Deterministic Bayesian inference methods for the Naomi model

HIV Inference Lab Group Meeting

Adam Howes

Imperial College London

April 2023

Bayesian modelling and inference

- As a statistical modeller, the bulk of our job is in constructing a generative model for data y using parameters ϑ
- This is the joint distribution $p(y, \vartheta) = p(y \mid \vartheta)p(\vartheta)$
- Then, we want to compute (ideally without having to think much about it) the posterior $p(\vartheta \mid y)$ which is **just**[1]

$$p(\vartheta \mid y) = \frac{p(y, \vartheta)}{p(y)} = \frac{p(y \mid \vartheta)p(\vartheta)}{p(y)}$$

- The central problem of Bayesian inference is doing the following integral

$$p(y) = \int p(y, \vartheta) \mathrm{d}\vartheta$$

[1] I've bolded this with sarcasm in mind: it's a difficult problem

Numerical integration

- If you want to integrate something deterministically, you could use numerical integration, otherwise called quadrature
- Select nodes $\vartheta \in \mathcal{Q} \subset \Theta$ and weights $\omega : \Theta \to \mathbb{R}$ then compute the sum

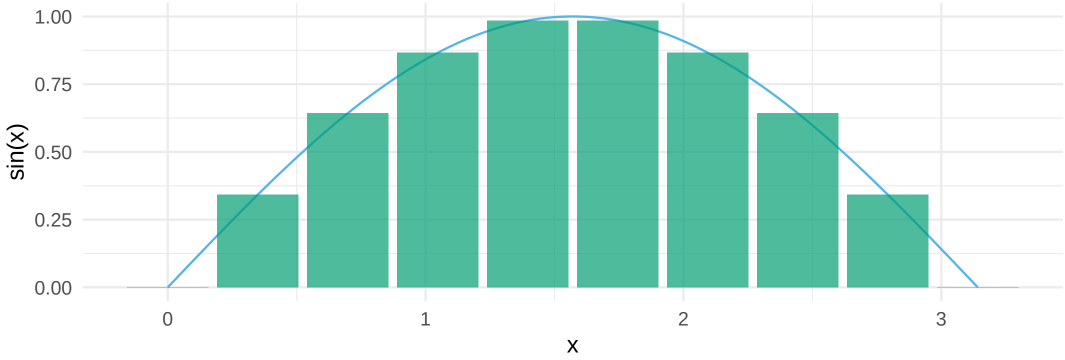
$$\tilde{p}(y) = \sum_{\vartheta \in \mathcal{Q}} p(y, \vartheta)\omega(\vartheta)$$

