theta \mid y) \propto \frac{\pi(y \mid x^*(\theta), \theta)\pi(x^*(\theta) \mid \theta)\pi(\theta)}{\pi_G(x^*(\theta) \mid y, \theta)}$$

has relative error of $n^{-3/2}$.

The integrated nested Laplace approximation (INLA) I

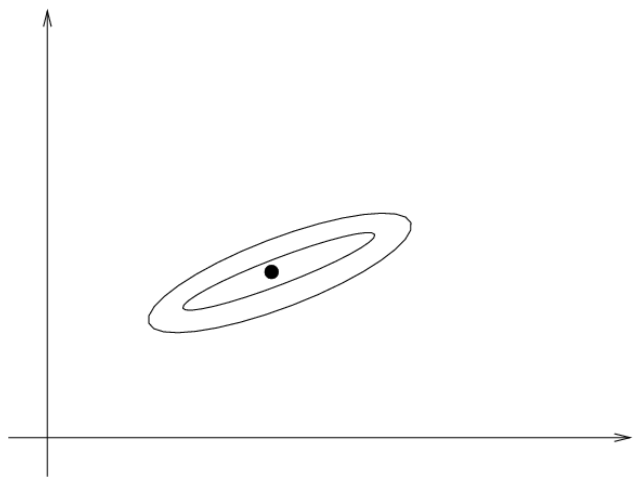
Step I Explore $\widetilde{\pi}(\boldsymbol{\theta}|\boldsymbol{y})$

- Locate the mode
- Use the Hessian to construct new variables
- Grid-search
- Can be case-specific

The integrated nested Laplace approximation (INLA) I

Step I Explore $\widetilde{\pi}(\boldsymbol{\theta}|\boldsymbol{y})$

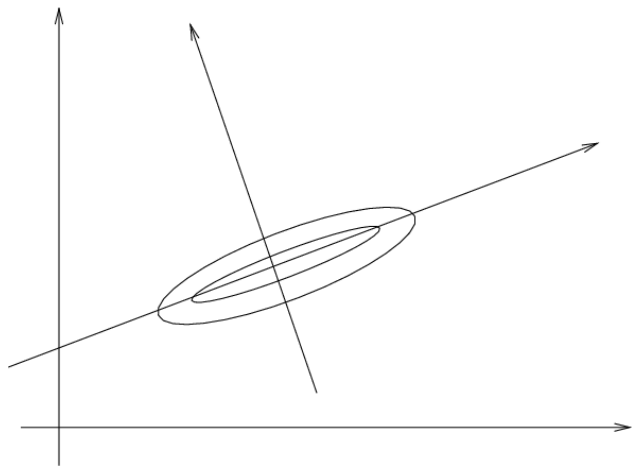
- Locate the mode
- Use the Hessian to construct new variables
- Grid-search
- Can be case-specific



The integrated nested Laplace approximation (INLA) I

Step I Explore $\widetilde{\pi}(\boldsymbol{\theta}|\boldsymbol{y})$

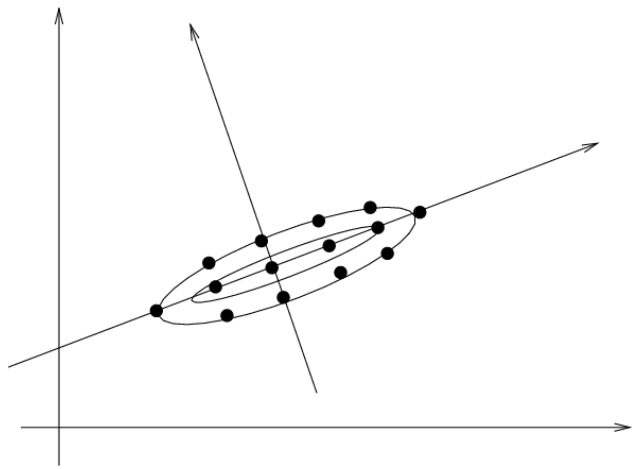
- Locate the mode
- Use the Hessian to construct new variables
- Grid-search
- Can be case-specific



