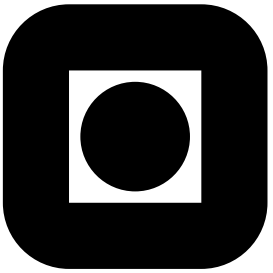
NORGES TEKNISK-NATURVITENSKAPELIGE
UNIVERSITET

Implementing Approximate Bayesian Inference using Integrated Nested Laplace Approximation: a manual for the `inla` program

by

Sara Martino & Håvard Rue

PREPRINT
STATISTICS NO. 2/2008



NORWEGIAN UNIVERSITY OF SCIENCE AND
TECHNOLOGY
TRONDHEIM, NORWAY

This preprint has URL `http://www.math.ntnu.no/preprint/statistics/2008/S2-2008.pdf`
Sara Martino has homepage: `http://www.math.ntnu.no/~martino`
E-mail: `martino@math.ntnu.no`
Address: Department of Mathematical Sciences, Norwegian University of Science and Technology, N-7491 Trondheim, Norway.

Implementing Approximate Bayesian Inference using Integrated Nested Laplace Approximation: a manual for the `inla` program

Sara Martino and Håvard Rue
Department of Mathematical Sciences
NTNU, Norway

January 2008

Abstract

This manual describes the `inla` program, a new instrument which allows the user to easily perform approximate Bayesian inference using integrated nested Laplace approximation (INLA). We describe the set of models which can be solved by the `inla` program and provide a series of worked out examples illustrating its usage in details. Appendix A contains a reference manual for the `inla` program.

This manual is for version 1.0-0 of the `inla` program.

1 Introduction

Integrated nested Laplace approximation (INLA) is a new approach to statistical inference for latent Gaussian Markov random field (GMRF) models introduced by Rue and Martino (2006) and Rue et al. (2007). It provides a fast, deterministic alternative to Markov chain Monte Carlo (MCMC) which, at the moment, is the standard tool for inference in such models. The main advantage of the INLA approach over MCMC is that it is much faster to compute; it gives answers in minutes and seconds where MCMC requires hours and days. The theory behind INLA is thoroughly described in Rue et al. (2007) and will not be repeated here.

In short, a latent GMRF model is a hierarchical model where, at the first stage we find a distributional assumption for the observables $\boldsymbol{y}$ usually assumed to be conditionally independent given some latent parameters $\boldsymbol{\eta}$ and, possibly, some additional parameters $\boldsymbol{\theta}_1$

$$\pi(\boldsymbol{y}|\boldsymbol{\eta}, \boldsymbol{\theta}_1) = \prod_j \pi(y_j|\eta_j, \boldsymbol{\theta}_1).$$

The latent parameters $\boldsymbol{\eta}$ are part of a larger latent random field $\boldsymbol{x}$, which constitutes the second stage of our hierarchical model. The latent field $\boldsymbol{x}$ is modelled as a GMRF with precision matrix $\boldsymbol{Q}$ depending on some hyperparameters $\boldsymbol{\theta}_2$

$$\pi(\boldsymbol{x}|\boldsymbol{\theta}_2) \propto \exp\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{Q}(\boldsymbol{x}-\boldsymbol{\mu})\}$$

The third, and last, stage of the model consists of the prior distribution for the hyperparameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$.

The INLA approach provides a recipe for fast Bayesian inference using accurate approximations to $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ and $\pi(x_i|\boldsymbol{y})$, $i = 0, \ldots, n-1$, i.e. the marginal posterior density for the hyperparameters and the posterior marginal densities for the latent variables. Different types of approximations are available, see Rue et al. (2007) for details. The approximate posterior marginals can then be used to compute summary statistics of interest, such as posterior means, variances or quantiles.

Using the INLA approach it is also possible to challenge the model itself. The model can be assessed through cross-validation in a reasonable time. Moreover, Bayes factors and deviance information criterion (DIC) can be computed in an efficient way providing tools for model comparison.

Computational speed is one of the most important components of the INLA approach, therefore special care has to be put in the implementation of the required algorithms. All procedures necessary to perform INLA are efficiently implemented in the GMRFLib library. This an open source library written in (ANSI) C and Fortran which is freely available on the web page `http://www.math.ntnu.no/~hrue/GMRFLib/`.

The `inla` program is a useful tool which allows the user to easily specify and solve a large class of models, using the algorithms in the GMRFLib library, without any need for C programming. The components of the model and the options for the INLA procedures are specified through a `ini` file. The `inla` program reads the `ini` file, then it builds and solves the model returning the required approximate posterior marginal densities and summary statistics.

The class of models which can be solved using the `inla` program is wide, covering *time series models*, *generalised additive models* (Hastie and Tibshirani, 1990), *generalised additive mixed models* (Lin and Zhang, 1999), *geoadditive models* (Kammand and Wand, 2003), *univariate volatility models* (Taylor, 1986). With the exception of univariate volatility models, the `inla` program supports a subset of the models supported by BayesX. BayesX is a software tool, developed in the University of Munich, for estimating structured additive regression models, Brezger et al. (2003).

In this tutorial we present the `inla` program and, through a series of worked out examples show the possible range of applications where approximate Bayesian inference using INLA can be useful. In Section 2 we discuss the class of models which can be defined and solved using the `inla` program. In Section 3 we describe the use of the `inla` program through a series of worked out examples of increasing complexity. The examples include all, but one, examples in Rue and Held (2005) and all examples in Rue et al. (2007), plus some more examples previously analysed with BayesX. Section 4 describes how to perform model assessment and model comparison. Appendix A consists of a reference manual for the `inla` program while appendix B describes some of the implememnted probability density.

2 Model description

The `inla` program supports hierarchical GMRF models of the following type

$$y_j|\eta_j, \boldsymbol{\theta}_1 \sim \pi(y_j|\eta_j, \boldsymbol{\theta}_1) \qquad j \in J \tag{1}$$

$$\eta_i = \sum_{k=0}^{n_f-1} f_k(c_{ki}) + \boldsymbol{z}_i^T\boldsymbol{\beta} + \epsilon_i \qquad i = 0, \ldots, n_\eta - 1 \tag{2}$$

where

- J is a subset of $\{0, 1, \ldots, n_\eta - 1\}$, that is, not necessarily all latent parameters $\boldsymbol{\eta}$ are observed through the data $\boldsymbol{y}$.
- $\pi(y_j|\eta_j, \boldsymbol{\theta}_1)$ is the likelihood of the observed data assumed to be conditional independent given the latent parameters $\boldsymbol{\eta}$, and, possibly, some additional parameters $\boldsymbol{\theta}_1$. The latent variable η_i enters the likelihood through a known link function, see Appendix A.1 for details.
- $\boldsymbol{\epsilon}$ is a vector of unstructured random effects of length n_η with i.i.d Gaussian priors with precision λ_η:

$$\boldsymbol{\epsilon}|\lambda_\eta \sim \mathcal{N}(\mathbf{0}, \lambda_\eta \boldsymbol{I}) \tag{3}$$

- $\boldsymbol{\eta} = (\eta_1, \eta_2, \dots)$ is a vector of predictors.
- $f_k(c_{ki})$ is the effect of a generic covariate k which assumes value c_{ki} for observation i. The functions f_k, $k = 0, \dots, n_f - 1$ comprise usual nonlinear effect of continuous covariates, time trends and seasonal effects, two dimensional surfaces, iid random intercepts and slopes and spatial random effects. The unknown functions, or more exactly the corresponding vector of function evaluations $\boldsymbol{f}_k = (f_{0k}, \dots, f_{(m_k-1)k})^T$, are modelled as GMRFs given some parameters $\boldsymbol{\theta}_{f_k}$, that is

$$\boldsymbol{f}_k | \boldsymbol{\theta}_{f_k} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}_k^{-1}) \tag{4}$$

- $\boldsymbol{z}_i$ is a vector of n_β covariates assumed to have a linear effect, and is $\boldsymbol{\beta}$ the corresponding vector of unknown parameters with independent zero-mean Gaussian prior with fixed precisions.

The full latent field, of dimension $n = n_\eta + \sum_{j=0}^{n_f-1} m_j + n_\beta$, is then

$$\boldsymbol{x} = (\boldsymbol{\eta}^T, \boldsymbol{f}_0^T, \dots, \boldsymbol{f}_{n_f-1}^T, \boldsymbol{\beta}^T).$$

Note that in the `inla` program the latent field $\boldsymbol{x}$ is parametrised using the predictors $\boldsymbol{\eta}$ instead of the unstructured terms $\boldsymbol{\epsilon}$.

All elements of vector $\boldsymbol{x}$ are defined as GMRFs, hence $\boldsymbol{x}$ is itself a GMRF with density:

$$\pi(\boldsymbol{x}|\boldsymbol{\theta}_2) = \prod_{i=0}^{n_\eta-1} \pi(\eta_i | \boldsymbol{f}_0, \dots, \boldsymbol{f}_{n_f-1}, \boldsymbol{\beta}, \lambda_\eta) \prod_{k=0}^{n_f-1} \pi(\boldsymbol{f}_k | \boldsymbol{\kappa}_{f_k}) \prod_{m=0}^{n_\beta-1} \pi(\beta_m) \tag{5}$$

where

$$\eta_i | \boldsymbol{f}_0, \dots, \boldsymbol{f}_{n_f-1}, \boldsymbol{\beta} \sim \mathcal{N}(\sum_{k=0}^{n_f-1} f_k(c_{ki}) + \boldsymbol{z}_i^T \boldsymbol{\beta}, \lambda_\eta) \tag{6}$$

and $\boldsymbol{\theta}_2 = \{\log \lambda_\eta, \boldsymbol{\theta}_{f_0}, \dots, \boldsymbol{\theta}_{n_f-1}\}$ is a vector of unknown hyperparameters. Note that we include the logarithm of the precision parameters in the vector of hyperparameters.

The last element in the definition of our hierarchical model is a prior distribution for the hyperparameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$. In the `inla` function all precisions are given a Gamma prior with parameters a and b so that the mean is a/b and the variance is a/b^2. See the Appendix for details about the prior distributions for all the hyperparameters of the model.

Many well known models from the literature can be written as special cases of (1) and (2)

- *Time series models*

  Time series models are obtained if $c_k = t$ represents time. The functions f_k can model nonlinear trends or seasonal effects

$$\eta_t = f_{trend}(t) + f_{seasonal}(t) + \boldsymbol{z}_t^T \boldsymbol{\beta}$$

- *Generalised additive models (GAM)*

  A GAM model is obtained if $\pi(y_i|\eta_i, \boldsymbol{\theta}_l)$ belongs to an exponential family, c_k are univariate, continuous covariates and f_k are smooth functions.

- *Generalised additive mixed models (GAMM) for longitudinal data*

  Consider longitudinal data for individuals $i = 0, \dots, n_i - 1$, observed at time points $t_0, t_1, \dots$. A GAMM model extends a GAM by introducing individual specific random effects, i.e.

$$\eta_{it} = f_0(c_{it0}) + \dots, + f_{n_f-1}(c_{it(n_f-1))}) + b_{0i} w_{it0} + \dots + b_{(n_b-1)i} w_{it(n_b-1)}$$

where η_{it} is the predictor for individual i at time t, x_{itk}, $k = 0, \ldots, n_f - 1$,w_{itq}, $q = 0, \ldots, n_b - 1$ are covariate values for individual i at time t, and $b_{0i}, \ldots, b_{(n_b-1)i}$ is a vector of n_b individual specific random intercepts (if $w_{itq} = 1$) or slopes. The above model can be written in the general form in equation (2) by defining $r = (i, t)$, $c_{rj} = c_{itj}$ for $j = 0, \ldots, n_f - 1$ and $c_{r,(n_f-1)+q} = w_{itq}$, $f_{(n_f-1)+q}(c_{r,(n_f-1)+q}) = b_{qi}w_{itq}$ for $q = 0, \ldots, n_b$. In the same way GAMM's for cluster data can be written in the general form (2).

- *Geoadditive models*

  If geographical information for the observations in the data set are available, they might be included in the model as

  $$\eta_i = f_1(c_{0i}) + \cdots + f_{n_f-1}(c_{(n_f-1)i}) + f_{spat}(s_i) + \boldsymbol{z}_i^T\boldsymbol{\beta}$$

  where s_i indicates the location of observation i and f_{spat} is a spatially correlated effect. Models where one of the covariate represent the spatial effect have recently been coined geoadditive by Kammann and Wand (2003).

- *ANOVA type interaction model*

  The effect of two continuous covariate w and v can be modelled as

  $$\eta_i = f_1(w_i) + f_2(v_i) + f_{1|2}(w_i, v_i) + \ldots$$

  where f_1 and f_2 are the main effects of the two covariates and $f_{1|2}$ is a two dimensional interaction surface. The above model can be written in the general form (2) simply by defining $c_{1i} = w_i$, $c_{2i} = v_i$, $c_{3i} = (w_i, v_i)$,

- *Univariate stochastic volatility model*

  Stochastic volatility models are time series models with Gaussian likelihood where it is the variance, and not the mean of the observed data, to be part of the latent GMRF model. That is

  $$y_i|\eta_i \sim \mathcal{N}(0, \exp(\eta_i))$$

  The latent field is then typically modelled as a autoregressive model of order 1.

3 Examples of application

In this section we present a series of worked out examples mostly taken from Rue and Held (2005), Rue et al. (2007) and from the BayesX web page. The aim is both to show the wide range of models which can be solved using the approximate Bayesian inference techniques presented in Rue et al. (2007), and to introduce the `inla` program which makes it possible for the user to apply the above mentioned approximation techniques, making use of the GMRFLib library, in an easy and painless way.

The only input required from the `inla` program is a `ini` file containing the description of the model, the location of the files where the data and the covariates are stored, and, possibly, some options to be passed to the underlying GMRFLib library. The `ini` file is organised in sections each of which either describes one element of the hierarchical model in equations (1) and (2), or specifies some global parameters for the underlying functions in the GMRFLib library. The user is required to specify the likelihood model for the data, the parameters for the prior distribution of the model hyperparameters $\boldsymbol{\theta}$, and to describe, one by one, all components of the latent GMRF $\boldsymbol{x}$ in (2). The `inla` program will then read the model specifications, build the joint probability distribution for the latent GMRF $\boldsymbol{x}$ in equation (5), compute approximations for the required posterior marginals and store the results in a user defined directory.

Before presenting the examples, we describe how the covariate values are stored in files. Each covariate has to be stored in a separate file. The format of the file depends on whether the covariate is assumed to have linear or non-linear effect:

Covariates with linear effect: The value of the covariate is simply stored in a file with n_η columns each row having the format:

$$i \quad z_i$$

where $i = 0, \ldots, n_\eta - 1$ and z_i is the value of the covariate for observation i.

Covariates with non-linear effect: Let $c \in \boldsymbol{C}$ and $\boldsymbol{C} = \{c^{(0)} < c^{(1)} < \cdots < c^{(idx)} < \cdots < c^{(m-1)}\}$. That is, covariate c takes one of the m values in the ordered vector $\boldsymbol{C}$. The file storing covariate c has n_η row, each with the following format:

$$i \quad (idx)_i$$

where $i = 0, \ldots, n_\eta - 1$ and $(idx)_i$ is the position of the observed value c_i in the vector $\boldsymbol{C}$. If the values in $\boldsymbol{C}$ are different from $0, 1, \ldots$, another file of m rows, is necessary to store the values of $\boldsymbol{C}$. A short example will be useful:

Example: *Let $n_\eta = 5$ and $\boldsymbol{C} = \{9, 10, 11\}$. Let the observed covariate values be $c_0 = 10$, $c_1 = 9$, $c_2 = 11$, $c_3 = 9$ and $c_4 = 10$. Then the covariate file will be as following*

$$\begin{array}{cc} 0 & 1 \\ 1 & 0 \\ 2 & 2 \\ 3 & 0 \\ 4 & 1 \end{array}$$

We would need also a file storing the values in $\boldsymbol{C}$:

$$\begin{array}{c} 9 \\ 10 \\ 11 \end{array}$$

Note that all indexes go from 0 to $n - 1$ and not from 1 to n.

We run each example in Section 3.1 on two different machines. The first, defined Machine 1, is a laptop with a Intel(R) Pentium(R) M processor 1.86GHz. The second one, defined Machine 2 is a Dell Poweredge 2950 equipped with two quad-core Itel Xeon 2.66GHz CPUs. For each of the examples we describe the model, the corresponding `ini` file and report some output results and the computation time for each of the two machines.

3.1 A simple time series: the Tokyo rainfall data

Our first example is a simple time series model, analysed, among others, in Rue and Held (2005, Sec. 4.3.4).

Example 1 *The number of occurrences of rainfall over 1 mm in the Tokyo area for each calendar year during two years (1983-84) are registered. It is of interest to estimate the underlying probability p_t of rainfall for calendar day t which is, apriori, assumed to change gradually over time. The likelihood model is binomial*

$$y_t | \eta_t \sim \mathit{Bin}(n_t, p_t)$$

with logit link function

$$p_t = \frac{\exp(\eta_t)}{1 + \exp(\eta_t)}.$$

The model for the latent variables can be written in the general form of equation (2) as

$$\eta_t = f(t)$$

where t is the observed time whose effect is modelled as a smooth function $f(\cdot)$. Following Rue and Held (2005), the random vector $\boldsymbol{f} = \{f_0, \ldots, f_{365}\}$ is assumed to have a circular random walk of order 2 (RW2) prior with unknown precision λ_f.

There is only one hyperparameter $\boldsymbol{\theta} = (\log \lambda_f)$ which we assign a LogGamma(a, b) prior distribution with $a = 1$ and $b = 0.0001$. The LogGamma distribution is defined such that if $X \sim$ LogGamma(a, b), the $Y = \exp(X) \sim$ Gamma(a, b) with E$(Y) = a/b$ and Var$(Y) = a/b^2$.

Figure 1, panel (a), displays the observed frequencies of rain for the 366 time points. The `TOKYO.ini` file which defines the above model for the `inla` program is:

```
1  [The Tokyo-rainfall example]
2  type = problem
3  dir = results
4
5  [Predictor-term]
6  type = predictor
7  initial = 10
8  fixed = 1
9  n = 366
10
11 [data]
12 type = data
13 likelihood = binomial
14 filename = tokyo.rainfall.data
15
16 [latent-RW2]
17 type = ffield
18 covariates = time.covariate
19 n=366
20 model = rw2
21 parameters = 1.0  0.0001
22 cyclic = 1
23 quantiles=0.025  0.975
```

In the following we guide the reader, section by section, through the above `ini` file and explain what the different fields represent. We then briefly illustrate how to run the `inla` program and how and where the output is stored.

Each section of the `ini` file starts with a tag (in square brackets) which is simply a user defined name for the section itself. The order of the sections is not important. The field named ***type*** is contained in each section. It defines the role of the section in the problem specification and, consequently, determines also the nature of all other fields in the same section. There are six specifications for the ***type*** field, see Appendix A.1 for details.

The first section in our `ini` file, defined by ***type***=*problem*, specifies some global parameters. The options specified in this section are valid for the whole problem. Here, the directory where the results will be stored is defined (line 3).

