# INLA: An Alternative to MCMC

Emily Wang

April 14, 2020

# Contents

- Background
    - Latent Gaussian models
    - Additive models
    - Gaussian Markov random fields
    - Laplace approximation
- INLA method
    - `R-INLA` package
- Simulation studies
    - Simple linear regression
    - Poisson regression
- Case application
    - Negative binomial regression

# Motivation

- Hierarchical models can easily represent very complex data.
- Useful priors make Bayesian inference very appealing.
- However, we often get complex posteriors, forcing use of MCMC.
    - Computationally intensive and slow.
    - Can't be parallelized.
    - Sometimes even Gibbs updates are slow.
- Integrated nested Laplace approximation (INLA) is a solution!
    - Approximates posterior marginals.
    - `R-INLA` package.

# Latent Gaussian Models (LGMs)

- Three-stage hierarchical model:

$$\begin{aligned} \boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta_1} &\sim \prod_i p(y_i|x_i,\boldsymbol{\theta_1}) \\ \boldsymbol{x}|\boldsymbol{\theta_2} &\sim N(\boldsymbol{\mu}(\boldsymbol{\theta_2}),\boldsymbol{Q}^{-1}(\boldsymbol{\theta_2})) \\ (\boldsymbol{\theta_1},\boldsymbol{\theta_2})^T &\sim p(\boldsymbol{\theta}) \end{aligned}$$

- Observations $\boldsymbol{y}$ are conditionally independent given $\boldsymbol{x}$ and hyperparameters $\boldsymbol{\theta_1}$.
- Latent Gaussian random field $\boldsymbol{x}$ describes all the random terms, specifies dependence structure of the model.
- Number of hyperparameters $|\boldsymbol{\theta}| < 20$.

# Additive models

- LGMs encompass many generalized additive models:

$$\boldsymbol{y} \sim \prod_i p(y_i|\mu_i)$$

$$g(\mu_i) = \eta_i = \alpha + \sum_j \beta_j z_{ij} + \sum_k f_k(w_{k,i})$$

  - Link function $g(\cdot)$
  - $z_{ij}$ covariates with linear fixed effects $\beta_j$
  - $f_k(w_{k,i})$ are "model components" on covariates $\boldsymbol{w}$.
    - Random effects, spatial effects, smoothing splines, etc.
- Then assuming Gaussian priors, the joint distribution of

$$\boldsymbol{x} = (\boldsymbol{\eta}, \alpha, \boldsymbol{\beta}, \boldsymbol{f_1}, \boldsymbol{f_2}, \ldots)$$

is Gaussian. This gives us our latent Gaussian random field $\boldsymbol{x}$.

# Gaussian Markov random fields (GMRFs)

- GMRF is a Gaussian random vector with Markov properties: for $i \neq j$ $x_i \perp x_j | x_{-ij}$.
- Recall latent Gaussian random field $\boldsymbol{x}|\boldsymbol{\theta_2} \sim N(\boldsymbol{\mu}(\boldsymbol{\theta_2}), \boldsymbol{Q}^{-1}(\boldsymbol{\theta_2}))$
- In the INLA framework, $\boldsymbol{x}$ ideally should be a sparse GMRF.
- Why? Nice properties:
  - Sparsity pattern allows fast computation.
  - $x_i \perp x_j | x_{-ij} \iff \boldsymbol{Q}_{ij} = 0$

# Laplace approximations

- Approximate target distribution with a Gaussian, matching posterior mode ($\hat{\boldsymbol{f}}$) and curvature at the mode ($\boldsymbol{A}$ is the negative Hessian of the log posterior at $\boldsymbol{f} = \hat{\boldsymbol{f}}$).

