INLA

as much as you can learn in 90 minutes

Tullia Padellini

tulliapadellini.github.io
tullia.padellini@uniroma1.it

SAPIENZA
UNIVERSITÀ DI ROMA

Dipartimento di Scienze Statistiche

Roma – 11 December 2019

Efficient (i.e. *fast*) and accurate computational tool for Bayesian Statistics.

- INLA - the method
  a **deterministic** algorithm to approximate the posterior distribution

- `R-INLA` - the implementation
  an `R` package to perform fit a large class of models in a Bayesian way

INLA -Integrated Nested Laplace Aproximation

you can have the cake and eat it too

- **it's fast**
  relies on numerical approximation and sparse matrices
- **it's accurate**
  empirically shows better performances than MCMC
- **it's flexible**
  can be used to fit any model formulated as a GAN
- **it is** (relatively) **easy to use**
  it's implemented as an `R` package

Motivation

isn't MCMC good enough?

Do we actually need **yet another way** to implement Bayesian Methods?

Yes if we think that MCMC methods are

- cumbersome to write
- slow

And this is very much true when dealing with Spatial Models

INLA vs MCMC take I

MCMC are cumbersome to write

JAGS code

```
model = function() {
  for(i in 1:N) {
    y[i] ~ dnorm(mu[i],tau)
    mu[i] <- alpha + beta*x[i]
  }
  alpha  ~ dnorm(0,0.001)
  beta  ~ dnorm(0,0.001)
  tau ~ dgamma(0.01,0.01)
}
params = c("alpha","beta","tau","mu")
jags(data=data,param=params,n.chains=3,n.iter=500
00, n.burnin=5000, model.file=model)
```

INLA Code

```
inla(y~x, family = c("gaussian"), data = data, co
ntrol.predictor=list(link=1))
```

INLA vs MCMC take II

MCMC are slow

n	rjags	r-inla
100	4.19	0.176
500	18.141	0.359
5000	381.573	2.787
25000	2203.679	13.27
100000	8873.836	52.787

INLA

INLA models

basically most of the models you have already seen

$$y|\theta,\psi \sim \pi(y;\theta,\psi) \qquad \text{Likelihood}$$
$$\theta|\psi \sim \pi(\theta;\psi) \qquad \text{Latent structure}$$
$$\psi \sim \pi(\psi) \qquad \text{Hyperprior}$$

`INLA` provides **numerical** approximations of the marginal posteriors

$$\pi(\theta_i|y) \qquad \pi(\psi_j|y)$$

INLA models

basically most of the models you have already seen

$$
\begin{aligned}
y|\theta,\psi &\sim \pi(y;\theta,\psi) && \text{Likelihood} \\
\theta|\psi &\sim N(\theta;0,\Sigma(\psi)) && \text{Latent structure} \\
\psi &\sim \pi(\psi) && \text{Hyperprior}
\end{aligned}
$$

`INLA` provides **numerical** approximations of the marginal posteriors

$$\pi(\theta_i|y) \qquad\qquad \pi(\psi_j|y)$$

INLA models

basically most of the models you have already seen

$$\begin{aligned} y|\theta,\psi &\sim \pi(y;\theta,\psi) && \text{Likelihood} \\ \theta|\psi &\sim N(\theta;0,\Sigma(\psi)) && \text{Latent structure} \\ \psi &\sim \pi(\psi) && \text{Hyperprior} \end{aligned}$$

`INLA` provides **numerical** approximations of the marginal posteriors

$$\pi(\theta_i|y) \qquad \pi(\psi_j|y)$$

Linear models naturally fall in the `INLA` framework when we consider $\theta = (\beta, f_1, f_2, \dots)$

$$y = k(\eta) + \epsilon \qquad \eta = x^t\beta + \sum_k f_k(z_k)$$

where $\sum_k f_k(z_k)$ can represent random effects, splines, anything you like.