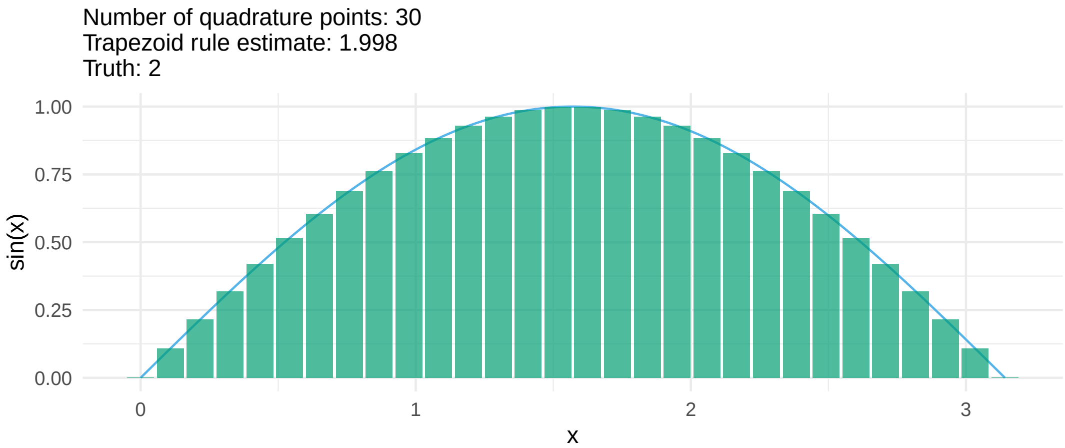
Naive quadrature example

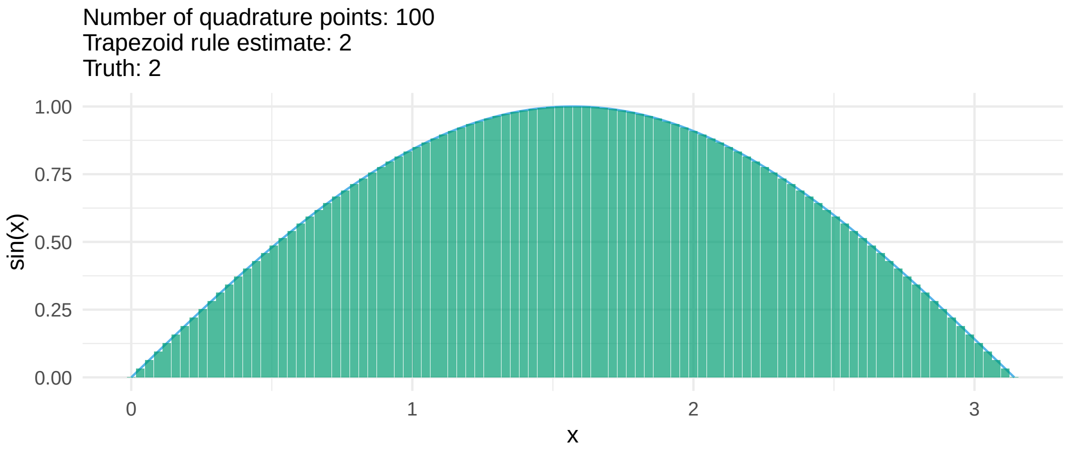
- Remember how integration is taught? (Rienmann sums)
- Try computing $\int_0^\pi \sin(x)\mathrm{d}x = 2$ using trapezoid rule

```
trapezoid_rule <- function(x, spacing) {
  w <- rep(spacing, length(x))
  w[1] <- w[1] / 2
  w[length(x)] <- w[length(x)] / 2
  sum(w * x)
}
```

Number of quadrature points: 10
Trapezoid rule estimate: 1.98
Truth: 2




Number of quadrature points: 30
Trapezoid rule estimate: 1.998
Truth: 2
sin(x)
1.00
0.75
0.50
0.25
0.00
0
1
2
3
x


Number of quadrature points: 100
Trapezoid rule estimate: 2
Truth: 2
sin(x)
1.00
0.75
0.50
0.25
0.00
0
1
2
3
x

Monte Carlo as an example of numerical integration

- Suppose we can sample $\vartheta_i \sim p(y, \vartheta)$ for $i = 1, \ldots, N$
- If we set $\omega(\vartheta_i) = 1/N$ for all i then we get a **Monte Carlo** (MC) estimate

$$\tilde{p}(y) = \frac{1}{N} \sum_i p(y, \vartheta_i)$$

- For complicated models[2] it's not possible to sample directly from $p(y, \vartheta)$, but we can usually sample from a Markov chain which if you squint a bit is good enough (MCMC)

[2] Or not even that complicated

Monte Carlo is fundamentally unsound

- "Monte Carlo ignores information" according to O'Hagan (1987)
- Suppose $N = 3$ and we sample $\vartheta_1, \vartheta_2, \vartheta_3$ with $\vartheta_2 = \vartheta_3$ then our MC estimate is
$$\tilde{p}(y) = \frac{1}{3}\left(p(y, \vartheta_1) + p(y, \vartheta_2) + p(y, \vartheta_3)\right)$$
- This is despite the fact that nothing new about the function has been learned by adding $\{\vartheta_3, p(y, \vartheta_3)\}$

