Session 3.1: Introduction to INLA and `R-INLA`

Bayesian modelling for Spatial and Spatio-temporal data, Imperial College

Learning objectives

After this lecture you should be able to

- Describe what MCMC is
- Present the class of latent Gaussian models
- Present the Laplace approximation and the INLA approach
- Use the basic functions of the `R-INLA` package

The topics treated in this lecture are presented in Chapter 4 of Blangiardo and Cameletti (2015).

Outline

1. MCMC
2. MCMC to INLA
3. Latent Gaussian models
4. The INLA approach
5. `R-INLA` package

MCMC

Why Markov Chain Monte Carlo?

- For all but trivial examples it will be difficult to draw an iid Monte Carlo sample directly from the posterior distribution.
  - This happens, for example, when the dimension of the parameter vector $\boldsymbol{\theta}$ is high
  - Also to use MC methods we must have a known form for the posterior distribution

Why Markov Chain Monte Carlo?

- For all but trivial examples it will be difficult to draw an iid Monte Carlo sample directly from the posterior distribution.
  – This happens, for example, when the dimension of the parameter vector $\boldsymbol{\theta}$ is high
  – Also to use MC methods we must have a known form for the posterior distribution

- Alternatively *correlated* values (via MCMC) can be (more easily) drawn to approximate the posterior distribution of the parameter of interest
- Instead of simulating independent values from the posterior distribution we draw a sample by running a Markov chain whose stationary distribution is the posterior density $p(\theta \mid y)$

Why Markov Chain Monte Carlo?

- For all but trivial examples it will be difficult to draw an iid Monte Carlo sample directly from the posterior distribution.
  – This happens, for example, when the dimension of the parameter vector $\boldsymbol{\theta}$ is high
  – Also to use MC methods we must have a known form for the posterior distribution

- Alternatively *correlated* values (via MCMC) can be (more easily) drawn to approximate the posterior distribution of the parameter of interest
- Instead of simulating independent values from the posterior distribution we draw a sample by running a Markov chain whose stationary distribution is the posterior density $p(\theta \mid y)$

- A sequence of values $\left\{\theta^{(1)}, \ldots, \theta^{(m)}\right\}$ generated from a Markov chain that has reached its stationary distribution (i.e. has converged) can be considered as an approximation to the posterior distribution and can be used to compute all the summaries of interest.

Why Markov Chain Monte Carlo?

- For all but trivial examples it will be difficult to draw an iid Monte Carlo sample directly from the posterior distribution.
  – This happens, for example, when the dimension of the parameter vector $\boldsymbol{\theta}$ is high
  – Also to use MC methods we must have a known form for the posterior distribution

- Alternatively *correlated* values (via MCMC) can be (more easily) drawn to approximate the posterior distribution of the parameter of interest
- Instead of simulating independent values from the posterior distribution we draw a sample by running a Markov chain whose stationary distribution is the posterior density $p(\theta \mid y)$

- A sequence of values $\left\{\theta^{(1)}, \ldots, \theta^{(m)}\right\}$ generated from a Markov chain that has reached its stationary distribution (i.e. has converged) can be considered as an approximation to the posterior distribution and can be used to compute all the summaries of interest.

- MCMC methods are very general and can effectively be applied to any model
- Even if **in theory**, MCMC can provide (nearly) exact inference, given perfect convergence and MC error $\to 0$, in practice, this has to be balanced with model complexity and running time
- This is an issue particularly for problems characterised by large data or very complex structure (e.g. hierarchical models)

MCMC: Gibbs sampling

The **Gibbs sampling** (GS) is one of the most popular schemes for MCMC. Consider the case of a generic J dimensional parameter set $(\theta_1, \theta_2, \ldots, \theta_J)$:

1. Select a set of initial values $(\theta_1^{(0)}, \theta_2^{(0)}, \ldots, \theta_J^{(0)})$
2. Sample $\theta_1^{(1)}$ from the conditional distribution $p(\theta_1 \mid \theta_2^{(0)}, \theta_3^{(0)}, \ldots, \theta_J^{(0)}, y)$; Sample $\theta_2^{(1)}$ from the conditional distribution $p(\theta_2 \mid \theta_1^{(1)}, \theta_3^{(0)}, \ldots, \theta_J^{(0)}, y)$; ...; Sample $\theta_J^{(1)}$ from the conditional distribution $p(\theta_J \mid \theta_1^{(1)}, \theta_2^{(1)}, \ldots, \theta_{J-1}^{(1)}, y)$

MCMC: Gibbs sampling

The **Gibbs sampling** (GS) is one of the most popular schemes for MCMC. Consider the case of a generic J dimensional parameter set $(\theta_1, \theta_2, \ldots, \theta_J)$:

1. Select a set of initial values $(\theta_1^{(0)}, \theta_2^{(0)}, \ldots, \theta_J^{(0)})$
2. Sample $\theta_1^{(1)}$ from the conditional distribution $p(\theta_1 \mid \theta_2^{(0)}, \theta_3^{(0)}, \ldots, \theta_J^{(0)}, y)$; Sample $\theta_2^{(1)}$ from the conditional distribution $p(\theta_2 \mid \theta_1^{(1)}, \theta_3^{(0)}, \ldots, \theta_J^{(0)}, y)$; ...; Sample $\theta_J^{(1)}$ from the conditional distribution $p(\theta_J \mid \theta_1^{(1)}, \theta_2^{(1)}, \ldots, \theta_{J-1}^{(1)}, y)$
3. Repeat step 2. for S times until convergence is reached to the target distribution $p(\boldsymbol{\theta} \mid y)$
4. Use the sample from the target distribution to compute all relevant statistics: (posterior) mean, variance, credibility intervals, etc.

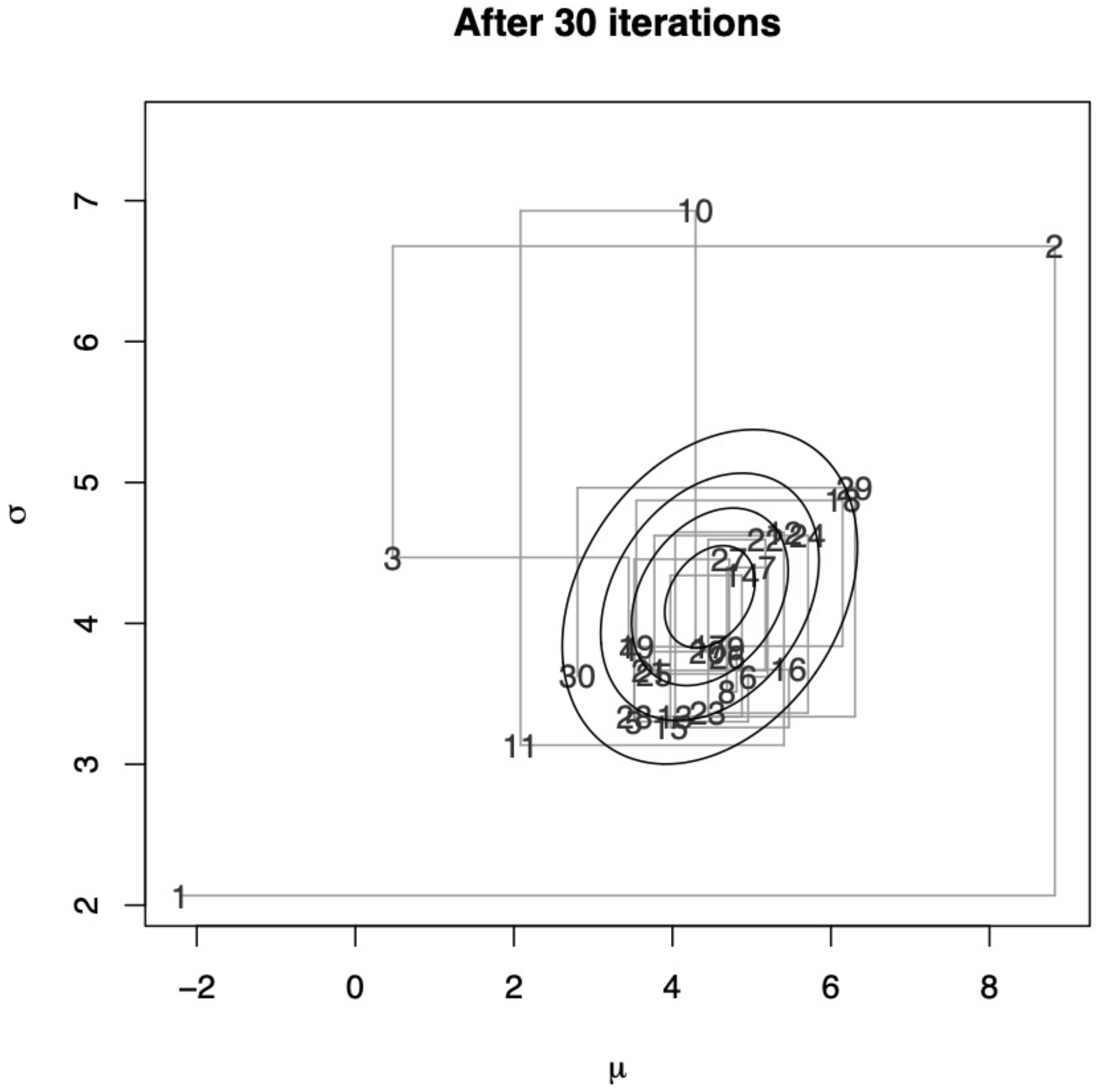
If the *full conditionals* are not readily available, they need to be estimated (eg via Metropolis-Hastings) before applying the GS

Easy references for MCMC are

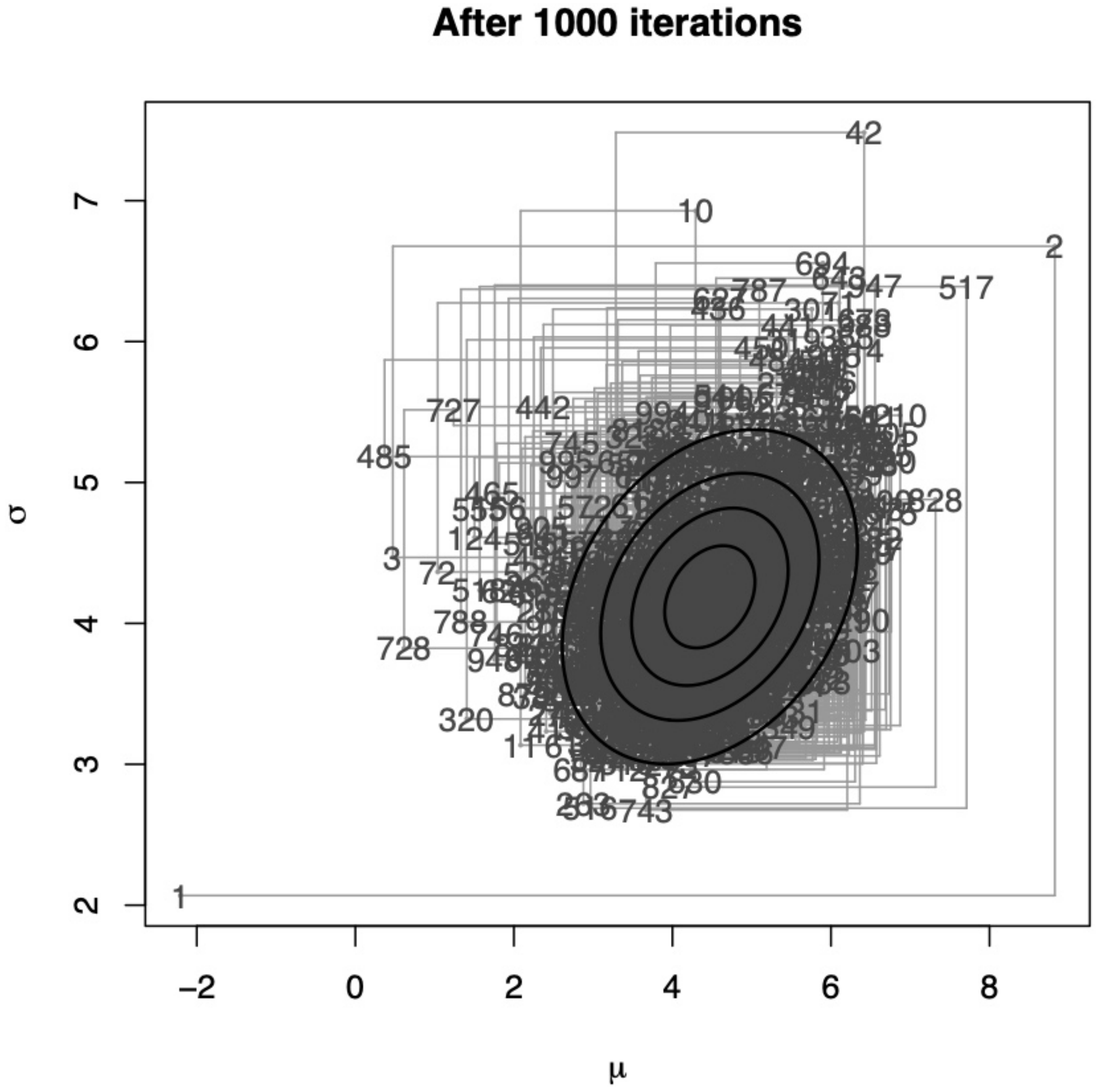
- Blangiardo and Cameletti (2015), Chapter 4
- Johnson, Ott, and Dogucu (2022), Chapters 6-7

MCMC: convergence

MCMC: convergence



MCMC: convergence



MCMC to INLA

From MCMC to INLA

- Standard MCMC sampler are generally easy-ish to program and are in fact implemented in readily available software
- MCMC methods are flexible and able to deal with virtually any type of data and model, but they involve computationally- and time- intensive simulations to obtain the posterior distribution for the parameters. For this reason the complexity of the model and the database dimension often remain fundamental issues.

From MCMC to INLA

- `Standard` MCMC sampler are generally easy-ish to program and are in fact implemented in readily available software
- MCMC methods are flexible and able to deal with virtually any type of data and model, but they involve computationally- and time- intensive simulations to obtain the posterior distribution for the parameters. For this reason the complexity of the model and the database dimension often remain fundamental issues.

- The INLA algorithm proposed by Rue, Martino, and Chopin (2009) is a *deterministic* algorithm for Bayesian inference and it represents an alternative to MCMC which is instead a simulation based algorithm.

From MCMC to INLA

- Standard MCMC sampler are generally easy-ish to program and are in fact implemented in readily available software
- MCMC methods are flexible and able to deal with virtually any type of data and model, but they involve computationally- and time- intensive simulations to obtain the posterior distribution for the parameters. For this reason the complexity of the model and the database dimension often remain fundamental issues.

