Bayesian hierarchical models

Practical 1: Implementing Bayesian models using R-INLA

Introduction

In this practical we will introduce the `R-INLA` software and demonstrate its functionality with some simple examples.

Preliminaries

We need the following packages

- `INLA` - Package to perform full Bayesian analysis of Latent Gaussian Models using Integrated Nested Laplace Approximations.

The `INLA` package is not hosted on CRAN, and we need to download it from another repository. We do this using the `install.packages()` function.

```
# Installing the R-INLA package
install.packages("INLA", repos="https://www.math.ntnu.no/inla/R/stable")
```

```
# Loading required packages
require(INLA)
```

Fitting a simple model in R-INLA

To demonstrate how to fit a simple model in R-INLA, we use the `Seeds` dataset. This dataset comprises of the number of seeds that were planted and germinated on each of 21 plates arranged according to a two by two factorial layout by seed and type of root extract. We can load this dataset, which is stored in `R`, using the `data()` function.

```
# Loading the trees dataset
data(Seeds)
```

More information on this dataset can be found by typing `?Seeds` into `R`. We can look at the structure of this dataset using the `str()` and `head()` functions.

```
# Viewing the structure of the dataset
str(Seeds)
'data.frame':   21 obs. of  5 variables:
 $ r    : int  10 23 23 26 17 5 53 55 32 46 ...
 $ n    : int  39 62 81 51 39 6 74 72 51 79 ...
 $ x1   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ x2   : int  0 0 0 0 0 1 1 1 1 1 ...
 $ plate: int  1 2 3 4 5 6 7 8 9 10 ...
```

```r
# Viewing the first six rows of 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
```

We can see that there are five variables:

- `r` - number of germinated seeds per plate
- `n` - number of total seeds per plate
- `x1` - seed type
- `x2` - root extracted
- `plate` - indicator for the plate

In this example, we are interested in the proportion of seeds that germinate at each plate location. Therefore, we use a binomial model,

$$r_i \sim Binomial(p_i, n_i),$$

where, p_i is the probability of a seed germinating on plate i. Here we will use the logit link function for modelling.

Model with Fixed Effects

We initally fit a very simple model without random effects. We include fixed effects for seed type and root extracted together with an interaction between them The model takes the form,

$$\text{logit}(p_i) = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_{12} x_{1i} x_{2i}$$

where p_i is the proportion of seeds germinating in plate i, x_{1i} is the type of seed in plate i and x_{2i} is the root extracted of seed i.

Specifying the model formula is done in the same way as specifying a model for `lm` or `glm`,

```r
# Model formula
formula1 <- r ~ 1 + x1 + x2 + x1*x2
```

We use the `inla()` function to fit this model using the `R-INLA`. This function takes a number of arguments. It requires the model formula, data, distribution of the response and the number of trials. We also include an option to calculate the Deviance Information Criterion (DIC), a measure of goodness-of-fit anagolous to the AIC and BIC.

```r
# Fitting the model in R-INLA
mod1 <- inla(formula1,
             data = Seeds,
             family = "binomial",
             Ntrials = n,
             control.compute=list(dic=TRUE))
```

Having fit the model, we can produce a summary using the `summary()` function.

```
# Creating summary of the fitted model
summary(mod1)
```

Activities

- Interpret the estimates of the parameters. Are the associations positive or negative?
- `R-INLA`creates a great deal of output from the model (you can see them using the `names()` function). Extract the different components and interpret them (you may find the help files useful) .

Model with fixed and random effects

We now extend the previous model to include a random effect to account for differences between plates. This model takes the form

$$\text{logit}(p_i) = \alpha_0 + \alpha_1 x_{1i} + \alpha_2 x_{2i} + \alpha_{12} x_{1i} x_{2i} + b_i$$

where p_i is the proportion of seeds germinating in plate i, x_{1i} is the type of seed in plate i, x_{2i} is the root extracted in plate i and b_i is the effect of plate i on germination.

To incorporate random effects, we use the `f()` function.

```
# Model formula
formula2 <- r ~ 1 + x1 + x2 + x1*x2 + f(plate, model='iid')
```

We assume that the random effects are iid.

We use the `inla()` function to fit this model.

```
# Fitting the model in R-INLA
mod2 <- inla(formula2,
             data = Seeds,
             family = "binomial",
             Ntrials = n,
             control.compute = list(dic=TRUE))
```

Once the model has been fit, we can summarise the model output using the `summary()` function.

```
# Creating summary of the fitted model
summary(mod2)
```

Activities

- Have the estimates of the model parameters changed?
- Looking at the values of the DIC from `mod1` and `mod2`, does it look like adding a random effect to the model has improved the fit?
- Adding a random effect to our model produces more objects in the output, have a look at the new objects related to the random effects in the model. *Which model has the best fit?

Fitting smoothing models in R-INLA

To demonstrate how to fit a smoothing model in R-INLA, we use the `trees` dataset. This dataset comprises the Volume, Girth and Height of 31 felled cherry trees and there is an interest in predicting Volume from the other two variables. We can load this dataset, which is stored in `R`, using the `data()` function.

```
# Loading the trees dataset
data(trees)
```

More information on this dataset can be found by typing `?trees` into `R`. We can look at the structure of this dataset using the `str()` and `head()` functions.

```
# Viewing the structure of the dataset
str(trees)
'data.frame':   31 obs. of  3 variables:
 $ Girth : num  8.3 8.6 8.8 10.5 10.7 10.8 11 11 11.1 11.2 ...
 $ Height: num  70 65 63 72 81 83 66 75 80 75 ...
 $ Volume: num  10.3 10.3 10.2 16.4 18.8 19.7 15.6 18.2 22.6 19.9 ...

# Viewing the first six rows of Seeds
head(trees)
  Girth Height Volume
1   8.3     70   10.3
2   8.6     65   10.3
3   8.8     63   10.2
4  10.5     72   16.4
5  10.7     81   18.8
6  10.8     83   19.7
```

We can see that there are three variables

- `Volume` - Volume in cubic ft
- `Height` - Height in ft
- `Girth` - Tree diameter in inches

In this example, we are interested in modelling the volume of the seeds that germinate at each plate location. To avoid non-negativity and skew of the response variable, we perform a log transformation. We use a Gaussian model to model the log of the volume of the trees

$$\log(Volume_i) \sim N(\mu_i, \sigma^2).$$

The relationship between the response and the explanatory variables may not be linear and hence we will model the log volume using smoothed functions of the height and girth of the trees. The model takes the form

$$\mu_i = \beta_0 + f_1(Girth_i) + f_2(Height_i)$$

We now specify the formula

```
# Model formula
formula3 <- logVolume ~ 1 +
  f(Height, model="rw2") + # Smoothed function of Girth
  f(Girth, model="rw2")   # Smoothed function of Height
```

we are fitting a smooth function to the two explanatory variables, each modelled as random walk of order 2.

```
# Creating a log volume variable
trees$logVolume <- log(trees$Volume)

# Fitting the model in `R-INLA`
mod3 <- inla(formula3,
             data = trees,
             family = "normal",
             control.compute = list(dic=TRUE))
```

Once the model has been fit, we can summarise the model output using the `summary()` function.

```
# Creating summary of the fitted model
summary(mod3)
```

Activities

- Fit a model which has fixed effects for height and girth.
- By comparing the DICs with those from your previous (smoothing) model, which model do you prefer?
- Is the extra complexity of the smoothing functions justified here?