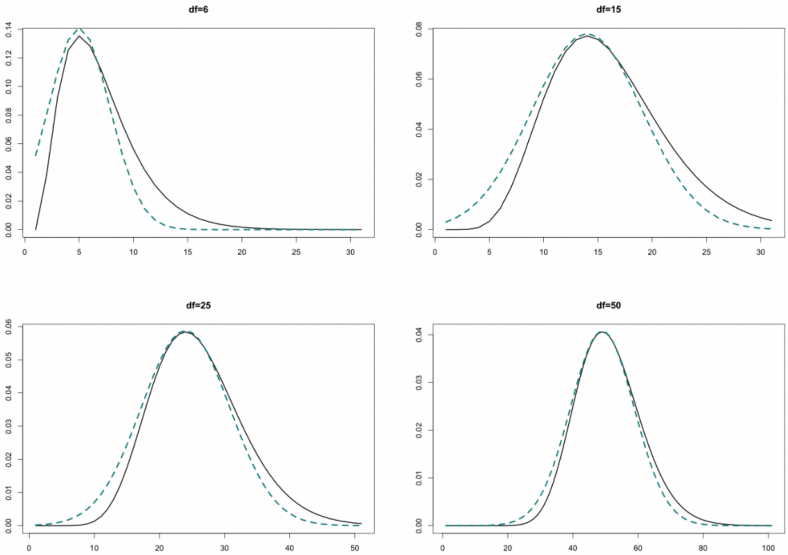
$$p(\boldsymbol{f}|y,x) \overset{approx.}{\sim} N(\hat{\boldsymbol{f}}, \boldsymbol{A}^{-1})$$

- Uses simple Taylor expansion trick on the log posterior.

$$log\, p(\boldsymbol{f}) \approx log\, p(\hat{\boldsymbol{f}}) + \frac{1}{2}(\boldsymbol{f} - \hat{\boldsymbol{f}})^T \boldsymbol{A}(\boldsymbol{f} - \hat{\boldsymbol{f}})$$
$$\boldsymbol{A} = -\nabla\nabla log\, p(\boldsymbol{f})\,|_{\boldsymbol{f}=\hat{\boldsymbol{f}}}$$

- The "more normal" the target distribution, the better the approximation.
- Use any numerical method (i.e. Newton-Raphson) to find the mode.

# Laplace approximations



# INLA

- We want:
  - $p(\boldsymbol{\theta}|\boldsymbol{y})$ – posterior marginals for hyperparameters
  - $p(x_i|\boldsymbol{y})$ – posterior marginals of latent field

# Getting marginals of hyperparameters $\boldsymbol{\theta}$

- Rewrite

$$p(\boldsymbol{\theta}|\boldsymbol{y}) \propto \frac{p(\boldsymbol{\theta})p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})}{p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})}$$

- Use Laplace approximation for the denominator

$$p(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta}) \propto exp\left(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}(\boldsymbol{\theta})\boldsymbol{x} + \sum_i log\, p(y_i|x_i,\boldsymbol{\theta})\right)$$
$$\overset{approx.}{\sim} N(\boldsymbol{\mu}(\boldsymbol{\theta}), \boldsymbol{P}^{-1}(\boldsymbol{\theta}))$$

- Approximation should be good because conditioning on observations should only shift the mean and affect only diagonal elements of $\boldsymbol{P}(\boldsymbol{\theta})$.

# Getting marginals of latent field $x$

- Rewrite

$$p(x_i|\boldsymbol{y}) = \int p(x_i|\boldsymbol{\theta}, \boldsymbol{y})p(\boldsymbol{\theta}|\boldsymbol{y})\, d\boldsymbol{\theta}$$

- Issue 1: integrating over $p(\boldsymbol{\theta}|\boldsymbol{y})$ is exponential in $|\boldsymbol{\theta}|$.
  - Solution: use composite design – integration points around a sphere (Box & Wilson 1951).
- Issue 2: approximating $p(x_i|\boldsymbol{\theta}, \boldsymbol{y})$ for each $x_i$ is expensive.
  - Solution: use 3rd order Taylor expansion around mode of Laplace approximation

$$\log p(x_i|\boldsymbol{\theta}, \boldsymbol{y}) \approx -\frac{1}{2}x_i^2 + b_i(\boldsymbol{\theta})x_i + \frac{1}{6}c_i(\boldsymbol{\theta})x_i^3$$

    Match this with skew-Normal distribution to correct for mean and skewness.
  - Called "simplified Laplace approximation".

# Integrating over $p(\boldsymbol{\theta}|\boldsymbol{y})$

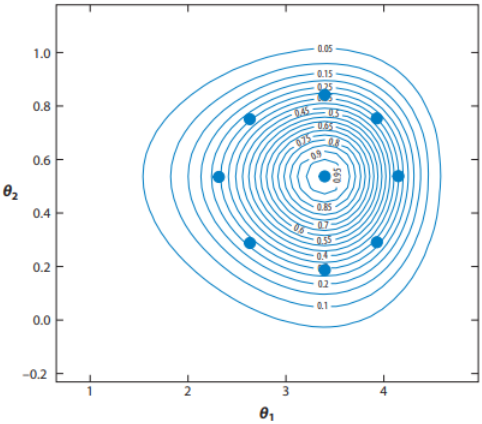



**Figure 1:** Contours of posterior marginal for $(\theta_1, \theta_2)$ and integration points (blue)

# R-INLA package

- Very dedicated community on this topic at http://www.r-inla.org/
- Model fitting procedure is similar to R `lm` function

```
library(INLA)
m1 <- inla(formula = y ~ x1 + x2 + f(...),
               data = data)
summary(m1)
```

