

Integrated nested Laplace approximations for extended latent Gaussian models with application to the Naomi HIV model

Waterloo SAS Student Seminar Series

Adam Howes

Imperial College London

November 2022

Motivation

- Surveillance of the HIV epidemic in sub-Saharan Africa
- Want to estimate indicators used for monitoring and response, including:
 - Prevalence ρ : the proportion of people who are HIV positive
 - Treatment coverage α : the proportion of PLHIV on treatment
 - Incidence λ : the proportion of people newly infected
- Aim to provide estimates at a district-level to enable precision public health

This is a challenging task! Data is noisy, sparse and biased \implies
compelling case for thoughtful Bayesian modelling

A simple small-area model for prevalence

- Consider “small-areas” $i = 1, \dots, n$ (e.g. districts of a country)
- Simple random sample household-survey¹ of size m_i^{HS} where y_i^{HS} people testing positive for HIV
- Could calculate direct estimates of prevalence by $y_i^{\text{HS}}/m_i^{\text{HS}}$

Because the survey is powered at a national-level, the m_i^{HS} are small and direct estimates would be noisy \implies use a model to smooth estimates

¹In reality a complex survey design is used, often with urban rural stratification.

A simple small-area model for prevalence

- We can use a binomial logistic regression of the form:

$$y_i^{\text{HS}} \sim \text{Bin}(m_i^{\text{HS}}, \rho_i^{\text{HS}}),$$
$$\text{logit}(\rho_i^{\text{HS}}) \sim g(\vartheta^{\text{HS}}), \quad i = 1, \dots, n,$$

- We usually set up g as a Gaussian spatial smoother
- This allows for **pooling of information** between districts

Geography



Graph

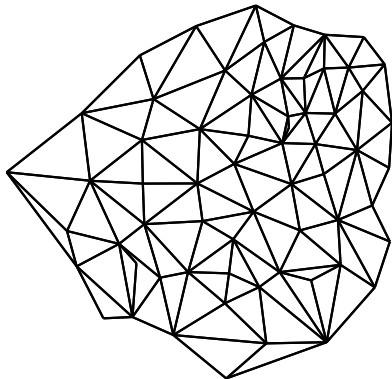


Figure 1: The Besag model, $\phi_i | \phi_{-i} \sim \mathcal{N}\left(\frac{1}{n_{\delta i}} \sum_{j:j \sim i} \phi_j, \frac{1}{n_{\delta i} \tau_\phi}\right)$.

Latent Gaussian models

- Three-stage Bayesian hierarchical model

$$\text{(Observations)} \quad \mathbf{y} \sim p(\mathbf{y} | \mathbf{x}),$$

$$\text{(Latent field)} \quad \mathbf{x} \sim p(\mathbf{x} | \boldsymbol{\theta}),$$

$$\text{(Hyperparameters)} \quad \boldsymbol{\theta} \sim p(\boldsymbol{\theta}),$$

where $\mathbf{y} = (y_1, \dots, y_n)$, $\mathbf{x} = (x_1, \dots, x_n)$, $\boldsymbol{\theta} = (\theta_1, \dots, \theta_m)$

- Interested in learning both $(\boldsymbol{\theta}, \mathbf{x})$ from data \mathbf{y}
- If the middle layer is Gaussian, then it's a **latent Gaussian model**

$$\text{(Latent field)} \quad p(\mathbf{x} | \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}(\boldsymbol{\theta}), \mathbf{Q}(\boldsymbol{\theta})^{-1}).$$

- Latent field is typically indexed by spatiotemporal location, such that $n > m$

Limitations of household surveys

- Household surveys cost millions to run so they don't happen very often
- e.g. DHS include 5k-30k households, and occurs around every 5 years

The snapshot they provide can be quite out of date, and difficult to base effective policy on \implies need to use routinely collected data to help here

Adding ANC surveillance

- Pregnant women attending antenatal care clinics are routinely tested for HIV, to avoid mother-to-child transmission. This data source is:
 1. More **real-time** than household surveys – can be collected e.g. monthly
 2. More **biased** than household surveys – attendees are not representative
- If the this bias is consistent, we can still ANC data to supplement our model