- The INLA algorithm proposed by Rue, Martino, and Chopin (2009) is a *deterministic* algorithm for Bayesian inference and it represents an alternative to MCMC which is instead a simulation based algorithm.

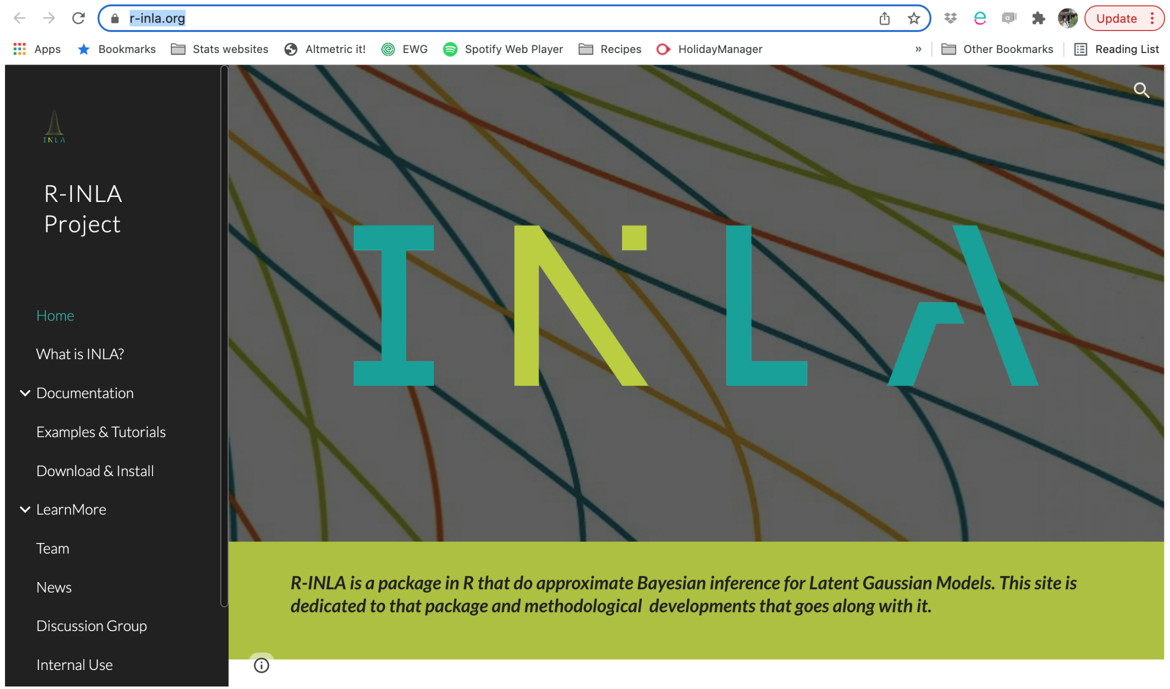
- The INLA algorithm is designed for the class of *latent Gaussian models* and compared to MCMC it provides (as) accurate results in a shorter time.

From MCMC to INLA

- `Standard` MCMC sampler are generally easy-ish to program and are in fact implemented in readily available software
- MCMC methods are flexible and able to deal with virtually any type of data and model, but they involve computationally- and time- intensive simulations to obtain the posterior distribution for the parameters. For this reason the complexity of the model and the database dimension often remain fundamental issues.
- The INLA algorithm proposed by Rue, Martino, and Chopin (2009) is a *deterministic* algorithm for Bayesian inference and it represents an alternative to MCMC which is instead a simulation based algorithm.
- The INLA algorithm is designed for the class of *latent Gaussian models* and compared to MCMC it provides (as) accurate results in a shorter time.
- INLA has become very popular among statisticians and applied researchers and in the past few years the number of papers reporting usage and extensions of the INLA method has increased considerably.

INLA website and community

- The website contains source code, examples, papers and reports discussing the theory and applications of INLA.
- There is also a discussion forum where users can post queries and requests of help.
- Almost each year there is an INLA-related scientific meeting.



INLA website

Latent Gaussian models

Latent Gaussian models (LGMs)

- The general problem of (parametric) inference is posited by assuming a probability model for the observed data $\boldsymbol{y} = (y_1, \ldots, y_n)$, as a function of some relevant parameters

$$\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\psi} \sim p(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\psi}) = \prod_{i=1}^{n} p(y_i \mid \boldsymbol{\theta}, \boldsymbol{\psi})$$

- The general problem of (parametric) inference is posited by assuming a probability model for the observed data $\boldsymbol{y} = (y_1, \ldots, y_n)$, as a function of some relevant parameters

$$\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\psi} \sim p(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\psi}) = \prod_{i=1}^{n} p(y_i \mid \boldsymbol{\theta}, \boldsymbol{\psi})$$

- Often (in fact for a surprisingly large range of models!), we can assume that the parameters are described by a **Gaussian Markov Random Field** (GMRF)

$$\boldsymbol{\theta} \mid \boldsymbol{\psi} \sim \text{Normal}(\mathbf{0}, \boldsymbol{Q}^{-\mathbf{1}}(\boldsymbol{\psi}))$$

$$\theta_i \perp\!\!\!\perp \theta_j \mid \boldsymbol{\theta}_{-i,j} \iff Q_{ij}(\boldsymbol{\psi}) = 0$$

where

- The precision matrix $\boldsymbol{Q}$ depends on some hyperparameters $\boldsymbol{\psi}$.
- The notation $-i, j$ indicates all the other elements of the parameters vector, excluding elements i and j
- The components of $\boldsymbol{\theta}$ are supposed to be *conditionally independent* with the consequence that $\boldsymbol{Q}$ is a sparse precision matrix.
- This kind of models is often referred to as **Latent Gaussian models**.

LGMs as a general framework

- In general

$$
\begin{aligned}
\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\psi} &\sim \prod_i p(y_i \mid \boldsymbol{\theta}, \boldsymbol{\psi}) \text{ (Data model)} \\
\boldsymbol{\theta} \mid \boldsymbol{\psi} &\sim p(\boldsymbol{\theta} \mid \boldsymbol{\psi}) = \text{Normal}(0, \boldsymbol{Q}^{-1}(\boldsymbol{\psi})) \text{ (Latent Gaussian Field)} \\
\boldsymbol{\psi} &\sim p(\boldsymbol{\psi}) \text{ (Hyperprior)}
\end{aligned}
$$

LGMs as a general framework

- In general

$$\begin{aligned} \boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\psi} &\sim \prod_i p(y_i \mid \boldsymbol{\theta}, \boldsymbol{\psi}) \text{ (Data model)} \\ \boldsymbol{\theta} \mid \boldsymbol{\psi} &\sim p(\boldsymbol{\theta} \mid \boldsymbol{\psi}) = \text{Normal}(0, \boldsymbol{Q}^{-1}(\boldsymbol{\psi})) \text{ (Latent Gaussian Field)} \\ \boldsymbol{\psi} &\sim p(\boldsymbol{\psi}) \text{ (Hyperprior)} \end{aligned}$$

- The dimension of $\boldsymbol{\theta}$ can be very large (eg 10 ^{2}-10 ^{5}).
- Conversely, the dimension of $\boldsymbol{\psi}$ must be relatively small (less than 20 is recommended) to avoid an exponential increase in the computational costs of the model.

LGMs as a general framework

- A very general way of specifying the problem is specifying a distribution for y_i characterized by a parameter ϕ_i (usually the mean) defined as a function of a structured additive predictor η_i, defined on a suitable scale, such that $g(\phi_i) = \eta_i$ (e.g. logistic for binomial data):

$$\eta_i = \beta_0 + \sum_{m=1}^{M} \beta_m x_{mi} + \sum_{l=1}^{L} f_l(z_{li})$$

where

- β_0 is the intercept;
- $\boldsymbol{\beta} = \{\beta_1, \ldots, \beta_M\}$ quantify the effect of the covariates $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M)$ on the response;
- $\boldsymbol{f} = \{f_1(\cdot), \ldots, f_L(\cdot)\}$ is a set of functions defined in terms of some covariates $\boldsymbol{z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_L)$

and then assume

$$\boldsymbol{\theta} = \{\beta_0, \boldsymbol{\beta}, \boldsymbol{f}\} \sim \text{Normal}(\boldsymbol{0}, \boldsymbol{Q}^{-1}(\boldsymbol{\psi})) = \text{GMRF}(\boldsymbol{\psi})$$

LGMs as a general framework

- A very general way of specifying the problem is specifying a distribution for y_i characterized by a parameter ϕ_i (usually the mean) defined as a function of a structured additive predictor η_i, defined on a suitable scale, such that $g(\phi_i) = \eta_i$ (e.g. logistic for binomial data):

$$\eta_i = \beta_0 + \sum_{m=1}^{M} \beta_m x_{mi} + \sum_{l=1}^{L} f_l(z_{li})$$

where

- β_0 is the intercept;
- $\boldsymbol{\beta} = \{\beta_1, \ldots, \beta_M\}$ quantify the effect of the covariates $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M)$ on the response;
- $\boldsymbol{f} = \{f_1(\cdot), \ldots, f_L(\cdot)\}$ is a set of functions defined in terms of some covariates $\boldsymbol{z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_L)$

and then assume

$$\boldsymbol{\theta} = \{\beta_0, \boldsymbol{\beta}, \boldsymbol{f}\} \sim \text{Normal}(\mathbf{0}, \boldsymbol{Q}^{-1}(\boldsymbol{\psi})) = \text{GMRF}(\boldsymbol{\psi})$$

- **NB**: This of course implies some form of Normally-distributed marginals for β_0, $\boldsymbol{\beta}$ and $\boldsymbol{f}$

LGMs as a general framework --- examples

Upon varying the form of the functions $f_l(\cdot)$, this formulation can accommodate a wide range of models (see Martins, Simpson, Lindgren, and Rue (2013) for a review)

- Standard regression
  - $f_l(\cdot) = \text{NULL}$

LGMs as a general framework --- examples

Upon varying the form of the functions $f_l(\cdot)$, this formulation can accommodate a wide range of models (see Martins, Simpson, Lindgren, and Rue (2013) for a review)

- Standard regression
  - $f_l(\cdot) = \mathrm{NULL}$
- Hierarchical models
  - $f_l(\cdot) \sim \mathrm{Normal}(0, \sigma_f^2)$ (Exchangeable)
    $\sigma_f^2 \mid \boldsymbol{\psi} \sim$ some common distribution

LGMs as a general framework --- examples

Upon varying the form of the functions $f_l(\cdot)$, this formulation can accommodate a wide range of models (see Martins, Simpson, Lindgren, and Rue (2013) for a review)

- Standard regression
  - $f_l(\cdot) = \text{NULL}$
- Hierarchical models
  - $f_l(\cdot) \sim \text{Normal}(0, \sigma_f^2)$ (Exchangeable)
    $\sigma_f^2 \mid \boldsymbol{\psi} \sim$ some common distribution
- Spatial and spatio-temporal models
  - Areal data: $f_1(\cdot) \sim \text{CAR}$ (Spatially structured effects)
    $f_2(\cdot) \sim \text{Normal}(0, \sigma_{f_2}^2)$ (Unstructured residual)
  - Geostatistical data: $f(\cdot) \sim \text{Gaussian field}$
  - Temporal component: $f(\cdot) \sim \text{RW}$

LGMs as a general framework --- examples

Upon varying the form of the functions $f_l(\cdot)$, this formulation can accommodate a wide range of models (see Martins, Simpson, Lindgren, and Rue (2013) for a review)

- Standard regression
  - $f_l(\cdot) = \text{NULL}$
- Hierarchical models
  - $f_l(\cdot) \sim \text{Normal}(0, \sigma_f^2)$ (Exchangeable)
    $\sigma_f^2 \mid \boldsymbol{\psi} \sim$ some common distribution
- Spatial and spatio-temporal models
  - Areal data: $f_1(\cdot) \sim \text{CAR}$ (Spatially structured effects)
    $f_2(\cdot) \sim \text{Normal}(0, \sigma_{f_2}^2)$ (Unstructured residual)
  - Geostatistical data: $f(\cdot) \sim \text{Gaussian field}$
  - Temporal component: $f(\cdot) \sim \text{RW}$
- Spline smoothing
  - $f_l(\cdot) \sim \text{AR}(\phi, \sigma_\varepsilon^2)$

LGMs as a general framework --- examples

Upon varying the form of the functions $f_l(\cdot)$, this formulation can accommodate a wide range of models (see Martins, Simpson, Lindgren, and Rue (2013) for a review)

- Standard regression
  - $f_l(\cdot) = \text{NULL}$
- Hierarchical models
  - $f_l(\cdot) \sim \text{Normal}(0, \sigma_f^2)$ (Exchangeable)

    $\sigma_f^2 \mid \boldsymbol{\psi} \sim$ some common distribution
- Spatial and spatio-temporal models
  - Areal data: $f_1(\cdot) \sim \text{CAR}$ (Spatially structured effects)

    $f_2(\cdot) \sim \text{Normal}(0, \sigma_{f_2}^2)$ (Unstructured residual)
  - Geostatistical data: $f(\cdot) \sim \text{Gaussian field}$
  - Temporal component: $f(\cdot) \sim \text{RW}$
- Spline smoothing
  - $f_l(\cdot) \sim \text{AR}(\phi, \sigma_\varepsilon^2)$
- Survival models, logGaussian Cox Processes, etc.

The INLA approach

- The first *ingredient* of the INLA approach is the definition of conditional probability, which holds for any pair of variables (x, z).

Technically, provided $p(z) > 0$

$$p(x \mid z) =: \frac{p(x, z)}{p(z)} \rightarrow p(x, z) = p(x \mid z)p(z)$$

$p(x \mid z)$ can be re-written as

$$p(z) = \frac{p(x, z)}{p(x \mid z)}$$

Integrated Nested Laplace Approximation (INLA)

- The first *ingredient* of the INLA approach is the definition of conditional probability, which holds for any pair of variables (x, z).

Technically, provided $p(z) > 0$

$$p(x \mid z) =: \frac{p(x, z)}{p(z)} \rightarrow p(x, z) = p(x \mid z)p(z)$$

$p(x \mid z)$ can be re-written as

$$p(z) = \frac{p(x, z)}{p(x \mid z)}$$

- In particular, a conditional version can be obtained further considering a third variable w as

$$p(z \mid w) = \frac{p(x, z \mid w)}{p(x \mid z, w)}$$

which is particularly relevant to the Bayesian case.

