*Journal of Statistical Software*

*January 2015, Volume 63, Issue 20.* http://www.jstatsoft.org/

# Spatial Data Analysis with R-INLA with Some Extensions

**Roger S. Bivand**
NHH Norwegian School of Economics

**Virgilio Gómez-Rubio**
Universidad de Castilla-La Mancha

**Håvard Rue**
Norwegian University for Science and Technology

**Abstract**

The integrated nested Laplace approximation (INLA) provides an interesting way of approximating the posterior marginals of a wide range of Bayesian hierarchical models. This approximation is based on conducting a Laplace approximation of certain functions and numerical integration is extensively used to integrate some of the models parameters out.

The R-**INLA** package offers an interface to INLA, providing a suitable framework for data analysis. Although the INLA methodology can deal with a large number of models, only the most relevant have been implemented within R-**INLA**. However, many other important models are not available for R-**INLA** yet.

In this paper we show how to fit a number of spatial models with R-**INLA**, including its interaction with other R packages for data analysis. Secondly, we describe a novel method to extend the number of latent models available for the model parameters. Our approach is based on conditioning on one or several model parameters and fit these conditioned models with R-**INLA**. Then these models are combined using Bayesian model averaging to provide the final approximations to the posterior marginals of the model.

Finally, we show some examples of the application of this technique in spatial statistics. It is worth noting that our approach can be extended to a number of other fields, and not only spatial statistics.

*Keywords*: INLA, spatial statistics, R.

## 1. Introduction

Bayesian inference has become very popular in spatial statistics in recent years. Part of this success is due to the availability of computation methods to tackle fitting of spatial models. Besag, York, and Mollié (1991) proposed in their seminal paper an appropriate way of fitting

a spatial model using Markov chain Monte Carlo methods. This model has been extensively used and extended to consider different types of fixed and random effects for spatial and spatio-temporal analysis.

In general, fitting these models has been possible because of the availability of different computational techniques, the most notable being Markov chain Monte Carlo (MCMC). For large models or big data sets, MCMC can be tedious and reaching the required number of samples can take a long time. Not to mention that autocorrelation may arise and that an increased number of iterations may be required.

Alternatively, the posterior distributions of the parameters may be approximated in some way. However, most models are highly multivariate and approximating the full posterior distribution may not be possible in practice. The integrated nested Laplace approximation (INLA, Rue, Martino, and Chopin 2009) focuses on the posterior marginals for latent Gaussian models. Although these models may seem rather restricted, they appear in a fair number of fields. This also means that INLA will be particularly useful when only marginal inference on the model parameters is needed.

The R-**INLA** package (Rue, Martino, Lindgren, Simpson, Riebler, and Krainski 2014; Lindgren and Rue 2015) for the R programming language (R Core Team 2014) provides an interface to INLA (a free-standing programme) so that models can be fitted using standard R commands. Results are readily available for plotting or further analysis. First of all, we describe how R-**INLA** can be used together with other R packages for spatial data analysis. It is often the case that spatial data are available in different formats that need to be loaded into R and some pre-processing is required. Also, once the results are available, it is helpful to explain how to display them on a map.

Although INLA is a general method to approximate the posterior marginals, R-**INLA** implements a number of popular latent models and prior distributions for the model parameters. It is, however, difficult to fit new models with INLA if these are based on other distributions not available in R-**INLA**. This may be an inconvenience when trying to develop new models as there is no easy way of extending R-**INLA** to fit other models without writing them into INLA itself.

This is why we also describe a way of extending the number of models that R-**INLA** can fit with little extra effort. First of all, we consider one (or more) parameters in our model so that, if they are fixed, the resulting model can be fitted with R-**INLA**. What we are doing here is, in fact, to fit a model conditioned on the assigned values to the parameters. Then, we can assign different values to these parameters and combine the resulting models in some way to obtain a fit of the original model. We have used Bayesian model averaging and numerical integration techniques to combine these models (Bivand, Gómez-Rubio, and Rue 2014b).

This paper is organized as follows. Section 2 describes the integrated nested Laplace approximation. In Section 3 the different latent models for spatial statistics are described. We describe how to extend R-**INLA** to fit new models in Section 4. Some examples are provided in Section 5. Finally, we discuss why our approach is relevant in Section 6.

## 2. Integrated nested Laplace approximation

Bayesian inference is based on computing the posterior distribution of a vector of model parameters $\mathbf{x}$ conditioned on the vector of observed data $\mathbf{y}$. Bayes' rule states that this

posterior distribution can be written down as

$$\pi(\mathbf{x}|\mathbf{y}) \propto \pi(\mathbf{y}|\mathbf{x})\pi(\mathbf{x}) \tag{1}$$

Here, $\pi(\mathbf{y}|\mathbf{x})$ is the likelihood of the model and $\pi(\mathbf{x})$ represents the prior distribution on the model parameters.

Usually, $\pi(\mathbf{x}|\mathbf{y})$ is a highly multivariate distribution and difficult to obtain. In particular, it is seldom possible to derive it in a closed form. For this reason, several computational approaches have been proposed to get approximations to it. MCMC is probably the most widely used family of computational approaches to estimate the posterior distribution.

The marginal distribution of parameter $x_i$ can be denoted by $\pi(x_i|\mathbf{y})$ and it can be easily derived from the full posterior by integrating out over the remaining set of parameters $\mathbf{x}_{-i}$.

Let us assume that we have a set of $n$ observations $\mathbf{y} = \{y_i\}_{i=1}^n$, whose distribution is of the exponential family. The mean of observation $i$ is $\mu_i$ and it can depend on a linear predictor $\eta_i$ via a link function. In turn, the linear predictor $\eta_i$ can be modelled as follows:

$$\eta_i = \alpha + \sum_{j=1}^{n_f} f^{(j)}(u_{ji}) + \sum_{k=1}^{n_\beta} \beta_k z_{ki} + \varepsilon_i \tag{2}$$

$\alpha$ is the intercept, $f^{(j)}$ are functions on a set of $n_f$ random effects on a vector of covariates $\mathbf{u}$, $\beta_k$ are coefficients on some covariates $\mathbf{z}$ and $\varepsilon_i$ are error terms. Hence, the vector of latent effects is $\mathbf{x} = \{\{\eta_i\}, \alpha, \{\beta_k\}, \ldots\}$. Note that given our particular interest in spatial models, terms $f^{(j)}(u_{ji})$ can be defined as to model spatial or spatio-temporal dependence.

$\mathbf{x}$ is modelled using a Gaussian distribution with zero mean and variance-covariance matrix $Q(\theta_1)$. Now, $\theta_1$ is a vector of hyperparameters. Furthermore, $\mathbf{x}$ is assumed to be a Gaussian Markov random field (GMRF, Rue and Held 2005). This means that $Q(\theta_1)$ will fulfil a number of Markov properties.

The distribution of observations $y_i$ will depend on the latent effects $\mathbf{x}$ and, possibly, a number of hyperparameters $\theta_2$. Taking the vector of hyperparameters $\theta = (\theta_1, \theta_2)$, observations $y_i$ will be independent of each other given $x_i$ and $\theta$ because of $\mathbf{x}$ being a GMRF.

Following Rue *et al.* (2009), the posterior distribution of the model latent effects $\mathbf{x}$ and hyperparameters $\theta$ can be written as

$$\pi(\mathbf{x}, \theta|\mathbf{y}) \propto \pi(\theta)\pi(\mathbf{x}|\theta) \prod_{i \in \mathcal{I}} \pi(y_i|x_i, \theta) \propto \tag{3}$$

$$\pi(\theta)|\mathbf{Q}(\theta)|^{1/2} \exp\{-\frac{1}{2}\mathbf{x}^T\mathbf{Q}(\theta)\mathbf{x} + \sum_{i \in \mathcal{I}} \log(\pi(y_i|x_i, \theta))\}$$

$\mathcal{I}$ represents an index of observed data (from 1 to $n$), $\mathbf{Q}(\theta)$ is a precision matrix on some hyperparameters $\theta$ and $\log(\pi(y_i|x_i, \theta))$ is the log-likelihood of observation $y_i$.

INLA allows different forms for the likelihood of the observations. This includes not only distributions from the exponential family but also mixtures of distributions. Also, INLA can handle observations with different likelihoods in the same model. Regarding the latent effects $\mathbf{x}$, different models can be used. We will describe some of these in more detail in Section 3.

The specification of the prior distributions $\pi(\theta)$ is also very flexible. These will often depend on the latent effect but, in principle, the most common distributions are available and the

user can define their own prior distribution in the R-**INLA** package (but we will return to this later).

Hence, we can write the marginals of the elements in $\mathbf{x}$ and $\theta$ (i.e., latent effects and hyperparameters) as

$$\pi(x_i|\mathbf{y}) = \int \pi(x_i|\theta,\mathbf{y})\pi(\theta|\mathbf{y})d\theta \tag{4}$$

and

$$\pi(\theta_j|\mathbf{y}) = \int \pi(\theta|\mathbf{y})d\theta_{-j} \tag{5}$$

In order to estimate the previous marginals, we need $\pi(\theta|\mathbf{y})$ or, alternatively, a convenient approximation that we will denote by $\tilde{\pi}(\theta|\mathbf{y})$. Initially, this approximation can be taken as

$$\tilde{\pi}(\theta|\mathbf{y}) \propto \left.\frac{\pi(\mathbf{x},\theta,\mathbf{y})}{\tilde{\pi}_G(\mathbf{x}|\theta,\mathbf{y})}\right|_{x=x^*(\theta)} \tag{6}$$

Here $\tilde{\pi}_G(\mathbf{x}|\theta,\mathbf{y})$ is a Gaussian approximation to the full conditional of $\mathbf{x}$ and $x^*(\theta)$ is the mode of the full conditional for a given value of $\theta$. Rue *et al.* (2009) take this approximation and use it to compute the marginal distribution of $x_i$ using numerical integration:

$$\tilde{\pi}(x_i|\mathbf{y}) = \sum_k \tilde{\pi}(x_i|\theta_k,\mathbf{y}) \times \tilde{\pi}(\theta_k|\mathbf{y}) \times \Delta_k \tag{7}$$

Here $\Delta_k$ are the weights associated with the ensemble of values $\theta_k$, defined on a multidimentional grid over the space of hyperparameters.

Note that in the previous equation it is important to have good approximations of $\pi(x_i|\theta_k,\mathbf{y})$. A Gaussian approximation $\tilde{\pi}_G(x_i|\theta_k,\mathbf{y})$, with mean $\mu_i(\theta)$ and variance $\sigma_i^2(\theta)$, may be a good starting point but a better approximation may be required in other cases. Rue *et al.* (2009) developed better approximations based on alternative approximation methods, such as the Laplace approximation. For example, they have used the Laplace approximation to obtain:

$$\tilde{\pi}_{LA}(x_i|\theta,\mathbf{y}) \propto \left.\frac{\pi(\mathbf{x},\theta,\mathbf{y})}{\tilde{\pi}_{GG}(\mathbf{x}_{-i}|x_i,\theta,\mathbf{y})}\right|_{\mathbf{x}_{-i}=\mathbf{x}^*_{-i}(x_i,\theta)} \tag{8}$$

$\tilde{\pi}_{GG}(\mathbf{x}_{-i}|x_i,\theta,\mathbf{y})$ is a Gaussian approximation to $\mathbf{x}_{-i}|x_i,\theta,\mathbf{y}$ around its mode $\mathbf{x}^*_{-i}(x_i,\theta)$.