The integrated nested Laplace approximation (INLA) I

Step I Explore $\widetilde{\pi}(\boldsymbol{\theta}|\boldsymbol{y})$

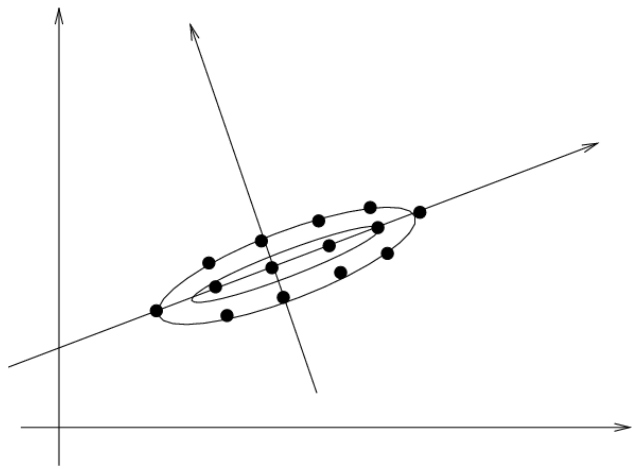
- Locate the mode
- Use the Hessian to construct new variables
- Grid-search
- Can be case-specific



The integrated nested Laplace approximation (INLA) I

Step I Explore $\widetilde{\pi}(\boldsymbol{\theta}|\boldsymbol{y})$

- Locate the mode
- Use the Hessian to construct new variables
- Grid-search
- Can be case-specific



The integrated nested Laplace approximation (INLA) II

Step II For each $\boldsymbol{\theta}_j$

- For each i, evaluate the Laplace approximation for selected values of x_i
- Build a Skew-Normal or log-spline corrected Gaussian

$$\mathcal{N}(x_i;\ \mu_i, \sigma_i^2) \times \exp(\text{spline})$$

to represent the conditional marginal density.

The integrated nested Laplace approximation (INLA) III

Step III Sum out $\boldsymbol{\theta}_j$

- For each i, sum out $\boldsymbol{\theta}$

$$\widetilde{\pi}(x_i \mid \boldsymbol{y}) \propto \sum_j \widetilde{\pi}(x_i \mid \boldsymbol{y}, \boldsymbol{\theta}_j) \times \widetilde{\pi}(\boldsymbol{\theta}_j \mid \boldsymbol{y})$$

- Build a log-spline corrected Gaussian

$$\mathcal{N}(x_i;\ \mu_i, \sigma_i^2) \times \exp(\text{spline})$$

to represent $\widetilde{\pi}(x_i \mid \boldsymbol{y})$.

Computing posterior marginals for θ_j (I)

Main idea

- Use the integration-points and build an interpolant
- Use numerical integration on that interpolant

Computing posterior marginals for θ_j (II)

Practical approach (high accuracy)

- Rerun using a fine integration grid
- Possibly with no rotation
- Just sum up at grid points, then interpolate

Outline

So where is the problem?

Outside of the standard "forward problem" challenges, we have one big numerical linear algebra problem.

$$\pi(\theta \mid y) \propto \left(\frac{|Q_x(\theta) + A^T Q_y A|}{|Q_x(\theta)|} \right)^{N/2} \times (\text{stuff})$$

- Q_x is the size of the latent structure. $(10^5 - 10^11)$
  - There is some structure, but it's unpleasant (fourth order PDEs, $(L_1(\partial_t) + L_2(\partial_s))^2$ etc)
  - There are also dense rows
  - Graph laplacians
  - Wavelet-y finger matrices
  - etc
- Q_y is the size of the data $(10^1 - 10^9)$
- Both of those determinants are infinite, but their ratio is finite.

So where is the problem?

Outside of the standard "forward problem" challenges, we have one big numerical linear algebra problem.

$$\pi(\theta \mid y) \propto \left(\frac{|Q_x(\theta) + A^T Q_y A|}{|Q_x(\theta)|} \right)^{N/2} \times (\text{stuff})$$

- Q_x is the size of the latent structure. $(10^5 - 10^11)$
  - There is some structure, but it's unpleasant (fourth order PDEs, $(L_1(\partial_t) + L_2(\partial_s))^2$ etc)
  - There are also dense rows
  - Graph laplacians
  - Wavelet-y finger matrices
  - etc
- Q_y is the size of the data $(10^1 - 10^9)$
- Both of those determinants are infinite, but their ratio is finite.

So where is the problem?

Outside of the standard "forward problem" challenges, we have one big numerical linear algebra problem.

$$\pi(\theta \mid y) \propto \left(\frac{|Q_x(\theta) + A^T Q_y A|}{|Q_x(\theta)|} \right)^{N/2} \times (\text{stuff})$$

- Q_x is the size of the latent structure. $(10^5 - 10^11)$
  - There is some structure, but it's unpleasant (fourth order PDEs, $(L_1(\partial_t) + L_2(\partial_s))^2$ etc)
  - There are also dense rows
  - Graph laplacians
  - Wavelet-y finger matrices
  - etc
- Q_y is the size of the data $(10^1 - 10^9)$
- Both of those determinants are infinite, but their ratio is finite.

