



INLA for Spatial Statistics

2. An introduction to R-INLA

Daniel Simpson

Department of Mathematical Sciences
University of Bath

INLA - Integrated nested Laplace approximations

Outline

- ▶ Describe the class of models INLA can be applied to.
- ▶ Look at simple examples in R-INLA.

Outline

Background

Latent Gaussian model

Gibbs samplers and MCMC for LGMs

A case for approximate inference

R-INLA

Summary

Two main paradigms for statistical analysis

- ▶ Let \mathbf{y} denote a set of observations, distributed according to a probability model $\pi(\mathbf{y}; \boldsymbol{\theta})$.
- ▶ Based on the observations, we want to estimate $\boldsymbol{\theta}$.

The classical approach:

$\boldsymbol{\theta}$ denotes **parameters** (unknown fixed numbers), estimated for example by maximum likelihood.

Two main paradigms for statistical analysis

- ▶ Let \mathbf{y} denote a set of observations, distributed according to a probability model $\pi(\mathbf{y}; \boldsymbol{\theta})$.
- ▶ Based on the observations, we want to estimate $\boldsymbol{\theta}$.

The classical approach:

$\boldsymbol{\theta}$ denotes **parameters** (unknown fixed numbers), estimated for example by maximum likelihood.

The Bayesian approach:

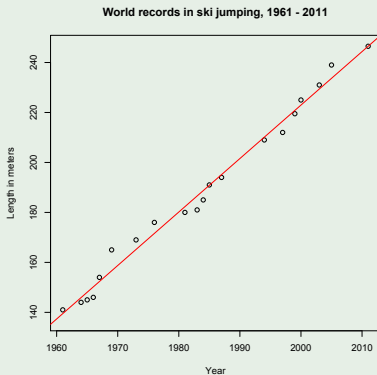
$\boldsymbol{\theta}$ denotes **random variables**, assigned a **prior** $\pi(\boldsymbol{\theta})$. Estimate $\boldsymbol{\theta}$ based on the **posterior**:

$$\pi(\boldsymbol{\theta} | \mathbf{y}) = \frac{\pi(\mathbf{y} | \boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\pi(\mathbf{y})} \propto \pi(\mathbf{y} | \boldsymbol{\theta})\pi(\boldsymbol{\theta}).$$

Example (Ski flying records)

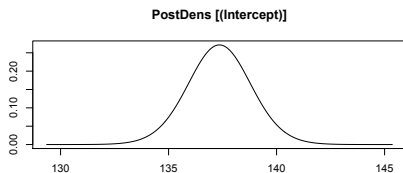
Assume a simple linear regression model with Gaussian observations $\mathbf{y} = (y_1, \dots, y_n)$, where

$$E(y_i) = \alpha + \beta x_i, \quad \text{Var}(y_i) = \tau^{-1}, \quad i = 1, \dots, n$$

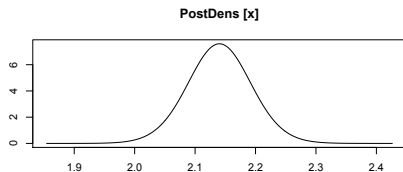


The Bayesian approach

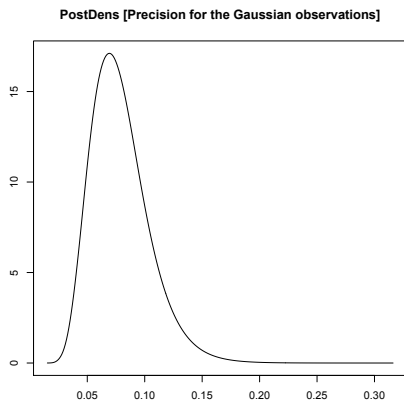
Assign priors to the parameters α , β and τ and calculate posteriors:



Mean = 137.354 SD = 1.508



Mean = 2.14 SD = 0.054



Real-world datasets are usually much more complicated!

Using a Bayesian framework:

- ▶ Build (hierarchical) models to account for potentially complicated dependency structures in the data.
- ▶ Attribute uncertainty to model parameters and latent variables using priors.

Real-world datasets are usually much more complicated!

Using a Bayesian framework:

- ▶ Build (hierarchical) models to account for potentially complicated dependency structures in the data.
- ▶ Attribute uncertainty to model parameters and latent variables using priors.

Two main challenges:

1. Need computationally efficient methods to calculate posteriors.
2. Select priors in a sensible way.

MCMC: Markov chain Monte Carlo methods

Based on **sampling**. Construct Markov chains with the target posterior as stationary distribution.

- ▶ Extensively used within Bayesian inference since the 1980's.
- ▶ Flexible and general, sometimes the only thing we can do!
- ▶ Available for specific models using e.g. BUGS, JAGS, BayesX.
- ▶ In general, not straightforward to implement. Slow, convergence issues, etc.

INLA: Integrated nested Laplace approximations

Introduced by Rue, Martino and Chopin (2009). Posteriors are estimated using numerical approximations. **No sampling** needed!

- ▶ Unified framework for analysing a general class of statistical models, named latent Gaussian models.
- ▶ Accurate and computationally superior to MCMC methods!
- ▶ Easily accessible using the R-interface R-INLA, see www.r-inla.org.

Reference:

Rue, H., Martino, S. and Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations (with discussion). *Journal of the Royal Statistical Society, Series B*, **71**, 319–392.

Outline

Background

Latent Gaussian model

Computational framework and approximations

Gibbs samplers and MCMC for LGMs

A case for approximate inference

R-INLA

Summary

What is a latent Gaussian model?

Classical multiple linear regression model

The mean $\boldsymbol{\mu}$ of an n -dimensional observational vector \mathbf{y} is given by

$$\mu_i = \mathbf{E}(Y_i) = \alpha + \sum_{j=1}^{n_\beta} \beta_j z_{ji}, \quad i = 1, \dots, n$$

where

α : Intercept

$\boldsymbol{\beta}$: Linear effects of covariates \mathbf{z}

Account for non-Gaussian observations

Generalized linear model (GLM)

The mean μ is linked to the linear predictor η_i :

$$\eta_i = g(\mu_i) = \alpha + \sum_{j=1}^{n_\beta} \beta_j z_{ji}, \quad i = 1, \dots, n$$

where $g(\cdot)$ is a link function and

α : Intercept

β : Linear effects of covariates z

Account for non-linear effects of covariates

Generalized additive model (GAM)

The mean μ is linked to the linear predictor η_i :

$$\eta_i = g(\mu_i) = \alpha + \sum_{k=1}^{n_f} f_k(c_{ki}), \quad i = 1, \dots, n$$

where $g(\cdot)$ is a link function and

α : Intercept

$\{f_k(\cdot)\}$: Non-linear smooth effects of covariates c_k

Structured additive regression models

GLM/GAM/GLMM/GAMM+++

The mean μ is linked to the linear predictor η_i :

$$\eta_i = g(\mu_i) = \alpha + \sum_{j=1}^{n_\beta} \beta_j z_{ji} + \sum_{k=1}^{n_f} f_k(c_{ki}) + \epsilon_i, \quad i = 1, \dots, n$$

where $g(\cdot)$ is a link function and

α : Intercept

β : Linear effects of covariates z

$\{f_k(\cdot)\}$: Non-linear smooth effects of covariates c_k

ϵ : iid random effects

Latent Gaussian models

- ▶ Collect all parameters (random variables) in the linear predictor in a **latent field**

$$\mathbf{x} = \{\alpha, \boldsymbol{\beta}, \{f_k(\cdot)\}, \boldsymbol{\eta}\}.$$

- ▶ A latent Gaussian model is obtained by assigning Gaussian priors to all elements of \mathbf{x} .