Rue *et al.* (2009) develop a simplified Laplace approximation to improve $\tilde{\pi}_{LA}(x_i|\theta,\mathbf{y})$ using a series expansion of the Laplace approximation around $x_i$. This approximation is computationally less expensive, and it also corrects for location and skewness.

## 2.1. The R-INLA package

An interface to INLA has been provided as an R package called R-**INLA**, which can be downloaded from `http://www.r-inla.org/`, together with the free-standing external INLA programme. R-**INLA** provides a user model interface similar to the one used to fit generalized additive models (GAM) with function `gam()` in the **mgcv** package (Wood 2006). It can handle fixed effects, non-linear terms and random effects in a `formula` argument. The interface is flexible enough to allow for the specification of different priors and model fitting options. Non-linear terms and random effects are included in the formula as calls to the `f()` function.

The model is fitted with a call to function `inla()`, which will return the fitted model as an `inla` object. Note that, by default, only some results will be returned. These include the marginal distributions of the latent effects and hyperparameters, as well as summary statistics.

In addition to the posterior marginals, R-**INLA** can provide a number of additional quantities on the fitted model. For example, it can provide the log-marginal likelihood $\pi(\mathbf{y})$ which can be used for model selection. Other model selection criteria such as the DIC (Spiegelhalter, Best, Carlin, and Van der Linde 2002) and CPO (Held, Schödle, and Rue 2010) have also been implemented.

Furthermore, R-**INLA** includes a number of options to define the prior distributions for the parameters in the model. Well-known prior distributions are available and the user can define their own prior distributions as well.

In the next Section we describe different examples of the use of R-**INLA** for spatial statistics, in which we have included a detailed description on how `inla()` should be called.

# 3. Spatial models with INLA

As discussed in Section 2, spatial dependence can be included as part of the vector of latent effects $\mathbf{x}$. In principle, any number of random effects can be included in the model. In this Section, we will describe the different options available, depending on the type of problem. A full description of the models described here can be found in the R-**INLA** website at `http://www.r-inla.org/`, but we have included a summary. Blangiardo, Cameletti, Baio, and Rue (2013) and Gómez-Rubio, Bivand, and Rue (2014b) also discuss the different spatial models included in R-**INLA**.

First we will briefly introduce other papers describing the use of INLA and R-**INLA** for spatial statistics. Schrödle and Held (2010) describe the use of spatial and spatio-temporal models for disease mapping, including ecological regression. Schrödle and Held (2011) expand the number of spatio-temporal models that can be used with R-**INLA**, and show the use of setting linear constraints to make complex spatio-temporal effects identifiable. Schrödle, Held, Riebler, and Danuser (2011) show how to use spatio-temporal models for disease surveillance. Eidsvik, Finley, Banerjee, and Rue (2012) focus on the use of R-**INLA** for the analysis of large spatial datasets. Finally, Ruiz-Cardenas, Krainski, and Rue (2012) develop spatio-temporal dynamic models with R-**INLA**.

## 3.1. Analysis of lattice data

First of all, we will discuss the analysis of lattice data because this will establish the basis for other types of analyses. In the analysis of lattice data observations are grouped according to a set of areas, which usually represent some sort of administrative region (neighborhoods, municipalities, provinces, countries, etc.).

R-**INLA** includes a latent model for uncorrelated random effects. In this case, the random effects $u_i$ are modelled as

$$u_i \sim N(0, \tau_u) \tag{9}$$

where $\tau_u$ refers to the precision of the Gaussian distribution. It should be noted that R-**INLA** assigns a prior to $\log(\tau_u)$ which, by default, is a log-gamma distribution. Although this model

is not spatial, it can be combined with other spatial models. Using $\log(\tau_u)$ instead of simply $\tau_u$ provides some advantages as $\log(\tau_u)$ is not constrained to be positive. This is particularly useful when optimising to find the mode of $\log(\tau_u)$, for example.

In order to model spatial correlation, neighborhoods must be defined among the study areas. It is often considered that two areas are neighbors is they share a common boundary. Spatial autocorrelation is modelled using a Gaussian distribution with zero mean and a precision matrix that will model correlation between neighbors. Given that latent effects are a GMRF, we can define the variance-covariance matrix of the random effects as

$$\Sigma = \frac{1}{\tau} Q^{-1} \tag{10}$$

where $\tau$ is a precision parameter and matrix $Q$ encodes the spatial structure. Given that we are assuming a latent GMRF, this also means that matrix $Q$ will be defined such as element $Q_{ij}$ is zero if areas $i$ and $j$ are not neighbors. This means that $Q$ is often a very sparse matrix. See, for example, Rue and Held (2005) for details.

Available specifications for spatial dependence includes the intrinsic conditional autoregressive (CAR) specification (Besag *et al.* 1991). This will produce a $Q$ matrix in which element $Q_{ii}$ is $n_i$ (the number of neighbors of area $i$) and element $Q_{ij}$ (with $i \neq j$) is -1 if areas $i$ and $j$ are neighbors and 0 otherwise. This means that the spatial random effects $v_i$ are distributed as

$$v_i | v_j, \tau_v \sim N(\frac{1}{n_i} \sum_{i \sim j} v_j, \frac{1}{\tau_v n_i}) \quad i \neq j \tag{11}$$

$\tau_v$ is the conditional precision of the random effects. As in the previous model, R-**INLA** uses a log-gamma prior on $\log(\tau_v)$.

In addition, a proper version of this model is available as well, for which the spatial random effects are distributed as

$$v_i | v_j, \tau_v \sim N(\frac{1}{n_i + d} \sum_{i \sim j} v_j, \frac{1}{\tau_v (n_i + d)}) \quad i \neq j \tag{12}$$

$d$ is a positive quantity to make the distribution proper. By default, a log-gamma distribution is assigned to $\log(d)$.

A more general approach is obtained with the following precision matrix:

$$Q = (I - \frac{\rho}{\lambda_{max}} C) \tag{13}$$

Here $I$ is the identity matrix, $\rho$ a spatial autocorrelation parameter, $C$ an adjacency matrix and $\lambda_{max}$ the maximum eigenvalue of $C$. R-**INLA** assigns a Gaussian prior on $\log(\rho/(1-\rho))$. This specification ensures that $\rho$ takes values between 0 and 1.

In the following example we use the Boston housing data, which is described in Harrison and Rubinfeld (1978), to develop an example on several spatial models. This data set records median price for houses that were occupied by their owners plus some other relevant covariates (see Harrison and Rubinfeld 1978; Pace and Gilley 1997, for details). Data have been recorded at the tract level and the neighborhood structure of the tracts is also available, and it is available in the `boston` data set from the R package **spdep** (Bivand 2014). In addition, this data set is also available in a shapefile, which is the one we will use in this example. This

will provide a more general example on how to load external data into R to fit models with R-**INLA**.

`readShapePoly()`, from package **maptools** (Bivand and Lewin-Koh 2014), can be used to load vector data from a shapefile. Alternatively, `readOGR()`, from package **rgdal** (Bivand, Keitt, and Rowlingson 2014a), provides a more general data loading framework for vector data since it supports a wider range of formats. This is the one we have used to load the Boston data set:

```
R> library("rgdal")
R> boston <- readOGR(system.file("etc/shapes", package = "spdep")[1],
+    "boston_tracts")
```

Here, `readOGR()` takes the directory where the layer (shapefile) is located and the layer name, which in this case is the name of the shapefile, as arguments and return an object of type `SpatialPolygonsDataFrame`. This data object is used to store the tract boundaries plus the associated data (tract name and other variables).

Before fitting any spatial model, the neighborhood structure needs to be defined. A common criterion is to consider that two areas are neighbors if they share a common boundary. Function `poly2nb()` will take the tract boundaries and perform this operation to provide us with the adjacency structure of the Boston tracts as a `nb` object:

```
R> library("spdep")
R> bostonadj <- poly2nb(boston, queen = FALSE)
```

Here, we have also set `queen = FALSE` so that queen adjacency is not used, i.e., in order to consider two areas as neighbors more than one shared point is required. We have converted this into a binary matrix to be used with R-**INLA** using function `nb2mat()`. Furthermore, the adjacency matrix is converted into a sparse matrix of class `dgTMatrix` to reduce memory usage. This will be passed to function `f()` when defining the spatial model.

```
R> adj <- nb2mat(bostonadj, style = "B")
R> adj <- as(adj, "dgTMatrix")
```

A summary of some latent models implemented in R-**INLA**, and that can be used within the `f()` function, is available in Table 1. Note that this is not an exhaustive list and that a complete list of the available latent models can be obtained from the R-**INLA** documentation. We have also included a column showing whether these models are restricted to a regular grid. Also, detailed examples are available from the R-**INLA** website at http://www.r-inla.org/.

Fixed effects (including the intercept) in R-**INLA** have a Gaussian prior with fixed mean and precision, which are 0 and 0.01 (or 0 for the intercept) by default, respectively. These values can be changed using option `control.fixed` in the `inla()` call. `control.fixed` must take a named list of arguments, which are used to control how to handle the fixed effects in the model.

In this named list, `mean.intercept` and `prec.intercept` can be used to set the parameters of the Gaussian prior of the intercept, whilst `mean` and `prec` are the analogous parameters to define the priors for the other fixed effects. These can be a numeric value or another named list, using the names of fixed effects, to set different priors for different effects. Note that

| Name in `f()` | Model | Regular grid |
|---|---|---|
| `besag` | Intrinsic CAR | No |
| `besagproper` | Proper CAR | No |
| `bym` | Convolution model | No |
| `generic0` | $\Sigma = \frac{1}{\tau}Q^{-1}$ | No |
| `generic1` | $\Sigma = \frac{1}{\tau}(I_n - \frac{\rho}{\lambda_{max}}C)^{-1}$ | No |
| `rw2d` | 2-D random walk | Yes |
| `matern2d` | Matérn correlation | Yes |

Table 1: Summary of some latent models implemented in R-**INLA** for spatial statistics.

precisions in the fixed effects priors cannot be estimated as was the case with the different random effects presented before.

The model that we are fitting is:

$$y_i = \alpha + \beta X_i + v_i + \varepsilon_i \tag{14}$$

where $\alpha$ is the model intercept, $\beta$ a vector of coefficients of the covariates $X_i$, $v_i$ a random effect with an intrinsic CAR specification and $\varepsilon_i$ is random Gaussian error term.

As `f()` needs an area index which must have different values for different areas, this is first defined in variable `idx`.

```
R> library("INLA")
R> boston$idx <- 1:nrow(boston)
R> form <- log(CMEDV) ~ CRIM + ZN + INDUS + CHAS + I(NOX^2) + I(RM^2) +
+    AGE + log(DIS) + log(RAD) + TAX + PTRATIO + B + log(LSTAT) +
+    f(idx, model = "besag", graph = adj)
R> btdf <- as.data.frame(boston)
R> m1 <- inla(form, data = btdf, control.predictor = list(compute = TRUE))
```

Note how the call to `inla()` is similar to fitting other regression models with R with `glm()` or `gam()`. Furthermore, it is very easy to include spatial random effects with function `f()` in the `formula` passed to `inla()`. Finally, `control.predictor = list(compute = TRUE)` is used to compute summary statistics on the fitted values.