Integrated Nested Laplace Approximation (INLA)

- The second *ingredient* is **Laplace approximation**

Integrated Nested Laplace Approximation (INLA)

- The second *ingredient* is **Laplace approximation**
- Main idea: approximate $\log f(x)$ using a quadratic function by means of a Taylor's series expansion around the mode $x^* = \text{argmax}_x \log f(x)$

$$
\begin{aligned}
\log f(x) &\approx \log f(x^*) + (x - x^*) \left.\frac{\partial \log f(x)}{\partial x}\right|_{x=x^*} + \frac{(x-x^*)^2}{2} \left.\frac{\partial^2 \log f(x)}{\partial x^2}\right|_{x=x^*} \\
&= \log f(x^*) + \frac{(x-x^*)^2}{2} \left.\frac{\partial^2 \log f(x)}{\partial x^2}\right|_{x=x^*} \qquad \left(\text{since } \left.\frac{\partial \log f(x)}{\partial x}\right|_{x=x^*} = 0\right)
\end{aligned}
$$

Integrated Nested Laplace Approximation (INLA)

- The second *ingredient* is Laplace approximation
- Main idea: approximate $\log f(x)$ using a quadratic function by means of a Taylor's series expansion around the mode $x^* = \text{argmax}_x \log f(x)$

$$\log f(x) \approx \log f(x^*) + (x - x^*) \left.\frac{\partial \log f(x)}{\partial x}\right|_{x=x^*} + \frac{(x-x^*)^2}{2} \left.\frac{\partial^2 \log f(x)}{\partial x^2}\right|_{x=x^*}$$

$$= \log f(x^*) + \frac{(x-x^*)^2}{2} \left.\frac{\partial^2 \log f(x)}{\partial x^2}\right|_{x=x^*} \qquad \left(\text{since } \left.\frac{\partial \log f(x)}{\partial x}\right|_{x=x^*} = 0\right)$$

- Setting ${\sigma^2}^* = -1 / \left.\frac{\partial^2 \log f(x)}{\partial x^2}\right|_{x=x^*}$ we can re-write

$$\log f(x) \approx \log f(x^*) - \frac{1}{2{\sigma^2}^*}(x - x^*)^2$$

or equivalently

$$\int f(x)dx = \int \exp[\log f(x)]dx \approx f(x^\star) \int \exp\left[-\frac{(x-x^*)^2}{2{\sigma^2}^*}\right] dx$$

Integrated Nested Laplace Approximation (INLA)

- The second *ingredient* is **Laplace approximation**
- Main idea: approximate $\log f(x)$ using a quadratic function by means of a Taylor's series expansion around the mode $x^* = \text{argmax}_x \log f(x)$

$$\log f(x) \approx \log f(x^*) + (x - x^*) \left.\frac{\partial \log f(x)}{\partial x}\right|_{x=x^*} + \frac{(x-x^*)^2}{2} \left.\frac{\partial^2 \log f(x)}{\partial x^2}\right|_{x=x^*}$$

$$= \log f(x^*) + \frac{(x-x^*)^2}{2} \left.\frac{\partial^2 \log f(x)}{\partial x^2}\right|_{x=x^*} \qquad \left(\text{since} \left.\frac{\partial \log f(x)}{\partial x}\right|_{x=x^*} = 0\right)$$

- Setting ${\sigma^2}^* = -1/\left.\frac{\partial^2 \log f(x)}{\partial x^2}\right|_{x=x^*}$ we can re-write

$$\log f(x) \approx \log f(x^*) - \frac{1}{2{\sigma^2}^*}(x - x^*)^2$$

or equivalently

$$\int f(x)dx = \int \exp[\log f(x)]dx \approx f(x^\star) \int \exp\left[-\frac{(x-x^*)^2}{2{\sigma^2}^*}\right] dx$$

- Thus, under LA, $f(x) \approx \text{Normal}(x^*, {\sigma^2}^*)$.

Laplace approximation --- example

- Consider a χ^2 distribution con k degrees of freedom: $f(x) = \frac{x^{\frac{k}{2}-1} e^{\frac{-x}{2}}}{2^{k/2}\Gamma(k/2)}$

Laplace approximation --- example

- Consider a χ^2 distribution con k degrees of freedom: $f(x) = \frac{x^{\frac{k}{2}-1} e^{\frac{-x}{2}}}{2^{k/2}\Gamma(k/2)}$

1. $l(x) = \log f(x) = \left(\frac{k}{2} - 1\right) \log x - \frac{x}{2} + \text{constant}$

Laplace approximation --- example

- Consider a χ^2 distribution con k degrees of freedom: $f(x) = \frac{x^{\frac{k}{2}-1} e^{\frac{-x}{2}}}{2^{k/2}\Gamma(k/2)}$

1. $l(x) = \log f(x) = \left(\frac{k}{2} - 1\right) \log x - \frac{x}{2} + \text{constant}$
2. $l'(x) = \frac{\partial \log f(x)}{\partial x} = \left(\frac{k}{2} - 1\right) x^{-1} - \frac{1}{2}$

Laplace approximation --- example

- Consider a χ^2 distribution con k degrees of freedom: $f(x) = \frac{x^{\frac{k}{2}-1} e^{\frac{-x}{2}}}{2^{k/2}\Gamma(k/2)}$

1. $l(x) = \log f(x) = \left(\frac{k}{2} - 1\right) \log x - \frac{x}{2} + \text{constant}$
2. $l'(x) = \frac{\partial \log f(x)}{\partial x} = \left(\frac{k}{2} - 1\right) x^{-1} - \frac{1}{2}$
3. $l''(x) = \frac{\partial^2 \log f(x)}{\partial x^2} = -\left(\frac{k}{2} - 1\right) x^{-2}$

Laplace approximation --- example

- Consider a χ^2 distribution con k degrees of freedom: $f(x) = \frac{x^{\frac{k}{2}-1} e^{\frac{-x}{2}}}{2^{k/2}\Gamma(k/2)}$

1. $l(x) = \log f(x) = \left(\frac{k}{2} - 1\right) \log x - \frac{x}{2} + \text{constant}$
2. $l'(x) = \frac{\partial \log f(x)}{\partial x} = \left(\frac{k}{2} - 1\right) x^{-1} - \frac{1}{2}$
3. $l''(x) = \frac{\partial^2 \log f(x)}{\partial x^2} = -\left(\frac{k}{2} - 1\right) x^{-2}$

- Then
  - Solving $l'(x) = 0$ we find the mode: $x^* = k - 2$
  - Evaluating $-\frac{1}{l''(x)}$ at the mode gives $\sigma^{2^*} = 2(k-2)$

Laplace approximation --- example

- Consider a χ^2 distribution con k degrees of freedom: $f(x) = \frac{x^{\frac{k}{2}-1}e^{\frac{-x}{2}}}{2^{k/2}\Gamma(k/2)}$

1. $l(x) = \log f(x) = \left(\frac{k}{2} - 1\right) \log x - \frac{x}{2} + \text{constant}$
2. $l'(x) = \frac{\partial \log f(x)}{\partial x} = \left(\frac{k}{2} - 1\right) x^{-1} - \frac{1}{2}$
3. $l''(x) = \frac{\partial^2 \log f(x)}{\partial x^2} = -\left(\frac{k}{2} - 1\right) x^{-2}$

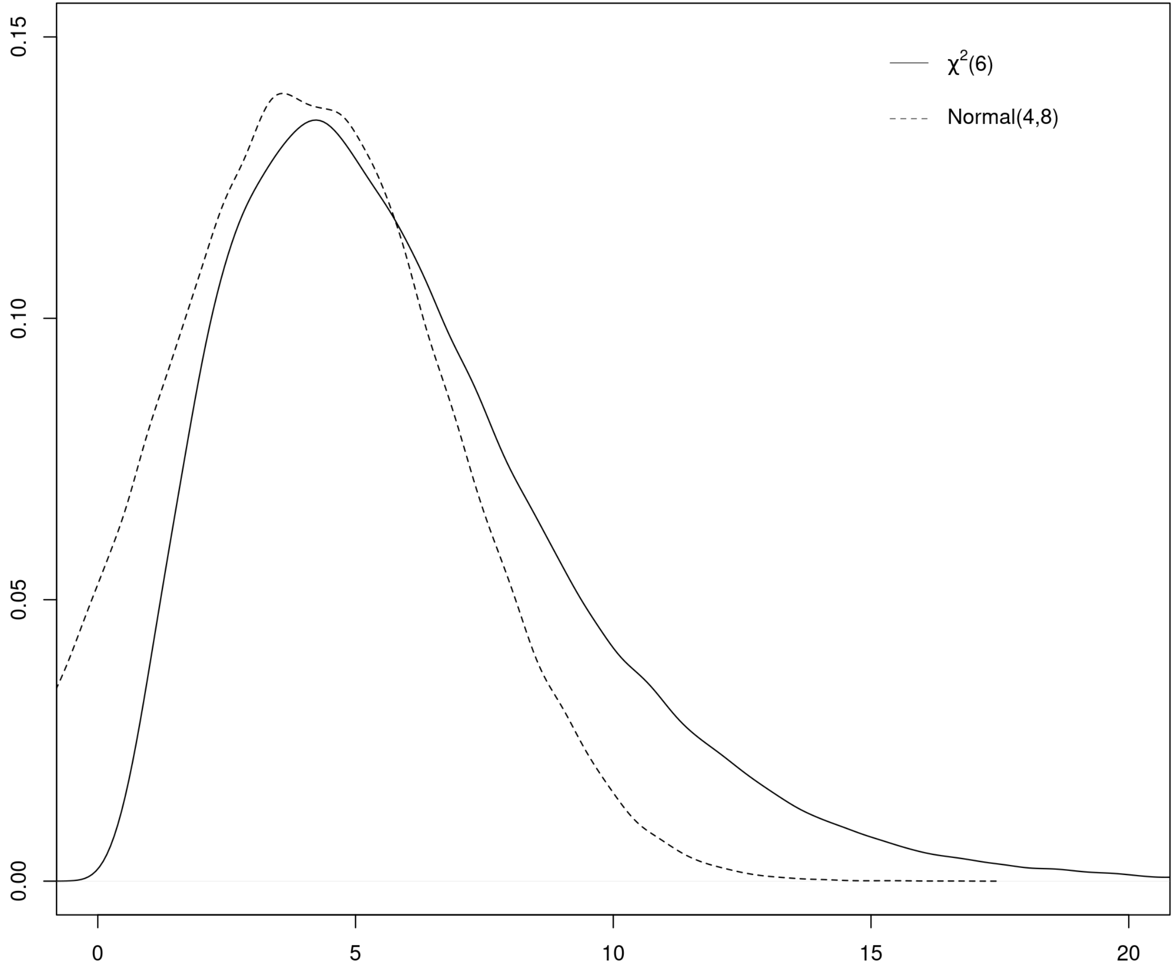
- Then
  - Solving $l'(x) = 0$ we find the mode: $x^* = k - 2$
  - Evaluating $-\frac{1}{l''(x)}$ at the mode gives $\sigma^{2^*} = 2(k-2)$