Latent Gaussian models

- ▶ Collect all parameters (random variables) in the linear predictor in a **latent field**

$$\mathbf{x} = \{\alpha, \beta, \{f_k(\cdot)\}, \eta\}.$$

- ▶ A latent Gaussian model is obtained by assigning Gaussian priors to all elements of \mathbf{x} .
- ▶ Very flexible due to many different forms of the unknown functions $\{f_k(\cdot)\}$:
 - ▶ Include temporally and/or spatially indexed covariates.

Latent Gaussian models

- ▶ Collect all parameters (random variables) in the linear predictor in a **latent field**

$$\mathbf{x} = \{\alpha, \beta, \{f_k(\cdot)\}, \boldsymbol{\eta}\}.$$

- ▶ A latent Gaussian model is obtained by assigning Gaussian priors to all elements of \mathbf{x} .
- ▶ Very flexible due to many different forms of the unknown functions $\{f_k(\cdot)\}$:
 - ▶ Include temporally and/or spatially indexed covariates.
- ▶ **Hyperparameters** account for variability and length/strength of dependence.

Some examples of latent Gaussian models

- ▶ Generalized linear and additive (mixed) models
- ▶ Semiparametric regression
- ▶ Disease mapping
- ▶ Survival analysis
- ▶ **Log-Gaussian Cox-processes**
- ▶ Geostatistical models
- ▶ Spatial and spatio-temporal models
- ▶ Stochastic volatility
- ▶ Dynamic linear models
- ▶ State-space models
- ▶ +++

Unified framework: A three-stage hierarchical model

1. Observations: \mathbf{y}

2. Latent field: \mathbf{x}

3. Hyperparameters: θ

Unified framework: A three-stage hierarchical model

1. Observations: \mathbf{y}

Assumed **conditionally independent** given \mathbf{x} and $\boldsymbol{\theta}_1$:

2. Latent field: \mathbf{x}

Assumed to be a **GMRF** with a sparse precision matrix $\mathbf{Q}(\boldsymbol{\theta}_2)$:

3. Hyperparameters: $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$

Precision parameters of the Gaussian priors:

Unified framework: A three-stage hierarchical model

1. Observations: \mathbf{y}

Assumed **conditionally independent** given \mathbf{x} and $\boldsymbol{\theta}_1$:

$$\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}_1 \sim \prod_{i=1}^n \pi(y_i \mid x_i, \boldsymbol{\theta}_1).$$

2. Latent field: \mathbf{x}

Assumed to be a **GMRF** with a sparse precision matrix $\mathbf{Q}(\boldsymbol{\theta}_2)$:

$$\mathbf{x} \mid \boldsymbol{\theta}_2 \sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{\theta}_2), \mathbf{Q}^{-1}(\boldsymbol{\theta}_2)).$$

3. Hyperparameters: $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$

Precision parameters of the Gaussian priors:

$$\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta}).$$

Model summary

The joint posterior for the latent field and hyperparameters:

$$\begin{aligned}\pi(\mathbf{x}, \boldsymbol{\theta} \mid \mathbf{y}) &\propto \pi(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})\pi(\mathbf{x}, \boldsymbol{\theta}) \\ &\propto \prod_{i=1}^n \pi(y_i \mid x_i, \boldsymbol{\theta})\pi(\mathbf{x} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})\end{aligned}$$

Remarks:

- ▶ $m = \dim(\boldsymbol{\theta})$ is often quite small, like $m \leq 6$.
- ▶ $n = \dim(\mathbf{x})$ is often large, typically $n = 10^2 - 10^6$.

Target densities are given as high-dimensional integrals

We want to estimate:

- ▶ The marginals of all components of the latent field:

- ▶ The marginals of all the hyperparameters:

Target densities are given as high-dimensional integrals

We want to estimate:

- ▶ The marginals of all components of the latent field:

$$\begin{aligned}\pi(x_i | \mathbf{y}) &= \int \int \pi(\mathbf{x}, \boldsymbol{\theta} | \mathbf{y}) d\mathbf{x}_{-i} d\boldsymbol{\theta} \\ &= \int \pi(x_i | \boldsymbol{\theta}, \mathbf{y}) \pi(\boldsymbol{\theta} | \mathbf{y}) d\boldsymbol{\theta}, \quad i = 1, \dots, n.\end{aligned}$$

- ▶ The marginals of all the hyperparameters:

$$\begin{aligned}\pi(\theta_j | \mathbf{y}) &= \int \int \pi(\mathbf{x}, \boldsymbol{\theta} | \mathbf{y}) d\mathbf{x} d\boldsymbol{\theta}_{-j} \\ &= \int \pi(\boldsymbol{\theta} | \mathbf{y}) d\boldsymbol{\theta}_{-j}, \quad j = 1, \dots, m.\end{aligned}$$

Outline

Background

Latent Gaussian model

Gibbs samplers and MCMC for LGMs

A case for approximate inference

R-INLA

Summary

Outline

Background

Latent Gaussian model

Gibbs samplers and MCMC for LGMs

A case for approximate inference

Gibbs samplers are bad!

Blocking

Gaussian approximations

Independence sampler

Deterministic inference

Conclusions

Assessing the error

R-INLA

Summary

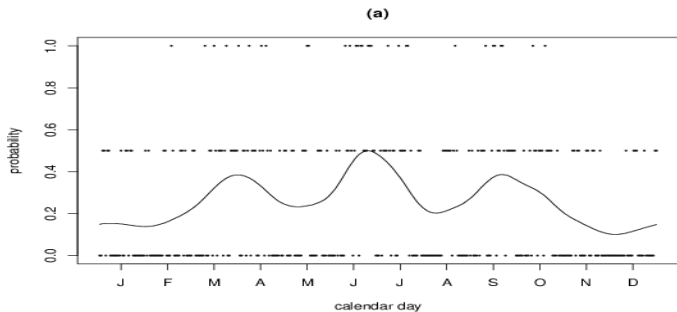
An MCMC case-study

- ▶ Study a seemingly trivial hierarchical model
 - ▶ Latent temporal Gaussian, with
 - ▶ Binary observations
- ▶ Develop a “standard” MCMC-algorithm for inference
 - ▶ Auxiliary variables
 - ▶ (Conjugate) single-site updates
- ▶ ..and study empirically its properties.