⇒ model the level using the household survey data, and the trend using the ANC data

Adding ANC surveillance

- Suppose of m_i^{ANC} ANC attendees, y_i^{ANC} are HIV positive, and model

$$\begin{aligned}y_i^{\text{ANC}} &\sim \text{Bin}(m_i^{\text{ANC}}, \rho_i^{\text{ANC}}), \\ \text{logit}(\rho_i^{\text{ANC}}) &= \text{logit}(\rho_i^{\text{HS}}) + b_i, \\ b_i &\sim \mathcal{N}(\beta_b, \sigma_b^2),\end{aligned}$$

- This is similar to using ρ_i^{ANC} as a covariate in the model for household survey prevalence, but this way takes into account sampling variation

Adding ART coverage

- Also interested in what proportion α_i of people living with HIV are receiving treatment, which may also be informative about prevalence
- If we record A_i attendees from a known population of N_i in each district, then this can be modelled by

$$A_i \sim \text{Bin}(N_i, \rho_i^{\text{HS}} \alpha_i),$$
$$\text{logit}(\alpha_i) \sim \mathcal{N}(\beta_\alpha, \sigma_\alpha^2).$$

- To be more sophisticated, you can also model the movement of people to receive treatment in districts other than the one they live in

Naomi evidence synthesis model

- Combining these three modules is the basis of the Naomi evidence synthesis model
- Used by countries to produce HIV estimates in a yearly process supported by UNAIDS
- Can't run long MCMC in this setting, so we **require fast, accurate, approximations**
- It's a complicated model, and requires something more flexible than R-INLA
- Currently using a package called Template Model Builder TMB



Figure 2: A supermodel

1

2

3

4

5

6

7

Upload inputs

Review inputs

Model options

Fit model

Calibrate model

Review output

Save results

BACK / CONTINUE

Spectrum file (required)

Select new file

Browse

Area boundary file (required)

Select new file

Browse

Population (required)

Select new file

Browse

Household Survey (required)

Select new file

Browse

ART

Select new file

Browse

ANC Testing

Select new file

Browse

BACK / CONTINUE

Figure 3: Example of the user interface from <https://naomi.unaids.org/>

Template Model Builder

- TMB (Kristensen et al. 2015) is an R package which implements the Laplace approximation for latent variable models
- I use “Laplace approximation” to mean approximating the normalising constant with Laplace’s method²
- To get started with TMB, write your $f(\mathbf{x}, \boldsymbol{\theta})$ in TMB’s C++ syntax
- As pseudo-Bayesians, we choose the log-posterior

$$f(\mathbf{x}, \boldsymbol{\theta}) = -\log p(\mathbf{y} | \mathbf{x}, \boldsymbol{\theta})p(\mathbf{x} | \boldsymbol{\theta})p(\boldsymbol{\theta})$$

- For example, for the model $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, 1)$ with $p(\boldsymbol{\mu}) \propto 1$ then the TMB user template looks as follows

²Rather than approximating the posterior with a Gaussian, which I call a Gaussian approximation.

```
#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);
}
```

Template Model Builder

- We can use TMB to obtain the Laplace approximation

$$\tilde{p}_{\text{LA}}(\boldsymbol{\theta} \mid \mathbf{y}) \propto \frac{p(\mathbf{y}, \mathbf{x}, \boldsymbol{\theta})}{\tilde{p}_{\text{G}}(\mathbf{x} \mid \boldsymbol{\theta}, \mathbf{y})} \Big|_{\mathbf{x}=\boldsymbol{\mu}(\boldsymbol{\theta})}$$

- Integrate out a Gaussian approximation $\tilde{p}_{\text{G}}(\mathbf{x} \mid \boldsymbol{\theta}, \mathbf{y})$ to the latent field
- TMB uses automatic differentiation (Griewank and Walther 2008) via CppAD to do this, as well as help with numerical optimisation routines
- TO-DO: how to describe the approximation for \mathbf{x} with TMB