- Consequently, we can approximate $f(x)$ as

$$f(x) \approx \text{Normal}(k-2, 2(k-2))$$

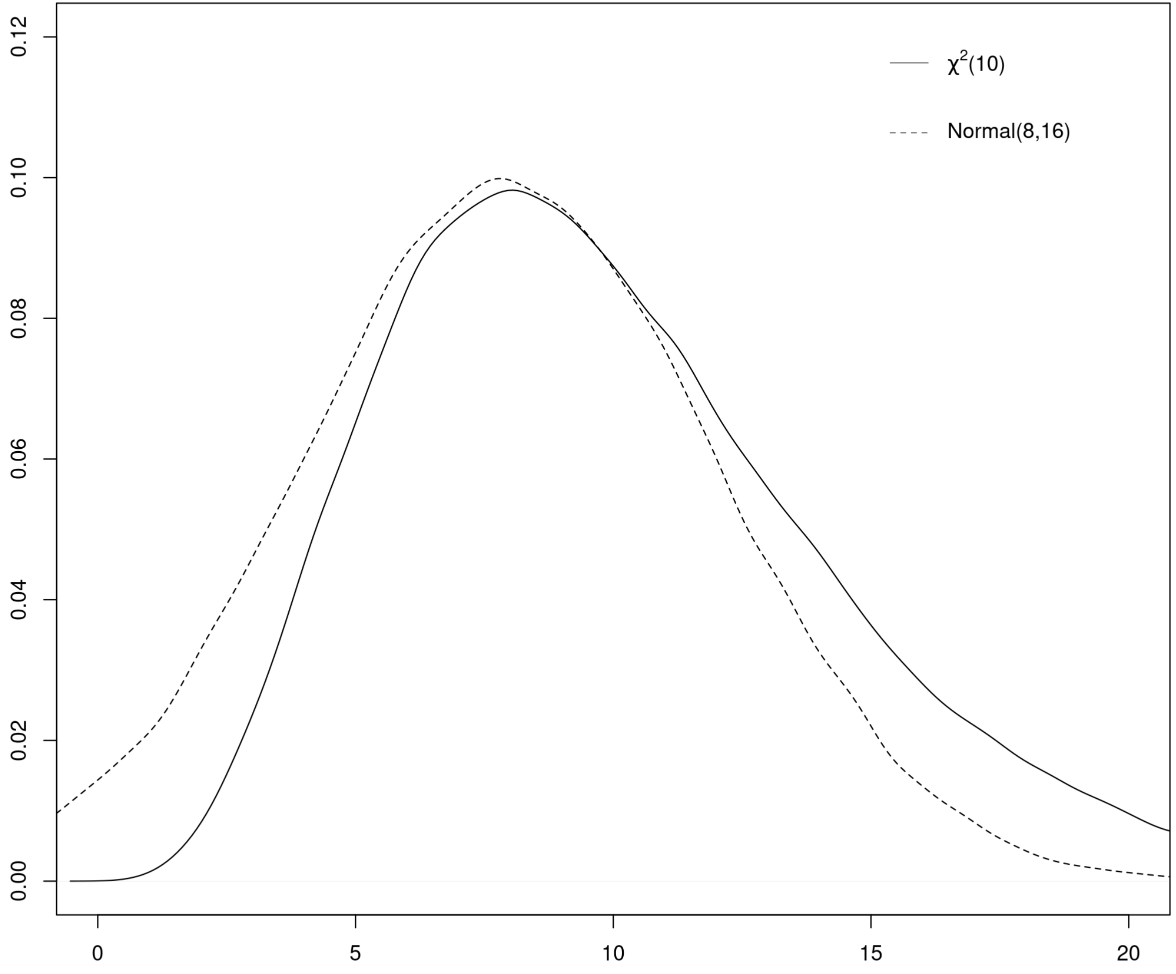
Laplace approximation --- example

Fig 1 Fig 2 Fig 3



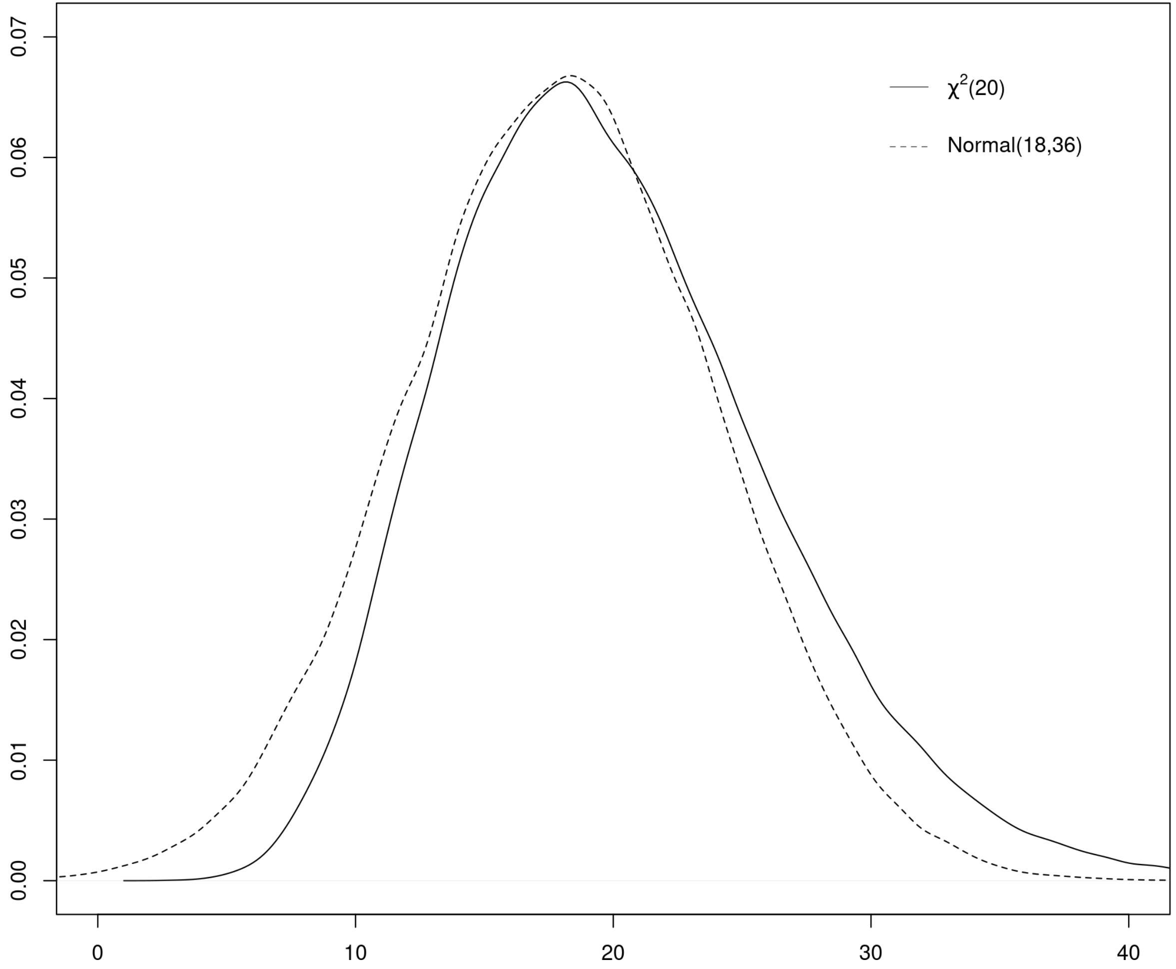
Laplace approximation --- example

Fig 1 | Fig 2 | Fig 3



Laplace approximation --- example

Fig 1 Fig 2 **Fig 3**



Laplace approximation --- example

- Consider a Gamma distribution: $f(x) = \frac{b^a}{\Gamma(a)} \exp\left(-bx\right) x^{a-1} \qquad x, a, b > 0.$

Laplace approximation --- example

- Consider a Gamma distribution: $f(x) = \frac{b^a}{\Gamma(a)} \exp\left(-bx\right) x^{a-1} \qquad x, a, b > 0.$

1. $\log f(x) = (a-1)\log x - bx + \text{constant}$

Laplace approximation --- example

- Consider a Gamma distribution: $f(x) = \frac{b^a}{\Gamma(a)} \exp\left(-bx\right) x^{a-1} \qquad x, a, b > 0.$

1. $\log f(x) = (a-1)\log x - bx + \text{constant}$
2. $l'(x) = \frac{\partial \log f(x)}{\partial x} = \frac{a-1}{x} - b$

Laplace approximation --- example

- Consider a Gamma distribution: $f(x) = \frac{b^a}{\Gamma(a)} \exp\left(-bx\right) x^{a-1} \qquad x, a, b > 0.$

1. $\log f(x) = (a-1)\log x - bx + \text{constant}$
2. $l'(x) = \frac{\partial \log f(x)}{\partial x} = \frac{a-1}{x} - b$
3. $l''(x) = \frac{\partial^2 \log f(x)}{\partial x^2} = -\frac{a-1}{x^2}$

Laplace approximation --- example

- Consider a Gamma distribution: $f(x) = \frac{b^a}{\Gamma(a)}\text{exp}\left(-bx\right)x^{a-1} \qquad x, a, b > 0.$

1. $\text{log} f(x) = (a-1)\text{log} x - bx + \text{constant}$
2. $l'(x) = \frac{\partial \text{log} f(x)}{\partial x} = \frac{a-1}{x} - b$
3. $l''(x) = \frac{\partial^2 \text{log} f(x)}{\partial x^2} = -\frac{a-1}{x^2}$

- Then
  - Solving $l'(x) = 0$ we find the mode: $\frac{a-1}{b}$
  - Evaluating $-\frac{1}{l''(x)}$ at the mode gives $\frac{a-1}{b^2}$

Laplace approximation --- example

- Consider a Gamma distribution: $f(x) = \frac{b^a}{\Gamma(a)} \exp(-bx)\, x^{a-1} \qquad x, a, b > 0.$

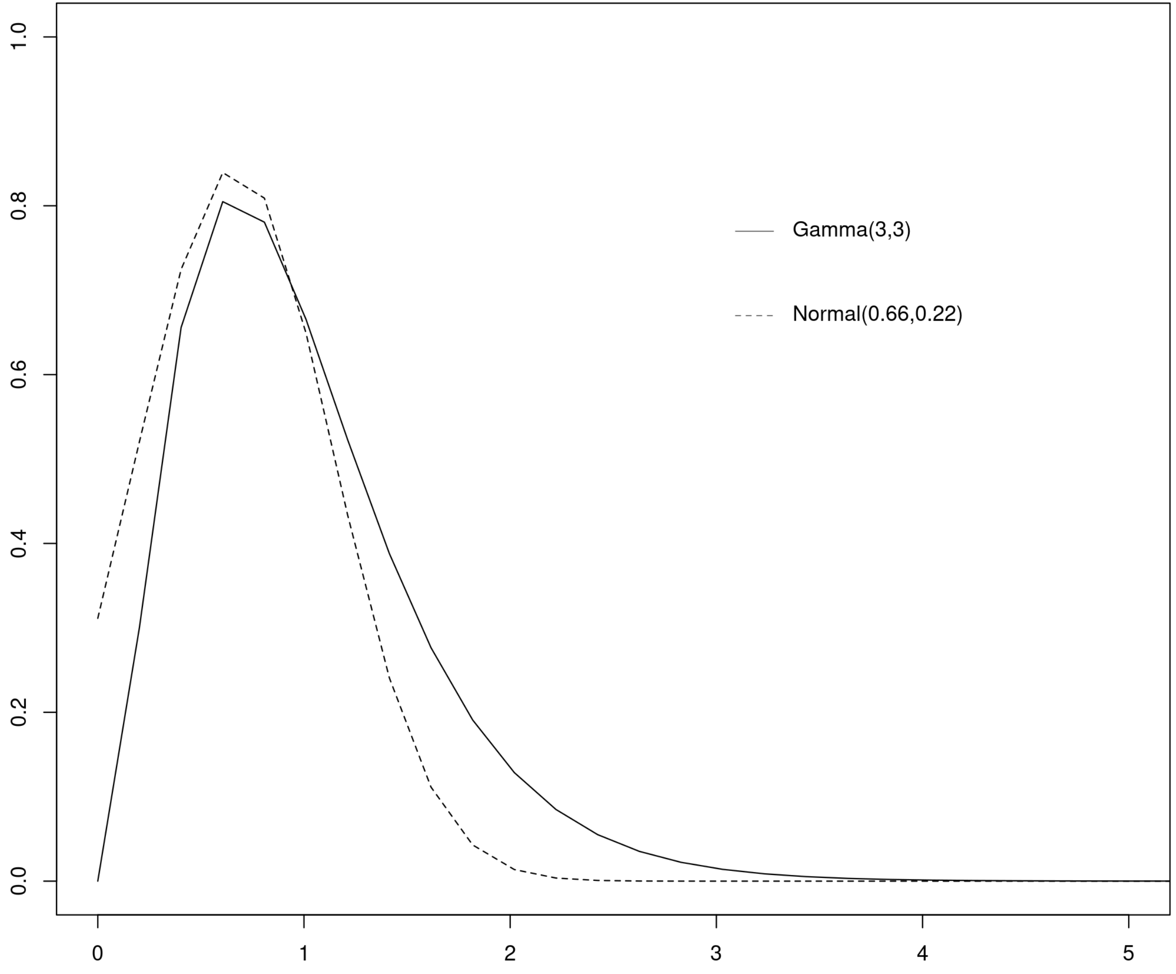
1. $\log f(x) = (a-1)\log x - bx + \text{constant}$
2. $l'(x) = \frac{\partial \log f(x)}{\partial x} = \frac{a-1}{x} - b$
3. $l''(x) = \frac{\partial^2 \log f(x)}{\partial x^2} = -\frac{a-1}{x^2}$

- Then
  - Solving $l'(x) = 0$ we find the mode: $\frac{a-1}{b}$
  - Evaluating $-\frac{1}{l''(x)}$ at the mode gives $\frac{a-1}{b^2}$
- Consequently, we can approximate $f(x)$ as

$$Gamma(a,b) \approx \text{Normal}(x^* = \frac{a-1}{b}, \sigma^{2*} = \frac{a-1}{b^2})$$