Application to HIV survey sampling

- This is a digression but...
- Say we're running a household survey, and sample the same individual twice
- We didn't learn anything new about HIV by surveying them again!
- This doesn't just bite for nodes or individuals which are exactly the same: an analogous argument can be made if they are close together and we expect their function evaluations to be similar

$\implies$ Bayesian quadrature, Bayesian survey design
For some half-baked thoughts, see `athowes.github.io/fourth-gen/paper.pdf`

Latent variables and hyperparameters

- Quadrature doesn't work very well when $\dim(\vartheta)$ gets even moderately sized
  - If you're using k points per dimension it's $k^{\dim(\vartheta)}$
- Previously all of the parameters were under the symbol ϑ – what if we split them up as being $\vartheta = (x, \theta)$
- The key part about this is that $\dim(x)$ is big and $\dim(\theta)$ is small

Names for x	Names for θ
Latent variables, random effects, latent field	Hyperparameters, fixed effects

Spatio-temporal statistics

- There is nothing inherently special about spatio-temporal statistics, but we do often end up in similar situations because of the structures of the problems we tackle
- We have observations indexed by space $s \in \mathcal{S}$ and time $t \in \mathcal{T}$
- Usually we associate parameters to spatio-temporal locations as well as observations
- This ends up with us having something like $\{x_{s,t}\}$

What's important about this?

1. There might be **a lot** of spatio-temporal locations, so $\dim(x)$ might be pretty big! If you have 100 districts and 10 years, that's already $100 \times 10 = 1000$ parameters
2. Perhaps we're willing to make quite strong assumptions about how things vary over space-time[3]

- Not IID anymore!

[3] Are there any slides about spatial statistics that don't describe Tobler's first law of geography?

Latent Gaussian models

- A latent Gaussian model (LGM) (Rue, Martino, and Chopin 2009) looks along these lines:

$$
\begin{aligned}
&\text{(Observations)} && y \sim p(y \mid x, \theta), \\
&\text{(Latent field)} && x \sim \mathcal{N}(x \mid \mu(\theta), Q(\theta)^{-1}), \\
&\text{(Hyperparameters)} && \theta \sim p(\theta).
\end{aligned}
$$

- Many models are LGMs, especially in spatio-temporal statistics

Laplace approximation

- Remember that we wanted to compute

$$p(y) = \int p(y, \vartheta)\mathrm{d}\vartheta$$

- One trick for doing this is to pretend $p(\vartheta \mid y)$ is Gaussian

  - Mode $\hat{\vartheta} = \arg\max_{\vartheta} \log p(y, \vartheta)$
  - Hessian $H(\hat{\vartheta}) = -\partial_{\vartheta}^2 \log p(y, \vartheta)|_{\vartheta=\hat{\vartheta}}$
  - Gaussian approximation $\implies \tilde{p}_{\mathrm{G}}(\vartheta \mid y) = \mathcal{N}(\vartheta \mid \hat{\vartheta}, H(\hat{\vartheta})^{-1})$

Example of computing the Laplace approximation

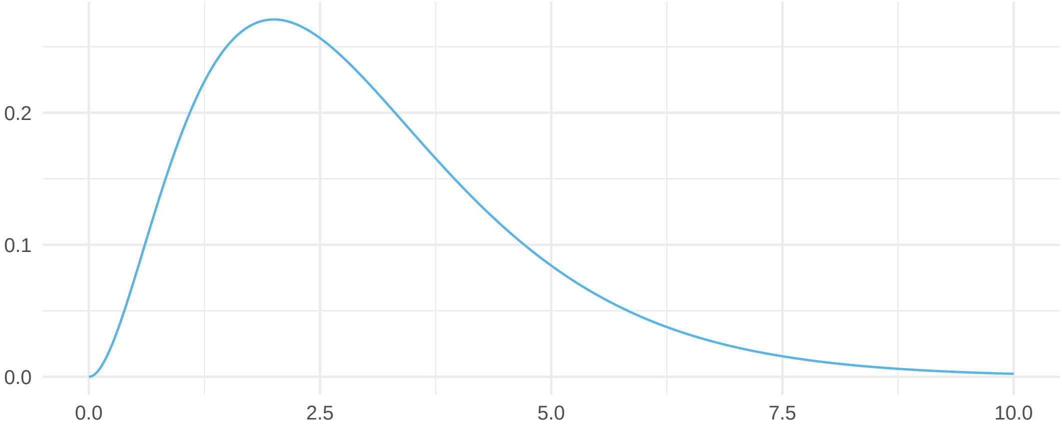
- Consider the following model for $i = 1, \ldots, n$ with fixed a and b