Integrated Nested Laplace Approximation

- Integrated nested Laplace approximation (INLA) (Rue, Martino, and Chopin 2009; Blangiardo and Cameletti 2015) is an approach to approximate inference which builds on the Laplace approximation
- Goal is to approximate **posterior marginals** $\{\tilde{p}(x_i | \mathbf{y})\}_{i=1}^n$ and $\{\tilde{p}(\theta_j | \mathbf{y})\}_{j=1}^m$

$$p(x_i | \mathbf{y}) = \int p(x_i, \boldsymbol{\theta} | \mathbf{y}) d\boldsymbol{\theta} = \int p(x_i | \boldsymbol{\theta}, \mathbf{y}) p(\boldsymbol{\theta} | \mathbf{y}) d\boldsymbol{\theta}, \quad i = 1, \dots, n, \quad (1)$$

$$p(\theta_j | \mathbf{y}) = \int p(\boldsymbol{\theta} | \mathbf{y}) d\boldsymbol{\theta}_{-j} \quad j = 1, \dots, m. \quad (2)$$

- To do so, we require the approximations $\tilde{p}(\boldsymbol{\theta} | \mathbf{y})$ and $\tilde{p}(x_i | \boldsymbol{\theta}, \mathbf{y})$
- There are four steps as to how the method works (bare with me!)

Step 1

1) First Laplace approximate hyperparameter posterior

$$\tilde{p}_{\text{LA}}(\boldsymbol{\theta} \mid \mathbf{y}) \propto \frac{p(\mathbf{y}, \mathbf{x}, \boldsymbol{\theta})}{\tilde{p}_{\text{G}}(\mathbf{x} \mid \boldsymbol{\theta}, \mathbf{y})} \Big|_{\mathbf{x}=\boldsymbol{\mu}(\boldsymbol{\theta})} \quad (3)$$

which can be marginalised to get $\tilde{p}(\theta_j \mid \mathbf{y})$

- Notice that this is the same object we had been working with in TMB
- We use this approximation **nested** within integrals – hence the name INLA

Step 2

- 2) In both Equations (1) and (2) we want to integrate w.r.t. θ , so choose integration nodes and weights $\{\theta(\mathbf{z}), \omega(\mathbf{z})\}_{\mathbf{z} \in \mathcal{Z}}$
- For low m R-INLA uses a grid-strategy
 - For larger m this becomes too expensive and R-INLA uses a CCD design
 - We plan to use adaptive Gaussian Hermite quadrature (AGHQ), which has recently been shown to have theoretical guarantees (Bilodeau, Stringer, and Tang 2021) and is implemented in the `aghq` R package (Stringer 2021)

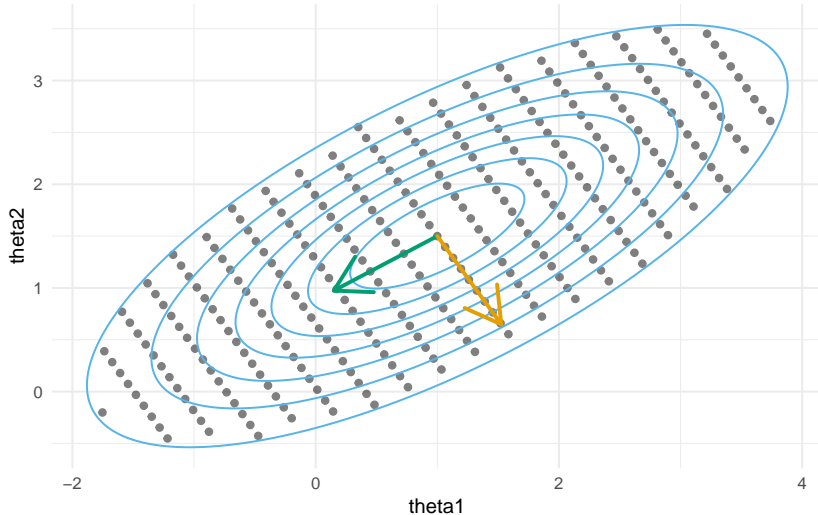


Figure 4: An illustration of the R-INLA grid method for selecting integration nodes using a toy bivariate Gaussian distribution for θ . Start at the mode and work outwards along the eigenvectors until the density drops sufficiently low.