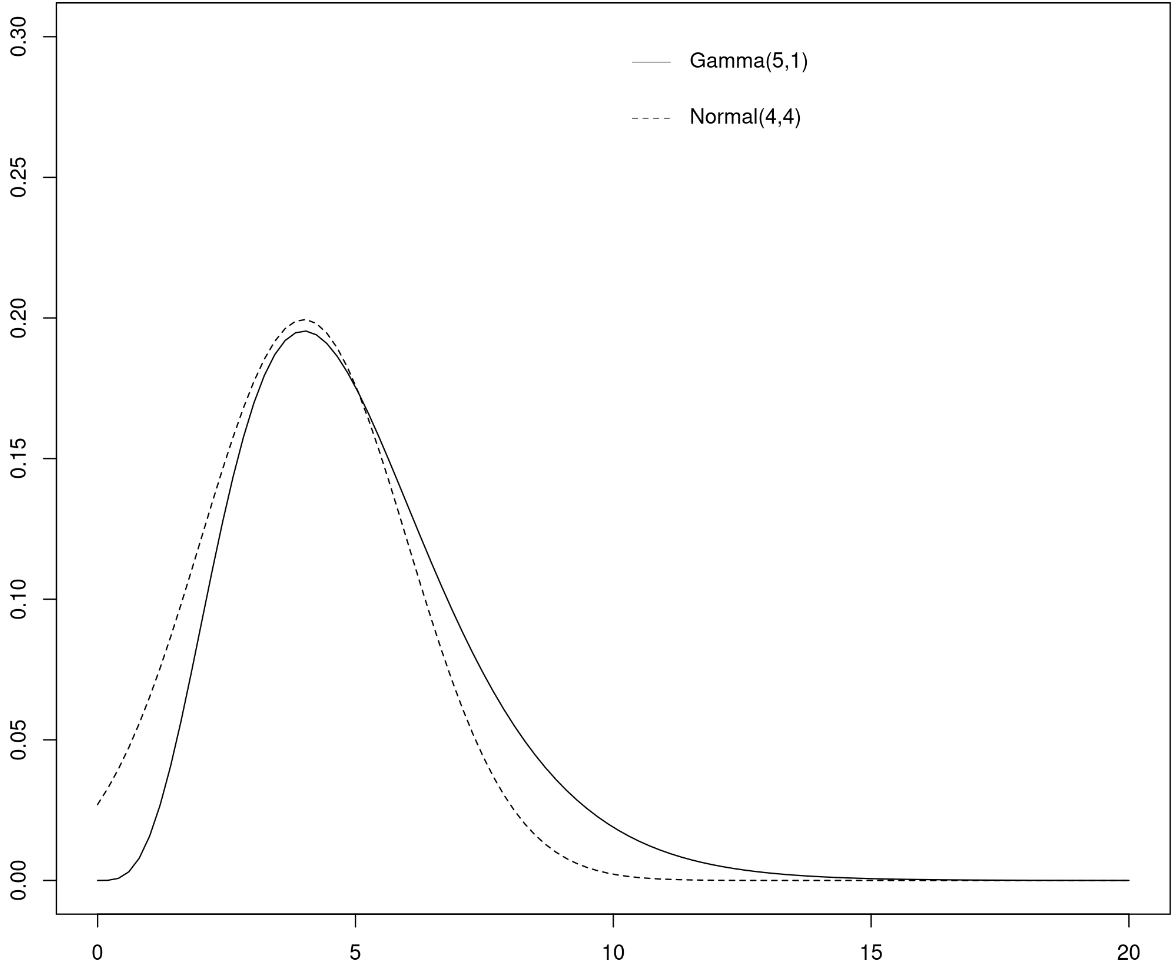
Laplace approximation --- example

Fig1 Fig 2 Fig 3



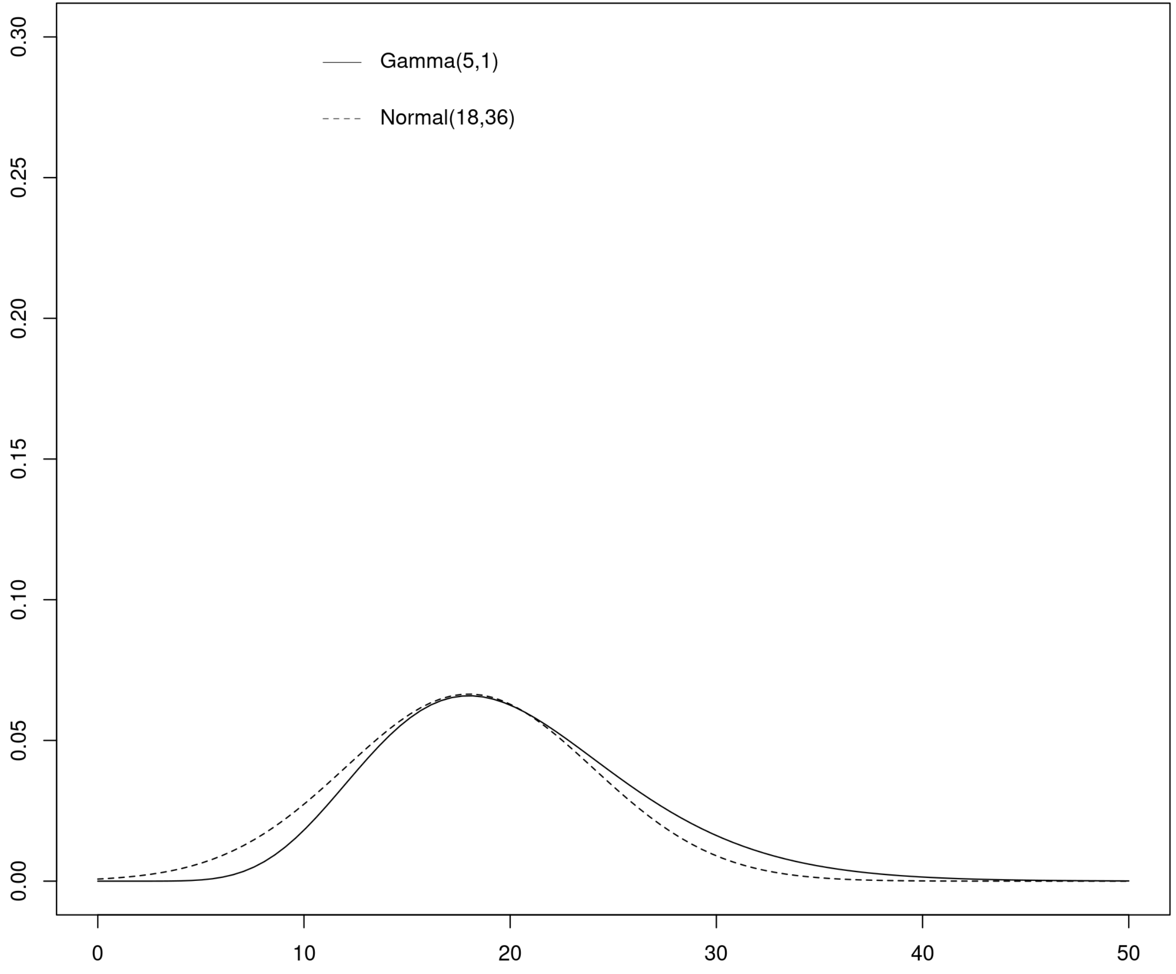
Laplace approximation --- example

Fig1 | **Fig 2** | Fig 3



Laplace approximation --- example

Fig1 Fig 2 **Fig 3**



Integrated Nested Laplace Approximation (INLA)

Objective of Bayesian estimation

- In a Bayesian LGM, the required distributions are

$$p(\theta_i \mid \boldsymbol{y}) = \int p(\theta_i, \boldsymbol{\psi} \mid \boldsymbol{y}) d\boldsymbol{\psi} = \int p(\boldsymbol{\psi} \mid \boldsymbol{y}) p(\theta_i \mid \boldsymbol{\psi}, \boldsymbol{y}) d\boldsymbol{\psi}$$

$$p(\psi_k \mid \boldsymbol{y}) = \int p(\boldsymbol{\psi} \mid \boldsymbol{y}) d\boldsymbol{\psi}_{-k}$$

Objective of Bayesian estimation

- In a Bayesian LGM, the required distributions are

$$p(\theta_i \mid \boldsymbol{y}) = \int p(\theta_i, \boldsymbol{\psi} \mid \boldsymbol{y}) d\boldsymbol{\psi} = \int p(\boldsymbol{\psi} \mid \boldsymbol{y}) p(\theta_i \mid \boldsymbol{\psi}, \boldsymbol{y}) d\boldsymbol{\psi}$$

$$p(\psi_k \mid \boldsymbol{y}) = \int p(\boldsymbol{\psi} \mid \boldsymbol{y}) d\boldsymbol{\psi}_{-k}$$

- Thus we need to estimate:
  1. $p(\boldsymbol{\psi} \mid \boldsymbol{y})$, from which also all the relevant marginals $p(\psi_k \mid \boldsymbol{y})$ can be obtained;

Integrated Nested Laplace Approximation (INLA)

Objective of Bayesian estimation

- In a Bayesian LGM, the required distributions are

$$p(\theta_i \mid \boldsymbol{y}) = \int p(\theta_i, \boldsymbol{\psi} \mid \boldsymbol{y}) d\boldsymbol{\psi} = \int p(\boldsymbol{\psi} \mid \boldsymbol{y}) p(\theta_i \mid \boldsymbol{\psi}, \boldsymbol{y}) d\boldsymbol{\psi}$$

$$p(\psi_k \mid \boldsymbol{y}) = \int p(\boldsymbol{\psi} \mid \boldsymbol{y}) d\boldsymbol{\psi}_{-k}$$

- Thus we need to estimate:
  1. $p(\boldsymbol{\psi} \mid \boldsymbol{y})$, from which also all the relevant marginals $p(\psi_k \mid \boldsymbol{y})$ can be obtained;
  2. $p(\theta_i \mid \boldsymbol{\psi}, \boldsymbol{y})$, which is needed to compute the marginal posterior for the parameters

Integrated Nested Laplace Approximation (INLA)

1. can be easily estimated as

$$\color{red}{p(\boldsymbol{\psi} \mid \boldsymbol{y})} = \frac{p(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \boldsymbol{y})}{p(\boldsymbol{\theta} \mid \boldsymbol{\psi}, \boldsymbol{y})} \text{ (Recall slide 17)}$$

Integrated Nested Laplace Approximation (INLA)

1. can be easily estimated as

$$
\begin{aligned}
\color{red}{p(\boldsymbol{\psi} \mid \boldsymbol{y})} &= \frac{p(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \boldsymbol{y})}{p(\boldsymbol{\theta} \mid \boldsymbol{\psi}, \boldsymbol{y})} \\
&= \frac{p(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\psi}) p(\boldsymbol{\theta}, \boldsymbol{\psi})}{p(\boldsymbol{y})} \frac{1}{p(\boldsymbol{\theta} \mid \boldsymbol{\psi}, \boldsymbol{y})} \quad \text{(Bayes' theorem)}
\end{aligned}
$$

Integrated Nested Laplace Approximation (INLA)

1. can be easily estimated as

$$
\begin{aligned}
{\color{red}p(\boldsymbol{\psi} \mid \boldsymbol{y})} &= \frac{p(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \boldsymbol{y})}{p(\boldsymbol{\theta} \mid \boldsymbol{\psi}, \boldsymbol{y})} \\
&= \frac{p(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\psi}) p(\boldsymbol{\theta}, \boldsymbol{\psi})}{p(\boldsymbol{y})} \frac{1}{p(\boldsymbol{\theta} \mid \boldsymbol{\psi}, \boldsymbol{y})} \\
&= \frac{p(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\psi}) p(\boldsymbol{\theta} \mid \boldsymbol{\psi}) p(\boldsymbol{\psi})}{p(\boldsymbol{y})} \frac{1}{p(\boldsymbol{\theta} \mid \boldsymbol{\psi}, \boldsymbol{y})} \quad \text{(Conditional probability)}
\end{aligned}
$$

1. can be easily estimated as

$$
\begin{aligned}
\color{red}{p(\boldsymbol{\psi} \mid \boldsymbol{y})} &= \frac{p(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \boldsymbol{y})}{p(\boldsymbol{\theta} \mid \boldsymbol{\psi}, \boldsymbol{y})} \\
&= \frac{p(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\psi}) p(\boldsymbol{\theta}, \boldsymbol{\psi})}{p(\boldsymbol{y})} \frac{1}{p(\boldsymbol{\theta} \mid \boldsymbol{\psi}, \boldsymbol{y})} \\
&= \frac{p(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\psi}) p(\boldsymbol{\theta} \mid \boldsymbol{\psi}) p(\boldsymbol{\psi})}{p(\boldsymbol{y})} \frac{1}{p(\boldsymbol{\theta} \mid \boldsymbol{\psi}, \boldsymbol{y})} \\
&\propto \frac{p(\boldsymbol{\psi}) p(\boldsymbol{\theta} \mid \boldsymbol{\psi}) p(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\psi})}{p(\boldsymbol{\theta} \mid \boldsymbol{\psi}, \boldsymbol{y})}
\end{aligned}
$$

1. can be easily estimated as

$$
\begin{aligned}
p(\boldsymbol{\psi} \mid \boldsymbol{y}) &= \frac{p(\boldsymbol{\theta}, \boldsymbol{\psi} \mid \boldsymbol{y})}{p(\boldsymbol{\theta} \mid \boldsymbol{\psi}, \boldsymbol{y})} \\
&= \frac{p(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\psi}) p(\boldsymbol{\theta}, \boldsymbol{\psi})}{p(\boldsymbol{y})} \frac{1}{p(\boldsymbol{\theta} \mid \boldsymbol{\psi}, \boldsymbol{y})} \\
&= \frac{p(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\psi}) p(\boldsymbol{\theta} \mid \boldsymbol{\psi}) p(\boldsymbol{\psi})}{p(\boldsymbol{y})} \frac{1}{p(\boldsymbol{\theta} \mid \boldsymbol{\psi}, \boldsymbol{y})} \\
&\propto \frac{p(\boldsymbol{\psi}) p(\boldsymbol{\theta} \mid \boldsymbol{\psi}) p(\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\psi})}{p(\boldsymbol{\theta} \mid \boldsymbol{\psi}, \boldsymbol{y})} \\
&\approx \left. \frac{p(\boldsymbol{\psi}) p(\boldsymbol{\theta} \mid \boldsymbol{\psi}) p(\boldsymbol{y} \mid \boldsymbol{\theta})}{\tilde{p}(\boldsymbol{\theta} \mid \boldsymbol{\psi}, \boldsymbol{y})} \right|_{\boldsymbol{\theta} = \boldsymbol{\theta}^*(\boldsymbol{\psi})} =: \tilde{p}(\boldsymbol{\psi} \mid \boldsymbol{y})
\end{aligned}
$$

where

- $\tilde{p}(\boldsymbol{\theta} \mid \boldsymbol{\psi}, \boldsymbol{y})$ is the Gaussian approximation given by the Laplace method of $p(\boldsymbol{\theta} \mid \boldsymbol{\psi}, \boldsymbol{y})$
- $\boldsymbol{\theta} = \boldsymbol{\theta}^*(\boldsymbol{\psi})$ is its mode for a given $\boldsymbol{\psi}$

Integrated Nested Laplace Approximation (INLA)

2. is slightly more complex, because in general there will be more elements in $\boldsymbol{\theta}$ than there are in $\boldsymbol{\psi}$ and thus this computation is more expensive.

2. is slightly more complex, because in general there will be more elements in $\boldsymbol{\theta}$ than there are in $\boldsymbol{\psi}$ and thus this computation is more expensive.

Gaussian Approximation: one easy and cheap possibility is to start from $\tilde{p}(\boldsymbol{\theta} \mid \boldsymbol{\psi}, \boldsymbol{y})$ and approximate the density of $\theta_i \mid \boldsymbol{\psi}, \boldsymbol{y}$, with the Gaussian marginals. While this is very fast, the approximation is generally not very good.

2. is slightly more complex, because in general there will be more elements in $\boldsymbol{\theta}$ than there are in $\boldsymbol{\psi}$ and thus this computation is more expensive.

Gaussian Approximation: one easy and cheap possibility is to start from $\tilde{p}(\boldsymbol{\theta} \mid \boldsymbol{\psi}, \boldsymbol{y})$ and approximate the density of $\theta_i \mid \boldsymbol{\psi}, \boldsymbol{y}$, with the Gaussian marginals. While this is very fast, the approximation is generally not very good.

Full Laplace Approximation: alternatively, we can write $\boldsymbol{\theta} = \{\theta_i, \boldsymbol{\theta}_{-i}\}$, use the definition of conditional probability and again Laplace approximation to obtain $\tilde{p}(\theta_i \mid \boldsymbol{\psi}, \boldsymbol{y})$ (very computationally expensive)

Integrated Nested Laplace Approximation (INLA)

2. is slightly more complex, because in general there will be more elements in $\boldsymbol{\theta}$ than there are in $\boldsymbol{\psi}$ and thus this computation is more expensive.

Gaussian Approximation: one easy and cheap possibility is to start from $\tilde{p}(\boldsymbol{\theta} \mid \boldsymbol{\psi}, \boldsymbol{y})$ and approximate the density of $\theta_i \mid \boldsymbol{\psi}, \boldsymbol{y}$, with the Gaussian marginals. While this is very fast, the approximation is generally not very good.

