Latent Gaussian modeling and INLA:
A review with focus on space-time applications

Titre: Modèles à processus gaussiens latents et inférence INLA :
un survol orienté vers les applications spatio-temporelles

Thomas Opitz[1]

Abstract: Bayesian hierarchical models with latent Gaussian layers have proven very flexible in capturing complex stochastic behavior and hierarchical structures in high-dimensional spatial and spatio-temporal data. Whereas simulation-based Bayesian inference through Markov Chain Monte Carlo may be hampered by slow convergence and numerical instabilities, the inferential framework of Integrated Nested Laplace Approximation (INLA) is capable to provide accurate and relatively fast analytical approximations to posterior quantities of interest. It heavily relies on the use of Gauss–Markov dependence structures to avoid the numerical bottleneck of high-dimensional nonsparse matrix computations. With a view towards space-time applications, we here review the principal theoretical concepts, model classes and inference tools within the INLA framework. Important building blocks for space-time models are certain spatial Matérn-like Gauss–Markov random fields, obtained as approximate solutions to a stochastic partial differential equation. Efficient implementation of statistical inference tools for a large variety of models is available through the `INLA` package of the R software. To showcase the practical use of `R-INLA` and to illustrate its principal commands and syntax, a comprehensive simulation experiment is presented using simulated non Gaussian space-time count data with a first-order autoregressive dependence structure in time.

Résumé : Les modèles bayésiens hiérarchiques structurés par un processus gaussien latent sont largement utilisés dans la pratique statistique pour caractériser des comportements stochastiques complexes et des structures hiérarchiques dans les données en grande dimension, souvent spatiales ou spatio-temporelles. Si des méthodes d'inférence bayésienne de type MCMC, basées sur la simulation de la loi a posteriori, sont souvent entravées par une covergeance lente et des instabilités numériques, l'approche inférentielle par INLA ("Integrated Nested Laplace Approximation") utilise des approximations analytiques, souvent très précises et relativement rapides, afin de calculer des quantités liées aux lois a posteriori d'intérêt. Cette technique s'appuie fortement sur des structures de dépendance de type Gauss–Markov afin d'éviter des difficultés numériques dans les computations matricielles en grande dimension. En mettant l'accent sur les applications spatio-temporelles, nous discutons ici les principales notions théoriques, les classes de modèles accessibles et les outils d'inférence dans le contexte d'INLA. Certains champs Gauss–Markov spatiaux, obtenues comme solution approximative d'une équation différentielle partielle stochastique, sont des briques de construction importantes dans la modélisation spatio-temporelle. Pour illustrer l'utilisation pratique du logiciel `R-INLA` et la syntaxe de ses commandes principales, un scénario de simulation-réestimation est présenté en détail, basé sur des données simulées, spatio-temporelles et non gaussiennes, avec une structure de dépendance autorégressive dans le temps.

Keywords: Integrated Nested Laplace Approximation, R-INLA, spatio-temporal statistics

Mots-clés : Integrated Nested Laplace Approximation, R-INLA, statistique spatio-temporelle

---

[1] BioSP, INRA, 84914, Avignon, France E-mail: Thomas.Opitz@paca.inra.fr

1. Introduction

The rapidly increasing availability of massive sets of georeferenced data has spawned a strong demand for suitable statistical modeling approaches to handle large and complex data. Bayesian hierarchical models have become a key tool for capturing and explaining complex stochastic structures in spatial or spatio-temporal processes. Many of these models are based on latent Gaussian processes, embedded in a parameter related to the mean of the distribution assumed for the likelihood of the data, and extend the Gaussian random field modeling brought forward by classical geostatistics. Using a conditional independence assumption for the data process with respect to the latent Gaussian layer makes inference tractable in many cases. Typically, closed-form expressions for the likelihood are not available for these complex models, and simulation-based inference through Markov chain Monte Carlo has become a standard approach for many models. An important alternative, superior to MCMC inference under certain aspects, has been developed through the idea of Integrated Nested Laplace Approximation, proposed in the JRSS discussion paper of Rue et al. (2009). Many case studies have been conducted through INLA in the meantime, with space-time applications to global climate data (Lindgren et al., 2011), epidemiology (Bisanzio et al., 2011), disease mapping and spread (Schrödle and Held, 2011; Schrödle et al., 2012), forest fires (Serra et al., 2014), air pollution risk mapping (Cameletti et al., 2013) or fishing practices (Cosandey-Godin et al., 2014). More generally, INLA has been successfully applied to generalized linear mixed models (Fong et al., 2010), log-Gaussian Cox processes (Illian et al., 2012), survival models (Martino et al., 2011) and quantile regression (Yue and Rue, 2011), amongst many other application fields. The recent monograph of Blangiardo and Cameletti (2015) reviews INLA in detail and gives many practical examples. Instead of applying simulation techniques to produce a representative sample of the posterior distribution, INLA uses analytic Laplace approximation and efficient numerical integration schemes to achieve highly accurate analytical approximation of posterior quantities of interest with relatively small computing times. In particular, we get approximations of univariate posterior marginals of model hyperparameters and of the latent Gaussian variables. By making use of latent Gauss–Markov dependence structures, models remain tractable even in scenarios that are very high-dimensional in terms of observed data and latent Gaussian variables.

The INLA-based inference procedures are implemented in the R-package `INLA` (referred to as `R-INLA` in the following) for a large variety of models, defined through basic building blocks of three categories: the (univariate) likelihood specification of data, the latent Gaussian model and prior distributions for hyperparameters. Functionality of `R-INLA` is continuously extended (Martins et al., 2013; Lindgren and Rue, 2015; Rue et al., 2016). This review and the code examples refer to `R-INLA` version 0.0-1463562937. The `R-INLA` software project is hosted on http://www.r-inla.org/, where one can find lots of INLA-related resources, amongst them details on the specification of likelihoods, latent models and priors, a discussion forum with very active participation of the members of the INLA core team, tutorials and codes, an FAQ section, etc.

2. Modeling and estimation with INLA

2.1. Latent Gaussian modeling

The structured latent Gaussian regression models amenable to INLA-based inference can be defined in terms of three layers: hyperparameters, latent Gaussian field, likelihood model. The univariate likelihood captures the marginal distribution of data and is usually chosen as an exponential family (Gaussian, gamma, exponential, Weibull, Cox, binomial, Poisson, negative binomial, ...), where models like the exponential, Weibull or Cox ones are available as survival models allowing for right- and left-side censoring. Following the mechanism of generalized linear models, the mean of the likelihood distribution is determined by the latent Gaussian predictor through a link function such that $\mathbb{E}(Y \mid \eta) = h^{-1}(\eta)$, where Y is the observation, η is a Gaussian predictor and h is an appropriately chosen link function. Hyperparameters can appear in the likelihood as dispersion parameters like the variance of the Gaussian distribution, the overdispersion parameter of the negative binomial one or the shape parameter of the gamma one, or they can characterize the structure of the latent Gaussian model, for instance through variances, spatial correlation parameters or autoregression coefficients. Formally, this hierarchical model can be written as

$$\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta}) \qquad \text{hyperparameters} \qquad (1)$$

$$\boldsymbol{x} \mid \boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}(\boldsymbol{\theta})^{-1}) \qquad \text{latent Gaussian field} \qquad (2)$$

$$\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta} \sim \prod_i \pi(y_i \mid \eta_i(\boldsymbol{x}), \boldsymbol{\theta}) \qquad \text{observations} \qquad (3)$$

where $\boldsymbol{Q}(\boldsymbol{\theta})$ is the precision matrix (*i.e.*, inverse covariance matrix) of the latent Gaussian vector $\boldsymbol{x}$, $\boldsymbol{\eta}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$ with the so-called *observation matrix* $\boldsymbol{A}$ that maps the latent variable vector $\boldsymbol{x}$ to the predictors $\eta_i = \eta_i(\boldsymbol{x})$ associated to observations y_i. If $\boldsymbol{y}$ and $\boldsymbol{x}$ can be high-dimensional when using INLA, an important limitation concerns the hyperparameter vector $\boldsymbol{\theta} \in \Theta$ whose dimension should be moderate, say ≤ 10, which is due to numerical integration that has to be carried out over the hyperparameter space Θ. Notice that the Gaussian likelihood is particular since, conditional to the hyperparameters, the observations are still Gaussian. In practice, the precision hyperparameter of the Gaussian likelihood (*i.e.*, the inverse of its variance) can correspond to a measurement error or a nugget effect, and we can fix a very high value for the precision hyperparameter if we want the model for data $\boldsymbol{y}$ to correspond exactly to the latent Gaussian predictors $\boldsymbol{\eta}$. The dependence structure between observations y_i is captured principally by the precision matrix $\boldsymbol{Q}(\boldsymbol{\theta})$ of the latent field $\boldsymbol{x}$. In practice, it is strongly recommended or even indispensable from the point of view of computation time and memory requirements to choose Gauss–Markov structures with sparse $\boldsymbol{Q}(\boldsymbol{\theta})$ whenever model dimension is high.