Adaptive Gaussian Hermite Quadrature

- Gauss-Hermite quadrature is one way to pick nodes and weights, and is based on the theory of polynomial interpolation
- The adaptive part means that it uses the location (mode) and curvature (Hessian) of the target (posterior) \implies does not require manual parameter tuning!
- Works particularly well when the integrand is pretty Gaussian
- Use k quadrature nodes per dimension, e.g. if $k = 3$ then 3^m total nodes

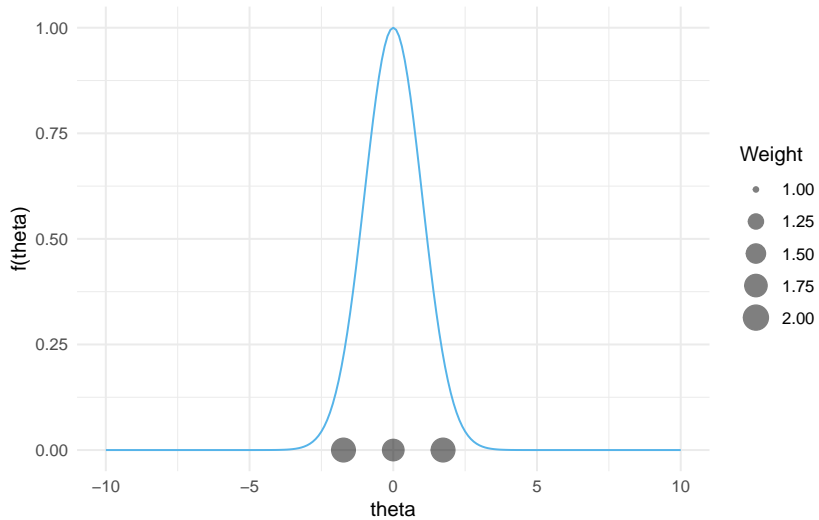


Figure 5: One dimensional example of AGHQ with $3^1 = 3$ nodes.

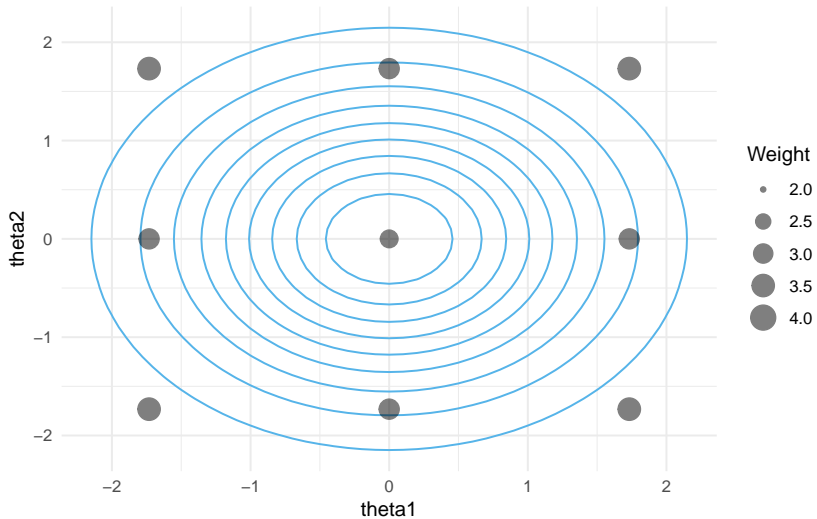


Figure 6: Two dimensional example of AGHQ with $3^2 = 9$ nodes.

