Bayesian spatial methods for large scale models: An introduction to `INLA` and `tmbstan`

Joaquin Cavieres

Universidad de Valparaíso
Ph.D (c) in Statistics

June 30, 2021

Contents

Introduction

Introduction

Commonly we have two paradigms for statistical modelling, for example:

Consider the following: $\boldsymbol{y}$ is a set of observations with distribution of probability $\pi(\boldsymbol{y} \mid \boldsymbol{\theta})$. For the above we can estimate $\boldsymbol{\theta}$ of two ways:

Frequentist approach

$\boldsymbol{\theta}$ denotes fixed and unknown parameters what can be estimated by maximum likelihood.

Bayesian approach

$\boldsymbol{\theta}$ denotes random variables with a prior $\pi(\boldsymbol{\theta})$ specification. We can estimate $\boldsymbol{\theta}$ based on the posterior:

$$\pi(\boldsymbol{\theta} \mid \boldsymbol{y}) = \frac{\pi(\boldsymbol{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\pi(\boldsymbol{y})} \propto \pi(\boldsymbol{y} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta}) \tag{1}$$

Introduction

Specifically, in the Bayesian framework we can use:

- Hierarchical models to consider complex structures and explain the behavior of our data
- Propose a model to calculate the uncertainty associated with the parameters and latent variables

Option 1: Integrated Nested Laplace Approximation (`INLA`)

This method was proposed by [Rue et al., 2009]. In summary, we can obtain the posterior distribution using numerical approximations.

Option 1: Integrated Nested Laplace Approximation (INLA)

This method was proposed by [Rue et al., 2009]. In summary, we can obtain the posterior distribution using numerical approximations. Advantage?

Option 1: Integrated Nested Laplace Approximation (INLA)

This method was proposed by [Rue et al., 2009]. In summary, we can obtain the posterior distribution using numerical approximations. Advantage?

Is not necessary sampling.

Option 1: Integrated Nested Laplace Approximation (INLA)

This method was proposed by [Rue et al., 2009]. In summary, we can obtain the posterior distribution using numerical approximations. Advantage?

Is not necessary sampling.

Software: INLA

- A Bayesian framework to analyze a specific class of models called Latent Gaussian Models
- Available in R (see `http://www.r-inla.org/`)

Option 2: Markov Chain Monte Carlo (MCMC)

The MCMC method use Markov chains to find a posterior stationary distribution based in sampling.

- The most used method to make Bayesian inference
- Commonly used in models where the likelihood is intractable
- It's very easy to use but with slow convergence in complex models

Option 2: Markov Chain Monte Carlo (MCMC)

The MCMC method use Markov chains to find a posterior stationary distribution based in sampling.

- The most used method to make Bayesian inference
- Commonly used in models where the likelihood is intractable
- It's very easy to use but with slow convergence in complex models

`Software: Stan`

- A full Bayesian framework that uses Hamiltoniano Monte Carlo.
- Available in `R` (see `https://mc-stan.org/`)

Introduction

In this presentation we will see two ways to fit efficient high dimensional models (large scale models):

Introduction

In this presentation we will see two ways to fit efficient high dimensional models (large scale models):

Option 1

- Integrated Nested Laplace Approximation (`INLA`)

In this presentation we will see two ways to fit efficient high dimensional models (large scale models):

Option 1

- Integrated Nested Laplace Approximation (INLA)

Option 2

- Template Model Builder (TMB) and Stan via $\Longrightarrow$ tmbstan

Spatial random field

Gaussian random field

The classic definition of a Gaussian random field is:

Let $\boldsymbol{s}$ a location in a particular area D and $u(\boldsymbol{s})$ is the random effect (spatial) at that location. $u(\boldsymbol{s})$ is a stochastic process with $\boldsymbol{s} \in \boldsymbol{D}$ and $\boldsymbol{D} \subset \mathbb{R}^d$ is the spatial domain where the observations were measured. $u(\boldsymbol{s}_i)$ is one realization of $u(\boldsymbol{s})$ where $i = 1, \ldots, n$ locations. We asumme that $u(\boldsymbol{s})$ has a multivariate Gaussian distribution (GRF), continuous over the space indexed by $\boldsymbol{s}$ and defined by the mean and the covariance ([Cressie, 1993])

Matérn family (covariance)

Continuous domain spatial

- Matérn covariance familiy on $s \in \mathbb{R}^d$:

$$r_M(\boldsymbol{s}_1, \boldsymbol{s}_2) = C(u(\boldsymbol{s}_1), u(\boldsymbol{s}_2))$$
$$= \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} (\kappa \|\boldsymbol{s}_2 - \boldsymbol{s}_1\|)^{\nu} K_{\nu}(\kappa \|\boldsymbol{s}_2 - \boldsymbol{s}_1\|)$$

with scale $\kappa > 0$, shape/smoothness $\nu > 0$ and K_ν a modified Bessel function.