Laplace approximation is based on the following two key idea:

$$f(x) = \exp[\log(f(x))]$$
$$g(x) = g(x^*) + g''(x^*)(x - x^*)^2 + \text{error} \approx g''(x^*)(x - x^*)^2$$

So that for every density f we have

$$f(x) \approx \exp[\log(f)''(x^*)(x - x^*)^2]$$

Laplace approximation is based on the following two key idea:

$$f(x) = \exp[\log(f(x))]$$
$$g(x) = g(x^*) + g''(x^*)(x - x^*)^2 + \text{error} \approx g''(x^*)(x - x^*)^2$$

So that for every density f we have

$$f(x) \approx \exp[\log(f)''(x^*)(x - x^*)^2]$$

Intuitively we can approximate any density f with a Gaussian by:

- matching the mode to the mean of the Gaussian, $\mu = x^*$
- setting the variance by looking at the curvaure at the mode $\sigma = -1/\log(f)''(x^*)$

Basic INLA assumptions

most verbose slide of the day

1. Each data point depends on only one of the elements in the latent Gaussian field θ, the linear predictor

2. The size of the hyperparameter vector ψ is small (say < 15)

3. The latent field θ, can be large but it is endowed with some conditional independence (Markov) properties so that the precision matrix $\Sigma^{-1}(\psi)$ is sparse.

4. The linear predictor depends linearly on the unknown smooth function of covariates.

5. The inferential interest lies in the univariate posterior marginals $\pi(\theta_i|y)$ and $\pi(\psi_j|y)$ rather than in the joint posterior $\pi(\theta, \psi|y)$.

$$\pi(\theta_i|y) = \int\int \pi(\theta,\psi|y) d\theta_{-i} d\psi = \int \pi(\theta_i|\psi,y)\pi(\psi|y) d\psi$$

$$\pi(\theta_i|y) = \int\int \pi(\theta, \psi|y) d\theta_{-i} d\psi = \int \pi(\theta_i|\psi, y)\pi(\psi|y) d\psi$$

$$\widehat{\pi}(\theta_i|y) = \sum_k \widehat{\pi}(\theta_i|\psi^{(k)}, y)\widehat{\pi}(\psi^{(k)}|y)\Delta^{(k)}$$

- Approximate $\pi(\psi|y)$ and $\pi(\theta_i|\psi, y)$ through Laplace Approximation

INLA

it's time for the formulas

$$\pi(\theta_i|y) = \int\int \pi(\theta,\psi|y)d\theta_{-i}d\psi = \int \pi(\theta_i|\psi,y)\pi(\psi|y)d\psi$$

$$\widehat{\pi}(\theta_i|y) = \sum_k \widehat{\pi}(\theta_i|\psi^{(k)},y)\widehat{\pi}(\psi^{(k)}|y)\Delta^{(k)}$$

- Approximate $\pi(\psi|y)$ and $\pi(\theta_i|\psi,y)$ through Laplace Approximation
- Approximate the integrals over ψ with summations over a finite set of values $\psi^{(1)},\dots,\psi^{(K)}$

Back to our Basic INLA assumptions

still most verbose slide of the day

1. Each data point depends on only one of the elements in the latent Gaussian field θ, the linear predictor
2. The size of the hyperparameter vector ψ is small (say < 15)
3. The latent field θ, can be large but it is endowed with some conditional independence (Markov) properties so that the precision matrix $\Sigma^{-1}(\psi)$ is sparse.
4. The linear predictor depends linearly on the unknown smooth function of covariates.
5. The inferential interest lies in the univariate posterior marginals $\pi(\theta_i|y)$ and $\pi(\psi_j|y)$ rather than in the joint posterior $\pi(\theta, \psi|y)$.