Auxillary aims

- ▶ Give a “historical” development of the ideas in INLA
- ▶ Show how to make good proposal distributions for latent Gaussian models
- ▶ Remind you not to make bad Gibbs samplers

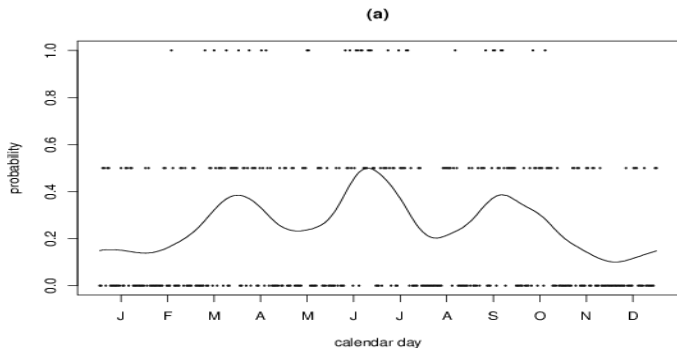
Tokyo rainfall data



Stage 1 Binomial data

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

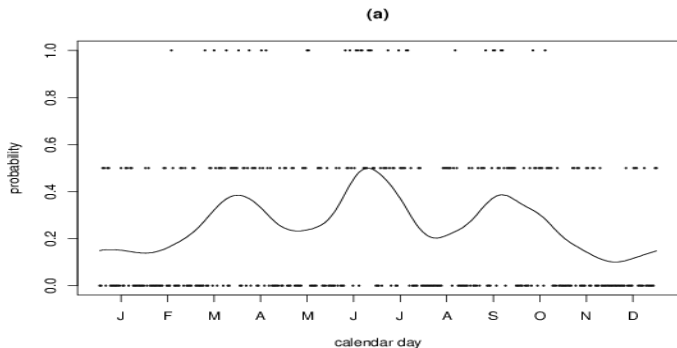
Tokyo rainfall data



Stage 2 Assume a smooth latent x ,

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

Tokyo rainfall data



Stage 3 Gamma(α, β)-prior on κ

Model summary

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

where

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

Construction of nice full conditionals

A popular approach is to introduce auxiliary variables w , so that

$x \mid \text{the rest}$

is Gaussian.

Example: Binary regression

GMRF x and Bernoulli data

$$y_i \sim \mathcal{B}(g^{-1}(x_i))$$
$$g(p) = \Phi(p) \quad \text{probit link}$$

Equivalent representation using auxiliary variables w

$$\epsilon_i \stackrel{\text{iid}}{\sim} \mathcal{N}(0, 1)$$
$$w_i = x_i + \epsilon_i$$
$$y_i = \begin{cases} 1 & \text{if } w_i > 0 \\ 0 & \text{otherwise.} \end{cases}$$

for the probit-link.

Single-site Gibbs sampling

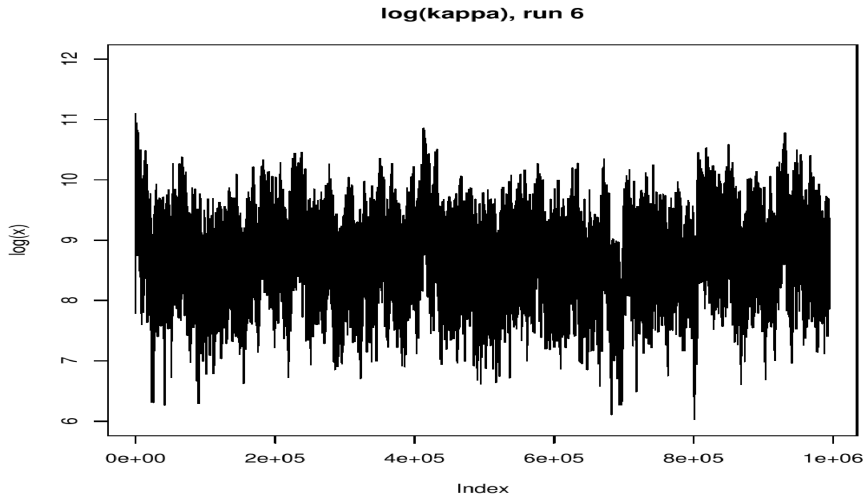
Auxiliary variables can be introduced for the logit-link¹, to achieve this sampler:

- ▶ $\kappa \sim \Gamma(\cdot, \cdot)$
- ▶ for each i
 - ▶ $x_i \sim \mathcal{N}(\cdot, \cdot)$
- ▶ for each i
 - ▶ $w_i \sim \mathcal{W}(\cdot)$

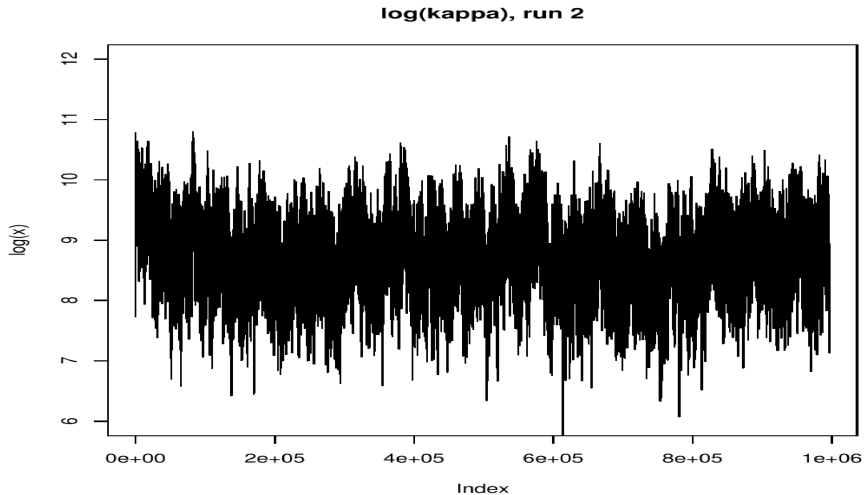
It is fully automatic; no tuning!!!

¹Held & Holmes (2006)