A summary of the model can be obtained as follows:

```
R> summary(m1)

Call:
"inla(formula = form, data = btdf, control.predictor = list(compute = TRUE))"

Time used:
 Pre-processing    Running inla Post-processing           Total
         0.5940          2.2684          0.1178          2.9802

Fixed effects:
```

```
               mean     sd 0.025quant 0.5quant 0.975quant    mode kld
(Intercept)  3.7798 0.1661     3.4536   3.7798     4.1058  3.7798   0
CRIM        -0.0073 0.0011    -0.0094  -0.0073    -0.0053 -0.0073   0
ZN           0.0003 0.0005    -0.0006   0.0003     0.0013  0.0003   0
INDUS       -0.0005 0.0024    -0.0051  -0.0005     0.0041 -0.0005   0
CHAS1       -0.0443 0.0299    -0.1030  -0.0443     0.0144 -0.0443   0
I(NOX^2)    -0.4209 0.1440    -0.7037  -0.4210    -0.1384 -0.4209   0
I(RM^2)      0.0101 0.0011     0.0080   0.0101     0.0122  0.0101   0
AGE         -0.0012 0.0005    -0.0022  -0.0012    -0.0003 -0.0012   0
log(DIS)    -0.1806 0.0719    -0.3217  -0.1806    -0.0395 -0.1806   0
log(RAD)     0.0483 0.0208     0.0076   0.0483     0.0891  0.0483   0
TAX         -0.0003 0.0001    -0.0005  -0.0003     0.0000 -0.0003   0
PTRATIO     -0.0162 0.0053    -0.0265  -0.0162    -0.0058 -0.0162   0
B            0.0006 0.0001     0.0003   0.0006     0.0008  0.0006   0
log(LSTAT)  -0.2434 0.0223    -0.2873  -0.2434    -0.1996 -0.2434   0

Random effects:
Name          Model
 idx   Besags ICAR model

Model hyperparameters:
                                          mean      sd
Precision for the Gaussian observations 1.626e+04 1.707e+04
Precision for idx                       1.222e+01 7.817e-01
                                        0.025quant 0.5quant
Precision for the Gaussian observations 7.582e+02  1.096e+04
Precision for idx                       1.074e+01  1.220e+01
                                        0.975quant mode
Precision for the Gaussian observations 6.180e+04  1.816e+03
Precision for idx                       1.381e+01  1.216e+01

Expected number of effective parameters(std dev): 501.85(5.348)
Number of equivalent replicates : 1.008

Marginal Likelihood:  -212.85
Posterior marginals for linear predictor and fitted values computed
```

The output includes summary statistics of the posterior marginals of the coefficients of the fixed effects plus the precisions of the error term and intrinsic CAR random effect. In addition, `kld` reports the Kullback-Leibler divergence between the Gaussian and the (simplified) Laplace approximation to the marginal posterior densities. This provides information about the accuracy of the Gaussian approximation.

The marginal likelihood of the model is also reported and it is computed by integrating all the model parameters out. Hence, it is not the *predictive* marginal likelihood and it can be used to perform model selection, for example. The effictive number of parameters, as defined in Spiegelhalter *et al.* (2002), and the associated number of equivalent replicates are also shown. See Martino and Rue (2010) for more details on the R-**INLA** output.

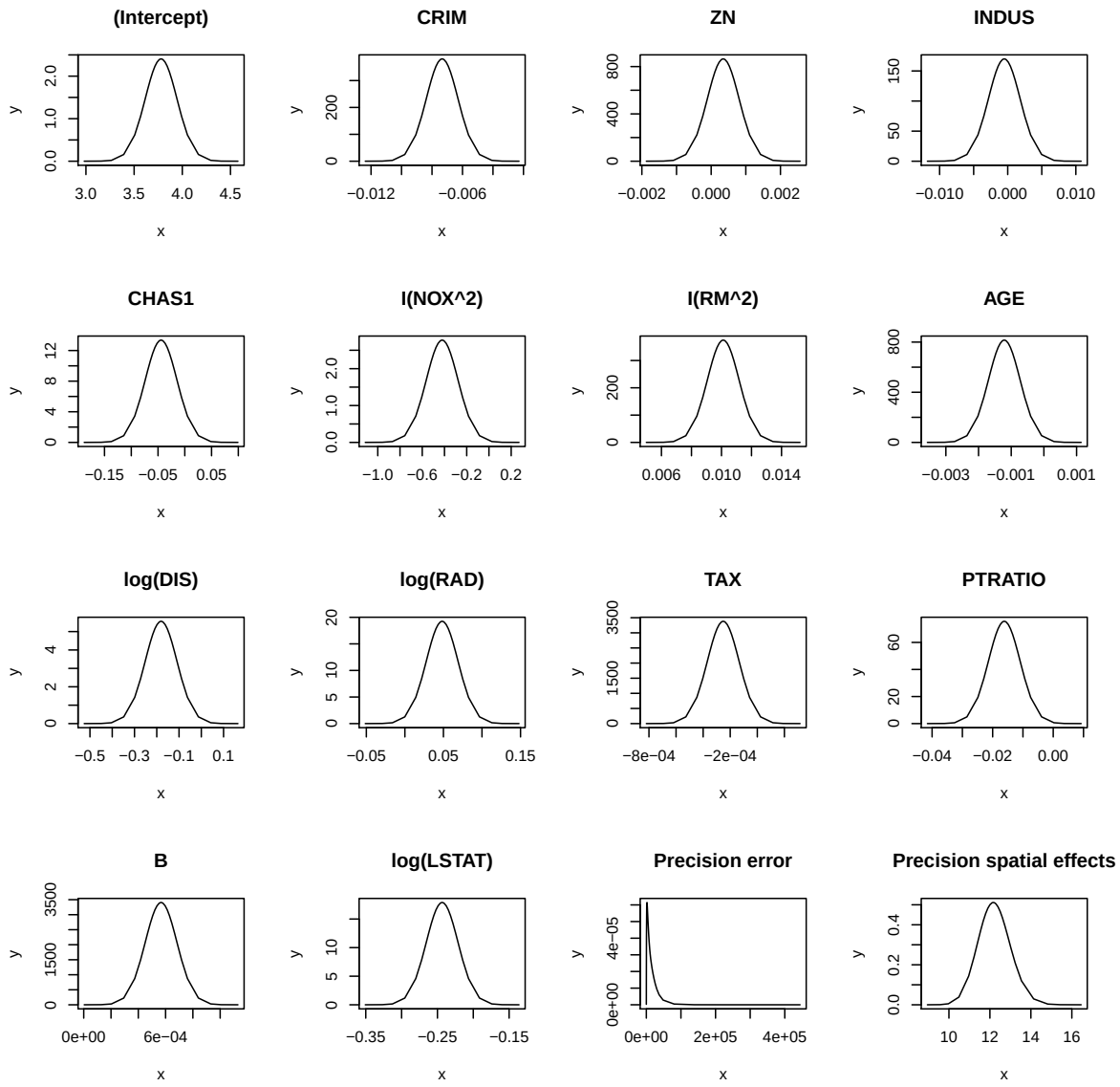


Figure 1: Marginals of the fixed effects, and the precisions of the error term and spatial random effects, Boston housing data.

Figure 1 shows the estimated marginals of the coefficients of the fixed effects and the precisions of the random effects in the model. These distributions can be used to compute summary statistics for the model parameters. In the previous R-**INLA** output these marginals have been used to compute the posterior mean, standard deviation, mode and some quantiles (0.025, 0.5 and 0.975).

Fitted values can be easily displayed in a map. First, we need to add all the required values to the `SpatialPolygonsDataFrame`:

```
R> boston$LOGCMEDV <- log(boston$CMEDV)
R> boston$FTDLOGCMEDV <- m1$summary.fitted[, "mean"]
```

Note that we will represent values in the log-scale. Next, we can use `spplot()` to display

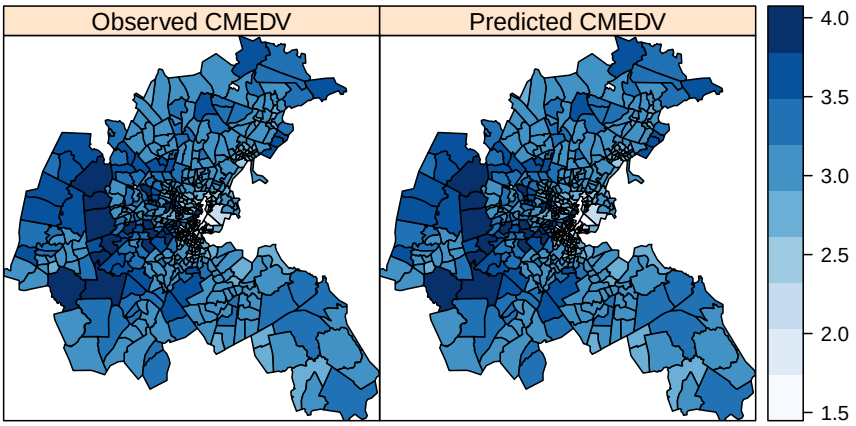


Figure 2: Observed and predicted median values, Boston housing data.

both the observed and the predicted values of house prices. In the following example, which can be seen in Figure 2, we have also used package **RColorBrewer** (Neuwirth 2014) to define a suitable color palette:

```
R> library("RColorBrewer")
R> spplot(boston, c("LOGCMEDV", "FTDLOGCMEDV"),
+    col.regions = brewer.pal(9, "Blues"), cuts = 8,
+    names.attr = c("Observed log-CMEDV", "Predicted log-CMEDV"))
```

To provide an alternative visualisation of the results, we have included a short example using function `qmap()` from the **ggmap** package (Kahle and Wickham 2013). First of all we will reproject our data to be WGS84. With `fortify()` the `boston` dataset is converted into a suitable format to be used when plotting and then the log median values are added to the new data.

```
R> bostonf <- spTransform(boston, CRS("+proj=longlat +datum=WGS84"))
R> library("ggmap")
R> bostonf <- fortify(bostonf, region = "TRACT")
R> idx <- match(bostonf$id, as.character(boston$TRACT))
R> bostonf$LOGCMEDV <- boston$LOGCMEDV[idx]
```

`qmap()` is based on the the grammar of graphics implemented in the **ggplot2** package (Wickham 2009). In the next example, `qmap()` is used to get satellite data from the Boston area, whilst `geom_polygon()` adds the boundaries:

```
R> qmap("boston", zoom = 10, maptype = "satellite") + geom_polygon(
+    data = bostonf, aes(x = long, y = lat, group = group, fill = LOGCMEDV),
+    colour = "white", alpha = 0.8, size = 0.3)
```

The resulting map can be seen in Figure 3.