Step 3

3) Choose approximation for $\tilde{p}(x_i | \boldsymbol{\theta}, \mathbf{y})$

- Simplest version (Rue and Martino 2007) is to marginalise $\tilde{p}_G(\mathbf{x} | \boldsymbol{\theta}, \mathbf{y})$

$$\tilde{p}_G(x_i | \boldsymbol{\theta}, \mathbf{y}) = \mathcal{N}(x_i | \mu_i(\boldsymbol{\theta}), 1/q_i(\boldsymbol{\theta})) \quad (4)$$

- In R-INLA, the above is referred to as `method = "gaussian"`
- This is also what is currently used in `aghq`

There are more accurate (and complicated) versions which I will talk about in a minute!

Step 4

4) Finally, use quadrature to combine

- our approximation $\tilde{p}_{\text{LA}}(\boldsymbol{\theta} \mid \mathbf{y})$ from Step 1,
- some choice of integration nodes and weights $\{\boldsymbol{\theta}(\mathbf{z}), \omega(\mathbf{z})\}$ Step 2,
- some choice of approximation $\tilde{p}(x_i \mid \boldsymbol{\theta}, \mathbf{y})$ from Step 3 to give

$$\tilde{p}(x_i \mid \mathbf{y}) = \sum_{\mathbf{z} \in \mathcal{Z}} \tilde{p}(x_i \mid \boldsymbol{\theta}(\mathbf{z}), \mathbf{y}) \times \tilde{p}_{\text{LA}}(\boldsymbol{\theta}(\mathbf{z}) \mid \mathbf{y}) \times \omega(\mathbf{z}) \quad (5)$$

Using a Laplace approximation for Step 3

- Rather than taking the marginals of $\tilde{p}_G(\mathbf{x} \mid \boldsymbol{\theta}, \mathbf{y})$ we could calculate a new Laplace approximation for each x_i

$$\tilde{p}_{\text{LA}}(x_i, \boldsymbol{\theta}, \mathbf{y}) = \frac{p(x_i, \mathbf{x}_{-i}, \boldsymbol{\theta}, \mathbf{y})}{\tilde{p}_G(\mathbf{x}_{-i} \mid x_i, \boldsymbol{\theta}, \mathbf{y})} \Big|_{\mathbf{x}_{-i} = \boldsymbol{\mu}_{-i}(x_i, \boldsymbol{\theta})}$$

where $\tilde{p}_G(\mathbf{x} \mid \boldsymbol{\theta}, \mathbf{y}) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_{-i}(x_i, \boldsymbol{\theta}), \mathbf{Q}_{-i}(x_i, \boldsymbol{\theta})^{-1})$

- Problem: n can be big, and we will need to recalculate this for each $(x_i, \boldsymbol{\theta})$
- So for example, if we use a nested AGHQ strategy with k_x and k_θ then we will need $k_x k_\theta^m$ recalculations for each i

Cheaper approximate approximations

- Rue, Martino, and Chopin (2009) found a way to do this in a cheaper and more approximate way based on assuming a **sparse** precision for \mathbf{x}
 - a.k.a. that \mathbf{x} is a Gaussian Markov random field (GMRF)
- Wood (2020) extended their approximation to work for the case when \mathbf{x} does not have a sparse precision

Epilepsy example

- Notebook for this example at athowes.github.io/elgm-inf/epil
- From Section 5.2. of Rue, Martino, and Chopin (2009) previously from BUGS manual
- Patients $i = 1, \dots, 59$ each either assigned treatment $\text{Trt}_i = 1$ or placebo $\text{Trt}_i = 0$ to help with seizures
- Visits to clinics $j = 1, \dots, 4$ times with y_{ij} the number of seizures of the i th person in the two weeks proceeding their j th visit to the clinic
- Covariates age Age_i , baseline seizure counts Base_i and an indicator for the final clinic visit V_4

