A short introduction to INLA* and R-INLA

* Integrated Nested Laplace Approximation

Thomas Opitz, BioSP, INRA Avignon

Workshop: Theory and practice of INLA and SPDE

November 7, 2018

Plan for this talk

- understand which models are suitable for estimation with INLA
- understand the basic mechanism of Integrated Nested Laplace Approximation
- learn how to run some simple models in `R-INLA` and how to conduct prediction

⇒ **general theoretical concepts**, and **practical implementation** in `R-INLA`

Some important resources for R-INLA with SPDE

http://www.r-inla.org/

- newsfeed
- discussion forum
- FAQ, examples and tutorials

A great book soon to come, already available for free online :

Advanced Spatial Modeling with Stochastic Partial Differential Equations Using R and INLA
by E.T. Krainski, V. Gómez-Rubio, H. Bakka, A. Lenzi, D. Castro-Camilo, D. Simpson, F. Lindgren and H. Rue
https://becarioprecario.bitbucket.io/spde-gitbook/

What kind of data/models can we fit with INLA ?

- **generalized additive (mixed) regression models**
  $\Rightarrow$ explain **observed response variable** through **covariates and random effects** :

$$y \sim \underbrace{\underbrace{\beta_0 + \beta_1 \times \text{covar.1} + \ldots + \beta_m \times \text{covar.m}}_{\textbf{fixed effects}} + \textbf{random effects}}_{\textbf{linear predictor}}$$

- large variety of **response distributions** for y
  - (log-)gaussian : (log) y =linear predictor+Gaussian error
  - Poisson : log(intensity) = log($\mathbb{E}y$) = linear predictor
  - generalized Pareto : log(α-quantile of y)= linear predictor

  and many more (gamma, skew normal, t, binomial), see `inla.list.models("likelihood")`
- **random effects** $\Rightarrow$ high flexibility :
  - model nonlinear trends with respect to space/time/covariates
  - model longitudinal effects, space/time dependence
  - structured (e.g., space/time dependence) or unstructured (e.g., measurement error)

  $\Rightarrow$ capture variability in response not explained by fixed effects and response distribution

  $\Rightarrow$ SPDE approach very useful for continuous random effects (Haakon's tutorial)

  see `inla.list.models("latent")`

Bayesian inference

INLA uses a **Bayesian framework** :

- we put **Gaussian prior distributions** on fixed effects β_j and random effects, which can be (relatively) uninformative or incorporate expert knowledge

  $\Rightarrow$ linear predictor is multivariate Gaussian

  $\Rightarrow$ when we have only few data, prior has strong influence

- we can estimate some **hyperparameters** controling
  - signal-to-noise ratio, smoothing $\Rightarrow$ precision parameters
  - range of dependence over space/time/..., e.g. range of Matérn correlation
  - shape of the response distribution, e.g. gamma shape, skew-normal

see `inla.list.models("prior")`

Bayesian hierarchical formulation of models

$$\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta}) \qquad \text{hyperparameters}$$

$$\boldsymbol{x} \mid \boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}(\boldsymbol{\theta})^{-1}) \qquad \text{latent Gaussian components}$$

$$\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta} \sim \prod_i \pi(y_i \mid \eta_i(\boldsymbol{x}), \boldsymbol{\theta}) \qquad \text{likelihood of observations}$$

- univariate **likelihood** $\pi(y_i \mid \eta_i(\boldsymbol{x}), \boldsymbol{\theta})$ for data
- **observation/projection matrix** $\boldsymbol{A}$ links latent Gaussian components to observations :

$$\boldsymbol{\eta}(\boldsymbol{x}) = (\eta_1(\boldsymbol{x}), \ldots, \eta_n(\boldsymbol{x})) = \boldsymbol{A}\boldsymbol{x} \quad \text{where} \quad y_i \sim \eta_i(\boldsymbol{x})$$

- $\boldsymbol{Q}(\boldsymbol{\theta})$ is called **precision matrix** (= inverse of variance-covariance matrix)
- R-INLA's speed, even in very high dimension, is based on
  - using **sparse matrices** for $\boldsymbol{Q}$ with mostly 0 entries
  - few non-0 entries in each row of $\boldsymbol{A}$
  - conditional independence of data $\boldsymbol{y}$ given the linear predictor $\boldsymbol{\eta}$ and $\boldsymbol{\theta}$ (⇒ sparsity is preserved during matrix computations in R-INLA)