So where is the problem?

Outside of the standard "forward problem" challenges, we have one big numerical linear algebra problem.

$$\pi(\theta \mid y) \propto \left(\frac{|Q_x(\theta) + A^T Q_y A|}{|Q_x(\theta)|} \right)^{N/2} \times (\text{stuff})$$

- Q_x is the size of the latent structure. $(10^5 - 10^11)$
  - There is some structure, but it's unpleasant (fourth order PDEs, $(L_1(\partial_t) + L_2(\partial_s))^2$ etc)
  - There are also dense rows
  - Graph laplacians
  - Wavelet-y finger matrices
  - etc
- Q_y is the size of the data $(10^1 - 10^9)$
- Both of those determinants are infinite, but their ratio is finite.

So where is the problem?

Outside of the standard "forward problem" challenges, we have one big numerical linear algebra problem.

$$\pi(\theta \mid y) \propto \left(\frac{|Q_x(\theta) + A^T Q_y A|}{|Q_x(\theta)|} \right)^{N/2} \times (\text{stuff})$$

- Q_x is the size of the latent structure. $(10^5 - 10^11)$
  - There is some structure, but it's unpleasant (fourth order PDEs, $(L_1(\partial_t) + L_2(\partial_s))^2$ etc)
  - There are also dense rows
  - Graph laplacians
  - Wavelet-y finger matrices
  - etc
- Q_y is the size of the data $(10^1 - 10^9)$
- Both of those determinants are infinite, but their ratio is finite.

So where is the problem?

Outside of the standard "forward problem" challenges, we have one big numerical linear algebra problem.

$$\pi(\theta \mid y) \propto \left(\frac{|Q_x(\theta) + A^T Q_y A|}{|Q_x(\theta)|} \right)^{N/2} \times (\text{stuff})$$

- Q_x is the size of the latent structure. $(10^5 - 10^11)$
  - There is some structure, but it's unpleasant (fourth order PDEs, $(L_1(\partial_t) + L_2(\partial_s))^2$ etc)
  - There are also dense rows
  - Graph laplacians
  - Wavelet-y finger matrices
  - etc
- Q_y is the size of the data $(10^1 - 10^9)$
- Both of those determinants are infinite, but their ratio is finite.

So where is the problem?

Outside of the standard "forward problem" challenges, we have one big numerical linear algebra problem.

$$\pi(\theta \mid y) \propto \left(\frac{|Q_x(\theta) + A^T Q_y A|}{|Q_x(\theta)|} \right)^{N/2} \times (\text{stuff})$$

- Q_x is the size of the latent structure. $(10^5 - 10^11)$
  - There is some structure, but it's unpleasant (fourth order PDEs, $(L_1(\partial_t) + L_2(\partial_s))^2$ etc)
  - There are also dense rows
  - Graph laplacians
  - Wavelet-y finger matrices
  - etc
- Q_y is the size of the data $(10^1 - 10^9)$
- Both of those determinants are infinite, but their ratio is finite.

So where is the problem?

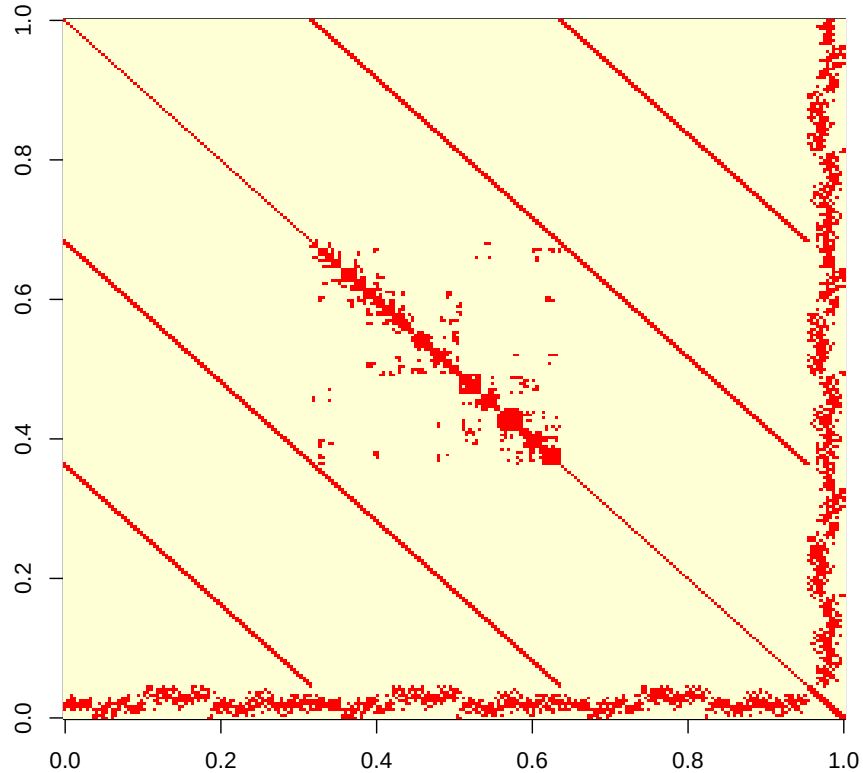
Outside of the standard "forward problem" challenges, we have one big numerical linear algebra problem.