Fields with Matérn covariances are solutions of the SPDE method (Whittle, 1954,1963) base on the Laplacian $\triangle = \nabla^T \nabla$:

$$(\kappa^2 - \Delta)^{\alpha/2} u(\boldsymbol{s}) = \mathscr{W}(\boldsymbol{s})$$

where $\mathscr{W}(\boldsymbol{s})$ is the spatial white noise, $\alpha = \nu + d/2$ and

$$\sigma^2 = \frac{\Gamma(\nu)}{\Gamma(\alpha)(4\pi)^{d/2}\kappa^{2\nu}\tau^2}$$

Hilbert space approximation

A finite Hilbert space uses a set of N basis functions $\{\psi_k\}$ and weights $\{w_k\}$ for that:

$$u(\boldsymbol{s}) = \sum_{k=1}^{n} \psi_k(\boldsymbol{s}) w_k$$

where $\psi(\cdot)$ are deterministic basis functions and $\{u_1, \ldots, u_n\}$ is a vector of weights that is chosen so that the distribution of the functions $u(\boldsymbol{s})$ approximates the distribution of solutions to the SPDE on the domain

SPDE

Construction of the $Q = \Sigma^{-1}$

To obtain a Markov structure we using piecewise polynomial basis functions with compact support (essentially it's a Finite Element method)

For a domain two dimensional we use picewise linear basis functions defined by a triangulation of the domain of interest

$$\boldsymbol{Q} = \tau^2 \left(\kappa^4 \boldsymbol{C} + 2\kappa^2 \boldsymbol{G}_1 + \boldsymbol{G}_2\right)$$

where by default $\alpha = 2$ so that the elements of $\boldsymbol{Q}$ have explicit expressions as functions of κ and τ. Assumming a multivariate Gaussian distribution $\boldsymbol{u} \sim \mathcal{N}(0, \boldsymbol{Q})$ now this is approximate solution to the SPDE (in a stochastically weak sense) [Lindgren et al., 2015]

More references: [Rue and Held, 2005], [Lindgren et al., 2011], [Blangiardo and Cameletti, 2015]

SPDE

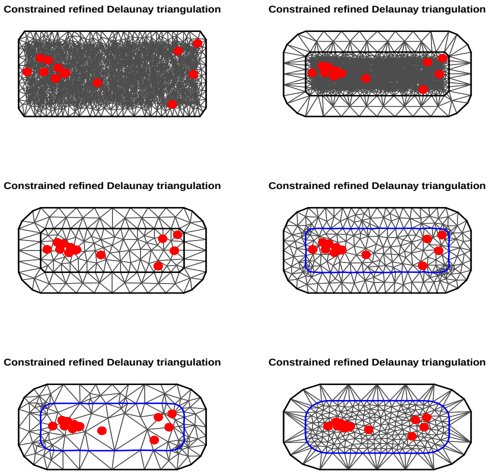


Figure: Differents SPDE/GMRF triangulations

INLA

Idea behind of a Latent Gaussian Model

For example: Multiple linear regression model

$$\mu_i = \mathbb{E}(y_i) = \beta_0 + \sum_{j=1}^{n_\beta} \beta_j x_{ji}, \quad i = 1, ..., n$$

where β_0 is the intercept and $\boldsymbol{\beta}$ are the parameters related to the covariates $\boldsymbol{x}$.

Idea behind of a Latent Gaussian Model

Generalized additive model (GAM)

$$\eta_i = g(\mu_i) = \beta_0 + \sum_{k=1}^{n_f} f_k(c_{ki}), \quad i = 1, ..., n$$

where $g(\cdot)$ es a link function, β_0 is the intercept, $f_k(\cdot)$ is the non-linear smopoth effects of the covariates $\boldsymbol{c}_k$.

Idea behind of a Latent Gaussian Model

A more complete general structure

$$\eta_i = g(\mu_i) = \beta_0 + \sum_{j=1}^{n_\beta} \beta_j x_{ji} + \sum_{k=1}^{n_f} f_k(c_{ki}), \quad i = 1, ..., n$$

where $g(\cdot)$ es a link function, β_0 is the intercept related to the covariates $\boldsymbol{x}$, $f_k(\cdot)$ is the non-linear smopoth effects of the covariates $\boldsymbol{c}_k$.

So, we collect all the parameters of the linear predictor in a latent field

$$\boldsymbol{u} = \{\beta_0, \boldsymbol{\beta}, \{f_k(\cdot)\}, \boldsymbol{\eta}\}$$

and, in this way, we can assign a Gaussian prior to all the elements of $\boldsymbol{u}$.