The second section, defined by ***type***=*predictor*, (lines 5-9), deals with vector $\boldsymbol{\eta}$ in (6). The field ***n*** is required and indicates the length n_η of the latent variable vector $\boldsymbol{\eta}$. The `inla` program requires a section of ***type***=*predictor* to always be present, even in cases, like the example we are presenting here, where there is no unstructured random effect $\boldsymbol{u}$ and therefore the predictor vector is a deterministic function of $\boldsymbol{f}_0, \ldots, \boldsymbol{f}_{n_f-1}, \boldsymbol{\beta}$. We mimic the absence of unstructured random effect by declaring the precision λ_η to be fixed and not random (***fixed***=*1*), and the value of the log precision $\log \lambda_\eta$ to be high (***initial***=*10*).

The following section, defined by ***type***=*data* (lines 11-14), specifies the model for the likelihood of the data $\pi(y_t|\eta_t)$ (line 13), and the name of the file where the data are stored (line 14). The format of the data file

depends on the likelihood model, see Appendix A.1.2. For binomial likelihood it is as following:

$$t \quad n_t \quad y_t$$

where t is the data index going from 0 to $(n_d - 1) = 365$.

The last section, defined by ***type***= *ffield* (lines 16-23) specifies the model for the random vector $\boldsymbol{f}$. In this example we have a second order random walk (***model***=*rw2*) of length 366 (***n***=*366*) which is cyclical (***cyclic*** =*1*). We also specify here the parameters a and b for the LogGamma prior for the log precision parameter $\log \lambda_f$ (line 21). We require the `inla` program to compute also the 0.025 and 0.975 quantiles for each of the posterior marginal densities in the latent RW2 field (line 23). The name of the file where the covariate values are stored (line 18) completes the model specification. In this case the covariate is just the observed time point. The covariate file consists of two identical columns with index going from 0 to 365.

$$\begin{array}{cc} 0 & 0 \\ 1 & 1 \\ 2 & 2 \end{array}$$

Once the `ini` file is ready, we can run the program using the following command line:
The option `-v` (verbose) makes the program print out some more information about the model while running. Only for this example, we reproduce the output of the `inla` program to make the reader familiar with it.

```
Processing file [TOKYO.ini]
inla_build...
        number of sections=[5]
        parse section=[0] name=[the tokyo-rainfall example] type=[PROBLEM]
        inla_parse_problem...
                name=[the tokyo-rainfall example]
                use.derivaties=[1]
                dof.max=[50]
                store results in directory=[results0]
                output:
                        cpo=[0]
                        dic=[0]
                        kld=[1]
                        mlik=[0]
                        hyperparameters=[0]
                        summary=[1]
                        density=[1]
                        nquantiles=[0]  [ ]
                        npercentiles=[0]  [ ]
        parse section=[1] name=[predictor-term] type=[PREDICTOR]
        inla_parse_predictor ...
                section=[predictor-term]
                PRIOR->name=[LOGGAMMA]
                PRIOR->PARAMETERS=[1, 0.001]
                initialise log_precision[10]
                fixed=[1]
                n=[366]
                compute=[0]
                output:
                        summary=[1]
                        density=[1]
                        nquantiles=[0]  [ ]
                        npercentiles=[0]  [ ]
        parse section=[2] name=[data] type=[DATA]
        inla_parse_data...
                tag=[data]
                likelihood=[BINOMIAL]
                file->name=[tokyo.rainfall.data]
                read n=[1098] entries from file=[tokyo.rainfall.data]
```

```
                        0/366  (idx,a,y) = (0, 2, 0)
                        1/366  (idx,a,y) = (1, 2, 0)
                        2/366  (idx,a,y) = (2, 2, 1)
        parse section=[3] name=[latent-rw2] type=[FFIELD]
        inla_parse_ffield...
                section=[latent-rw2]
                model=[rw2]
                PRIOR->name=[LOGGAMMA]
                PRIOR->PARAMETERS=[1, 0.000289]
                constr=[0]
                diagonal=[0]
                compute=[1]
                fixed=[0]
                read covariates from file=[time.covariate]
                read n=[732] entries from file=[time.covariate]
                file=[time.covariate] 0/366  (idx,y) = (0, 0)
                file=[time.covariate] 1/366  (idx,y) = (1, 1)
                n=[366]: use default locations, if required
                cyclic=[1]
                initialise log_precision[1]
                output:
                        summary=[1]
                        density=[1]
                        nquantiles=[2]  [ 0.025 0.975 ]
                        npercentiles=[0]  [ ]
        parse section=[4] name=[inla] type=[INLA]
        inla_parse_INLA...
                section[inla]
Contents of ai_param 0x9aa3428
        Strategy:       Use a mean-skew corrected Gaussian by fitting a Skew-Normal
        Fast mode:      On
        Use linear approximation to log(|Q+c|)? No
        Parameters for improved approximations
                Number of points evaluate:      9
                Step length to compute derivatives numerically: 0.000018
                Cutoff value to construct local neigborhood:    0.001000
                Limit to accept a Gaussian fit: 0.010000
                Limit to accept a Skew-Normal fit:      0.010000
        Log calculations:       On
        Log calculated marginal for the hyperparameters:        Off
        Integration strategy:   Use adaptive grid-approach (GRID)
                f0 (CCD only):  1.100000
                dz (GRID only): 1.000000
                Adjust weights (GRID only):     On
                Difference in log-density limit (GRID only):    2.500000
                Skip configurations with (presumed) small density (GRID only):  On
        Gradient is computed using Forward difference with step-length 0.001000
        Hessian is computed using Central difference with step-length 0.001000
        Hessian matrix is forced to be a diagonal matrix? [No]
        Compute effective number of parameters? [Yes]
        Perform a Monte Carlo error-test? [No]
        Interpolator [Auto]

inla_build: check for unused entries in[TOKYO.ini]
inla_INLA...
        Size of full graph=[732]
        Found optimal reordering=[amd]
        List of hyperparameters:
                theta[0] = [log-precision for latent-rw2]
Maximise marginal for hyperparam: log(density) = -332.2833  theta =  8.826705
Maximise marginal for hyperparam: log(density) = -332.2833  theta =  8.826704
Compute the Hessian using central differences and step_size[0.001]. Matrix-type [dense]
     3.757422
```

```
Eigenvectors of the Hessian
        1.000000
Eigenvalues of the Hessian
        3.757422
StDev/Correlation matrix (scaled inverse Hessian)
     0.515887
Search: coordinate 0 direction -1
        config  0=[ -1] log(rel.dens)= -0.46, accept, compute, 0.10s
        config  1=[ -2] log(rel.dens)= -1.68, accept, compute, 0.10s
        config  2=[ -3] log(rel.dens)= -3.44, diff to large, stop searching
Search: coordinate 0 direction 1
        config  3=[  1] log(rel.dens)= -0.54, accept, compute, 0.11s
        config  4=[  2] log(rel.dens)= -2.35, accept, compute, 0.10s
        config  5=[  3] log(rel.dens)= -5.90, diff to large, stop searching
Fill-in computations
        config  6=[  0] log(rel.dens)= -0.00, accept, compute, 0.10s
Combine the densities with relative weights:
        config  0/ 5=[ -1.00] weight = 0.632 adjusted weight = 0.633  neff = 12.49
        config  1/ 5=[ -2.00] weight = 0.186 adjusted weight = 0.209  neff = 14.19
        config  2/ 5=[  1.00] weight = 0.584 adjusted weight = 0.585  neff = 9.69
        config  3/ 5=[  2.00] weight = 0.095 adjusted weight = 0.107  neff = 8.53
        config  4/ 5=[  0.00] weight = 1.000 adjusted weight = 0.963  neff = 11.00
Expected effective number of parameters: 11.233,  #data/#eff.params: 32.58
Done.
        store results in directory[results0]
                store summary results in[results0/latent-rw2/summary.dat]
                store summary (gaussian) results
                    in[results0/latent-rw2/summary-gaussian.dat]
                store marginals in[results0/latent-rw2/marginal-densities.dat]
                store marginal-densities (gaussian)
                    in[results0/latent-rw2/marginal-densities-gaussian.dat]
                store (symmetric) kld's in[results0/latent-rw2/symmetric-kld.dat]
                store quantiles in[results0/latent-rw2/quantiles.dat]
                store quantiles (gaussian)
                    in[results0/latent-rw2/quantiles-gaussian.dat]

Wall-clock time used on[TOKYO.ini]
        Preparations     :   0.025 seconds
        Approx inference:   5.007 seconds
        Output           :   5.848 seconds
        ------------------------------------
        Total            :  10.880 seconds
```

¿¿From the above output we can follow what the `inla` program does: it first reads the different sections, builds the model for the full latent field x, performs the INLA approximation and, finally, stores the results in the appropriate directories. The whole procedure takes less than 10 seconds on Machine 1 and about 2 seconds on Machine 2.

Note that in the output is also reported, for each computed configuration of the hyperparameters, the estimated number of effective parameters (neff), Rue et al. (2007) suggest these as a way to check the accuracy of the approximation of $\pi(\boldsymbol{\theta}|\boldsymbol{y})$. Namely, if the number of effective parameters is small compared to the number of data, then we can expect the approximation to be accurate. In this case the ratio between the number of data and the effective number of parameters is around 32, thus suggesting a good quality of the approximation.

The results are stored in the the directory *results*. The program creates sub-directories to store separately results for each component of the model. In our Tokyo example we have two sub-directories:

- *predictor −term/*
- *latent −rw2/*

The first one is an empty directory since by default the marginals for the predictor term are not computed, see Appendix A.1.3. The second directory contains results for the latent RW2 model. The sub-directories where the results are stored are printed in the last part of the output of the `inla` function.

The default results consist of five files for each sub-directory created, namely:

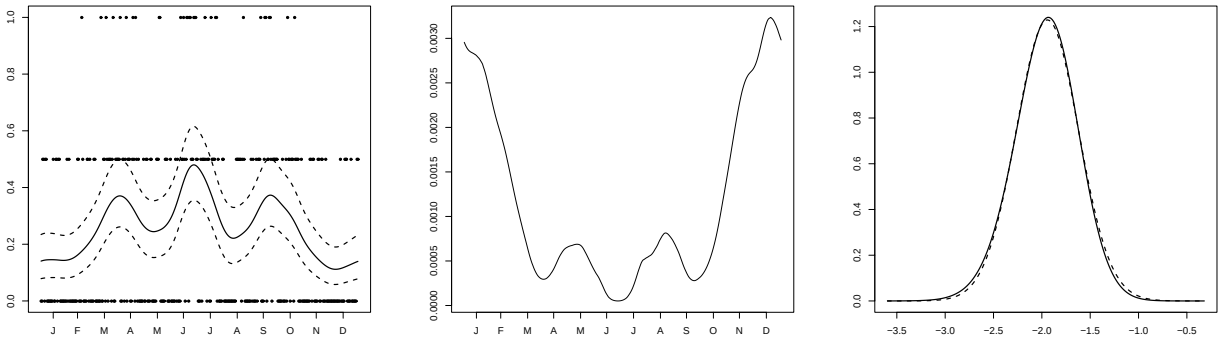
- *marginal−densities−gaussian.dat*
- *summary−gaussian.dat*
- *marginal−densities . dat*
- *summary.dat*
- *symmetric−kld.dat*

Moreover we have two files containing the quantiles

- *quantiles −gaussian.dat*
- *quantiles . dat*

The names of the files are always the same for each sub-directory created. The files whose names ends with *−gaussian.dat* contain results obtained using the Gaussian approximation to approximate the density of $x_t|\boldsymbol{y}, \boldsymbol{\theta}$ (see Rue et al. (2007), Section 3.2.1) while the other files contain results obtained using one of the improved approximations for $x_t|\boldsymbol{y}, \boldsymbol{\theta}$ described in Rue et al. (2007), i.e. the Laplace approximation or its simplified version (default).

The file *symmetric−kld.dat* contains the (symmetric) Kullback-Leibler (KL) divergence between the Gaussian and the (simplified) Laplace approximation to the marginal posterior densities, which we have plotted in Figure 1, panel (b). In this example the divergence is larger for the winter months (November to February), when the observed frequencies are lower, but it stays always very low. Rue et al. (2007) propose to use the Kullback-Leibler distance to check the accuracy of the Gaussian approximation.



(a) Observed frequencies and fitted probabilities with uncertainty bounds

(b) KL-divergence between Gaussian and simplified Laplace approximation for $\pi(x_t|\boldsymbol{y})$

Gaussian (broken line) and simplified Laplace approximation (solid line) for $\pi(x_0|\boldsymbol{y})$

Figure 1: Results for the Tokyo rainfall example

The "*summary*" files contain the mean and the standard deviation for each posterior density. There is one line for each node in the RW2 model and each line is structured as follows:

$$t \quad \mathrm{E}(x_t|\boldsymbol{y}) \quad \sigma(x_t|\boldsymbol{y})$$

Also in the "*quantiles*" files each line refers to one node and is structured as follows:

$$t \quad p(0) \quad x_t(0) \quad p(1) \quad x_t(1) \ldots$$

where $p(j)$ and $x_t(j)$ are such that Prob$(x_t < x_t(j)|\boldsymbol{y}) = p(j)$, $j = 0, 1, \ldots$. The number of columns in the "*quantiles*" files depends on how many quantile values the user choose to compute. In our example there are 5 columns.

Figure 1, panel (a), displays the binomial frequencies and the approximated posterior mean with uncertainty bounds for the underlying probabilities p_t. The probability of rain is smaller in the winter months.

The "*marginal-densities*" files contain the approximated marginal posterior densities. Again each line refers to a different node in the RW2 model and the structure of each line is as follows

$$t \quad x_{t0} \quad \widetilde{\pi}(x_{t0}|\boldsymbol{y}) \quad x_{t1} \quad \widetilde{\pi}(x_{t1}|\boldsymbol{y}) \quad \ldots \quad x_{tK} \quad \widetilde{\pi}(x_{t(K-1)}|\boldsymbol{y})$$

where $(x_{t0}, x_{t1}, \ldots, x_{t(K-1)})$ are $K = 201$ selected values of the variable x_t and $(\widetilde{\pi}(x_{t0}), \widetilde{\pi}(x_{t1}), \ldots, \widetilde{\pi}(x_{t(K-1)}))$ are the corresponding values of the density. Figure 1 (right) displays the Gaussian approximation (broken line) and the simplified Laplace approximation (solid line) for the marginal posterior density of $x_{365}|\boldsymbol{y}$, this node is chosen for being the one for which the KL divergence is maximised. The following R code can be used to reproduce this figure

```
>marginal<-read.table("results0/latent-rw2/marginal.densities.dat")
>gaus.marginal<-read.table("results0/latent-rw2/marginal.densities-gaussian.dat")
>
>
>plot(marginal[1,seq(2,403,2)],marginal[1,seq(3,403,2)],type="l",
      lwd=2,ylab="",xlab="")
>lines(gaus.marginal[1,seq(2,403,2)],gaus.marginal[1,seq(3,403,2)],type="l",
      lwd=2,lty=2)
```

3.2 A time series with seasonal component: the drivers data

The second example is also taken from Rue and Held (2005, Sec 4.4.2). It is again a time series but here we decompose the latent variables η_t into a trend and a seasonal component.

Example 2 *The data consist in monthly counts of car drivers in Great Britain killed or seriously injured in car accidents from January 1969 to December 1984. The time series has $n_d = 192$ data points and exhibits a strong seasonal pattern. One of our goals is to predict the pattern of counts in the 12 month following the last observation.*

We assume the squared root of the counts y_t to be conditionally independent Gaussian random variables:

$$y_t|\eta_t, \lambda_y \sim \mathcal{N}(\eta_t, 1/\lambda_y), \quad t = 0, \ldots, n_d - 1$$

The conditional mean η_t is then a sum of a smooth trend and a seasonal component:

$$\eta_t = season_t + trend_t, \quad t = 0, \ldots, n_\eta - 1 \tag{7}$$

*We assume the vector **season** $= (season_0, \ldots, season_{n_\eta - 1})$ to follow the seasonal model in (3.58) of Rue and Held (2005), with length 12 and unknown precision $\lambda_{\textbf{season}}$, and the vector **trend** $= (trend_0, \ldots, trend_{n_\eta - 1})$ to follow a RW2 with unknown precision $\lambda_{\textbf{trend}}$.*

Note that we have that $n_\eta = n_d + 12 = 204$, since no observations y_t are available for $t = n_d, n_d + 1, \ldots, n_d + 11$. For prediction we are interested in the posterior marginals of $(\eta_{n_d}, \ldots, \eta_{n_d+11})$.

There are three hyperparameters in the model $\boldsymbol{\theta} = (\log \lambda_y, \log \lambda_{\textbf{season}}, \log \lambda_{\textbf{trend}})$ for which we choose the following prior distributions:

$$\begin{array}{lcl} \lambda_y & \sim & \textit{LogGamma}(4, 4) \\ \lambda_{\textbf{season}} & \sim & \textit{LogGamma}(1, 0.1) \\ \lambda_{\textbf{trend}} & \sim & \textit{LogGamma}(1, 0.0005) \end{array}$$

See Rue and Held (2005) for more details.

The corresponding `DRIVERS.ini` file is as follows:

```
[Drivers data]
type = problem
dir = results-%d
quantiles = 0.025 0.975

[Predictor]
type = predictor
parameters = 1 0.0005
initial = 13
fixed = 1
n = 204
compute=1

[data]
type = data
likelihood = gaussian
filename = sqrt-drivers.dat
parameters = 4 4
initial = -2

[trend]
```

```
22 type = ffield
23 covariates = time.dat
24 n=204
25 model = rw2
26 parameters = 1 0.0005
27 initial = 7
28
29 [seasonal]
30 type = ffield
31 model = seasonal
32 covariates = time.dat
33 n = 204
34 season=12
35 parameters = 1 0.01
36 initial = 10
37
38 [INLA parameters]
39 type = INLA
40 h = 0.001
```

We go briefly through the `ini` file ,section by section, highlighting the difference with the previous example.

- *[Drivers data]* section: specifying the quantiles in ***type****=problem* section (line 4) , will make the program compute quantiles for all nodes in the latent field.
- *[Predictor]* section: the precision is fixed to a high value (lines 9-12) to mimic the absence of an unstructured term in the model. Anyway, since our goal is to predict the expected counts we ask the program to compute posterior marginals for $\boldsymbol{\eta}$ as well (***compute****=1*).
- *[data]* section: for Gaussian likelihood the data file has the following format

  $$t \quad w_t \quad y_t$$

  where w_t are fixed weights, see Appendix A.1.2. Note that in this example the length of the observed data (194) is smaller than the length of the latent variables vector $\boldsymbol{\eta}$ (204).
- *[trend]* section: defines the RW2 model for the trend component. At line 26 we also define a starting value for $\log \lambda_{trend}$ for the optimiser.
- *[seasonal]* section: defines the model for the seasonal component of the model, the parameter ***season*** at line 34 defines the season length
- *[INLA parameters]*: this is an optional section, defined by ***type****=INLA*, which specifies some parameters to be passed to the GMRFLib library, in this case we specify the step length for the numerical computation of the gradient and the Hessian of $\widetilde{\pi}(\boldsymbol{\theta}|\boldsymbol{y})$ at its mode, see Appendix for details.