$$\pi(\theta \mid y) \propto \left(\frac{|Q_x(\theta) + A^T Q_y A|}{|Q_x(\theta)|} \right)^{N/2} \times (\text{stuff})$$

- Q_x is the size of the latent structure. $(10^5 - 10^11)$
  - There is some structure, but it's unpleasant (fourth order PDEs, $(L_1(\partial_t) + L_2(\partial_s))^2$ etc)
  - There are also dense rows
  - Graph laplacians
  - Wavelet-y finger matrices
  - etc
- Q_y is the size of the data $(10^1 - 10^9)$
- Both of those determinants are infinite, but their ratio is finite.

Knowing me, knowing you

What does the precision matrix (usually) look like?



NB: It's good to consider the whole (jointly) Gaussian part: fixed + random effects + noise.

Linear algebra problems (ranked)

1. Solve large, SPD linear systems $(Q_x(\theta) + A^T Q_y A)u = b$
2. Compute determinants of large SPD matrices
3. Sample from large multivariate Gaussians $x \sim N(0, Q^{-1})$

The first task is "standard", the second two are HARD.

Linear algebra problems (ranked)

1. Solve large, SPD linear systems $(Q_x(\theta) + A^T Q_y A)u = b$
2. Compute determinants of large SPD matrices
3. Sample from large multivariate Gaussians $x \sim N(0, Q^{-1})$

The first task is "standard", the second two are HARD.

Linear algebra problems (ranked)

1. Solve large, SPD linear systems $(Q_x(\theta) + A^T Q_y A)u = b$
2. Compute determinants of large SPD matrices
3. Sample from large multivariate Gaussians $x \sim N(0, Q^{-1})$

The first task is "standard", the second two are HARD.

The village green preservation society

Direct methods

All methods for computing with Gaussians require a factorisation of the covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{R}\boldsymbol{R}^T$ or the precision matrix $\boldsymbol{Q} = \boldsymbol{\Sigma}^{-1} = \boldsymbol{L}\boldsymbol{L}^{\boldsymbol{T}}$. This is always† done with a Cholesky factorisation.

- $\log(\det(Q)) = 2\sum_i L_{ii}$
- We can do direct factorisations of quite large matrices
- This is stable.
- This is the only reliable way to solve these problems
- Can we do better?

A remarkable result

If the Cholesky decomposition is unavailable, a better way is to use the identity

$$\log(\det(A)) = \operatorname{tr}(\log(A)) = \sum_{i=1}^{n} e_i^T \log(A) e_i.$$

Is there a cheap way to approximate $\operatorname{tr}(\log(A))$*?*

A Stochastic Estimator of the Trace

Theorem (Hutchinson '90)

Let $B \in \mathbb{R}^{n \times n}$ be a symmetric matrix with non-zero trace. Let Z be the discrete random variable which takes the values $-1, 1$ each with probability $1/2$ and let z be a vector of n independent samples from Z. Then $z^T Bz$ is an unbiased estimator of $\mathrm{tr}(B)$ and Z is the unique random variable amongst zero mean random variables for which $z^T Bz$ is a minimum variance, unbiased estimator of $\mathrm{tr}(B)$.

Therefore

$$\log(\det(A)) = \mathbb{E}\left(z^T \log(A) z\right).$$

With probability $(1 - \delta)$, $m > \mathcal{O}(\epsilon^{-2} \log(1/\delta))$ probing vectors is enough to reduce the error to ϵ.

Nobody does it better?

The advantage of the MC scheme is that it is *unbiased* and, should you so desire, you can account for the extra randomness in an MCMC scheme to keep it asymptotically exact.

But it is slow!

As with all other things, it turns out that if you chose "better" than random vectors, you can get a method that is practically much better.