Example : linear model with two covariates z_1, z_2

$y_i \mid (\boldsymbol{x}, \boldsymbol{\theta}) \sim \mathcal{N}(\eta_i(\boldsymbol{x}), \sigma^2), \qquad i = 1, \ldots, n$
linear predictor $\eta_i(\boldsymbol{x}) = \beta_0 + \beta_1 z_{1i} + \beta_2 z_{2i}$
latent Gaussian vector $\boldsymbol{x} = (\beta_0, \beta_1, \beta_2)$

- Gaussian likelihood $\pi(y \mid \eta_i(\boldsymbol{x}), \theta) = \frac{1}{\sqrt{2\pi \exp(-\theta)}} \exp\left(\frac{1}{2}\frac{y-\eta(\boldsymbol{x})}{\exp(-\theta)}\right)$

- $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}^{-1})$ with $\boldsymbol{Q} = \begin{pmatrix} \tau_0 & 0 & 0 \\ 0 & \tau_\beta & 0 \\ 0 & 0 & \tau_\beta \end{pmatrix}$
  for fixed effects, we usually fix the precision hyperparameters τ_0, τ_β to a very low value, e.g. 10^{-3} ($\Rightarrow$ non-informative priors)

- hyperparameter $\theta = \log(1/\sigma^2) \in \mathbb{R}$ (**log-precision** of Gaussian likelihood), for instance with an exponential prior with rate λ on σ such that

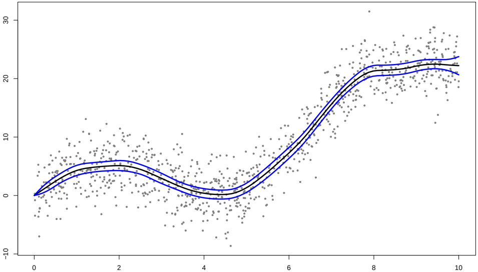
$$\pi(\theta) = \frac{\lambda}{2} \exp\left(-\lambda \exp(-\theta/2) - \theta/2\right), \quad \lambda > 0$$

(**Penalized Complexity Prior**, see later talks)

Example : estimating a nonlinear regression curve

1D SPDE models are random effects in `R-INLA`, useful to estimate continuous curves :

- spline-like model with different degrees of smoothness
- Bayesian framework allows using many knots and correlating spline coefficients using a Matérn-like correlation model
- two hyperparameters : range, precision



INLA fit of data (gray dots) with 1D spline prior for latent curve.
Pointwise 95% credible envelope in blue.

What random effect models are available in R-INLA ?

Output of `inla.list.models("latent")` :

ar	Auto-regressive model of order p (AR(p))
ar1	Auto-regressive model of order 1 (AR(1))
besag	The Besag area model (CAR-model)
generic	A generic model
iid	vector of independent effects
meb	Berkson measurement error model
ou	The Ornstein-Uhlenbeck process
rw1	Random walk of order 1
rw2	Random walk of order 2
rw2d	Thin-plate spline model
slm	Spatial lag model
spde2	A SPDE2 model

and many more...

Instances of such models can be linked together through a **group model** (AR, RW, exchangeable, ...).

Example : first-order random walks

First-order Gaussian random walks (`rw1` in `R-INLA`) are useful prior models for capturing nonlinear covariate effects or time trends.

They can be interpreted as very special and simple cases of 1D SPDE models.

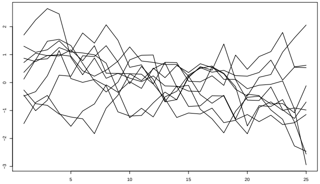
A `rw1` is **defined intrinsically over m classes** through its "innovations"

$$x_{i+1} - x_i \sim \mathcal{N}(0, 1/\tau_{RW}), \quad i = 2, \ldots, m.$$

It has one hyperparameter τ_{RW} measuring **precision**.