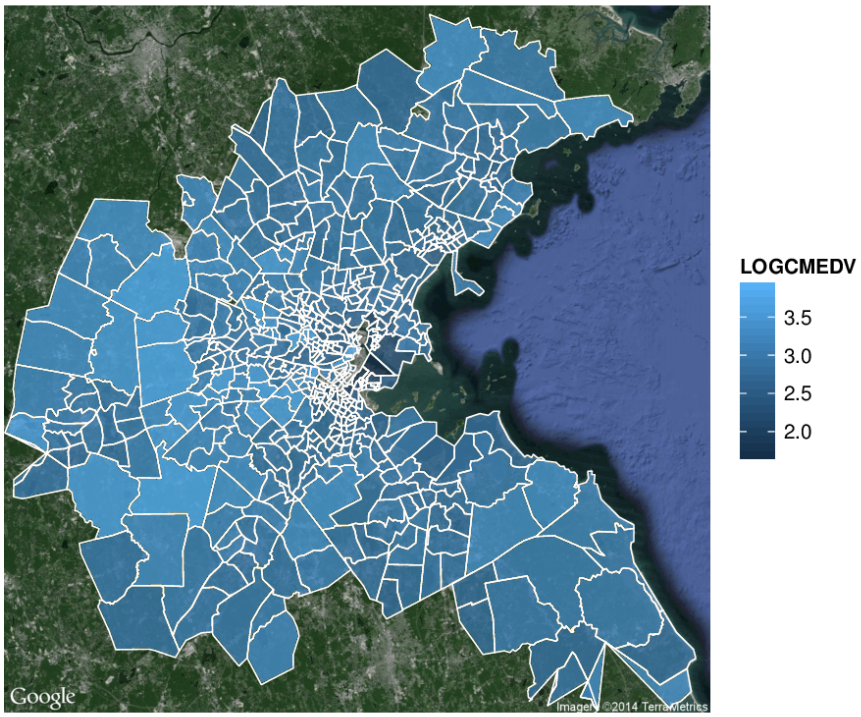


Figure 3: Display of the Boston housing data set using `ggmap` and Google Maps.

### 3.2. Point patterns

Point patterns are analyzed with INLA as the result of a counting process, i.e., points are not modelled directly but they are aggregated over a a grid of small squares. For this reason, the analysis of point patterns is conducted similarly to that of lattice data: counts are available for each square and these are assigned neighbors according to the adjacent squares. Then, counts can be smoothed using an appropriate non-linear term, such as spatial random effects. Hossain and Lawson (2009) compare different approximations to the analysis of point patterns, including methods that are based on discretisation of the study region.

In the following example we use the Japanese black pine data set from R package **spatstat** (Baddeley and Turner 2005). This data set records the location of Japanese black pine saplings in a square sampling in a natural forest. This example is reproduced from Gómez-Rubio *et al.* (2014b).

Hence, we first split the study area into smaller squares to create a grid of $10 \times 10$ squares.

```
R> library("spatstat")
R> data("japanesepines")
R> japd <- as.data.frame(japanesepines)
R> Nrow <- 10
R> Ncol <- 10
R> n <- Nrow * Ncol
R> grd <- GridTopology(cellcentre.offset = c(0.05, 0.05),
+    cellsize = c(1/Nrow, 1/Ncol), cells.dim = c(Nrow, Ncol))
```

After the creation of the grid, we have used function `over()` on the set of points and the newly defined squares to find how many points can be found in each square.

```
R> polygrdjap <- as(grd, "SpatialPolygons")
R> idxpp <- over(SpatialPoints(japd), polygrdjap)
R> japgrd <- SpatialGridDataFrame(grd, data.frame(Ntrees = rep(0, n)))
R> tidxpp <- table(idxpp)
R> japgrd$Ntrees[as.numeric(names(tidxpp))] <- tidxpp
```

Next, an index variable is built to create the spatial neighborhood structure to be passed to the `f()` function. Note that care must be taken as R and R-**INLA** may have a different ordering of the areas when defining the adjacency matrix.

```
R> japgrd$SPIDX <- 1:n
R> japnb <- poly2nb(polygrdjap, queen = FALSE, row.names = 1:100)
R> adjpine <- nb2mat(japnb, style = "B")
R> adjpine <- as(adjpine, "dgTMatrix")
```

Here we have avoided using a queen adjacency as this will consider as neighbors two areas which only share a corner.

Finally, we define the call to `inla()` using a `formula` which includes spatial random effects based on the grid of squares. In addition, we have set other options to compute the DIC, with `control.compute = list(dic = TRUE)`, and the marginals of the linear predictors, using `control.predictor = list(compute = TRUE)`. We have included the specification of the prior distributions of the log-precisions of unstructured and spatial random effects as well.

```
R> fpp <- Ntrees ~ 1 + f(japgrd$SPIDX, model = "bym", graph = adjpine,
+    hyper = list(prec.unstruct = list(prior = "loggamma",
+      param = c(0.001, 0.001)),
+    prec.spatial = list(prior = "loggamma", param = c(0.1, 0.1))))
R> japinlala <- inla(fpp, family = "poisson", data = as.data.frame(japgrd),
+    control.compute = list(dic = TRUE),
+    control.inla = list(tolerance = 1e-20, h = 1e-08),
+    control.predictor = list(compute = TRUE))
R> japgrd$INLALA <- japinlala$summary.fitted.values[, "mean"]
```

The former model is the one that we have employed with the Boston data set on an irregular lattice. Given that now we are considering a regular lattice it is also possible to use a two-dimensional random walk for spatial smoothing:

```
R> fpprw2d <- Ntrees ~ 1 + f(japgrd$SPIDX, model = "rw2d", nrow = 10,
+    ncol = 10, hyper = list(prec = list(prior = "loggamma",
+      param = c(0.001, 0.001))))
R> japinlalarw2d <- inla(fpprw2d, family = "poisson",
+    data = as.data.frame(japgrd), control.compute = list(dic = TRUE),
+    control.inla = list(tolerance = 1e-20, h = 1e-08),
+    control.predictor = list(compute = TRUE))
R> japgrd$INLALARW2D <- japinlalarw2d$summary.fitted.values[, "mean"]
```

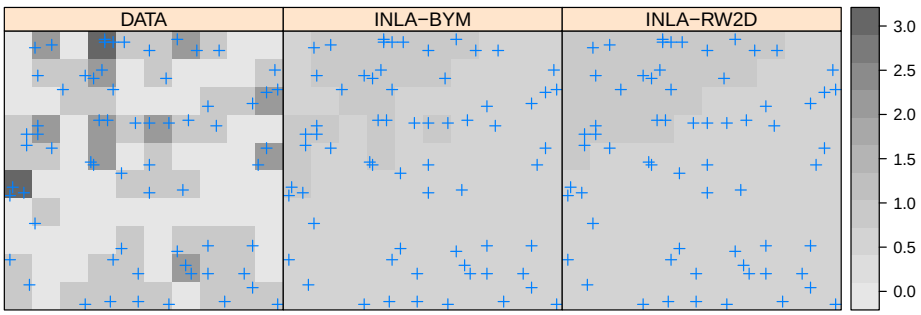


Figure 4: Estimation of the intensity of a point pattern with R-**INLA**, Japanese black pine dataset.

Figure 4 shows the original counts and the smoothed counts. Note that this is similar to estimating the intensity of an inhomogeneous point pattern using a smoothing method.

### 3.3. Geostatistics

R-**INLA** deals with geostatistical data on a regular grid. This means that observations need to be matched to the points in the grid and that those points with no observations attached are considered as missing values. Hence, this is somewhat similar to the analysis of lattice data and point patterns. However, R-**INLA** provides a number of options to build model-based geostatistical models (Diggle and Ribeiro 2007). First of all, different likelihoods can be used. Secondly, there are different options to define the spatial random effects. Although it is still possible to model spatial dependence in the grid of points using a CAR specification, R-**INLA** provides a two-dimensional Matérn covariance function.

This correlation allows, for example, the use of exponentially decaying functions such as

$$\Sigma_{ij} = \sigma^2 \exp(-d_{ij}/\varphi) \tag{15}$$

where $d_{ij}$ is the distance between points $i$ and $j$, and $\varphi$ is a parameter that controls the scale of the spatial dependence.

More recently, Lindgren, Rue, and Lindström (2011) follow a different approach based on a triangulation on the sampling points and the use of stochastic partial differential equations. Now, the spatial effects are defined as

$$u(s) = \sum_{k=1}^{n} \psi_k(s) w_k, \;\; s \in \mathbb{R}^2 \tag{16}$$

Here, $\{\psi_k(s)\}$ are a basis of functions and $w_k$ are associated weights. Weights are assumed to be Gaussian. The advantages of this approach for spatial statistics are fully described in Cameletti, Lindgren, Simpson, and Rue (2013).

In order to show how to fit geostatistical models with R-**INLA** we reproduce here an example from Gómez-Rubio *et al.* (2014b) based on the Rongelap data set (Diggle and Ribeiro 2007), which records radionuclide concentration at 157 different locations in Rongelap island. We have restricted the analysis to one of the clusters in the north-east part of the island because

observations need to be matched to a regular grid of points. For this analysis we have used R packages **geoR** (Ribeiro and Diggle 2001) and **geoRglm** (Christensen and Ribeiro 2002).

First of all, data are loaded and the data from the desired clusters are extracted from the original data set by checking that their coordinates are in the window $(-700, -500) \times (-1900, -1700)$.

```
R> library("geoR")
R> library("geoRglm")
R> data("rongelap")
R> rgldata <- as.data.frame(rongelap)
R> xy <- rongelap[[1]]
R> idx1 <- (xy[, 1] < -500 & xy[, 1] > -700 & xy[, 2] > -1900 &
+    xy[, 2] < -1700)
R> rgldata <- rgldata[idx1, ]
```

The next step is to define the grid topology for the grid that will be used to match these points to. The grid is defined to be of dimension $5 \times 5$.

```
R> Nrow <- 5
R> Ncol <- 5
R> n <- Nrow * Ncol
R> grdoffset <- c(min(rgldata$X1), min(rgldata$X2))
R> csize1 <- diff(range(rgldata$X1))/(Nrow - 1)
R> csize2 <- diff(range(rgldata$X2))/(Ncol - 1)
R> grd <- GridTopology(cellcentre.offset = grdoffset,
+    cellsize = c(csize1, csize2), cells.dim = c(Nrow, Ncol))
```

Data will be placed in a `SpatialGridDataFrame` (using the previously defined grid topology) and re-organized according to what R-**INLA** expects for this model (i.e., grid data stored by column). An index variable `IDX` is added to be used in `f()` when defining the model. However, R-**INLA** will rely on how the rows are ordered in the data passed to `inla()` when defining distances and adjacencies (i.e., the index variable ordering will not be considered).

```
R> inla2sp <- inla.lattice2node.mapping(Nrow, Ncol)[, Ncol:1]
R> inla2sp <- as.vector(inla2sp)
R> spgrd <- SpatialGridDataFrame(grd, as.data.frame(rgldata[inla2sp, ]))
R> spgrd$IDX <- 1:nrow(spgrd@data)
```

Next, we create a `SpatialPolygons` with the boundaries of the squares in the grid. This way, it is easy to match the data to the newly created grid using function `over()`.

```
R> polygrd <- as(grd, "SpatialPolygons")
R> dataidx <- over(SpatialPoints(as.matrix(rgldata[, 1:2])), polygrd)
```

It should be noted that radionuclide concentration is measured at each square by the average of the observations in the square, and this needs to be computed beforehand.

```
R> yag <- by(rgldata$data, dataidx, sum)
R> umag <- by(rgldata$units.m, dataidx, sum)
R> ratioag <- yag/umag
```

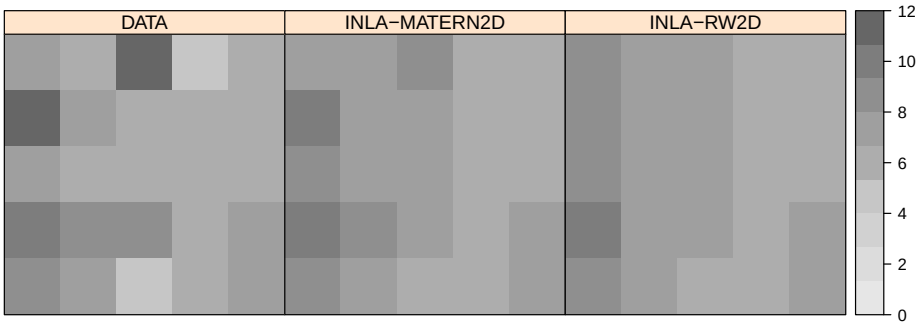


Figure 5: Observed and estimated radionuclide concentration in Rongelap island.