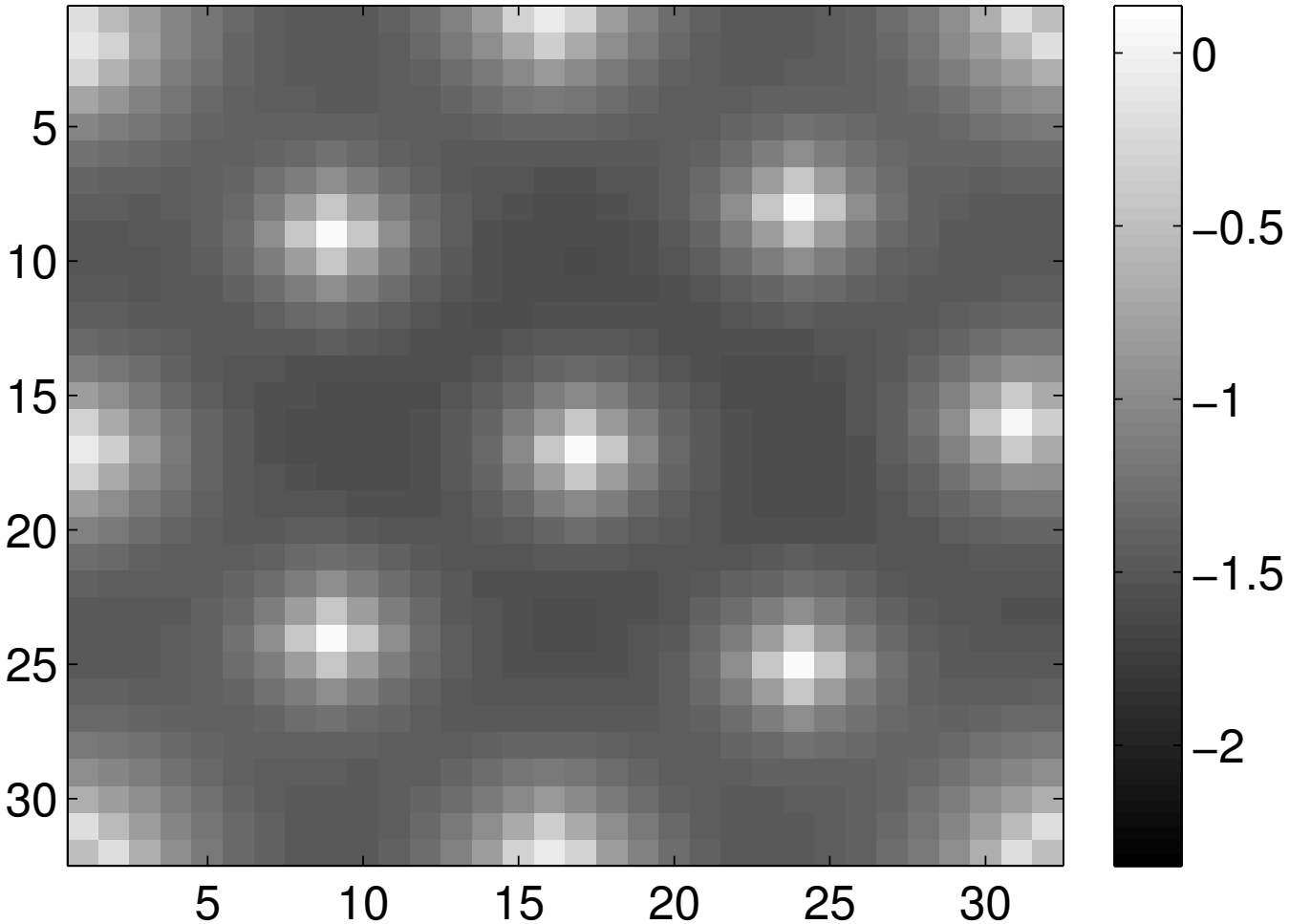
Putting it together

Here is the procedure that works best:

1. Pick a value p and produce a graph colouring of $\boldsymbol{Q}^p$.
2. For each colour c , construct a vector $\boldsymbol{z}_c$ that is randomly ± 1 (w.p. $1/2$) at the vertices of that colour and zero everywhere else
3. Use these vectors in Hutchinson's estimator of $\log(\det(\boldsymbol{Q}))$

Sometimes it's worth doing a change of basis (wavelet transform).