To make it identifiable, we have to impose a constraint such as $\sum_{i=1}^{m} x_i = 0$.

For easier interpretation, we may rescale to **marginal distribution** $\mathcal{N}(1/\tau_{RW})$ (i.e., control precision τ_{RW} averaged over all x_i).



10 realizations of a constrained `rw1` over $t = 1, \ldots, 25$, rescaled, $1/\tau_{RW} = 1$

Estimating Bayesian additive regression models

We obtain the **posterior estimations through Bayes' formula**.

⚠ Usually, we cannot calculate the **posterior estimations** in closed form :

- **posterior densities** $\pi(\theta_j \mid \boldsymbol{y})$, $\pi(x_k \mid \boldsymbol{y})$, $\pi(\eta_i \mid \boldsymbol{y})$
- **posterior mean estimates** $\mathbb{E}(\theta_j \mid \boldsymbol{y})$, $\mathbb{E}(x_k \mid \boldsymbol{y})$, $\mathbb{E}(h(\eta_i) \mid \boldsymbol{y})$

⚠ Moreover, the latent Gaussian components $\boldsymbol{x}$ are often very high-dimensional.

$\Rightarrow$ use **numerical approximation of complicated integrals :**

- **Markov-chain Monte-Carlo** : iteratively simulate and update values $\boldsymbol{x}^{(j)}, \boldsymbol{\theta}^{(j)}$
  $\Rightarrow$ generate a large representative sample of posterior distribution
  - theoretical convergence garantueed, but often too slow and unstable in practice
    ⚠ mixing (i.e., exploration of the space of parameters) can be too slow

- **Integrated Nested Laplace Approximation :** use astute numerical integration

Laplace approximation

Suppose that we want to calculate

$$\int_{-\infty}^{\infty} \exp(g(\boldsymbol{x}))\,\mathrm{d}\boldsymbol{x}$$

where $\boldsymbol{x}$ is a vector with many components, such as the latent Gaussian components.

If **g has maximum at x_0 and its values decrease fast and smoothly around x_0**, we can replace $g(\boldsymbol{x})$ by a curvature approximation around $\boldsymbol{x}_0$:

$$g(\boldsymbol{x}) \approx g(\boldsymbol{x}_0) + 0.5(\boldsymbol{x} - \boldsymbol{x}_0)'\boldsymbol{H}(g)(\boldsymbol{x}_0)(\boldsymbol{x} - \boldsymbol{x}_0)$$

with Hessian matrix H containing second partial derivatives of g at $\boldsymbol{x}_0$.

Using this approximation, the function to integrate becomes an (unnormalized) Gaussian density :

$$\int_{-\infty}^{\infty} \exp(g(\boldsymbol{x}))\,d\mathrm{x} \approx (2\pi)^{d/2}\,|\boldsymbol{H}(g)(\boldsymbol{x}_0)|^{-1/2}\exp(g(\boldsymbol{x}_0)).$$

In practice, it remains to determine $\boldsymbol{x}_0$ and $\boldsymbol{H}(g)(\boldsymbol{x}_0)$
$\Rightarrow$ iterative Newton–Raphson algorithm.

A schematic overview of how INLA works

All hyperparameters are fixed : skip 1 and 2.2. Exactly 1 hyperparameter : skip 1.2.

1. Calculate posterior density of hyperparameter θ_j :

$$\pi(\theta_j \mid \boldsymbol{y}) = \int \int \pi(\boldsymbol{\theta}, \boldsymbol{x} \mid \boldsymbol{y}) \, \mathrm{d}\boldsymbol{x} \mathrm{d}\boldsymbol{\theta}_{-j}$$

   1. $d\boldsymbol{x}$: use Laplace approximation for each configuration $\boldsymbol{\theta}$
   2. $\mathrm{d}\boldsymbol{\theta}_{-j}$: use numerical integration

2. Calculate posterior density of latent components x_i or predictors η_i :

$$\pi(x_i \mid \boldsymbol{y}) = \int \int \pi(\boldsymbol{x}, \boldsymbol{\theta} \mid \boldsymbol{y}) \mathrm{d}\boldsymbol{x}_{-i} \, \mathrm{d}\boldsymbol{\theta}$$