Then, a new column is added to the `SpatialGridDataFrame` with these averages. `NA` will be used for the squares with no data so that these values will be imputed from the model.

```
R> spgrd$ratioag <- NA
R> spgrd$ratioag[as.numeric(names(ratioag))] <- ratioag
```

Here we define a model with an intercept term and a random effect of the Matérn class. Note how we have fixed, for convenience, the value of the range and precision.

```
R> formula1 <- ratioag ~ 1 + f(spgrd$IDX, model = "matern2d", nrow = Nrow,
+    ncol = Ncol, hyper = list(range = list(initial = log(sqrt(8)/0.5),
+      fixed = TRUE), prec = list(initial = log(1), fixed = TRUE)))
R> rglinlala <- inla(formula1, family = "poisson",
+    control.predictor = list(compute = TRUE),
+    control.compute = list(dic = TRUE),
+    data = as.data.frame(spgrd))
R> spgrd$INLALA <- rglinlala$summary.fitted.values[, "mean"]
```

Similarly as in the point patterns example, here we have also used a two dimensional random walk for spatial smoothing.

```
R> formularw2d <- ratioag ~ 1 + f(spgrd$IDX, model = "rw2d", nrow = Nrow,
+    ncol = Ncol, hyper = list(prec = list(prior = "loggamma",
+      param = c(1, 1))))
R> rglinlalarw2d <- inla(formularw2d, family = "poisson",
+    control.predictor = list(compute = TRUE),
+    control.compute = list(dic = TRUE),
+    data = as.data.frame(spgrd))
R> spgrd$INLALARW2D <- rglinlalarw2d$summary.fitted.values[, "mean"]
```

Figure 5 shows the observed and estimated radionuclide concentration in Rongelap island. It can be seen how our model has spatially smoothed the observed values.

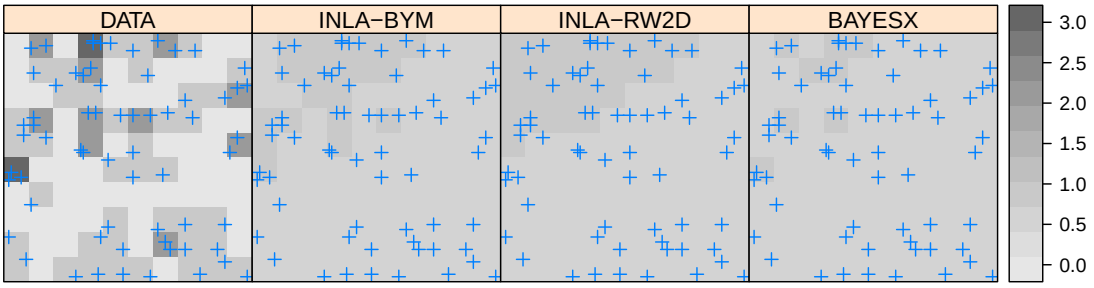


Figure 6: Estimation of the intensity of a point pattern with R-**INLA** and **BayesX**, Japanese black pine dataset.

### 3.4. R-INLA and other packages for Bayesian spatial modelling

R-**INLA** is not the only package for Bayesian spatial modelling. Bivand, Pebesma, and Gómez-Rubio (2013, Chapter 10) compare different packages for Bayesian modelling in the contect of disease mapping. We wil focus here in **R2BayesX** (Umlauf, Kneib, Lang, and Zeileis 2013; Umlauf, Adler, Kneib, Lang, and Zeileis 2015) because it provides a way to defining spatial models as R-**INLA**.

For example, in order to reproduce the example on the Japanese black pine data with **R2BayesX** we can do the following:

```
R> library("R2BayesX")
R> bayesxadj <- nb2gra(japnb)
R> japbyx <- bayesx(Ntrees ~ 1 + sx(SPIDX, bs = "re") +
+    sx(SPIDX, bs = "spatial", map = bayesxadj), family = "poisson",
+    data = as.data.frame(japgrd))
```

Function `nb2gra()` is used to convert our adjacency matrix into an object of class `gra`, which is used in **R2BayesX** to store adjacencies. `bayesx()` takes similar arguments as `inla()` and the model can be expressed using a `formula`, with `sx()` used to define the random effects. `sx(idx, bs = "re")` defines independent Gaussian random effects and the spatial random effects are defined in `sx(TRACT, bs = "spatial", map = bayesxadj)` using adjancency matrix defined in `bayesxadj`.

Retrieving the predicted data requires some care as they are reordered, but is is as simple as:

```
R> japgrd$BAYESX <- japbyx$fitted.values[
+    order(japbyx$bayesx.setup$order), "mu"]
```

Finally, we compare the fitted values obtained with R-**INLA** and **R2BayesX** in Figure 6. Note that differences appear not only because of the different models used but also because of the choice of prior distributions.

# 4. Extending R-INLA to fit new models

Although the current implementation of INLA in the R-**INLA** package provides a reasonable number of models for spatial dependence it may be the case that we need to include some other models. As it is now, this is not possible without adding to the code of the external INLA programme.

Bivand *et al.* (2014b) describe a simple way of extending INLA to use other latent models. In particular they focus on some latent models used in spatial econometrics that are not available as part of the R-**INLA** package at the moment. A new latent class has been added recently and it is described in Gómez-Rubio, Bivand, and Rue (2014a).

This approach is based on considering a model where one or several parameters have been fixed in a way that makes the conditioned model fittable with R-**INLA**. If we denote by $\rho$ the vector of parameters to fix and by $\hat{\rho}$ a specific set of fixed parameter values, the full posterior marginal could be written as

$$\pi(\mathbf{x}, \theta|\mathbf{y}, \hat{\rho}) \tag{17}$$

Taking this into account, it is clear that when conditioning on $\rho = \hat{\rho}$ R-**INLA** will give us an approximation to $\pi(x_i|\mathbf{y}, \hat{\rho})$ and $\pi(\theta_i|\mathbf{y}, \hat{\rho})$.

Note that the full posterior distribution can be obtained by integrating $\rho$ out, i.e.,

$$\pi(\mathbf{x}, \theta|\mathbf{y}) = \int \pi(\mathbf{x}, \theta|\mathbf{y}, \rho)\pi(\rho|\mathbf{y})d\rho \tag{18}$$

where $\pi(\rho|\mathbf{y})$ is the posterior distribution of $\rho$. Also, note that this can be written as

$$\pi(\rho|\mathbf{y}) \propto \pi(\mathbf{y}|\rho)\pi(\rho) \tag{19}$$

Here $\pi(\rho)$ is a prior distribution on $\rho$ and $\pi(\mathbf{y}|\rho)$ is the marginal likelihood of the model, which is reported by R-**INLA**. Hence, $\pi(\rho|\mathbf{y})$ can be estimated by re-scaling the expression in Equation 19.

The posterior distribution of $\rho$ can be estimated by defining a fine grid of values $S = \{\rho_i\}_{i=1}^r$ so that $\pi(\rho_i|\mathbf{y}),\ i = 1, \ldots, r$ are computed. Then $\pi(\rho|\mathbf{y})$ can be obtained by fitting and re-scaling a spline (or other non-linear function) to the previous values. Using simple numerical integration techniques we can obtain an approximation to $\pi(\mathbf{x}, \theta|\mathbf{y})$ as follows:

$$\pi(\mathbf{x}, \theta|\mathbf{y}) = \int \pi(\mathbf{x}, \theta|\mathbf{y}, \rho)\pi(\rho|\mathbf{y})d\rho \simeq \sum_{\rho_i \in S} \pi(\mathbf{x}, \theta|\mathbf{y}, \rho_i)\pi(\rho_i|\mathbf{y})\Delta_i \tag{20}$$

where $\Delta_i$ is the amplitude of the interval used in the discretisation of $\rho$.

Note that the previous expression can be regarded as a weighted average of the different models fitted after conditioning on different values of $\rho$.

From Equation 20 it is clear that we can obtain the following approximations to the posterior marginals of the individual latent parameters and hyperparameters:

$$\hat{\pi}(x_i|\mathbf{y}) = \sum_j \pi(x_i|\mathbf{y}, \rho_j)w_j \tag{21}$$

$$\hat{\pi}(\theta_i|\mathbf{y}) = \sum_j \pi(\theta_i|\mathbf{y}, \rho_j)w_j \tag{22}$$

$w_j$ is a weight associated with $\rho_j$ as follows:

$$w_j = \pi(\rho_j|\mathbf{y})\Delta_j \tag{23}$$

This is like carrying out Bayesian model averaging (Hoeting, Madigan, Raftery, and Volinsky 1999) on the different conditioned models fitted with R-**INLA**. Altogether, this provides a way of combining simpler models to obtain our desired model. In Section 5 we show how to apply these ideas to different models in spatial statistics.

Note that this approach can be easily extended to the case of $\rho$ being a discrete random variable.

### 4.1. Implementation

We have implemented this approach in an **R** package called **INLABMA**, available from CRAN. The package includes some general functions to conduct Bayesian model averaging of models fitted with INLA. In addition, we have included some wrapper functions to fit the models described in Section 5.

## 5. Examples

### 5.1. Leroux model

Leroux, Lei, and Breslow (1999) propose a model for the analysis of spatial data in a lattice which is similar to the one by Besag *et al.* (1991), in the sense that they split variation according to spatial and non-spatial patterns. Rather than including the spatial and non-spatial random effect as a sum in the linear term they consider a single random effect as follows:

$$u \sim \text{MVN}(0, \Sigma);\ \Sigma = \sigma^2((1-\lambda)I_n + \lambda M)^{-1} \tag{24}$$

Here $M$ is the precision matrix of a process with spatial structure and we will take that of an intrinsic CAR specification. Hence, the precision matrix is, in a sense, a mixture of the precisions of a non-spatial and a spatial one. $\lambda$ controls how strong the spatial structure is. For $\lambda = 1$ the effect is entirely spatial whilst for $\lambda = 0$ there is no spatial dependence.