The resulting joint posterior density of latent variables $\boldsymbol{x}$ and hyperparameters $\boldsymbol{\theta}$ is

$$\pi(\boldsymbol{x}, \boldsymbol{\theta} \mid \boldsymbol{y}) \propto \exp\left(-0.5\boldsymbol{x}'\boldsymbol{Q}(\boldsymbol{\theta})\boldsymbol{x} + \sum_i \log \pi(y_i \mid \eta_i, \boldsymbol{\theta}) + \log \pi(\boldsymbol{\theta}) \right). \qquad (4)$$

This density over a high-dimensional space does usually not characterize one of the standard multivariate families and is therefore difficult to interpret and to manipulate. In practice, the main

interest lies in the marginal posteriors of hyperparameters θ_j, of latent variables x_i and of the resulting predictors η_i, which can be included into $\boldsymbol{x}$ for notational convenience. Calculation of these univariate posterior densities requires integrating with respect to $\boldsymbol{\theta}$ and $\boldsymbol{x}$:

$$\pi(\theta_j \mid \boldsymbol{y}) = \int \int \pi(\boldsymbol{x}, \boldsymbol{\theta} \mid \boldsymbol{y}) \mathrm{d}\boldsymbol{x} \, \mathrm{d}\boldsymbol{\theta}_{-j} = \int \pi(\boldsymbol{\theta} \mid \boldsymbol{y}) \mathrm{d}\boldsymbol{\theta}_{-j}, \tag{5}$$

$$\pi(x_i \mid \boldsymbol{y}) = \int \int \pi(\boldsymbol{x}, \boldsymbol{\theta} \mid \boldsymbol{y}) \mathrm{d}\boldsymbol{x}_{-i} \mathrm{d}\boldsymbol{\theta} = \int \pi(x_i \mid \boldsymbol{\theta}, \boldsymbol{y}) \pi(\boldsymbol{\theta} \mid \boldsymbol{y}) \mathrm{d}\boldsymbol{\theta}. \tag{6}$$

We notice that the use of astutely designed numerical integration schemes with respect to the moderately dimensioned hyperparameter space Θ can yield satisfactorily accurate approximation of the outer integral. On the other hand, calculating the inner integral with respect to $\boldsymbol{x}$, often of very high dimension ($\approx 10^2$ to 10^8), is intricate.

2.2. Gauss–Markov models

We say that a random vector $\boldsymbol{x} \mid \boldsymbol{\theta} \sim \mathscr{N}(\boldsymbol{0}, \boldsymbol{Q}^{-1})$ is Gauss–Markov if the number of nonnull entries of its $n \times n$ precision matrix $\boldsymbol{Q}$ is $\mathscr{O}(n)$. Such sparse precision matrices allow efficient numerical computation of matrix operations like *LR*-decomposition (with sparse factors *L* and *R*), determinant calculation, matrix-vector products, etc. For instance, complexity of matrix inversion decreases from $\mathscr{O}(n^3)$ for matrices without any structural constraints to around $\mathscr{O}(n^{3/2})$ for sparse matrices. Using Gauss–Markov structures fundamentally shifts the dependence characterization from covariance matrices $\boldsymbol{Q}^{-1}$ to precision matrices $\boldsymbol{Q}$. Notice that the conditional expectation is easily expressed through the regression $\mathbb{E}(x_i \mid \boldsymbol{x}_{-i}) = -\sum_{j \neq i}(q_{ij}/q_{ii})x_j$, where only a small number of the sum terms, also called the neighborhood of x_i, are non-zero owing to the sparse structure of $\boldsymbol{Q}$. The conditional variance is $\mathbb{V}(x_i \mid \boldsymbol{x}_{-i}) = 1/q_{ii}$. R-INLA uses fast and efficient algorithms for sparse matrix calculations (Rue and Held, 2005), already implemented in the GMRFLib library. For efficient calculations, it is important to make the precision matrix $\boldsymbol{Q}$ "as diagonal as possible" by reordering variables to regroup nonzero elements as close as possible to the diagonal. R-INLA has implemented several of those reordering strategies; see Rue and Held (2005) for more details on reordering algorithms. If certain Gauss–Markov models exist for spatially indexed graphs, useful covariance functions defined over $\mathbb{R}^d$ and leading to Gauss–Markov covariance matrices are difficult to establish. An exception is the very flexible approximate Gauss–Markov representation of Matérn-like covariances based on certain stochastic partial differential equations (often referred to as the *SPDE approach* in the literature), which is also implemented in R-INLA; see Section 3.1 for more details.

2.3. INLA

The fundamental idea of INLA consists in applying the device of Laplace approximation to integrate out high-dimensional latent components. This theoretical foundation is combined with efficient algorithms and numerical tricks and approximations to ensure a fast yet accurate approximation of posterior marginal densities of interest like those of the predictors η_i in (6) or of hyperparameters θ_j in (5). Since the details of methods implemented in the INLA approximation are quite technical, we here content ourselves with a presentation of the main ideas and the related options available in R-INLA.

2.3.1. The principle of Laplace approximation

We first recall the principle of the Laplace approximation and its calculation in practice. Typically, one seeks to evaluate an integral $\int f(\boldsymbol{x}) \, d\boldsymbol{x}$, where the positive integrand function f, here written as $f(\boldsymbol{x}) = \exp(g(\boldsymbol{x}))$, is defined over a high-dimensional space and is "well-behaved" in the sense that it is unimodal and that its shape is not too far from gaussianity. Since the integral value is mainly determined by the behavior around the mode of g, a second-order Taylor approximation of g can be substituted for g to calculate an approximate value of the integral. Assuming that $\boldsymbol{x}^{\star}$ is the unique global maximum of g, we get $g(\boldsymbol{x}) \approx g(\boldsymbol{x}^{\star}) + 0.5(\boldsymbol{x} - \boldsymbol{x}^{\star})' \boldsymbol{H}(g)(\boldsymbol{x}^{\star})(\boldsymbol{x} - \boldsymbol{x}^{\star})$ for values $\boldsymbol{x}$ close to $\boldsymbol{x}^{\star}$ with the Hessian matrix $\boldsymbol{H}(g)(\boldsymbol{x}^{\star})$. Notice that $-\boldsymbol{H}(g)(\boldsymbol{x}^{\star})$ is positive definite. More generally, we now write $f(\boldsymbol{x}) = \exp(kg(\boldsymbol{x}))$ with a scale variable $k > 0$. An approximate value of the integral can be calculated using the fact that a multivariate Gaussian density integrates to 1. The resulting following integral approximation in dimension d is expected to become more and more accurate for higher values of $k \gg 1$:

$$\int_{-\infty}^{\infty} f(\boldsymbol{x}) \, d\boldsymbol{x} = \int_{-\infty}^{\infty} \exp(kg(\boldsymbol{x})) \, d\boldsymbol{x} \tag{7}$$

$$\stackrel{k \gg 0}{\approx} \int_{-\infty}^{\infty} \exp(kg(\boldsymbol{x}^{\star}) + 0.5k(\boldsymbol{x} - \boldsymbol{x}^{\star})' \boldsymbol{H}(g)(\boldsymbol{x}^{\star})(\boldsymbol{x} - \boldsymbol{x}^{\star})) \, d\boldsymbol{x}$$

$$= \left(\frac{2\pi}{k}\right)^{d/2} |\boldsymbol{H}(g)(\boldsymbol{x}^{\star})|^{-1/2} \exp(kg(\boldsymbol{x}^{\star})). \tag{8}$$

In statistical practice, k may represent the number of i.i.d. replications, each of which has density $f(\boldsymbol{x}) = \exp(g(\boldsymbol{x}))$. Higher values of k usually lead to better approximation, and formal results on the quality of approximation have been derived (Tierney and Kadane, 1986; Rue et al., 2009). Many of the models commonly estimated with INLA have no structure of i.i.d. replication, but the Laplace approximation remains sufficiently accurate in most cases since there usually still is a structure of internal replication, for instance when there are blocks of data that are close to independence.

In the context of INLA, the following observation will be interesting and useful. Fix $k = 1$ in (7) and suppose that $f(\boldsymbol{x}) = \exp(g(\boldsymbol{x})) = \pi(\boldsymbol{x}, \boldsymbol{\theta})$, where $\pi(\boldsymbol{x}, \boldsymbol{\theta})$ is the joint probability density of a random vector $(\boldsymbol{x}, \boldsymbol{\theta})$. Then, in (8), the term $\exp(g(\boldsymbol{x}^{\star}))$ is the value of π at its mode $\boldsymbol{x}^{\star}$ for fixed $\boldsymbol{\theta}$, whereas $(2\pi)^{d/2} |\boldsymbol{H}(g)(\boldsymbol{x}^{\star})|^{-1/2}$ is $1/\pi_G(\boldsymbol{x}^{\star} \mid \boldsymbol{\theta})$ with π_G a Gaussian approximation with mean vector $\boldsymbol{x}^{\star}$ to the conditional density of $\boldsymbol{x} \mid \boldsymbol{\theta}$. In practice, we can determine the mean $\boldsymbol{\mu}^{\star} = \boldsymbol{x}^{\star}$ and the precision matrix $\boldsymbol{Q}^{\star} = -\boldsymbol{H}(g)(\boldsymbol{x}^{\star})$ of π_G through an iterative Newton–Raphson optimisation. Starting from the joint posterior (4) of our latent Gaussian model, we set $g(\boldsymbol{x}) = -0.5\boldsymbol{x}'\boldsymbol{Q}(\boldsymbol{\theta})\boldsymbol{x} + \sum_i \log \pi(y_i \mid \eta_i, \boldsymbol{\theta})$. We further write $g_i(x_i) = \log \pi(y_i \mid x_i, \boldsymbol{\theta})$ and calculate its second-order Taylor expansion $g_i(x_i) \approx g_i(\mu_i^{(0)}) + b_i x_i - 0.5 c_i x_i^2$. Without loss of generality, we here assume that the linear predictor $\boldsymbol{\eta}$ corresponds to the latent Gaussian vector $\boldsymbol{x}$. We start the iterative optimisation with initial values $\boldsymbol{Q}^{(1)} = \boldsymbol{Q} + \text{diag}(\boldsymbol{c})$ and $\boldsymbol{\mu}^{(1)}$, where $\boldsymbol{Q}^{(1)}\boldsymbol{\mu}^{(1)} = \boldsymbol{b}$. We then iterate this procedure until convergence such that $\boldsymbol{\mu}^{(i)} \to \boldsymbol{\mu}^{\star} = \boldsymbol{x}^{\star}$ and $\boldsymbol{Q}^{(i)} \to \boldsymbol{Q}^{\star} = \boldsymbol{Q} + \text{diag}(\boldsymbol{c}^{\star})$, where an appropriate convergence criterion must be used. Notice that the conditional independence assumption of observations y_i with respect to $(\eta_i, \boldsymbol{\theta})$ allows preserving the sparse structure in $\boldsymbol{Q}^{\star}$. It is further possible to impose linear constraints $\boldsymbol{M}\boldsymbol{x} = \boldsymbol{e}$

Soumis au Journal de la Société Française de Statistique
File: INLASFDS.tex, compiled with jsfds, version : 2009/12/09
date: July 25, 2016

onto $\boldsymbol{x}$ and $\boldsymbol{x}^{\star}$ with given matrix $\boldsymbol{M}$ and vector $\boldsymbol{e}$ by using the approach of conditioning through kriging (Rue et al., 2009).