- Specify latent field with `formula`
- Specify likelihood type with `family`
- Multi-core compute with `num.threads`
- Specify link function with `control.predictor`

# Challenges of using INLA

- Priors are set by default, any changes are difficult and must be done manually.
- Cannot deal with missing data.
- Fitting complex models besides LGMs can be done but is very difficult.
- Some likelihoods (e.g. multinomial) not supported.
- Issues arise if full conditional density for the latent field is not "near" Gaussian.

# Simulation 1 - Simple linear regression

- **Model**:

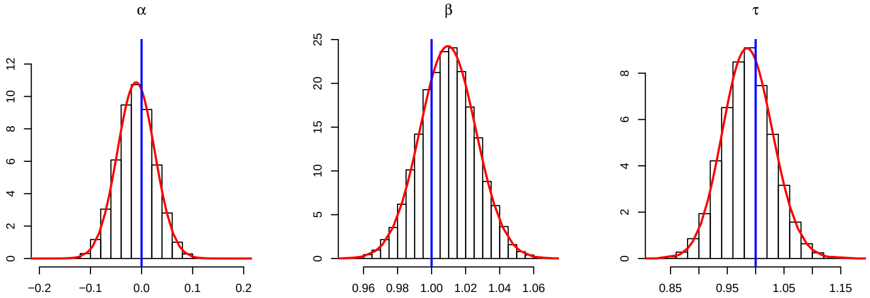
$$y_i = \alpha + \beta x_i + \epsilon_i$$
$$\epsilon_i \sim N(0, \sigma^2)$$
$$\pi(\tau, \beta, \alpha | X) \propto Gamma(a, b)$$
$$\tau = \sigma^{-2}$$

- **Gibbs updates**:

$$\beta | \tau, y \sim N\left(\hat{\beta}, \tau^{-1}(X^T X)^{-1}\right)$$
$$\hat{\beta} = (X^T X)^{-1} X^T y$$
$$\tau \sim Gamma(a + \frac{N}{2}, b + \frac{(y - X\hat{\beta})^T (y - X\hat{\beta})}{2})$$

# Simulation 1 - Simple linear regression

| | INLA | Gibbs |
|---|---|---|
| $N = 10^2$ | 1.310 s | 1.196 s |
| $N = 10^3$ | 1.580 s | 1.416 s |
| $N = 10^4$ | 7.150 s | 3.796 s |



# Simulation 2 - Poisson regression with iid random effect

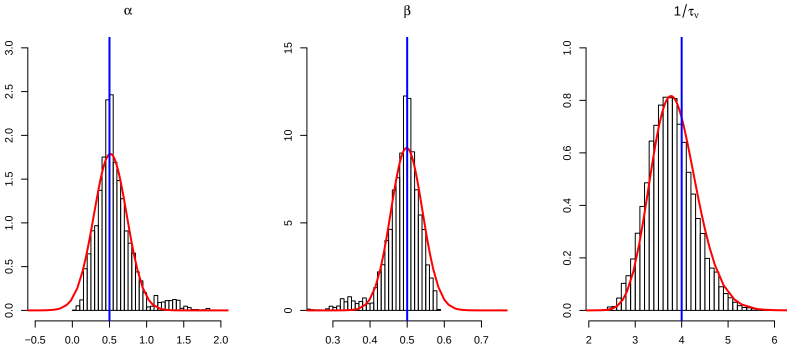
- **Model**

$$y_i|x_i \sim Poisson(\mu_i)$$
$$log(\mu_i) = \alpha + \beta x_i + \nu_i$$
$$\nu_i \sim N(0, \tau_\nu)$$

- Ordinary Poisson GLM with an added iid random effect $\nu_i$.
- MCMC using JAGS ("Just Another Gibbs Sampler").
    - Automatic MCMC sampler for hierarchical models.
    - Uses Gibbs, adaptive rejection, slice sampling, or Metropolis-Hastings, depending on situation.
    - Similar to BUGS and Stan.