In principle, this is not a model that R-**INLA** can fit. However, if $\lambda$ is fixed, then the random effects are Gaussian with a known structure for the variance-covariance matrix which can be fitted using a `generic0` latent model.

*Boston housing data*

Here we revisit the Boston housing data to fit the Leroux *et al.* model. First of all, it is worth mentioning that the model needs a wrapper function to be fitted for a given value of the spatial parameter $\lambda$. This wrapper function is included in the R package R-**INLA** and it is based on the `generic0` latent model available in R-**INLA**. Once $\lambda$ is fixed the model can be easily fitted with R-**INLA**, as the latent effect is a multivariate Gaussian random effect with zero mean and precision matrix as in Equation 24. We repeat this procedure for different values of $\lambda$ to obtain a list of fitted models to be combined later.

Hence, we have written a simple wrapper function which is included in package **INLABMA** (Gómez-Rubio and Bivand 2014):

```
R> library("INLABMA")
R> leroux.inla

function (formula, d, W, lambda, improve = TRUE, fhyper = NULL, ...)
{
    W2 <- diag(apply(W, 1, sum)) - W
    Q <- (1 - lambda) * diag(nrow(W)) + lambda * W2
    assign("Q", Q, environment(formula))
    if (is.null(fhyper)) {
        formula <- update(formula, . ~ . + f(idx, model = "generic0",
            Cmatrix = Q))
    }
    else {
        formula <- update(formula, . ~ . + f(idx, model = "generic0",
            Cmatrix = Q, hyper = fhyper))
    }
    res <- INLA::inla(formula, data = d, ...)
    if (improve)
        res <- INLA::inla.rerun(res)
    res$logdet <- as.numeric(Matrix::determinant(Q)$modulus)
    res$mlik <- res$mlik + res$logdet/2
    return(res)
}
<environment: namespace:INLABMA>
```

In the previous code, the precision matrix `Q` is created using the adjacency matrix `W` and the value of $\lambda$. Then the `generic0` model is added to the formula with the fixed effects. Finally we correct the marginal log-likelihood $\pi(\mathbf{y}|\lambda)$ (conditioned on the value of $\lambda$) by adding half the log-determinant of $((1-\lambda)I_n + \lambda M)$. Note that, in principle, this is not needed to fit a single model and obtain the approximations to the posterior marginals as it is a constant. However, we are fitting and combining several models so we need to correct for this because this scaling factor will change with the value of $\lambda$. Argument `...` is used to pass any other options to `inla()`. This can be used to tune and set a number of other options.

Also, the adjacency matrix is taken from the data provided in the `boston` data set. Note that we will be using a binary adjacency matrix as the random effects have an intrinsic CAR specification:

```
R> boston.matB <- listw2mat(nb2listw(bostonadj, style = "B"))
R> bmspB <- as(boston.matB, "CsparseMatrix")
```

Function `inla.leroux` is used in the example below to compute the fitted models for the Leroux *et al.* model. In this case, we take $\lambda$ to be in the interval $(0.8, 0.99)$ after previous assessment on where $\pi(\lambda|\mathbf{y})$ has its mode. Also, we define a prior for the precision of the random effects in variable `fhyper`. The prior for the precision of the error term is defined in `errorhyper`. In addition, we have used `mclapply` to parallelize the computations on operating systems supporting forking (not Windows). Note that this is an advantage of fitting these conditioned models compared with standard MCMC methods.

| | Mean (INLA) | SD (INLA) | Mean (MCMC) | SD (MCMC) |
|---|---|---|---|---|
| Intercept | 3.8055 | 0.1770 | 3.8018 | 0.1651 |
| CRIM | −0.0076 | 0.0011 | −0.0077 | 0.0011 |
| ZN | 0.0003 | 0.0005 | 0.0003 | 0.0004 |
| INDUS | −0.0006 | 0.0023 | −0.0007 | 0.0023 |
| CHAS1 | −0.0338 | 0.0302 | −0.0342 | 0.0305 |
| I(NOX^2) | −0.4374 | 0.1412 | −0.4467 | 0.1380 |
| I(RM^2) | 0.0099 | 0.0011 | 0.0099 | 0.0011 |
| AGE | −0.0011 | 0.0005 | −0.0011 | 0.0005 |
| log(DIS) | −0.1663 | 0.0616 | −0.1624 | 0.0618 |
| log(RAD) | 0.0506 | 0.0205 | 0.0484 | 0.0220 |
| TAX | −0.0003 | 0.0001 | −0.0003 | 0.0001 |
| PTRATIO | −0.0164 | 0.0053 | −0.0164 | 0.0053 |
| B | 0.0006 | 0.0001 | 0.0006 | 0.0001 |
| log(LSTAT) | −0.2534 | 0.0227 | −0.2521 | 0.0221 |
| $\lambda$ | 0.9520 | 0.0315 | 0.9401 | 0.0342 |

Table 2: Point estimates of fixed effects and $\lambda$ using INLA and MCMC.

```
R> rlambda <- seq(0.8, 0.99, length.out = 20)
R> fhyper <- list(prec = list(prior = "loggamma", param = c(0.001, 0.001),
+    initial = log(1), fixed = FALSE))
R> errorhyper <- list(prec = list(prior = "loggamma",
+    param = c(0.001, 0.001), initial = log(1), fixed = FALSE))
R> form2 <- log(CMEDV) ~ CRIM + ZN + INDUS + CHAS + I(NOX^2) + I(RM^2) +
+    AGE + log(DIS) + log(RAD) + TAX + PTRATIO + B + log(LSTAT)
R> lerouxmodels <- mclapply(rlambda, function(lambda) {
+    leroux.inla(form2, d = as.data.frame(boston), W = bmspB,
+      lambda = lambda, fhyper = fhyper, improve = TRUE,
+      family = "gaussian", control.fixed = list(prec.intercept = 0.001,
+        prec = 0.001), control.family = list(hyper = errorhyper),
+      control.predictor = list(compute = TRUE),
+      control.compute = list(dic = TRUE, cpo = TRUE),
+      control.inla = list(print.joint.hyper = TRUE))
+  })
```

Following this, we need to combine the different models to obtain the final model by Bayesian Model Averaging. We will take a uniform prior on $\rho$ and set the third argument in the following function to $\log(1) = 0$. Note that another prior can be used here by giving the log-density of the prior at the different values of $\rho$.

```
R> bmaleroux <- INLABMA(lerouxmodels, rlambda, 0, impacts = FALSE)
```

`bmaleroux` is similar to the object returned by `inla()` and it includes the posterior marginals and summary statistics for $\lambda$ in a list element named `rho`. This provides summary statistics (mean, standard deviation and some quantiles) and the posterior marginal.

The same model can be fitted using package **CARBayes** (Lee 2013) as follows:

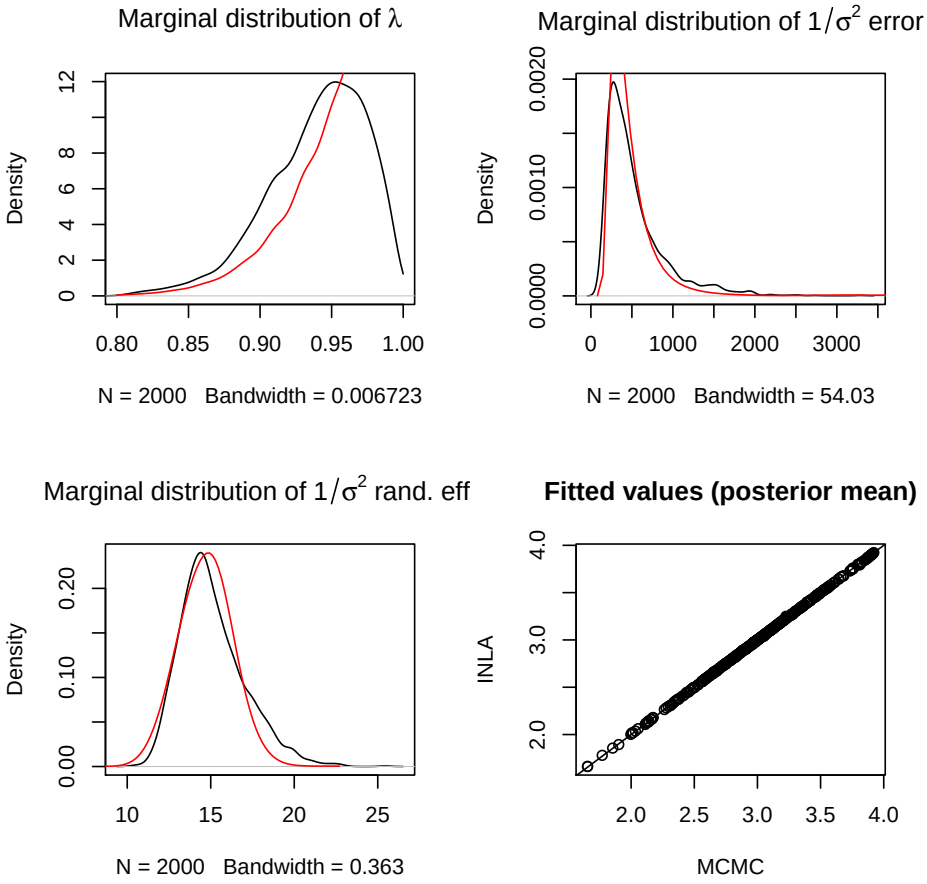


Figure 7: Comparisson between the posterior marginals of several parameters in Leroux *et al.*'s model and fitted values with INLA (red) and MCMC (black).

```
R> library("CARBayes")
R> lcarbayes <- S.CARleroux(form2,  data = boston, W = boston.matB,
+     family = "gaussian", burnin = 1000, n.sample = 11000, thin = 5)
```

Table 2 shows point estimates and standard deviations of the fixed effects and parameter $\lambda$ (bottom line) in the model. It is clear that there are no significant differences between the estimates computed with MCMC and our method. Furthermore, Figure 7 shows the marginal distribution of $\lambda$ and it shows how our estimate is very close to that provided by MCMC. Figure 7 also shows a good agreement between the posterior means of the fitted values.

While writting this paper, Lee and Mitchell (2013) have come up with an alternative way of fitting this model using R-**INLA** and a `generic1` latent model. In general, the results obtained for $\lambda$ with our approach are very similar to the ones obtained with theirs for the Boston housing data.

## 5.2. Spatial econometrics models

LeSage and Pace (2009) describe in detail a range of models used in spatial econometrics. These models have been developed to make the dependence among the observed values $y_i$

explicit. First of all, spatial dependence can be assumed on the error term (SEM model), so that we have a slightly different model:

$$y = \alpha + \beta X + u; u = \rho W u + \varepsilon \tag{25}$$

Here the error term $u$ is assumed to have spatial dependence. $\rho$ is a parameter that controls spatial autocorrelation, $\alpha$ is the intercept, $X$ a design matrix of covariates and $\beta$ a vector of associated coefficients. $\varepsilon$ is an error term which is Gaussian with zero mean and variance-covariance matrix $\Sigma$. Here, $\Sigma = \sigma^2 I_n$ with $I_n$ being a $n \times n$ identity matrix and $\sigma^2$ is the variance of the error term.

