Inference methods for extended latent Gaussian models

Waterloo SAS Student Seminar Series

Adam Howes

Imperial College London

October 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
  - Incidence λ: the proportion of people newly infected
  - Treatment coverage α: the proportion of PLHIV on treatment
- We would like to provide them at a local, district-level
- This is a challenging task! Data is noisy, sparse and biased

A simple small-area model for prevalence

- Consider areas $i = 1, \dots, n$
- Simple random sample household-survey taken in each area, with sample sizes m_i^{HS}
- The number of people testing positive is y_i^{HS}
- Then we can use a binomial logistic regression of the form:

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

- If g is Gaussian then this is a latent Gaussian model in the sense of Rue, Martino, and Chopin (2009)
- We usually set up g as a spatial smoother, because the sample sizes in each area are too small to get reliable direct estimates
- One problem with the above model is that household surveys are expensive to run, so they only happen rarely

Latent Gaussian models

- Three-stage Bayesian hierarchical model

$$
\begin{aligned}
&\text{(Observations)} && \mathbf{y} \sim p(\mathbf{y} \mid \mathbf{x}), \\
&\text{(Latent field)} && \mathbf{x} \sim p(\mathbf{x} \mid \boldsymbol{\theta}), \\
&\text{(Hyperparameters)} && \boldsymbol{\theta} \sim p(\boldsymbol{\theta}),
\end{aligned}
$$

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

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

- Covers most of the models commonly used in spatiotemporal statistics
- Latent field is typically indexed by spatiotemporal location, such that $n > m$

Adding ANC surveillance

- Pregnant women attending antenatal care clinics are routinely tested for HIV, to avoid mother-to-child transmission
- This data source is more real-time than household surveys, but it's also more biased, because attendees are unlikely to be as representative of the population
- But perhaps this bias is consistent, in which case we can still make use of the ANC data to improve our model!

Adding ANC surveillance

- Suppose of m_i^{ANC} women attending ANC, y_i^{ANC} are HIV positive, then we can use another binomial logistic regression:

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

- Similar to using ρ_i^{ANC} as a covariate in our model for household survey prevalence, but this way we allow it to vary rather than be fixed

Adding ART coverage

- Remember that we're also interested in what proportion of PLHIV are receiving treatment
- To estimate this, we need first to know the number of PLHIV

$$H_i = N_i \rho_i^{\mathsf{HS}}$$

- Then

$$\begin{aligned} A_i &\sim \mathrm{Bin}(N_i, \rho_i^{\mathsf{HS}} \alpha_i), \\ \mathrm{logit}(\alpha_i) &\sim \mathcal{N}(\beta_\alpha, \sigma_\alpha^2), \end{aligned}$$

Naomi evidence synthesis model

- Combining these three modules is the basis of the Naomi evidence synthesis model
- Used by countries (which provide their own data) to produce HIV estimates in a yearly process supported by UNAIDS
- Can't run long MCMC in this setting, requires fast, accurate, approximations
- It's a complicated model, and requires something more flexible than `R-INLA`
- Currently using Template Model Builder `TMB` (Kristensen et al. 2015)

Figure 1: A supermodel

Template Model Builder

- So what is TMB?
- R package which implements the Laplace approximation for latent variable models using automatic differentiation (via CppAD)
  - For more about AD see e.g. Griewank and Walther (2008)
  - Useful for getting the mode, Hessian
- Write an objective function $f(\mathbf{x}, \boldsymbol{\theta})$ in C++ ("user template")
  - We select $f(\mathbf{x}, \boldsymbol{\theta}) = -\log p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}) p(\mathbf{x} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})$