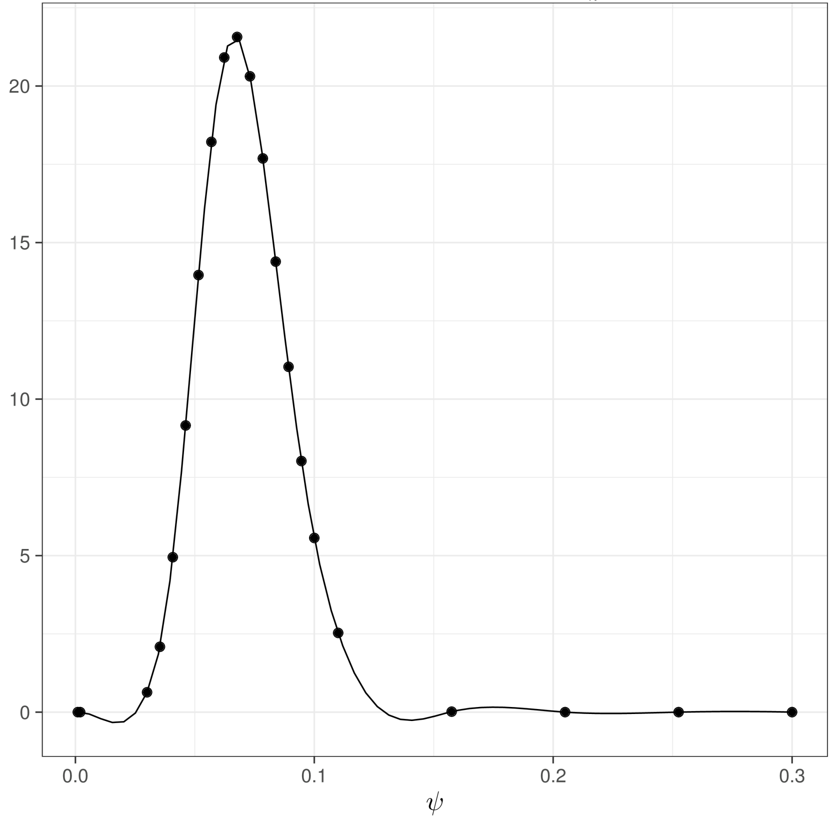
Full Laplace Approximation: alternatively, we can write $\boldsymbol{\theta} = \{\theta_i, \boldsymbol{\theta}_{-i}\}$, use the definition of conditional probability and again Laplace approximation to obtain $\tilde{p}(\theta_i \mid \boldsymbol{\psi}, \boldsymbol{y})$ (very computationally expensive)

The most efficient algorithm is the **Simplified Laplace Approximation**:

- Based on a Taylor's series expansion of the Laplace approximation $\tilde{p}(\theta_i \mid \boldsymbol{\psi}, \boldsymbol{y})$.
- The accuracy of this approximation is sufficient in many applied cases and that the computing time is considerably shorter, it is the standard option.

1. Select a grid of H points $\{\psi_h^*\}$ and area weights $\{\Delta_h\}$; interpolate the density to approximate to the posterior

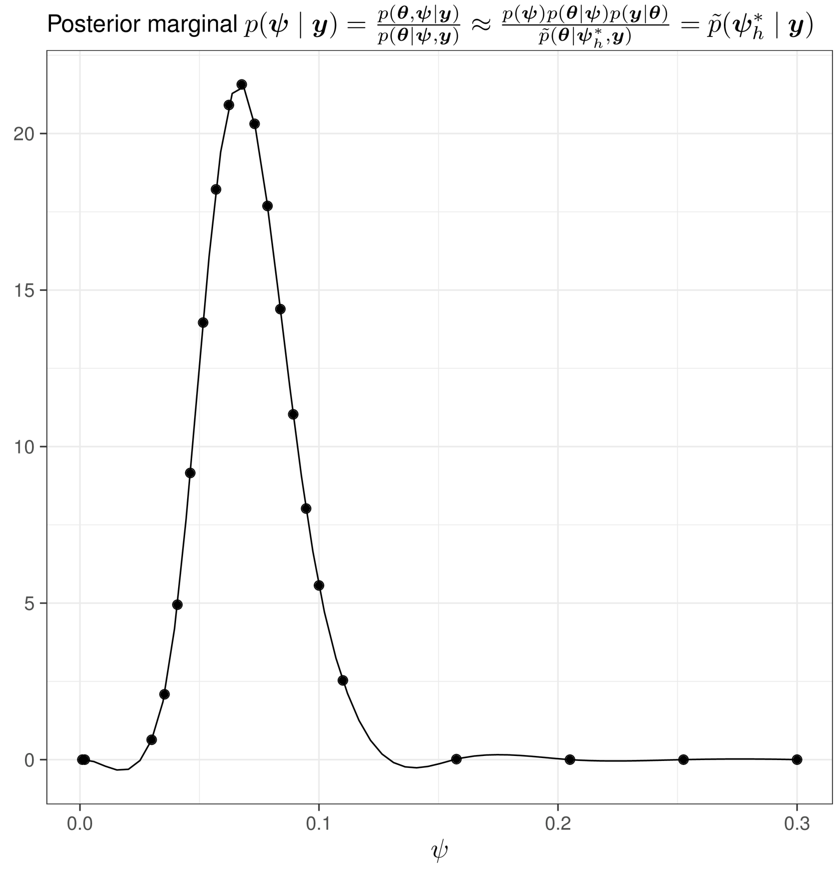
Posterior marginal $p(\psi \mid \boldsymbol{y}) = \frac{p(\boldsymbol{\theta},\boldsymbol{\psi}|\boldsymbol{y})}{p(\boldsymbol{\theta}|\boldsymbol{\psi},\boldsymbol{y})} \approx \frac{p(\boldsymbol{\psi})p(\boldsymbol{\theta}|\boldsymbol{\psi})p(\boldsymbol{y}|\boldsymbol{\theta})}{\tilde{p}(\boldsymbol{\theta}|\boldsymbol{\psi}_h^*,\boldsymbol{y})} = \tilde{p}(\boldsymbol{\psi}_h^* \mid \boldsymbol{y})$

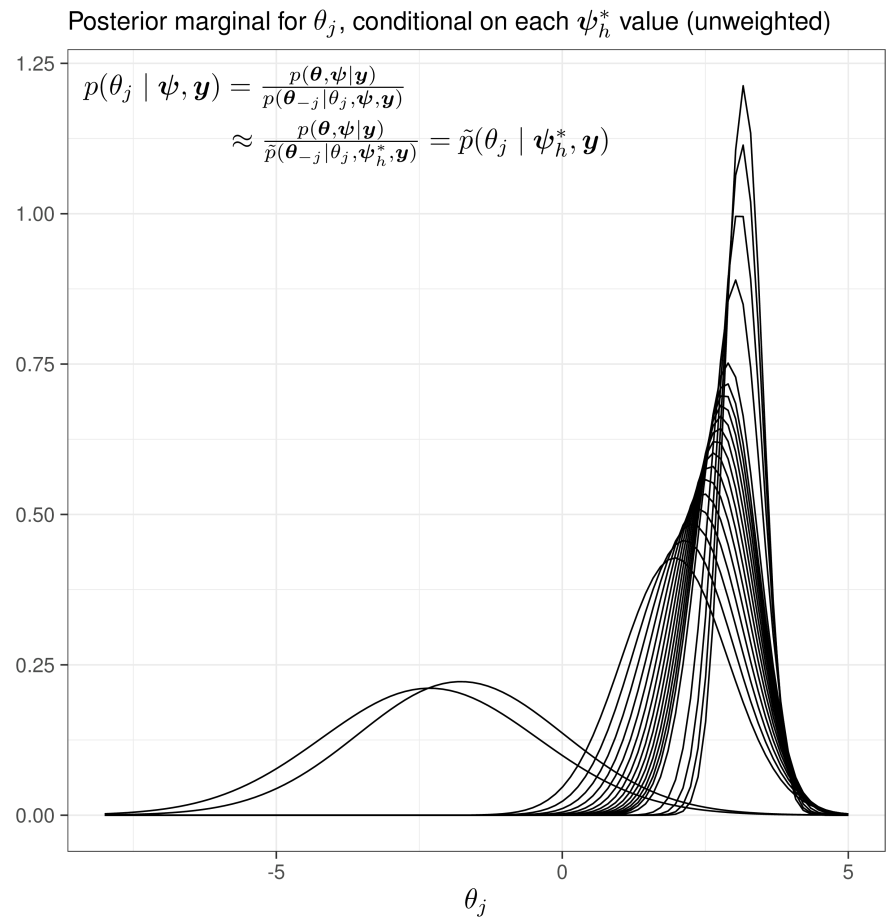


INLA – in a nutshell...

1. Select a grid of H points $\{\boldsymbol{\psi}_h^*\}$ and area weights $\{\Delta_h\}$; interpolate the density to approximate to the posterior

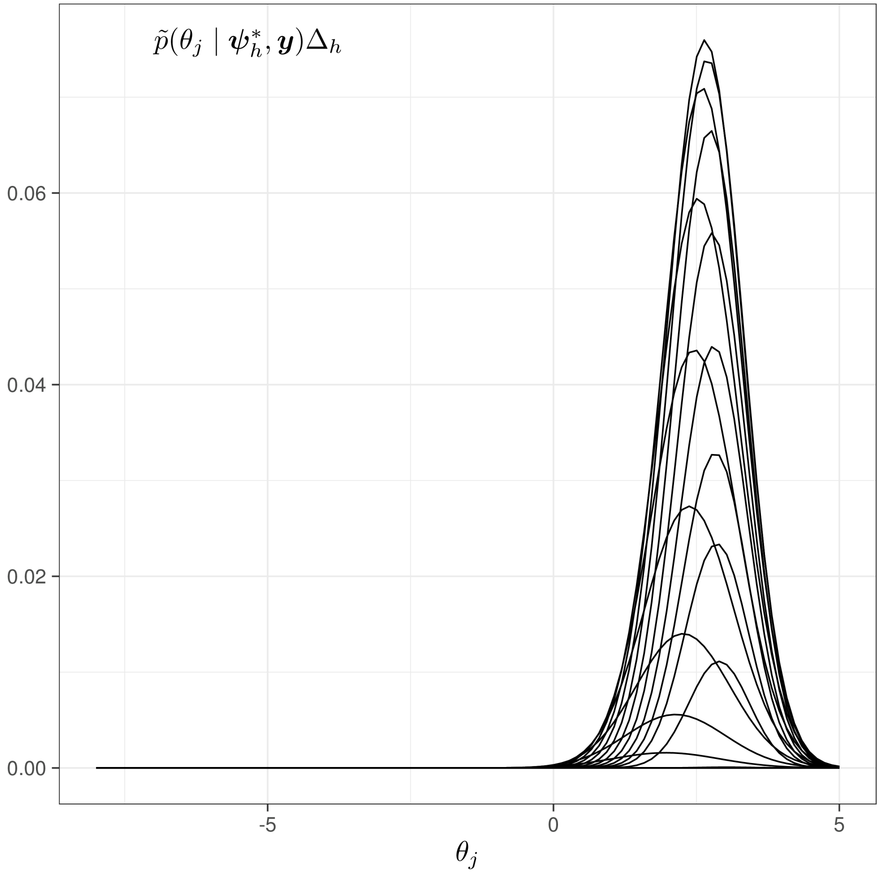
2. Approximates the conditional posterior of each θ_j, given $\boldsymbol{\psi}, \boldsymbol{y}$ on the $H-$dimensional grid





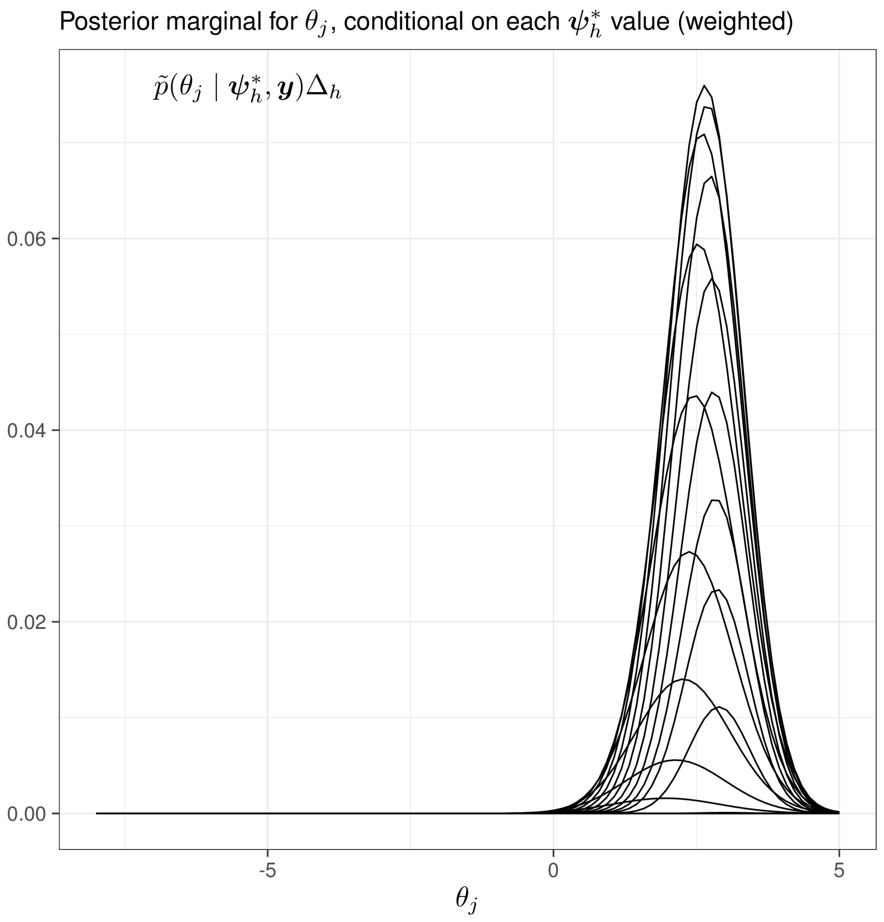
3. Weight the conditional marginal posteriors by the density associated with each ψ_h^*

Posterior marginal for θ_j, conditional on each ψ_h^* value (weighted)

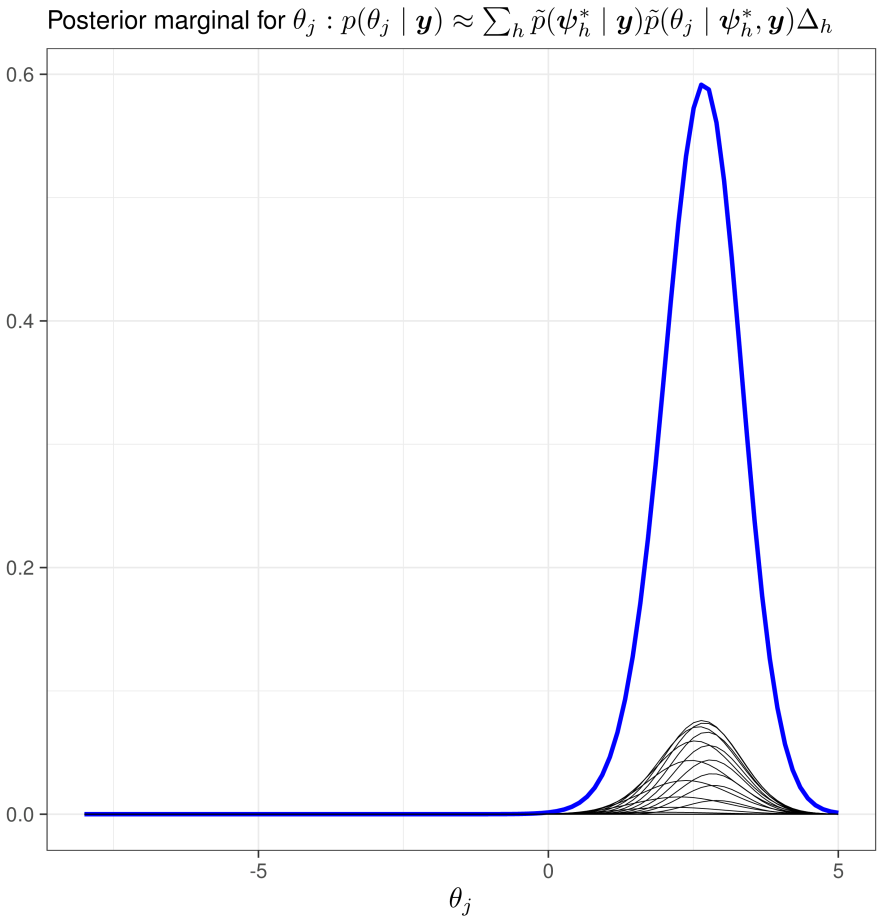


INLA – in a nutshell...