A probing vector



What I know

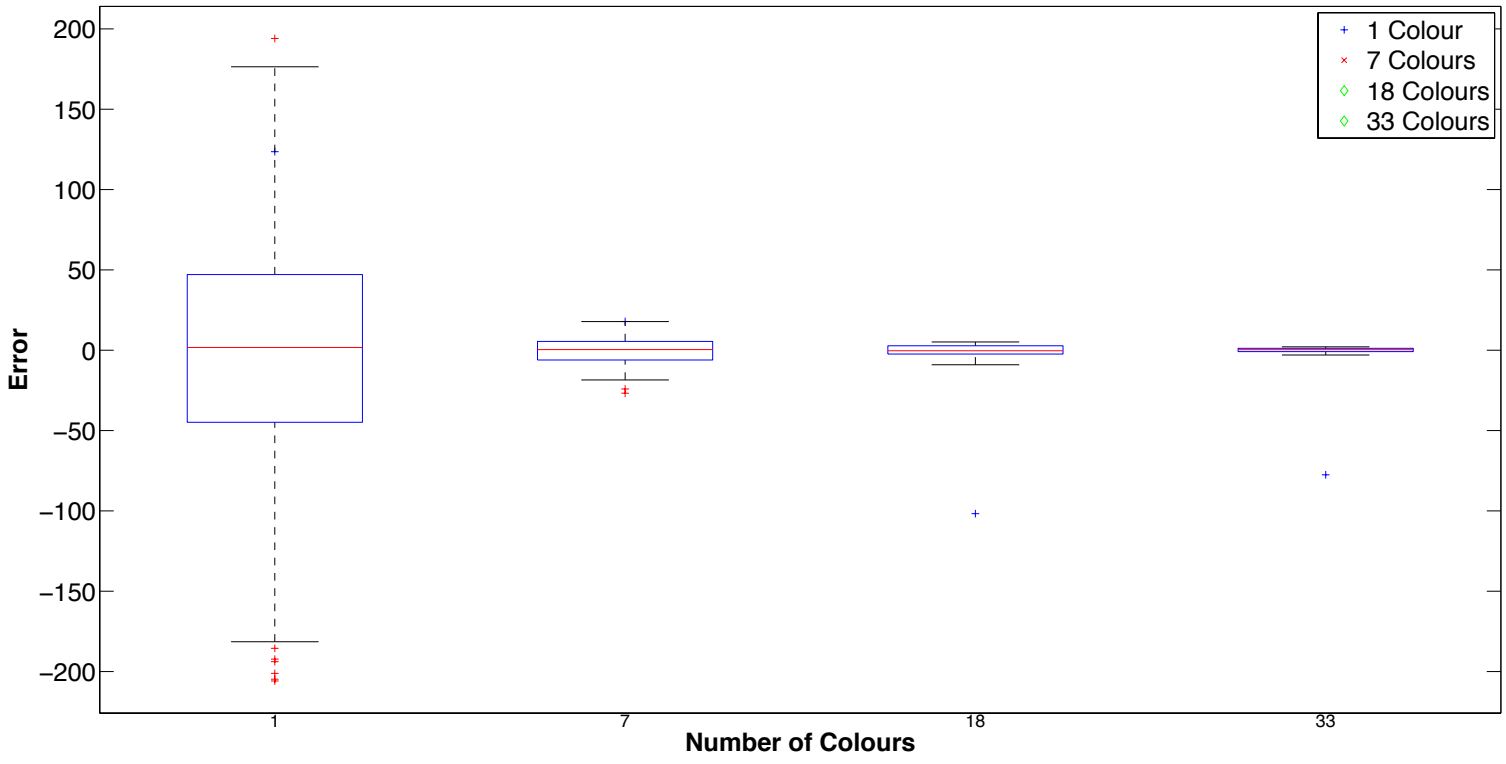
- The elements of $\log(Q)$ decay exponentially away from the non-zero entries of $\boldsymbol{Q}$
- For each colour c,

$$z_c^T \log(Q) z_c = \sum_{i \in c} [\log(\boldsymbol{Q}]_{ii} + 2 \sum_{i,j \in c} (\pm 1) \log(\boldsymbol{Q})_{ij}$$

  and the first term will dominate asymptotically.
- The "accidental" off diagonals cancel and there are fewer of them than in the basic sampler and they are smaller
- High p means more colours, but fewer vertices with each colour. If $p = n$ then you recover the trace formula.
- We can show

$$\mathrm{Var}(\mathcal{V}^T \log(\boldsymbol{Q}) \mathcal{V} \leq C^2 \sum_{i \neq j \in \mathfrak{c} \times \mathfrak{c}} (2R)^{-2d(i,j)}.$$