Posterior of ψ

starting from the "deepest" level

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

Posterior of ψ

starting from the "deepest" level

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

Here comes the **Laplace approximation:**

Approximate $\pi(\theta|\psi, y)$ with a Gaussian $\widehat{\pi}_G(\theta|\psi, y) = N(\theta; \mu, Q^{-1})$ wherer

- μ is the mode of $\pi(\theta|\psi, y)$
- $-Q$ is the curvature of $\log[\pi(\theta|\psi, y)]$ at the mode μ

Posterior of ψ

starting from the "deepest" level

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

Here comes the **Laplace approximation:**

Approximate $\pi(\theta|\psi,y)$ with a Gaussian $\widehat{\pi}_G(\theta|\psi,y) = N(\theta;\mu,Q^{-1})$ wherer

- μ is the mode of $\pi(\theta|\psi,y)$
- $-Q$ is the curvature of $\log[\pi(\theta|\psi,y)]$ at the mode μ

$$\widehat{\pi}(\psi|y) = \frac{\pi(\theta,\psi|y)}{\widehat{\pi}_G(\theta|\psi,y)} \propto \frac{\pi(y|\theta,\psi)\pi(\theta|\psi)\pi(\psi)}{\widehat{\pi}_G(\theta|\psi,y)}$$

$$\pi(\theta_i|y) = \int\int \pi(\theta, \psi|y) d\theta_{-i} d\psi = \int \pi(\theta_i|\psi, y)\pi(\psi|y) d\psi$$

- Approximate $\pi(\psi|y)$ and $\pi(\theta_i|\psi, y)$ through Laplace Approximation
- Approximate the integrals over ψ with summations over a set of *carefully chosen* values $\psi^{(1)}, \ldots, \psi^{(K)}$

$$\widehat{\pi}(\theta_i|y) = \sum_k \widehat{\pi}(\theta_i|\psi^{(k)}, y)\widehat{\pi}(\psi^{(k)}|y)\Delta^{(k)}$$

Approximate the Posterior Latent Field

skipping all the details

$$\pi(\theta_i|\psi, y) = \frac{\pi(\theta|\psi, y)}{\pi(\theta_{-i}|\theta_i, \psi, y)}$$

Approximate the Posterior Latent Field

skipping all the details

$$\pi(\theta_i|\psi, y) = \frac{\pi(\theta|\psi, y)}{\pi(\theta_{-i}|\theta_i, \psi, y)} \propto \frac{\pi(y|\theta, \psi)\pi(\theta|\psi)\pi(\psi)}{\pi(\theta_{-i}|\theta_i, \psi, y)}$$

- **Gaussian**: use the marginals of $\widehat{\pi}_G(\theta|\psi, y)$ computed before
- **Laplace approximation**: use a Gaussian approximation for the denominator $\pi(\theta_{-i}|\theta_i, \psi, y)$
- **Simplified Laplace approximation**: a mix of the two

Putting everything together

1. Explore the space of ψ through the approximation $\widehat{\pi}(\psi|y)$.

   Find the mode of $\widehat{\pi}(\psi|y)$
   Select $\psi^{(1)}, \dots, \psi^{(K)}$ in the area of high density of $\widehat{\pi}(\psi|y)$

2. Compute $\widehat{\pi}(\psi^{(k)}|y)$ for each $\psi^{(1)}, \dots, \psi^{(K)}$

3. Compute $\widehat{\pi}(\theta_i|\psi^{(k)}, y)$ for each $\psi^{(1)}, \dots, \psi^{(K)}$

4. Approximate $\pi(\theta_i|y)$ as

$$\widehat{\pi}(\theta_i|y) = \sum_k \widehat{\pi}(\theta_i|\psi^{(k)}, y)\widehat{\pi}(\psi^{(k)}|y)\Delta^{(k)}$$

R-INLA

Installation

it is non-trivial already

INLA is not on CRAN, so you need to specify the repository when you install it:

```
install.packages("INLA",
                 repos = "https://inla.r-inla-download.org/R/stable",
                 dep = TRUE)
```

`INLA` gets constant updating - check your version

Setting up the model

building blocks of the `inla` call

The generic `inla` call is structured as follows:

```
inla(formula, data, family)
```