2.3.2. Posterior marginal densities of hyperparameters

To calculate

$$\pi(\theta_j \mid \boldsymbol{y}) = \int \int \pi(\boldsymbol{x}, \boldsymbol{\theta} \mid \boldsymbol{y}) \mathrm{d}\boldsymbol{x} \, \mathrm{d}\boldsymbol{\theta}_{-j} = \int \pi(\boldsymbol{\theta} \mid \boldsymbol{y}) \mathrm{d}\boldsymbol{\theta}_{-j}, \tag{9}$$

we use the Laplace approximation of the inner integral $\int \pi(\boldsymbol{x}, \boldsymbol{\theta} \mid \boldsymbol{y}) \mathrm{d}\boldsymbol{x} = \pi(\boldsymbol{\theta} \mid \boldsymbol{y})$ as described in Section 2.3.1 such that the approximated density $\tilde{\pi}$ satisfies

$$\tilde{\pi}(\boldsymbol{\theta} \mid \boldsymbol{y}) \propto \frac{\pi(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{y})}{\pi_G(\boldsymbol{x} \mid \boldsymbol{\theta}, \boldsymbol{y})} \mid_{\boldsymbol{x} = \boldsymbol{x}^{\star}(\boldsymbol{\theta})} \tag{10}$$

with $\boldsymbol{x}^{\star}(\boldsymbol{\theta})$ the mode of the joint density $\pi(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{y})$ for fixed $(\boldsymbol{\theta}, \boldsymbol{y})$ and a Gaussian density π_G that approximates $\pi(\boldsymbol{x} \mid \boldsymbol{\theta}, \boldsymbol{y})$:

$$\pi_G(\boldsymbol{x} \mid \boldsymbol{\theta}, \boldsymbol{y}) = (2\pi)^{n/2} |\boldsymbol{Q}^{\star}(\boldsymbol{\theta})|^{1/2} \exp\left(-0.5(\boldsymbol{x} - \boldsymbol{x}^{\star}(\boldsymbol{\theta}))'\boldsymbol{Q}^{\star}(\boldsymbol{\theta})(\boldsymbol{x} - \boldsymbol{x}^{\star}(\boldsymbol{\theta}))\right). \tag{11}$$

Notice that the Gaussian approximation π_G is exact if the data likelihood $\pi(y_i \mid \eta_i, \boldsymbol{\theta})$ itself is Gaussian. An approximation of the posterior marginal of θ_j in (9) is now obtained through a numerical integration with a set of integration nodes $\boldsymbol{\theta}_{\ell}$ chosen from a numerical exploration of the surface of the density $\tilde{\pi}(\boldsymbol{\theta}_{-j}, \theta_j \mid \boldsymbol{y})$ (with θ_j held fixed). This yields

$$\tilde{\pi}(\theta_j \mid \boldsymbol{y}) = \sum_{\ell=1}^{L} \omega_{\ell} \tilde{\pi}(\boldsymbol{\theta}_{\ell} \mid \boldsymbol{y}) \tag{12}$$

with weights ω_{ℓ} (which are chosen to be 1 in the approaches implemented in R-INLA). In R-INLA, $\boldsymbol{\theta}_{\ell}$ can either be chosen as a grid around the mode of $\tilde{\pi}(\boldsymbol{\theta} \mid \boldsymbol{y})$ (int.strategy="grid", the most costly variant), or through a simpler so-called *complete composite design* which is less costly when the dimension of $\boldsymbol{\theta}$ is relatively large (int.strategy="ccd", the default approach), or we may use only one integration node given as the mode value (int.strategy="eb", corresponding to the idea of an *empirical Bayes* approach).

2.3.3. Posterior marginal densities of the latent Gaussian field

For calculating the marginal density $\pi(x_i \mid \boldsymbol{y})$ of a latent variable x_i, we lean on representation (6). Numerical integration with respect to $\boldsymbol{\theta}$ can be done in analogy to the procedure described in Section 2.3.2, and the Laplace approximation (10) allows approximating $\pi(\boldsymbol{\theta} \mid \boldsymbol{y})$. It thus remains to (approximately) evaluate $\pi(x_i \mid \boldsymbol{\theta}, \boldsymbol{y})$. A simple and fast solution would be to use the univariate Gaussian approximation resulting from the multivariate Gaussian approximation (11) whose mean value is $x_i^{\star}(\boldsymbol{\theta})$ and whose variance can easily and quickly be calculated from a partial inversion of the precision $\boldsymbol{Q}^{\star}(\boldsymbol{\theta})$ (Rue, 2005) (strategy="gaussian" in R-INLA). However, this Gaussian approximation often fails to capture skewness behavior and can generate

Soumis au Journal de la Société Française de Statistique
File: INLASFDS.tex, compiled with jsfds, version : 2009/12/09
date: July 25, 2016

nonnegligible bias in certain cases – an important exception to this issue being the case where the data likelihood is Gaussian. In the general case, using again a Laplace-like approximation

$$\frac{\pi(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{y})}{\pi_G(\boldsymbol{x}_{-i} \mid x_i, \boldsymbol{\theta}, \boldsymbol{y})} \mid_{\boldsymbol{x}_{-i} = \boldsymbol{x}^{\star}_{-i}(x_i, \boldsymbol{\theta})} \tag{13}$$

with mode $\boldsymbol{x}^{\star}_{-i}(x_i, \boldsymbol{\theta})$ of $\pi(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{y})$ for fixed $(x_i, \boldsymbol{\theta}, \boldsymbol{y})$ would be preferable, but is relatively costly (`strategy="laplace"` in R-INLA). Instead, Rue et al. (2009) propose a so-called *simplified Laplace approximation* based on third-order Taylor developments of numerator and denominator in (13) that satisfactorily remedies location and skewness inaccuracies of the Gaussian approximation (`strategy="simplified.laplace"` in R-INLA, the default). Notice that the "Nested" in INLA refers to this second Laplace-like approximation.

3. Space-time modeling approaches

Modeling nonstationarities over time and spatio-temporal dependence in repeated spatial observations is paramount to understanding the dynamics of processes observed over space and time. We here review approaches to integrating the time component into the latent Gaussian predictor η_{st} that are suitable for high-dimensional space-time inference with INLA. In principle, any space-time Gaussian process η_{st} could be used, but the requirement of a Gauss–Markov structure for fast matrix calculations and the current scope of models implemented in R-INLA impose some constraints. Flexible, Matérn-like spatial Gauss–Markov models with continuous sample paths are available in R-INLA and will be discussed in Section 3.1. A generalization of such purely spatial models to flexible nonseparable space-time dependence structures is still pending, but R-INLA allows extending such spatial models to capture temporal dependence through autoregressive structures, which will be discussed in the following and explored in the examples of Section 4. New approaches to generic nonseparable space-time Gauss–Markov model classes and their implementation in R-INLA are projected by members from the scientific community working on the SPDE approach. Any other nonseparable covariance model would in principle be amenable to INLA-based inference without difficulty as long as it keeps a Markovian structure when using a high-dimensional latent model.

We start by formulating a generic latent Gaussian space-time model for the predictor η_{st} that covers many of the models that can be fitted with R-INLA. We denote by ζ_{stk}, $k = 1, \dots, K$ given covariate data, which may depend on space or time only. Notice that in cases where covariates are not available except at the observation points (s_i, t_i), $i = 1, \dots, n$ of y_i, one may have to include them into into the observation vector $\boldsymbol{y}$ and formulate a latent Gaussian model for their interpolation over space and time. This would give a model that achieves both interpolation of covariate values and prediction of $\boldsymbol{y}$ at the same time. Here, we consider

$$\eta_{st} = x_0 + \sum_{k=1}^{K} x_k \zeta_{stk} + \sum_{k=1}^{L} f_k(\zeta_{stk}) + x_t + x_s + x_{st} \tag{14}$$

The linear coefficients $x_0, \dots, x_K$ are known as *fixed effects*, whereas functions $f_k(\cdot)$ and the processes x_t, x_s and x_{st} are referred to as *random effects*. Notice that x_t and x_s are to be understood

Soumis au Journal de la Société Française de Statistique
File: INLASFDS.tex, compiled with jsfds, version : 2009/12/09
date: July 25, 2016

as marginal models that capture purely spatial or temporal marginal effects. We now shortly present some typical examples of Gaussian prior models that are commonly used and available in R-INLA for the intercept x_0, the linear covariate effects x_k, the nonlinear covariate effects $f_k(\cdot)$, the marginal temporal effect x_t, the marginal spatial effect x_s and the space-time effect x_{st}.

A temporal effect x_t could be modeled as an autoregressive process or as a random walk, where autoregression coefficients and the step variance of the random walk represent hyperparameters that could be estimated. A nonparametric spatial effect x_s could be modeled with a Gauss–Markov Matérn-like prior random field, the details of whose construction are presented in the following section 3.1. A simple extension to a space-time effect x_{st} is obtained from considering independent replicates of the spatial field for each time point. An important class of space-time models x_{st} that allows for temporal dependence and preserves the Gauss–Markov structure are the stationary first-order autoregressive models

$$x_{st} = ax_{s,t-1} + \sqrt{1-a^2}\varepsilon_{st}, \tag{15}$$

where $a \in (-1,1]$ and ε_{st} is a stationary spatial innovation process, i.i.d. in time, typically chosen as the Gauss–Markov Matérn-like field. We here allow for $a = 1$ to include the case of temporal independence as mentioned before. This AR model is a *group model* where spatial groups ε_{st}, $t = 1, 2, \ldots$, are joined through an AR group model. In R-INLA, it is possible to work with more general group models that define a type of dependence like "autoregressive", "exchangeable", "random walk" or "besag" between certain groups of latent variables. It is further possible to specify certain graph structures among latent variables; we here refer to www.r-inla.org for full details about the specification of a large variety of available latent models. Owing to issues of identifiability and model complexity, usual only a subset of the terms in (14) is used to construct the latent field in practical applications.

