Integreted Nested Laplace Approximations, The R-program

Ingelin Steinsland

Department of Mathematical Sciences, NTNU

November 2012

Outline

1. Latent Gaussian Models
2. Examples
3. INLA for Gaussian likelihoods
4. Review key assumptions and ideas
5. INLA for non-Gaussian likelihoods
6. r-INLA, with toturial

Bayesian inference

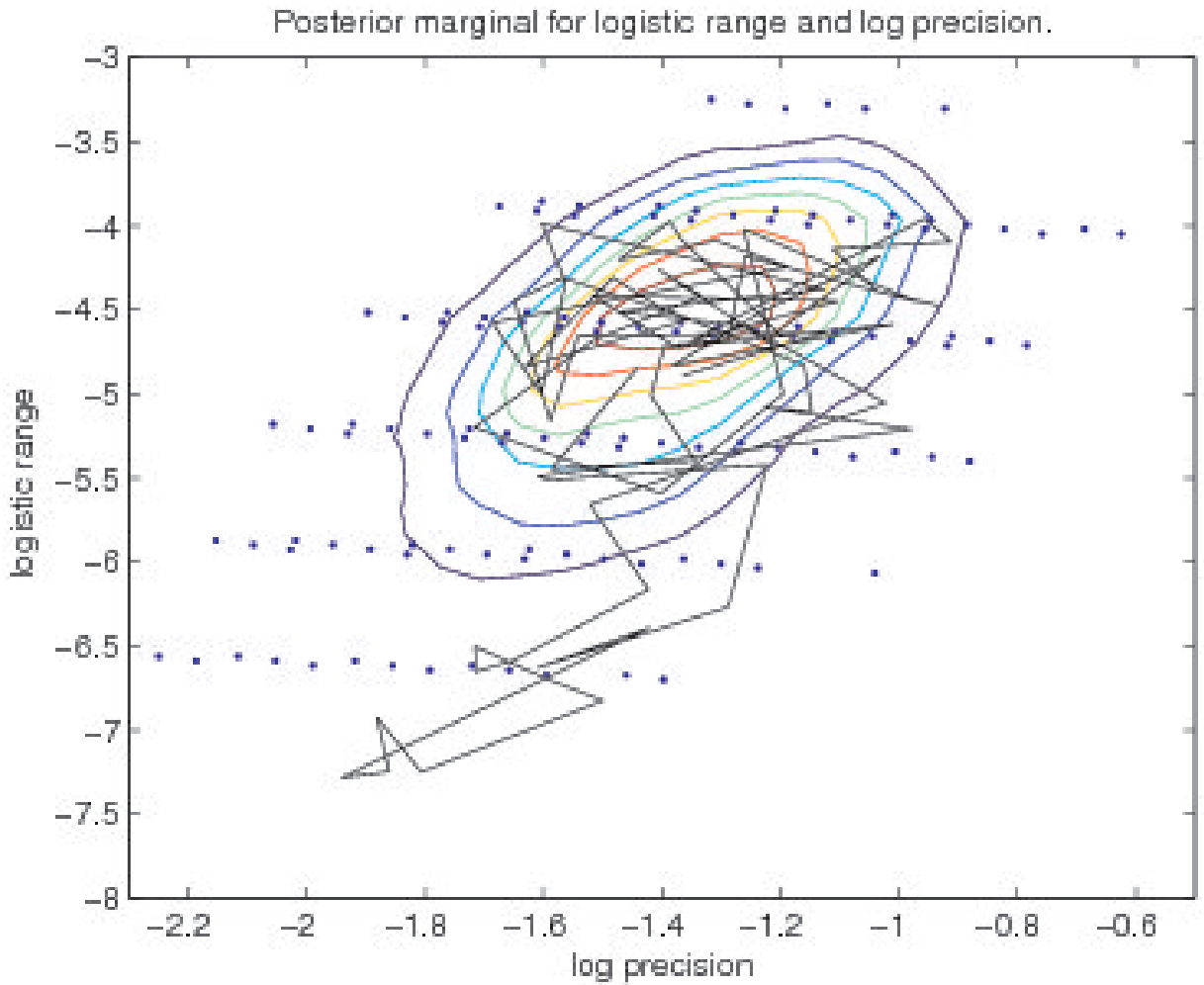
- Data; y
- Likelihood; $\pi(y|\theta)$
- Prior; $\pi(\theta)$
- Posterior $\pi(\theta|y)$

$$\pi(\theta|y) = \frac{\pi(y|\theta)\pi(\theta)}{\pi(y)}$$

If we can not solve this anaytically;

- MCMC
- INLA

INLA vs MCMC 2D



What is INLA?

INLA provides a recipe for computing in a fast and accurate way, approximations to marginal posterior densities for latent Gaussian models. The approximations are based on a smart use of Laplace or other related analytical approximations and of numerical integration schemes.

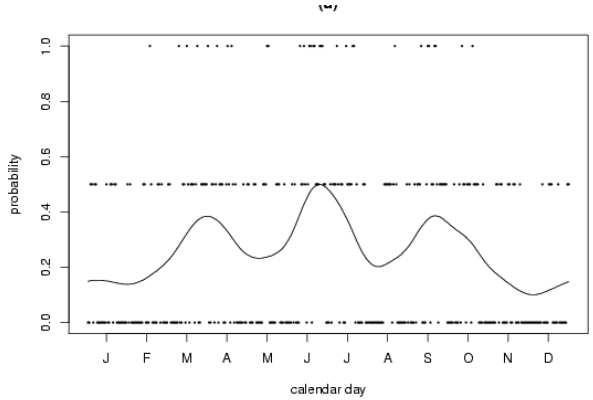
- H. Rue, S. Martino & N. Chopin *Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations*, Journal of the Royal Statistical Society -Series B, 2009

Bayesian hierarchical model

Bayesian latent GMRF models are a subclass of Bayesian hierarchical models:

1. Observation process: $y \sim \pi(y|x,\theta) = \prod_{i=1}^{n} \pi(y_i|x_i)$
2. Latent field: $x \sim \pi(x|\theta)$, $x \sim N(\mu, Q^{-1})$
3. Hyper-parameters: $\theta \sim \pi(\theta)$

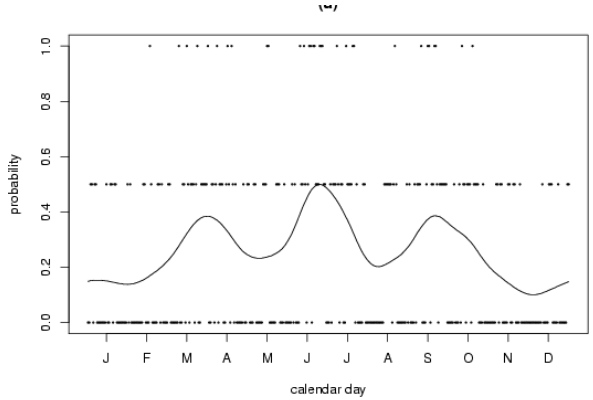
Example(1): Tokyo rainfall data



Stage 1 Binomial data

$$y_i \sim \begin{cases} \text{Binomial}(2, p(x_i)) \\ \text{Binomial}(1, p(x_i)) \end{cases}$$

Example(1): Tokyo rainfall data



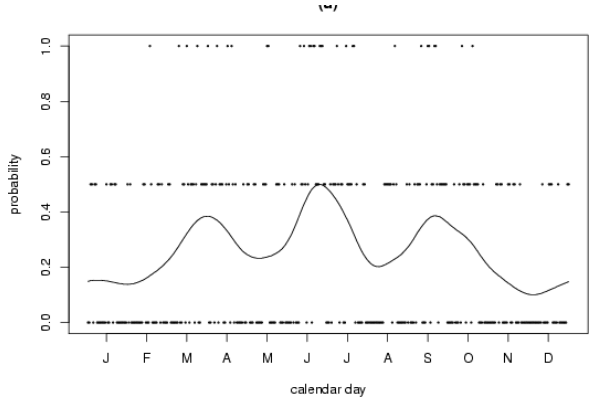
Stage 1 Binomial data

$$y_i \sim \begin{cases} \text{Binomial}(2, p(x_i)) \\ \text{Binomial}(1, p(x_i)) \end{cases}$$

Stage 2 Assume a smooth latent $\boldsymbol{x}$,

$$\boldsymbol{x} \sim RW2(\kappa), \qquad \text{logit}(p_i) = x_i$$

Example(1): Tokyo rainfall data



Stage 1 Binomial data

$$y_i \sim \begin{cases} \text{Binomial}(2, p(x_i)) \\ \text{Binomial}(1, p(x_i)) \end{cases}$$

Stage 2 Assume a smooth latent $\boldsymbol{x}$,

$$\boldsymbol{x} \sim RW2(\kappa), \qquad \text{logit}(p_i) = x_i$$

Stage 3 Gamma(α, β)-prior on κ

Model summary, Tokyo rainfall

$$\pi(\boldsymbol{x} \mid \kappa)\, \pi(\kappa) \prod_i \pi(y_i \mid x_i)$$

where

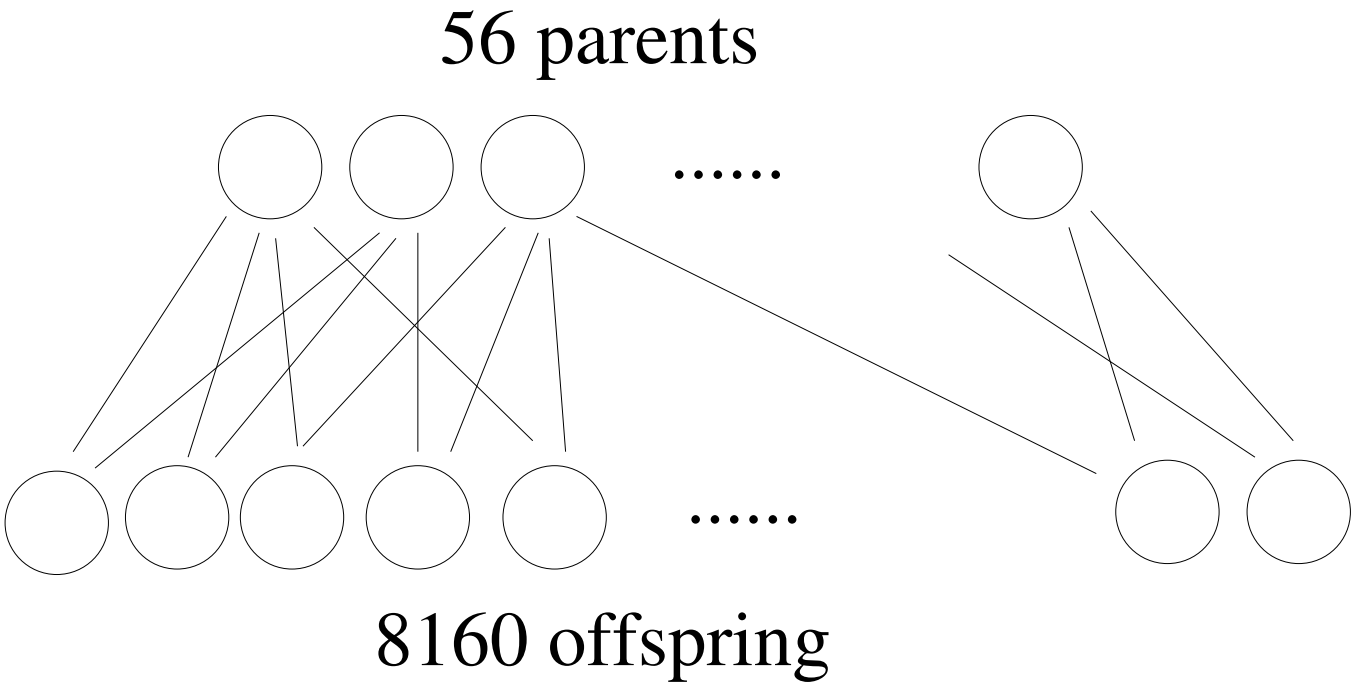
- $y_i|x_i$ is Binomial with $p(x_i)$
- $\boldsymbol{x} \mid \kappa$ is Gaussian (Markov) with dimension 366
- κ is Gamma

Example (II): Scots Pine Breeding

Pedigree 56 unrelated parents, partial diallel design. Original 8160 seedlings.

Spatial location 2.2×2.2m grid, two trail sites.

Data Hight of 4970 26-years-old scots pine.

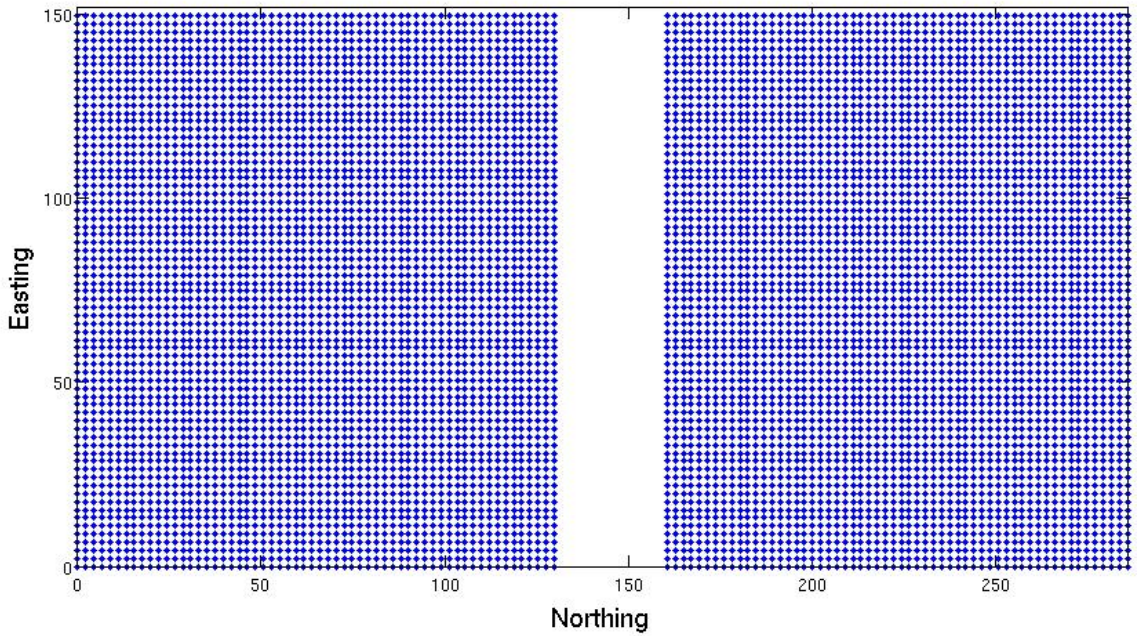


Example (II): Scots Pine Breeding

Pedigree 56 unrelated parents, partial diallel design. Original 8160 seedlings.

Spatial location 2.2×2.2m grid, two trail sites.

Data Hight of 4970 26-years-old scots pine.