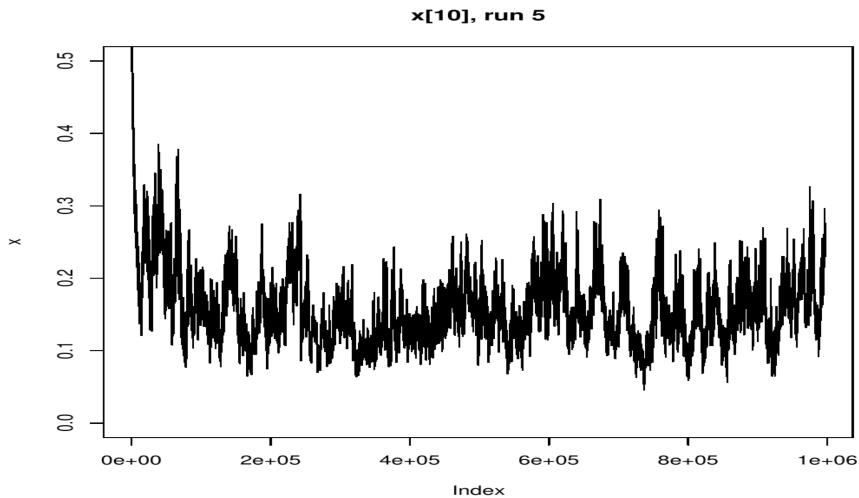
Results: hyper-parameter $\log(\kappa)$



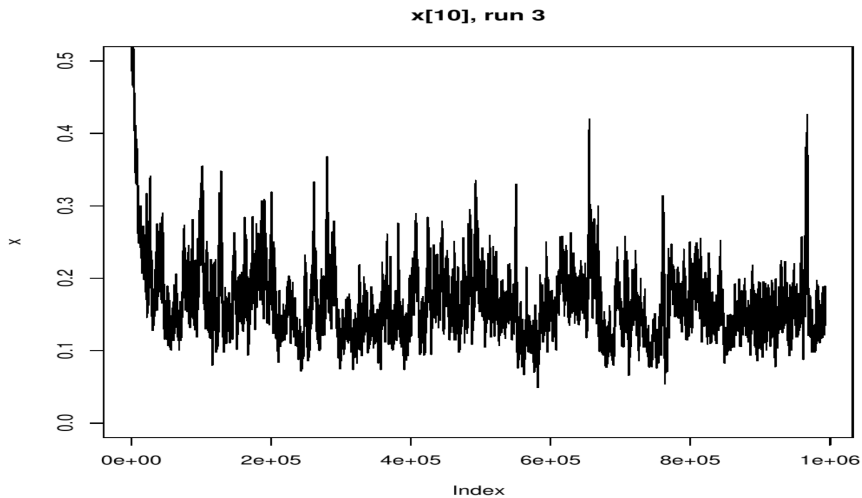
Results: hyper-parameter $\log(\kappa)$



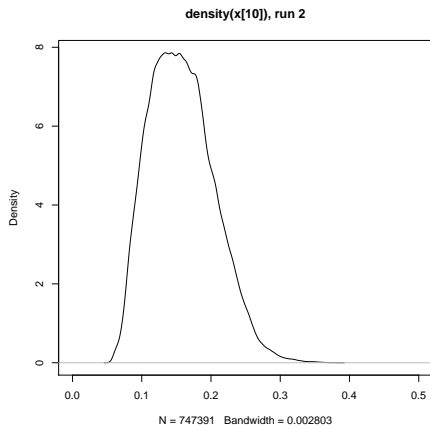
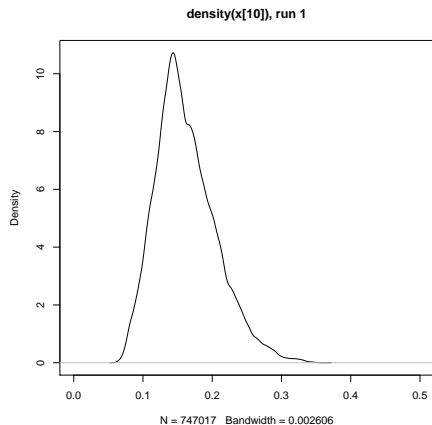
Results: latent node x_{10}



Results: latent node x_{10}



Results: density for latent node x_{10}



Discussion

Single-site sampler with auxiliary variables:

- ▶ Even *long runs* shows large variation
- ▶ “Long” range dependence
- ▶ *Very* slowly mixing

But:

- ▶ Easy to be “fooled” running shorter chains
- ▶ The variability can be underestimated.

What is causing the problem?

Two issues

1. Slow mixing within the latent field x
2. Slow mixing between the latent field x and θ .

Blocking is the “usual” approach to resolve such issues, if possible.

Note: blocking mainly helps within the block only.

Strategies for blocking

Slow mixing due to the latent field x only:

- ▶ Block x

Slow mixing due to the interaction between the latent field x and θ :

- ▶ Block (x, θ) .

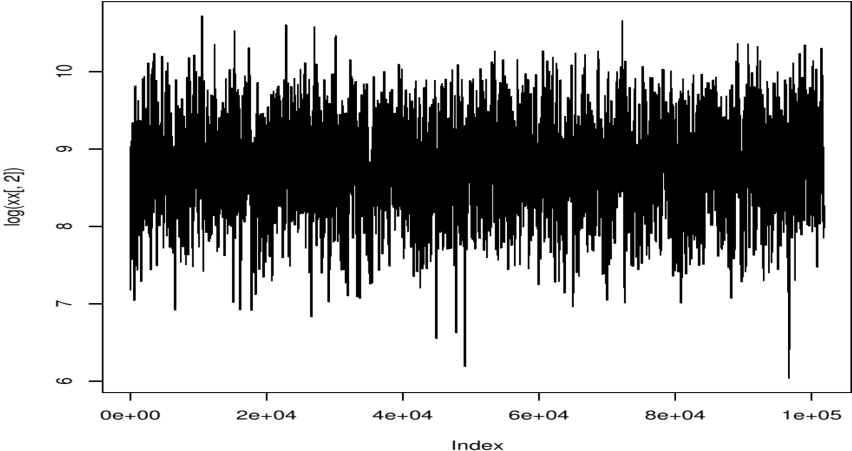
In most cases: if you can do one, you can do both.

Blocking scheme I

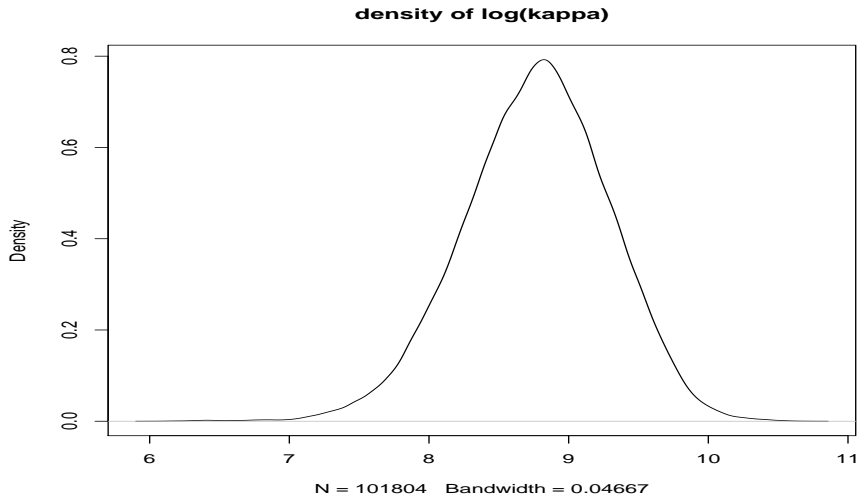
- ▶ $\kappa \sim \Gamma(\cdot, \cdot)$
- ▶ $\mathbf{x} \sim \mathcal{N}(\cdot, \cdot)$
- ▶ $\mathbf{w} \sim \mathcal{W}(\cdot)$ (conditional independent)

Results

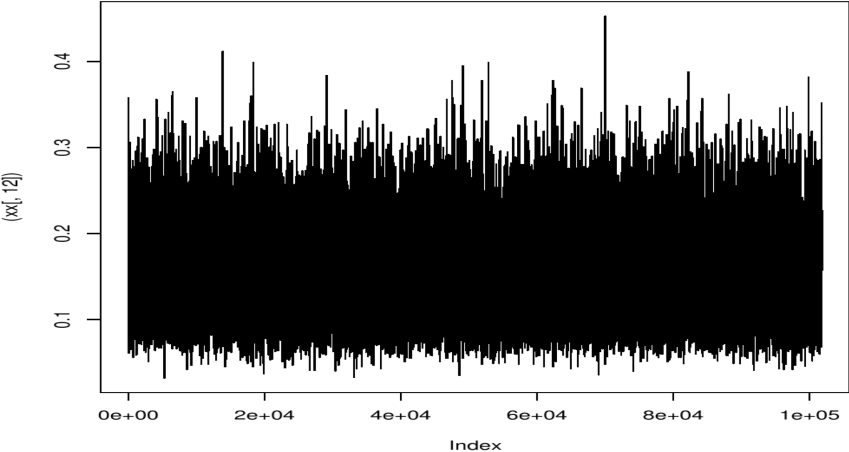
trace of log(kappa)