Building and solving the model takes about 10 seconds on Machine 1 and about 3 seconds on Machine 2.

Figure 2 displays the observed and expected counts in the squared root scale (together with 0.025 and 0.975 quantiles). Following is the R code used to produce Figure 2:

```
# Read the files
> data=read.table("sqrt-drivers.dat")
> pred=read.table("results-0/predictor/summary.dat")
> quant=read.table("results-0/predictor/quantiles.dat")
```

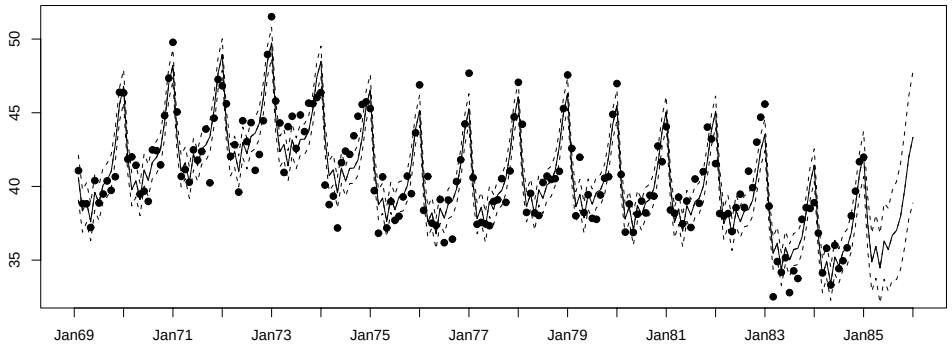


Figure 2: Observed and predicted counts (posterior mean within 0.025 and 0.975 quantiles) for the drivers data example without the seat belt covariate

```
#Make the plot
> plot(data[,3],pch=19,xlim=c(0,205))
> lines(pred[,2])
> lines(quant[,3],lty=2)
> lines(quant[,5],lty=2)
```

We consider now a slight modification of Example 2 as discussed by Rue and Held (2005, Sec 4.2.2):

Example 2 cont. *On January 1983 wearing seat belt became compulsory. To check whether this law had an effect on the number of serious accidents we modify the model as follows:*

$$\eta_t = \begin{cases} \textit{season}(t) + \textit{trend}(t) & t = 0, \ldots, 168 \\ \textit{season}(t) + \textit{trend}(t) + \beta & t = 169, \ldots, 204. \end{cases}$$

We assign additional parameter β a Gaussian distribution with 0 precision, equivalent to a flat prior.

Modifying the `DRIVERS.ini` file to account for the extended model is really easy; it is enough to add a new section as below:

```
[belt]
type=linear
covariates = belt.dat
precision=0
```

The ***type***=*linear* parameter specifies that the new covariate has a lines effect, the file *belt.dat* is as follows

$$\begin{array}{cc} 0 & 0 \\ \vdots & \vdots \\ 168 & 0 \\ 169 & 1 \\ \vdots & \vdots \\ 203 & 1 \end{array}$$

Figure 3 displays the approximate posterior marginal density for β together with 0.025 and 0.975 quantiles. The 95% confidence region is well below 0 indicating a significant effect of the seat belt law in reducing

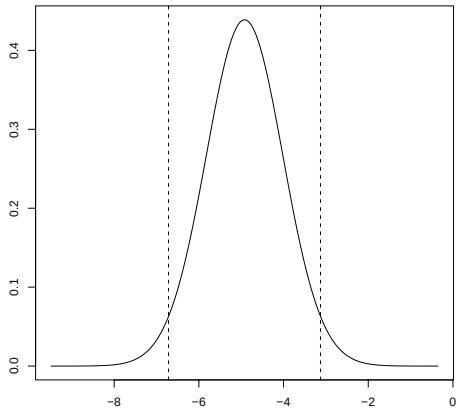


Figure 3: Approximate posterior marginal for parameter β with 0.025 and 0.975 quantiles

the number of dead or injured drivers. Finally, the observed and expected counts in the squared root scale (together with 0.025 and 0.975 quantiles) for the model with the seat belt covariate are displayed in Figure 4, a slightly better fit of this model before and after January 1983 is visible.

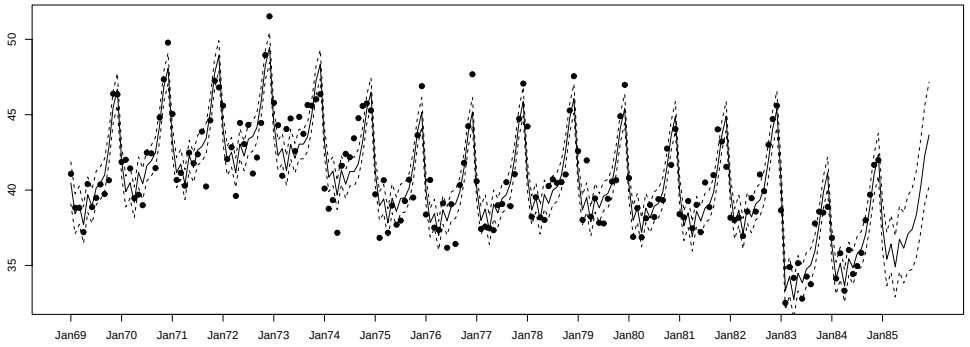


Figure 4: Observed and predicted counts (posterior mean within 0.025 and 0.975 quantiles) for the drivers data example with seat belt covariate

3.3 Stochastic volatility models

Stochastic volatility models are common models in financial time series analysis, lately much interest has been shown in developing efficient MCMC methods for such models, e.g. Shephard and Pitt (1997) and Chib et al. (2002). In the following example, we show how easily a univariate stochastic volatility model can be solved using the `inla` program. The example is taken from Rue et al. (2007) but the model is slightly modified here.

Example 3 *The data consist in 945 observed logarithms of the daily difference of the dollar-pound exchange rate from October 1st, to June 28th, 1985. The data are displayed in Figure 5, panel (a). We analyse this data set using a univariate stochastic volatility model (Taylor, 1986). The likelihood of the data, conditional on the latent variables is:*

$$y_t|\eta_t \sim \mathcal{N}(0, \exp(\eta_t)), \quad t = 0, \ldots, n_d - 1 \tag{8}$$

and the model for the latent variables:

$$\eta_t = \mu + f_t \quad t = 0, \quad , n_\eta - 1 \tag{9}$$

where μ is an unknown common mean with vague Gaussian prior and $\boldsymbol{f} = (f_0, \ldots, f_{n_\eta - 1})$ is modelled as an auto regressive process of order 1 (AR1) with persistence parameter $\phi \in (-1, 1)$ to ensure stationarity, and precision parameter λ_f.

The model has two hyperparameters, $(\log \lambda_f, \phi)$. We re-parametrise the persistence parameter ϕ as

$$\kappa = \textit{logit}\left(\frac{\phi + 1}{2}\right)$$

and assign the following prior distributions

$$\log \lambda_f \sim \textit{LogGamma}(1, 0.0005)$$
$$\kappa \sim \mathcal{N}(0, 1/0.0001)$$

The `VOLATILITY.ini` file defining the model is the following:

```
[Standard Volatility]
type = problem
dir = results-%d

[Predictor term]
type = predictor
n = 1001
initial = 13
fixed = 1
compute=1

[Data]
type = data
likelihood = stochvol
filename = poundd.dat

[AR1]
type = ffield
model = ar1
covariates=time.dat
n=1001
prior0=loggamma                 ;prior for the log-precision
initial0=3                      ;initial value for the log-precision
```

```
24 parameters0 = 1.0 0.0005   ;parameters for the Gamma prior of the precision
25
26 prior1=gaussian              ;prior for \kappa
27 initial1=4                   ;initial value for \kappa
28 parameters1 = 0 0.0001       ;paramters for the Gaussian prior of \kappa
29
30 [Common mean]
31 type=linear
```

The likelihood for the stochastic volatility model is named *stochvol* (line 14) and the format of the data file is

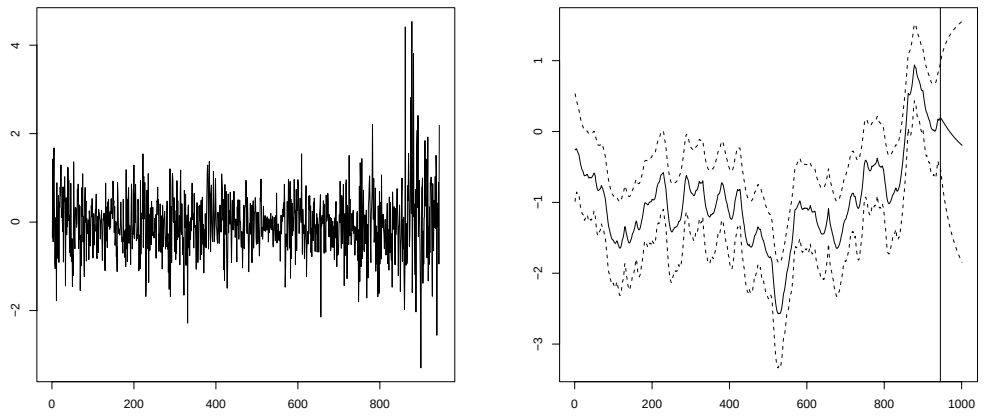
$$t \quad y_t$$

As in Example 2, the precision for the unstructured term λ_η is fixed, but we compute the marginal posteriors distributions for the elements of vector $\boldsymbol{\eta}$.

The AR1 model for $\boldsymbol{f}$ is defined in lines 17-28. Unlike all other models at the moment available for the *ffield* section, the AR1 has two hyperparameters, namely the precision parameter λ_f, and the transformed persistence parameter κ. Lines 22-24 specify the prior and the starting value for the precision parameter $\lambda_{\boldsymbol{f}}$, and lines 26-28 do the same for parameter κ.

The last section of the `ini` file describes the model for the common mean, the default value for the precision is used here.

Note that the length of the data set n_d is 945 but we have set the length of the latent variable vector $\boldsymbol{\eta}$, to be $n_\eta = 1001$ (lines 7 and 21). In this way we obtain also predictions for the unobserved volatility for the 56 days following the last observation.



(a) Log of the daily difference in the Pound/Dollar exchange rate

(b) Posterior mean of $\boldsymbol{\eta}$ together with 0.025 and 0.975 quantiles.

Figure 5: Data and results for the volatility model in Example 3

Building and running the model takes around 110 seconds on Machine 1 and 26 seconds on Machine 2.

Figure 5, panel (b), display the approximate posterior mean for the logarithm of the unobserved volatility, together with 0.025 and 0.975 posterior quantiles. The vertical line indicates the last observed data point.

An alternative model for the response variable y_t is a Student-t. This allows heavier tail, a feature which is often observed in financial time series. The observation model in equation (8) then becomes

$$y_t = \exp(\eta_t/2)\, \mathcal{T}_t(\nu) \quad t = 1, \ldots, T \tag{10}$$

where $\mathcal{T}_t(\nu)$ is a random variable having a Student-t distribution having ν degree of freedom and standardised so that its variance is 1 for any value of $\nu > 2$. To implement the new model it is sufficient to substitute the *[Data]* section (lines 12-15) with

```
1 [Data]
2 type=data
3 likelihood=stochvol_t
4 filename=poundd.dat
```

Yet another model is the *normal inverse Gaussian* (NIG) distribution, for which

$$y_t = \exp(\eta_t/2)\ NIG, \quad t = 1, \ldots, T \tag{11}$$

where NIG is a standarised NIG distrubution with two parameters, which (essentially) are skewness and shape-parameters. To implement the NIG model it is sufficient to substitute the *[Data]* section (lines 12-15) with

```
1 [Data]
2 type=data
3 likelihood=stochvol_nig
4 filename=poundd.dat
```

3.4 Bayesian multiscale analysis for time series data

In the previous examples we were interested in the posterior marginals $\pi(x_i|\boldsymbol{y})$ where the uncertainty about the hyperparameter $\boldsymbol{\theta}$ is integrated out. We present here one example where it is important to be able to precisely estimate posterior marginals for a fixed value of the hyperparameter $\boldsymbol{\theta}$, that is $\pi(x_i|\boldsymbol{y}, \boldsymbol{\theta})$. The example is taken from Rue et al. (2007).

Example 4 *A signal is observed with noise and the goal of the analysis is to detect significant features and structures in the signal. Since some features might be visible only at some specific level of smoothing it is interesting to consider several levels of smoothing simultaneously. This is the idea behind the SIZer (Significant ZERo crossing of derivatives) methodology, see Chaudhuri and Marron (1999) and Erästö (2005).*

In our example the data are Gamma ray burst intensity, plotted in Figure 6 (panel (a)). The observations are assumed to be conditionally independent Poisson random variables

$$y(t_i)|\eta(t_i) \sim Po\{\exp(\eta(t_i)\} \quad i = 0, 1, \ldots$$

Where $\eta(t)$ is the underlying signal of interest. We assume $\eta(t)$ to be continuous with derivatives $\eta'(t)$, and level of smoothing κ. The derivative is said to be "significant positive" at time t if

$$Prob(\eta'(t) > 0|\boldsymbol{y}, \kappa) > 1 - \alpha/2$$

with α being the level of significance. A similar definition holds for "significant negative".

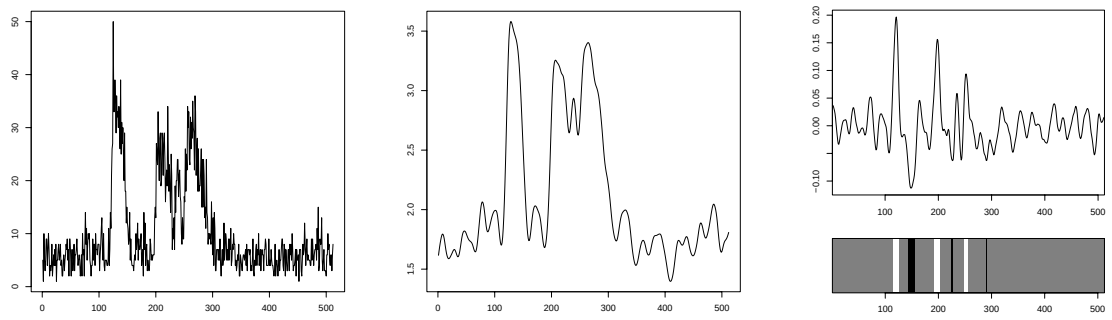
We model $\eta(t)$ as an integrated Wiener process with precision κ which is Markov if augmented with derivatives (Wecker and Ansley, 1983), hence a discretely observed Wiener process observed in n time points is a GMRF of dimension $2n$, see Rue and Held (2005, Sec. 3.5). Our latent GMRF is then $\boldsymbol{x} = (\boldsymbol{\eta}, \boldsymbol{\eta}')$, that is the log-mean of the data augmented with its derivatives.

In this example the precision κ is fixed therefore there are no random hyperparameters in the model.

The file *BURST.ini* is as follows:

```
[Burst data example]
type = problem
dir = results-%d
smtp = GMRFLib_SMTP_BAND

[Poisson data]
type = data
likelihood = poisson
filename = burst.dat

[Predictor term]
type = predictor
n = 512
initial = 10
fixed = 1

[Smoother]
type = ffield
model = crw2
n = 512
covariates = covar.dat
initial = 7
fixed = 1
percentiles = 0
```

The ***smtp*** field in the *[Burst data example]* section (line 4) determines the type of solver for dealing with sparse matrices, in this case, since we know that the precision matrix of the problem is a band matrix, we can use the *GMRFLib_SMTP_BAND* solver which is optimal for band matrices.



(a) (b) (c)

Figure 6: Multiscale analysis example: (a) observed Gamma ray burst intensity, (b) posterior mean for the underlying signal $\eta(t)$ for level of smoothing given by $\log \kappa = 7$, (c) posterior mean of the derivatives $\eta'(t)$ is displayed. The band in the lower part of the figure indicates where the derivatives are found to be significantly positive (white), negative (black) or none (grey).

Notice that all precision parameters are defined ***fixed*** in the `ini` file (lines 15 and 23). The log-precision of the *[Predictor term]* section is fixed to a high value (line 14) again to mimic the absence of the unstructured component in the model, while the log-precision in the *[Smoother]* section is fixed to a user defined value, in this case $\log \kappa = 7$. This determines the level of smoothing in the result. The continuous time random walk model is defined in line 19. Note that even if the length of the smoother term is declared to be 512 (line 20) the actual length of the output file is 1024 since the derivatives are also included. The derivatives constitutes the second half of the output file.

Since we are interested in checking where the derivatives are significantly positive or negative, we compute also the percentiles Prob$(x(t) < 0)$ for the smoother term (line 24). Figure 6 (panel (b)) displays the posterior mean of $\eta(t)$ for $\log \kappa = 7$. In Figure 6(panel(c)) the posterior mean of the derivatives $\eta'(t)$ is displayed. The band in the lower part of Figure 6(c) indicates where the derivatives are found to be significantly positive, negative or none. Figure 6(c) is produced using the following R code:

```
#Read the file containing approximate mean and sd
>smooth=read.table("results-0/smoother/summary.dat")
>
#select the approximations for the derivatives
>deriv=smooth[513:1024,]
>
>xx=deriv[,2]
>
# Create the graph
>split.screen( rbind(c(0,1,0.3,1), c(0,1,0,0.3)))
>
>screen(1)
>par( mar=c(2,2,2,2), oma=c(3,3,2,3) )
>plot(xx[,2],type="l",ylab="",xlab="",xaxs="i")
>
>screen(2)
>par( mar=c(2,2,2,2), oma=c(3,3,2,3) )
>image(1:512,1,mm,axes=F,col=gray(seq(0,1,len=3)))
```

The `inla` program runs in about 7 seconds on Machine 1 and about 2 seconds on Machine 2.

3.5 Disease mapping

Our next example is taken from (Rue and Held, 2005, Sec. 4.4.2). The data are collected over a spatial domain rather than over a time period. The data are georeferenced and we want to include the knowledge of the spatial location of the data in the model.

Each observed data y_i is linked to a spatial region $s \in \boldsymbol{S} = (0, \ldots, S-1)$, so that s_i indicates the region the ith data belongs to. A common way to introduce a spatially correlated effect is to assume that neighbouring sites are more alike than two arbitrary sites, therefore for a valid prior definition, a neighbourhood has to be defined for each site s. In geographical applications a common assumption is that two sites are neighbours if they share a common border.

Let $f_s(s_i)$ indicate the spatial effect. The prior model for $\boldsymbol{f}_s = (f(0), \ldots, f(s), \ldots, f(S-1))$ implemented in the `inla` program is a simple (but most often used) intrinsic GMRF model, see (Rue and Held, 2005, Ch. 3), defined as:

$$f_s(s)|f_s(s'), s \neq s', \lambda_s \sim \mathcal{N}(\frac{1}{n_s}\sum_{s\sim s'} f_s(s'), \frac{1}{n_s\lambda_s}) \tag{12}$$

where n_s is the number of neighbours of site s, $s \sim s'$ indicates that the two sites s and s' are neighbours. λ_s is the unknown precision parameter.