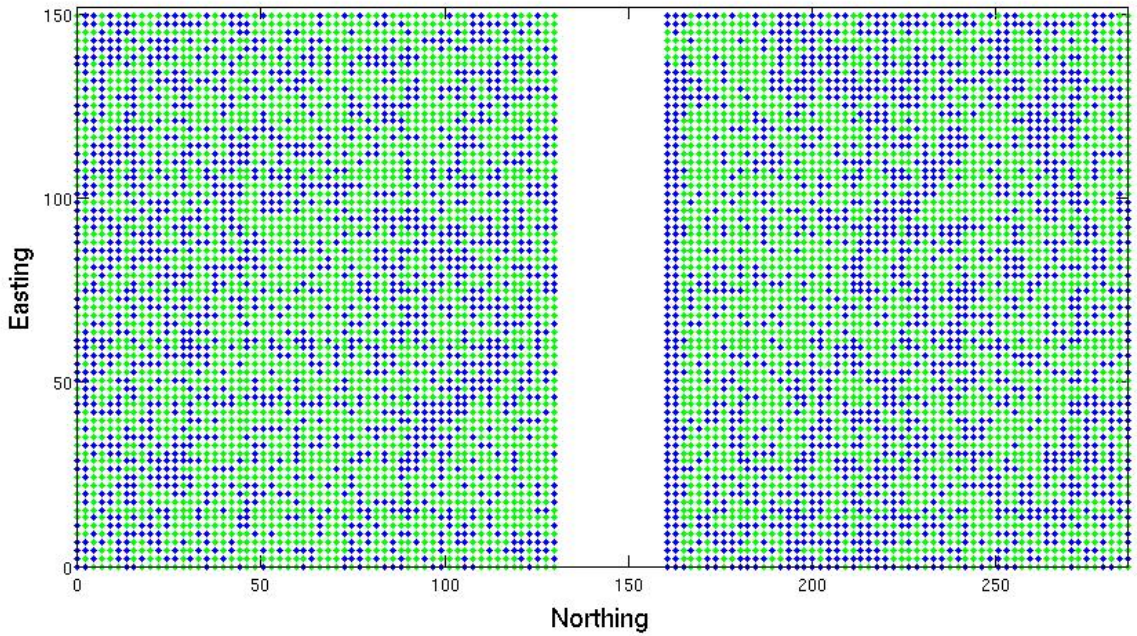


Example (II): Scots Pine Breeding

Pedigree 56 unrelated parents, partial diallel design. Original 8160 seedlings.

Spatial location 2.2×2.2m grid, two trail sites.

Data Hight of 4970 26-years-old scots pine.



Example (II): Scots Pine Breeding

Pedigree 56 unrelated parents, partial diallel design. Original 8160 seedlings.

Spatial location 2.2×2.2m grid, two trail sites.

Data Hight of 4970 26-years-old scots pine.

Provided by Tore Ericsson, Skogforsk, Sweden

Model, hight

Genetic and spatial model:

- Genetic effects
  - Additive, a
  - Dominant, d
- Spatial structure, w

For one tree:

$$y_i = \beta_0 + a_i + d_i + w(s_i) + \epsilon_i$$

Model, hight

Genetic and spatial model:

- Genetic effects
  - Additive, *a*
  - Dominant, *d*
- Spatial structure, *w*

For one tree:

$$y_i = \beta_0 + a_i + d_i + w(s_i) + \epsilon_i$$

For a population:

$$Y = \beta_0 + Za + Zd + w + E$$

$$\begin{aligned} \text{Additive:} \quad & a \sim N(0, \sigma_a^2 A), \\ \text{Dominance:} \quad & d \sim N(0, \sigma_d^2 D) \\ \text{Space:} \quad & w \sim N(0, \sigma_w^2 R(\phi)) \\ \text{Individual effect:} \quad & E \sim N(0, \tau I) \end{aligned}$$

Model summary, Scots pine

$$\pi(\boldsymbol{\eta} \mid \theta)\, \pi(\theta) \prod_i \pi(y_i \mid \eta_i, \theta)$$

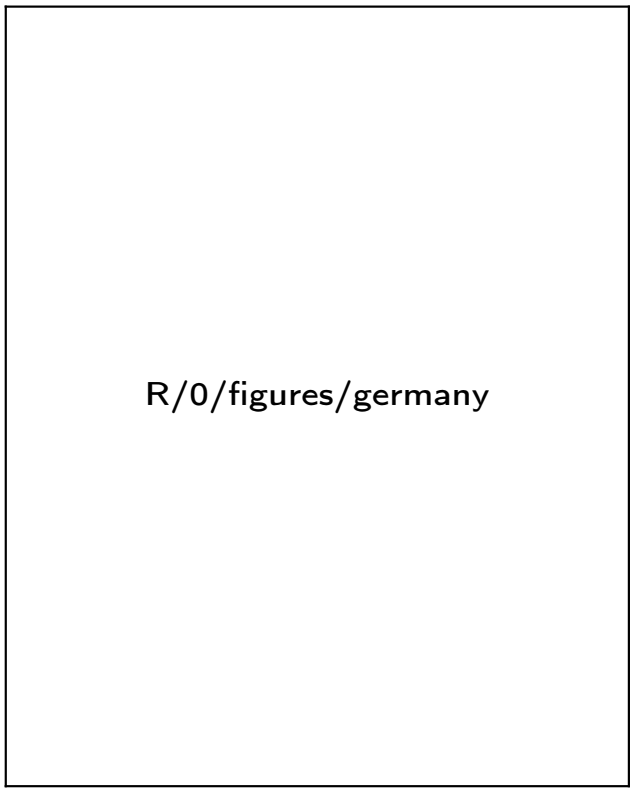
where

- Stage 1: $y_i|x_i$ is Gaussian with mean η_i and variance τ
- Stage 2: $\boldsymbol{x} \mid \theta$ is Gaussian
- Stage 3: θ is inverse Gammas and Gamma

Hyper-parameters: $\theta = (\sigma_a, \sigma_d, \sigma_w, \phi, \tau)$
Latent field: $\boldsymbol{x} = (\boldsymbol{\eta}, \boldsymbol{a}, \boldsymbol{d}, \boldsymbol{w}, \beta_0)$
$\beta_0 + a_i + d_i + w(s_i)$

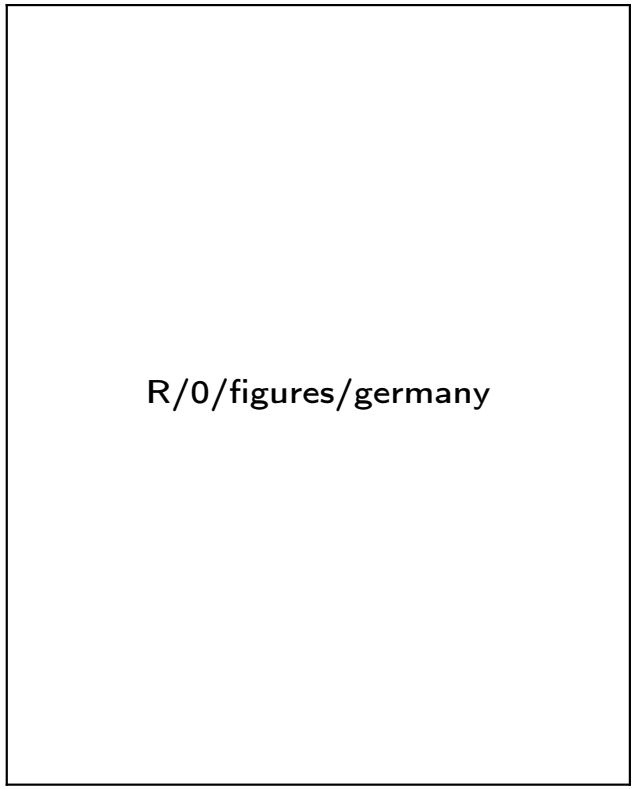
Example (III): Disease mapping in Germany

- Data $y_i \sim \text{Poisson}(E_i exp(\eta_i))$



Example (III): Disease mapping in Germany

- Data $y_i \sim$ Poisson$(E_i exp(\eta_i))$
- Log-relative risk
  $\eta_i = \mu + u_i + v_i + f(c_i)$



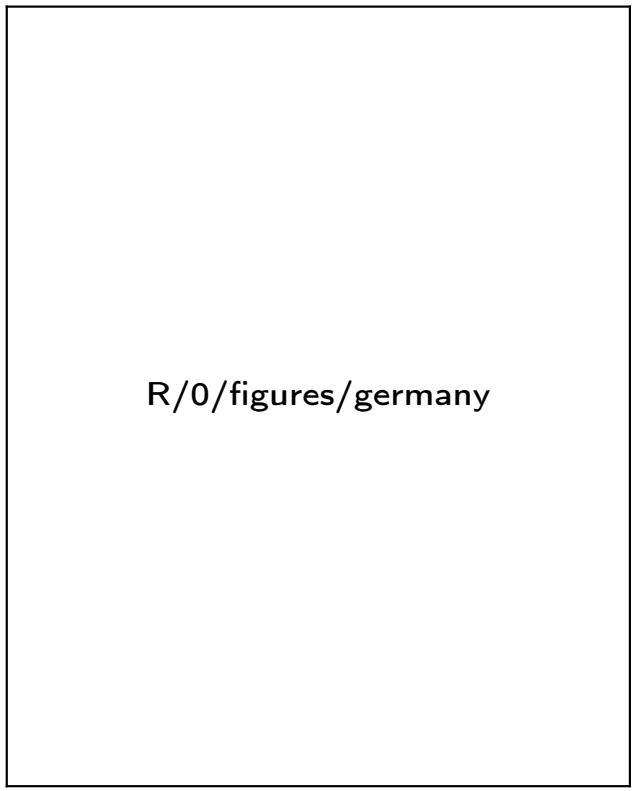
Model summary

- Stage 1: $y_i|x_i$ is Poisson
- Stage 2: $\boldsymbol{x} \mid \theta$ is Gaussian
- Stage 3: θ is ....

Latent field: $x = (\boldsymbol{\eta}, \boldsymbol{u}, \boldsymbol{v}, \mu, \boldsymbol{f})$

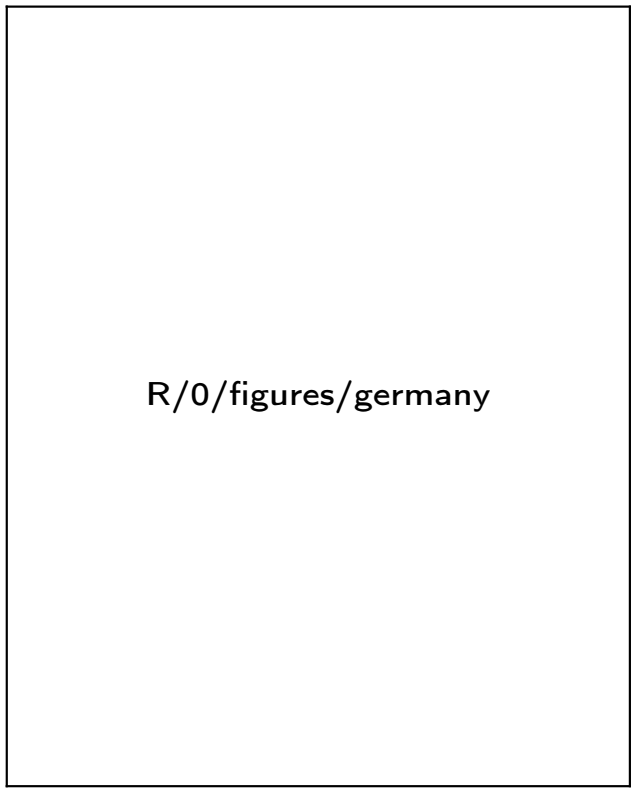
Example (III): Disease mapping in Germany

- Data $y_i \sim$ Poisson$(E_i exp(\eta_i))$
- Log-relative risk
  $\eta_i = \mu + u_i + v_i + f(c_i)$
- Spatially structured component
  $\boldsymbol{u}$



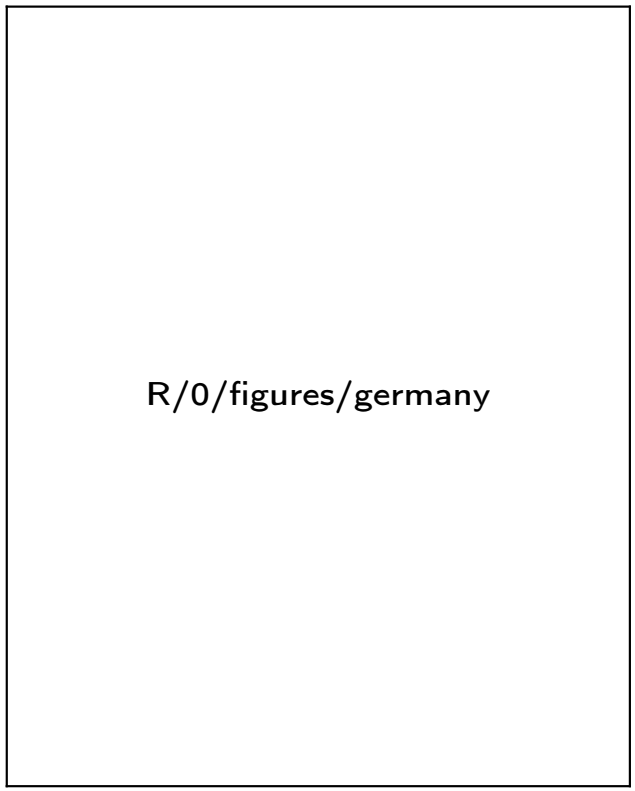
Example (III): Disease mapping in Germany

- Data $y_i \sim$ Poisson$(E_i exp(\eta_i))$
- Log-relative risk
  $\eta_i = \mu + u_i + v_i + f(c_i)$
- Spatially structured component $\boldsymbol{u}$
- Unstructured component $\boldsymbol{v}$

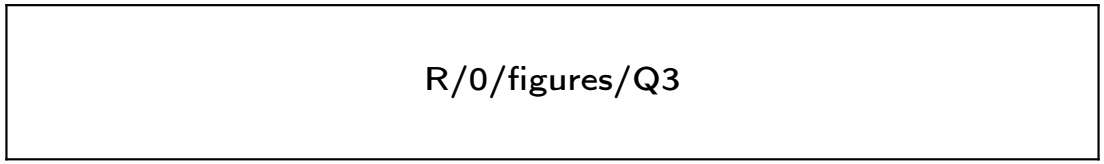


Example (III): Disease mapping in Germany

- Data $y_i \sim$ Poisson$(E_i exp(\eta_i))$
- Log-relative risk
  $\eta_i = \mu + u_i + v_i + f(c_i)$
- Spatially structured component $\boldsymbol{u}$
- Unstructured component $\boldsymbol{v}$
- Smooth effect of a covariate (smoking) $\boldsymbol{c}$



Precision matrix $(\boldsymbol{\eta}, \boldsymbol{u}, \boldsymbol{v}, \mu, \boldsymbol{f})$



We'll just keep reaching for a framework

We can reinterpret the model as

$$\begin{aligned} \boldsymbol{\theta} &\sim \pi(\boldsymbol{\theta}) \\ \boldsymbol{x} \mid \boldsymbol{\theta} &\sim \pi(\boldsymbol{x} \mid \boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}(\boldsymbol{\theta})) \\ \boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta} &\sim \prod_i \pi(y_i \mid \eta_i, \boldsymbol{\theta}) \end{aligned}$$

- $\dim(\boldsymbol{x})$ could be large 10^2-10^5
- $\dim(\boldsymbol{\theta})$ is small 1-5

GxxMs—Different names for the same thing

GLM/GAM/GLMM/GAMM/++

- Perhaps the most important class of statistical models
- Many "models" can be cast in to this class without knowing
- No good (enough) MCMC solution around
- Even frequentist approaches does not scale well computationally

Bayesian GLM/GAM/GLMM/GAMM/+++

Linear predictor

$$\boldsymbol{\eta} = \mu \mathbf{1} + \boldsymbol{A}\boldsymbol{\beta} + \sum_i \boldsymbol{B}_i \boldsymbol{v}_i + \boldsymbol{\epsilon}$$

where

- $\boldsymbol{A}$: covariates, fixed effects $\boldsymbol{\beta}$
- $\boldsymbol{B}_i$: weights, random effects $\boldsymbol{v}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$
- $\boldsymbol{\epsilon}_i$: Possible noise

Observations

$$\boldsymbol{y} \sim \pi(\boldsymbol{y} \mid \boldsymbol{\eta},) = \prod_i \pi(y_i \mid \eta_i)$$

Latent Gaussian Models

This model-construct

Observation process: $\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta} \sim \prod_i \pi(y_i \mid \eta_i, \boldsymbol{\theta})$

Latent field: $\boldsymbol{x} \mid \boldsymbol{\theta} \sim \pi(\boldsymbol{x} \mid \boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}(\boldsymbol{\theta}))$

Hyper-parameters: $\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta})$

Occurs in many, seemingly unrelated, statistical models.