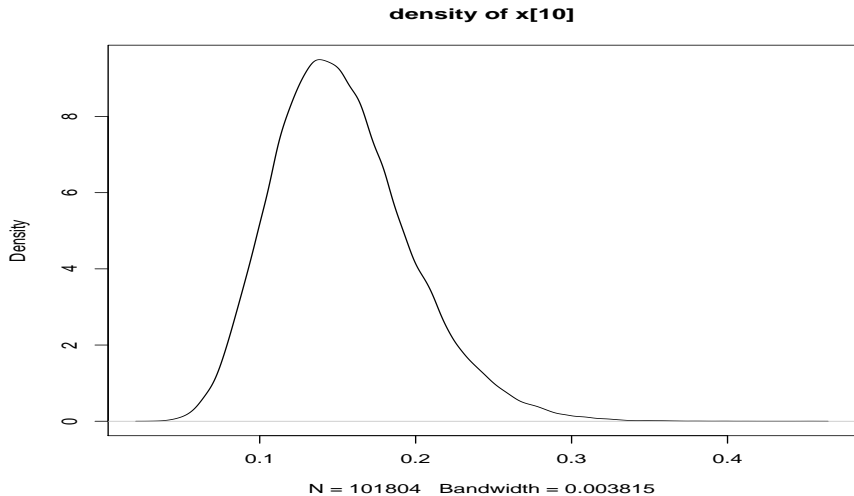
Results



Results



Results



Blocking scheme II

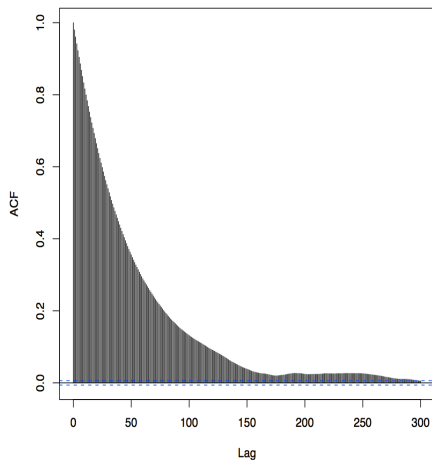
- ▶ Sample
 - ▶ $\kappa' \sim q(\kappa'; \kappa)$
 - ▶ $\mathbf{x}' | \kappa', \mathbf{y} \sim \mathcal{N}(\cdot, \cdot)$and then accept/reject (\mathbf{x}', κ') jointly
- ▶ $\mathbf{w} \sim \mathcal{W}(\cdot)$ (conditional independent)

Remarks

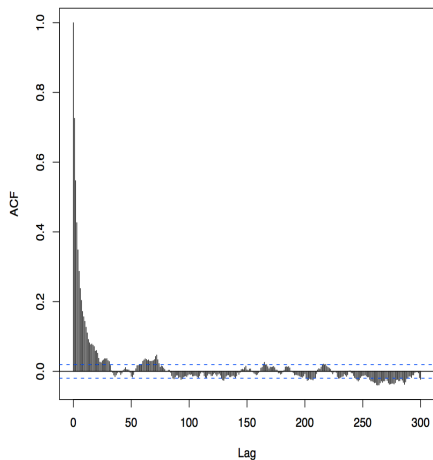
- ▶ If the normalising constant for $\mathbf{x}|\cdot$ is available, then this is an EASY FIX of scheme I.
- ▶ Usually makes a huge improvement
- ▶ Automatic “reparameterisation”
- ▶ Doubles the computational costs

Results

ACF(log(kappa) scheme I)



ACF(log(kappa) scheme II)



Removing the auxiliary variables

- ▶ The auxiliary variables makes the full conditional for \mathbf{x} Gaussian
- ▶ If we do not use them, the full conditional for \mathbf{x} looks like

$$\begin{aligned}\pi(\mathbf{x} \mid \dots) &\propto \exp\left(-\frac{1}{2}\mathbf{x}^T \mathbf{Q} \mathbf{x} + \sum_i \log(\pi(y_i \mid x_i))\right) \\ &\approx \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T (\mathbf{Q} + \text{diag}(\mathbf{c}))(\mathbf{x} - \boldsymbol{\mu})\right) \\ &= \pi_G(\mathbf{x} \mid \dots)\end{aligned}$$

- ▶ The Gaussian approximation is constructed by matching the
 - ▶ mode, and the
 - ▶ curvature at the mode.

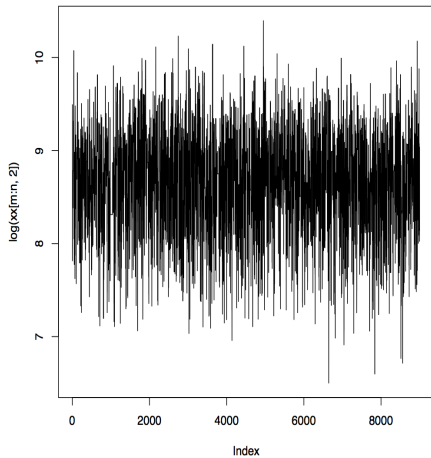
Improved one-block scheme

- ▶ $\kappa' \sim q(\cdot; \kappa)$
- ▶ $\mathbf{x}' \sim \pi_G(\mathbf{x} \mid \kappa', \mathbf{y})$
- ▶ Accept/reject (\mathbf{x}', κ') jointly

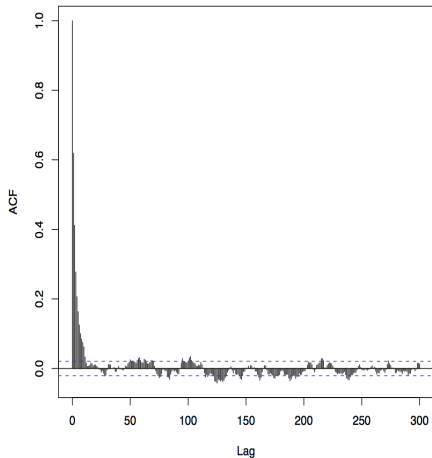
Note: $\pi_G(\cdot)$ is indexed by κ' , hence we need to compute one for each value of κ' .

Results

Trace of log(kappa)



ACF(log(kappa))



Independence sampler

We can construct an independence sampler, using $\pi_G(\cdot)$.
The Laplace-approximation for $\kappa|\mathbf{x}$:

$$\begin{aligned}\pi(\kappa | \mathbf{y}) &\propto \frac{\pi(\kappa) \pi(\mathbf{x}|\kappa) \pi(\mathbf{y}|\mathbf{x})}{\pi(\mathbf{x}|\kappa, \mathbf{y})} \\ &\approx \frac{\pi(\kappa) \pi(\mathbf{x}|\kappa) \pi(\mathbf{y}|\mathbf{x})}{\pi_G(\mathbf{x}|\kappa, \mathbf{y})} \Bigg|_{\mathbf{x}=\text{mode}(\kappa)}\end{aligned}$$

Hence, we do first

- ▶ Evaluate the Laplace-approximation at some “selected” points
- ▶ Build an interpolation log-spline
- ▶ Use this parametric model as $\tilde{\pi}(\kappa|\mathbf{y})$

Independence sampler

- ▶ $\kappa' \sim \tilde{\pi}(\kappa|\mathbf{y})$
- ▶ $\mathbf{x}' \sim \pi_G(\mathbf{x}|\kappa', \mathbf{y})$
- ▶ Accept/reject (κ', \mathbf{x}') jointly