The neighbourhood structure has to be passed to the `inla` program through a file which describes the graph of the spatial component of the model. We describe the required format for such a file using a small example. Let the file *gra.dat*, relative to a small graph, be

```
1  5
2  0  1  1
3  1  2  0  2
4  2  3  1  3  4
5  3  1  2
6  4  1  2
```

Line 1 declares the total number of nodes in the graph, then, in lines 2-6 each node is described. For example, line 4 states that node 2 has 3 neighbours and these are nodes 1, 3 and 4. This is the same format used in the GMRFLib library.

Example 5 *The number of cases of oral cavity cancer is observed for a 5 year period (1986-1990) in the 544 districts of Germany. The goal of the analysis is to explore the spatial distribution of the data. The common approach is to assume that the data are conditionally independent Poisson counts*

$$y_i|\eta_i \sim \textit{Po}(E_i \exp(\eta_i)) \quad i = 0, \ldots, 543$$

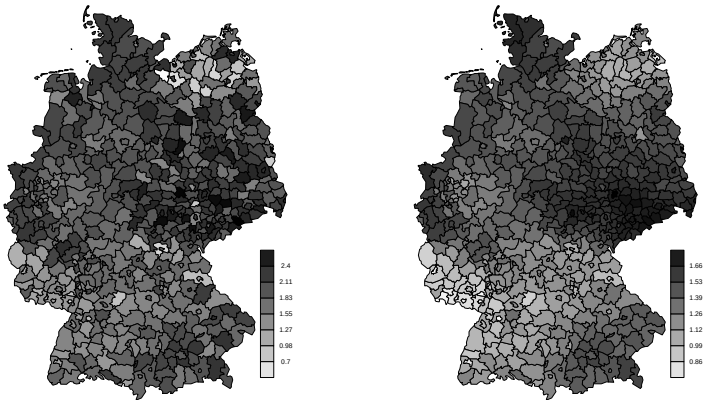
where E_i is a fixed quantity which accounts for number of people in district i, age distribution etc. The standardised mortality ratios y_i/E_i are displayed in Figure 7, panel (a).

The model for the latent variable η_i takes the following form

$$\eta_i = \mu + f_s(s_i) + u_i \tag{13}$$

where μ is th common mean, $\boldsymbol{f}_s$ is a spatially structured term and $\boldsymbol{u}$ is the unstructured term which accounts for non-observed variability. The prior model for $\boldsymbol{f}_s$ is the intrinsic GMRF in equation (12). We impose a sum-to-zero restriction on $\boldsymbol{f}_s$ ($\sum_s f(s) = 0$) to ensure identifiably of μ.

Following Rue and Held (2005), the two precision hyperparameters of the model $(\log\lambda_u, \log\lambda_s)$ are both given LogGamma priors with $a = 1$ and $b = 0.01$.



(a) (b)

Figure 7: Standardised mortality ratio for oral cavity cancer, panel (a) and estimated relative risks (posterior mean) of the spatial component $\exp(\boldsymbol{f}_s)$.

The `DISEASE-oral.ini` file describing the model for the `inla` program is:

```
1  [Oral-cavity cancer data]
2  type = problem
3  dir = results-for-oral-%d
4
5  [Predictor]
6  type = predictor
7  prior = loggamma
8  parameters = 1 0.01
9  n = 544
10
11 [data]
12 type = data
13 likelihood = poisson
14 filename = oral.txt
15
16 [Spatial]
17 type = ffield
18 model = besag
19 covariates=spatial.covariate
20 parameters = 1 0.01
21 constraint = 1
22 graph = germany.gra
23
24 [Constant]
25 type = linear
```

The *[predictor]* section (lines 5-9) defines the model for η_i. Unlike the previous examples, here there actually is an unstructured component, therefore in this case λ_η is not fixed.

The model for the spatial component of $f_s(\cdot)$ is defined in lines 16-22. The section is defined by ***type*** *= ffield* . The intrinsic GMRF model in equation (12) is named *besag* in the `inla` program. Line 21 defines the sum-to-zero constraint for $\boldsymbol{f}_s$. The graph of $\boldsymbol{f}_s$ is read from a file (line 22). The last section, lines 24-25 defines

the model for the common mean μ. Figure 7, panel (b), displays the posterior mean of the spatial component $\exp(\boldsymbol{f}_s)$.

A different parametrisation would have been possible for the same model. Namely we could have dropped the common mean μ and the sum-to-zero constraint. Modifying the `ini` file to account for this other parametrisation is extremely easy; it is, in fact, sufficient to remove lines 24-25 defining the common mean and line 21 defining the constraint.

The `inla` program allows also the possibility to introduce a user defined model for some functions $f(\cdot)$ in equation (2). This is done in a ***type*** = *ffield* section specifying the field ***model*** = *generic*. The user then has to provide the precision function $\boldsymbol{Q}$, corresponding to the stochastic vector $\boldsymbol{f}$, in a file with the following format

$$i \quad j \quad \boldsymbol{Q}_{ij}$$

where i and j are the row and column index and $\boldsymbol{Q}_{ij}$ is the corresponding element of the precision matrix. Only the non-zero elements of the precision matrix need to be stored in the file. For example, we could have stored the precision matrix corresponding to the spatial effect in (13) in a file, named `Qmat.dat`. We report the few first lines of such file:

```
1   0 0 1
2   0 11 -1
3   1 1 2
4   1 9 -1
```

The same model as in (13) can then be defined in a new `ini` file as following:

```
1  [Oral-cavity cancer - User defined Q matrix]
2  type = problem
3  dir = results-%1d
4
5  [Predictor]
6  type = predictor
7  prior = loggamma
8  parameters = 1 0.01
9  n = 544
10
11 [data]
12 type = data
13 likelihood = poisson
14 filename = oral.txt
15
16 [Spatial]
17 type = ffield
18 model = generic
19 Qmatrix = Qmat.dat
20 rankdef = 1
21 covariates = spatial.covariate
22 parameters = 1 0.01
23 constraint = 1
24
25 [Constant]
26 type = linear
```

Notice that the only difference with respect to the `ini` file previously used is in the section *[Spatial]*. Here we declare ***model*** = *generic* and specify the file containing the $\boldsymbol{Q}$ function in line 19. The `inla` program then builds a graph based on the non-zero pattern of the specified precision matrix. The optional argument ***rankdef***, in line 20, specifies the rank deficiency of the precision matrix. For the intrinsic model in equation (12) the rank deficiency is 1.

3.6 Disease mapping with covariate

We present now an extension of the model in Example 5 which allows for adjusting the log-relative risk by a semi-parametric function of a covariate which is believed to influence the risk. The model is a Bayesian semiparametric model with an additional spatial effect. These kinds of models have been named "geoadditive models" in Kammann and Wand (2003). For an introduction to the subject see, for example, Fahrmeir and Tutz (2001). The example below is taken from Rue et al. (2007).

Example 6 *Larynx cancer mortality counts are observed in the 544 district of Germany from 1986 to 1990. As in Example 5 we assume the data to be conditionally independent Poisson random variables with mean $E_i \exp(\eta_i)$, where E_i is fixed and accounts for demographic variation, and η_i is the log-relative risk. Together with the counts, for each district, the level of smoking consumption c is registered.*

The model for η_i takes the following form

$$\eta_i = \mu + f_s(s_i) + f(c_i) + u_i \tag{14}$$

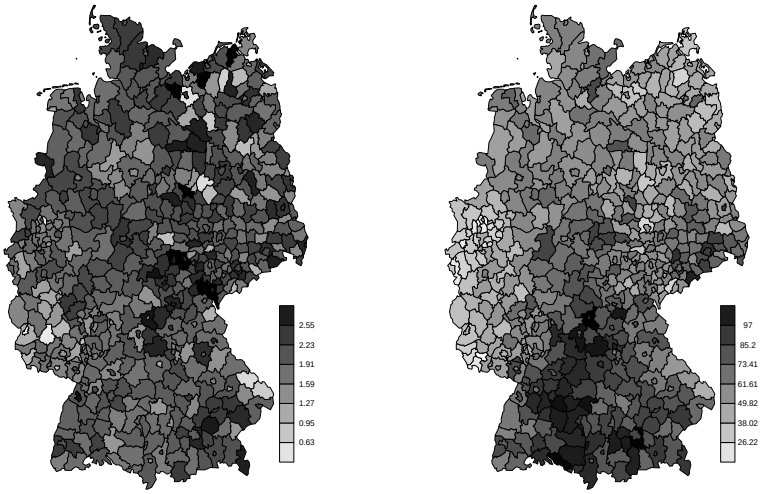
where, as in Example 5, $f_s(\cdot)$ is the spatial effect modelled according to (12), and u_i is the unstructured random effect. The remaining term in (14), $f(c_i)$, is the unknown effect of of the exposure covariate which assumes value c_i for observation i. The effect of covariate c is modelled as a smooth function $f(\cdot)$ parametrised as unknown values $\boldsymbol{f} = (f_0, \ldots, f_{m-1})^T$ at $m = 100$ equidistant values of c_i. We have scaled the covariate values so that they belong to the interval $[0, 10]$. The vector $\boldsymbol{f}$ is modelled with a second-order random walk (RW2) prior with unknown precision λ_f. A sum-to-zero constraint is imposed on $\boldsymbol{f}_s$ and $\boldsymbol{f}$ separate out the spatial effect and the effect of the covariate from the common mean μ.

The model has three hyperparameters $\boldsymbol{\theta} = (\log \lambda_s, \log \lambda_f, \log \lambda_\eta)$. Following Rue et al. (2007) we assign a vague LogGamma prior to each element of $\boldsymbol{\theta}$.

In Figure 9 the standardised mortality ratios, y_i/E_i are displayed (panel (a)) together with the observed values of the covariate c (panel (b)).

The `DISEASE-COVARIATE.ini` file defining the model is the following:

```
[Disease mapping with covariate]
type = problem
dir = results-%d

[Predictor term]
type = predictor
n = 544
prior = loggamma
initial=9
parameters = 1.0 0.00005

[Data]
type = data
likelihood = poisson
filename = larynx.dat

[Spatial]
type = ffield
model = besag
covariates=spatial-covariate.dat
prior = loggamma
parameters = 1.0 0.00005
```



(a) (b)

Figure 8: Standardised mortality ratio for larynx cancer, panel (a) and observed covariate values, panel(b)

```
23 graph = germany.gra
24 constraint = 1
25 initial=3
26 diagonal = 0.001
27
28 [Covariate]
29 type =ffield
30 model = rw2
31 covariates = covariate.dat
32 locations=covariate.value
33 prior = loggamma
34 parameters = 1 0.05
35 initial=9
36 diagonal = 0.00001
37 quantiles=0.025 0.975
38 constraint = 1
39
40 [Constant linear]
41 type = linear
42
43 [INLA param]
44 type = INLA
45 h = 0.001
```

The section *[Spatial]* defines the model for the structured spatial component $\boldsymbol{f}_s$. We recognise the intrinsic GMRF model in line 19 and the graph file in line 23. The field ***diagonal*** at line 36 indicates a (small) number to be added to the diagonal of the precision matrix for $\boldsymbol{f}_s$ to ensure that it is positive definite.

The model for the semi-parametric function $\boldsymbol{f}$, which is the new feature introduced by this example, is defined in the section tagged *[Covariate]*. The file *covariate.value* declared in line 32 contains all values that the covariate c could assume, they are ordered from the lower to higher. In this case the file contains one sequence

of numbers from 0 to 9.9 with step 0.1. The file *covariate . dat* contains information on which values of c is actually observed in each district. We report the first 5 lines of the file to better explain the format of such files

```
1 0  56
2 1  65
3 2  50
4 3  63
5 4  65
```

For example, line 3 tells us that for district 2 the observed value of the covariate c is the 50th element of the series in file *covariate . value*, that is 0.5.

In the last section, tagged *[INLA param]* we define the step length for the numerical computation of the gradient and Hessian of $\widetilde{\pi}(\boldsymbol{\theta}|\boldsymbol{y})$ at the mode. This is necessary because the default values do not always ensure a positive definite Hessian matrix.

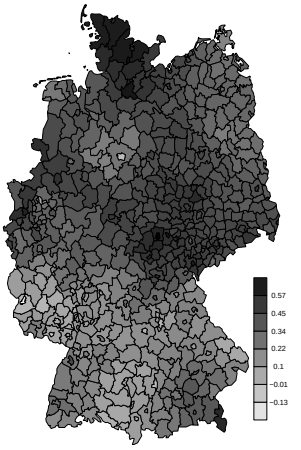


Figure 9: Posterior mean for the structured spatial effect $\boldsymbol{f}_s$

The computation time is about 30 seconds on Machine 1 and 15 seconds on Machine 2.

Figure 9 displays the posterior mean of the spatial effect $\boldsymbol{f}_s$ for all districts. To reproduce Figure 9 the following R code has been used:

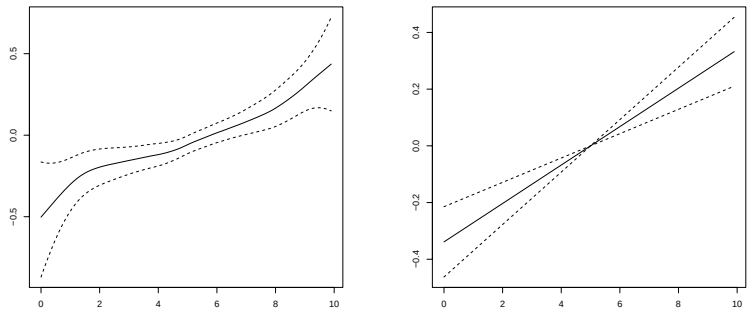
```
> source("draw-map.r")
> spatial=read.table("results-0/spatial/summary.dat")
> germany.map(spatial[,2])
```

The R code `draw.map.r` can be downloaded together with all the other example files.

Figure 10, panel (a), displays the effect of the covariate c (posterior mean) within 2.5 and 97.5% confidence intervals. The covariate effect is not too far from a linear effect. We might, therefore, want to run a modified version of the model in which the effect of c is modelled as a linear function, that is

$$\eta_i = \mu + f_s(s_i) + \beta c_i + u_i \tag{15}$$

To modify the `DISEASE-COVARIATE.ini` file in order to fit the new model it is enough to delete the *[Covariate]* section, lines 28-38 and instead add the following section where β is defined.



(a) (b)

Figure 10: Effect of the covariate. Panel (a) nonparametric model and panel (b) linear model: posterior mean within 2.5 and 97.5% confidence interval.

```
[Covariate linear]
type=linear
covariates=covariate-linear.dat
```

The file *covariate−linear.dat* has the format

$$i \quad c_i$$

The computation time for the linear-effect model reduces to 11 seconds for Machine 1 and to 6 seconds on Machine 2. This is due to the fact that in the linear model both the latent field $\boldsymbol{x}$ and the vectors of hyperparameters $\boldsymbol{\theta}$ are of lower dimensionality.

The estimated posterior mean for the slope parameter β is 0.0677 with posterior standard deviation 0.0126. Figure 10, panel (b), displays the linear effect of the covariate within 0.025 and 0.975 quantiles. To compute the quantiles for the regression line in Figure 10, panel (b), we have run the model described in the `DISEASE-COVARIATE.ini` file fixing the log precision of the RW2 model to a high value. In this way the RW2 is forced to be a straight line.

3.7 Mapping cancer incidence

We present a little more complicated example on the same line of examples 5 and 6. Instead of observing only one data point for each district, in the next example there are multiple observations sharing the same spatial location. Therefore, a possible unstructured spatial effect needs to be coded in a different way than in the two previous examples. The example is taken from Rue and Held (2005, Sec 4.3.5).

Example 7 *The data are incident cases of cervical cancer in the former East German Republic (GDR) from 1979, stratified by district and age group. Each cases was classified as pre-malignant (coded as 0) or malignant (coded as 1). For each of the $n_d = 6\,690$ cases in the data set, the age, age$_i$, and the district, s_i, of the patient are available. The age was categorised into 15 age groups.*

The data are assumed to be conditionally independent Bernoulli random variables:

$$y_i|\eta_i \sim \mathcal{B}(p_i) \quad i = 0, \ldots, n_d$$

with logit link function

$$p_i = \frac{\exp(\eta_i)}{1 + \exp(\eta_i)}$$

The model for the latent variables is:

$$\eta_i = \mu + f(\text{age}_i) + f_s(s_i) + f_u(s_i)$$

where $f(\text{age})$ is the age group effect, modelled as a RW2 with precision parameter λ_f. The spatial effect of the district s_i is split into a spatially correlated part and an uncorrelated one. The spatially correlated element, $f_s(\cdot)$, is modelled as the intrinsic GMRF in equation (12) with given neighbouring structure. The uncorrelated part, $f_u(\cdot)$, is modelled as by a i.i.d Gaussian effect. Note that, in this model, the unstructured spatial effect $f_u(\cdot)$, does not coincide with the unstructured term u_i in equation (2), which was the case in Examples 5 and 6.

There are three hyperparameters in the model $\boldsymbol{\theta} = (\log \lambda_f, \log \lambda_s, \log \lambda_u)$. Following Rue and Held (2005), we assume a LogGamma$(1.0, 0.01)$ prior distribution for $\log \lambda_s$ and $\log \lambda_u$ and a LogGamma$(1.0, 0.00005)$ prior for $\log \lambda_f$. Moreover we impose a sum-to-zero constraint on both $\boldsymbol{f}$ and $\boldsymbol{f}_s$

The file *CANCER−INCIDENCE.ini* defining the model is:

```
[Cancer incidence]
type = problem
dir  = results-%d

[Predictor]
type = predictor
n = 6690
initial = 15
fixed = 1

[Likelihood model]
type = data
likelihood = binomial
filename = cancer.dat

[Age classes]
type = ffield
model = rw2
covariates = age-group-cov.dat
```

```
n=15
constraint = 1
diagonal = 1.0e-4
parameters = 1 0.001
initial = 6.456745
quantiles =0.025 0.975

[Spatial]
type = ffield
model = besag
graph = ddr.gra
covariates = spatial-cov.dat
constraint = 1
diagonal = 1.0e-4
parameters = 1 0.0005
initial = 8.006793

[Spatial random effect]
type = ffield
model = iid
n = 216
parameters = 1 0.01
covariates = spatial-cov.dat
initial = 4.512093

[constant]
type = linear

[Parameters for INLA]
type = INLA
h = 0.01
```

Note that while in Examples 5 and 6 the spatial unstructured component in the model was coded in the ***type***=*predictor* section of the `ini` file, here, for the same purpose, we have to include a ***type***= *ffield* section where ***model***=*iid* (lines 37-43).

The model runs in about 90 seconds on Machine 1 and about 30 seconds on Machine 2.