3.1. *Spatial Gauss–Markov random fields based on the SPDE approach*

The spatial SPDE model of Lindgren et al. (2011) defines a Gauss–Markov random field as the approximate solution to a certain stochastic partial differential equation. It is an important building block for latent Gaussian models with spatial and spatio-temporal effects. Contrary to classical covariance function models, this approach provides sparse precision matrices that make numerical procedures efficient even for very high-dimensional problems. Formally, a Gaussian process $x(s)$ on $\mathbb{R}^D$ is defined through

$$\left(\kappa^2 - \Delta\right)^{\alpha/2} x(s) = W(s), \quad \alpha = \nu + D/2, \quad s \in \Omega \tag{16}$$

with the Laplace operator $\Delta y = \sum_{j=1}^{D} \partial^2 y / \partial^2 x_j$, a standard Gaussian white noise process $W(s)$ and a nonempty spatial domain $\Omega \subset \mathbb{R}^D$ with regular boundary. Depending on the value of ν and D, the Laplace operator $\left(\kappa^2 - \Delta\right)^{\alpha/2}$ is fractionary with noninteger exponent $\alpha/2$, and it must be defined in an appropriate way (Lindgren et al., 2011). The only stationary solution to (16) for $\Omega = \mathbb{R}^D$ is a Gaussian random field with the Matérn covariance function whose shape parameter is ν (with $\nu = 0.5$ yielding the exponential covariance model). The marginal variance is $\Gamma(\nu)/[\Gamma(\nu + D/2)(4\pi)^{D/2}\kappa^{2\nu}]$, and the "empirical range" where a correlation of approximately

Soumis au Journal de la Société Française de Statistique
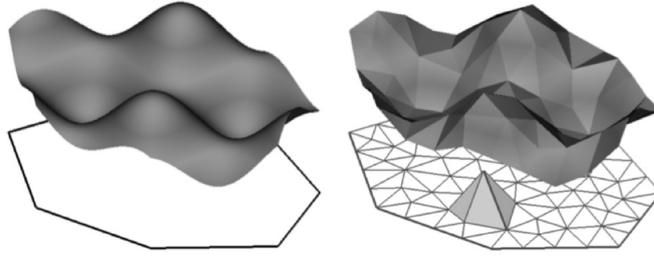File: INLASFDS.tex, compiled with jsfds, version : 2009/12/09
date: July 25, 2016

FIGURE 1. *Illustration taken from Blangiardo et al. (2013). Example of a Gaussian surface (left) and its finite element approximation (right). On the right display, we can further see the triangulation and one of the finite element pyramid-shaped basis functions.*

0.1 is attained between two points is around $\sqrt{8\nu}/\kappa^2$. The Matérn model is known to be very flexible through its scale and shape parametrization, with regularity properties of sample paths governed by the shape parameter ν.

In practice, when working on a finite domain $\Omega \subset \mathbb{R}^2$, boundary effects come into play. One can assume a polygon-shaped boundary $\partial\Omega$, as it is implemented in R-INLA. An interesting choice of boundary condition is the Neumann condition with zero normal derivatives at the boundary such that the Gaussian field is "reflected" at the boundary. Lindgren et al. (2011) show that Neumann conditions principally lead to an increase in variance close to the boundary, the factor being approximately 2 when there is one close linear boundary segment, and 4 when we are close to the 90-degree angle of a rectangle where two linear segments meet. Whereas such boundary conditions may be interesting for some applications, we often prefer to extend the domain Ω beyond the study region towards a larger domain, such that boundary effects become negligible within the study region. This requires that the extended domain's boundary is separated by a distance superior to the empirical range from the study region.

Approximate solutions to (16) are obtained based on the finite element approach commonly used in the numerical solution of partial differential equations. Using a triangulation of the spatial domain leads to a high-dimensional multivariate representation with Gaussian variables located in the triangulation nodes s_i. Spatial interpolation between nodes is achieved by considering these Gaussian variables as weights for piecewise linear basis functions $\psi_i(s)$, one for each node. Basis functions are of compact support giving by the triangles touching the node s_i ("pyramid functions"); see Figure 1 for an example of a basis function and of the approximation of a spatial surface through a linear combination of such basis functions. By "projecting" the SPDE (16) on the space spanned by the Gaussian variables, one can calculate the precision matrix $\boldsymbol{Q}$ governing the dependence structure between these variables. There are certain rules of thumb to be respected for a construction of the triangulation that does not strongly distort the actual dependence structure of the exact solution to (16) and that remains numerically stable with respect to certain matrix computations, mainly concerning maximum sizes of triangles and minimum sizes of interior angles. For numerical efficiency, overly fine triangulations can be avoided by requiring minimum edge lengths, for instance; see Lindgren et al. (2011) for further details on the finite element

Soumis au Journal de la Société Française de Statistique
File: INLASFDS.tex, compiled with jsfds, version : 2009/12/09
date: July 25, 2016

construction. The approximate Gauss–Markov solution has mean zero and its precision matrix $\boldsymbol{Q}$ has entries that are determined by the SPDE. We refer to the Appendix section of Lindgren et al. (2011) for the calculation of $\boldsymbol{Q}$, which is explicit. Based on the approximate solutions for $\alpha = 0$ and $\alpha = 1$, an approximate solution of the SPDE for $\alpha + 2$ can be obtained by injecting the solution for α at the place of the white noise $W(s)$ in (16). For non-integer values of $\alpha > 0$, additional approximations are necessary to obtain a solution, see the authors' response in the discussion of the Lindgren et al. (2011) paper. We have here considered κ and τ to be constant over space. It is possible to allow spatial variation of these parameters for nonstationary models with $\kappa = \kappa(s)$ as in (16) and a precision-related parameter $\tau = \tau(s)$ that varies over space. To wit, Ingebrigtsen et al. (2014) apply such second-order nonstationary modeling to precipitation in Norway, where altitude is used as a covariate that acts on the local covariance range in the dependence structure. The SPDE (16) is also well-defined over manifolds Ω, for instance the sphere $\mathbb{S}_2$ embedded in $\mathbb{R}^3$. In general, Lindgren et al. (2011) show that the approximate solution converges to the true solution of the SPDE for an adequately chosen norm when the triangulation is refined in a way such that the maximum diameter of a circle inscribed into one of the triangles tends to 0.

R-INLA currently implements calculation, estimation and simulation of the Gauss–Markov SPDE solution for $\alpha \leq 2$, for Ω a subset of $\mathbb{R}^1$, $\mathbb{R}^2$ or a two-dimensional manifold embedded into $\mathbb{R}^3$, and for spatially constant or varying values $\kappa(s)$ and $\tau(s)$. A large number of tools is available to construct numerically efficient and stable triangulations.

4. Using **R-INLA** on a complex simulated space-time data set

To execute the following code, the INLA package of R must be installed; see www.r-inla.org for information on the R command recommended for installing this package, which is not hosted on the site of the Comprehensive R Archive Network (CRAN) due to its use of external libraries. We here illustrate the powerful estimation and inference tools of R-INLA in a controlled simulation experiment with data simulated from a latent Gaussian space-time model with the Poisson likelihood. The full code for the simulation study below can be requested from the author.

4.1. Simulating the data

We simulate a space-time count model based on a latent first-order autoregressive Gaussian process defined on $[0,1]^2$ for $t = 1, \ldots, 60$. We use two covariates, given as $z_1(t) = t/60$ and $z_2(t)$ simulated according to an autoregressive time series model. The simulated model is

$$
\begin{aligned}
Y(s,t) \mid \eta(s,t) &\sim \text{Pois}(\exp(\eta(s,t))) \quad \text{i.i.d.} \\
\eta(s,t) &= -1 + z_1(t) + 0.5 z_2(t) + W(s,t) \\
W(s,1) &= \varepsilon(s,1) \\
W(s,t) &= 0.5 W(s,t-1) + (1-0.5)^2 \varepsilon(s,t), \quad t = 2, \ldots, 60.
\end{aligned}
$$

The covariance function of the standard Gaussian field $\varepsilon(s,t)$ is chosen of Matérn type with shape parameter $\nu = 1$ and effective range 0.25. We now fix $50 \times 60 = 3000$ observation points (s_i, t_i)

Soumis au Journal de la Société Française de Statistique
File: INLASFDS.tex, compiled with jsfds, version : 2009/12/09
date: July 25, 2016

of $Y(s,t)$, determined as the Cartesian product of 50 sites uniformly scattered in $[0,1]$ and $t_i = i$, $i = 1, \ldots, 60$.

To illustrate the simulation capacities of `R-INLA`, we here use the SPDE approach to achieve simulation based on the Gauss–Markov approximation of the Matérn correlation structure. After loading the `INLA`-package and fixing a random seed for better reproducibility of results, we start by defining the κ, τ and α parameters of the SPDE. To avoid boundary effects in the SPDE simulation, we wil use the square $[-0.5, 1.5]^2$ as spatial support.

```
library(INLA)
seed=2;set.seed(seed)
n.repl=60;n.sites=50
nu=1;alpha=nu+1;range0=0.25;sigma0=1;a=.5
kappa=sqrt(8*nu)/range0
tau=1/(2*sqrt(pi)*kappa*sigma0)
```

Next, we create a fine 2D triangulation mesh for relatively accurate simulation, with maximum edge length 0.04 within $[0,1]^2$ and 0.2 in $[-0.5, 1.5]^2 \setminus [0,1]^2$. The minimum angle between two edges is set to 21, a value recommended to avoid ill-conditioned triangulations containing very elongated triangles. Polygon nodes should be given in counterclockwise order, whereas as polygon-shaped holes in the support would be specified by clockwise order of nodes; see the left display of Figure 2 for the resulting triangulation.

```
nodes.bnd=matrix(c(0,0,1,0,1,1,0,1),ncol=2,byrow=T)
segm.bnd=inla.mesh.segment(nodes.bnd)
nodes.bnd.ext=matrix(c(-.5,-.5,1.5,-.5,1.5,1.5,-.5,1.5),ncol=2,byrow=T)
segm.bnd.ext=inla.mesh.segment(nodes.bnd.ext)
mesh.sim=inla.mesh.2d(boundary=list(segm.bnd,segm.bnd.ext),
   max.edge=c(.04,.2),min.angle=21)
plot(mesh.sim)
```