We call these models *latent Gaussian models*!

Latent Gaussian Models

This model-construct

Observation process: $\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta} \sim \prod_i \pi(y_i \mid \eta_i, \boldsymbol{\theta})$

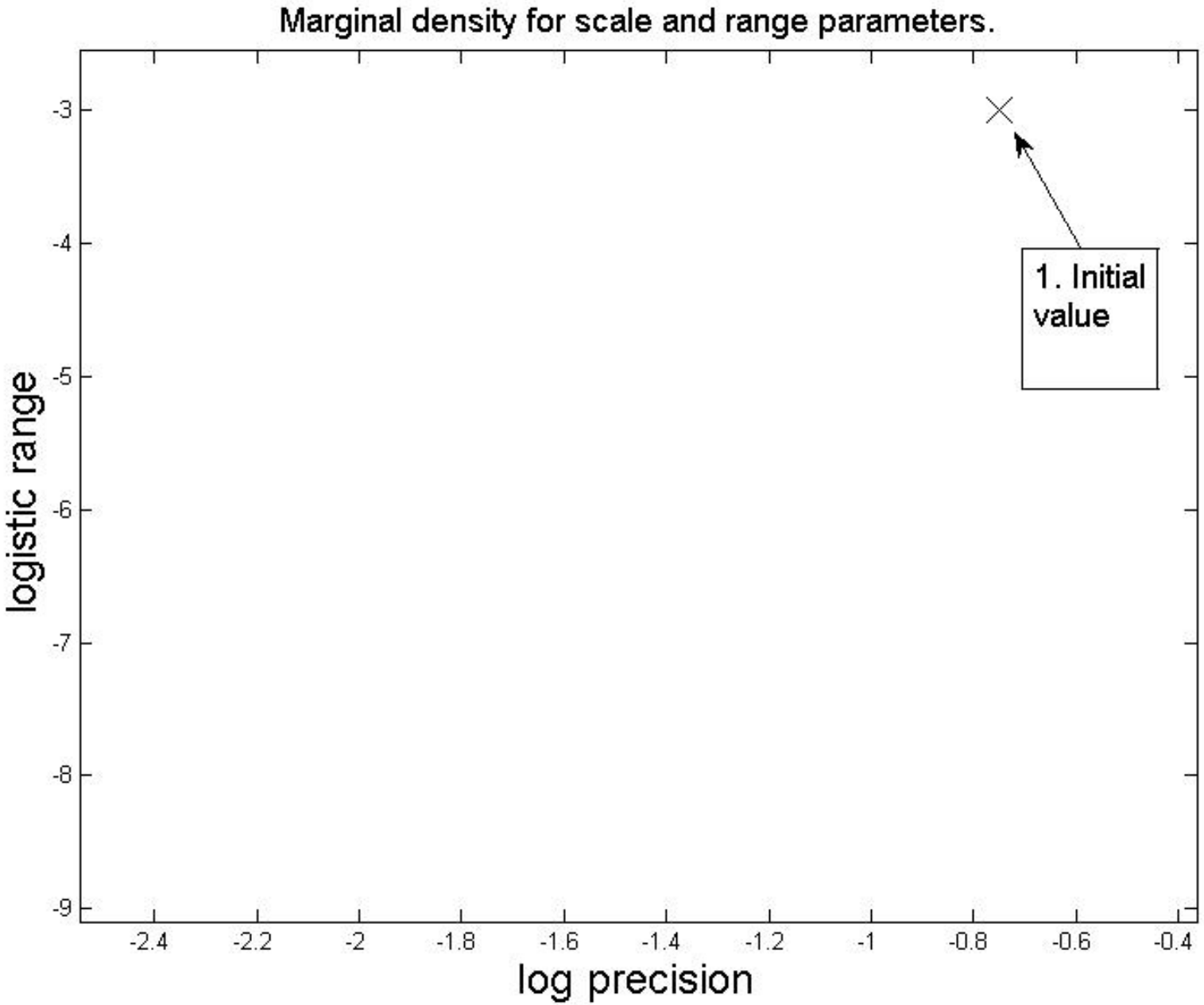
Latent field: $\boldsymbol{x} \mid \boldsymbol{\theta} \sim \pi(\boldsymbol{x} \mid \boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}(\boldsymbol{\theta}))$

Hyper-parameters: $\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta})$

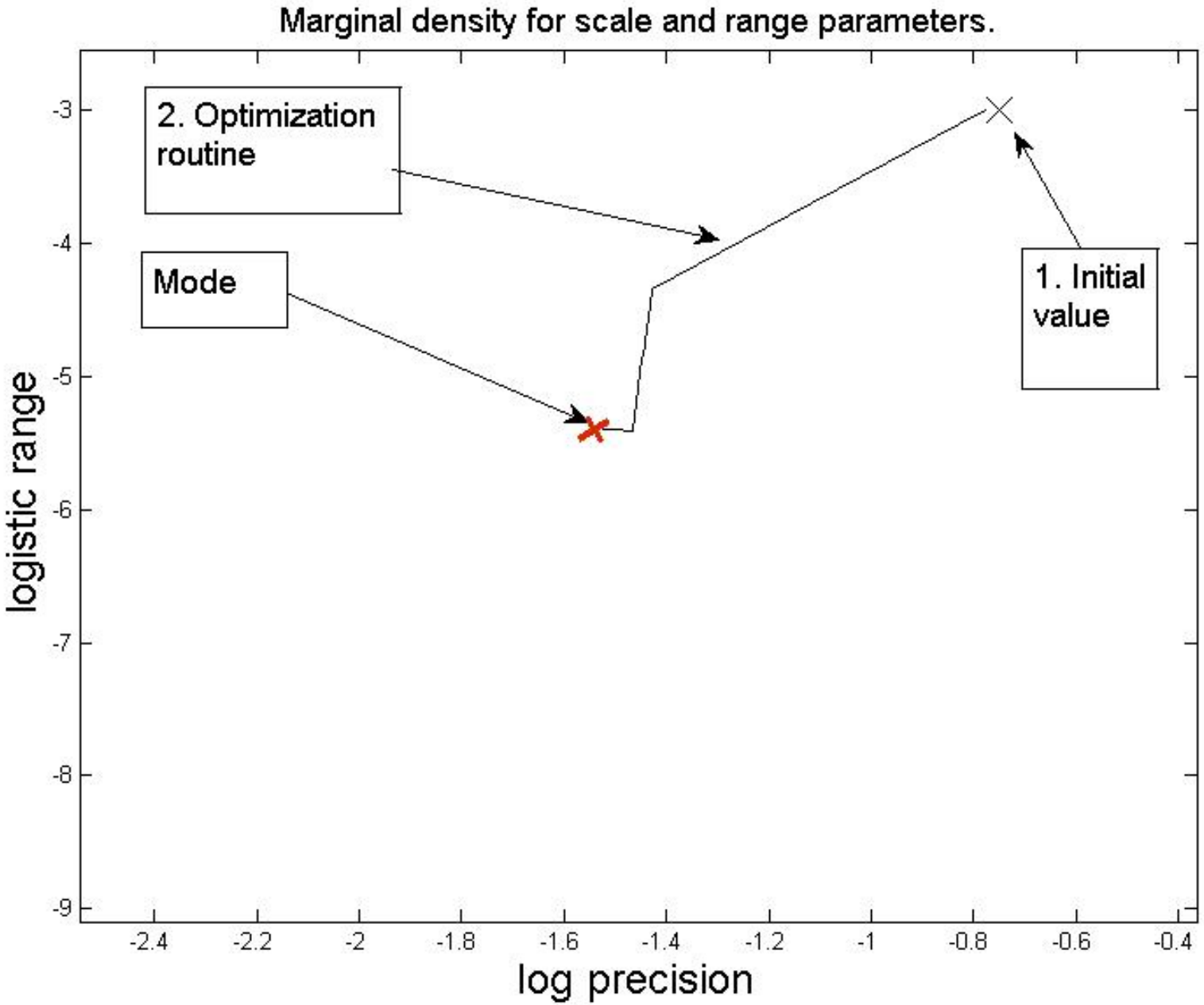
Of interest

- Marginal posterior for hyper-parameters $\pi(\theta|\boldsymbol{y})$
- Marginal posterior for latent variables $\pi(x_i|\boldsymbol{y})$