In Figure 11 the posterior mean of the non-parametric effect of the age group within 2.5 and 97.5% confidence band is dispayed.

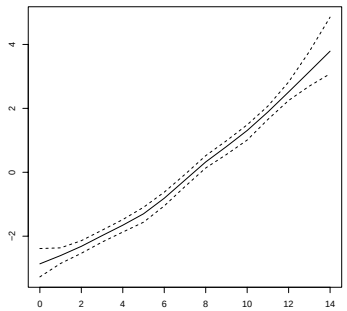


Figure 11: Nonparametric effect of age group. Posterior mean within 2.5 and 97.5% quantiles.

3.8 Geoadditive model: Munich rental guide

In this section we present a slightly more complex example of geoadditive models where we have a higher number of covariates in the data set. The example is taken from Rue and Held (2005, Sec. 4.2.1).

Example 8 - Munich rental guide

The response variable y_i is the rent (Euro per square meter) for a flat in Munich. There are three covariates to be included in the model: the spatial location (s_i), the floor space (size$_i$) and the year of construction (year$_i$). Moreover for each data point we have a set of indicator variables such as whether or not the flat has central heating, bathroom, a large balcony, etc. The data set consist in $n_d = 2\,035$ observations. There are 380 district in Munich, the floor size varies from 17 to 185 square meters and the year of construction goes from 1918 to 2001.

The model for the data is:

$$y_i|\eta_i \sim \mathcal{N}(\eta_i, 1/\lambda_y)$$

with

$$\eta_i = \mu + f_s(s_i) + f_0(\text{size}_i) + f_1(\text{year}_i) + \boldsymbol{z}_i^T\boldsymbol{\beta} \tag{16}$$

where $f_s(\cdot)$ is the spatial effect modelled as the intrinsic GMRF in equation (12), $f_0(\cdot)$ is the non parametric effect of the floor size and $f_1(\cdot)$ is the non parametric effect of the year of construction. Both $f_0(\cdot)$ and $f_1(\cdot)$ are modelled as RW2 with unknown precision. The last term in (16) models the covariates assumed to have a linear effect. As usual we choose a Gaussian prior with known precision for the elements of vector $\boldsymbol{\beta}$. We impose a sum-to-zero constraint on $f_s(\cdot)$, $f_0(\cdot)$ and $f_1(\cdot)$.

The model has four hyperparameters $\boldsymbol{\theta} = (\log\lambda_y, \log\lambda_s, \log\lambda_0, \log\lambda_1)$. We assign to each precision a LogGamma$(1.0, 0.001)$ prior. In this example we approximate also the posterior marginals for the four hyperparameters $\boldsymbol{\theta}$.

In the following we report part of the `RENT.ini` file which defines the model. We have omitted the part defining most of the indicator variables since they are all defined in the same way.

```
[Rent in Munich]
type = problem
dir = results-%d
hyperparameters = 1

[Predictor term]
type = predictor
n = 2035
parameters = 1.0 0.001
initial = 10
fixed = 1

[Data]
type = data
likelihood = gaussian
filename = rent.dat
parameters = 1 0.001
initial = -1

[floor-size]
type =ffield
model = rw2
covariates = size-covariate.dat
```

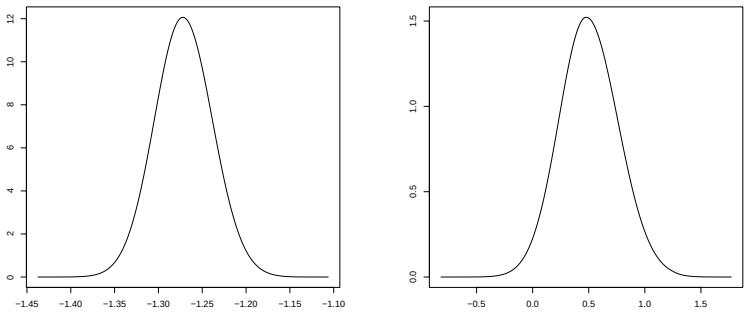
```
24 locations = size-loc.dat
25 diagonal = 1.0e-6
26 initial = 7
27 constraint = 1
28 parameters = 1 0.001
29 quantiles = 0.25 0.975
30
31 [spatial]
32 type = ffield
33 model = besag
34 graph = munich.gra
35 covariates = spatial-covariate.dat
36 diagonal = 0.00001
37 constraint = 1
38 initial = 0.4
39 parameters = 1 0.001
40 compute=1
41
42 [year]
43 type =ffield
44 model = rw2
45 covariates = year-covariate.dat
46 locations = year-loc.dat
47 diagonal = 1.0e-6
48 initial = 7
49 constraint = 1
50 parameters = 1 0.001
51 quantiles = 0.25 0.975
52
53 [constant]
54 type = linear
55 precision = 0.01
56
57 [linear-beste.dat]
58 type = linear
59 covariates = beta-beste.dat
60 precision = 0.01
61
62 .
63 .
64 .
65
66 [INLA param]
67 type = INLA
68 int_strategy = GMRFLib_AI_INT_STRATEGY_CCD;
69 h = 0.01
```

The flag ***hyperparameters*** in line 4 section is turned on to indicate that also posterior marginals for the hyperparameters have to be computed. The results are displayed in Figure 12 and they agree well with tho use found by Rue and Held (2005).

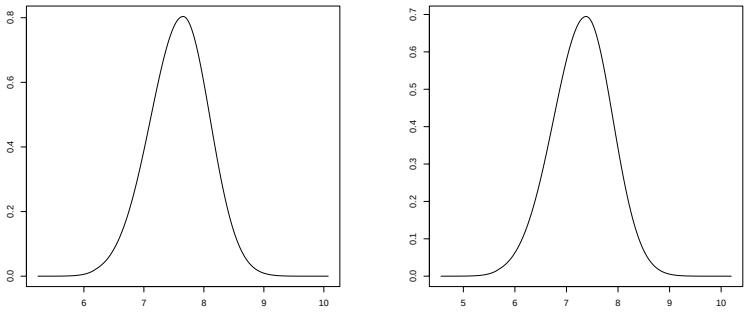
The new feature introduced in this example is the use of a different integration scheme to compute

$$\widetilde{\pi}(x_i|\boldsymbol{y}) = \sum_k \widetilde{\pi}(x_i|\boldsymbol{y}, \boldsymbol{\theta}_k)\widetilde{\pi}(\boldsymbol{\theta}_k|\boldsymbol{y})\Delta_k \tag{17}$$

When the dimension of the hyperparameters space grows, in fact, the grid integration scheme, which was used in all previous examples and which is the default choice in the `inla` program, soon becomes too computa-



(a) $\widetilde{\pi}(\log \lambda_y|\boldsymbol{y})$ (b) $\widetilde{\pi}(\log \lambda_s|\boldsymbol{y})$



(c) $\widetilde{\pi}(\log \lambda_0|\boldsymbol{y})$ (d) $\widetilde{\pi}(\log \lambda_1|\boldsymbol{y})$

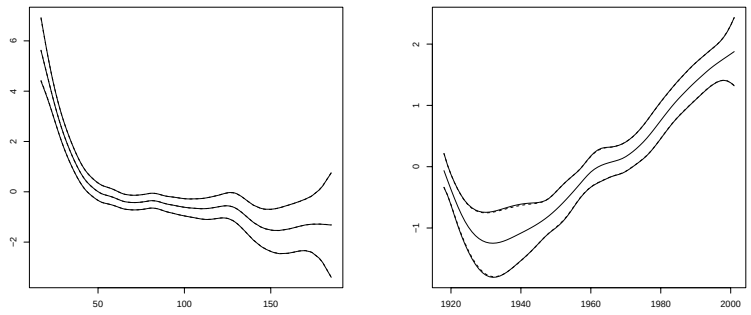
Figure 12: Munich rent example: approximate posterior marginals for the hyperparameter of the model.

tionally intensive. The central composit design (CCD) integration scheme, defined in line 68, is an alternative integration scheme which computes the integral in (17) using much less points, still providing useful results. Both integration schemes are described in Rue et al. (2007).

Figure 16, panels (a) and (b), displays the posterior mean, within 0.25 and 0.975 quantiles, of the effect of the floor size and the year of construction respectively.

To check the quality of the CCD integration scheme we run the model once more using the default grid scheme (to do so it is enough to delete line 67). The results are plotted in Figure 13 as dotted lines, they are indistinguishable from the CCD results despite the fact that the grid integration scheme used 115 evaluation points to compute the integral in (17) and the CCD one only 15.

The computing time for this model on Machine 1 is of 80 seconds if we use the CCD scheme and 250 seconds using the grid scheme. On Machine 2 the computational time reduces to 30 seconds in the first case and 70 in the second case.



(a) (b)

Figure 13: Munich rent example: semiparametric effect of the floor size (a) and of the year of construction (b). The posterior mean within 0.025 and 0.975 quantiles is displayed. The solid line is the result of the CCD integration scheme and the dotted line is the result of the grid integration scheme.

3.9 Geoadditive model: Zambia children undernutrition

The second example of geoadditive model with several covariates is from Kandala et al. (2001) and is one of the worked out examples in the BayesX web page.

Example 9 - Undernutrition of children in Zambia. *Undernutrition in children is measured determining the anthropometric status of the child relative to a reference standard. In our example undernutrition is measured by stunting, or inefficiency height for age, indicating chronic undernutrition. Stunting for a child i is determined using a Z score defined as*

$$Z_i = \frac{AI_i - MAI}{\sigma}$$

where AI refers to the child's anthropometric indicator, MAI refers to the median of the reference population and σ refers to the deviation of the standard population.

The main interest is on modelling the dependence of undernutrition on a set of covariates including the age of the child (age_i), the body mass index of the child's mother (bmi_i), the district the child lives in (s_i) and some further categorical covariates. The data set consists in $n_d = 4846$ observations. For more details about the data set see Kandala et al. (2001) and Kneib et al. (2004).

We assume the scores Z_i to be conditionally independent Gaussian random variables

$$Z_i|\eta_i \sim \mathcal{N}(\eta_i, 1/\lambda_y)$$

and

$$\eta_i = \mu + f_0(bmi_i) + f_1(agc_i) + f_s(s_i) + f_u(s_i) + \boldsymbol{z}_i^T\boldsymbol{\beta}$$

where $f_0(\cdot)$ and $f_1(\cdot)$ are the semi parametric effect of the mother's body mass index and the age of the child respectively. $f_s(\cdot)$ is the structured spatial effect of the district, $f_u(\cdot)$ is an unstructured spatial effect and $\boldsymbol{z}_i$ are a set of categorical covariates. We model the spatial structured effect $f_s(s_i)$ as the intrinsic GMRF in equation (12) and $f_0(\cdot)$ and $f_1(\cdot)$ as RW2. The unstructured spatial effect $f_u(s_i)$ is modelled by i.i.d. Gaussian random variables. We impose a sum-to-zero constraint for $f_s(\cdot)$, $f_0(\cdot)$ and $f_1(\cdot)$.

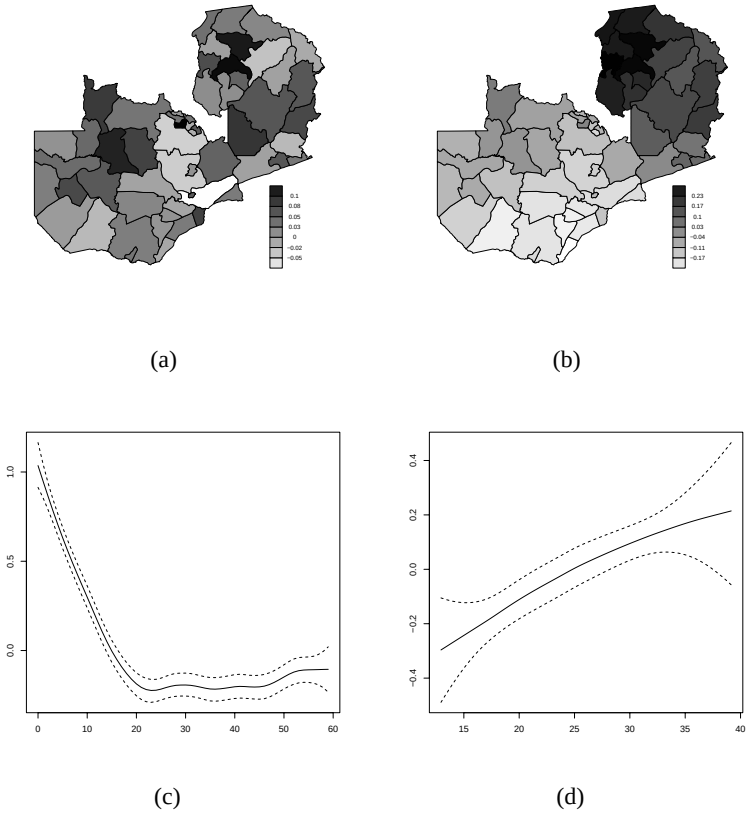
In this model there are five hyperparameters $\boldsymbol{\theta} = (\log\lambda_y, \log\lambda_s, \log\lambda_u, \log\lambda_0, \log\lambda_1)$ and we assign a vague LogGamma prior distribution to each of them.

```
1  [Zambia model]
2  type = problem
3  dir = results-%d
4
5  [Predictor term]
6  type = predictor
7  n = 4846
8  prior = loggamma
9  parameters = 1.0 0.005
10 initial = 10
11 fixed = 1
12
13 [Data]
14 type = data
15 likelihood = gaussian
16 filename = zambia.dat
17 parameters = 1 0.005
18 initial = 0.2
19
```

```
[spatial]
type = ffield
model = besag
graph = zambia.gra
covariates = spatial_covariate.dat
diagonal = 0.00001
constraint = 1
initial = 3.6
parameters = 1 0.005

[spatial predictor]
type = ffield
model = iid
covariates = spatial_covariate.dat
n =57
diagonal = 0.00001
initial = 5.4
parameters = 1 0.005

[agc]
type = ffield
model = rw2
covariates = agc.dat
n=60
diagonal = 0.0001
constraint = 1
initial = 6.6
parameters = 1 0.005
quantiles = 0.025 0.975

[bmi]
type = ffield
model = rw2
covariates = bmi_covariate.dat
locations = bmi.location
diagonal = 0.00001
constraint = 1
initial = 6.2
parameters = 1 0.005
quantiles = 0.025 0.975

[beta]
type=linear

[rcw]
type=linear
covariates = rcw.dat

[edu1]
type=linear
covariates = edu1.dat

[edu2]
type=linear
covariates = edu2.dat

[sex]
```

```
type=linear
covariates = sex.dat

[tpr]
type=linear
covariates = tpr.dat

[INLA param]
type = INLA
int_strategy = CCD;
```

Also in this example we use the CCD integration scheme to compute the integral in (17).



(a) (b)

(c) (d)

Figure 14: Results for the Zambia example. Panel (a) and (b) displays the posterior mean of predictor $\boldsymbol{\eta}$ and of structured spatial effect respectively. Panel (c) and (d) display the posterior mean, within 0.025 and 0.975 quantiles, of the age effect (c) and of the mother's body mass index (d)

In Figure 14, panels (a) and (b), the posterior mean of the predictor and of the structured spatial effect is displayed. The effect of the age of the children is in Figure 14, panel (c). It shows a clear non linear pattern. The effect of the mother's body mass index (Figure 14, panel (d)) instead is more regular and could probably be substitute in the model formulation by a linear effect.

The computation time is about 4 minutes on Machine 1 and 1 minute on Machine 2.

3.10 Log-Gaussian Cox processes

The particular feature of our next example is that data are registered on a regular grid of dimension $n_{row} \times n_{col}$, where n_{row} is the number of row and n_{col} the number of columns. Unlike all the previous examples then, each data is identified by two indexes (i, j) indicating respectively the row and column the data point belongs to. This example is taken from Rue et al. (2007).

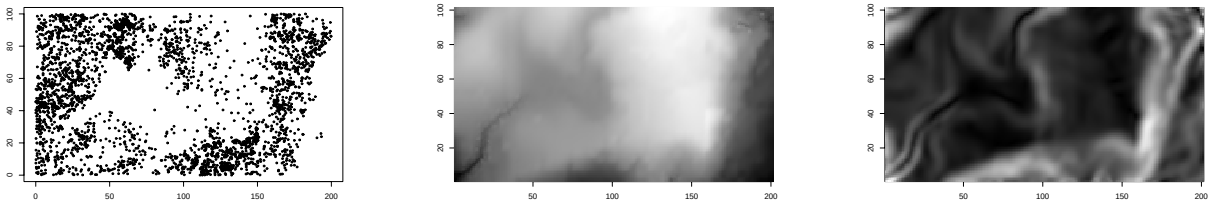
Example 10 *Log-Gaussian Cox processes (LGCP) are a class of models used for modelling spatial point processes, see for example Møller and Waagepetersen (2003). A LGCP is a Poisson point process.* $\boldsymbol{Y} \in W \subset \mathcal{R}^d$. *with random intensity function* $\lambda(\boldsymbol{\xi}) = \exp(Z(\boldsymbol{\xi}))$, *where* $Z(\boldsymbol{\xi})$ *is a Gaussian field and* $\boldsymbol{\xi} \in W$. *It is common practice to discretise the observation windows* W *into* $N = n_{row} \times n_{col}$ *disjoint cells* $\{s_{ij}\}$*with area* $|s_{ij}|$ *where* $i = 0, \ldots, n_{row} - 1$ *and* $j = 0, \ldots, n_{col} - 1$.

Let y_{ij} *be the observed number of occurrences of the realised point pattern within* s_{ij}. *Let* η_{ij} *be the random variable* $Z(\boldsymbol{\xi}_{ij})$. *The likelihood of the model is*

$$y_{ij}|\eta_{ij} \sim Po(|s_{ij}| \exp(\eta_{ij}))$$

while, as usual the latent variable vector $\boldsymbol{\eta}$ *is part of a larger GMRF.*

In this example, the data consist in the locations of a particular tropical tree species (Beilschmiedia pendula Lauraceae) registered in a 50-hectares plot in the tropical moist forest of Barro Colorado Island in central Panama. For more information about this study see Waagepetersen (2006). The 3605 tree locations are plotted in Figure 6, panel (a). We divide our region of interest into a 201×101 *regular grid, where each square pixel represent an area of 25 squares meters. Together with the data* y_{ij}, *we observe, the mean elevation and the mean norm of the gradient for each area on the grid. These covariates are believed to influence the behaviour of the tree under examination. A scaled version of these covariates is displayed in Figure 15, panels (b) and (c). The model for the latent variable* η_{ij} *is*



(a) (b) (c)

Figure 15: Data and covariate for the LGCP example: panel (a) displays locations for the 3065 trees, panel (b) displays the altitude and panel (c) the norm of the gradient.

$$\eta_{ij} = \mu + \beta_1 alt_{ij} + \beta_2 grad_{ij} + f_s(s_{ij}) + u_{ij}$$

where alt_{ij} *and* $grad_{ij}$ *are the values for the two covariates at location* (i, j), f_s *is the spatial structured effect of the location and* u_{ij} *is the unstructured random effect.*

For the spatial structured term $\boldsymbol{f}_s$ *we use a second order polynomial intrinsic GMRF with unknown precision* λ_f. *See Rue and Held (2005, Sec 3.4.2) for a thorough definition of intrinsic GMRF models on a lattice. We use vague Gaussian priors for* μ, β_1 *and* β_2. *The unstructured terms* u_{ij} *are independent* $\mathcal{N}(0, 1/\lambda_u)$ *random variables. Notice that the latent field* $\boldsymbol{x} = (\boldsymbol{\eta}, \boldsymbol{f}_s, \mu, \beta_1, \beta_2)$ *in this example has dimension* $40\,605$.