Template Model Builder

```
#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

- Performs the Laplace approximation $L_f(\boldsymbol{\theta}) \approx L_f^\star(\boldsymbol{\theta})$ and use R to optimise this with respect to $\boldsymbol{\theta}$ to give $\hat{\boldsymbol{\theta}}$ (the central point in Figure 2)
  - This is done by specifying the `random` argument to be the parameters that you want to integrate out with a Laplace approximation (the latent field)
- MAP estimate of $\mathbf{x}$ conditional on $\hat{\boldsymbol{\theta}}$
- Standard errors calculated using the δ-method (a Gaussian assumption)

Integrated Nested Laplace Approximation

- Suggested reading: the paper (Rue, Martino, and Chopin 2009) or a book e.g. Blangiardo and Cameletti (2015)
- Approximate Bayesian inference for latent Gaussian models (LGMs), which recall are three-stage models with middle layer

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

- R-INLA implementation takes advantage of sparsity properties of $\boldsymbol{Q}(\boldsymbol{\theta})$, i.e. if $\mathbf{x}$ is a Gaussian Markov random field (GMRF)

Integrated Nested Laplace Approximation

- Gives approximate posterior marginals $\{\tilde{p}(x_i \mid \mathbf{y})\}_{i=1}^{n}$ and $\{\tilde{p}(\theta_j \mid \mathbf{y})\}_{j=1}^{m}$
- To approximate posterior marginals below requires $\tilde{p}(\boldsymbol{\theta} \mid \mathbf{y})$ and $\tilde{p}(x_i \mid \boldsymbol{\theta}, \mathbf{y})$

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

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

Integrated Nested Laplace Approximation

1) First Laplace approximate hyperparameter posterior

$$\tilde{p}(\boldsymbol{\theta} \mid \mathbf{y}) \propto \left. \frac{p(\mathbf{y}, \mathbf{x}, \boldsymbol{\theta})}{\tilde{p}_G(\mathbf{x} \mid \boldsymbol{\theta}, \mathbf{y})} \right|_{\mathbf{x}=\boldsymbol{\mu}^\star(\boldsymbol{\theta})} \tag{3}$$

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

- Note: this involves integrating out a Gaussian approximation to the latent field, which we denote by $p_G(\mathbf{x} \mid \boldsymbol{\theta}, \mathbf{y})$

Integrated Nested Laplace Approximation

2) In both (1) and (2) we want to integrate w.r.t. (3), so choose integration points and weights $\{\boldsymbol{\theta}^{(k)}, \Delta^{(k)}\}$

- For low m `R-INLA` uses a grid-strategy (illustrated in the next slide)
- For larger m this becomes too expensive and `R-INLA` uses a CCD design is used
- Other approaches, like adaptive Gaussian Hermite quadrature (AGHQ) have recently been shown to have theoretical guarantees (Bilodeau, Stringer, and Tang 2021) and may be preferable

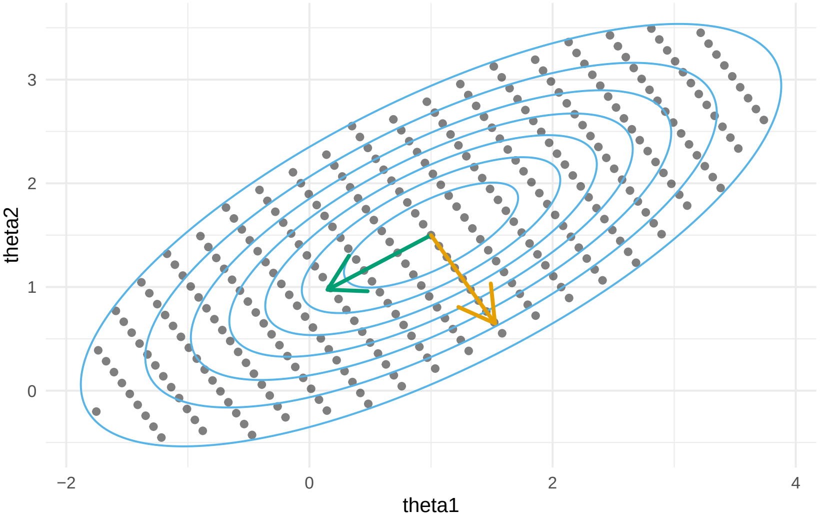

Figure 2: An illustration of the R-INLA grid method for selecting integration points using a toy bivariate Gaussian distribution for $\boldsymbol{\theta}$. Start at the mode and work outwards along the eigenvectors until the density drops sufficiently low.

Adaptive Gaussian Hermite Quadrature

- `aghq` R package and vignette (Stringer 2021)
- Gauss-Hermite quadrature is a way of picking 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)
- Use k quadrature points
  - If k is odd then they include the mode
  - If $k = 1$ then it's a Laplace approximation
  - In the vignette $k = 3$ (for each dimension, so 3^m total) is chosen quite often

Integrated Nested Laplace Approximation

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

- Simplest version (Rue and Martino 2007) is to marginalise the $p_G(\mathbf{x} \mid \boldsymbol{\theta}, \mathbf{y})$

$$\tilde{p}(x_i \mid \boldsymbol{\theta}, \mathbf{y}) = \mathcal{N}(x_i \mid \mu_i^\star(\boldsymbol{\theta}), 1/q_i^\star(\boldsymbol{\theta})) \tag{4}$$

- The above is referred to as `method = "gaussian"` in `R-INLA`
- There are two better, more complex approximations
  - Confusingly called `"simplified laplace"` and `"laplace"`

Integrated Nested Laplace Approximation

4) Finally (!) use quadrature combining

- our approximation $p(\boldsymbol{\theta} \mid \mathbf{y})$ from step 1),
- our integration points and weights $\{\boldsymbol{\theta}^{(k)}, \Delta^{(k)}\}$ from step 2),
- and our approximation $\tilde{p}(x_i \mid \boldsymbol{\theta}, \mathbf{y})$ from step 3) to give

$$\tilde{p}(x_i \mid \mathbf{y}) = \sum_{k=1}^{K} \tilde{p}(x_i \mid \boldsymbol{\theta}^{(k)}, \mathbf{y}) \times \tilde{p}(\boldsymbol{\theta}^{(k)} \mid \mathbf{y}) \times \Delta^{(k)} \tag{5}$$

Experiments

- We would like to find an accurate way to do inference for the Naomi model
- It also needs to be fast enough to use in production!

Experiments

- We wrote a simplified version of the Naomi model up in `TMB`
- This allowed us to test three inference methods **all using precisely the same model and C++ code**

1. A direct Gaussian approximation via `TMB`
2. Adaptive Gaussian Hermite quadrature via `aghq`
3. No-U-Turn Sampling (NUTS – a type of Hamiltonian Monte Carlo) via `tmbstan`

- Note: using different software it is often very difficult to ensure the model is precisely the same, so we're very fortunate here

Comparison approach

- You could look at the summaries like the mean and standard deviation of each of the posterior marginals
  - Any approximation method should be pretty good at getting the mean right
  - Gaussian approximations should be good at getting the second moment right
- It's probably 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

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.