$$y_i \sim \text{Poisson}(\lambda), \quad \lambda \sim \text{Gamma}(a, b).$$

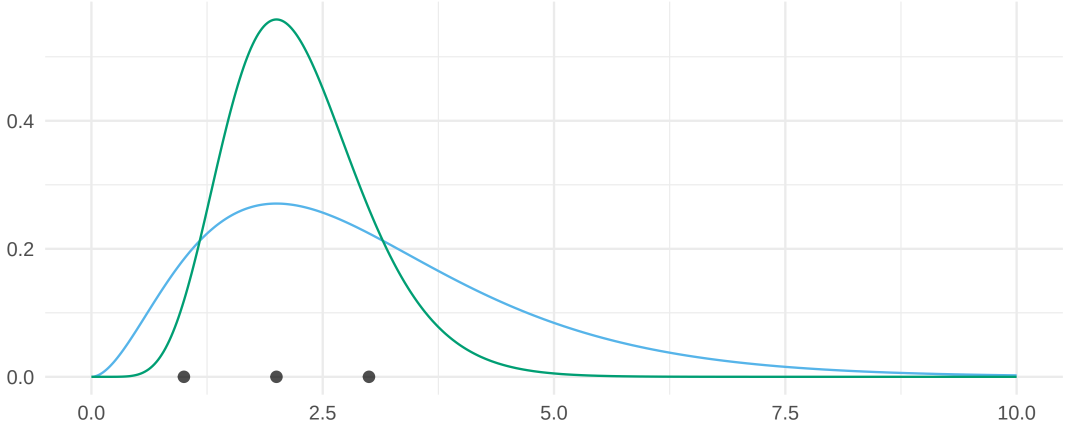
- It's conjugate so we directly know that

$$\lambda \mid y \sim \text{Gamma}(a + \sum_i y_i, b + n)$$

Prior



Posterior



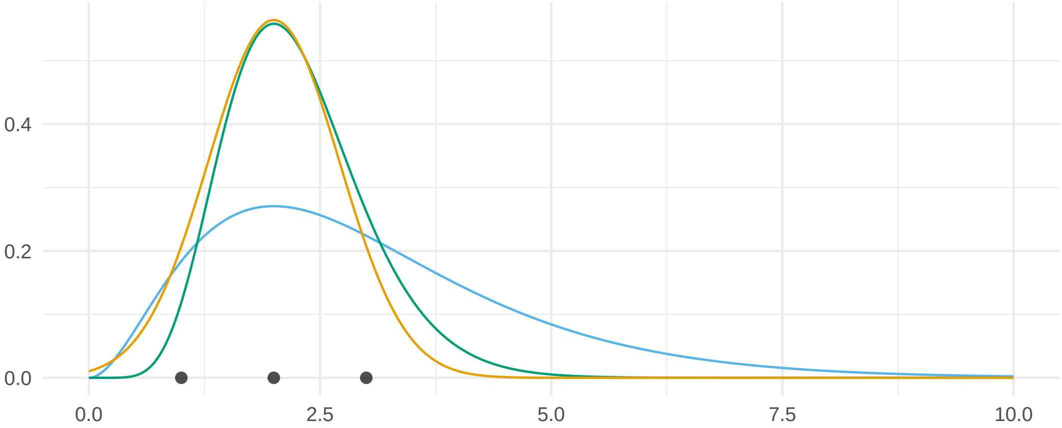
Laplace approximation

```r
fn <- function(x) dgamma(x, a + sum(y), b + length(y), log = TRUE)

# Here we are using numerical derivatives rather than automatic
ff <- list(
  fn = fn,
  gr = function(x) numDeriv::grad(fn, x),
  he = function(x) numDeriv::hessian(fn, x)
)

opt_bfgs <- aghq::optimize_theta(
  ff, 1, control = aghq::default_control(method = "BFGS")
)
```