The hyperparameters are $\boldsymbol{\theta} = (\log \lambda_f, \log \lambda_u)$ *are are assigned vague LogGamma priors.*

```
1  [Tropical rainforest data]
2  type = problem
3  dir = results-%d
4
5  [Poisson data]
6  type = data
7  likelihood = poisson
8  filename = data-full.dat
9
10 [Predictor term]
11 type = predictor
12 nrow = 101
13 ncol = 201
14 initial = 0.4
15
16 [Spatial smoother]
17 type = ffield
18 covariates=spatial-full.dat
19 nrow=101
20 ncol=201
21 model = rw2d
22 constraint=1
23 initial=0.7
24
25 [Constant]
26 type = linear
27
28 [Altitude Covariate]
29 type = linear
30 covariates = altitude-full.dat
31
32 [Gradient Covariate]
33 type = linear
34 covariates = gradient-full.dat
35
36 [INLA parameters]
37 type = INLA
38 h = 0.001
```

The data file *data−full.dat* has the following format

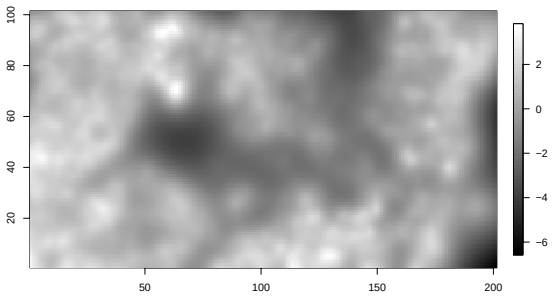
$$i \quad j \quad |s_{ij}| \quad y_{ij}$$

where $i = 0, \ldots, n_{row} - 1$ is the row index and $j = 0, \ldots, n_{col} - 1$ is the column index. Notice then, that for data observed on a grid the data file has four columns instead of three (see Appendix A.1.2). The data are stored by row, so that the first n_{row} lines of the data file refer to row 1, the second n_{row} lines to row 2 etc. The same also for the covariate files.

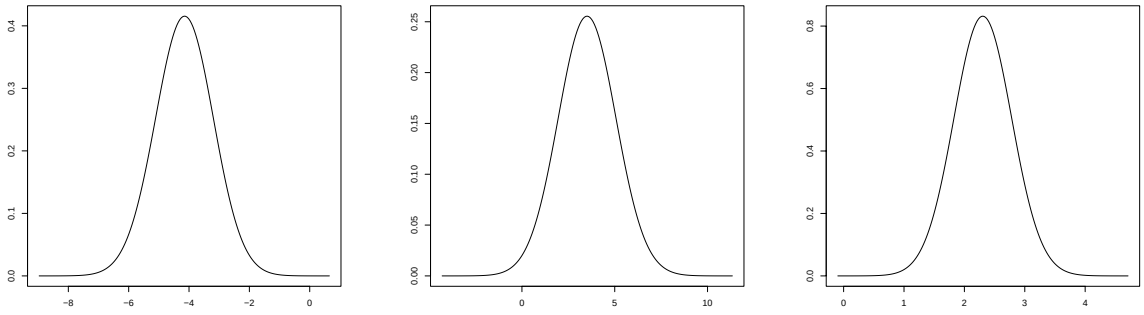
Notice also that it is required for the user to specify the number of rows and columns in the data set (lines 12-13 and 19-20). For grid observed data, the fields ***nrow*** and ***ncol*** substitute the field ***n*** which we have used in all previous examples. The prior model for the spatial effect is defined in line 21.

The results are displayed in Figure 16. Panel (a) shows the posterior mean of the structured spatial effect. Following is the R code used to produce Figure 16(a):

```
> library(fields)
```



(a)



(b) (c) (d)

Figure 16: LGCP example: (a) posterior mean of the spatial effect $f_s(\cdot)$, (b)-(d) posterior marginals for μ, β_1 and β_2

```
> xcoord=5*seq(0,200)
> ycoord=5*seq(0,100)
> space=read.table("results-linear0/spatial-smoother/summary.dat")
> image.plot(xcoord,ycoord,matrix(space[,3],ncol=101,byrow=F),
             col=gray(seq(0,1,len=1000)))
```

Panels (b)-(d) show the posterior marginal distributions for the parameters μ, β_1 and β_2.

The graph of the full model for this example contains 40605 nodes, this makes the computation procedures heavier that for all other examples considered here. The computational time required to solve the model grows then to about 1 hour and 30 minutes on Machine 2. We have not run the model on Machine 1.

3.11 A longitudinal study example - Forest health data

Our last example is a longitudinal study on forest health. The aim of the study if to identify potential factors influencing the health status of the trees. In addition to covariates characterising a tree and its stand, spatial and temporal information are also available. The example is taken from Kneib and Fahrmeir (2008), an earlier version of the data set is analysed in Kneib and Fahrmeir (2006).

Example 11 *The data have been collected annually in a visual forest health inventories between 1983 and 2004 in a northern Bavarian district. There are 83 observations plots within an area of around 15 squared kilometres.*

Every year, in some of the 83 observations plots the health status of the tree y_{it}, $i = 0, \ldots, 83$, $t = 0, \ldots, 21$, *is registered. Not all plots are observed every year, so the data set has in total* $n_d = 1796$ *observations. In the original data set there are 9 categories for tree health, anyway, here we consider only two: healthy or non-healthy. Together with the tree health status, several covariates are registered year after year at the different observation plot. All covariates are summarised in Table 1. Moreover the location of each registration plot* s_i

Covariate	Description
Age	age of the stand in years (continuous between 7 and 234 years)
elevation	elevation above the sea level (continuous, between 250 and 480 meters)
inclination	inclination of the terrain in percent (continuous between 0 and 1)
soil	depth of soil level (continuous, between 9 and 51 cm)
ph	ph-value in 0-2cm depth (continuous, between 3.28 and 5.05)
canopy	density of forest canopy in percent (continuous, between 0 and 1)
stand	type of stand (categorical, 3 categories)
fertilisation	fertilisation (categorical: yes or no)
humus	thickness of humus (categorical, 5 categories)
moisture	level of moisture (categorical, 3 categories)
saturation	base saturation (ordinal)

Table 1: Forest health data: description of covariates.

is known. The spatial distribution of the locations is displayed in Figure 17.

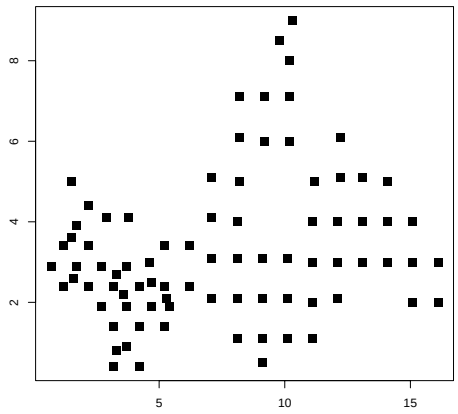


Figure 17: Forest health example: location of the 83 observation plots.

The likelihood of the data is binomial:

$$y_{it}|\eta_{it} \sim \textit{Bin}(p_{it})$$

with logit link

$$p_{it} = \frac{\exp(\eta_{it})}{1+\exp(\eta_{it})} \quad i = 0,\ldots,82, t = 0,\ldots,21.$$

Following Kneib and Fahrmeir (2008) we model the latent variables as:

$$\eta_{it} = \mu + f_0(\textit{age}_{it}) + f_1(\textit{inclination}_i) + f_2(\textit{canopy}_{it}) + f_{\textit{time}}(t) + f_s(s_i) + f_u(s_i) + \boldsymbol{z}_{it}^T\boldsymbol{\beta} \qquad (18)$$

where $f_0(\cdot)$, $f_1(\cdot)$, $f_2(\cdot)$ are the semiparametric effect of age of the tree, inclination and canopy of the location respectively, while $f_{\textit{time}}(\cdot)$ is the non parametric effect of time. Each semiparametric function is modelled as a RW2 with unknown precision parameter. The vector $\boldsymbol{z}_{it}^T$ includes all covariates in Table 2 not mentioned before which are assumed to have a linear effect. Finally $f_s(\cdot)$ and $f_u(\cdot)$ indicate the structured spatial effect and the unstructured one.

We models the spatial structured effect as the intrinsic GMRF in equation (12). We build the graph for such a model by considering two observation plots as neighbours if their distance is less than 1200 meters. The spatial unstructured effect is modelled as a series of uncorrelated Gaussian random variable.

We can cast the model in (18) in the general formulation in equation (2) by defining a new index $r = (i,t)$, $r = 0,\ldots,n_d - 1$, and rewriting the model as

$$\eta_r = \mu + f_0(\textit{age}_r) + f_1(\textit{inclination}_r) + f_2(\textit{canopy}_r) + f_{\textit{time}}(r) + f_s(s_r) + f_u(s_r) + \boldsymbol{z}_r^T\boldsymbol{\beta} \qquad (19)$$

The above model has six precision hyperparameters $\boldsymbol{\theta} = (\log\lambda_0, \log\lambda_1, \log\lambda_2, \log\lambda_{\textit{time}}, \log\lambda_s, \log\lambda_u)$, each is given a vague LogGamma prior.

We report part of the `ini` file which defines the model. We have omitted the definition of almost all covariates with linear effect.

```
[Forest damage]
type=problem
dir=results-%d

[predictor term]
type=predictor
n=1796
initial = 10
fixed=1

[Data]
type=data
likelihood=binomial
filename=damage.dat

[spatial]
type=ffield
model=besag
graph=forest.gra
covariates=spatial.covariate
diagonal = 0.00001
constraint = 1
initial = -3.346165
parameters = 1 0.001

```

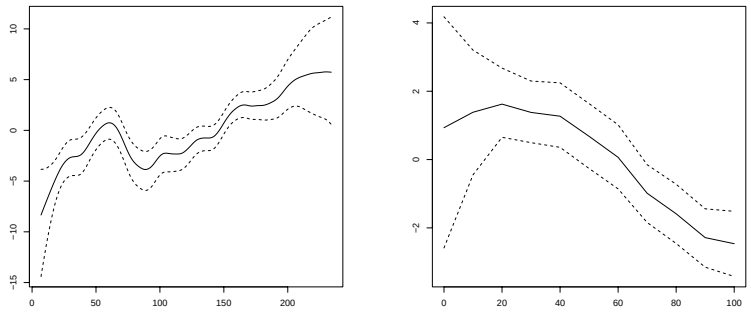
```
[spatial-unstruct]
type=ffield
model=iid
n=83
covariates=spatial.covariate
diagonal = 0.00001
constraint = 1
initial = 7.324791
parameters = 1 1

[age]
type = ffield
model = rw2
covariates = age.covariate
locations=age.location
diagonal = 0.0001
constraint = 1
initial = 5.674807
parameters = 1 0.001
quantiles = 0.025 0.975

[canopy]
type = ffield
model = rw2
covariates = canopy.covariate
locations=canopy.location
diagonal = 0.0001
constraint = 1
initial = 13.763045
parameters = 1 0.001
quantiles = 0.025 0.975

[inclination]
type = ffield
model = rw2
covariates = inclination.covariate
n=47
diagonal = 0.0001
constraint = 1
initial = 6.422709
parameters = 1 0.001
quantiles = 0.025 0.975

[time]
type = ffield
model = rw2
covariates = year.covariate
locations=year.location
diagonal = 0.0001
constraint = 1
initial = 1.211905
parameters = 1 0.001
quantiles = 0.025 0.975

[common mean]
type=linear

```

```
[soil]
type = linear
covariates = soil.cov
.
.
.
[INLA parameters]
type = INLA
int_strategy = CCD;
h = 1.0e-2;
```

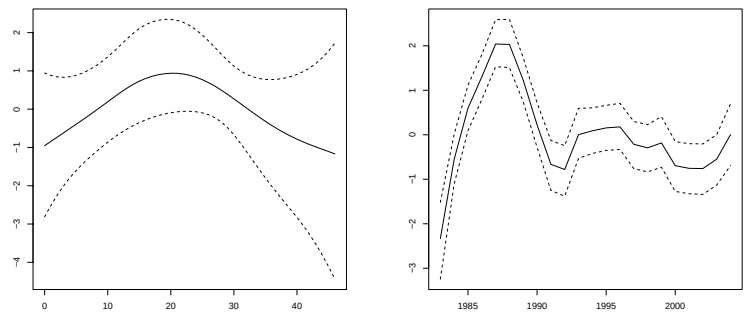
Notice that when using the `inla` program we treat all covariates, including space and time in the same way. All covariates files have the same structure.

Again we use the CCD strategy in order to integrate out the uncertainty about the hyperparameters $\boldsymbol{\theta}$. Given the high dimension of the hyperparameters space, the CCD strategy gives a much lower computation time if compared to the grid strategy. We have compared the results coming from the two integration strategies and the differences are irrelevant.

In Figure 18 the results about the semiparametric effects are displayed. The posterior mean is plotted within 0.025 and 0.975 posterior quantiles. The results agree very well with those found by Kneib and Fahrmeir (2008).



(a) Age of the tree (b) Canopy



(c) Inclination (d) Time

Figure 18: Results for the forest health example, semiparametric effect of covariates, posterior mean within 0.025 and 0.975 quantiles: age of the tree, panel (a), canopy, panel (b), inclination panel (c) and time panel (d).

The model runs in around 9 minutes on Machine 1 and around 4 minutes on Machine 2. Much of the time is used by the optimiser to find the maximum of $\widetilde{\pi}(\boldsymbol{\theta}|\boldsymbol{y})$ and to compute the Hessian at the modal configuration. When the hyperparameter space is high dimensional it is possible that the optimiser fails to succeed at a first attempt. The problem is usually solved by running the `inla` program again starting from different initial values for the hyperparameters. It is, usually, a good idea to start from the best configuration found during the previous run.

If one is interested in spatial prediction of tree health outside the observation plots, the spatial model in (12) is not very useful. We could instead use a second order random walk defined on a regular grid (Rue and Held, 2005, Sec 3.4.2) built as following. We divide the region of interest in $n_{row} \times n_{col}$ cells, with $n_{row} = 50$ and $n_{col} = 100$. We then build a new covariate file, *spatial−covariate−rw2.dat*, where, to each data point y_r are assigned two indexes n^r_{row} and n^r_{col} indicating its the location of the data on the $n_{row} \times n_{col}$ grid.

The code for the `ini` file substituting section *[spatial]* (lines 16-24) and *[spatial −unstruct]* (lines 26-34) is the following:

```
1 [spatial]
2 type=ffield
3 model=rw2d
4 covariates=spatial-covariate-rw2.dat
5 nrow=50
6 ncol=100
7 constraint=1
8 parameters = 1 0.001
9 initial= -1.570568
```

The new model has one hyperparameter less than the previous one since no spatial unstructured effect is present, but the number of nodes in the latent field $\boldsymbol{x}$ is increased, therefore running the new model will take longer time.

The results for the spatial effect in the new model is displayed in Figure 19. The non parametric effects of the other covariates do not change significantly.

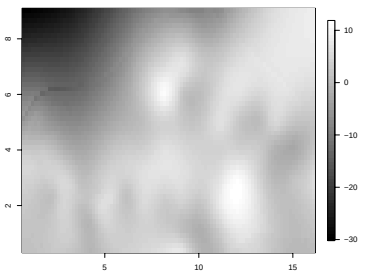


Figure 19: Posterior mean estimate for the spatial effect modelled as a RW2d

Kneib and Fahrmeir (2008) propose to include in the model for the latent variable an interaction between the age of the tree and the calendar time, so that the model becomes:

$$\begin{aligned} \eta_{it} = \quad & \mu + f_1(\text{inclination}_i) + f_2(\text{canopy}_{it}) \\ & f_3(t, \text{age}_{it}) + f_s(s_i) + f_u(s_i) + \boldsymbol{z}^T_{it}\boldsymbol{\beta} \end{aligned} \tag{20}$$

where the spatial effect $f_s(\cdot)$ is modelled as in (12) and $f_4(\cdot)$ is the interaction effect between time and age of the tree modelled as a RW2d.

We can include the term $f_4(\cdot)$ in equation (20) in a similar way as we did earlier in this same example for the RW2d spatial effect. We just create a new covariate file, *year.age−covariate*, with the format

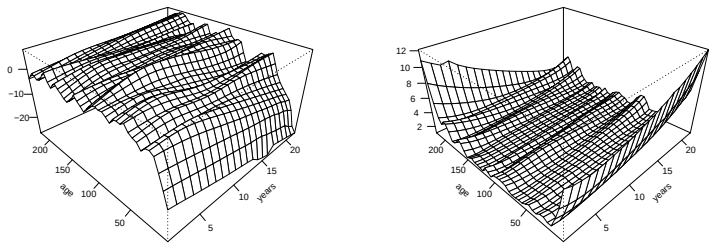
$$r \quad t \quad \text{age}_r$$

where both time and age are recorded, and delete from the `ini` on page 42 section *[age]* and *[time]* while adding the the following lines:

```
[year−age interaction]
type=ffield
model=rw2d
covariates=year.age−covariate
nrow=22
ncol=223
constraint=1
diagonal = 0.01
parameters = 1 0.01
initial= 2.025712
```

The new model has 5 hyperparameters and the total number of nodes in the latent field is 6939. We run the model on Machine 2 and the computation time was around 30 minutes using a CCD integration strategy.

The posterior mean and standard deviation of the interaction effect are displayed in Figure 20, panel (a) and (b) respectively.



(a) Posterior mean (b) Posterior standard deviation

Figure 20: Interaction effect between age of the tree and calendar time in Model (20). Panel (a) posterior mean, panel (b) posterior standard deviation.

4 Model assessment and model choice

For the material in this section refer to the revised version of Rue et al. (2007)

4.1 Marginal Likelihood

The marginal likelihood for a certain model $\mathcal{M}$, defined as

$$\pi(\boldsymbol{y}|\mathcal{M}) = \int \pi(\boldsymbol{y}, \boldsymbol{y}, \boldsymbol{\theta}|\mathcal{M})\, d\boldsymbol{x}\, d\boldsymbol{\theta}$$

can be used as a basis for model comparison. The Bayes factor for two competing models is in fact defined as

$$\mathcal{B}(i,j) = \frac{\pi(\mathcal{M}_i|\boldsymbol{y})\pi(\mathcal{M}_i)}{\pi(\mathcal{M}_j|\boldsymbol{y})\pi(\mathcal{M}_j)} \tag{21}$$

If we choose the models to be apriori equiprobable,$\pi(\mathcal{M}_1) = \cdots = \pi(\mathcal{M}_K)$, then the Bayes factor reduces to

$$\mathcal{B}(i,j) = \frac{\pi(\boldsymbol{y}|\mathcal{M}_i)}{\pi(\boldsymbol{y}|\mathcal{M}_j)}$$

Hence, we can compare models by comparing their marginal likelihood $\pi(\boldsymbol{y}|\mathcal{M}_k)$.