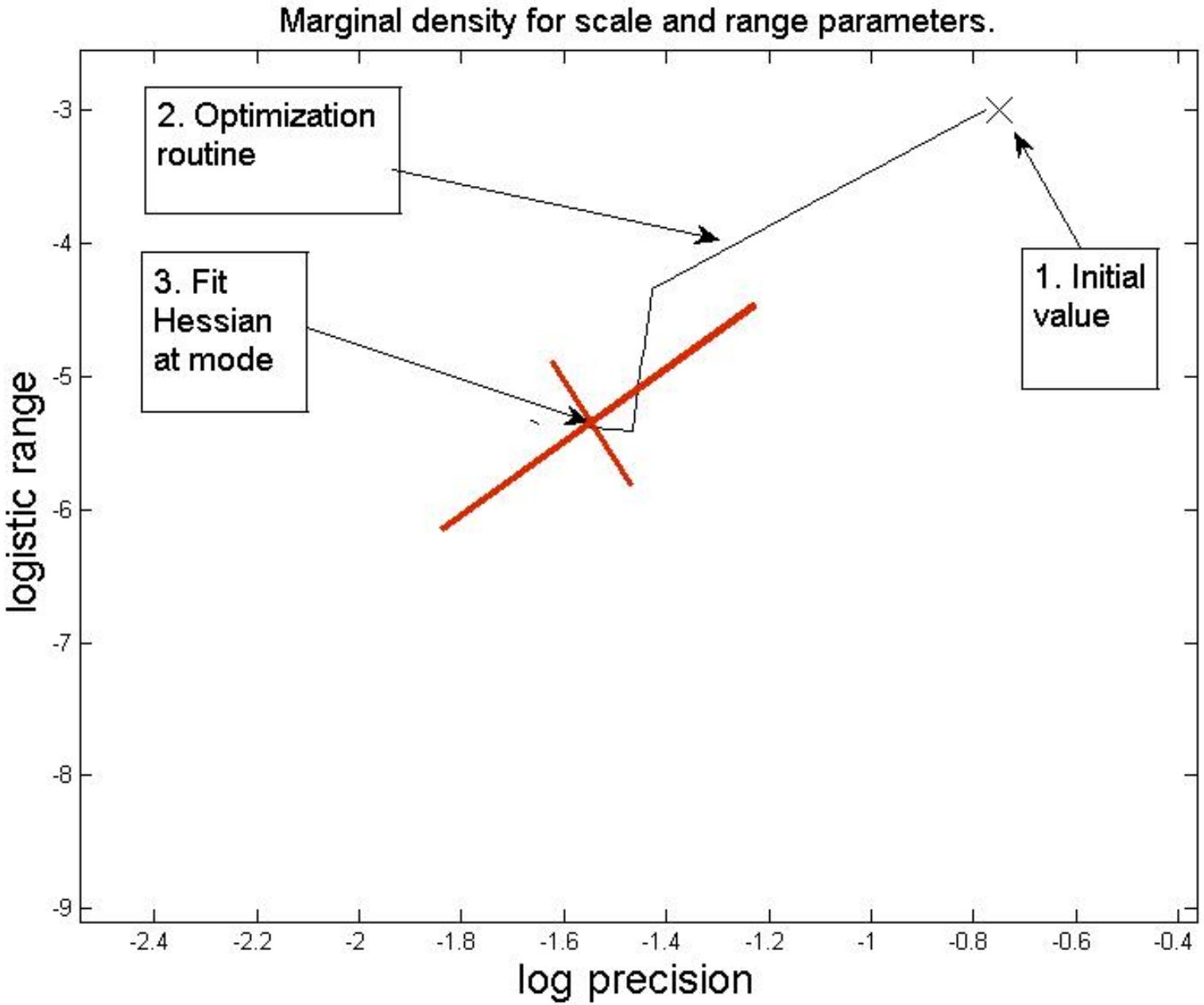
INLA, 2 hyper-parameters example



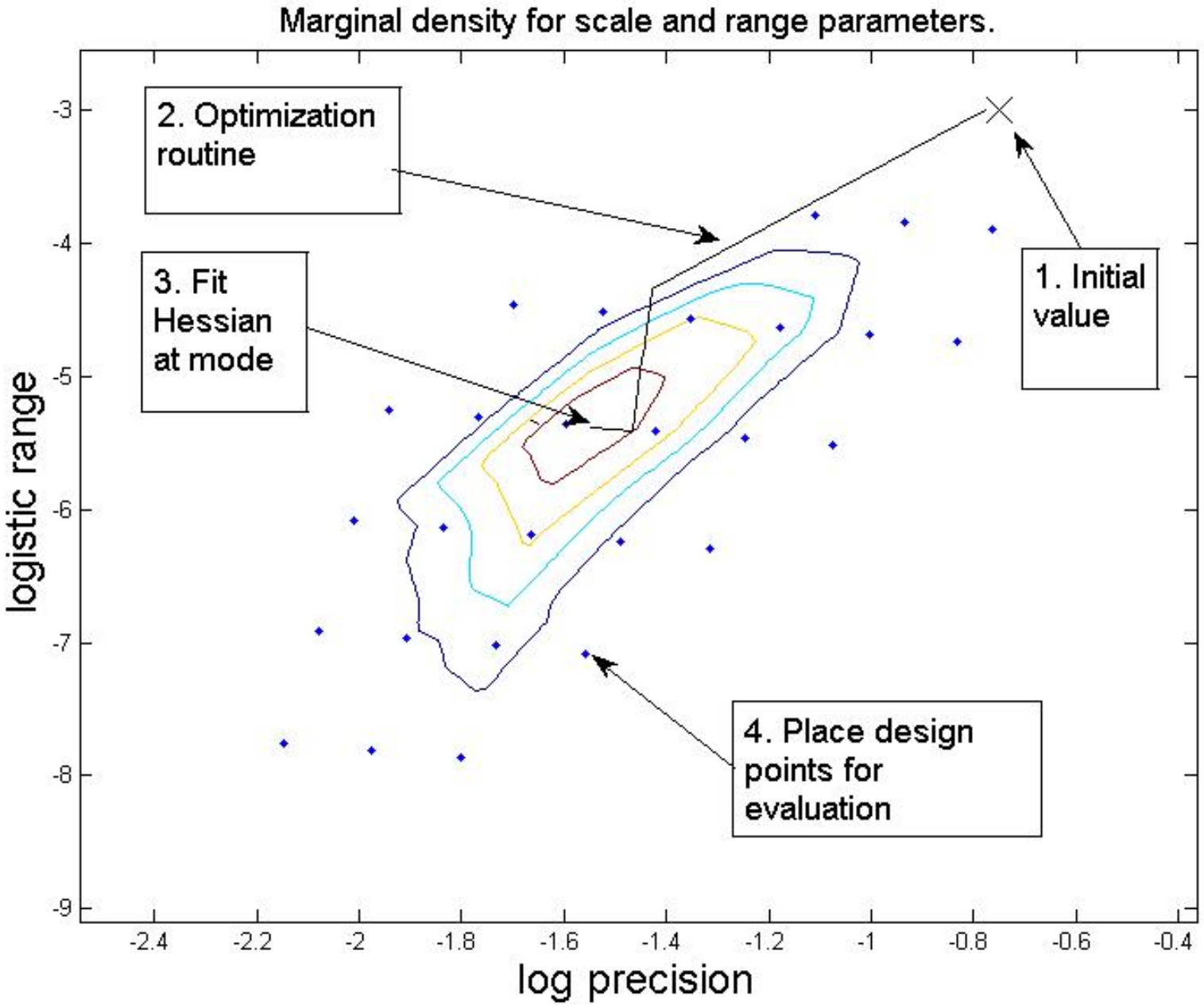
INLA, 2 hyper-parameters example



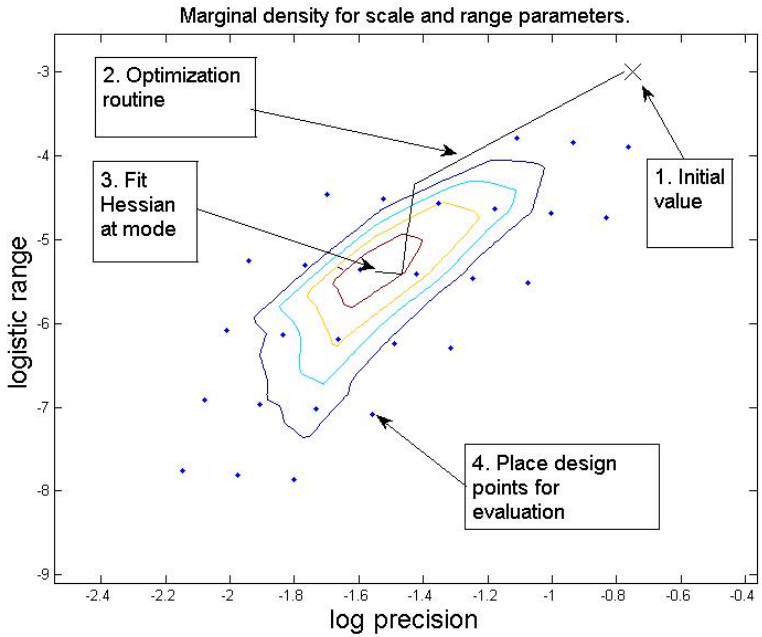
INLA, 2 hyper-parameters example



INLA, 2 hyper-parameters example



INLA, 2 hyper-parameters example



We need: $\pi(\boldsymbol{\theta}|\boldsymbol{y})$

We have: $\pi(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}) = \pi(\boldsymbol{\theta})\pi(\boldsymbol{x}|\boldsymbol{\theta})\pi(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta})$

From $\pi(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})$ to $\pi(\boldsymbol{\theta}|\boldsymbol{y})$

Consider the general problem

- θ is hyper-parameter with prior $\pi(\theta)$
- x is latent with density $\pi(x|\theta)$
- y is observed with likelihood $\pi(y|x)$

then

$$\pi(\boldsymbol{\theta}|\boldsymbol{y}) = \int \pi(\boldsymbol{\theta}, \boldsymbol{x}|\boldsymbol{y}) dx$$

From $\pi(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})$ to $\pi(\boldsymbol{\theta}|\boldsymbol{y})$

Consider the general problem

- θ is hyper-parameter with prior $\pi(\theta)$
- x is latent with density $\pi(x|\theta)$
- y is observed with likelihood $\pi(y|x)$

then

$$\pi(\boldsymbol{\theta}|\boldsymbol{y}) = \int \pi(\boldsymbol{\theta}, \boldsymbol{x}|\boldsymbol{y}) dx$$

or

$$\pi(\theta|y) = \frac{\pi(x, \theta|y)}{\pi(x|\theta, y)}$$

for any x!

$(\pi(A) = \frac{\pi(A,B)}{\pi(B|A)})$

From $\pi(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})$ to $\pi(\boldsymbol{\theta}|\boldsymbol{y})$

Further,

$$\begin{aligned}
\pi(\theta|y) &= \frac{\pi(x,\theta|y)}{\pi(x|\theta,y)} \\
&\propto \frac{\pi(\theta)\,\pi(x|\theta)\,\pi(y|x)}{\pi(x|\theta,y)} \\
&\approx \left.\frac{\pi(\theta)\,\pi(x|\theta)\,\pi(y|x)}{\pi_G(x|\theta,y)}\right|_{x=x^*(\theta)}
\end{aligned}$$

where $\pi_G(x|\theta,y)$ is the Gaussian approximation of $\pi(x|\theta,y)$ and $x^*(\theta)$ is the mode.

The GMRF-approximation

$$\pi(\boldsymbol{x} \mid \boldsymbol{y}, \theta) \propto \exp\left(-\frac{1}{2}\boldsymbol{x}^T \boldsymbol{Q}\boldsymbol{x} + \sum_i \log \pi(y_i|x_i)\right)$$
$$\approx \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T(\boldsymbol{Q} + \text{diag}(c_i))(\boldsymbol{x}-\boldsymbol{\mu})\right) = \widetilde{\pi}(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y})$$

Constructed as follows:

- Locate the mode $\boldsymbol{x}^*$
- Expand to second order

Markov and computational properties are preserved

INLA

Ideas:

1. Laplace approximation

2. Numerical integration

INLA

Ideas:

1. Laplace approximation

$$\pi(\theta|y) = \frac{\pi(\theta, x|y)}{\pi(x|y,\theta)} \approx \frac{\pi(\theta, x|y)}{\hat{\pi}(x|y,\theta)}$$

   where $\hat{\pi}(x|y,\theta)$ Gaussian approximation to $\pi(x|y,\theta)$

2. Numerical integration

INLA

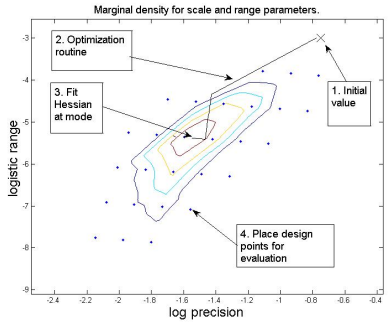
Ideas:

1. Laplace approximation

$$\pi(\theta|y) = \frac{\pi(\theta, x|y)}{\pi(x|y,\theta)} \approx \frac{\pi(\theta, x|y)}{\hat{\pi}(x|y,\theta)}$$

where $\hat{\pi}(x|y,\theta)$ Gaussian approximation to $\pi(x|y,\theta)$

2. Numerical integration



INLA

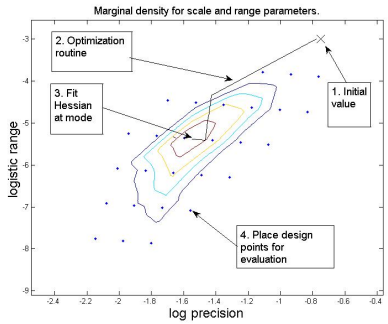
Ideas:

1. Laplace approximation

$$\pi(\theta|y) = \frac{\pi(\theta, x|y)}{\pi(x|y,\theta)} \approx \frac{\pi(\theta, x|y)}{\hat{\pi}(x|y,\theta)}$$

where $\hat{\pi}(x|y,\theta)$ Gaussian approximation to $\pi(x|y,\theta)$

2. Numerical integration



(normalizing is trivial)

INLA

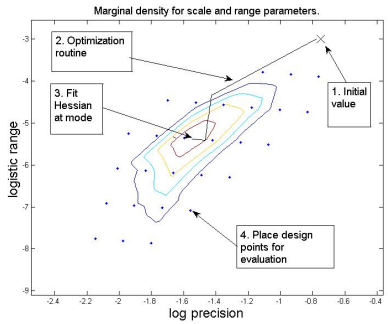
Ideas:

1. Laplace approximation

$$\pi(\theta|y) = \frac{\pi(\theta, x|y)}{\pi(x|y,\theta)} \approx \frac{\pi(\theta, x|y)}{\hat{\pi}(x|y,\theta)}$$

where $\hat{\pi}(x|y,\theta)$ Gaussian approximation to $\pi(x|y,\theta)$
$\pi(y_i|x_i)$ Gaussian $\Rightarrow \pi(x|y,\theta)$?

2. Numerical integration



(normalizing is trivial)

INLA

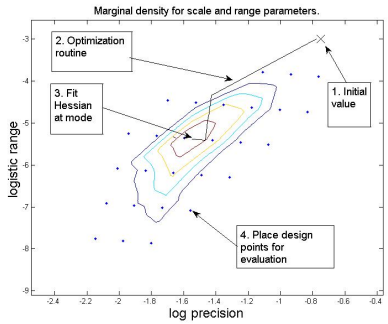
Ideas:

1. Laplace approximation

$$\pi(\theta|y) = \frac{\pi(\theta, x|y)}{\pi(x|y,\theta)} \approx \frac{\pi(\theta, x|y)}{\hat{\pi}(x|y,\theta)}$$

where $\hat{\pi}(x|y,\theta)$ Gaussian approximation to $\pi(x|y,\theta)$
$\pi(y_i|x_i)$ Gaussian $\Rightarrow$ $\pi(x|y,\theta)$ Gaussian

2. Numerical integration



(normalizing is trivial)

Important observation

If $\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}$ is *Gaussian*, the "approximation" is exact.

Important observation

If $\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}$ is *Gaussian*, the "approximation" is exact.

Today we only consider estimation for latent Gaussian models with Gaussian likelihood.

INLA-Integrated Laplace

Want to find $\pi(\theta|y)$, $\pi(x_i|y)$,....

INLA-Integrated Laplace

Want to find $\pi(\theta|y)$, $\pi(x_i|y)$,....

$$\pi(x_i|y) = \int \pi(x_i|\theta, y)\ \pi(\theta|y)\ d\theta$$

INLA-Integrated Laplace

Want to find $\pi(\theta|y)$, $\pi(x_i|y)$,....

$$\pi(x_i|y) = \int \pi(x_i|\theta, y)\ \pi(\theta|y)\ d\theta$$

- $\pi(\theta|y) \Rightarrow$ Laplace approximations

INLA-Integrated Laplace

Want to find $\pi(\theta|y)$, $\pi(x_i|y)$,....

$$\pi(x_i|y) = \int \pi(x_i|\theta, y)\ \pi(\theta|y)\ d\theta$$

- $\pi(\theta|y) \Rightarrow$ Laplace approximations
- $\int\ \mathrm{d}\theta \Rightarrow$ Numerical integration $\int\ d\theta \approx \sum_{\Theta}$

INLA-Integrated Laplace

Want to find $\pi(\theta|y)$, $\pi(x_i|y)$,....

$$\pi(x_i|y) = \int \pi(x_i|\theta, y)\ \pi(\theta|y)\ d\theta$$

- $\pi(\theta|y) \Rightarrow$ Laplace approximations
- $\int\ \mathrm{d}\theta \Rightarrow$ Numerical integration $\int\ d\theta \approx \sum_{\Theta}$
- $\pi(x_i|\theta, y) \Rightarrow$ Laplace approximations
  $\pi(x_i|\theta, y) \approx \frac{\pi(x_i, x_{-i}|\theta, y)}{\hat{\pi}(x_{-i}|\theta, y)}$

INLA-Integrated Laplace

Want to find $\pi(\theta|y)$, $\pi(x_i|y)$,....

$$\pi(x_i|y) = \int \pi(x_i|\theta, y)\ \pi(\theta|y)\ d\theta$$

- $\pi(\theta|y) \Rightarrow$ Laplace approximations
- $\int\ \mathrm{d}\theta \Rightarrow$ Numerical integration $\int\ d\theta \approx \sum_{\Theta}$
- $\pi(x_i|\theta, y) \Rightarrow$ Laplace approximations
  $\pi(x_i|\theta, y) \approx \frac{\pi(x_i, x_{-i}|\theta, y)}{\hat{\pi}(x_{-i}|\theta, y)}$ And when $\pi(x|\theta, y)$ is Gaussian....

INLA-Integrated Laplace

Want to find $\pi(\theta|y)$, $\pi(x_i|y)$,....

$$\pi(x_i|y) = \int \pi(x_i|\theta, y)\ \pi(\theta|y)\ d\theta$$

- $\pi(\theta|y) \Rightarrow$ Laplace approximations
- $\int\ \mathrm{d}\theta \Rightarrow$ Numerical integration $\int\ d\theta \approx \sum_{\Theta}$
- $\pi(x_i|\theta, y) \Rightarrow$ Laplace approximations
  $\pi(x_i|\theta, y) \approx \frac{\pi(x_i, x_{-i}|\theta, y)}{\hat{\pi}(x_{-i}|\theta, y)}$ And when $\pi(x|\theta, y)$ is Gaussian....
- $\pi(x_i|\theta, y)$ is Gaussian

INLA-Integrated Laplace

Want to find $\pi(\theta|y)$, $\pi(x_i|y)$,....

$$\pi(x_i|y)=\int \pi(x_i|\theta,y)\ \pi(\theta|y)\ d\theta$$

- $\pi(\theta|y) \Rightarrow$ Laplace approximations
- $\int\ \mathrm{d}\theta \Rightarrow$ Numerical integration $\int d\theta \approx \sum_{\Theta}$
- $\pi(x_i|\theta,y) \Rightarrow$ Laplace approximations
  $\pi(x_i|\theta,y) \approx \frac{\pi(x_i,x_{-i}|\theta,y)}{\hat{\pi}(x_{-i}|\theta,y)}$ And when $\pi(x|\theta,y)$ is Gaussian....
- $\pi(x_i|\theta,y)$ is Gaussian

$$\pi(\boldsymbol{x}|\theta,y) \sim N(\mu, Q^{-1})$$

We know $\boldsymbol{\mu}$ and Q.

$$x_i \sim N(?,?)$$

INLA-Integrated Laplace

Want to find $\pi(\theta|y)$, $\pi(x_i|y)$,....

$$\pi(x_i|y) = \int \pi(x_i|\theta, y)\ \pi(\theta|y)\ d\theta$$

- $\pi(\theta|y) \Rightarrow$ Laplace approximations
- $\int\ \mathrm{d}\theta \Rightarrow$ Numerical integration $\int d\theta \approx \sum_{\Theta}$
- $\pi(x_i|\theta, y) \Rightarrow$ Laplace approximations
  $\pi(x_i|\theta, y) \approx \frac{\pi(x_i, x_{-i}|\theta, y)}{\hat{\pi}(x_{-i}|\theta, y)}$ And when $\pi(x|\theta, y)$ is Gaussian....
- $\pi(x_i|\theta, y)$ is Gaussian

$$\pi(\boldsymbol{x}|\theta, y) \sim N(\mu, Q^{-1})$$

We know $\boldsymbol{\mu}$ and Q.

$$x_i \sim N(\mu_i, ?)$$

Model requirements

Hierarchical model:

1. Data: $y \sim \pi(y|x, \theta)$

2. Latent field $\pi(\boldsymbol{x}|\theta)$
3. Hyper-parameters: $\theta \sim \pi(\theta)$

Model requirements

Hierarchical model: INLA requires

1. Data: $y \sim \pi(y|x,\theta) = \prod_{i=1}^{n} \pi(y_i|x_i)$

2. Latent field $\pi(\boldsymbol{x}|\theta)$ $x \sim N(\mu, \Sigma)$
3. Hyper-parameters: $\theta \sim \pi(\theta)$ only few non-Gaussian