Laplace approximation

```
laplace <- posterior +
  stat_function(
    data = data.frame(x = c(0, 10)),
    aes(x),
    fun = dnorm,
    n = 500,
    args = list(mean = opt_bfgs$mode, sd = sqrt(1 / opt_bfgs$hessian)),
    col = cbpalette[3]
  )
```

Laplace approximation



Laplace approximation

- Now
$$p(y) = \frac{p(\vartheta, y)}{p(\vartheta \mid y)} \approx \frac{p(\vartheta, y)}{p_{\mathrm{G}}(\vartheta \mid y)}$$
and we can evaluate RHS where we would like, so let's pick the point at which the Gaussian is most accurate, which is $\hat{\vartheta}$
$$p_{\mathrm{LA}}(y) = \frac{p(\vartheta, y)}{p_{\mathrm{G}}(\vartheta \mid y)}\Big|_{\vartheta=\hat{\vartheta}} = (2\pi)^{\dim(\vartheta)/2} \det(H(\hat{\vartheta}))^{-1/2} p(\hat{\vartheta}, y)$$

Marginal Laplace approximation

- Hey wait a second, is it reasonable to just assume $p(\vartheta \mid y)$ is Gaussian?
- No, not in general. But...

1. We just described a class of models (LGMs) where some subset of the parameters (x) have a Gaussian prior $\implies$ it's a lot more reasonable to think that they would have a marginal posterior which is close to Gaussian
2. We just talked about how big x is in comparison to θ! $\implies$ most of the work in our integral can be done using a **marginal Laplace** approximation to get rid of x

Generalist versus specialist methods

Dichotomy in statistical inference methods:

1. **Generalist** works in all situations
2. **Specialist** "exploits" properties of the problem at hand

We are taking approach 2!

Marginal Laplace approximation

- What does this look like? Instead of assuming $p(\vartheta \mid y) = p(x, \theta \mid y)$ is Gaussian we assume $p(x \mid \theta, y)$ is

$$\tilde{p}_{\mathrm{G}}(x \mid \theta, y) = \mathcal{N}(x \mid \hat{x}, H(\hat{x}))^{-1}$$

where $\hat{x} = \hat{x}(\theta)$
- Now the marginal Laplace approximation is

$$p_{\mathrm{LA}}(\theta, y) = \frac{p(x, \theta, y)}{\tilde{p}_{\mathrm{G}}(x \mid \theta, y)}|_{x=\hat{x}} = (2\pi)^{\dim(x)/2} \det(H(\hat{x}))^{-1/2} p(\hat{x}, \theta, y)$$

Integrated nested Laplace approximation

- Now we can compute $p_{\mathrm{LA}}(\theta, y)$ but what we really want is still $p(y)$
- But hopefully[4] the dimension of θ is small enough that we can now tackle this with quadrature
- So pick some nodes $\mathcal{Q}$ and a weighting function ω and away we go

$$p(y) \approx \sum_{\theta \in \mathcal{Q}} p_{\mathrm{LA}}(\theta, y)\omega(\theta)$$

- This is the famous integrated nested Laplace approximation (INLA)

[4] Really: hopefully

Taking stock

1. Bayesian inference is integration
2. Spatial statistics has parameters (x, θ)
3. Integrate x cheaply using a Gaussian assumption
4. Try a bit harder with θ performing quadrature

Now let's apply this to a difficult problem in HIV inference!