NB: For (21) to be well defined it is necessary for the prior of the latent field $\pi(\boldsymbol{x}|\boldsymbol{\theta})$ to be **proper**. For intrinsic models, in fact, there is an arbitrary missing constant which cannot be determined, see for example Gelfand (1996).

Using the INLA approach, the marginal likelihood for a certain model $\mathcal{M}$, $\pi(\boldsymbol{y}|\mathcal{M})$ can be computed as the normalising constant of $\widetilde{\pi}(\boldsymbol{\theta}|\boldsymbol{y})$ using two different approaches:

1. Via numerical integration of $\widetilde{\pi}(\boldsymbol{\theta}|\boldsymbol{y})$
2. Assuming a Gaussian approximation to $\widetilde{\pi}(\boldsymbol{\theta}|\boldsymbol{y})$

see Rue et al. (2007, Revised version) for details. Using the `inla` program it is enough to set the ***mlik*** flag in the section ***type***=*problem* to 1 for the marginal likelihood to be computed. The first approximation (which is more accurate) is computed only if in the ***type*** = *INLA* section, ***int_strategy*** = *CCD* is selected.

Example 12 *We want to check which one, between a Gaussian and a Student-t error is more appropriate to describe the dollar-pound exchange rate data set in Example 3.*

It is enough to add to the first section of the corresponding `ini` the line

```
1 mlik = 1
```

The output is stored in the file *results/marginal-likelihood/marginal-likelihood.dat* which contains both approximations for the log marginal likelihood of the model, $\log \widetilde{\pi}(\boldsymbol{y}|\mathcal{M})$.

For the Gaussian error model in (8) we have

```
log marginal-likelihood (integration): -933.258
log marginal-likelihood (Gaussian): -933.324
```

while for the Student-t model in (10) the result is

```
log marginal-likelihood (integration): -934.997
log marginal-likelihood (Gaussian): -935.233
```

In this case the Gaussian error model is preferred.

Note that in the volatility model example we have considered, the prior for the latent model $\boldsymbol{x}$ is an autoregressive model of order one. This is a proper model, therefore the marginal likelihood gives a reasonable tool for model comparison. If we would have chosen, for example, a RW1 model (or any other intrinsic distribution) as prior for the latent volatility the marginal likelihood computed would have been meaningless.

4.2 Deviance Information Criterion (DIC)

Deviance information criterion (DIC) is a criterion for comparing complex hierarchical models introduced in Spiegelhalter et al. (2002) and defined as:

$$\text{DIC} = \bar{D} + \text{p}_D \tag{22}$$

where $\bar{D}$ is the posterior mean of the deviance of the model and p_D is the effective number of parameters in the model, see Spiegelhalter et al. (2002). Details on how to compute the quantities in equation (22) using the INLA approach are described in Rue et al. (2007, Revised version).

To compute the DIC using the `inla` program it is enough to set the flag *dic* in the section *type=problem* to 1.

For example, if we want to compute the DIC for the two disease mapping models considered in Section 3.6, it is enough to add to the first section of the corresponding `ini` file the line:

```
2 dic = 1
```

The result is printed in the output of the `inla` program, moreover it is stored in the file:
results/dic/dic.dat
which contains four quantities: the mean of the deviance, the deviance of the mean, the effective number of parameters and the DIC.

For model (14) in Section 3.6, which assumes a non-linear effect of the covariate, the *dic.dat* file is the following:

```
mean of the deviance: 2652.86
deviance of the mean: 2563.22
effective number of parameters: 89.6438
dic: 2742.51
```

We can compute the DIC also for model (15), which assumes a linear effect of the covariate, obtaining:

```
mean of the deviance: 2655.87
deviance of the mean: 2552.33
effective number of parameters: 103.542
dic: 2759.42
```

The difference in DIC values is 16.91 in favour of model (14) which suggests that the effect of the exposure covariate is better represented by a non linear function.

4.3 Predictive measures

Predictive measures can be used both to validate and to compare models (Gelfand, 1996; Gelman et al., 2004) and as a device to detect possible outliers or surprising observations (Pettit and Young, 1990). Using `inla` it is possible to compute Conditional Predictive Ordinates (CPOs) and Probability Integral Transforms (PIT).

Conditional predictive ordinates (CPOs) are defined as:

$$\text{CPO}_i = \pi(y_i|\boldsymbol{y}_{-i})$$

where the subfix $-i$ indicates that element i of the vector is removed. CPOs are discussed among others by Pettit (1990) and Gelfand (1996).

Unusually small or large values of CPO_i indicate a surprising observation. Anyway, before being compared, the CPOs have to be calibrated. One of the possible calibration procedures is to compute the probability integral transform

$$\text{PIT}_i = \text{Prob}(y_i^{\text{new}} \leq y_i|\boldsymbol{y}_{-i})$$

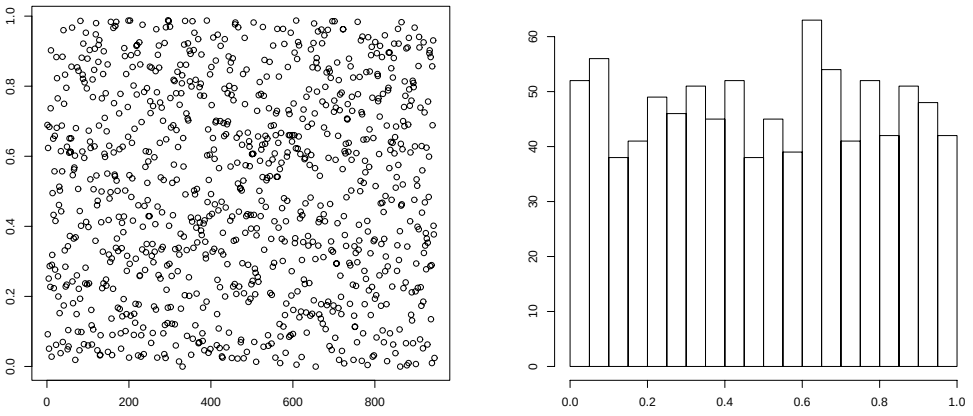
see also Gneiting and Raftery (2007). An unusual large or small value indicates possible outliers. Furthermore, an histogram of the PITs far from uniform might indicate a questionable model (Czado et al., 2007).

In the `inla` program to compute CPOs and PITs it is sufficient to add in the *type=problem* section of the `ini` file the line

```
3 cpo = 1
```

The results will be stored in the *results/cpo/* directory in the two files *cpo.dat* and *pit.dat*.

As an example we consider the volatility model with Gaussian observation in Example 3. The corresponding PIT values are plotted in Figure 21(a). There are three observation whose PIT is close to 0, namely 331 656 and 862. Figure 21(b) displays the histogram of the PIT values which is reasonably close to uniform.



(a) (b)

Figure 21: PIT values for the volatility model with Gaussian observation in Example 3 (panel (a)) and corresponding histogram (panel (b)).

NB: In some cases the value of PIT_i might result exactly equal to 1 or 0, this simply means that its real value is very close to 1 or 0.

References

Brezger, A., Kneib, T., and Lang, S. (2003). *BayesX: Software for Bayesian inference*. Department of statistics, University of Munich, version 1.1 edition. http://www.stat.uni-muenchen.de/∼lang/bayesx.

Chaudhuri, P. and Marron, J. S. (1999). SiZer for exploration of structures in curves. *Journal of the American Statistical Association*, 94:807–823.

Chib, S., Nardari, F., and Shepard, N. (2002). Markov chain Monte Carlo methods for stochastic volatility models. *Journal of Econometrics*, 108:281–316.

Czado, C., Gneiting, T., and Held, L. (2007). Predictive model assessment for count data. Technical Report 518, University of Washington, Dep. of Statistics.

Erästö, P. (2005). *Studies in trend detection of scatter plots with visualization*. PhD thesis, Department of Mathematics and Statistics, University of Helsinki, Finland.

Fahrmeir, L. and Tutz, G. (2001). *Multivariate Statistical Modelling Based on Generalized Linear Models*. Springer-Verlag, Berlin, 2nd edition.

Gelfand, A. E. (1996). Model determination using sampling-based methods. In Gilks, W. R., Richardson, S., and Spiegelhalter, D. J., editors, *Markov Chain Monte Carlo in practice*. Chapman and Hall.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004). *Bayesian Data Analysis*. Texts in Statistical Science Series. Chapman & Hall/CRC, Boca Raton, FL, 2nd edition.

Gneiting, T. and Raftery, A. E. (2007). Stricly proper scoring rules, prediction and estimation. *Journal of the American Statistical Association*, 102:359–378.

Hastie, T. J. and Tibshirani, R. J. (1990). *Generalized Additive Models*, volume 43 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, London.

Kammand, E. E. and Wand, M. P. (2003). Geoadditive models. *Journal of Royal Statistical Society C*, 52:1–18.

Kammann, E. E. and Wand, M. P. (2003). Geoadditive models. *Journal of the Royal Statistical Society, Series C*, 52(1):1–18.

Kandala, N. B., Lang, S., Klasen, S., and Fahrmeir, L. (2001). Semiparametric analysis of the socio-demographic and spatial determinants of undernutrition in two african countries. *Research in Official Statistics*, 1:81–100.

Kneib, T. and Fahrmeir, L. (2006). Structured additive regression for categorical space-time data: A mixed model approach. *Biometrics*, 62:109–118.

Kneib, T. and Fahrmeir, L. (2008). A space-time study on forest health. In Chandler, R. E. and Scott, M., editors, *Statistical Methods for Trend Detection and Analysis in the Environmental Sciences*. Wiley. (to appear).

Kneib, T., Lang, S., and Brezger, A. (2004). Bayesian semiparametric regression based on mcmc techniques: A tutorial. Technical report, Department of statistics, University of Munich.

Lin, X. and Zhang, D. (1999). Inference for generalized additive mixed models by using smoothing splines. *Journal of Royal Statistical Society B*, 61:381–400.

Martino, S. (2007). Approximate bayesian inference for multivariate stochastic volatility models. Technical report, Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway.

Møller, J. and Waagepetersen, R. (2003). *Statistical inference and simulation for spatial point processes*, volume 100 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, London.

Pettit, L. I. (1990). The conditional predictive ordinate for the normal distribution. *Journal of the Royal Statistical Society, Series B*, 1(52):175–184.

Pettit, L. I. and Young, K. D. (1990). Measuring the effect of observations on bayes factors. *Biomatrika*, 77:455–466.

Rue, H. and Held, L. (2005). *Gaussian Markov Random Fields: Theory and Applications*, volume 104 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, London.

Rue, H. and Martino, S. (2006). Approximate Bayesian inference for hierarchical Gaussian Markov random fields models. *Journal of Statistical Planning and Inference*, 137:3177–3192.

Rue, H., Martino, S., and Chopin, N. (2007). Approximate bayesian inference for latent gaussian models using integrated nested laplace approximations. Statistics Report No. 1, Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway.

Shephard, N. and Pitt, M. K. (1997). Likelihood analysis of non-Gaussian measurement time series. *Biometrika*, 84(3):653–667.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and van der Linde, A. (2002). Bayesian measures of model complexity and fit. *Journal Royal Statistical Society series B*, 64(4):583–639.

Taylor, S. J. (1986). *Modelling Stochastic volatility*. John Wiley.

Waagepetersen, R. P. (2006). An estimating function approach to inference for inhomogeneous Neyman-Scott processes. *Biometrics*, xx(xx):xx–xx. (to appear).

Wecker, W. E. and Ansley, C. F. (1983). The signal extraction approach to nonlinear regression and spline smoothing. *Journal of the American Statistical Association*, 78(381):81–89.

A Reference manual for the `inla` program

A.1 Structure of the `ini` file

The `ini` file describes the model and sets some additional parameters to be passed to the GMRFLib library. It is divided in several sections. Each section starts with a tag written between squared brackets (*[tag]*) which is simply a user defined name for the section itself.

Each section contains the field *type* which determines the role of the section in the problem definition and also the structure of the section itself. The six different types of section are described in details below.

A.1.1 The *type=problem* section

This sections specifies some global parameters which are valid for the whole problem. It consists of the following fields:

dir: A string indicating the name of the directory where the results are stored. The directory is created when the `inla` program is run. The directory name can include `%d`

hyperparameters: A Boolean variable indicating whether or not to compute the marginals for the hyperparameters $\boldsymbol{\theta}$ of the model.

Default = 0

summary: A Boolean variable indicating whether or not to output a short summary of the posterior density for *all* the nodes in the GMRF $\boldsymbol{x}$. Currently the summary contains the posterior mean and standard deviation.

Default = 1

density: A Boolean variable indicating whether or not to output the marginal densities for *all* nodes in the latent GMRF $\boldsymbol{x}$.

Default = 1

quantiles: A list of maximum 10 quantiles, $p(0), p(1), \ldots,$ to compute for each posterior marginal. The function returns, for each posterior marginal, the values $x(0), x(1), \ldots$ such that

$$\text{Prob}(X < x(p)) = p$$

Default: Empty

percentiles: A list of maximum 10 percentiles, $x(0), x(1), \ldots,$ to compute for each posterior marginal. The function returns, for each posterior marginal, the probabilities $\text{Prob}(X < x(p))$.

Default: Empty

smtp: A string indicating which type of solver for sparse matrices should be used. The available choices are:

- *GMRFLib_SMTP_BAND* Lapack's band-solver. This is optimal for band matrices
- *GMRFLib_SMTP_TAUCS* The solver in the TAUCS-library. This is generic for all kind of sparse matrices.

Default: *GMRFLib_SMTP_TAUCS*

dic: A Boolean variable indicating whether or not to compute the deviance information criterion (DIC) for the model.

Default: 0

cpo: A Boolean variable indicating whether or not to compute the conditional predictive ordinates for the model

Default: 0

mlik A Boolean variable indicating whether or not to compute the marginal likelihood for the model

NB: this quantity is meaningful ONLY if in all the sections *type= ffield* present in the `ini` file *model=ar1* is chosen.

Default: 0

A.1.2 The *type=data* section

This section specifies the model for the likelihood of the data $\pi(y_i|\eta_i, \boldsymbol{\theta}_1)$ in equation (1). It consists of the following fields:

likelihood : A string indicating the name of the required likelihood model. The available choices are listed in Table 2.

prior: Prior distribution for hyperparameter $\boldsymbol{\theta}_1$ in one-parameter likelihood models:
likelihood =*gaussian* or *stochvol* $-t$.

If *likelihood* =*gaussian* then $\theta_1 = \log \lambda_y$ and the corresponding prior is a LogGamma(a, b)[1].
If *likelihood* =*stochvol*$-t$ then $\theta_1 = \nu'$ (see Table 2) and the corresponding prior is a 0 mean Gaussian distribution $\mathcal{N}(0, 1/\text{prec})$.

initial : Initial value for hyperparameter $\boldsymbol{\theta}_1$ in one-parameter likelihood models:
likelihood =*gaussian* or *stochvol* $-t$.

Initial value for $\log \lambda_y$ (if *likelihood* =*gaussian*) or for ν' (if *likelihood* =*stochvol*$-t$).

parameters: Parameters for $\pi(\boldsymbol{\theta}_1)$ in one-parameter likelihood models:
likelihood =*gaussian* or *stochvol* $-t$.

If *likelihood* =*gaussian*: parameters a and b for the LogGamma prior of the log-precision $\log \lambda_y$.
If *likelihood* = *stochvol_t* : parameter prec for the Gaussian prior of the ν'.

prior0: Prior distribution for the first hyperparameter θ_{11} in vector $\boldsymbol{\theta}_1 = (\theta_{11}, \theta_{12})$ in two-parameter likelihood models:
likelihood =*T* or *stochvol_nig* .

If *likelihood* =*T* then $\theta_{11} = \log \lambda_y$ and the corresponding prior is a LogGamma(a, b).
If *likelihood* = *stochvol_nig* then $\theta_{11} = \beta$ (see Table 2) and the corresponding prior is a 0 mean Gaussian distribution $\mathcal{N}(0, 1/\text{prec})$.

initial0 : Initial value for hyperparameter θ_{11} in two-parameter likelihood models:
likelihood =*T* or *stochvol_nig* .

Initial value for $\log \lambda_y$ (if *likelihood* =*T*) or for β (if *likelihood* = *stochvol_nig*).

parameters0: Parameters for $\pi(\theta_{11})$ in two-parameter likelihood models:
likelihood =*T* or *stochvol_nig* .

If *likelihood* =*T*: parameters a and b for the LogGamma prior of the log-precision $\log \lambda_y$.
If *likelihood* = *stochvol_nig* : parameter prec for the Gaussian prior of the β.

prior1: Prior distribution for the second hyperparameter θ_{12} in vector $\boldsymbol{\theta}_1 = (\theta_{11}, \theta_{12})$ in two-parameter likelihood models:
likelihood =*T* or *stochvol_nig* .

If *likelihood* =*T* then $\theta_{12} = \nu'$ and the corresponding prior is a 0 mean Gaussian distribution $\mathcal{N}(0, 1/\text{prec})$.
If *likelihood* = *stochvol_nig* then $\theta_{12} = \psi'$ and the corresponding prior is a 0 mean Gaussian distribution $\mathcal{N}(0, 1/\text{prec})$.

initial1 : Initial value for hyperparameter θ_{12} in two-parameter likelihood models:
likelihood =*T* or *stochvol_nig* .

Initial value for ν' (if *likelihood* =*T*) or for ψ' (if *likelihood* = *stochvol_nig*).

parameters1: Parameters for $\pi(\theta_{12})$ in two-parameter likelihood models:
likelihood =*T* or *stochvol_nig* .

If *likelihood* =*T*: parameter prec for the Gaussian prior of ν'.
If *likelihood* = *stochvol_nig* : parameter prec for the Gaussian prior of ψ'.

[1] See Appendix B for a definition.

fixed: A Boolean variable indicating whether the hyperparameters of the likelihood model are fixed or random.