The list slot `mesh.sim$n` informs us that there are 2401 triangulation nodes. The mesh object `mesh.sim` has a slot `mesh$loc` which contains a three column matrix. The first two columns indicate the 2D coordinates of mesh nodes. In our case, the third coordinate, useful for specifying 2D manifolds in 3D space, is constant 0. We continue by creating the SPDE model through an `R-INLA` function `inla.spde2.matern` whose main arguments are used to pass the mesh object and to fix parameters α, τ and κ. Its arguments `B.tau` and `B.kappa` are matrices with one row for each mesh node. If only 1 row is given, it describes a model with stationary values of κ and τ, which will be duplicated internally for all nodes. For simulating a model with fixed parameters κ and τ, these matrices have only one column that must be specified as $\log \kappa$ or $\log \tau$ respectively. Then, we extract the precision matrix $\boldsymbol{Q}$ of the resulting SPDE model and use it to create independent samples of $\varepsilon(s,t)$, $t = 1, \ldots, 60$ through the function `inla.qsample`. We fix the random seed for simulation through its `seed=...` argument. Finally, we manually create the first order AR model with coefficient 0.5.

```
B.kappa=matrix(log(kappa),1,1)
B.tau=matrix(log(tau),1,1)
model.sim=inla.spde2.matern(mesh.sim,alpha=alpha,
   B.tau=B.tau,B.kappa=B.kappa)
Q=inla.spde2.precision(model.sim)
x=inla.qsample(n=n.repl,Q,seed=seed)
a=.5
for(i in 2:n.repl){x[,i]=a*x[,i-1]+sqrt(1-a^2)*x[,i]}
```

It remains to fix covariate values and to generate the time trend in the mean to add it to the centered Gauss–Markov space-time field. We fix an intercept -1 and the two covariates `covar1` `covar2`.
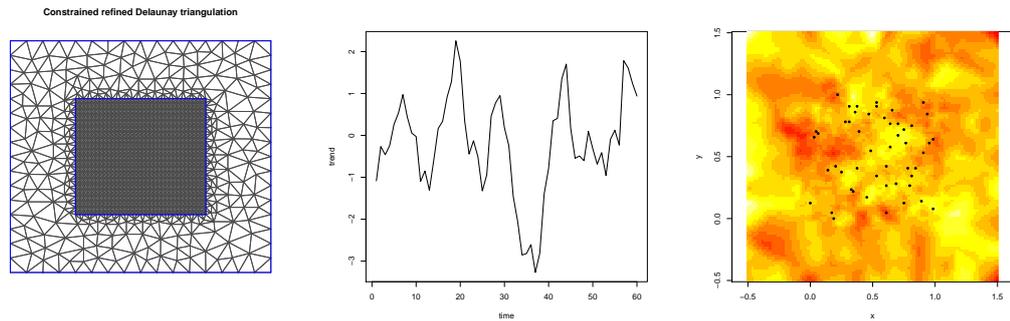
FIGURE 2. *Simulated latent Gaussian model. Left: triangulation used for the approximate SPDE simulation. Middle: time trend. Right: simulation for $t = 19$, with observation sites marked by black dots.*

```
covar1=1:n.repl/n.repl
covar2=as.numeric(arima.sim(n=n.repl,model=list(ma=rep(1,5))))
xtrend=-1+covar1+0.5*covar2
x=t(t(x)+xtrend)
plot(xtrend,type="l",xlab="time",ylab="trend",lwd=2)
```

For the observed data **y** to be used in estimation, we sample uniformly 50 sites among the triangulation nodes contained in $[0,1]^2$. By using R-INLA's methods inla.mesh.projector and inla.mesh.project to project a spatial field with known values for triangulation nodes onto a regular grid necessary for standard plotting methods, we further provide a plot of $W(s,19)$ and the observation sites. At $t = 19$, we observe the maximum value of the time trend, and we will later use R-INLA to do spatial prediction for $t = 19$.

```
nodes=mesh.sim$loc[,1:2]
idx.inside=which(pmin(nodes[,1],nodes[,2])>=0&pmax(nodes[,1],nodes[,2])<=1)
idx.obs=sample(idx.inside,size=n.sites)
sites=nodes[idx.obs,]
eta.i=as.numeric(x[idx.obs,])
y=rpois(length(eta.i),lambda=exp(eta.i))
t.pred=which.max(xtrend)
n.grid=100
grid=inla.mesh.projector(mesh.sim,dims=c(n.grid,n.grid))
image(grid$x,grid$y,inla.mesh.project(grid,field=x[,t.pred]),xlab="x",
   ylab="y",asp=1)
points(sites,pch=19, cex=.5)
```

Figure 2 shows the trend component $-1 + z_1(t) + 0.5z_2(t)$ (middle display) and the spatial field $W(s,19)$ at a fixed instant $t = 19$ with observation sites indicated (right display).

4.2. Fitting a complex space-time model with INLA

We now define and fit different candidate models to the above data y and its covariates covar1 and covar2. In Section 4.4, we will then explore tools for goodness-of-fit checks and model selection within the R-INLA package. We will first consider a model with prior structure similar to the simulated one, but we will also compare it to models where covariates are missing, where

Soumis au Journal de la Société Française de Statistique
File: INLASFDS.tex, compiled with jsfds, version : 2009/12/09
date: July 25, 2016

the Matérn shape parameter ν takes a different value or where the likelihood is not Poisson, but of the negative binomial type (with an additional hyperparameter for overdispersion).

First, let us define the triangulation mesh and the corresponding prior spatial SPDE model for estimation. For estimation, we must be careful about the dimension of the latent model to minimize memory requirements and high-dimensional matrix calculations. Therefore, will use a mesh with a lower resolution than for simulation, which may slightly increase the approximation error with respect to the stationary Matérn correlation structure. It is often reasonable to use observation sites as initial nodes of the triangulation and to refine it by adding nodes where necessary, or by removing nodes too close together which could be source for numerical instabilitities. R-INLA implements standard methods from the finite element literature and offers a conveniently parametrized interface to produce numerically stable and moderately dimensioned triangulations. R-INLA's prior parametrization of κ and τ is a bit technical; it essentially assumes that $\log(\tau) = b_{\tau,0} + \theta_1 b_{\tau,1} + \theta_2 b_{\tau,2} + ...$ and $\log(\kappa) = b_{\kappa,0} + \theta_1 b_{\kappa,1} + \theta_2 b_{\kappa,2} + ...$, where the values b_i correspond to the columns of B.tau and B.kappa. The first column b_0 is a fixed offset that must always be specified (even if it is 0), whereas the following columns correspond to hyperparameters θ_i that are estimated when $b_i \neq 0$ for τ or κ. For instance, specifying B.tau=B.kappa=matrix(c(0,1),nrow=1) would lead to a model where $\kappa = \tau = \exp(\theta_1)$. In our model, we fix the offset $b_0 = 0$ and we estimate two hyperparameters, one corresponding to $\log(\tau)$, the other to $\log(\kappa)$. This can be seen as the standard prior specification of the SPDE model in R-INLA. We fix the "correct" simulated value of α in the SPDE model.

```
mesh=inla.mesh.2d(sites,offset=c(-0.125,-.25),cutoff=0.075,min.angle=21,
    max.edge=c(.1,.25))
plot(mesh)
points(sites,col="red",pch=19,cex=.5)
spde=inla.spde2.matern(mesh=mesh,alpha=alpha,B.tau=matrix(c(0,1,0),nrow=1),
    B.kappa=matrix(c(0,0,1),nrow=1))
```

The mesh counts 294 nodes. Notice that further arguments of the inla.spde2.matern(...) function can be set to modify default priors, impose "integrate-to-zero" constraints, etc. We now run a first estimation by considering a model with Poisson likelihood and prior structure of the latent Gaussian field corresponding to the model that we simulated to generate the data. Naturally, this model should provide a good fit and we will later compare it to a number of alternative models. The observation matrix A links observations y to the latent variables x through $Ax = y$ and must therefore be of dimension $(50 \times 60) \times$(number of latent variables) for our model. Since construction of this matrix and certain preprocessing steps before estimation like the removal of duplicate rows is rather complicated for complex models involving the spatial SPDE solution, R-INLA has helper methods that allow constructing this matrix and keeping track of latent variable indices more easily. In the following, the inla.spde.make.index command creates an index named "spatial", *i.e.*, a data frame with a vector spatial (an index to match latent variables and triangulation nodes), a vector spatial.group (an index that indicates the membership of a latent variable in a "group" , here given as the instant $t \in \{1,\ldots,60\}$), and a vector spatial.repl (an index that indicates the group membership when groups are i.i.d. replicates). In our case, all values of spatial.repl are 1 since there is no structure of i.i.d. blocks in our space-time model due to the non-zero autoregression coefficient.

```
idx.spatial=inla.spde.make.index("spatial",n.spde=spde$n.spde,n.group=n.repl)
A.obs=inla.spde.make.A(mesh,loc=sites,index=rep(1:nrow(sites),n.repl),
    group=rep(1:n.repl,each=nrow(sites)))
stack.obs=inla.stack(data=list(y=y),A=list(A.obs,1),effects=list(idx.spatial,
    data.frame(intercept=1,covar1=rep(covar1,each=n.sites),
        covar2=rep(covar2,each=n.sites))),tag="obs")
```

In practice, we may want to use a fitted model for prediction at unobserved sites. A natural