Model requirements

Hierarchical model: INLA requires

1. Data: $y \sim \pi(y|x,\theta) = \prod_{i=1}^{n} \pi(y_i|x_i)$

2. Latent field $\pi(\boldsymbol{x}|\theta)$ $x \sim N(\mu, \Sigma)$ GMRF
3. Hyper-parameters: $\theta \sim \pi(\theta)$ only few non-Gaussian

Model requirements

Hierarchical model: INLA requires

1. Data: $y \sim \pi(y|x,\theta) = \prod_{i=1}^{n} \pi(y_i|x_i)$ can be relaxed: $\pi(y|x,\theta) = \prod_{i=1}^{n} \pi(y_i|\boldsymbol{Ax},\theta)$ with a sparse A.
2. Latent field $\pi(\boldsymbol{x}|\theta)$ $x \sim N(\mu,\Sigma)$ GMRF
3. Hyper-parameters: $\theta \sim \pi(\theta)$ only few non-Gaussian

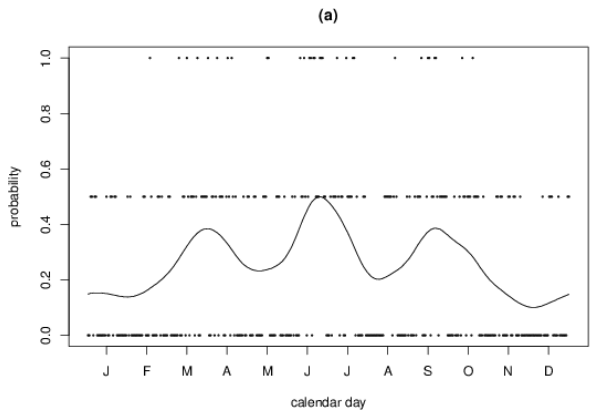
Model requirements

Hierarchical model: INLA requires

1. Data: $y \sim \pi(y|x,\theta) = \prod_{i=1}^{n} \pi(y_i|x_i)$ can be relaxed: $\pi(y|x,\theta) = \prod_{i=1}^{n} \pi(y_i|\boldsymbol{Ax},\theta)$ with a sparse A.
2. Latent field $\pi(\boldsymbol{x}|\theta)$ $x \sim N(\mu,\Sigma)$ GMRF
3. Hyper-parameters: $\theta \sim \pi(\theta)$ only few non-Gaussian

We call these models *Latent GMRF models*

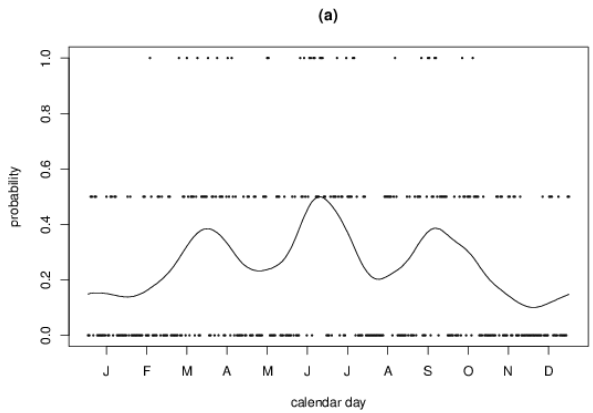
Example(1): Tokyo rainfall data



Stage 1 Binomial data

$$y_i \sim \begin{cases} \text{Binomial}(2, p(x_i)) \\ \text{Binomial}(1, p(x_i)) \end{cases}$$

Example(1): Tokyo rainfall data



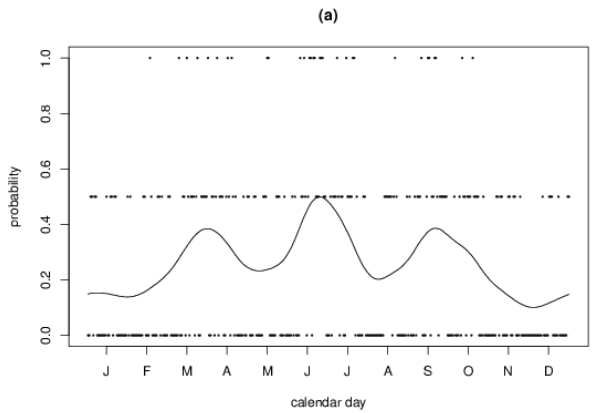
Stage 1 Binomial data

$$y_i \sim \begin{cases} \text{Binomial}(2, p(x_i)) \\ \text{Binomial}(1, p(x_i)) \end{cases}$$

Stage 2 Assume a smooth latent $\boldsymbol{x}$,

$$\boldsymbol{x} \sim RW2(\kappa), \qquad \text{logit}(p_i) = x_i$$

Example(1): Tokyo rainfall data



Stage 1 Binomial data

$$y_i \sim \begin{cases} \text{Binomial}(2, p(x_i)) \\ \text{Binomial}(1, p(x_i)) \end{cases}$$

Stage 2 Assume a smooth latent $\boldsymbol{x}$,

$$\boldsymbol{x} \sim RW2(\kappa), \qquad \text{logit}(p_i) = x_i$$

Stage 3 Gamma(α, β)-prior on κ

Key ideas

Want to find $\pi(\theta|y)$, $\pi(x_i|y)$,....

Key ideas

Want to find $\pi(\theta|y)$, $\pi(x_i|y)$,....

1. Laplace approximation

2. Numerical integration

Key ideas

Want to find $\pi(\theta|y)$, $\pi(x_i|y)$,....

1. Laplace approximation

$$\pi(\theta|y) = \frac{\pi(\theta, x|y)}{\pi(x|y, \theta)} \approx \frac{\pi(\theta, x|y)}{\hat{\pi}(x|y, \theta)}$$

where $\hat{\pi}(x|y, \theta)$ Gaussian approximation to $\pi(x|y, \theta)$

2. Numerical integration

Key ideas

Want to find $\pi(\theta|y)$, $\pi(x_i|y)$,....

1. Laplace approximation

$$\pi(\theta|y) = \frac{\pi(\theta, x|y)}{\pi(x|y, \theta)} \approx \frac{\pi(\theta, x|y)}{\hat{\pi}(x|y, \theta)}$$

   where $\hat{\pi}(x|y, \theta)$ Gaussian approximation to $\pi(x|y, \theta)$
   $\pi(y_i|x_i)$ Gaussian $\Rightarrow$ $\pi(x|y, \theta)$ Gaussian
2. Numerical integration

Key ideas

Want to find $\pi(\theta|y)$, $\pi(x_i|y)$,....

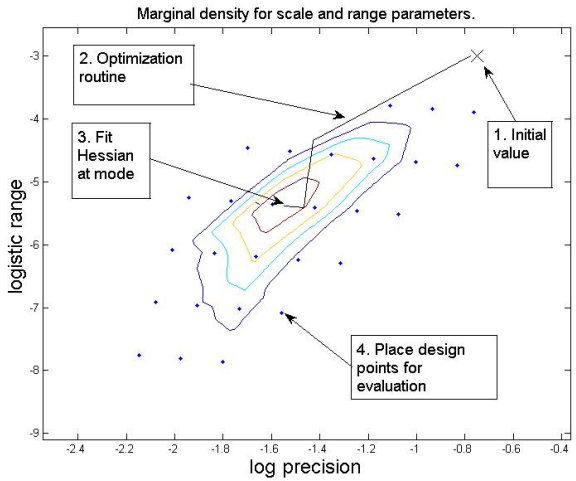
1. Laplace approximation

$$\pi(\theta|y) = \frac{\pi(\theta, x|y)}{\pi(x|y, \theta)} \approx \frac{\pi(\theta, x|y)}{\hat{\pi}(x|y, \theta)}$$

where $\hat{\pi}(x|y, \theta)$ Gaussian approximation to $\pi(x|y, \theta)$

$\pi(y_i|x_i)$ Gaussian $\Rightarrow$ $\pi(x|y, \theta)$ Gaussian

2. Numerical integration



Key ideas

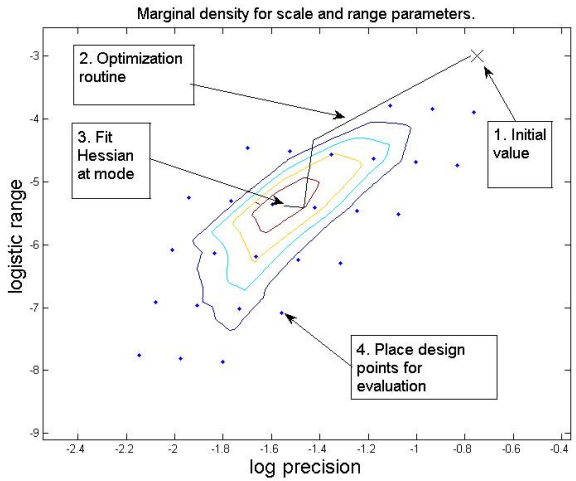
Want to find $\pi(\theta|y)$, $\pi(x_i|y)$,....

1. Laplace approximation

$$\pi(\theta|y) = \frac{\pi(\theta, x|y)}{\pi(x|y, \theta)} \approx \frac{\pi(\theta, x|y)}{\hat{\pi}(x|y, \theta)}$$

where $\hat{\pi}(x|y, \theta)$ Gaussian approximation to $\pi(x|y, \theta)$
$\pi(y_i|x_i)$ Gaussian $\Rightarrow$ $\pi(x|y, \theta)$ Gaussian

2. Numerical integration



Utilize the sparse Q that a GMRF model gives.

INLA-Integrated Laplace

Want to find $\pi(\theta|y)$, $\pi(x_i|y)$,....

INLA-Integrated Laplace

Want to find $\pi(\theta|y)$, $\pi(x_i|y)$,....

$$\pi(x_i|y) = \int \pi(x_i|\theta, y)\ \pi(\theta|y)\ d\theta$$

INLA-Integrated Laplace

Want to find $\pi(\theta|y)$, $\pi(x_i|y)$,....

$$\pi(x_i|y) = \int \pi(x_i|\theta, y)\ \pi(\theta|y)\ d\theta$$

- $\pi(\theta|y) \Rightarrow$ Laplace approximations

INLA-Integrated Laplace

Want to find $\pi(\theta|y)$, $\pi(x_i|y)$,....

$$\pi(x_i|y) = \int \pi(x_i|\theta, y)\ \pi(\theta|y)\ d\theta$$

- $\pi(\theta|y) \Rightarrow$ Laplace approximations
- $\int\ \mathrm{d}\theta \Rightarrow$ Numerical integration $\int\ d\theta \approx \sum_{\Theta}$

INLA-Integrated Laplace

Want to find $\pi(\theta|y)$, $\pi(x_i|y)$,....

$$\pi(x_i|y) = \int \pi(x_i|\theta, y)\ \pi(\theta|y)\ d\theta$$

- $\pi(\theta|y) \Rightarrow$ Laplace approximations
- $\int\ \mathrm{d}\theta \Rightarrow$ Numerical integration $\int\ d\theta \approx \sum_{\Theta}$
- $\pi(x_i|\theta, y) \Rightarrow$ Laplace approximations
  $\pi(x_i|\theta, y) \approx \frac{\pi(x_i, x_{-i}|\theta, y)}{\hat{\pi}(x_{-i}|\theta, y)}$

INLA-Integrated Laplace

Want to find $\pi(\theta|y)$, $\pi(x_i|y)$,....

$$\pi(x_i|y) = \int \pi(x_i|\theta, y)\ \pi(\theta|y)\ d\theta$$

- $\pi(\theta|y) \Rightarrow$ Laplace approximations
- $\int\ \mathrm{d}\theta \Rightarrow$ Numerical integration $\int\ d\theta \approx \sum_{\Theta}$
- $\pi(x_i|\theta, y) \Rightarrow$ Laplace approximations
  $\pi(x_i|\theta, y) \approx \frac{\pi(x_i, x_{-i}|\theta, y)}{\hat{\pi}(x_{-i}|\theta, y)}$ And when $\pi(x|\theta, y)$ is Gaussian....

INLA-Integrated Laplace

Want to find $\pi(\theta|y)$, $\pi(x_i|y)$,....

$$\pi(x_i|y) = \int \pi(x_i|\theta, y)\ \pi(\theta|y)\ d\theta$$

- $\pi(\theta|y) \Rightarrow$ Laplace approximations
- $\int \mathrm{d}\theta \Rightarrow$ Numerical integration $\int d\theta \approx \sum_{\Theta}$
- $\pi(x_i|\theta, y) \Rightarrow$ Laplace approximations
  $\pi(x_i|\theta, y) \approx \frac{\pi(x_i, x_{-i}|\theta, y)}{\hat{\pi}(x_{-i}|\theta, y)}$ And when $\pi(x|\theta, y)$ is Gaussian....
- $\pi(x_i|\theta, y)$ is Gaussian

INLA-Integrated Laplace

Want to find $\pi(\theta|y)$, $\pi(x_i|y)$,....

$$\pi(x_i|y) = \int \pi(x_i|\theta, y)\ \pi(\theta|y)\ d\theta$$

- $\pi(\theta|y) \Rightarrow$ Laplace approximations
- $\int\ \mathrm{d}\theta \Rightarrow$ Numerical integration $\int d\theta \approx \sum_{\Theta}$
- $\pi(x_i|\theta, y) \Rightarrow$ Laplace approximations
  $\pi(x_i|\theta, y) \approx \frac{\pi(x_i, x_{-i}|\theta, y)}{\hat{\pi}(x_{-i}|\theta, y)}$ And when $\pi(x|\theta, y)$ is Gaussian....
- $\pi(x_i|\theta, y)$ is Gaussian

  And if $\pi(x|\theta, y)$ is not Gaussian...?

Approximating $\pi(x_i|\boldsymbol{y}, \boldsymbol{\theta})$

This task is more challenging, since

- dimension of $\boldsymbol{x}$, n is large
- and there are potential n marginals to compute, or at least $\mathcal{O}(n)$.

An obvious simple and fast alternative, is to use the GMRF-approximation

$$\widetilde{\pi}(x_i|\boldsymbol{\theta}, \boldsymbol{y}) = \mathcal{N}(x_i;\ \mu(\boldsymbol{\theta}), \sigma^2(\boldsymbol{\theta}))$$

Laplace approximation of $\pi(x_i|\boldsymbol{\theta}, \boldsymbol{y})$

- The Laplace approximation:

$$\widetilde{\pi}(x_i \mid \boldsymbol{y}, \boldsymbol{\theta}) \approx \left. \frac{\pi(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y})}{\widetilde{\pi}(\boldsymbol{x}_{-i}|x_i, \boldsymbol{y}, \boldsymbol{\theta})} \right|_{\boldsymbol{x}_{-i}=\boldsymbol{x}^*_{-i}(x_i,\boldsymbol{\theta})}$$

- Again, approximation is very good, as $\boldsymbol{x}_{-i}|\boldsymbol{x}_i, \theta$ is 'almost Gaussian',
- but it is expensive. In order to get the n marginals:
  - perform n optimisations, and
  - n factorisations of $n-1 \times n-1$ matrices.

  Can be "solved".