Default: 0

filename: The name of the file which contains the data for the model. The format of the file depends on the likelihood model chosen and is indicated in Table 2

Model name	Distribution	Link function	Parameters $\boldsymbol{\theta}_1$	Input File format	Input File format (on a grid)
gaussian	$y_i \sim \mathcal{N}(\mu_i, \frac{\lambda_y^{-1}}{w_i})$	$\mu_i = \eta_i$	$\theta_1 = \log \lambda_y$	$i \quad w_i \quad y_i$	$i \quad j \quad w_{ij} \quad y_{ij}$
poisson	$y_i \sim \text{Po}(E_i \lambda_i)$	$\lambda_i = \exp(\eta_i)$	-	$i \quad E_i \quad y_i$	$i \quad j \quad E_{ij} \quad y_{ij}$
binomial	$y_i \sim \text{Bin}(n_i, p_i)$	$p_i = \frac{\exp(\eta_i)}{(1+\exp(\eta_i))}$	-	$i \quad n_i \quad y_i$	$i \quad j \quad n_{ij} \quad y_{ij}$
T	$y_i = x_i + \frac{1}{\sqrt{\lambda_y w_i}}\mathcal{T}$ $\mathcal{T} \sim t_\nu$ (*)	$x_i = \eta_i$	$\boldsymbol{\theta}_1 = (\log \lambda_y, \nu')$ $\nu' = \log(\nu - 2)$	$i \quad w_i \quad y_i$	$i \quad j \quad w_{ij} \quad y_{ij}$
stochvol	$y_i \sim \mathcal{N}(0, \sigma_i^2)$	$\sigma_i = \exp(\eta_i/2)$	-	$i \quad y_i$	-
stochvol_t	$y_i = \sigma_i \mathcal{T}$ $\mathcal{T} \sim t_\nu$ (*)	$\sigma_i = \exp(\eta_i/2)$	$\theta_1 = \nu'$ $\nu' = \log(\nu - 2)$	$i \quad y_i$	-
stochvol_nig	$y_i = \sigma_i \times \mathcal{T}$ $\mathcal{T} \sim \text{NIG}(\beta, \psi)$(**)	$\sigma_i = \exp(\eta_i/2)$	$\boldsymbol{\theta}_1 = (\beta, \psi')$ $\psi' = \log(\psi - 1)$	$i \quad y_i$	-

(*) t_ν is a scaled Student-t distribution, see Appendix B for definition

(**) See Appendix B for definition of a NIG distribution.

Table 2: Likelihood models supported in the `inla` program.

Hyperparameter	Prior distribution	Default param
Log-Precision $\log \lambda_y$	LogGamma(a, b)(*)	$a = 1, b = 0.001$
$\nu',\ \psi',\ \beta$	$\mathcal{N}(0, 1/\text{prec})$	prec $= 0.001$

(*)) See Appendix B for definition of a LOgGamma distribution.

Table 3: Prior distributions for the hyperparameters in the likelihood models

A.1.3 The *type=predictor* section

This section defines the model for the unstructured term u_i in equation (2). The `inla` program requires a section of *type=predictor* to always be present. It consists of the following fields:

prior: Name of the prior for the log-precision parameter $\log \lambda_\eta$. At the moment only the LogGamma(a, b) prior is implemented.

Default: *loggamma*

parameters: Parameters a and b for the LogGamma prior of the log-precision λ_η.
Default: $a = 1.0$ and $b = 0.001$

fixed : A Boolean variable indicating whether the precision parameter $\log \lambda_\eta$ is fixed or random.
Default: 0.

initial : Starting value for $\log \lambda_\eta$

n: Length of the latent variable vector $\boldsymbol{\eta}$. Either *n*, or *nrow* and *ncol* are required.

nrow: Number of rows of the latent variable vector $\boldsymbol{\eta}$. Either *n*, or *nrow* and *ncol* are required.

ncol: Number of columns of the latent variable vector $\boldsymbol{\eta}$. Either *n*, or *nrow* and *ncol* are required.

compute: A Boolean variable indicating whether or not the marginals for vector $\boldsymbol{\eta}$ have to be computed.
Default: 0 section

summary:A Boolean variable indicating whether or not to output a short summary of the posterior density for $\boldsymbol{\eta}$.
Default: *compute*

density : A Boolean variable indicating whether or not to output the marginal densities for $\boldsymbol{\eta}$.
Default: *compute*

quantiles : A list of maximum 10 quantiles, $p(0), p(1), \ldots$, to compute for each node in $\boldsymbol{\eta}$.
Default: Empty

percentiles :A list of maximum 10 percentiles, $x(0), x(1), \ldots$, to compute for each node in $\boldsymbol{\eta}$.
Default: Empty

A.1.4 The *type= ffield* type section

A section of *type= ffield* specifies the model for one of the function f in equation (2). Hence, in a `ini` file there must be n_f sections of *type= ffield* . Each *type= ffield* section consists of the following fields:

model: A a string indicating the name of the chosen model. All available choices are listed in Table 4.

prior: Name of the prior for the log-precision parameter $\log \lambda_f$. At the moment only the LogGamma(a, b) prior is implemented (not in use if *model=ar1*)
Default: *loggamma*

parameters: Parameters a and b for the LogGamma prior of the log-precision $\log \lambda_f$ (not in use if *model=ar1*)
Default: $a = 1.0$ and $b = 0.001$

initial : Starting value for $\log \lambda_f$ (not in use if *model=ar1*)

prior0: Name of the prior for the log-precision parameter λ_f if *model=ar1*. At the moment only the LogGamma(a, b) prior is implemented
Default: *loggamma*

prior1: Name of the prior for the precision parameter κ if *model=ar1*. At the moment only the Gaussian$(0, \text{prec}_\kappa)$ prior is implemented
Default: *gaussian*

parameters0:Parameters a and b for the LogGamma prior of the precision $\log \lambda_f$ (**only** for *model=ar1*)
Default: $a = 1.0$ and $b = 0.001$

parameters1: Parameter prec_κ for parameter κ (**only** for *model=ar1*)
Default: $\text{prec}_k = 0.001$

initial0 : Starting value for $\log \lambda_f$ (**only** for *model=ar1*)

initial1 : Starting value for κ (**only** for *model=ar1*)

Model Type	Model Name	Parameters	Reference
Independent random noise	*iid*	log-precision $\log \lambda_f$	
Random Walk of order 1	*rw1*	log-precision $\log \lambda_f$	(Rue and Held, 2005, Ch. 3.3.1)
Random Walk of order 2	*rw2*	log-precision $\log \lambda_f$	(Rue and Held, 2005, Ch. 3.4.1)
First order Intrinsic GMRF on a irregular lattice	*besag*	log-precision $\log \lambda_f$	(Rue and Held, 2005, Ch. 3.3.2)
Continuous random walk	*crw2*	log-precision $\log \lambda_f$	(Rue and Held, 2005, Ch. 3.5)
Autoregressive of order 1 $x_t = \phi x_{t-1} + \epsilon_t$	*ar1*	log-precision $\log \lambda_f$ $\kappa = \text{logit}\frac{\phi+1}{2}$	(Rue and Held, 2005, Ch. 1.1)
User defined precision matrix	*generic*	log-precision $\log \lambda_f$	(see Example 5)

Table 4: Models for the *type= ffield* section implemented in the `inla` program.

Hyperparameter	Prior distribution	Default param
Log-Precision $\log \lambda_f$	LogGamma(a, b)	
κ (only for AR1)	$\mathcal{N}(0, 1/\text{prec}_k)$	$\text{prec}_k = 0.001$

Table 5: Prior distributions for the hyperparameters

rankdef: A number indicating the rank deficiency of the user defined $\boldsymbol{Q}$ matrix (0only used if *model=generic*).
Default: 0.

fixed : A Boolean variable indicating whether the precision parameter λ_f is fixed or random.
Default: 0.

constraint : A Boolean variable indicating whether or not to impose a sum-to-zero constraint $\sum f_j = 0$
Default: 0.

diagonal: Additional constraint to add on the diagonal
Default: 0.

graph: The name of the file where the graph is stored (only if *model=besag*)

n: Length m of vector $\boldsymbol{f}$. Only if *model=rw1,rw2,crw2* and no *locations* is specified.

locations : The name of the file where the value of the covariate are stored, only if *model=rw1,rw2* or *crw2*. If no file is specified the covariate are assumed to take values in $\{0, 1, \ldots, m-1\}$.

cyclic : A Boolean variable specifying whether the model is cyclical, only if *model=rw1,rw2* and no *locations* is specified.

compute: A Boolean variable indicating whether or not the marginals for vector $\boldsymbol{f}$ have to be computed.

Default: 1

summary:A Boolean variable indicating whether or not to output a short summary of the posterior density for $\boldsymbol{f}$.

Default: *compute*

density : A Boolean variable indicating whether or not to output the marginal densities for $\boldsymbol{f}$.

Default: *compute*

quantiles : A list of maximum 10 quantiles, $p(0), p(1), \ldots$, to compute for each node in $\boldsymbol{f}$.

Default: Empty

percentiles :A list of maximum 10 percentiles, $x(0), x(1), \ldots$, to compute for each node in $\boldsymbol{f}$.

Default: Empty

A.1.5 The *type=linear* section

A section of *type=linear* specifies the model for one of the element β_k of vector $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_{n_\beta - 1})$ in equation (2). Hence a `ini` file will contain n_β sections of *type=linear*. Each section consists of the following fields:

covariates : Name of the file where covariate are stored. If empty, then all covariates are assumed to be 1.

precision : Fixed precision for the Gaussian prior distribution of β.

Default: 0.001

compute: A Boolean variable indicating whether or not the marginal for β_k has to be computed.

Default: 1

summary:A Boolean variable indicating whether or not to output a short summary of the posterior density for β_k.

Default: *compute*

density : A Boolean variable indicating whether or not to output the marginal densities for β_k.

Default: *compute*

quantiles : A list of maximum 10 quantiles, $p(0), p(1), \ldots$, to compute for each node in β_k.

Default: Empty

percentiles :A list of maximum 10 percentiles, $x(0), x(1), \ldots$, to compute for each node in β_k.

Default: Empty

A.1.6 The *type=INLA* section

This section is optional, it specifies parameters to be passed to the GMRFLib library. It is possible to specify here all parameters in the *GMRFLib_ai_param_tp* structure. We describe here the most used and useful ones, for more details see the on-line documentation for the GMRFLib library: `http://www.math.ntnu.no/~hrue/GMRFLib/doc/html/`

strategy : The strategy used to compute approximations to the posterior marginals $\pi(x_i|\boldsymbol{y}, \boldsymbol{\theta})$. The three main choice are:

- *GMRFLib_AI_STRATEGY_GAUSSIAN*: computes the Gaussian approximation
- *GMRFLib_AI_STRATEGY_MEANSKEWCORRECTED_GAUSSIAN*: computes the simplified Laplace approximation.
- *GMRFLib_AI_STRATEGY_ADAPTIVE*: Computes the full Laplace approximation.

The three approximation types are described in Rue et al. (2007).

Default: *GMRFLib_AI_STRATEGY_MEANCORRECTED_GAUSSIAN*

int_strategy : The strategy used to integrate out the hyperparameters $\boldsymbol{\theta}$ when computing $\widetilde{\pi}(x_i|\boldsymbol{y})$. There are two possible choices:

- *GMRFLib_AI_INT_STRATEGY_GRID* (or *grid*) : Use a grid strategy, slower and somehow more accurate.
- *GMRFLib_AI_INT_STRATEGY_CCD* (or *ccd*) : Use a central composite design strategy, faster and especially useful for problems with higher dimension of the hyperparameter vector $\boldsymbol{\theta}$.

Both strategies are described in Rue et al. (2007).

Default: *GMRFLib_AI_INT_STRATEGY_GRID*

dz : Step length for the integration procedure, only if *int_strategy =grid*.

Default: 1

diff_logdens : Only used if *int_strategy = grid*. Threshold for accepting a configuration.

Default: 2.5

skip_configurations : Only used if *int_strategy = grid*. Skip fill-in configuration larger than a non-accepted one.

Default: *GMRFLib_TRUE*

gradient_finite_difference_step_len (or *h*): Step length to compute the gradient of $\widetilde{\pi}(\boldsymbol{\theta})$.

Default: 1.0e-4

hessian_finite_difference_step_len (or *h*): Step length to compute the Hessian of $\widetilde{\pi}(\boldsymbol{\theta}|\boldsymbol{y})$ at the mode.

Default: 1.0e-4

interpolator Type of interpolator used to compute marginals for each hyperparameter $\widetilde{\pi}(\theta_m|\boldsymbol{y})$, the available choices are:

- *GMRFLib_AI_INTERPOLATOR_AUTO*: Chose interpolation type based on the integration strategy. If *int_strategy =grid*, then choose *GMRFLib_AI_INTERPOLATOR_WEIGHTED_DISTANCE*. If *int_strategy =ccd*, then the choice is *GMRFLib_AI_INTERPOLATOR_CCD*
- *GMRFLib_AI_INTERPOLATOR_LINEAR*: Linear interpolation using the $(M+1)$ nearest points, where M is the dimension of the hyperparameters space.
- *GMRFLib_AI_INTERPOLATOR_QUADRATIC*: Quadratic interpolation using the $(M+1)$ nearest points.
- *GMRFLib_AI_INTERPOLATOR_WEIGHTED_DISTANCE*: Linear interpolation using weighted distance.
- *GMRFLib_AI_INTERPOLATOR_CCD*: Special interpolation for the CCD integration scheme.

The interpolations are described in Martino (2007).

A.2 Format of the input files

There are five type of input files which can be read from the `inla` program: the data file, the covariate file, the covariate locations type, the graph file and the $\boldsymbol{Q}$-matrix file, each with its own format required. The formats have been already presented in different examples but are all collected here.

Data file The format of the data file depends on the likelihood model chosen and on whether the data are collected on a grid or not. The format of the data file is displayed in Table 2.

Covariate and location file Each covariate has to be stored in a separate file. The format of the file depends on whether the covariate is assumed to have linear or non-linear effect:

Covariates with linear effect: The value of the covariate is simply stored in a file with n_η columns each row having the format:

$$i \quad z_i$$

where $i = 0, \ldots, n_\eta - 1$ and z_i is the value of the covariate for node i.

Covariates with non-linear effect: Let $c \in \boldsymbol{C}$ and $\boldsymbol{C} = \{c^{(0)} < c^{(1)} < \cdots < c^{(idx)} < \cdots < c^{(m-1)}\}$. That is, covariate c takes one of the m values in the ordered vector $\boldsymbol{C}$. The file storing covariate c has n_η row, each with the following format:

$$i \quad (idx)_i$$

where $i = 0, \ldots, n_\eta - 1$ and $(idx)_i$ is the position of the observed value c_i in the vector $\boldsymbol{C}$. If the values in $\boldsymbol{C}$ are different from $0, 1, \ldots$ then another file (the locations file) of m rows, is necessary to store the values of $\boldsymbol{C}$. A short example will be useful:

Example: *Let $n_\eta = 5$ and $\boldsymbol{C} = \{9, 10, 11\}$. Moreover assume that the observed covariate values are $c_0 = 10$, $c_1 = 9$, $c_2 = 11$, $c_3 = 9$ and $c_4 = 10$. Then the covariate file will be as following*

$$\begin{array}{cc} 0 & 1 \\ 1 & 0 \\ 2 & 2 \\ 3 & 0 \\ 4 & 1 \end{array}$$

We would need also a file storing the values in $\boldsymbol{C}$:

$$\begin{array}{c} 9 \\ 10 \\ 11 \end{array}$$

Graph file The graph file contains information on the neighbourhood structure of the spatial effect We describe the required format for such a file using a small example. Let the file *graph.dat*, relative to a small graph, be

```
1  5
2  0 1 1
3  1 2 0 2
4  2 3 1 3 4
5  3 1 2
6  4 1 2
```

Line 1 declares the total number of nodes in the graph, then, in lines 2-6 each node is described. For example, line 4 states that node 2 has 3 neighbours and these are nodes 1, 3 and 4. This is the same format used in the GMRFLib library.

$\boldsymbol{Q}$-matrix file This file is only needed if the field *model* in a *ffield* -type section is defined as *generic*. The file should contain all non-zero entries of the user specified precision matrix $\boldsymbol{Q}$ in the following format

$$i \quad j \quad \boldsymbol{Q}_{ij}$$

where i and j are the row and column index and $\boldsymbol{Q}_{ij}$ is the corresponding entry in the precision matrix.

A.3 Some possible problems and solutions

1. The `inla` function checks that all entries in the `ini` file are used while building the models, so an error message like

   ```
   inla_build: [ZAMBIA.ini] contain[1] unused entries. PLEASE CHECK
   ```

   probably means that some of the fields in the `ini` file have been misspelled.

2. In our experience the most common problems with the `inla` function comes from the optimisation procedure and the numerical computation of the Hessian of $\log \widetilde{\pi}(\boldsymbol{\theta}|\boldsymbol{y})$ at the modal configuration.

   The optimiser might not converge, thus producing an error message like:

   ```
   GMRFLib version 3.0-0-snapshot, has received error no [12]
         Reason    : The Newton-Reason optimiser did not converge
         Function  : GMRFLib_optimize_store
         File      : optimize.c
         Line      : 460
         RCSId     : $Id: tutorial_inla.tex,v 1.8 2008/02/11 08:46:05 hrue Exp $
   ```

Usually restarting the `inla` function assigning different starting values for the hyperparameters vector $\boldsymbol{\theta}$ (field *initial*), will solve the problem.

3. Another error which might happen is that the computed numerical Hessian for $\log \widetilde{\pi}(\boldsymbol{\theta}|\boldsymbol{y})$ in not positive definite. This produces the following error message:

```
GMRFLib version 3.0-0-snapshot, has received error no [2]
        Reason     : Matrix is not positive definite
        Message    : Condition 'gsl_vector_get(eigen_values, (unsigned int) i) >
            0.0' is not TRUE
        Function   : GMRFLib_ai_INLA
        File       : approx-inference.c
        Line       : 2689
        RCSId      : $Id: tutorial_inla.tex,v 1.8 2008/02/11 08:46:05 hrue Exp $
```

To solve this problem it is usually enough to increase the step length used to numerically compute the Hessian and the gradient. These quantities can be re-defined in the *type=INLA* section by using the parameter *h* (or equivalently the two parameters *gradient_finite_difference_step_len* and *hessian_finite_difference_step_len*).

B Some Distribution Functions

B.1 Log Gamma distribution

A random variable X has a LogGamma distribution with parameters a and b (LogGamma(a,b))) if $Y = \exp(X)$ has a Gamma distribution with mean a/b and variance a/b^2.

B.2 Scaled Student-t distribution

A scaled Student-t distribution is a Student-t distribution with ν degrees of freedom scaled so that its mean is 0 and its variance is 1 for any value of the parameter ν.

B.3 NIG distribution

A random variable X is saied to have a standarised normal inverse Gaussian distribution $NIG(\beta, \psi)$ with hyperparameters $\boldsymbol{\theta}_1 = (\beta, psi)$ if its density is given by

$$f(y; \boldsymbol{\theta}_1) = \frac{\gamma\psi}{\pi} \sqrt{\frac{\beta^2 + \psi^2}{(\gamma x + \beta)^2 + \psi^2}} \; \exp\left(\psi^2 + \beta(\gamma x + \beta)\right) \; K_1\left(\sqrt{(\beta^2 + \psi^2)\left((\gamma x + \beta)^2 + \psi^2\right)}\right)$$

where $\gamma^2 = 1 + \beta^2/\psi^2$. The above density has zero mean and unit variance. The parameter β controls (essentially) the skewness of the density, while the parameter ψ is (essentially) a shape parameter. This density is used in financial applications.