Soumis au Journal de la Société Française de Statistique
File: INLASFDS.tex, compiled with jsfds, version : 2009/12/09
date: July 25, 2016

way to achieve prediction in the Bayesian framework of INLA is to add the prediction points
to the data by considering the associated observations as missing data. To illustrate `R-INLA`'s
facilities for this approach, we here consider prediction at instant $t = 19$ over a regular spa-
tial grid covering $[0,1]^2$. Therefore, we first create a separate observation matrix and a stack
for the prediction points with missing data, and we will then use `R-INLA`'s *join*-mecanism
that allows regrouping several groups of predictors $\boldsymbol{\eta}_k$ through their observation matrices $\boldsymbol{A}_k$.
The `inla.stack.join(stack1,stack2,...)`-function creates a single model corre-
sponding to a structure $(\boldsymbol{A}'_1, \boldsymbol{A}'_2, ...)' \boldsymbol{x} = (\boldsymbol{\eta}'_1, \boldsymbol{\eta}'_2, ...)'$, where information relative to $\boldsymbol{A}_k \boldsymbol{x} = \boldsymbol{\eta}_k$ is
regrouped in `stackk` for $k = 1, 2, \ldots$

```
n.grid=51
xgrid=0:(n.grid-1)/(n.grid-1)
grid.pred=as.matrix(expand.grid(xgrid,xgrid))
A.pred=inla.spde.make.A(mesh,loc=grid.pred,index=1:nrow(grid.pred),
   group=rep(t.pred,nrow(grid.pred)),n.spde=spde$n.spde,n.group=n.repl)
stack.pred=inla.stack(data=list(y=rep(NA, n.grid^2)),A=list(A.pred,1),
   effects=list(idx.spatial,data.frame(intercept=1,
      covar1=rep(covar1[t.pred],n.grid^2),
         covar2=rep(covar2[t.pred],n.grid^2))),tag="pred")
stack=inla.stack(stack.obs,stack.pred)
```

We now run the estimation with the `inla(...)` function. Its syntax ressembles that of R's
`glm(...)`-function for generalized linear models, although with a variety of extensions and addi-
tional arguments. We need a model formula given in the usual R notation. For better handling of the
intercept term, it is often preferable to make it appear explicity (`form=y -1+intercept+...`),
such that it can later be directly included into the latent space(-time) model. Fixed effects (*i.e.*,
the intercept and covariates whose linear regression coefficients are estimated) are added to
the formula in the usual way, whereas random effects are added with the `f(...)` function. In
our model, the approximate SPDE solution is a random effect. The first argument of `f` is the
name of the "covariate" associated to the random effect. Having created an index with name
`spatial` beforehand, we now have a covariate `spatial` in the data set that indicates the
triangulation node index of the spatial SPDE model (repeated 60 times since the spatial model is
duplicated for each observation time). For prior SPDE models, we further specify the argument
`model=spde` in `f`, where `spde` is the R object already created for the SPDE prior model.
The SPDE model is of purely spatial nature whereas we have observations in space and time,
such that we can use the *group*-functionality of `R-INLA` to define the type of dependence be-
tween the 60 groups of spatially indexed Gaussian variables. Corresponding to the simulated
model, we here use an AR(1)-group model that models site-wise first-order autoregression of
variables over time. Since we have created the index `spatial`, we can specify the argument
`group=spatial.group` to indicate group membership of the covariates, and we have to set
`control.group=list(model="ar1")` for the AR(1)-model between groups.

Data must be passed to the fitting function `inla(...)` as a `data.frame` or `list`, and the
`inla.stack.data`-function allows convenient extraction of the preprocessed `data` object
from the stack. Further control arguments to `inla(...)` can be specified explicitly through
R's usual `control.*=list(...)` syntax, which allows overriding the default control argu-
ments. Here, $\star$ should be replaced by one of `inla` (for controling INLA-related details like the
reordering scheme used for making the precision matrix as diagonal as possible), `compute`
(for specifying which quantities should be calculated, *e.g.* goodness-of-fit and model selection
criteria), `predictor` (for specifying the observation matrix $\boldsymbol{A}$ if there is one, and for indi-

Soumis au Journal de la Société Française de Statistique
File: INLASFDS.tex, compiled with jsfds, version : 2009/12/09
date: July 25, 2016

cating which posterior quantities should be calculated for the predictor vector $\boldsymbol{\eta}$), `family` (for modifying the default prior of likelihood hyperparameters), amongst others. In the following, we fix the METIS-reordering strategy in `control.inla` to avoid instabilities that have appeared during a preliminary fitting approach with the default reordering scheme. In `control.predictor`, we pass the observation matrix $\boldsymbol{A}$ that can be extracted from the stack via `inla.stack.A(stack)`, and we advise the program to calculate (discretized) posterior densities for the predictor variables η_i (`compute=T`). In `control.compute`, we here demand the calculation of CPO-values (cross-validated predictive measures, see Held et al. (2010) for a comparison of Markov chain Monte Carlo and INLA), the marginal likelihood $\pi(\boldsymbol{y})$, the Deviance Information Criterion (DIC) and the Watanabe–Akaike Information Criterion (WAIC, Watanabe (2010); Gelman et al. (2014)), where the latter can be considered as a Bayesian variant of the common AIC. The CPO-related values are the density and cdf of the posterior distribution $\pi(y_i \,|\, \boldsymbol{y}_{-i})$, evaluated at the observed y_i. With INLA, these cross-validation quantities can be calculated quickly without explicitly reestimating the model, see Rue et al. (2009) for details. However, certain theoretical assumptions might be violated such that some or all of these CPO-related values are not trustworthy, which is then indicated in the `inla`-output in `fit$cpo$failure`. In such a cases, the `inla.cpo(...)`-function can be used for "manual" reestimation of the cross-validated model for the concerned data points y_i. We now construct the `inla(...)`-call. For later use, we here store the data and parameters of this first model in an object `mod1`:

```
data=inla.stack.data(stack,spde=spde)
form=y~-1+intercept+covar1+covar2+f(spatial,model=spde,
   group=spatial.group,control.group=list(model="ar1"))
cc=list(cpo=T,dic=T,mlik=T,waic=T)
cp=list(A=inla.stack.A(stack),compute=T)
ci=list(reordering="metis")
mod1=list(stack=stack,data=data,A.pred=A.pred,A.obs=A.obs,
   idx.spatial=idx.spatial,spde=spde,form=form,cp=cp,ci=ci,cc=cc)
fit=inla(form,family="poisson",data=data,control.compute=cc,
   control.predictor=cp,control.inla=ci)
```

Since our likelihood is not Gaussian (but Poisson) and since the latent model is quite complex, the `inla` run takes some time (around 50 minutes on a state-of-the art 4 core machine), and memory requirements are rather high. We remark that `inla(...)` has a `num.threads` argument which allows fixing the maximum number of computation threads used by `R-INLA`. By default, `R-INLA` takes control over all available cores of the machine for parallel execution, which can lead to problems in terms of too high memory requirements since each thread occupies a certain amount of memory. As is usual in `R`, we can now call `summary(fit)` to obtain principal results of the fit. Part of its output is as follows:

[...]
Time used:
Pre-processing	*Running inla*	*Post-processing*	*Total*
1.0956	*2766.6987*	*1.0865*	*2768.8808*

Fixed effects:
	mean	*sd*	*0.025quant*	*0.5quant*	*0.975quant*	*mode*	*kld*
intercept	*-0.9094*	*0.1451*	*-1.1959*	*-0.9091*	*-0.6251*	*-0.9084*	*0*
covar1	*0.7827*	*0.2352*	*0.3193*	*0.7829*	*1.2444*	*0.7833*	*0*
covar2	*0.5092*	*0.0262*	*0.4579*	*0.5091*	*0.5610*	*0.5089*	*0*

Random effects:
Name	*Model*
spatial	*SPDE2 model*

Model hyperparameters:
	mean	*sd*	*0.025quant*	*0.5quant*	*0.975quant*	*mode*
Theta1 for spatial	*-3.5143*	*0.0670*	*-3.6459*	*-3.5142*	*-3.3828*	*-3.5139*

Theta2 for spatial   2.2750 0.0696   2.1385   2.2750   2.4119 2.2750
GroupRho for spatial 0.4892 0.0325   0.4256   0.4889   0.5532 0.4875

Expected number of effective parameters(std dev): 1020.18(24.28)
Number of equivalent replicates : 2.941
Deviance Information Criterion (DIC) ...: 8544.98
Effective number of parameters ....: 979.41
Watanabe-Akaike information criterion (WAIC) ...: 8522.71
Effective number of parameters ....: 745.45
Marginal log-Likelihood: -4752.98
[...]

We see that the posterior means of the autoregression coefficient (`GroupRho`) and of co-variate coefficients of `covar1` and `covar2` are not far from the actually simulated values. The object `fit` is of `list` type; its various slots contain a multitude of information. For our model, we could be interested in a better interpretable representation of the hyperparameter estimates of the spatial SPDE model in terms of effective range and variance. In the following, `inla.spde.result(...)` extracts the fitting result for the spatial index `spatial` associated to the SPDE. Then, for instance, `inla.qmarginal` allows extracting posterior marginal quantiles, and `inla.emarginal(FUN, ...)` calculates posterior marginal expectations of FUN(X), where FUN is a function and X is the posterior marginal distribution (note that expectations are particular since $\mathbb{E}\text{FUN}(X) \neq \text{FUN}(\mathbb{E}X)$).

```
result.spatial=inla.spde.result(fit,"spatial",spde)
inla.emarginal(identity,result.spatial$marginals.range.nominal[[1]])
```

[1] 0.2914313

```
inla.qmarginal(c(0.025,0.25,0.5,0.75,0.975),
   result.spatial$marginals.range.nominal[[1]])
```

[1] 0.2539053 0.2773829 0.2906817 0.3046177 0.3327678

```
inla.emarginal(identity,result.spatial$marginals.variance.nominal[[1]])
```

[1] 0.9515285

```
inla.qmarginal(c(0.025,0.25,0.5,0.75,0.975),
   result.spatial$marginals.variance.nominal[[1]])
```

[1] 0.8368486 0.9083829 0.9490067 0.9918896 1.0788070

We also plot a histogram of the cross-validated $\pi(y_i \mid \boldsymbol{y}_{-i})$ cdf values, and give a histogram of the `fit$cpo$failure` values (between 0 and 1), where values far from 0 indicate a violation of internal assumptions in the calculation of cdf values (see Figure 3 for the resulting displays):

```
hist(fit$cpo$pit,breaks=50,main="",xlab="PIT value",ylab="number")
hist(fit$cpo$failure,breaks=50,main="",xlab="failure indicator",ylab="number")
```

Cdf values are not very far from being uniform, which indicates that there are no strong systematic biases in posterior predictions made with the model. A certain proportion of the failure indicator values are above zero and some are even 1, meaning that there are some cdf values that must be interpreted with caution.