Epilepsy example

The model is a Poisson GLMM:

$$y_{ij} \sim \text{Poisson}(\lambda_{ij}),$$

$$\lambda_{ij} = e^{\eta_{ij}},$$

$$\eta_{ij} = \beta_0 + \beta_{\text{Base}} \log(\text{Baseline}_j/4) + \beta_{\text{Trt}} \text{Trt}_i + \beta_{\text{Trt} \times \text{Base}} \text{Trt}_i \times \log(\text{Baseline}_j/4) \\ + \beta_{\text{Age}} \log(\text{Age}_i) + \beta_{\text{V}_4} \text{V}_{4j} + \epsilon_i + \nu_{ij}, \quad i = 1 : 59, \quad j = 1 : 4,$$

$$\beta \sim \mathcal{N}(0, 100^2), \quad \forall \beta,$$

$$\epsilon_i \sim \mathcal{N}(0, 1/\tau_\epsilon),$$

$$\nu_{ij} \sim \mathcal{N}(0, 1/\tau_\nu),$$

$$\tau_\epsilon \sim \Gamma(0.001, 0.001),$$

$$\tau_\nu \sim \Gamma(0.001, 0.001).$$

Inference

Implement the following inference procedures:

1. HMC NUTS via `tmbstan` and TMB
2. Grid with Gaussian marginals via R-INLA
3. Grid with simplified Laplace marginals via R-INLA
4. Grid with Laplace marginals via R-INLA
5. EB with Gaussian marginals via TMB
6. AGHQ with Gaussian marginals via `aghq` and TMB
7. **EB with Laplace marginals via `aghq` and TMB³**

³I'm working on AGHQ with Gaussian marginals via `aghq` and TMB. I am using the `aghq` package, just with $k = 1$ corresponding to EB

Results

Comparison approach

- You could look at the summaries like the mean and standard deviation of each of the posterior marginals but it's better to compare the whole posterior distributions
- One way to do this is via Kolmogorov-Smirnov statistics, which give the maximum difference between two empirical CDFs

Prevalence, ANC, ART example

- Notebook for this example at athowes.github.io/elgm-inf/prev-anc-art
- Simulate data from model with all three components and particular (known) parameter values

Inference

Implement the following inference procedures:

1. HMC NUTS via `tmbstan` and TMB
 2. EB with Gaussian marginals via TMB
 3. AGHQ with Gaussian marginals via `aghq` and TMB
- All of these approaches share the same C++ template, so the models are identical! This is often very difficult to ensure, so we're very fortunate here⁴

⁴i.e. thanks to Kasper and Alex for making `tmbstan` and `'aghq` respectively!

Results

Conclusions

*My main comment is that several aspects of the computational machinery 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)

Conclusions

- Hopeful that we'll give fast, accurate inferences for Naomi!
- Implementation as a part of `aghq` combining simplified INLA and AGHQ, enabled by automatic differentiation, will provide flexible use of the method
 - Will be of interest to advanced users of R-INLA who would like specify models outside a formula interface (similar to users of `brms` v.s. Stan)
 - This describes many in the HIV inference group hiv-inference.org⁵

⁵See athowes.github.io/inla-sandbox/ for some examples of understanding R-INLA internals.

References I

- Bilodeau, Blair, Alex Stringer, and Yanbo Tang. 2021. "Stochastic Convergence Rates and Applications of Adaptive Quadrature in Bayesian Inference." <https://arxiv.org/abs/2102.06801>.
- Blangiardo, Marta, and Michela Cameletti. 2015. *Spatial and spatio-temporal Bayesian models with R-INLA*. John Wiley & Sons.
- Griewank, Andreas, and Andrea Walther. 2008. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. Vol. 105. Siam.
- Kristensen, Kasper, Anders Nielsen, Casper W Berg, Hans Skaug, and Brad Bell. 2015. "TMB: automatic differentiation and Laplace approximation." *arXiv Preprint arXiv:1509.00660*.
- Rue, Håvard, and Sara Martino. 2007. "Approximate Bayesian inference for hierarchical Gaussian Markov random field models." *Journal of Statistical Planning and Inference* 137 (10): 3177–92.

References II

- 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.
- Stringer, Alex. 2021. "Implementing Approximate Bayesian Inference Using Adaptive Quadrature: The Aghq Package."
<https://arxiv.org/abs/2101.04468>.
- Wood, Simon N. 2020. "Simplified Integrated Nested Laplace Approximation." *Biometrika* 107 (1): 223–30.