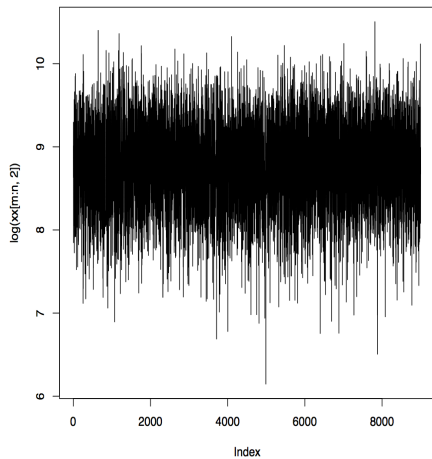
Note:

$$\text{Corr}(x(t+k), x(t)) \approx (1 - \alpha)^{|k|}$$

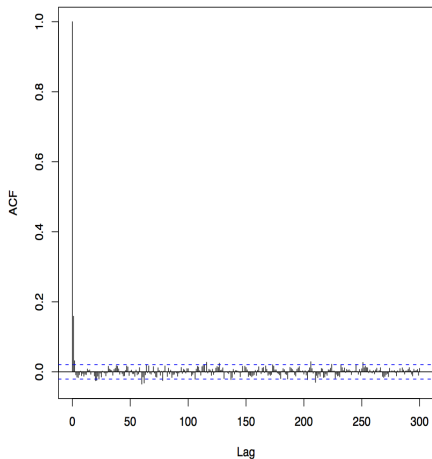
In this example, $\alpha = 0.83\dots$

Results

Trace log(kappa); independence sampler



ACF(log(kappa)); independence sampler



Can we improve this sampler?

- ▶ Yes, if we are interested in the posterior marginals for κ and $\{x_i\}$.
- ▶ The marginals for the Gaussian proposal $\pi_G(\mathbf{x}|\dots)$, are known analytically.
- ▶ Just use numerical integration!

Deterministic inference

Posterior marginal for κ :

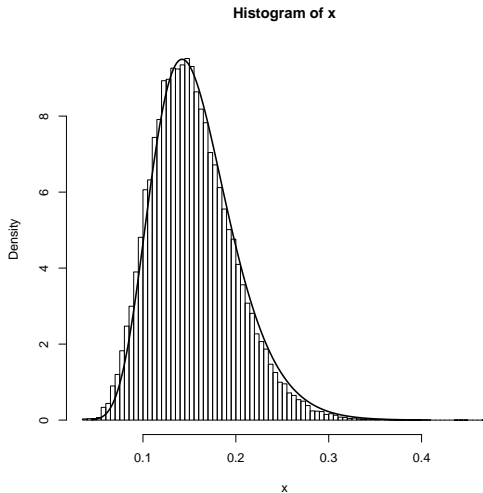
- ▶ Compute $\tilde{\pi}(\kappa | \mathbf{y})$

Posterior marginal for x_i :

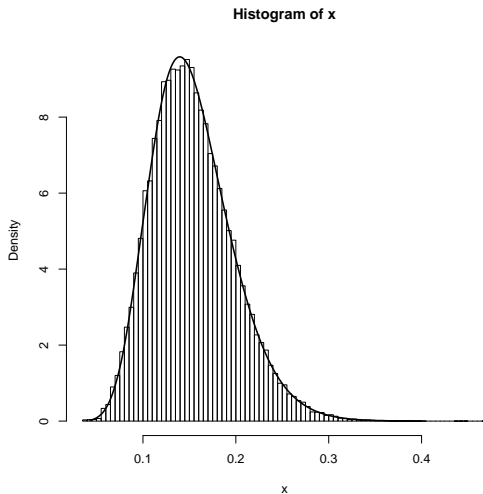
- ▶ Use numerical integration

$$\begin{aligned}\pi(x_i | \mathbf{y}) &= \int \pi(x_i | \mathbf{y}, \kappa) \pi(\kappa | \mathbf{y}) d\kappa \\ &\approx \sum_k \mathcal{N}(x_i; \mu_{\kappa_k}, \sigma^2(\kappa_k)) \times \tilde{\pi}(\kappa_k | \mathbf{y}) \times \Delta_k\end{aligned}$$

Results: Mixture of Gaussians



Results: Improved....



What can be learned from this exercise?

For a relative simple model, we have implemented

- ▶ single-site with auxiliary variables (long time; hours)
- ▶ various forms for blocking (long time; many minutes)
- ▶ independence sampler (long time; many minutes)
- ▶ approximate inference (nearly instant; one second)

What can be learned from this exercise? ...

Single-site Gibbs samplers don't work for when there's correlation.

This is completely unsurprising!

But they still get used. Which implies

- ▶ Most probably, the results would be not correct.
- ▶ They “accept” the long running-time.
- ▶ Trouble: such MCMC-schemes is not useful for routine analysis of similar data.

What can be learned from this exercise? ...

- ▶ In many cases, the situation is much worse in practice; this was a very simple model.
- ▶ Single-site MCMC is still the default choice for the non-expert user.
- ▶ Hierarchical models are popular, but they are difficult for MCMC.

Perhaps the development of models is not in sync with the development of inference? We cannot just wait for more powerful computers...

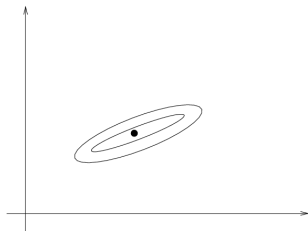
The integrated nested Laplace approximation (INLA) I

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

The integrated nested Laplace approximation (INLA) I

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

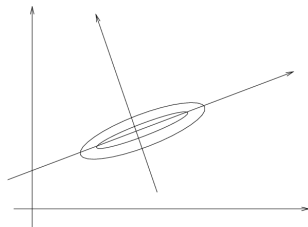
- ▶ Locate the mode



The integrated nested Laplace approximation (INLA) I

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

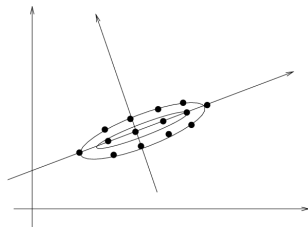
- ▶ Locate the mode
- ▶ Use the Hessian to construct new variables



The integrated nested Laplace approximation (INLA) I

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

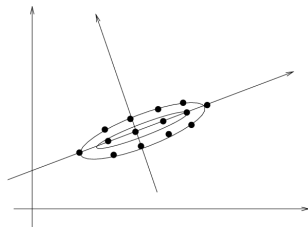
- ▶ Locate the mode
- ▶ Use the Hessian to construct new variables
- ▶ Grid-search



The integrated nested Laplace approximation (INLA) I

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

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



The integrated nested Laplace approximation (INLA) II

Step II For each θ_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 θ_j

- ▶ For each i , sum out θ

$$\tilde{\pi}(x_i | \mathbf{y}) \propto \sum_j \tilde{\pi}(x_i | \mathbf{y}, \theta_j) \times \tilde{\pi}(\theta_j | \mathbf{y})$$

- ▶ Build a log-spline corrected Gaussian

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

to represent $\tilde{\pi}(x_i | \mathbf{y})$.