The Naomi model

- Naomi is a spatial evidence synthesis model
- Used by countries to produce HIV estimates in a yearly process supported by UNAIDS
- Inference for Naomi is currently conducted using `TMB` by optimising $p_{\text{LA}}(\theta, y)$, and has to be pretty quick to allow for interactive review and development of estimates

Figure 1: A supermodel

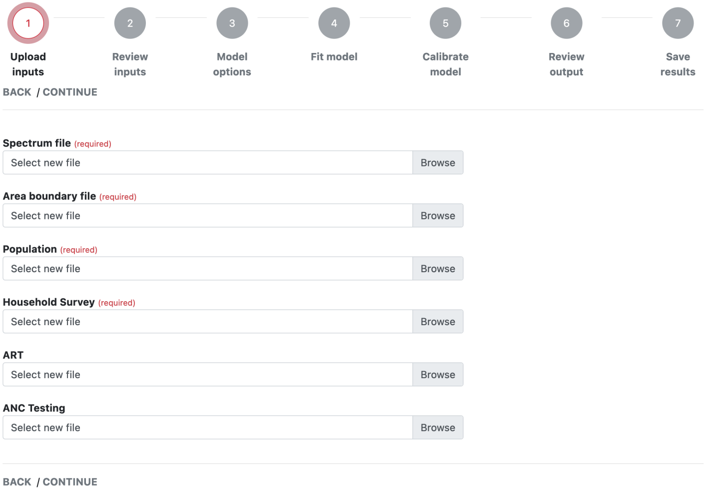


Figure 2: Example of the user interface from `https://naomi.unaids.org/`

Template Model Builder refresher

- TMB (Kristensen et al. 2015) is an R package which implements the Laplace approximation for latent variable models
- To use TMB, you write your objective function $-\log p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}) p(\mathbf{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})$ in TMB's C++ syntax
- For example, for the model $\mathbf{y} \sim \mathcal{N}(\mu, 1)$ with $p(\mu) \propto 1$ then the TMB user template looks as follows

```
#include <TMB.hpp>

template <class Type>
Type objective_function<Type>::operator()() {
  // Define data e.g.
  DATA_VECTOR(y);
  // Define parameters e.g.
  PARAMETER(mu);
  // Calculate negative log-likelihood e.g.
  nll = Type(0.0);
  nll -= dnorm(y, mu, 1, true).sum()
  return(nll);
}
```

Why do we use TMB

- It runs quickly, and is flexible enough to write the model
- Another answer: we don't have any better options
  - Markov chain Monte Carlo (MCMC) is accurate (eventually) but takes too long
  - Though Naomi is a spatial model with a large Gaussian latent field, it isn't technically a latent Gaussian model, and isn't compatible with the R-INLA implementation of INLA

Idea

- Implement an algorithm similar to INLA which

1. is compatible with Naomi
2. uses `TMB` to perform Laplace approximation

- Does this improves the quality of the inferences we get?

Not a new idea!

> *My main comment is that several aspects of the computational machineery that is presented by Rue and his colleagues* ***could benefit from the use of a numerical technique known as automatic differentiation (AD)*** *. . . By the use of AD one could obtain a system that is automatic from a user's perspective. . . the benefit would be a fast, flexible and easy-to-use system for doing Bayesian analysis in models with Gaussian latent variables*

– Hans J. Skaug (coauthor of `TMB`), RSS discussion of Rue, Martino, and Chopin (2009)

Meanwhile in Canada. . .

- Alex Stringer in Toronto $\to$ Waterloo had been thinking along similar lines, and made a lot of progress

1. Implementing an algorithm similar to INLA, using a specific quadrature rule $\mathcal{Q}$ called adaptive Gauss-Hermite quadrature (AGHQ) which he and others argue should be the default for this problem
2. Defining a class of models called extended latent Gaussian models (ELGMs) which Naomi fits into

- I will explain what both of these acronyms (AGHQ[5] and ELGM) mean!