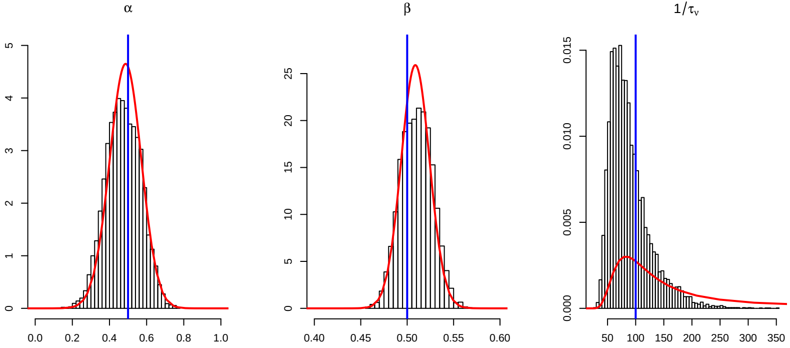
# Simulation 2 - Poisson regression with iid random effect

| | INLA | JAGS |
|---|---|---|
| $N = 20$ | 1.150 s | 0.622 s |
| $N = 200$ | 1.540 s | 6.729 s |
| $N = 2000$ | 3.190 s | 73.020 s |



# Simulation 2 - Poisson regression with iid random effect

| | INLA | JAGS |
|---|---|---|
| $N = 20$ | 1.150 s | 0.622 s |
| $N = 200$ | 1.540 s | 6.729 s |
| $N = 2000$ | 3.190 s | 73.020 s |



# Case study

- $N$ patients with epilepsy, each patient $i$ with $T_i$ time points of:
  - Recorded seizure counts $Y_{it}$
  - Clinical covariates $\boldsymbol{X_{it}}$
- Using Poisson regression to model counts requires assuming variance $=$ mean.
- Traditional negative binomial regression has no closed-form update and requires use of Metropolis algorithm.

# Case study

- Reparametrize the NB with dispersion $r$ and success probability $\psi_{it}$:

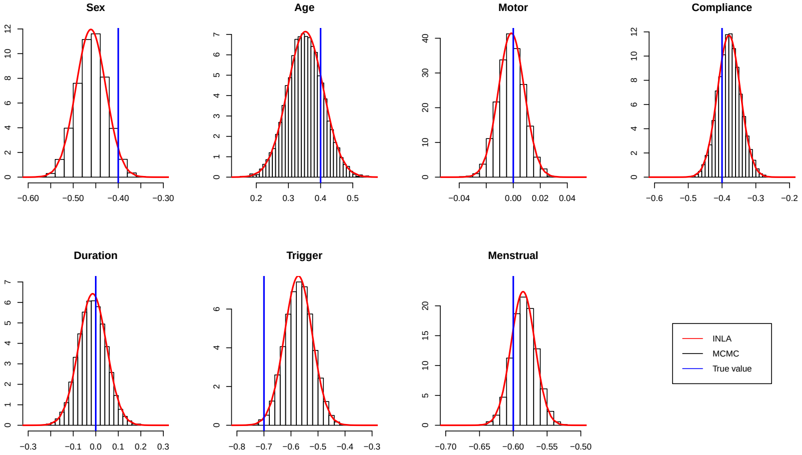
$$p(Y_{it}|\psi_{it}, r) = \frac{\Gamma(Y_{it} + r)}{\Gamma(r) Y_{it}!} (1 - \psi_{it})^r \psi_{it}^{Y_{it}}$$

$$\psi_{\text{it}} = \frac{exp(\boldsymbol{X}_{it}^T \boldsymbol{\beta})}{1 + exp(\boldsymbol{X}_{it}^T \boldsymbol{\beta})}$$

- Gibbs updates for NB regression coefficients $\boldsymbol{\beta}$ via Polya-Gamma data augmentation (Pillow and Scott, 2012):
  1. Draw $\omega_{it} \sim PG(Y_{it} + r, \boldsymbol{X}_{it}^T \boldsymbol{\beta})$.
  2. Define $\kappa_{it} = \frac{Y_{it} - r}{2\omega_{it}}$.
  3. Draw $\boldsymbol{\beta} \sim N(\boldsymbol{\mu}, \Sigma)$ where $\boldsymbol{\mu} = \Sigma \left( \Sigma_\beta^{-1} \boldsymbol{\mu}_\beta + \mathbb{X}^T \boldsymbol{\Omega} \boldsymbol{\kappa} \right)$ and $\Sigma = \left( \Sigma_\beta^{-1} + \mathbb{X}^T \boldsymbol{\Omega}_2 \mathbb{X} \right)^{-1}$.