- `formula`: formula object that specifies the linear predictor
- `data`: data frame with the data
- `family`: string that indicate the likelihood family (default is Gaussian)

The basic formulation of a linear regression model is almost the same as the canonical `lm` function:

```
library(INLA)

data(iris)

mod1  = inla(Petal.Length ~ 1 + Petal.Width, data = iris)

mod1_lm = lm(Petal.Length ~ 1 + Petal.Width, data = iris)
```

The `formula` argument

how to specify the model components

The formula object specifies the building blocks of the linear predictor

$$y = k(\eta) + \epsilon \qquad \eta = x^t\beta + \sum_k f_k(z_k)$$

`formula = y ~ x + f(id, model)`

The `f` terms contains random effect

- `id` name of the variable
- `model` name of the model of the random effect corresponding to `id`

Toy Example

most famous dataset ever

```
formula = Petal.Length ~ 1 + Petal.Width + f(Species, model = "iid")

mod2= inla(formula, data = iris)
```

NB: The list of all possible latent models can be found using:

```
names(inla.models()$latent)
inla.doc("ar1")
```

The `data` argument

how to input the observations to ‘inla‘

Data are typically provided through a `data.frame` (although named `list` can also be used).

- If the response is a factor it must be converted to $\{0, 1\}$ before calling `inla()`, as this conversion is not done automatic (as for example in `glm()`).

- If the covariate is binary it has to be converted to a `factor`, otherwise inla will treat it as numeric

- If we wish to predict the response variable for some observations, we need to specify the response variable of these observations as `NA`

The family argument is a string defining the likelihood of our model.

- each observation can have a different likelihood: vector of strings that indicate the likelihood family
- depending on the likelihood we are using, we may have additional arguments to provide to the `inla()` call

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

- we may have more than one link function corresponding to each family (as in the `logit` or `probit` case).
  `control.family=list(control.link=list(model="model"))`

NB: The list of all possible likelihoods can be found using:

```
names(inla.models()$link)
```

```r
data("Seeds")

res = inla(formula= r ~ x1 + x2, data = Seeds,
           family = "binomial", Ntrials = n,
           control.family = list(control.link=list(model = "logit")))
summary(res)
```

To see all available likelihood and links you can use:

```r
names(inla.models()$link)
names(inla.models()$likelihood)
```

Additional Arguments

- `control.compute`: list with the specification of several computing variables such as dic which is a Boolean variable indicating whether the DIC of the model should be computed

```
res = inla(Petal.Length ~ 1 + Petal.Width, data = iris,
           control.compute = list(dic = TRUE))
```

- `control.predictor`: list with the specification of several predictor variables such as link which is the link function of the model, and compute which is a Boolean variable that indicates whether the marginal densities for the linear predictor should be computed.

```
res = inla(Petal.Length ~ 1 + Petal.Width, data = iris,
          control.predictor = list(compute = TRUE))
```

Additional Arguments

- `control.compute`: list with the specification of several computing variables such as dic which is a Boolean variable indicating whether the DIC of the model should be computed

```
res = inla(Petal.Length ~ 1 + Petal.Width, data = iris,
           control.compute = list(dic = TRUE))
```

- `control.predictor`: list with the specification of several predictor variables such as link which is the link function of the model, and compute which is a Boolean variable that indicates whether the marginal densities for the linear predictor should be computed.

```
res = inla(Petal.Length ~ 1 + Petal.Width, data = iris,
          control.predictor = list(compute = TRUE))
```

Even more additional arguments

- `inla.emarginal()` and `inla.qmarginal()` calculate the expectation and quantiles, respectively, of the posterior marginals
- `inla.smarginal()` can be used to obtain a spline smoothing of the whole marginal
- `inla.tmarginal()` can be used to transform the marginals
- `inla.zmarginal()` provides summary statistics
- `inla.dmarginal()` computes the density at particular values