[5] Amusingly similar to AGYW: adolescent girls and young women

Gauss-Hermite quadrature

Adaptation

- The nodes and weights we use should probably depend on the integrand
  - Especially when the integrand is also a funciton of y as in $p(y, \vartheta)$

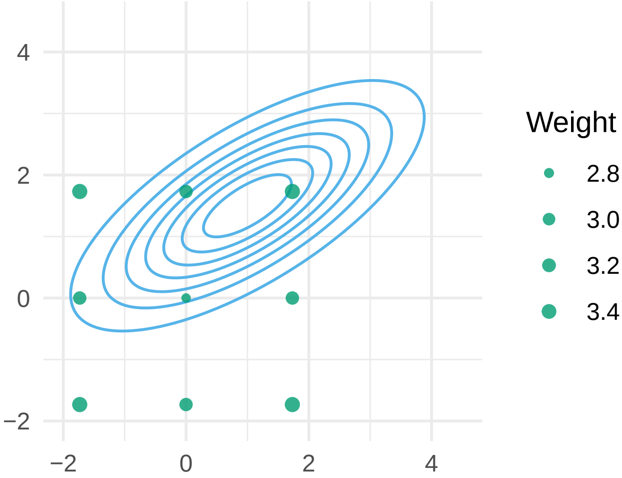


Figure 3: Unadapted points.

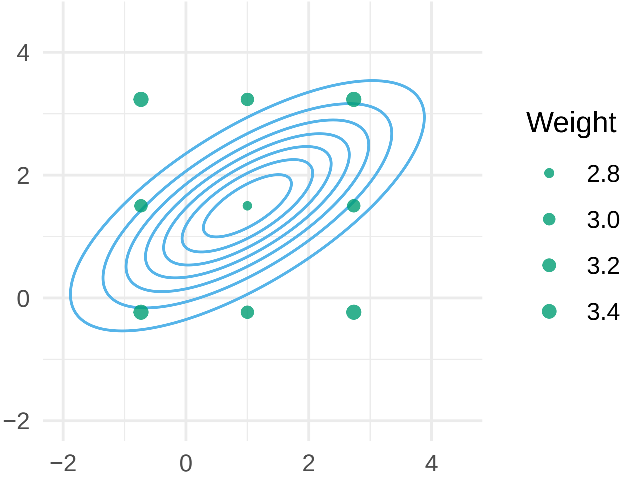


Figure 4: Add the mean.

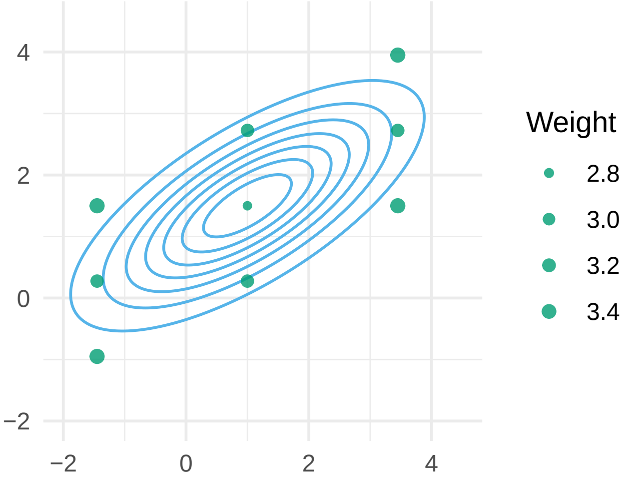


Figure 5: Rotate by the lower Cholesky.

Extended latent Gaussian models

Why is Naomi an ELGM?