The adjacency matrix $W$ is often taken to be row-standardized (see, for example, Haining 2003) to ensure that $\rho$ is in the interval $(-1, 1)$. Also, when $\rho$ is equal to zero there is no spatial dependence.

This can be reformulated as

$$y = \alpha + \beta X + \varepsilon' \tag{26}$$

$\varepsilon'$ is now an error term with a Gaussian distribution with zero mean and variance-covariance matrix $\Sigma = \sigma^2((I_n - \rho W)(I_n - \rho W^\top))^{-1}$. Note that this variance-covariance encodes spatial dependence in a particular way and that this is often referred to as Simultaneous Autoregressive (SAR) specification (see, for example, Cressie 1993).

Alternatively, autocorrelation can be modelled explicitly so that the variable response $y$ depends on itself. This is the Spatial Lag Model (SLM model) and it can be defined as follows:

$$y = \alpha + \beta X + \rho W y + \varepsilon \tag{27}$$

This model can be reformulated as

$$y = (I_n - \rho W)^{-1}(\alpha + \beta X) + \varepsilon' \tag{28}$$

where $\varepsilon'$ is a Gaussian term with zero mean and variance-covariance matrix with a SAR specification.

In addition to the previous models, sometimes lagged covariates are added to include the effects of neighboring covariates. This is known as Spatial Durbin Model (SDM) and it is often expressed as

$$y = \rho W y + \alpha + \beta X + \gamma W X + \varepsilon \tag{29}$$

which leads to

$$y = (I_n - \rho W)^{-1}(\alpha + \beta X + \gamma W X) + \varepsilon' \tag{30}$$

Note that this is like our previous model in Equation 28 using an extended design matrix that includes both $X$ and $WX$.

These models cannot be fitted with R-**INLA** for two reasons. First of all, the SAR specification is not implemented, so we cannot consider it for our error terms. Furthermore, it is not possible to define a model where the linear predictor is multiplied by $(I_n - \rho W)^{-1}$.

Following Section 4, it should be noted that, for a fixed $\rho$, these models become standard linear models with a particular (but known) design matrix for the fixed effects and a multivariate Gaussian distribution for the error term with zero mean and variance-covariance matrix.

This model can be easily fitted using R-**INLA** for different values of $\rho$. Note that, according to Haining (2003), $\rho$ is constrained to be in the interval $(1/\lambda_{min}, 1/\lambda_{max})$, with $\lambda_{min}$ and $\lambda_{max}$

the minimum and maximum eigenvalues of the adjacency matrix $W$. For row standardized adjacency matrices we have that $\lambda_{max} = 1$ and then $\rho$ is lower than 1. Furthermore, we are often interested in positive spatial autocorrelation and, in practice, we will assume that $\rho$ is in the interval $[0, 1)$.

Finally, we have considered here the Gaussian case, but it is very easy to extend this model to consider other likelihoods. In particular, LeSage, Pace, Lam, Campanella, and Liu (2011) describe a Spatial Probit model where the outcome is whether an event occurred. Hence, the response and the linear predictors are linked via a probit function as follows:

$$y_i = \begin{cases} 1 & \text{if } y_i^* \geq 0 \\ 0 & \text{if } y_i^* < 0 \end{cases} \tag{31}$$

The previous approach can be used here as well. The same code will still work as we can define a different family and link function to be used when calling `inla()`.

*Boston housing data*

Here we revisit the Boston housing data to fit the three models on Spatial Econometrics previously described. First of all, it is worth mentioning that each models needs a wrapper function to be fitted for a given value of the spatial autocorrelation parameter $\rho$. These wrapper functions are included in R package R-**INLA**.

All functions are based on the `generic0` model available in R-**INLA**. This model implements a multivariate Gaussian random effect with zero mean and precision matrix $\tau Q$. Once $\rho$ is fixed, $Q$ is known and the model can be easily fitted with R-**INLA** for that value of $\rho$. We repeat this procedure for different values of $\rho$ to obtain a list of fitted models to be combined later.

A simple wrapper function can be defined as follows for the SEM model:

```
R> semwr.inla <- function(formula, d, W, rho, ...) {
+    IrhoW <- diag(nrow(W)) - rho * W
+    IrhoW2 <- t(IrhoW) %*% IrhoW
+    environment(formula) <- environment()
+    formula <- update(formula, . ~ . + f(idx, model = "generic0",
+      Cmatrix = IrhoW2))
+    res <- inla(formula, data = d, ...)
+    res$logdet <- as.numeric(determinant(IrhoW2)$modulus)
+    res$mlik <- res$mlik + res$logdet/2
+    return(res)
+  }
```

In the previous code, the precision matrix `IrhoW2` is created using the adjacency matrix and the value of $\rho$ and the `generic0` model is added to the formula after the fixed effects. As discussed with the Leroux model, the marginal log-likelihood $\pi(\mathbf{y}|\rho)$ (note that now we are conditioning on $\rho$) is corrected by adding half the log-determinat of $(I_n - \rho W^\top)(I_n - \rho W)$. Argument `...` is used, again, to pass any other options to `inla()`.

Similar wrapper functions can be written for the other models. The functions included in package **INLABMA** are similar, but include further options, to improve fitting or compute the impacts (see Bivand *et al.* 2014b, for details).

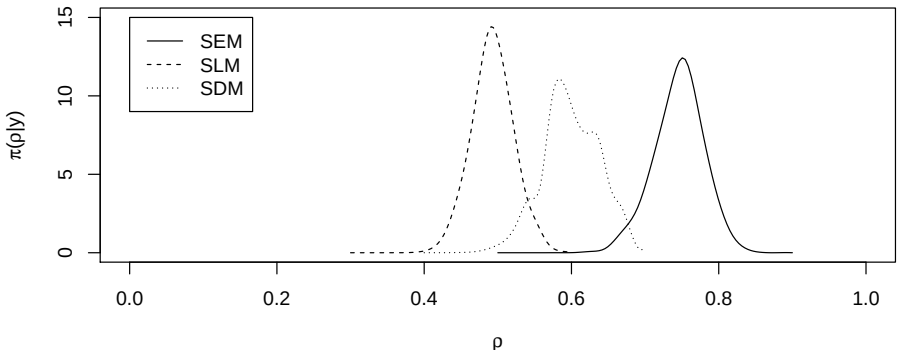


Figure 8: Posterior marginal of the spatial autocorrelation parameter $\rho$ for different spatial econometrics models, Boston housing data.

This is used in the following example to compute the fitted models for the SEM model. Note that here `zero.variance` is used to set to zero the additional error term which is added by default by R-**INLA**. Also, the adjacency matrix is taken from the data provided in the `boston` data set.

```
R> zero.variance <- list(prec = list(initial = 25, fixed = TRUE))
R> boston.mat <- nb2mat(bostonadj)
R> bmsp <- as(boston.mat, "CsparseMatrix")
R> boston$idx <- 1:nrow(boston)
R> fform <- log(CMEDV) ~ CRIM + ZN + INDUS + CHAS + I(NOX^2) + I(RM^2) +
+    AGE + log(DIS) + log(RAD) + TAX + PTRATIO + B + log(LSTAT)
```

Next, we define a fine grid to give values to $\rho$ and compute the different models under these values. Again, we have used `mclapply` to speed up computations.

```
R> rrho1 <- seq(0.5, 0.9, len = 20)
R> semmodels <- mclapply(rrho1, function(rho) {
+    sem.inla(fform, d = as.data.frame(boston), W = bmsp, rho = rho,
+      family = "gaussian", impacts = FALSE,
+      control.family = list(hyper = zero.variance),
+      control.predictor = list(compute = TRUE),
+      control.compute = list(dic = TRUE, cpo = TRUE),
+      control.inla = list(print.joint.hyper = TRUE))
+  })
```

Following this, we need to combine the different models to obtain the final model by Bayesian model averaging. As in the example using the Leroux model, we will take a uniform prior on $\rho$ and set the third argument in the following function to $\log(1) = 0$.

```
R> bmasem <- INLABMA(semmodels, rrho1, 0, impacts = FALSE)
```

`bmasem` is similar to the object returned by `inla()` and it includes the posterior marginals and summary statistics for $\rho$.

SLM and SDM models can be fitted using similar code. In the case of the SLM, first we compute the design matrix (of the fixed effects) to be used later when fitting the models. This will speed up the computations.

```
R> mmatrix <- model.matrix(fform, as.data.frame(boston))
R> rrho2 <- seq(0.3, 0.6, len = 20)
R> slmmodels <- mclapply(rrho2, function(rho) {
+    slm.inla(form, d = as.data.frame(boston), W = bmsp, rho = rho,
+      mmatrix = mmatrix, family = "gaussian", impacts = TRUE,
+      control.family = list(hyper = zero.variance),
+      control.predictor = list(compute = TRUE),
+      control.compute = list(dic = TRUE, cpo = TRUE),
+      control.inla = list(print.joint.hyper = TRUE))
+  })
R> bmaslm <- INLABMA(slmmodels, rrho2, 0, impacts = FALSE)
```

In the case of the SDM, it is very helpful to compute $X$ and $WX$ beforehand to reduce computation time, as this is common to all the fitted models regardless of the value of $\rho$. This can be later passed to wrapper function `sdm.inla`.

```
R> mmW <- bmsp %*% mmatrix[, -1]
R> mmatrixsdm <- cbind(mmatrix, as.matrix(mmW))
R> rrho3 <- seq(0.4, 0.7, len = 20)
R> sdmmodels <- mclapply(rrho3, function(rho) {
+    sdm.inla(form, d = as.data.frame(boston), W = bmsp, rho = rho,
+      mmatrix = mmatrixsdm, family = "gaussian", impacts = TRUE,
+      control.family = list(hyper = zero.variance),
+      control.predictor = list(compute = TRUE),
+      control.compute = list(dic = TRUE, cpo = TRUE),
+      control.inla = list(print.joint.hyper = TRUE))
+  })
R> bmasdm <- INLABMA(sdmmodels, rrho3, 0, impacts = FALSE)
```

As we have seen in the previous examples, combining the different values for the different values of $\rho$ provides accurate estimates of the posterior marginals for the model parameters. The estimates of the marginals of $\rho$ can be found in Figure 8.

## 6. Discussion

The integrated nested Laplace approximation has provided a new paradigm to model fitting in Bayesian analysis. By focusing on the marginals and providing an interesting computational approach, it has become a reasonable and faster alternative to MCMC. Furthermore, the R-**INLA** package makes model fitting in R a very simple task. Although R-**INLA** implements some key models, there are some others that have not been implemented yet. In addition, it is not easy to implement completely new models within R-**INLA**. Latent mixture models are an important example of models that cannot be fitted with R-**INLA**.

In this paper we have shown an innovative approach to model fitting with INLA and R-**INLA** which increases the number of latent models that can be fitted with INLA. We have been able to do so by fitting conditional models on some model parameters and combining the resulting models using simple methods for Bayesian model averaging. We have considered the case in which conditioning is carried out on a single parameter but our approach can be easily extended to the case of more than one parameter.