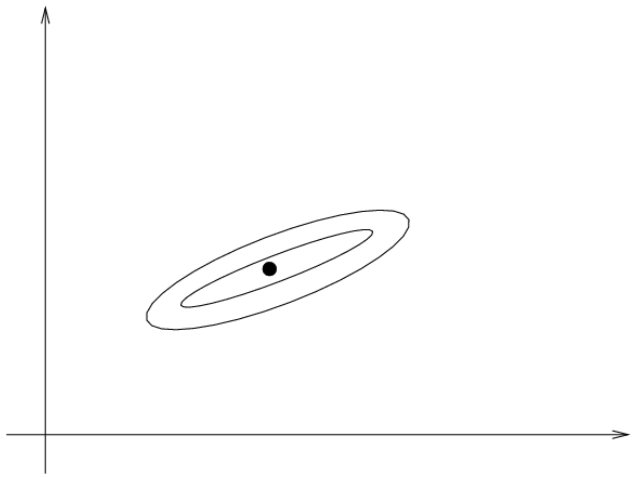
The integrated nested Laplace approximation (INLA) I

Step I Explore $\widetilde{\pi}(\boldsymbol{\theta}|\boldsymbol{y})$

The integrated nested Laplace approximation (INLA) I

Step I Explore $\widetilde{\pi}(\boldsymbol{\theta}|\boldsymbol{y})$

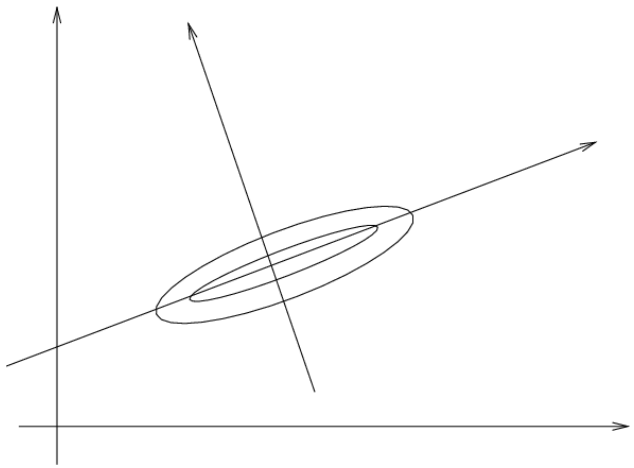
- Locate the mode



The integrated nested Laplace approximation (INLA) I

Step I Explore $\widetilde{\pi}(\boldsymbol{\theta}|\boldsymbol{y})$

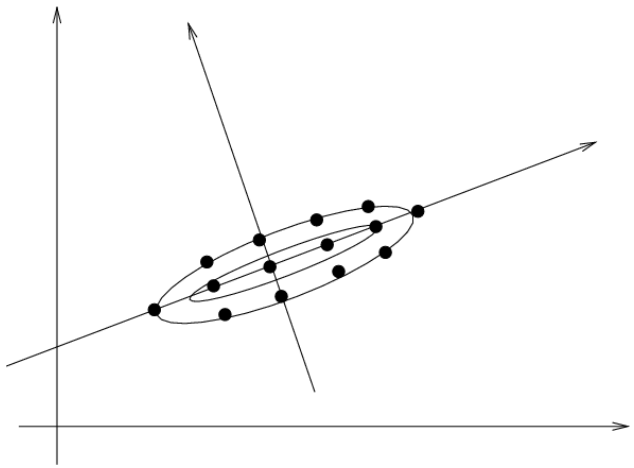
- Locate the mode
- Use the Hessian to construct new variables



The integrated nested Laplace approximation (INLA) I

Step I Explore $\widetilde{\pi}(\boldsymbol{\theta}|\boldsymbol{y})$

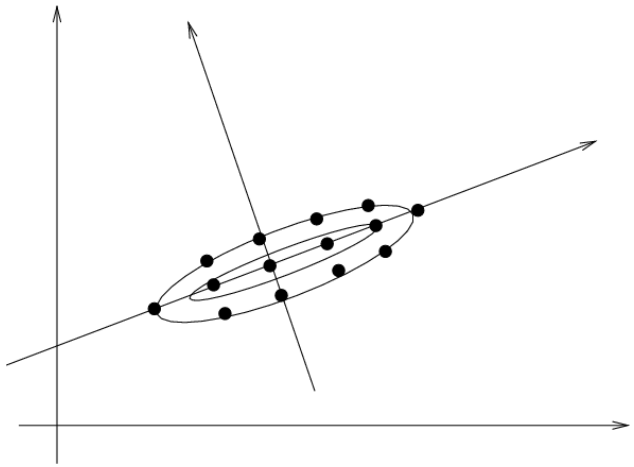
- Locate the mode
- Use the Hessian to construct new variables
- Grid-search



The integrated nested Laplace approximation (INLA) I

Step I Explore $\widetilde{\pi}(\boldsymbol{\theta}|\boldsymbol{y})$

- Locate the mode
- Use the Hessian to construct new variables
- Grid-search
- Can be case-specific



INLA II

Step II For each $\boldsymbol{\theta}_j$

- For each i, evaluate the Laplace approximation for selected values of x_i
- Build a Skew-Normal or log-spline corrected Gaussian

$$\mathcal{N}(x_i;\ \mu_i, \sigma_i^2) \times \exp(\text{spline})$$

to represent the conditional marginal density.

The integrated nested Laplace approximation (INLA) III

Step III Sum out $\boldsymbol{\theta}_j$

- For each i, sum out $\boldsymbol{\theta}$

$$\widetilde{\pi}(x_i \mid \boldsymbol{y}) \propto \sum_j \widetilde{\pi}(x_i \mid \boldsymbol{y}, \boldsymbol{\theta}_j) \times \widetilde{\pi}(\boldsymbol{\theta}_j \mid \boldsymbol{y})$$

- Build a log-spline corrected Gaussian

$$\mathcal{N}(x_i;\ \mu_i, \sigma_i^2) \times \exp(\text{spline})$$

to represent $\widetilde{\pi}(x_i \mid \boldsymbol{y})$.

Outline

Today:
1. Simulations and evaluations of GMRFs
2. Latent Gaussian Models
3. INLA for Gaussian likelihoods

Tomorrow:
1. Review key assumptions and ideas
2. INLA for non-Gaussian likelihoods
3. r-inla, with toturial

Go to `http://www.math.ntnu.no/~ingelins/Heidelberg/Tutorial/` and to `www.r-inla.org`

Some history

In the beginning there was

GMRFLib

A C library for fast computations for GMRFs.

Some history

GMRFLib begot

INLA

A C library for fast approximate inference,
accessesed through .ini files.

Some history

After much wailing and gnashing of teeth there came

R-INLA

which takes `R` code and writes an appropriate .ini file for the INLA C–program to read. (This is why INLA is not on CRAN)

The structure of an R program using INLA

There are essentially three parts to an INLA program:

1. The data organisation (important! will not be spoken of again)
2. The *formula*—notation inherrited from R's native `glm` function
3. The call to the INLA program.

The inla function

```
>  result <- inla(
        formula,   #This describes your latent field
        family = "gaussian", #The likelihood distribution.
        data = dat #A list or dataframe
        #This is all that's needed for a basic call

       verbose = TRUE, # I use this a lot!
        keep = FALSE, #Keeps the output

        #Then there are some "control statements"
        #that allow you to customise some things
        control.predictor=list(A = ObservationMatrix)
          )
```

Likelihood functions

- "binomial"
- "coxph"
- "Exponential"
- "gaussian" item "gev"
- "laplace"
- "sn" (Skew Normal)
- "stochvol", "stochvol.nig", "stochvol.t"
- "T"
- "weibull"
- Many others: go to `http://r-inla.org/`

`formula`: Specifying the latent field

The latent field is specifeid using the "standard" `R` method `formula = y ~ 1 + covariate + f(...)`.

- y is the name of your data in the data frame.

`formula`: Specifying the latent field

The latent field is specifeid using the "standard" `R` method
`formula = y ~ 1 + covariate + f(...)`.

- y is the name of your data in the data frame.
- An intercept is fitted *automatically*! Use `-1` in your formula to avoid it.

`formula`: Specifying the latent field

The latent field is specifeid using the "standard" `R` method
`formula = y ~ 1 + covariate + f(...)`.

- y is the name of your data in the data frame.
- An intercept is fitted *automatically*! Use `-1` in your formula to avoid it.
- The fixed effects (covariates) are taken as i.i.d. normal with a common prior. (This can be changed)

`formula`: Specifying the latent field

The latent field is specifeid using the "standard" `R` method
`formula = y ~ 1 + covariate + f(...)`.

- *y* is the name of your data in the data frame.
- An intercept is fitted *automatically*! Use `-1` in your formula to avoid it.
- The fixed effects (covariates) are taken as i.i.d. normal with a common prior. (This can be changed)
- The `f` function contains the random effect specifications.

The simplest case: Linear regression, `linear-regression.R`

Specifying random effects

Random effects are added to the formula through the function

```
f(name, model="...", hyper = ...,
        replicate = ...,  constr =FALSE, cyclic = FALSE)
```

- `name`—the name of the random effect. Also refers to the values in data which are used for various things.

Specifying random effects

Random effects are added to the formula through the function

```
f(name, model="...", hyper = ...,
        replicate = ...,  constr =FALSE, cyclic = FALSE)
```

- `name`—the name of the random effect. Also refers to the values in data which are used for various things.
- `model`—the latent model. Eg. "iid", "rw2", "ar1", "bym" etc

Specifying random effects

Random effects are added to the formula through the function

```
f(name, model="...", hyper = ...,
        replicate = ...,  constr =FALSE, cyclic = FALSE)
```

- `name`—the name of the random effect. Also refers to the values in data which are used for various things.
- `model`—the latent model. Eg. "iid", "rw2", "ar1", "bym" etc
- `hyper`—specify the prior on the hyperparameters

Specifying random effects

Random effects are added to the formula through the function

```
f(name, model="...", hyper = ...,
        replicate = ...,  constr =FALSE, cyclic = FALSE)
```

- `name`—the name of the random effect. Also refers to the values in data which are used for various things.
- `model`—the latent model. Eg. "iid", "rw2", "ar1", "bym" etc
- `hyper`—specify the prior on the hyperparameters
- `replicate`—for replicates (what else :p)

Specifying random effects

Random effects are added to the formula through the function

```
f(name, model="...", hyper = ...,
        replicate = ...,  constr =FALSE, cyclic = FALSE)
```

- `name`—the name of the random effect. Also refers to the values in data which are used for various things.
- `model`—the latent model. Eg. "iid", "rw2", "ar1", "bym" etc
- `hyper`—specify the prior on the hyperparameters
- `replicate`—for replicates (what else :p)
- `constr`—Sum to zero constraint?

Specifying random effects

Random effects are added to the formula through the function

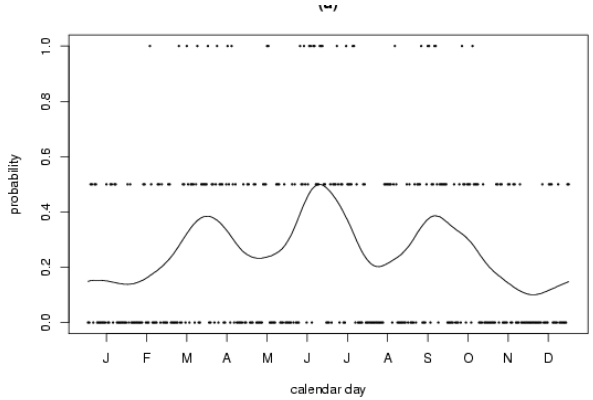
```
f(name, model="...", hyper = ...,
        replicate = ...,  constr =FALSE, cyclic = FALSE)
```

- `name`—the name of the random effect. Also refers to the values in data which are used for various things.
- `model`—the latent model. Eg. "iid", "rw2", "ar1", "bym" etc
- `hyper`—specify the prior on the hyperparameters
- `replicate`—for replicates (what else :p)
- `constr`—Sum to zero constraint?
- `cyclic`—Are you cyclic? (RW1, RW2 and AR1)

Smoothing binary times series, Tokyo rainfall data

Number of days in Tokyo with rainfall above 1 mm in 1983-84.
We want to estimate the probability of rain p_t for calendar day $t = 1, \ldots, 366$

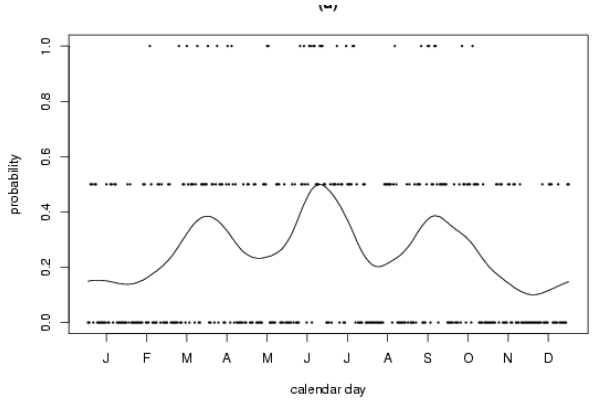
Model: Tokyo rainfall data



Stage 1 Binomial data

$$y_i \sim \begin{cases} \text{Binomial}(2, p(x_i)) \\ \text{Binomial}(1, p(x_i)) \end{cases}$$

Model: Tokyo rainfall data



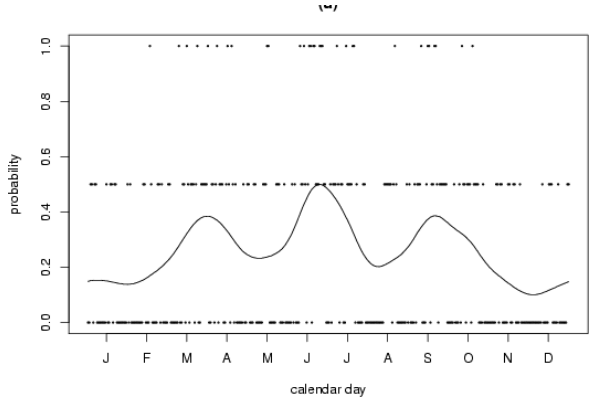
Stage 1 Binomial data

$$y_i \sim \begin{cases} \text{Binomial}(2, p(x_i)) \\ \text{Binomial}(1, p(x_i)) \end{cases}$$

Stage 2 Assume a smooth latent $\boldsymbol{x}$,

$$\boldsymbol{x} \sim RW2(\kappa), \qquad \text{logit}(p_i) = x_i$$

Model: Tokyo rainfall data



Stage 1 Binomial data

$$y_i \sim \begin{cases} \text{Binomial}(2, p(x_i)) \\ \text{Binomial}(1, p(x_i)) \end{cases}$$

Stage 2 Assume a smooth latent $\boldsymbol{x}$,

$$\boldsymbol{x} \sim RW2(\kappa), \qquad \text{logit}(p_i) = x_i$$

Stage 3 Gamma(α, β)-prior on κ

Smoothing binary time series, `Tokyo.R`

The `Tokyo` data frame:

y	n	time
0	2	1
0	2	2
1	2	3
⋮		

Smoothing binary time series, Tokyo.R

The Tokyo data frame:

y	n	time
0	2	1
0	2	2
1	2	3
⋮		

Specifying the model:

```
formula = y ~ f(time, model="rw2", cyclic=TRUE)-1
```

Smoothing binary time series, Tokyo.R

The Tokyo data frame:

y	n	time
0	2	1
0	2	2
1	2	3
⋮		

Specifying the model:

```
formula = y ~ f(time, model="rw2", cyclic=TRUE)-1
```

Running inla

```
result = inla(formula,family="binomial", Ntrials=Tokyo$n,
data=Tokyo)
```

Using different link functions

INLA typically implements the canonical link functions, which, in this case, is the logit link. Sometimes, you want other things.

```
control.data = list(link = "logit")
control.data = list(link = "probit")
control.data = list(link = "cloglog")
```