Computing posterior marginals for θ_j (I)

Main idea

- ▶ Use the integration-points and build an interpolant
- ▶ Use numerical integration on that interpolant

Computing posterior marginals for θ_j (II)

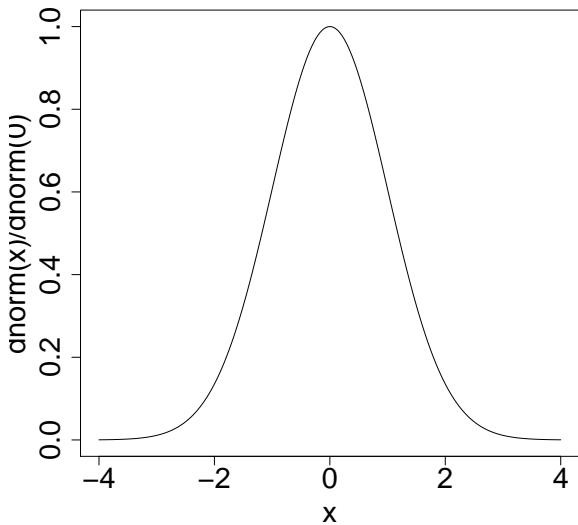
Practical approach (high accuracy)

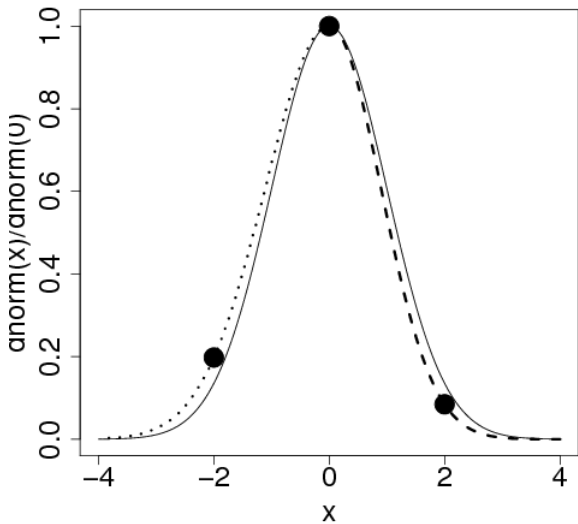
- ▶ Rerun using a fine integration grid
- ▶ Possibly with no rotation
- ▶ Just sum up at grid points, then interpolate

Computing posterior marginals for θ_j (II)

Practical approach (lower accuracy)

- ▶ Use the Gaussian approximation at the mode θ^*
- ▶ ...BUT, adjust the standard deviation in each direction
- ▶ Then use numerical integration





How can we assess the error in the approximations?

Tool 1: Compare a sequence of improved approximations

1. Gaussian approximation
2. Simplified Laplace
3. Laplace

How can we assess the error in the approximations?

Tool 3: Estimate the “effective” number of parameters as defined in the *Deviance Information Criteria*:

$$p_D(\boldsymbol{\theta}) = \overline{D}(\boldsymbol{x}; \boldsymbol{\theta}) - D(\overline{\boldsymbol{x}}; \boldsymbol{\theta})$$

and compare this with the number of observations.

Low ratio is good.

This criteria has theoretical justification.

Important observation

If $y|x, \theta$ is *Gaussian*, the “approximation” is exact.

Outline

Background

Latent Gaussian model

Gibbs samplers and MCMC for LGMs

A case for approximate inference

R-INLA

Logistic regression

Semiparametric regression

Summary

Anyone can use INLA!

The R-INLA project

- The home of the R-INLA project
 - Contact us, stay updated, get help or report an error
 - Discussion forum
 - Download
- ▼ Examples and tutorials
 - Case Studies
 - Tutorials
 - Volume I
 - Volume II
- FAQ
 - Getting started
 - Help
 - Internal use
- ▼ Models
 - Latent models
 - Likelihoods
 - Priors
 - Tools to manipulate models and likelihoods
- ▼ News
 - 10th Applied Statistics 2013 International conference, Slovenia
 - Bayes 2013: An introduction to INLA with a comparison to JAGS (hands on)
 - Bayesian Biostatistics short course: WinBUGS/SAS/INLA/MUSC May 2012
 - Bayesian Computing with INLA & Spatial Modeling Using

Bayesian computing with INLA!

This site provides documentation to the [R-INLA package](#) which solves a large class of statistical models using the [INLA](#) approach.

[Here](#) is a short introduction describing the class of models which can be solved using R-INLA.

All [models](#) implemented in R-INLA are described in details, moreover a large series of worked out [examples](#) are provided and we hope that this will help the user's familiarity with the library. Recent changes in the code can be [viewed here](#).

Recent posts to the discussion group

Google Group

NYTT EMNE


Hjelp


R-inla discussion group Delt offentlig Om

30 av 376 emner

Welcome to this discussion group about r-inla. Please ask your questions here in case you think they will be useful for others, otherwise send them to help@inla.org. You are of course free to comment on questions from others as well.


Best,
H


 Factor variable and random slope
Av Eric Coker - 2 innlegg - 4 visninger 13. mars


 Hyperparameters for rw1 and SPDE models
Av Ian Renner - 3 innlegg - 29 visninger 12. mars


Recent announcements

Recent Announcements

 [Open PhD-grants!](#) as attached. If you're interested, please contact inla.org
Posted 3 Mar 2014 19:59 by Havard Rue

 [Spatial Modelling with INLA Workshop, 2-4 June, St. Andre](#)
The link to the official page
Posted 23 Feb 2014 09:02 by Havard Rue

 [Short-course at the University of Girona, Spain April 24, 2014](#)
Gianluca Baio, details as attached.
Posted 29 Jan 2014 08:27 by Havard Rue

 [INLA-lectures in Florence, 28-29 Jan, 2014. The web-page is](#)
Posted 20 Jan 2014 10:03 by Havard Rue

Getting started with R-INLA

- ▶ Only once:

```
> source("http://www.math.ntnu.no/inla/givemeINLA.R")
```

- ▶ Load package and upgrade:

```
> library(INLA)
```

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

- ▶ Help and examples at www.r-inla.org

Basic structure to run a model

- ▶ Define the **formula**, specifying non-linear functions using `f(.)`, including the latent model and priors for hyperparameters:

```
> formula = y ~ 1 + z
           + f(c, model = "...",
              hyper = list(theta =
                            list(prior = "...", param = ...)))
```

- ▶ Call `inla(.)`, where you specify the relevant likelihood

```
> inla(formula, data=data.frame(...), family = "...")
```

Implemented models

Different likelihoods, latent models and (hyper)priors:

```
> names(inla.models())$likelihood)
```

```
> names(inla.models())$latent)
```

```
> names(inla.models())$prior)
```

Documentation (not complete):

```
> inla.doc("....")
```

Example: Logistic regression, 2×2 factorial design

Example (Seeds)

Consider the proportion of seeds that germinates on each of 21 plates. We have two seed types (x_1) and two root extracts (x_2).

```
> data(Seeds)
```

```
> head(Seeds)
```

	r	n	x1	x2	plate
1	10	39	0	0	1
2	23	62	0	0	2
3	23	81	0	0	3
4	26	51	0	0	4
5	17	39	0	0	5
6	5	6	0	1	6

Summary data set

Number of seeds that germinated in each group:

		Seed types	
		$x_1 = 0$	$x_1 = 1$
Root extract	$x_2 = 0$	99/272	49/123
	$x_2 = 1$	201/295	75/141

Statistical model

- ▶ Assume that the number of seeds that germinate on plate i is binomial

$$r_i \sim \text{Binomial}(n_i, p_i), \quad i = 1, \dots, 21,$$

- ▶ Logistic regression model:

$$\text{logit}(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \alpha + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{1i} x_{2i} + \epsilon_i$$

where $\epsilon_i \sim N(0, \sigma_\epsilon^2)$ are iid.