3. Weight the conditional marginal posteriors by the density associated with each ψ_h^*



4. (Numerically) sum over all the conditional densities to obtain the marginal posterior for θ_j



Integrated Nested Laplace Approximation (INLA)

So, it's all in the name...

Integrated Nested **Laplace Approximation**

- Because Laplace approximation is the basis to estimate the unknown distributions

Integrated Nested Laplace Approximation (INLA)

So, it's all in the name...

Integrated **Nested** Laplace Approximation

- Because Laplace approximation is the basis to estimate the unknown distributions
- Because the Laplace approximations are nested within one another
  - Since (1.) is needed to estimate (2.)
  - NB: Consequently the estimation of (1.) might not be good enough, but it can be refined (eg using a finer grid)

Integrated Nested Laplace Approximation (INLA)

So, it's all in the name...

Integrated Nested Laplace Approximation

- Because Laplace approximation is the basis to estimate the unknown distributions

- Because the Laplace approximations are nested within one another
  – Since (1.) is needed to estimate (2.)
  – NB: Consequently the estimation of (1.) might not be good enough, but it can be refined (eg using a finer grid)

- Because the required marginal posterior distributions are obtained by (numerical) integration.

The `R-INLA` package

The INLA package for R

- Good news is that all the procedures needed to perform INLA are implemented in a R package. This is effectively made by two components

The `INLA` package for R

- Good news is that all the procedures needed to perform INLA are implemented in a `R` package. This is effectively made by two components

1. The `GMRFLib` library

- A `C` library for fast and exact simulation of GMRFs

The `INLA` package for `R`

- Good news is that all the procedures needed to perform INLA are implemented in a `R` package. This is effectively made by two components

1. The `GMRFLib` library
   - A `C` library for fast and exact simulation of GMRFs
2. The `inla` program
   - A standalone `C` program build upon the `GMRFLib` library (it performs the relevant computation and returns the results in a standardised way)

The INLA package for R

- Good news is that all the procedures needed to perform INLA are implemented in a `R` package. This is effectively made by two components

1. The `GMRFLib` library
   - A `C` library for fast and exact simulation of GMRFs
2. The `inla` program
   - A standalone `C` program build upon the `GMRFLib` library (it performs the relevant computation and returns the results in a standardised way)

NB: Because the package `R-INLA` relies on a standalone `C` program (and other reasons...), it is not available directly from `CRAN`.

`R-INLA` runs natively under Linux, Windows and Mac and it is possible to do multi-threading using `OpenMP`

The INLA package R - Installation

- From R, installation of the stable version is performed typing

```
install.packages("INLA", repos="http://www.math.ntnu.no/inla/R/stable")
```

- Later, you can upgrade the package by typing

```
library(INLA)
inla.upgrade()
```

- A test-version (which may contain unstable updates/new functions) can be obtained by typing

```
inla.upgrade(testing=TRUE)
```

- Type `inla.version()` to find out the installed version

Step by step guide to using R-INLA

1. The first thing to do is to **specify the model**
   - For example, assume we have a generic model

$$
\begin{aligned}
y_i &\overset{iid}{\sim} p(y_i \mid \theta_i) \\
\eta_i &= \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + f(z_i)
\end{aligned}
$$

where

- $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2)$ are observed covariates
- $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2) \sim \text{Normal}(0, \tau_1^{-1})$ are unstructured (*fixed*) effects
- $\boldsymbol{z}$ is an **index**. This can be used to include structured (*random*), spatial, spatio-temporal effect, etc.
- $f \sim \text{Normal}(0, \boldsymbol{Q}_f^{-1}(\tau_2))$ is a suitable function used to model the structured effects

Step by step guide to using R-INLA

1. The first thing to do is to **specify the model**

- For example, assume we have a generic model

$$
\begin{aligned}
y_i &\overset{iid}{\sim} p(y_i \mid \theta_i) \\
\eta_i &= \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + f(z_i)
\end{aligned}
$$

where

- $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2)$ are observed covariates
- $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2) \sim \text{Normal}(0, \tau_1^{-1})$ are unstructured (*fixed*) effects
- $\boldsymbol{z}$ is an **index**. This can be used to include structured (*random*), spatial, spatio-temporal effect, etc.
- $f \sim \text{Normal}(0, \boldsymbol{Q}_f^{-1}(\tau_2))$ is a suitable function used to model the structured effects

- As mentioned earlier, this formulation can actually be used to represent quite a wide class of models!

Step by step guide to using R-INLA

- The model is translated in R code using a formula
- This is sort of standard in R (you would do pretty much the same for calls to functions such as lm or glm)

```
formula = y ~ 1+ x1 + x2 + f(name=z, model="...",hyper=...)
```

Step by step guide to using R-INLA

- The model is translated in R code using a `formula`
- This is sort of standard in R (you would do pretty much the same for calls to functions such as `lm` or `glm`)

  `formula = y ~ 1+ x1 + x2 + f(name=z, model="...",hyper=...)`

- The `f()` function can account for several structured nonlinear effects. We have for example:
  - `iid` specify independent random effects
  - `rw1`, `rw2`, `ar1` are smooth effect of covariates or time effects
  - `besag` models spatially structured effects (CAR)
  - `generic` is a user-defined precision matrix
- Type `inla.list.models("latent")` for the complete list and find descriptions at https://www.r-inla.org/documentation
- Some options can be specified for the `f()` term: for example `hyper` is used to specify the prior on the hyperparameters (more later).

Step by step guide to using R-INLA

- The model is translated in R code using a `formula`
- This is sort of standard in R (you would do pretty much the same for calls to functions such as `lm` or `glm`)

```
formula = y ~ 1+ x1 + x2 + f(name=z, model="...",hyper=...)
```

- The `f()` function can account for several structured nonlinear effects. We have for example:
  - `iid` specify independent random effects
  - `rw1`, `rw2`, `ar1` are smooth effect of covariates or time effects
  - `besag` models spatially structured effects (CAR)
  - `generic` is a user-defined precision matrix
- Type `inla.list.models("latent")` for the complete list and find descriptions at https://www.r-inla.org/documentation
- Some options can be specified for the `f()` term: for example `hyper` is used to specify the prior on the hyperparameters (more later).
- It is possible to include in the `formula` several `f()` terms specifying them separately, e.g.

```
formula <- y ~ x1 + x2 + f(z1,...) + f(z2,...) + f(z3,...)
```

Step by step guide to using R-INLA

2. Call the function `inla` to fit the model, specifying the data and options (more on this later),e.g.

```
m = inla(formula, family="...", data=data.frame(y,x1,x2,z))
```

Step by step guide to using `R-INLA`

2. Call the function `inla` to fit the model, specifying the data and options (more on this later),e.g.

```
m = inla(formula, family="...", data=data.frame(y,x1,x2,z))
```

- The data need to be included in a suitable `data.frame`

Step by step guide to using R-INLA

2. Call the function `inla` to fit the model, specifying the data and options (more on this later),e.g.

```
m = inla(formula, family="...", data=data.frame(y,x1,x2,z))
```

- The data need to be included in a suitable `data.frame`
- The distribution of the data (i.e. the likelihood) is specified with the `family` option.
  - Type `` `inla.list.models("likelihood") `` for the complete list of likelihood function (we have `poisson, binomial, gamma, beta, gaussian` and many others).

Step by step guide to using R-INLA

2. Call the function `inla` to fit the model, specifying the data and options (more on this later),e.g.

```
m = inla(formula, family="...", data=data.frame(y,x1,x2,z))
```

- The data need to be included in a suitable `data.frame`
- The distribution of the data (i.e. the likelihood) is specified with the `family` option.
  - Type `` `inla.list.models("likelihood") `` for the complete list of likelihood function (we have `poisson, binomial, gamma, beta, gaussian` and many others).
- Visit https://www.r-inla.org/documentation for the complete description

Step by step guide to using `R-INLA`

The `control.xxx=list(...)` statements in the `inla` function control various part of the INLA program:

- `control.compute`: for computing measures of fit (eg DIC)
- `control.predictor`: for specifying the *Observation matrix* $\boldsymbol{A}$ which links the latent field to the data
- `control.family`: for changing the prior distribution of the likelihood hyperparameters
- `control.fixed`: for changing the prior distribution of the fixed effects
- `control.inla`: for changing the strategy to use for the approximations ('gaussian', 'simplified.laplace' (default) or 'laplace') or the grid exploration strategy
- and many others for expert use.

Step by step guide to using R-INLA

R returns an object `m` in the class `inla`, which has some methods available as for example `summary()` and `plot()`.

Some of the elements in an `inla` object returned by a call to `inla()` are:

Name	Description
`summary.fitted.values`	Summary statistics of fitted values.
`summary.fixed`	Summary statistics of fixed effects.
`summary.random`	Summary statistics of random effects.
`summary.linear.predictor`	Summary statistics of linear predictors.
`marginals.fitted.values`	Posterior marginals of fitted values.
`marginals.fixed`	Posterior marginals of fixed effects.
`marginals.random`	Posterior marginals of random effects.
`marginals.linear.predictor`	Posterior marginals of linear predictors.
`mlik`	Estimate of marginal likelihood.

Source: Gómez-Rubio (2020), Section 2.3.1

Toy example

We consider the `iris` dataset included in the `R` datasets (see `?iris`), regarding the measurements in centimeters of the variables *sepal length* and *width* and *petal length* and *width*, respectively, for 50 flowers from each of 3 species of iris. The species are Iris setosa, versicolor, and virginica. See Section 2.6 of the INLA book.

```
> summary(iris)
```

```
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species  
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50  
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50  
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50  
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199                  
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800                  
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500                  
```

We specify a simple regression model with `Petal.length` and `Petal.width` as dependent and independent variables, respectively:

$$\begin{aligned}\text{Petal.length}_i &\sim \text{Normal}(\eta_i, 1/\sigma_e^2)\\ \eta_i &= \beta_0 + \beta_1\,\text{Petal.width}_i\end{aligned}$$

Toy example: run INLA + exploring the fixed effects output

```
> library(INLA)
> formula <- Petal.Length ~  1 + Petal.Width
> output <- inla(formula, family="gaussian", data=iris)
```

```
> output$summary.fixed
```

```
                mean         sd 0.025quant 0.5quant 0.975quant     mode          kld
(Intercept) 1.083565 0.07293226  0.9401746 1.083563   1.226833 1.083565 1.284615e-06
Petal.Width 2.229935 0.05137177  2.1289336 2.229933   2.330849 2.229935 1.284522e-06
```

- For each unstructured *fixed* effect, `R-INLA` reports a set of summary statistics from the posterior distribution.

Toy example: run INLA + exploring the fixed effects output

```
> library(INLA)
> formula <- Petal.Length ~  1 + Petal.Width
> output <- inla(formula, family="gaussian", data=iris)
```

```
> output$summary.fixed
```

```
                mean         sd 0.025quant 0.5quant 0.975quant     mode          kld
(Intercept) 1.083565 0.07293226  0.9401746 1.083563   1.226833 1.083565 1.284615e-06
Petal.Width 2.229935 0.05137177  2.1289336 2.229933   2.330849 2.229935 1.284522e-06
```

- For each unstructured *fixed* effect, `R-INLA` reports a set of summary statistics from the posterior distribution.
- The value of the Kullback-Leibler divergence `kld` describes the difference between the Gaussian approximation and the Simplified Laplace Approximation (SLA) to the marginal posterior densities:
  - Small values indicate that the posterior distribution is well approximated by a Normal distribution
  - If so, the more sophisticated SLA gives a *good* error rate and therefore there is no need to use the more computationally intensive *full* Laplace approximation.

Exploring the output: hyperparatemers

```
> output$summary.hyperpar
```

```
                                            mean        sd 0.025quant 0.5quant 0.975quant     mode
Precision for the Gaussian observations 4.432532 0.5109594    3.48813 4.412588    5.48995 4.372888
```

- For each hyperparameter the summary statistics are reported to describe the posterior distribution.
- **NB**: `INLA` reports results on the **precision** scale (more on this later).