Since we have done prediction for $t = 19$, we now extract and visualize the marginal posterior mean and standard deviation over the prediction grid. To get the index of predicted variables, we can apply the `inla.stack.index` function to the `stack` object by indicating the prediction sub-stack through the argument `tag="pred"`. The following code visualizes the predictions $\hat{\eta}_i$ and the originally simulated Gaussian values on the prediction grid. The
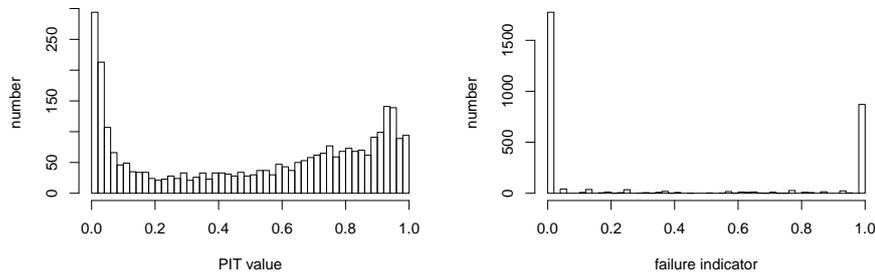
Soumis au Journal de la Société Française de Statistique
File: INLASFDS.tex, compiled with jsfds, version : 2009/12/09
date: July 25, 2016

FIGURE 3. *Left: histogram of Internally cross-validated cdf values for $\pi(y_i \mid \mathbf{y}_{-i})$. Right: histogram of the failure indicator of trustworthiness of calculated cdf values.*

`inla.mesh.projector` function allows switching between the finite element representation and a regular grid by calculating the Gauss–Markov finite element approximation value to the SPDE for the grid positions by using the "pyramid" basis functions to interpolate between triangulation nodes. We use the `inla.emarginal`-function for calculating posterior expectations, where the standard deviation is calculated as $\sqrt{\mathbb{E}X^2 - (\mathbb{E}X)^2}$ in the following code:

```
idx.pred=inla.stack.index(stack, tag="pred")$data
eta.marginals=fit$marginals.linear.predictor[idx.pred]
eta.mean=unlist(lapply(eta.marginals,function(x) inla.emarginal(identity,x)))
eta.mean=matrix(eta.mean,n.grid,n.grid)
image(x=xgrid,y=xgrid,eta.mean,asp=1,xlab="x",ylab="y",main="posterior mean")
proj=inla.mesh.projector(mesh.sim,xlim=c(0,1),ylim=c(0,1),dims=c(n.grid,n.grid))
image(grid$x,grid$y,inla.mesh.project(proj,field=x[,t.pred]),xlab="x",
   ylab="y",xlim=c(0,1),ylim=c(0,1), asp=1,main="original")
points(sites,pch=19,cex=1)
eta2.mean=unlist(lapply(eta.marginals,function(x) inla.emarginal("^",x,2)))
eta2.mean=matrix(eta2.mean,n.grid,n.grid)
eta.sd=sqrt(eta2.mean-eta.mean^2)
image(x=xgrid,y=xgrid,eta.sd,asp=1,xlab="x",ylab="y",main="posterior sd")
points(sites,pch=19,cex=1)
```

Figure 4 shows the resulting displays. As expected, uncertainty is lower close to observation sites. The prediction captures the spatial variation of the actual data, and a deeper analysis shows that the predicted surface is smoother than the original values: since the Gaussian prior on the predictor is centered at 0, the spatial variation in posterior predictions is naturally less strong owing to the Bayesian approach.

4.3. *Other candidate models*

In Section 4.2, we have used our knowledge about the simulated data model to construct the Bayesian model that should be the most appropriate. For comparison and to illustrate `R-INLA`'s functionality for other types of models, we here propose to fit some alternative candidate models.

For Model 2, we drop the temporal autoregression and consider the spatial blocks of SPDE variables as independent in the prior. This necessitates some (slight) adaptations in the code since
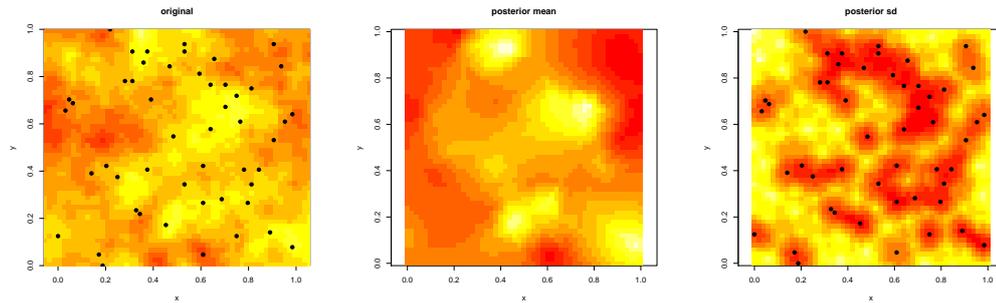
Soumis au Journal de la Société Française de Statistique
File: INLASFDS.tex, compiled with jsfds, version : 2009/12/09
date: July 25, 2016

FIGURE 4. *Predictions of the latent Gaussian field for t = 19. Left: original simulated data. Middle: posterior mean. Right: posterior standard deviation.*

there is no more group model with dependence between blocks, but we now have replicates, *i.e.*, independent blocks.

```
idx.spatial=inla.spde.make.index("spatial",n.spde=spde$n.spde,n.repl=n.repl)
A.obs=inla.spde.make.A(mesh,loc=sites,index=rep(1:nrow(sites),n.repl),
   repl=rep(1:n.repl,each=nrow(sites)))
stack.obs=inla.stack(data=list(y=y),A=list(A.obs,1),effects=list(idx.spatial,
   data.frame(intercept=1,covar1=rep(covar1,each=n.sites),
      covar2=rep(covar2,each=n.sites))),tag="obs")
A.pred=inla.spde.make.A(mesh,loc=grid.pred,index=1:nrow(grid.pred),
   repl=rep(t.pred,nrow(grid.pred)),n.spde=spde$n.spde,n.repl=n.repl)
stack.pred=inla.stack(data=list(y=NA),A=list(A.pred,1),effects=
   list(idx.spatial,data.frame(intercept=1,covar1=rep(covar1[t.pred],n.grid^2),
      covar2=rep(covar2[t.pred],n.grid^2))),tag="pred")
stack=inla.stack(stack.obs,stack.pred)
data=inla.stack.data(stack,spde=spde)
form=y~-1+intercept+covar1+covar2+f(spatial,model=spde,replicate=spatial.repl)
cp=list(A=inla.stack.A(stack),compute=T)
fit=inla(form,family="poisson",data=data,control.compute=mod1$cc,
   control.predictor=cp,control.inla=mod1$ci)
```

Model 3 focuses on the temporal trend, neglects spatial variation and supposes that covariates are not known. Here we use a temporal first-order random walk as prior model. We set a rather high initial prior value for the precision of the random walk innovations (corresponding to a standard deviation of 0.01). A sum-to-zero constraint is added (`constr=T`) for better identifiability.

```
data3=data.frame(y=y,covar1=rep(covar1,each=n.sites))
form=y~f(covar1,model="rw1",hyper=list(prec=list(initial=log(1/.01^2),
   fixed=F)),constr=T)
cp=list(compute=T)
fit=inla(form,family="poisson",data=data3,control.predictor=cp)
```

Another interesting model could be to obtained from combining Models 2 and 3, *i.e.*, using a random walk in time and a spatial SPDE model. Such a model is relatively complex if not overly complicated and its estimation is computationally demanding, so we do not consider it here. In Models 4 and 5, we specify a shape parameter of the Gauss–Markov Matérn model in the prior that is different from the simulated model, using either $\nu = 0.5$ (exponential model) or $\nu = 0$ (not a proper Matérn model, but still a valid covariance model).

```
nu=0.5 #model 4
```

Soumis au Journal de la Société Française de Statistique
File: INLASFDS.tex, compiled with jsfds, version : 2009/12/09
date: July 25, 2016

```
nu=0 #model 5
alpha=nu+1
spde=inla.spde2.matern(mesh=mesh,alpha=alpha,B.tau=matrix(c(0,1,0),nrow=1),
   B.kappa=matrix(c(0,0,1),nrow=1))
form=y~-1+intercept+covar1+covar2+f(spatial,model=spde,group=spatial.group,
   control.group=list(model="ar1"))
fit=inla(form,family="poisson",data=mod1$data,control.compute=mod1$cc,
   control.predictor=mod1$cp,control.inla=mod1$ci)
```

Finally, Model 6 uses not the Poisson likelihood but the negative binomial one that has an additional overdispersion parameter θ_{disp}, where the variance of the negative binomial distribution is $\mu + \mu^2/\theta_{\text{disp}}$ and μ is its mean. Notice that the Poisson distribution arises in the limit for $\theta_{\text{disp}} = \infty$. The internal parametrization of θ_{disp} considers $\log\theta_{\text{disp}}$ as a hyperparameter. We here fix a relatively high initial value $\log(10)$ and use a log-gamma prior with shape 10 and rate 1, such that the prior expectation of θ is 10 and yields a prior likelihood model relatively close to the Poisson one:

```
cf=list(hyper=list(list(theta=list(initial=log(10),prior="loggamma",
   param=c(10,1)))))
fit=inla(mod1$form,
family="nbinomial",data=mod1$data,control.compute=mod1$cc,
   control.family=cf,control.predictor=mod1$cp, control.inla=mod1$ci)
```

4.4. Analyzing fitted models

We now compare the 6 fitted models. For the purely temporal model 3, the following code plots a posterior mean estimate of the fitted temporal curve and the simulated data, see Figure 5:

```
plot(1:n.repl,fit$summary.fixed[1,1]+fit$summary.random$covar1$mean,type="l",
   xlab="t",ylab="time trend",lwd=2)
lines(1:n.repl,xtrend,col="blue",lwd=2)
```