Given that the different models can be run in parallel, this is not a real computational burden. In our examples, 20 models seemed to be enough to obtain good approximations. Implementing new models is very simple and the wrapper functions included in the examples and the **INLABMA** package provide templates to start with.

Other computationally efficient approaches for Bayesian inference on spatial models include, for example, **RStan** (**Stan** Development Team 2013). It provides an efficient MCMC algorithm and provides a good number of options for Gaussian processes which can be used to implement some of the models described in this paper. It is worth mentioning that **RStan** models are not based on R code but on C++ code that is compiled once the main model is defined.

We have shown that this is a practical approach using different examples in spatial statistics. Similarly, complex spatio-temporal models could be fitted using a similar approach. However, it should be noted that our approach can be extended to other areas.

## Acknowledgments

We would like to thank Duncan Lee and Lola Ugarte for their help and comments on how to fit the Leroux model with R-**INLA**. Virgilio Gómez-Rubio acknowledges partial funding from project PPIC-2014-001, funded by Consejería de Educación, Cultura y Deportes (JCCM, Spain) and FEDER.

## References

Baddeley A, Turner R (2005). "**spatstat**: An R Package for Analyzing Spatial Point Patterns." *Journal of Statistical Software*, **12**(6), 1–42. URL http://www.jstatsoft.org/v12/i06/.

Besag J, York J, Mollié A (1991). "Bayesian Image Restoration, with Two Applications in Spatial Statistics." *Annals of the Institute of Statistical Mathematics*, **43**(1), 1–59.

Bivand R (2014). ***spdep**: Spatial Dependence: Weighting Schemes, Statistics and Models*. R package version 0.5-74, URL http://CRAN.R-project.org/package=spdep.

Bivand R, Keitt T, Rowlingson B (2014a). ***rgdal**: Bindings for the Geospatial Data Abstraction Library*. R package version 0.8-16, URL http://CRAN.R-project.org/package=rgdal.

Bivand R, Lewin-Koh N (2014). ***maptools**: Tools for Reading and Handling Spatial Objects*. R package version 0.8-30, URL http://CRAN.R-project.org/package=maptools.

Bivand RS, Gómez-Rubio V, Rue H (2014b). "Approximate Bayesian Inference for Spatial Econometrics Models." *Spatial Statistics*, **9**, 146–165.

Bivand RS, Pebesma E, Gómez-Rubio V (2013). *Applied Spatial Data Analysis with R*. 2nd edition. Springer-Verlag, New York.

Blangiardo M, Cameletti M, Baio G, Rue H (2013). "Spatial and Spatio-Temporal Models with R-**INLA**." *Spatial and Spatio-Temporal Epidemiology*, **4**, 33 – 49.

Cameletti M, Lindgren F, Simpson D, Rue H (2013). "Spatio-Temporal Modeling of Particulate Matter Concentration through the SPDE Approach." *Advances in Statistical Analysis*, **97**(2), 109–131.

Christensen OF, Ribeiro PJ (2002). "**geoRglm** – A Package for Generalised Linear Spatial Models." *R News*, **2**(2), 26–28. URL http://CRAN.R-project.org/doc/Rnews/.

Cressie NAC (1993). *Statistics for Spatial Data*. John Wiley & Sons, New York.

Diggle PJ, Ribeiro PJ (2007). *Model-Based Geostatistics*. Springer-Verlag, New York.

Eidsvik J, Finley AO, Banerjee S, Rue H (2012). "Approximate Bayesian Inference for Large Spatial Datasets Using Predictive Process Models." *Computational Statistics & Data Analysis*, **56**(6), 1362–1380.

Gómez-Rubio V, Bivand R (2014). ***INLABMA**: Bayesian Model Averaging with INLA*. R package version 0.1-5, URL http://CRAN.R-project.org/package=INLABMA.

Gómez-Rubio V, Bivand RS, Rue H (2014a). "Estimating Spatial Econometrics Models with Integrated Nested Laplace Approximation." *Submitted for publication*.

Gómez-Rubio V, Bivand RS, Rue H (2014b). "Spatial Models Using Laplace Approximation Methods." In MM Fischer, P Nijkamp (eds.), *Handbook of Regional Science*, pp. 1401–1417. Springer-Verlag, New York.

Haining R (2003). *Spatial Data Analysis: Theory and Practice*. Cambridge University Press, Cambridge.

Harrison D, Rubinfeld DL (1978). "Hedonic Housing Prices and the Demand for Clean Air." *Journal of Environmental Economics and Management*, **5**(1), 81–102.

Held L, Schödle B, Rue H (2010). "Posterior and Cross-Validatory Predictive Checks: A Comparison of MCMC and INLA." In T Kneib, G Tutz (eds.), *Statistical Modelling and Regression Structures – Festschrift in Honour of Ludwig Fahrmeir*, pp. 91–110. Springer-Verlag, Berlin.

Hoeting J, Madigan D, Raftery A, Volinsky C (1999). "Bayesian Model Averaging: A Tutorial." *Statistical Science*, **14**(4), 382–401.

Hossain MM, Lawson AB (2009). "Approximate Methods in Bayesian Point Process Spatial Models." *Computational Statistics & Data Analysis*, **53**(8), 2831–2842.

Kahle D, Wickham H (2013). "**ggmap**: Spatial Visualization with **ggplot2**." *The R Journal*, **5**(1), 144–162. URL http://journal.R-project.org/archive/2013-1/kahle-wickham.pdf.

Lee D (2013). "**CARBayes**: An R Package for Bayesian Spatial Modeling with Conditional Autoregressive Priors." *Journal of Statistical Software*, **55**(13), 1–24. URL http://www.jstatsoft.org/v55/i13/.

Lee D, Mitchell R (2013). "Locally Adaptive Spatial Smoothing Using Conditional Auto-Regressive Models." *Journal of the Royal Statistical Society C*, **62**(4), 593–608.

Leroux B, Lei X, Breslow N (1999). "Estimation of Disease Rates in Small Areas: A New Mixed Model for Spatial Dependence." In M Halloran, D Berry (eds.), *Statistical Models in Epidemiology, the Environment and Clinical Trials*, pp. 135–178. Springer-Verlag, New York.

LeSage J, Pace RK (2009). *Introduction to Spatial Econometrics*. Chapman & Hall/CRC, Boca Raton.

LeSage JP, Pace RK, Lam N, Campanella R, Liu X (2011). "New Orleans Business Recovery in the Aftermath of Hurricane Katrina." *Journal of the Royal Statistical Society A*, **174**(4), 1007–1027.

Lindgren F, Rue H (2015). "Bayesian Spatial Modelling with R-**INLA**." *Journal of Statistical Software*, **63**(19), 1–25. URL http://www.jstatsoft.org/v63/i19/.

Lindgren F, Rue H, Lindström J (2011). "An Explicit Link between Gaussian Fields and Gaussian Markov Random Fields: The SPDE Approach." *Journal of the Royal Statistical Society B*, **73**(4), 423–498.

Martino S, Rue H (2010). "Case Studies in Bayesian Computation Using **INLA**." In P Mantovan, P Secchi (eds.), *Complex Data Modeling and Computationally Intensive Statistical Methods*, Contributions to Statistics, pp. 99–114. Springer-Verlag, New York.

Neuwirth E (2014). ***RColorBrewer*: *ColorBrewer* Palettes**. R package version 1.1-2, URL http://CRAN.R-project.org/package=RColorBrewer.

Pace RK, Gilley OW (1997). "Using the Spatial Configuration of the Data to Improve Estimation." *Journal of the Real Estate Finance and Economics*, **14**(3), 333–340.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Ribeiro PJ, Diggle PJ (2001). "**geoR**: A Package for Geostatistical Analysis." *R News*, **1**(2), 15–18. URL http://CRAN.R-project.org/doc/Rnews/.

Rue H, Held L (2005). *Gaussian Markov Random Fields – Theory and Applications*. Chapman & Hall, New York.

Rue H, Martino S, Chopin N (2009). "Approximate Bayesian Inference for Latent Gaussian Models by Using Integrated Nested Laplace Approximations." *Journal of the Royal Statistical Society B*, **71**(2), 319–392.

Rue H, Martino S, Lindgren F, Simpson D, Riebler A, Krainski ET (2014). ***INLA*: *Functions Which Allow to Perform a Full Bayesian Analysis of Structured Additive Models Using Integrated Nested Laplace Approximaxion***. R package version 0.0-1406288160, URL http://www.r-inla.org/.

Ruiz-Cardenas R, Krainski ET, Rue H (2012). "Direct Fitting of Dynamic Models Using Integrated Nested Laplace Approximations – INLA." *Computational Statistics & Data Analysis*, **56**(6), 1808–1828.

Schrödle B, Held L (2010). "A Primer on Disease Mapping and Ecological Regression Using INLA." *Computational Statistics*, **26**(2), 241–258.

Schrödle B, Held L (2011). "Spatio-Temporal Disease Mapping Using INLA." *Environmetrics*, **22**(6), 725–734.

Schrödle B, Held L, Riebler A, Danuser J (2011). "Using Integrated Nested Laplace Approximations for the Evaluation of Veterinary Surveillance Data from Switzerland: A Case-Study." *Journal of the Royal Statistical Society C*, **60**(2), 261–279.

Spiegelhalter DJ, Best NG, Carlin BP, Van der Linde A (2002). "Bayesian Measures of Model Complexity and Fit." *Journal of the Royal Statistical Society B*, **64**(4), 583–616.

**Stan** Development Team (2013). "**Stan**: A C++ Library for Probability and Sampling, Version 1.3." URL http://mc-stan.org/.

Umlauf N, Adler D, Kneib T, Lang S, Zeileis A (2015). "Structured Additive Regression Models: An R Interface to **BayesX**." *Journal of Statistical Software*, **63**(21), 1–46. URL http://www.jstatsoft.org/v63/i21/.

Umlauf N, Kneib T, Lang S, Zeileis A (2013). ***R2BayesX**: Estimate Structured Additive Regression Models with BayesX*. R package version 0.3-1, URL http://CRAN.R-project.org/package=R2BayesX.

Wickham H (2009). ***ggplot2**: Elegant Graphics for Data Analysis*. Springer-Verlag, New York. URL http://had.co.nz/ggplot2/book/.

Wood SN (2006). *Generalized Additive Models: An Introduction with R*. Chapman & Hall/CRC, Boca Raton.

## Affiliation:

Roger S. Bivand
Department of Economics
NHH Norwegian School of Economics
Helleveien 30
N-5045 Bergen, Norway
E-mail: roger.bivand@nhh.no

Virgilio Gómez-Rubio
Department of Mathematics
School of Industrial Engineering
University of Castilla-La Mancha
02071 Albacete, Spain
E-mail: virgilio.gomez@uclm.es

Håvard Rue
Department of Mathematical Sciences
Norwegian University for Science and Technology
N-7491 Trondheim, Norway
E-mail: hrue@math.ntnu.no

*Journal of Statistical Software* http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 63, Issue 20 *Submitted:* 2013-06-05
January 2015 *Accepted:* 2014-09-01