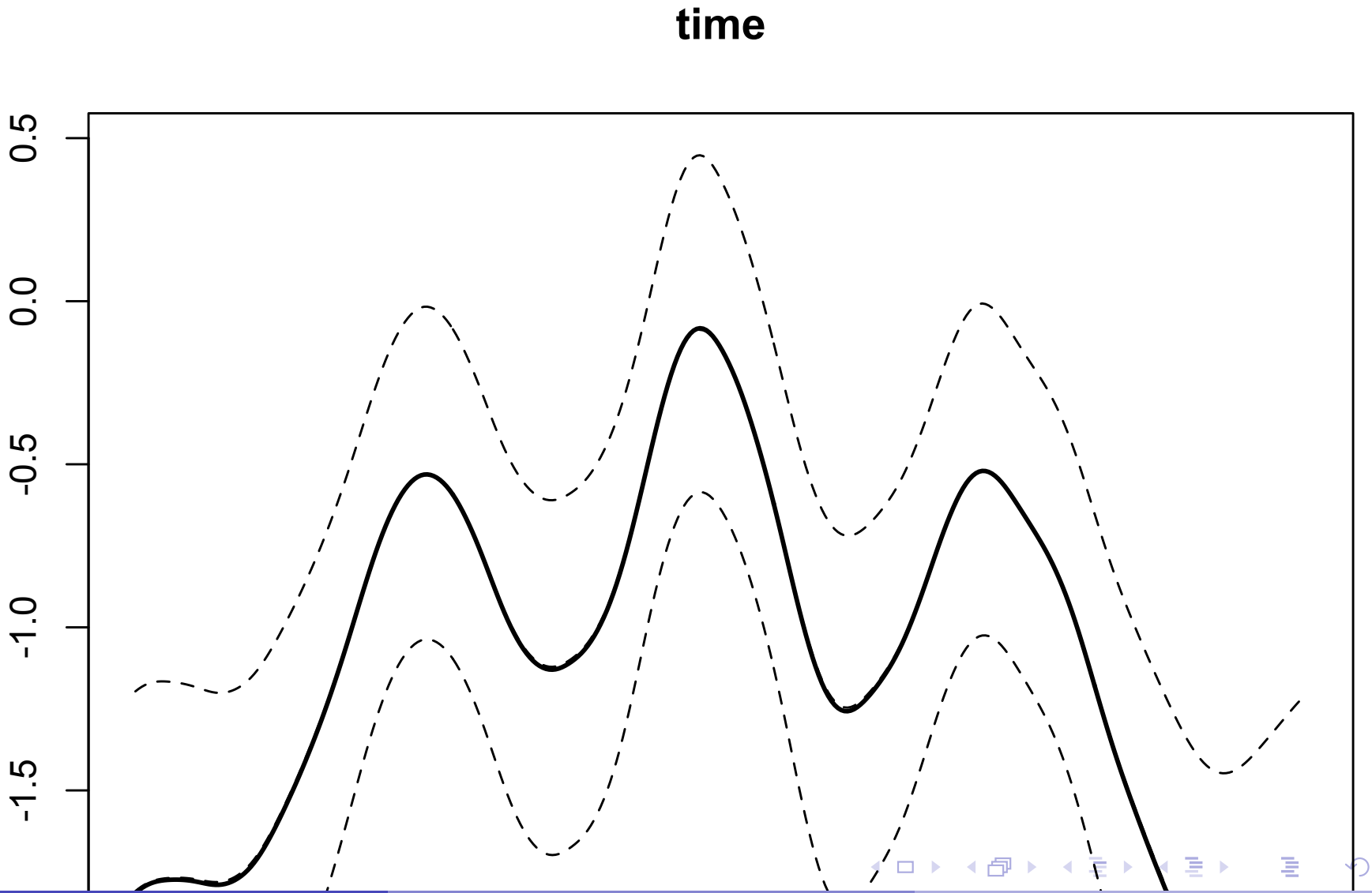
Tokyo rainfall

```
require(INLA)
data(Tokyo)

## Define the model
formula = y ~ f(time, model="rw2", cyclic=TRUE, param=c(1,0.0001)) - 1

## Once more: the call to inla in verbose mode
result = inla(formula, family="binomial", Ntrials=Tokyo$n, data=Tokyo, verbose = TRUE)

## Summarise the results
summary(result)

## Plot the results
plot(result)

## Improve estimate of the hyperparameters
h = inla.hyperpar(result)

## Summarise improved estimates
summary(h)

## Plot improved estimates
plot(h)
```

Tokyo rainfall

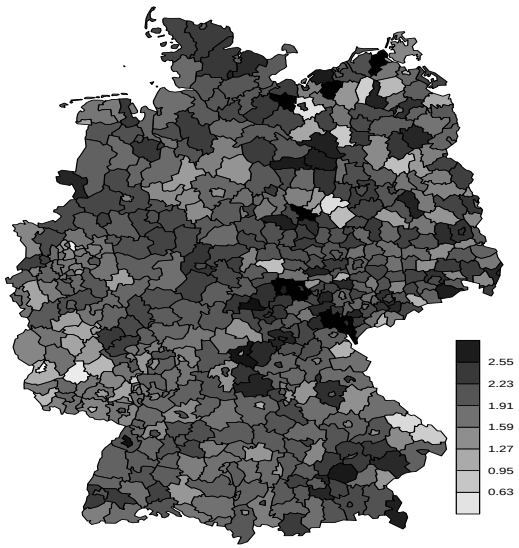
```
> summary(h)

Call:
c("inla(formula = formula, family = \"binomial\", data = Tokyo, Ntrials = Tokyo$n, ",  "    verbose = TRUE)")

Time used:
 Pre-processing    Running inla Post-processing           Total
      0.1774325       0.6074882       0.3358390       1.1207597

The model has no fixed effects

Random effects:
Name   Model    Max KLD
time   RW2 model

Model hyperparameters:
                    mean     sd       0.025quant 0.5quant 0.975quant
Precision for time 19640.63 14983.25  3823.52    15375.70 60561.84

Expected number of effective parameters(std dev): 9.106(1.289)
Number of equivalent replicates : 40.20

Marginal Likelihood:  -331.18
Warning: Interpret the marginal likelihood with care if the prior model is improper.
```

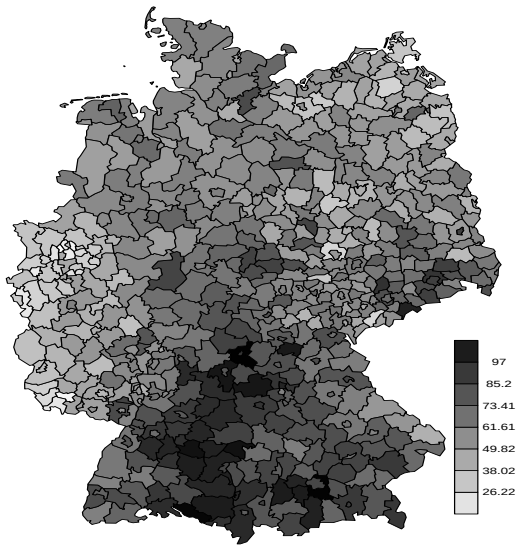
Posterior for temporal effect



Disease mapping in Germany, `bym.R`

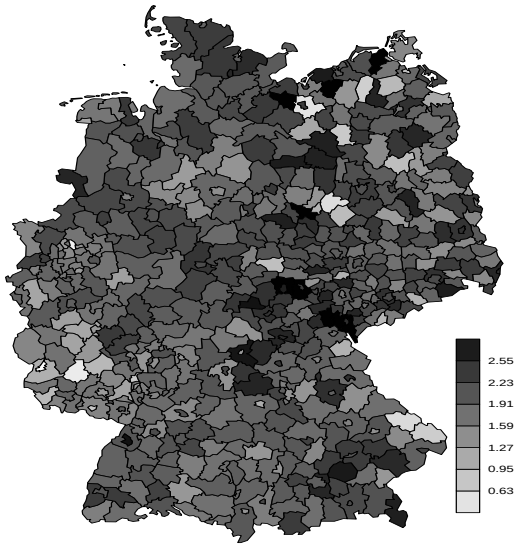
Larynx cancer mortality counts are observed in the 544 district of Germany from 1986 to 1990 and level of smoking consumption (100 possible values).

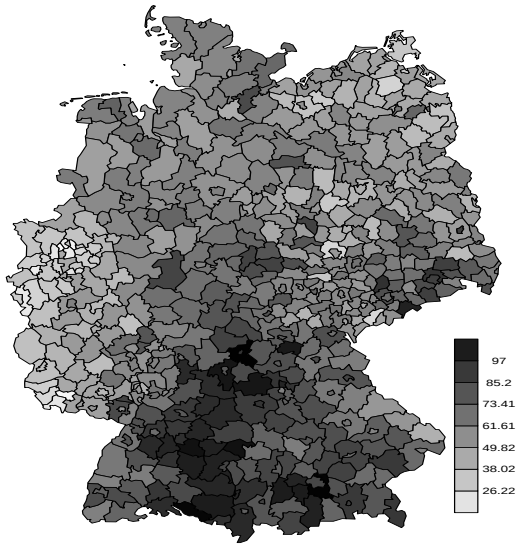




Disease mapping in Germany, Data

y_i, $i = 1, \ldots, 544$ counts of cancer mortality in Region i
E_i, $i = 1, \ldots, 544$ known variable accounting for demographic variation in Region i
c_i, $i = 1, \ldots, 544$ level of smoking consumption registered in Region i





The model

$$
\begin{aligned}
y_i &\sim \text{Poisson}\{E_i \exp(\eta_i)\};\ i = 1, \ldots, 544 \\
\eta_i &= \mu + f(c_i) + f_s(s_i) + f_u(s_i)
\end{aligned}
$$

where:

The model

$$\begin{aligned} y_i &\sim \text{Poisson}\{E_i \exp(\eta_i)\}; \; i = 1, \ldots, 544 \\ \eta_i &= \mu + f(c_i) + f_s(s_i) + f_u(s_i) \end{aligned}$$

where:

- $f(c_i)$ is a smooth effect of the covariate

$$\boldsymbol{f} = \{f_1, \ldots, f_{100}\} \sim \text{RW2}(\tau_f)$$

The model

$$\begin{aligned} y_i &\sim \text{Poisson}\{E_i \exp(\eta_i)\};\ i = 1, \ldots, 544 \\ \eta_i &= \mu + f(c_i) + f_s(s_i) + f_u(s_i) \end{aligned}$$

where:

- $f(c_i)$ is a smooth effect of the covariate

$$\boldsymbol{f} = \{f_1, \ldots, f_{100}\} \sim \text{RW2}(\tau_f)$$

- $f_s(s_i)$ is a spatial effect modelled as an intrinsic GMRF

$$f_s(s)|f_s(s'), s \neq s', \lambda_s \sim \mathcal{N}(\frac{1}{n_s}\sum_{s \sim s'} f_s(s'), \frac{\tau_{f_s}}{n_s})$$

The model

$$\begin{aligned} y_i &\sim \text{Poisson}\{E_i \exp(\eta_i)\};\ i = 1, \ldots, 544 \\ \eta_i &= \mu + f(c_i) + f_s(s_i) + f_u(s_i) \end{aligned}$$

where:

- $f(c_i)$ is a smooth effect of the covariate

$$\boldsymbol{f} = \{f_1, \ldots, f_{100}\} \sim \text{RW2}(\tau_f)$$

- $f_s(s_i)$ is a spatial effect modelled as an intrinsic GMRF

$$f_s(s)|f_s(s'), s \neq s', \lambda_s \sim \mathcal{N}(\frac{1}{n_s}\sum_{s\sim s'} f_s(s'), \frac{\tau_{f_s}}{n_s})$$

- $f_u(s_i)$ is a random effect

$$\boldsymbol{f}_u = \{f_u(s_1), \ldots, f_u(s_{544})\} \sim \mathbf{N}(0, \tau_{f_u}\boldsymbol{I})$$

The model

$$\begin{aligned} y_i &\sim \text{Poisson}\{E_i \exp(\eta_i)\};\ i = 1, \ldots, 544 \\ \eta_i &= \mu + f(c_i) + f_s(s_i) + f_u(s_i) \end{aligned}$$

where:

- $f(c_i)$ is a smooth effect of the covariate

$$\boldsymbol{f} = \{f_1, \ldots, f_{100}\} \sim \text{RW2}(\tau_f)$$

- $f_s(s_i)$ is a spatial effect modelled as an intrinsic GMRF

$$f_s(s)|f_s(s'), s \neq s', \lambda_s \sim \mathcal{N}(\frac{1}{n_s}\sum_{s\sim s'} f_s(s'), \frac{\tau_{f_s}}{n_s})$$

- $f_u(s_i)$ is a random effect

$$\boldsymbol{f}_u = \{f_u(s_1), \ldots, f_u(s_{544})\} \sim \mathbf{N}(0, \tau_{f_u}\boldsymbol{I})$$

- μ is an intercept term $\mu \sim \mathcal{N}(0, 0.0001)$

For identifiably we define a sum-to-zero constraint for all intrinsic models, so

$$\begin{aligned} \sum_s f_s(s) &= 0 \\ \sum_i f_i &= 0 \end{aligned}$$

The `Germany` data frame:

region	E	Y	x
0	7.965008	8	56
1	22.836219	22	65

The model is:

$$\eta_i = \mu + f(c_i) + f_s(s_i) + f_u(s_i)$$

The `Germany` data frame:

region	E	Y	x
0	7.965008	8	56
1	22.836219	22	65

The model is:

$$\eta_i = \mu + f(c_i) + f_s(s_i) + f_u(s_i)$$

- The data set has to contain *one separate column for each term specified through* `f()` so in this case we have to add one column.

```
> Germany = cbind(Germany, region.struct=Germany$region)
```

The `Germany` data frame:

region	E	Y	x
0	7.965008	8	56
1	22.836219	22	65

The model is:

$$\eta_i = \mu + f(c_i) + f_s(s_i) + f_u(s_i)$$

- The data set has to contain *one separate column for each term specified through* `f()` so in this case we have to add one column.
  `> Germany = cbind(Germany, region.struct=Germany$region)`
- We also need the graph file where the neighbourhood structure is specified `germany.graph`

The new data set is:

region	E	Y	x	region.struct
0	7.965008	8	56	0
1	22.836219	22	65	1

Then the formula is

```
formula <- Y ~
f(region.struct,model="besag",graph.file="germany.graph")+
f(x,model="rw2")+f(region)
```

The new data set is:

region	E	Y	x	region.struct
0	7.965008	8	56	0
1	22.836219	22	65	1

Then the formula is

```
formula <- Y ~
f(region.struct,model="besag",graph.file="germany.graph")+
f(x,model="rw2")+f(region)
```

The sum-to-zero constraint is *default* in the `inla` function for all *intrinsic models*.

The new data set is:

region	E	Y	x	region.struct
0	7.965008	8	56	0
1	22.836219	22	65	1

Then the formula is

```
formula <- Y ~
f(region.struct,model="besag",graph.file="germany.graph")+
f(x,model="rw2")+f(region)
```

The sum-to-zero constraint is *default* in the `inla` function for all *intrinsic models*.