1. ANC offset from household survey
2. Incidence depends on adult prevalence and coverage
3. ART attendance is a product
4. ART attendance uses a multinomial
5. Aggregation of finer processes

ANC offset from household survey

$$\mathrm{logit}(\rho_{x,a}^{\mathrm{ANC}}) = \mathrm{logit}(\rho_{x,F,a}) + \beta^{\rho^{\mathrm{ANC}}} + u_x^{\rho^{\mathrm{ANC}}} + \eta_{R_x,a}^{\rho^{\mathrm{ANC}}},$$
$$\mathrm{logit}(\alpha_{x,a}^{\mathrm{ANC}}) = \mathrm{logit}(\alpha_{x,F,a}) + \beta^{\alpha^{\mathrm{ANC}}} + u_x^{\alpha^{\mathrm{ANC}}} + \eta_{R_x,a}^{\alpha^{\mathrm{ANC}}}.$$

Incidence depends on adult prevalence and coverage

$$\log(\lambda_{x,s,a}) = \beta_0^{\lambda} + \beta_S^{\lambda, s=\text{M}} + \log(\rho_x^{\text{15-49}}) + \log(1 - \omega \cdot \alpha_x^{\text{15-49}}) + u_x^{\lambda} + \eta_{R_x,s,a}^{\lambda}.$$

ART attendance is a product

$$\dot{A}_{x',x,s,a} \sim \mathrm{Bin}(N_{x',s,a}, \pi_{x',x,s,a})$$

where $\pi_{x',x,s,a} = \rho_{x',s,a} \cdot \alpha_{x',s,a} \cdot \gamma_{x',x,s,a}$.

ART attendance multinomial

Aggregation of finer processes

The algorithm

$$
\begin{tikzcd}
p(\theta, x, y) \arrow[d, "\text{Laplace}"] & \tilde{p}_{\texttt{AQ}}(\theta \mid y) \arrow[r, "\text{Marginalise}"] & \tilde{p}_{\texttt{AQ}}(\theta_j \mid y) \arrow[d, "\text{Spline}"] \\
\tilde{p}_{\texttt{LA}}(\theta, y) \arrow[r, "\text{AGHQ}"] & \tilde{p}_{\texttt{AQ}}(y) \arrow[u, "\text{Normalise}"] & \text{Output}
\end{tikzcd}
$$

Figure 6: Inference for the hyperparameters. Shaped like a snake for no real reason.

$$
\begin{tikzcd}
p(\theta, x, y) \arrow[d, "\text{Laplace}"] & \tilde{p}_{\texttt{AQ}}(\theta \mid y) \arrow[r, "\text{Marginalise}"] & \tilde{p}_{\texttt{AQ}}(\theta_j \mid y) \arrow[d, "\text{Spline}"] \\
\tilde{p}_{\texttt{LA}}(\theta, y) \arrow[r, "\text{AGHQ}"] & \tilde{p}_{\texttt{AQ}}(y) \arrow[u, "\text{Normalise}"'] & \text{Output}
\end{tikzcd}
$$

Figure 7: Inference for the latent field.

Thanks for listening!

- Working on a paper “Fast approximate Bayesian inference for small-area estimation of HIV indicators using the Naomi model” based on this work, joint with Alex Stringer (Waterloo) and my PhD supervisors Seth Flaxman (Oxford) and Jeff Eaton (Imperial)
- Let me know if you’d be up for being an early reader!
- Code for this project is at `athowes.github.io/elgm-inf`

References I

Kristensen, Kasper, Anders Nielsen, Casper W Berg, Hans Skaug, and Brad Bell. 2015. "TMB: automatic differentiation and Laplace approximation." *arXiv Preprint arXiv:1509.00660*.

O'Hagan, Anthony. 1987. "Monte Carlo Is Fundamentally Unsound." *The Statistician*, 247–49.

Rue, Håvard, Sara Martino, and Nicolas Chopin. 2009. "Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations." *Journal of the Royal Statistical Society: Series b (Statistical Methodology)* 71 (2): 319–92.