   1. $\mathrm{d}\boldsymbol{x}_{-i}$: use Laplace approximation (`strategy="laplace"`), but often too expensive
      - cheap but often less accurate : use conditional Gaussian from $d\boldsymbol{x}$ above (`strategy="gaussian"`)
      - `R-INLA` default is a simplified Laplace approximation (`strategy="simplified.laplace"`)
   2. $\mathrm{d}\boldsymbol{\theta}$: use numerical integration

`R-INLA` proposes three variants for numerical integration in 1.2 and 2.2 :

- grid of $\boldsymbol{\theta}$ configurations (`int.strategy="grid"`, most costly)
- central composite design (`int.strategy="ccd"`, the default)
- only the mode of $\pi(\boldsymbol{\theta} \mid \boldsymbol{y})$ (empirical Bayes, `int.strategy="eb"`)

Prediction with R-INLA

The output of `inla(...)` contains the posterior estimates of fitted values for observations, but we often want to **predict where no observations are available :**

- spatial interpolation
- temporal prediction
- fill missing data values

The simplest prediction strategy in R-INLA is to **add response data with `NA` values**.
⚠ covariates cannot be missing, only responses !

Another simple possibility :

1. "construct" the posterior mean of the linear predictor for new unobserved data with observation matrix A_{new} by transforming the posterior mean of latent Gaussian components,
$$\hat{\eta}_{\text{new}} = A_{\text{new}}\, \mathbb{E}(\boldsymbol{x} \mid \boldsymbol{y})$$
2. transform to the scale of the response if necessary

⚠ propagation of posterior uncertainty of $\boldsymbol{x}$ given $\boldsymbol{y}$ in Step 2 is disregarded !
e.g. with log-link, $\exp(\hat{\eta}_{\text{new}})$ will tend to underestimate the response values

Yet another possibility : generate a large sample from the posterior distribution of model components using `inla.posterior.sample(...)`, and then conduct Monte-Carlo calculation of predictions etc.

Practical examples

We will now treat several code examples ⇒ open `examples.R` in `RStudio`.

Example 1 : monthly temperatures in Nottingham 1920–1939

`nottem` dataset of `R`

Objective : model seasonal behavior

- use monthly random walk ($m = 12$) to capture seasonality in data
  $\Rightarrow$ 12 latent Gaussian variables $x_1, \ldots, x_{12}$ with

$$x_{t+1} - x_t \sim \mathcal{N}(0, 1/\tau_{RW})$$

- random walk values for January and December should be linked
  $\Rightarrow$ make the random walk **cyclic** by setting

$$x_1 - x_{12} \sim \mathcal{N}(0, 1/\tau_{RW})$$

Example 2 : Daily air quality data for New York, May-September 1973

`airquality` dataset of `R` m containing Ozone (with missing values), temperature, wind speed,...

Objective : predict missing values of Ozone

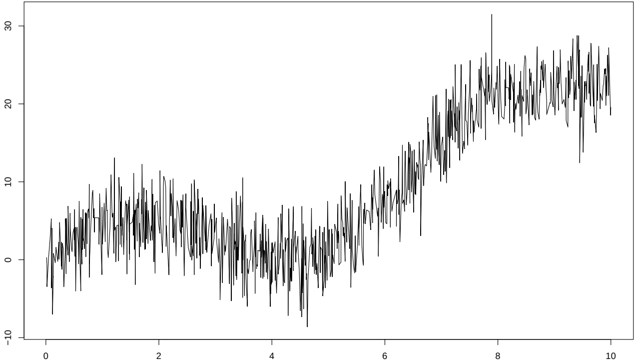
- **model 1** : temperature and wind as linear covariates (fixed effects)
- **model 2** : temperature and wind as nonlinear covariates
  $\Rightarrow$ use `rw1` over classes of covariate values
- **model 3** : temperature and wind (fixed effects), `ar1` for time dependence
- could combine two of the preceding models, e.g. model 2 and model 3

$\Rightarrow$ select best model through cross-validation of root-mean squared error
(or of any other good metric of predictive performance)

Example 3 : 1D SPDE for nonlinear regression curve

simulated dataset

Objective : fit a smooth nonlinear curve



1000 data points