# Case study - NB regression with known dispersion

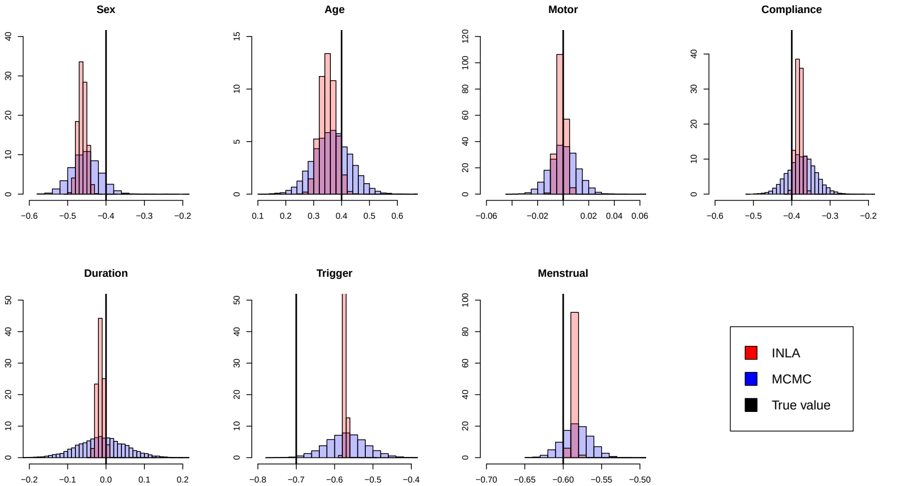
- INLA takes 1.62 s, Gibbs takes 191.7 s (for $10^4$ iterations).



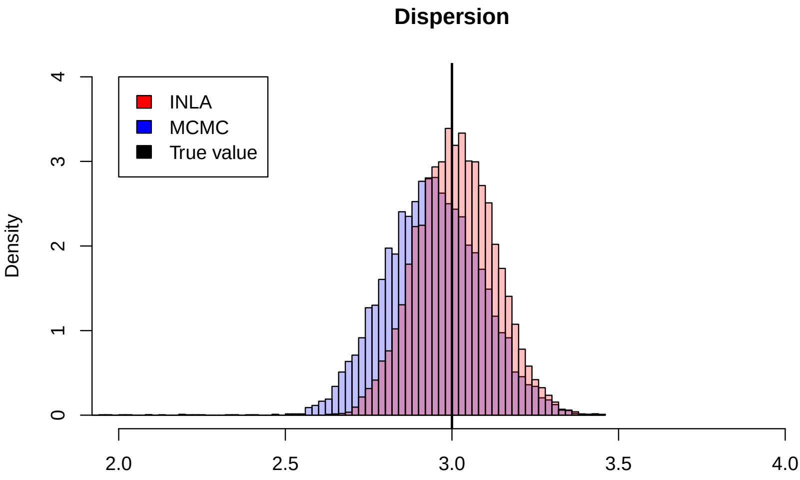
# Case study - NB regression with unknown dispersion

- Previous simulation assumes dispersion is known.
- If dispersion is unknown, it must be sampled.
- We can update dispersion using data augmentation method proposed by Zhou et al. (2012):
  1. First, draw $L_{it} \sim CRT(Y_{it}, r)$, where $CRT$ is the Chinese restaurant table distribution.
  2. Then, draw $r \sim Gamma(e + \sum_{i,t} L_{it}, f - \sum_{i,t} log(1 - \psi_{it}))$.

# Case study - NB regression with unknown dispersion



# Case study - NB regression with unknown dispersion



# References


Box, G. E. P., & Wilson, K. B. (1951). On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society. Series B (Methodological)*, *13*(1), 1–45.

Pillow, J. W., & Scott, J. (2012). Fully bayesian inference for neural models with negative-binomial spiking. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 25* (pp. 1898–1906). Curran Associates, Inc.

Polson, N. G., Scott, J. G., & Windle, J. (2013). Bayesian inference for logistic models using pólya–gamma latent variables. *Journal of the American Statistical Association*, *108*(504), 1339–1349.

Rue, H., Martino, S., & Chopin, N. (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), 319–392.

Rue, H., Riebler, A., Sørbye, S. H., Illian, J. B., Simpson, D. P., & Lindgren, F. K. (2017). Bayesian computing with inla: A review. *Annual Review of Statistics and Its Application*, *4*(1), 395–421.