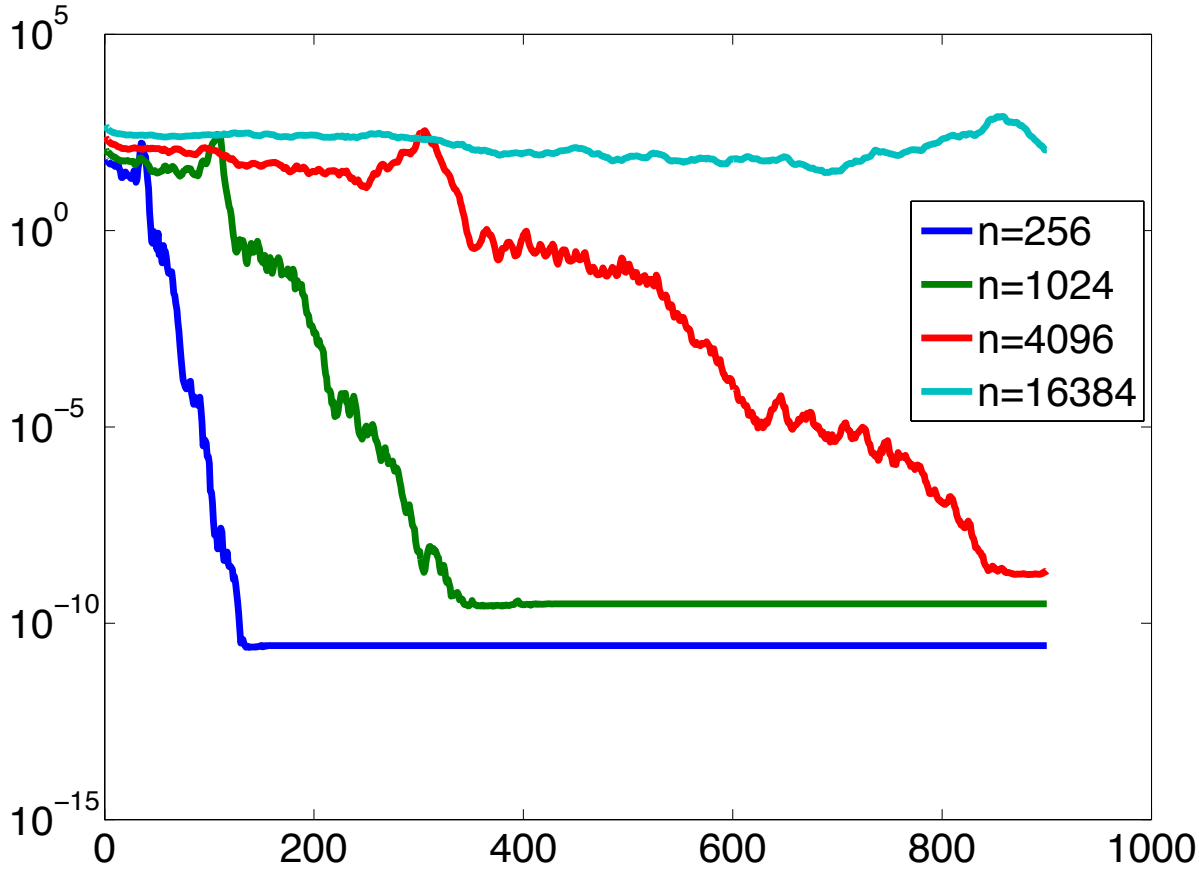
REDUCTION!


1 Colour
7 Colours
18 Colours
33 Colours
200
150
100
50
0
−50
−100
−150
−200
Error
1
7
18
33
Number of Colours

But let's talk about the logarithms

- We can compute $z^T \log(Q) z$ using a Krylov method
- Actually, we want $z^T \log(I + Q_x^{-1} A^T Q_y A) z$
- This is a Stieltjes function of a self adjoint matrix, so the convergence exactly tracks the convergence of FOM for solving $Q_x^{-1} A^T Q_y A u = b$ with the Q_x inner product.
- This also holds in floating point arithmetic and can be used a a termination criterion.