Statistical model

- ▶ Assume that the number of seeds that germinate on plate i is binomial

$$r_i \sim \text{Binomial}(n_i, p_i), \quad i = 1, \dots, 21,$$

- ▶ Logistic regression model:

$$\text{logit}(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \alpha + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{1i} x_{2i} + \epsilon_i$$

where $\epsilon_i \sim N(0, \sigma_\epsilon^2)$ are iid.

Aim:

Estimate the main effects, β_1 and β_2 and a possible interaction effect β_3 .

Using R-INLA

```
> formula = r ~ x1 + x2 + x1*x2 + f(plate, model="iid")
> result = inla(formula, data = Seeds,
                family = "binomial",
                Ntrials = n,
                control.predictor =
                    list(compute = T, link=1),
                control.compute = list(dic = T))
```

Default priors

Default prior for fixed effects is

$$\beta \sim N(0, 1000).$$

Change using the `control.fixed` argument in the `inla`-call.

Output

```
> summary(result)
```

```
Call:
```

```
"inla(formula = formula, family = \"binomial\", data = Seeds, Ntrials = n)"
```

```
Time used:
```

Pre-processing	Running inla	Post-processing	Total
0.1354	0.0911	0.0347	0.2613

```
Fixed effects:
```

	mean	sd	0.025quant	0.5quant	0.975quant	kld
(Intercept)	-0.5581	0.1261	-0.8076	-0.5573	-0.3130	0e+00
x1	0.1461	0.2233	-0.2933	0.1467	0.5823	0e+00
x2	1.3206	0.1776	0.9748	1.3197	1.6716	1e-04
x1:x2	-0.7793	0.3066	-1.3799	-0.7796	-0.1774	0e+00

```
Random effects:
```

Name	Model
plate	IID model

```
Model hyperparameters:
```

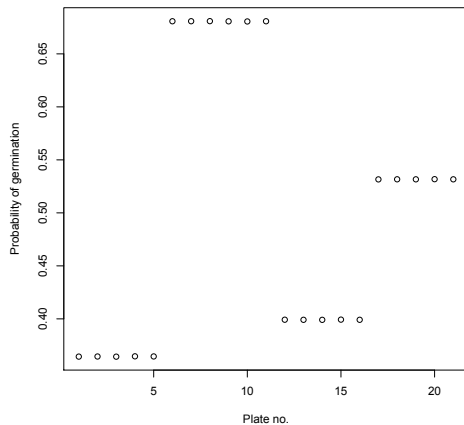
	mean	sd	0.025quant	0.5quant	0.975quant
Precision for plate	18413.03	18280.63	1217.90	13003.76	66486.29

```
Expected number of effective parameters(std dev): 4.014(0.0114)
```

```
Number of equivalent replicates : 5.231
```

Estimated germination probabilities

```
> result$summary.fitted.values$mean
```

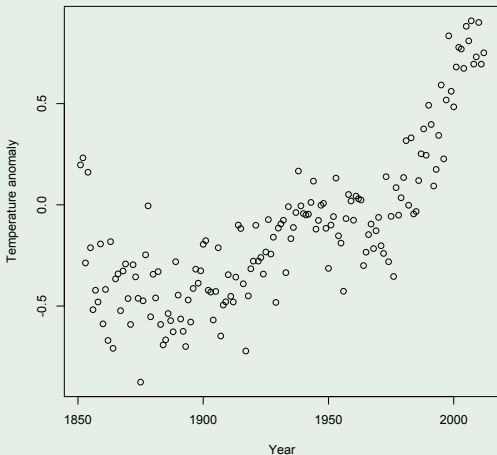


More in the practicals ...

```
> plot(result)
> result$summary.fixed
> result$summary.random
> result$summary.linear.predictor
> result$summary.fitted.values
> result$marginals.fixed
> result$marginals.hyperpar
> result$marginals.linear.predictor
> result$marginals.fitted.values
```

Example: Semiparametric regression

Example (Annual global temperature anomalies)



Estimating a smooth non-linear trend

- ▶ Assume the model

$$y_i = \alpha + f(x_i) + \epsilon_i, \quad i = 1, \dots, n,$$

where the errors are iid, $\epsilon_i \sim N(0, \sigma_\epsilon^2)$.

- ▶ Want to estimate the true underlying curve $f(\cdot)$.

R-code

- ▶ Define formula and run model

```
> formula = y ~ f(x, model = "rw2", hyper = ...)
```

```
> result = inla(formula, data = data.frame(y, x))
```

- ▶ The **default prior** for the hyperparameter of `rw2`:

```
hyper = list(prec =  
             list(prior = "loggamma",  
                  param = c(1, 0.00005)))
```


Output

- ▶ `> summary(result)`
`> plot(result)`
- ▶ The mean effect of x :
`> result$summary.random$x$mean`
Note that this effect is constrained to sum to 0.
- ▶ Resulting fitted curve
`> result$summary.fitted.values$mean`

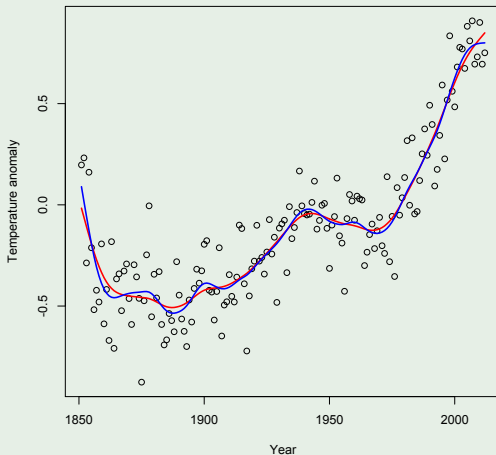
Estimated fit using the default prior

Example (Annual global temperature anomalies)



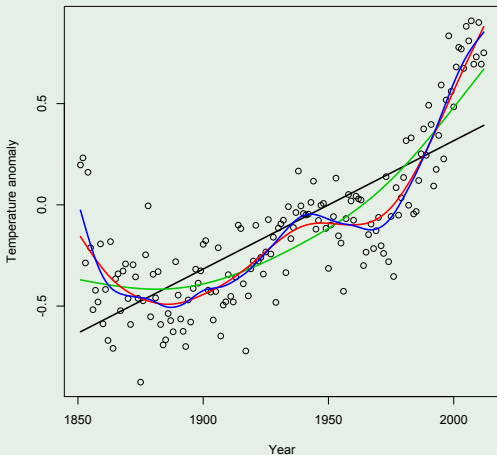
Estimated fit using R-INLA compared with smooth.spline

Example (Annual global temperature anomalies)



Using different priors for the precision

Example (Annual global temperature anomalies)



Default prior choices in R-INLA is about to change

The smoothness of the estimated curve is tuned by the prior for the precision of the `rw2` model.

New way to construct priors - More later!

- ▶ **Penalised complexity priors!**
- ▶ Smoothness is tuned in terms of an intuitive scaling parameter.

Outline

Background

Latent Gaussian model

Gibbs samplers and MCMC for LGMs

A case for approximate inference

R-INLA

Summary

Summary

- ▶ INLA is used to analyse a broad class of statistical models, named latent Gaussian models.
- ▶ Unified computational framework with three levels:
 - **Likelihood** for the observations.
 - **Latent field**, model dependency structures.
 - **Hyperparameters**, tune smoothness.
- ▶ Efficient and accurate. Easily available using R-INLA.