Manipulating the marginals from R-INLA: fixed effects

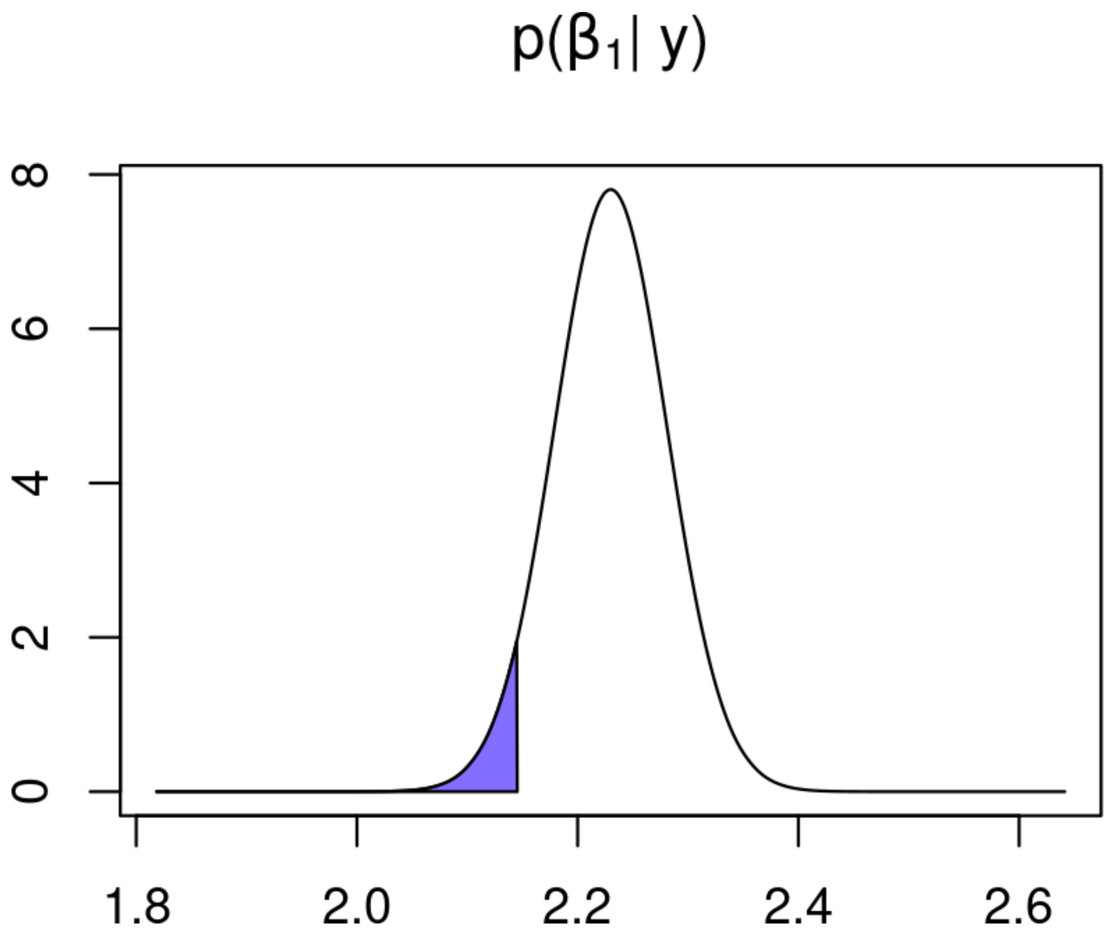
marginals1 marginals2

```
> names(output$marginals.fixed)
```

```
[1] "(Intercept)" "Petal.Width"
```

```
> beta1_post <-output$marginals.fixed[[2]]
> marg <- inla.smarginal(beta1_post)
> q <-inla.qmarginal(0.05,beta1_post)
```

```
> plot(marg,t="l",
+      ylab="",xlab="",
+      main=expression(paste("p(",beta[1], "| y)'
> polygon(c(marg$x[marg$x <= q ], q),
+         c(marg$y[marg$x <= q ], 0),
+         col = "slateblue1", border = 1)
```



Manipulating the marginals from R-INLA: fixed effects

marginals1 marginals2

```
> inla.pmarginal(q,beta1_post)
```

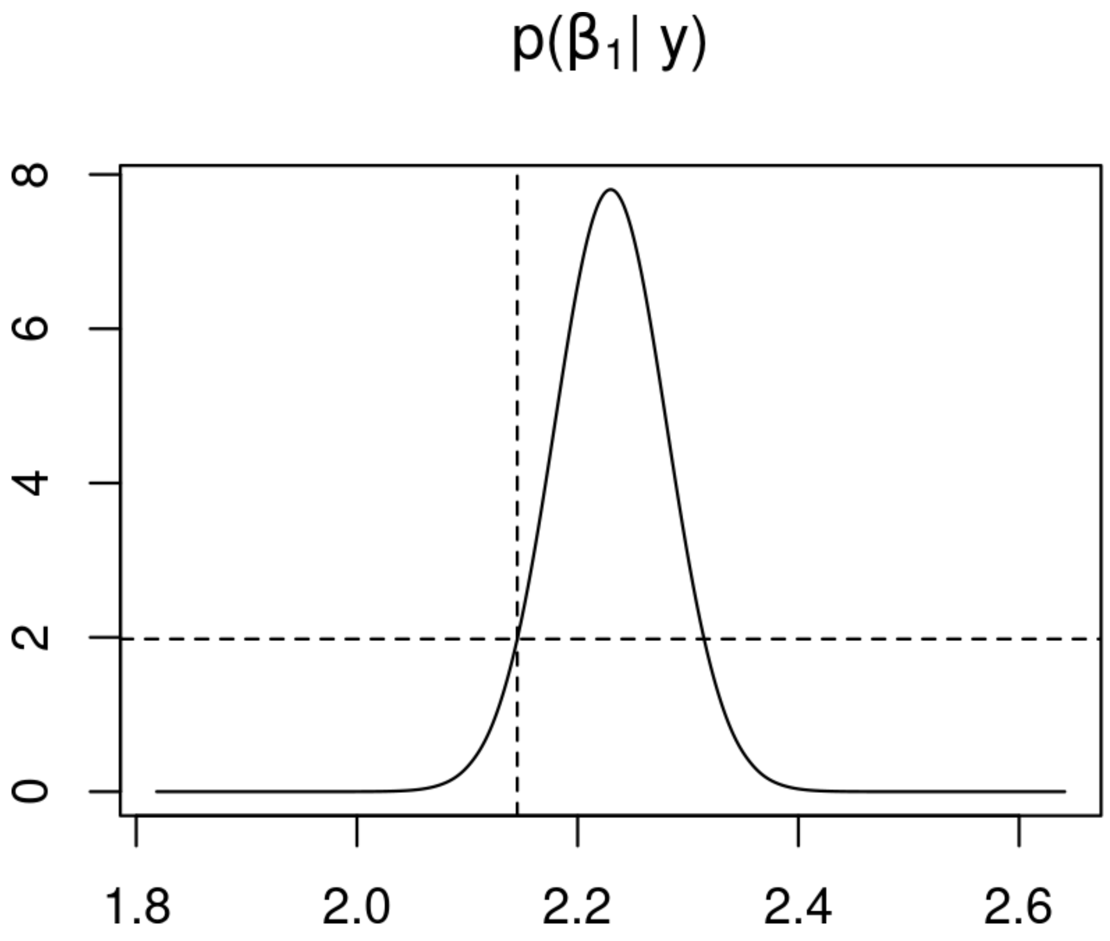
```
[1] 0.05
```

```
> d <-inla.dmarginal(q,beta1_post)
> d
```

```
[1] 1.978538
```

```
> inla.rmarginal(4, beta1_post)
```

```
[1] 2.160636 2.190686 2.205257 2.170303
```



INLA works with the precision by default

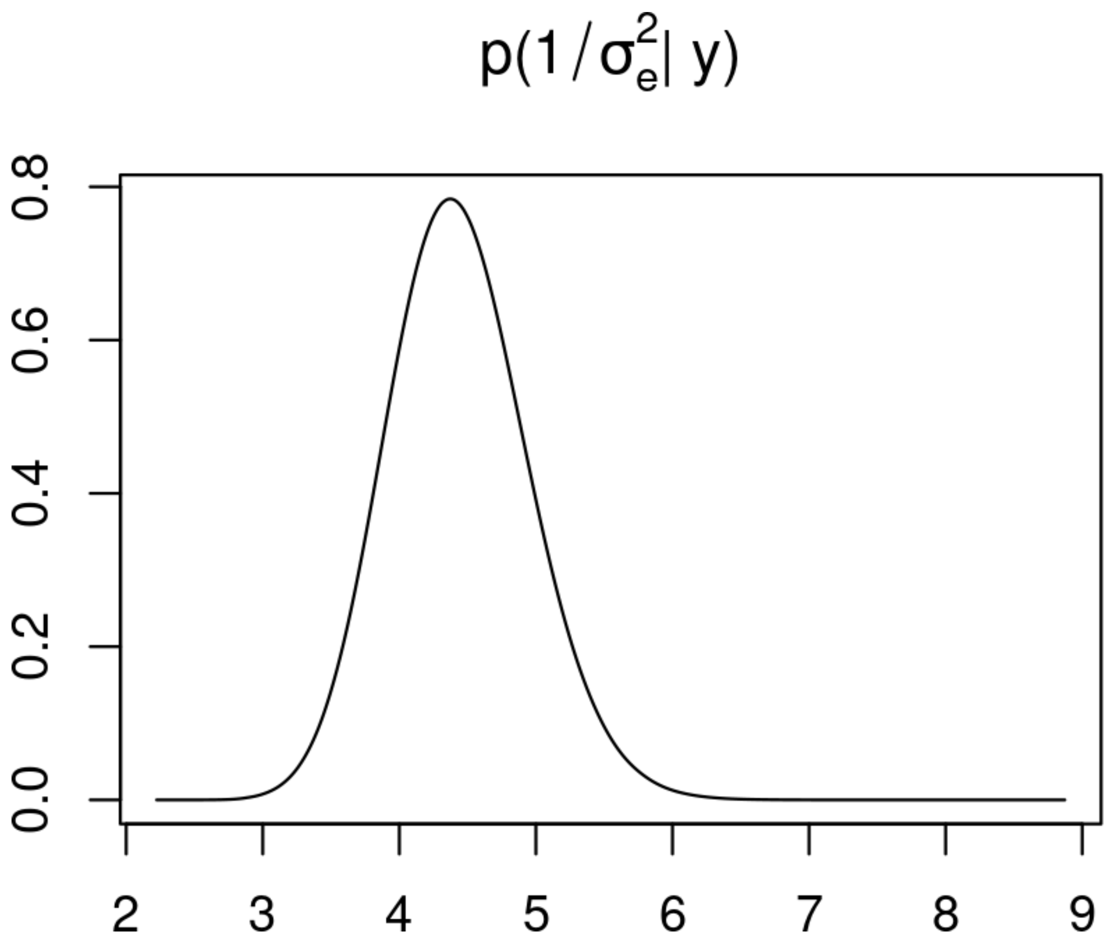
Precision | Variance

```
> names(output$marginals.hyperpar)
```

[1] "Precision for the Gaussian observations"

```
> prec_post <- output$marginals.hyperpar[[1]]
```

```
> plot(inla.smarginal(prec_post),t="l",
+       ylab="",xlab="",
+       main=expression(paste("p(",1/sigma[e]^2, '
```



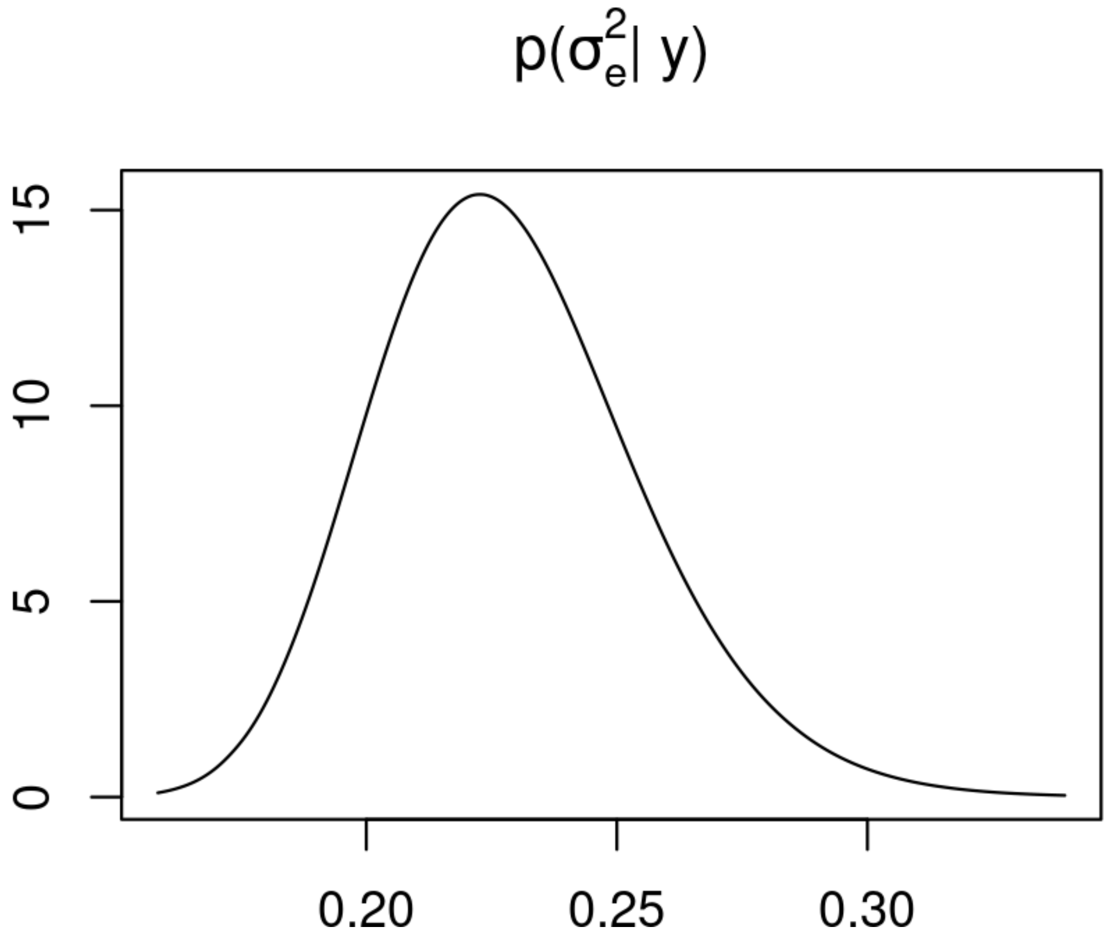
INLA works with the precision by default

Precision | **Variance**

```
> var_post = inla.tmarginal(fun=function(x)
+               1/x, mar=prec_post)
> inla.emarginal(fun=function(x) 1/x,
+ marg=prec_post)
```

```
[1] 0.2286685
```

```
> plot(inla.smarginal(var_post),t="l",
+       ylab="",xlab="",
+       main=expression(paste("p(",sigma[e]^2, "|
```



Summary

The INLA approach is not a rival/competitor/replacement to/of MCMC, just a better option for the class of LGMs.

- The basic idea behind the INLA procedure is simple
  - Repeatedly use Laplace approximation and take advantage of computational simplifications due to the structure of the model
  - Use numerical integration to compute the required posterior marginal distributions
- Complications are mostly computational and occur when
  - Extending to a large number of hyperparameters
  - Markedly non-Gaussian observations

References

Blangiardo, M. and M. Cameletti (2015). *Spatial and spatio-temporal Bayesian models with R-INLA*. John Wiley & Sons.

Gómez-Rubio, V. (2020). *Bayesian inference with INLA*. CRC Press.

Johnson, A. A., M. Q. Ott, and M. Dogucu (2022). *Bayes Rules!: An Introduction to Applied Bayesian Modeling*. CRC Press.

Martins, T. G., D. Simpson, F. Lindgren, et al. (2013). "Bayesian computing with INLA: New features". In: *Computational Statistics & Data Analysis* 67, pp. 68-83.

Rue, H., S. Martino, and N. Chopin (2009). "Approximate Bayesian inference for latent Gaussian model by using integrated nested Laplace approximations (with discussion)". In: *J. R. Statist. Soc. B* 71, pp. 319-392.