The new data set is:

region	E	Y	x	region.struct
0	7.965008	8	56	0
1	22.836219	22	65	1

Then the formula is

```
formula <- Y ~
f(region.struct,model="besag",graph.file="germany.graph")+
f(x,model="rw2")+f(region)
```

The new data set is:

region	E	Y	x	region.struct
0	7.965008	8	56	0
1	22.836219	22	65	1

Then the formula is

```
formula <- Y ~
f(region.struct,model="besag",graph.file="germany.graph")+
f(x,model="rw2")+f(region)
```

The location of the graph file has to be provided here (the graph file cannot be loaded in R)

The graph file

The `germany.graph` file:

```
544
 1    1   12
 2    2   10   11
 3    4    6    8   15   387
 ⋮
```

The graph file

The `germany.graph` file:

544

1 1 12

2 2 10 11

3 4 6 8 15 387

⋮

- Total number of nodes in the graph

The graph file

The `germany.graph` file:

```
544
1   1   12
2   2   10   11
3   4   6    8    15   387
⋮
```

- Total number of nodes in the graph
- Identifier for the node

The graph file

The `germany.graph` file:

```
544
 1    1   12
 2    2   10   11
 3    4    6    8   15   387
 ⋮
```

- Total number of nodes in the graph
- Identifier for the node
- Number of neighbours

The graph file

The `germany.graph` file:

544
1 1 12
2 2 10 11
3 4 6 8 15 387
⋮

- Total number of nodes in the graph
- Identifier for the node
- Number of neighbours
- Identifiers for the neighbours

```
data(Germany)
g = system.file("demodata/germany.graph", package="INLA")
source(system.file("demodata/Bym-map.R", package="INLA"))
Germany = cbind(Germany, region.struct=Germany$region)

# standard BYM model
formula1 = Y ~ f(region.struct,model="besag",graph.file=g) +
               f(region,model="iid")

# with linear covariate
formula2 = Y ~ f(region.struct,model="besag",graph.file=g) +
               f(region,model="iid") + x

# with smooth covariate
formula3 = Y ~ f(region.struct,model="besag",graph.file=g) +
               f(region,model="iid") + f(x, model="rw2")

result1 = inla(formula1,family="poisson",data=Germany,E=E,
               control.compute=list(dic=TRUE))
result2 = inla(formula2,family="poisson",data=Germany,E=E,
               control.compute=list(dic=TRUE))
```

```
result1 = inla(formula1,family="poisson",data=Germany,E=E,
               control.compute=list(dic=TRUE))

result2 = inla(formula2,family="poisson",data=Germany,E=E,
               control.compute=list(dic=TRUE))

result3 = inla(formula3,family="poisson",data=Germany,E=E,
               control.compute=list(dic=TRUE))
```

Survival models

patient	time	event	age	sex
1	8,16	1,1	28,28	0
2	23,13	1,0	48,48	1
3	22,18	1,1	32,32	0

- Times of infection from the time of insertion of catheter on 38 kidney patients using portable dialysis equipment.
- 2 observation for each patient (38 patients).
- Each time can be an *event* (infection) or a *censoring* (no infection)

Hazard rate and survival function

Density function:

$$y \sim f(y)$$

Survival function:

$$S(y) = 1 - F(y) = \int_y^\infty f(u)\ du$$

Hazard function:

$$\begin{aligned} h(y)\ dy &= \text{Prob}(y \leq Y < y + \ dy | Y > y) \\ h(y) &= \frac{f(t)}{S(t)} \end{aligned}$$

Cox proportinal hazards model

Write the hazard function for each patient as:

$$h(y_{ij}|w_i, \boldsymbol{x}_{ij}) = h_0(y_{ij})\ w_i\ \exp(\boldsymbol{x}_{ij}^T \boldsymbol{\beta});\ \ i = 1, \ldots, 38;\ \ j = 1, 2$$

where

$h_0(\cdot)$	is the baseline hazard function
w_i	is the log-Normal frailty effect associated with patient i
$\boldsymbol{x}_{ij}$	is the vector of observed covariates for patient i at observation j
$\boldsymbol{\beta}$	is a vector of unknown parameters

Cox proportinal hazard model

Can rewrite this a Poisson regression, augmenting data for each part of the picewise-constant baseline hazard.

The Kidney data

The `Kidney` data frame

time	event	age	sex	ID
8	1	28	0	1
16	1	28	0	1
23	1	48	1	2
13	0	48	1	2
22	1	32	0	3
28	1	32	0	3

Controlling $\boldsymbol{\theta}$

- We often need to set our own priors and using our own parameters in these.

Controlling $\boldsymbol{\theta}$

- We often need to set our own priors and using our own parameters in these.
- These can be set in two ways

Controlling $\boldsymbol{\theta}$

- We often need to set our own priors and using our own parameters in these.
- These can be set in two ways

  Old style using `prior=..`, `param=...`, `initial=...`, `fixed=...`

Controlling $\boldsymbol{\theta}$

- We often need to set our own priors and using our own parameters in these.
- These can be set in two ways

  Old style using `prior=.., param=..., initial=..., fixed=...`

  New style using `hyper = list(prec = list(initial=2, fixed=TRUE, ....))`

Controlling $\boldsymbol{\theta}$

- We often need to set our own priors and using our own parameters in these.
- These can be set in two ways

  Old style using `prior=.., param=..., initial=..., fixed=...`

  New style using `hyper = list(prec = list(initial=2, fixed=TRUE, ....))`

Controlling $\boldsymbol{\theta}$

- We often need to set our own priors and using our own parameters in these.
- These can be set in two ways

  Old style using `prior=.., param=..., initial=..., fixed=...`

  New style using `hyper = list(prec = list(initial=2, fixed=TRUE, ....))`

The old-style is there for backward-compatibility only. The two styles can also be mixed.

Example: AR1 model I

hyper

theta1

name	log precision
short.name	prec
prior	loggamma
param	1 5e-05
initial	4
fixed	FALSE
to.theta	`function(x) log(x)`
from.theta	`function(x) exp(x)`

Example: AR1 model

hyper

theta2

name	logit lag one correlation
short.name	rho
prior	normal
param	0 0.15
initial	2
fixed	FALSE
to.theta	`function(x) log((1+x)/(1-x))`
from.theta	`function(x) 2*exp(x)/(1+exp(x))-1`

Example: AR1 model

constr	FALSE
nrow.ncol	FALSE
augmented	FALSE
aug.factor	1
aug.constr	
n.div.by	
n.required	FALSE
set.default.values	FALSE
pdf	ar1

Example

```
    hyper = list(
                rho = list(
                            prior = "normal",
                            param = c(0,1),
                            initial = 3,
                            fixed = FALSE
                           )
                )

formula = y ~ f(i, model="ar1", hyper = hyper) + ...
```

These values are also available within R as

```
inla.model.properties("ar1", "latent")
```

```
inla.model.properties(<name>, <section>)
```

where *section* is one of *latent*, *group*, *predictor*, *hazard*, *likelihood*, *prior*, *wrapper*
Possible names can be extracted as

```
names(inla.models()$<section>)
```

Interal and external scale

Hyperparamters, like ρ and τ is represented internally using a "good" transformation, like

$$\theta_1 = \log(\tau)$$

$$\theta_2 = \log(\frac{1+\rho}{1-\rho})$$

Interal and external scale

Hyperparamters, like ρ and τ is represented internally using a "good" transformation, like

$$\theta_1 = \log(\tau)$$

$$\theta_2 = \log(\frac{1+\rho}{1-\rho})$$

- Initial values are given in the internal scale

Interal and external scale

Hyperparamters, like ρ and τ is represented internally using a "good" transformation, like

$$\theta_1 = \log(\tau)$$

$$\theta_2 = \log(\frac{1+\rho}{1-\rho})$$

- Initial values are given in the internal scale
- the *to.theta* and *from.theta* functions can be used to map between the external and internal scale.

Some more advanced features

- replicate
- more than one "family"
- copy
- linear combinations
- $\boldsymbol{A}$ matrix in the linear predictor
- remote computing

Feature: replicate

"replicate" generates iid replicates from the same model with the same hyperparameters.

If $\boldsymbol{x} \mid \boldsymbol{\theta} \sim$ AR(1), then nrep=3, makes

$$\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3)$$

with mutually independent $\boldsymbol{x}_i$'s from AR(1) with the same $\boldsymbol{\theta}$

Most `f()`-models can be replicated

Example: replicate.R

```
n=100
x1 = arima.sim(n, model=list(ar=0.9)) + 1
x2 = arima.sim(n, model=list(ar=0.9)) - 1
y1 = rpois(n,exp(x1))
y2 = rpois(n,exp(x2))
y = c(y1,y2)
i = rep(1:n,2)
r = rep(1:2,each=n)
intercept = as.factor(r)
formula = y ~ f(i, model="ar1", replicate=r) + intercept -1
result = inla(formula, family = "poisson",
              data = data.frame(y=y,i=i,r=r))
```

More than one family, `nfamily.R`

Every observation could have its own likelihood!

- Response is a matrix or list
- Each "column" defines a separate "family"
- Each "family" has its own hyperparameters

```
n=100
s = 0.5
phi = 0.9
x1 = 1   + s * arima.sim(n, model=list(ar=phi)) * sqrt(1-phi^2)
x2 = 0.5 + s * arima.sim(n, model=list(ar=phi)) * sqrt(1-phi^2)
y1 = rbinom(n,size=1, prob=exp(x1)/(1+exp(x1)))
y2 = rpois(n,exp(x2))
y = matrix(NA, 2*n, 2)
y[  1:n, 1] = y1
y[n+1:n, 2] = y2
i = rep(1:n,2)
r = rep(1:2,each=n)
intercept = as.factor(r)
Ntrials = c(rep(1,n), rep(NA,n))

formula = y ~ f(i, model="ar1", replicate=r) + intercept -1
result = inla(formula, family = c("binomial", "poisson"),
                Ntrials = Ntrials, data = data.frame(y,i,r),
        verbose=T)
```

More examples

Some rather advanced examples on `www.r-inla.org` using this feature

- Preferential sampling, geostatistics (marked point process)
- Weibull-survival data and "longitudinal" data

Feature: copy

The model

```
formula = y ~ f(i, ...) + ...
```

Only allow ONE element from each sub-model, to contribute to the linear predictor for each observation.

Sometimes this is not sufficient.

Feature: copy

Suppose

$$\eta_i = u_i + u_{i+1} + ...$$

Then we can code this as

```
formula = f(i, model="iid") + f(i.plus, copy="i")
```

- The copy-feature, creates an additional sub-model which is ϵ-close to the target.
- Many copies allowed
- Copy with unknown scaling (default scaling is fixed to 1).

A toy-example using *copy*

State-space model

$$y_t = x_t + v_t$$

$$x_t = 2x_{t-1} - x_{t-2} + w_t$$

A toy-example using *copy*

State-space model

$$y_t = x_t + v_t$$

$$x_t = 2x_{t-1} - x_{t-2} + w_t$$

Rewrite this as

$$y_t = x_t + v_t$$

$$0 = x_t - 2x_{t-1} + x_{t-2} + w_t$$

and implement this as two families

1. Observations y_t with precision Prec(v_t)
2. Observations 0 with precision Prec(w_t), or Prec=HIGH.

```
n = 100
m = n-2
y = sin((1:n)*0.2) + rnorm(n, sd=0.1)
formula = Y ~ f(i, model="iid", initial=-10, fixed=TRUE) +
              f(j, w, copy="i") + f(k, copy="i") +
              f(l, model ="iid") -1
Y = matrix(NA, n+m, 2)
Y[1:n,     1] = y
Y[1:m + n, 2] = 0
i = c(1:n, 3:n)                 # x_t
j = c(rep(NA,n), 3:n -1)        # x_t-1
w = c(rep(NA,n), rep(-2,m)) # weights for j
k = c(rep(NA,n), 3:n -2)        # x_t-2
l = c(rep(NA,n), 1:m)           # v_t
r = inla(formula, data = data.frame(i,j,w,k,l,Y),
         family = c("gaussian", "gaussian"),
         control.data = list(list(), list(initial=10, fixed=TRUE)))
```

Linear combinations (I)

Possible to extract extra information from the model through linear combinations of the latent field, say

$$\boldsymbol{v} = \boldsymbol{B}\boldsymbol{x}$$

for a $k \times n$ matrix $\boldsymbol{B}$.

Linear combinations, `linear-combinations.R`

Two different approaches.

1. Most "correct" is to do the computations on the enlarged field

$$\widetilde{\boldsymbol{x}} = (\boldsymbol{x}, \boldsymbol{v})$$

   But this often lead to more dense precision matrix.

Linear combinations, `linear-combinations.R`

Two different approaches.

1. Most "correct" is to do the computations on the enlarged field

$$\widetilde{\boldsymbol{x}} = (\boldsymbol{x}, \boldsymbol{v})$$

   But this often lead to more dense precision matrix.

2. The second option is to compute these "offline", as (conditionally on $\boldsymbol{\theta}$)

$$\text{Var}(v_1) = \text{Var}(\boldsymbol{b}_1^T \boldsymbol{x}) \approx \boldsymbol{b}_1^T \boldsymbol{Q}^{-1}_{\text{GMRFapprox}} \boldsymbol{b}_1$$

   and

$$\mathbb{E}(v_1) = \boldsymbol{b}_1 \mathbb{E}(\boldsymbol{x})$$

   Approximate density of v_1 with a Normal.

```
data(Epil)
my.center = function(x) (x - mean(x))

Epil$CTrt    = my.center(Epil$Trt)
Epil$ClBase4 = my.center(log(Epil$Base/4))
Epil$CV4     = my.center(Epil$V4)
Epil$ClAge   = my.center(log(Epil$Age))

formula = y ~ ClBase4*CTrt + ClAge + CV4 +
    f(Ind, model="iid") + f(rand, model="iid")

## Now I want the posterior for
##
##   1)    2*CTrt - CV4
##   2)    Ind[2] - rand[2]
##
lc1 = inla.make.lincomb( CTrt = 2, CV4 = -1)
names(lc1) = "lc1"
lc2 = inla.make.lincomb( Ind = c(NA,1), rand = c(NA,-1))
names(lc2) = "lc2"
```

$\boldsymbol{A}$-matrix in the linear predictor

Usual formula

$$\eta = ...$$

and

$$y_i \sim \pi(y_i \mid \eta_i, ...)$$

$\boldsymbol{A}$-matrix in the linear predictor

Extended formula

$$\eta = ...$$

$$\eta^* = \boldsymbol{A}\eta$$

and

$$y_i \sim \pi(y_i \mid \eta_i^*, ...)$$

$\boldsymbol{A}$-matrix in the linear predictor

Extended formula

$$\eta = ...$$

$$\eta^* = \boldsymbol{A}\eta$$

and

$$y_i \sim \pi(y_i \mid \eta_i^*, ...)$$

Implemented as

```
A = matrix(...)
A = sparseMatrix(...)
result = inla(formula, ...,
              control.predictor = list(A = A))
```

$\boldsymbol{A}$-matrix in the linear predictor, `example-A.R`

- Can *really* simplify model-formulations
- Duplicate to some extent the "copy" feature
- Really usefull for some models; the $\boldsymbol{A}$-matrix need not to be a square matrix...

```
n = 100
phi = 0.9
x = as.numeric(arima.sim(n=n,  model = list(ar = phi)))
prec.x = 1 - phi^2
eta = x

h = 15L
nh = 2L*h + 1L
kern = dnorm(-h:h, sd= h/3)
kern = kern / sum(kern)

A = toeplitz(c(kern[-(1:h)],  rep(0,  n-nh),  kern[1:h]))
y = rpois(n, lambda = exp(A %*% eta))

i = 1:n
formula = y ~ f(i,  model="ar1") -1
r = inla(formula, control.predictor = list(A = A),
        data = data.frame(y, i), family = "poisson")



dev.new()
```

Feature: remote computing

For large/huge models, its more convenient to run the computations on the remote (Linux/Mac) computational server

```
inla(...., inla.call="remote")
```

using `ssh` (and `Cygwin` on windows).