We see that neglecting the spatial variation in data and using a relatively simple, purely temporal model here permits to reconstruct quite accurately the simulated temporal trend. Finally, we can compare the information criteria DIC and WAIC, marginal likelihoods, estimates of spatial range, variance and temporal autocorrelation over the candidate models, see Table 1. The marginal likelihood is relatively close for all space-time models with an explicit spatial structure and temporal autoregression (Models 1,4,5,6), and has its by far lowest value for the purely temporal model 3. Model 1 whose prior structure is closest to the simulated model turns out best in terms of DIC, but has slightly higher WAIC values than models 4 and 5 characterized by a different fixed shape parameter ν in the SPDE solution, leading to less smooth sample paths in the spatial prior random field. Notice however that differences in the estimated WAIC values between models 1,4 and 5 are relatively small such that they should be interpreted with caution. Model 6 with the negative binomial likelihood but with the same latent Gaussian prior model as Model 1 has rather high DIC and WAIC values, maybe due to approximations and computations that are less accurate for this fitted model. Concerning range and variance parameters of the fitted spatial models, we find that they are close to simulated values in all cases except Model 5, where the different prior shape parameter $\nu = 0$ in the SPDE seems having perturbed the calculations of the dependence structure and the variance. Throughout, the posterior distribution of the autoregression coefficient a (if estimated) is concentrated around the true value 0.5.
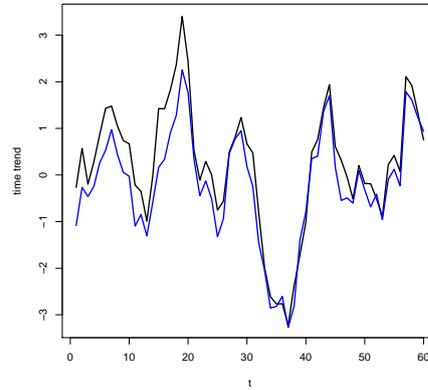
Soumis au Journal de la Société Française de Statistique
File: INLASFDS.tex, compiled with jsfds, version : 2009/12/09
date: July 25, 2016

FIGURE 5. *Purely temporal random walk model 3. Posterior mean curve (black) and simulated curve (blue).*

Model	DIC	WAIC	mlik	range	variance	*a*
1	8540	8520	−4750	0.29(0.25;0.33)	0.95(0.84;1.08)	0.49(0.43;0.55)
2	8660	8630	−4830	0.30(0.26;0.34)	0.92(0.82;1.03)	—
3	9530	9270	−5860	—	—	—
4	8550	8510	−4750	0.32(0.26;0.38)	1.03(0.91;1.16)	0.49(0.42;0.55)
5	8550	8500	−4760	0.59(0.43;0.79)	0.11(0.09;0.14)	0.48(0.42;0.55)
6	8890	8810	−4750	0.34(0.29;0.40)	0.82(0.70;0.96)	0.53(0.46;0.60)

TABLE 1. *Comparison of fitted models:* DIC, WAIC, *marginal likelihood* mlik, *spatial range, variance of spatial model, autoregression coefficient a. All values are rounded to three significant digits and to at most two decimals. Estimates are posterior means and 95% credible intervals are in parentheses.*

Soumis au Journal de la Société Française de Statistique
File: INLASFDS.tex, compiled with jsfds, version : 2009/12/09
date: July 25, 2016

5. Discussion

We have illustrated the theory and practice of Integrated Nested Laplace Approximation, implemented in the powerful `R-INLA` software library, which enables fast and accurate inference of complex Bayesian models. The dynamic behavior and dependence structure in the models covered by `R-INLA` is primarily governed by a latent Gaussian process for the mean of the univariate data distribution. In view of the near endless range of models that are available, one should perhaps also sound a note of caution since users may be mislead to construct overly complicated models in practice.

`R-INLA` has mechanisms to manage several different likelihoods in the same model, to use the same latent variables in different parts of the latent model ("copying") or to deal with non-informative missing data. Although the INLA approach does not directly provide the posterior dependence structure between predictors η_i or between the data distributions for different y_i, one could assume a Gaussian copula model with the Gaussian dependence given by the precision matrix $\boldsymbol{Q}^*$ of the Gaussian approximation (11) to obtain a practically useful approximation of the posterior dependence. Moreover, `R-INLA` has inbuilt functionality for calculating posterior densities of user-defined linear combinations of the latent variables, *i.e.*, for latent Gaussian variables other than the predictors η_i or the components of the latent model; see the `lincomb`-argument of `inla(...)`. New users should look around at www.r-inla.org, the main hub for staying informed about new INLA-related developments, for finding implementation details of `R-INLA` and for getting advice on specific problems via its discussion forum.

The SPDE approach, providing flexible spatial Gauss–Markov models, if of interest in its own far beyond the INLA framework where Markovian structures lead to fast high-dimensional matrix computations. Multivariate extensions (Hu et al., 2013) and certain nested version of SPDEs (Bolin and Lindgren, 2011) have already been proposed in the literature, although they are not yet available in `R-INLA`. Further modeling extensions in terms of data likelihoods and latent models can be expected in the near future. In particular, the construction of more complex and realistic Gauss–Markov space-time dependence structures based on the SPDE approach, capable to model effects like the nonseparability of space and time would be another big step forward.

References

Bisanzio, D., Giacobini, M., Bertolotti, L., Mosca, A., Balbo, L., Kitron, U., and Vazquez-Prokopec, G. M. (2011). Spatio-temporal patterns of distribution of West Nile virus vectors in eastern Piedmont Region, Italy. *Parasit Vectors*, 4:230.

Blangiardo, M. and Cameletti, M. (2015). *Spatial and Spatio-temporal Bayesian Models with R-INLA*. John Wiley & Sons.

Blangiardo, M., Cameletti, M., Baio, G., and Rue, H. (2013). Spatial and spatio-temporal models with R-INLA. *Spatial and spatio-temporal epidemiology*, 7:39–55.

Bolin, D. and Lindgren, F. (2011). Spatial models generated by nested stochastic partial differential equations, with an application to global ozone mapping. *The Annals of Applied Statistics*, pages 523–550.

Cameletti, M., Lindgren, F., Simpson, D., and Rue, H. (2013). Spatio-temporal modeling of particulate matter concentration through the SPDE approach. *AStA Advances in Statistical Analysis*, 97(2):109–131.

Cosandey-Godin, A., Krainski, E. T., Worm, B., and Flemming, J. M. (2014). Applying Bayesian spatiotemporal models to fisheries bycatch in the Canadian Arctic. *Canadian Journal of Fisheries and Aquatic Sciences*, 72(999):1–12.

Soumis au Journal de la Société Française de Statistique
File: INLASFDS.tex, compiled with jsfds, version : 2009/12/09
date: July 25, 2016

Fong, Y., Rue, H., and Wakefield, J. (2010). Bayesian inference for generalized linear mixed models. *Biostatistics*, 11(3):397–412.

Gelman, A., Hwang, J., and Vehtari, A. (2014). Understanding predictive information criteria for bayesian models. *Statistics and Computing*, 24(6):997–1016.

Held, L., Schrödle, B., and Rue, H. (2010). Posterior and cross-validatory predictive checks: a comparison of MCMC and INLA. In *Statistical modelling and regression structures*, pages 91–110. Springer.

Hu, X., Simpson, D., Lindgren, F., and Rue, H. (2013). Multivariate Gaussian random fields using systems of stochastic partial differential equations. *arXiv preprint arXiv:1307.1379*.

Illian, J. B., Sørbye, S. H., Rue, H., et al. (2012). A toolbox for fitting complex spatial point process models using integrated nested Laplace approximation (INLA). *The Annals of Applied Statistics*, 6(4):1499–1530.

Ingebrigtsen, R., Lindgren, F., and Steinsland, I. (2014). Spatial models with explanatory variables in the dependence structure. *Spatial Statistics*, 8:20–38.

Lindgren, F. and Rue, H. (2015). Bayesian spatial modelling with R-INLA. *Journal of Statistical Software*, 63(19).

Lindgren, F., Rue, H., and Lindström, J. (2011). An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498.

Martino, S., Akerkar, R., and Rue, H. (2011). Approximate Bayesian inference for survival models. *Scandinavian Journal of Statistics*, 38(3):514–528.

Martins, T. G., Simpson, D., Lindgren, F., and Rue, H. (2013). Bayesian computing with INLA: new features. *Computational Statistics & Data Analysis*, 67:68–83.

Rue, H. (2005). Marginal variances for Gaussian Markov random fields. Technical report. Norwegian Institute of Science and of Technology, Trondheim.

Rue, H. and Held, L. (2005). *Gaussian Markov random fields: theory and applications*. CRC Press.

Rue, H., Martino, S., and Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B*, 71(2):319–392.

Rue, H., Riebler, A., Sørbye, S. H., Illian, J. B., Simpson, D. P., and Lindgren, F. K. (2016). Bayesian computing with INLA: A review. *arXiv preprint arXiv:1604.00860*.

Schrödle, B. and Held, L. (2011). Spatio-temporal disease mapping using INLA. *Environmetrics*, 22(6):725–734.

Schrödle, B., Held, L., and Rue, H. (2012). Assessing the impact of a movement network on the spatiotemporal spread of infectious diseases. *Biometrics*, 68(3):736–744.

Serra, L., Saez, M., Mateu, J., Varga, D., Juan, P., Díaz-Ávalos, C., and Rue, H. (2014). Spatio-temporal log-Gaussian Cox processes for modelling wildfire occurrence: the case of Catalonia, 1994–2008. *Environmental and Ecological Statistics*, 21(3):531–563.

Tierney, L. and Kadane, J. B. (1986). Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, 81(393):82–86.

Watanabe, S. (2010). Asymptotic equivalence of bayes cross validation and widely applicable information criterion in singular learning theory. *The Journal of Machine Learning Research*, 11:3571–3594.

Yue, Y. R. and Rue, H. (2011). Bayesian inference for additive mixed quantile regression models. *Computational Statistics & Data Analysis*, 55(1):84–96.

Soumis au Journal de la Société Française de Statistique
File: INLASFDS.tex, compiled with jsfds, version : 2009/12/09
date: July 25, 2016