No easy way down



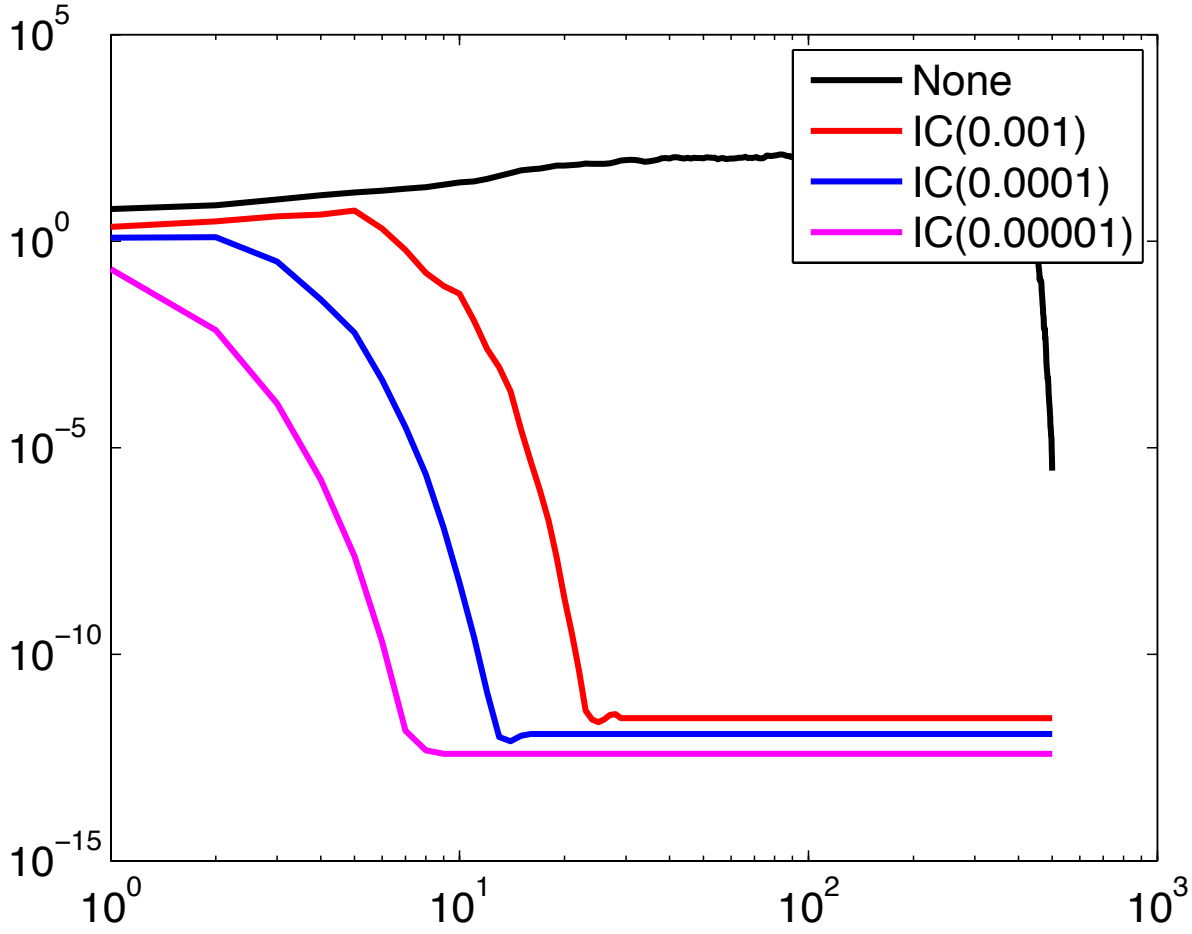
Precondtioning?

If $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^T$ is a preconditioner, then

$$\log(\det(\boldsymbol{Q})) = 2\log(\det(\boldsymbol{L})) + \log(\det(\boldsymbol{L}^{-1}\boldsymbol{Q}\boldsymbol{L}^{-T})).$$

- Typically, the first term is easy to compute, while the second is much better conditioned!
- We need efficient preconditioners that we can compute the determinant of...
- Incomplete Cholesky? Factored Sparse Approximate Inverses?
- Circulant preconditioners?

Speed Lab


None
IC(0.001)
IC(0.0001)
IC(0.00001)
10^{5}
10^{0}
10^{-5}
10^{-10}
10^{-15}
10^{0}
10^{1}
10^{2}
10^{3}

Outline

But none of this works

- While all of this works great on a single problem, when embedded in an optimiser or an MCMC algorithm, these methods do not work
- They are too slow
- They are too unstable
- The spectral properties of Q_x depend on the parameter θ, and the algorithms are not robust to it

Questions

- Can we do better with the preconditioning?
  - Sometimes we can compute Cholesky decompositions of blocks, but not the whole matrix
  - How can we use this? (nb: not a saddle point system!)
- Is the scope for a multilevel decomposition?
  - For a lot of inverse problems, we only need to compute a series of Fredholm determinants
  - There should be enough structure here to make MLMC work
  - But in general, there won't be enough structure to make multilevel methods work