Latent Gaussian Models

Hierarchical model

Likelihood

$$\boldsymbol{y} \mid \boldsymbol{u}, \theta_1 \sim \prod_{i \in \mathrm{I}} \pi(y_i \mid x_i, \theta_1)$$

LGM

$$\boldsymbol{u} \mid \theta_2 \sim \mathcal{N}(\mu(\theta_2), \boldsymbol{Q}^{-1}(\theta_2))$$

Posterior

$$\pi(\boldsymbol{u}, \boldsymbol{\theta} \mid \boldsymbol{y}) \propto \pi(\boldsymbol{\theta})\pi(\boldsymbol{u} \mid \boldsymbol{\theta}) \prod_{i \in I} \pi(y_i \mid u_i, \boldsymbol{\theta})$$

Integrated Nested Laplace Approximation (INLA)

`INLA` is an alternative method of estimation to the classical algorithms implemented in the MCMC method for Bayesian inference. The algorithms in the MCMC method are asymptotically exact, instead INLA is a method of approximation.

Types of models that we can use with `INLA`:

- Generalized Linear Models (GLM)
- Generalized Linear Mixed Models (GLMM)
- Time series models
- Spatial models
- Spatio-temporal models

How works INLA?

From the posterior distribution

$$\pi(\boldsymbol{u}, \boldsymbol{\theta} \mid \boldsymbol{y}) \propto \pi(\boldsymbol{\theta}) \pi(\boldsymbol{u} \mid \boldsymbol{\theta}) \pi(\boldsymbol{y} \mid \boldsymbol{u}, \boldsymbol{\theta}) \tag{2}$$

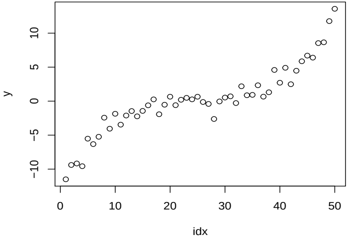
we can approximate the posterior marginals directly

$$\pi(u_i \mid y) \quad \text{and} \quad \pi(\theta_j \mid y) \tag{3}$$

How works INLA?

$$y_i = u(i) + \epsilon_i \quad i = 1, ..., n$$

We assumed that $u(i)$ es a smooth function indexed by i and $\epsilon_i \sim \mathcal{N}(0, \tau_0)$ with known parameter τ_0.



Figure

How we only are interesed in posterior marginals, the INLA method can be summarized as:

1 Step 1: Find a Laplace approximation to $\pi(\boldsymbol{\theta} \mid \boldsymbol{y})$.

2 Step 2: Find an approximation to $\pi(u_i \mid \boldsymbol{\theta}, \boldsymbol{y})$:

- Gaussian approximation
- Laplace approximation
- Simplified Laplace approximation

3 Step 3: Numerical integration

- Grid strategy
- Central composite design (CCD)

Step 1: Laplace approximation to $\pi(\boldsymbol{\theta} \mid \boldsymbol{y})$

$$\tilde{\pi}(\boldsymbol{\theta}|\mathrm{y}) = \left.\frac{\pi(\mathrm{u}, \boldsymbol{\theta}|\mathrm{y})}{\tilde{\pi}_G(\mathrm{u}|\boldsymbol{\theta}, \mathrm{y})}\right|_{\mathrm{u}=\mathrm{u}^*(\boldsymbol{\theta})}$$

and evaluate this at the mode $\boldsymbol{u}^*(\boldsymbol{\theta})$. Consider,

$$\pi(\mathrm{u}, \boldsymbol{\theta}|\mathrm{y}) = \pi(\boldsymbol{\theta})\pi(\mathrm{u}|\boldsymbol{\theta}) \prod_i \pi\left(y_i|u_i, \boldsymbol{\theta}\right)$$

Step 2: Marginals of the latent field

Marginals of u_i:

$$\tilde{\pi}\left(u_i|\mathrm{y}\right) = \int \pi\left(u_i|\boldsymbol{\theta}, \mathrm{y}\right)\tilde{\pi}(\boldsymbol{\theta}|\mathrm{y})d\boldsymbol{\theta}$$

We have three alternatives:

- Gaussian approximation
- Laplace approximation
- Simplified Laplace approximation

Step 2: Marginals of the latent field

- Gaussian distribution derived from $\hat{\pi}_G(u \mid \boldsymbol{\theta}, y)$

$$\hat{\pi}(u_i \mid \boldsymbol{\theta}, y) = \mathcal{N}(u_i; \mu_i(\boldsymbol{\theta}), \sigma^2(\boldsymbol{\theta}))$$

with mean $\mu_i(\boldsymbol{\theta})$ and marginal variance $\sigma^2(\boldsymbol{\theta})$.

INLA

Step 2: Marginals of the latent field

- Laplace approximation

$$\hat{\pi}_{LA}(u_i \mid \boldsymbol{\theta}, y) \propto \frac{\pi(u, \boldsymbol{\theta}, y)}{\hat{\pi}_G(u_{-1} \mid u_i, \boldsymbol{\theta}, y)}\Bigg|_{u_{-i}=x^\star(u_i, \boldsymbol{\theta})}$$

The approximation is very good but expensive due to "n" factorizations of $(n-1) \times (n-1)$ matrices (required to get the n marginals).

Step 2: Marginals of the latent field

- Simplified Laplace approximation
  - Based on a series expansion up to third order of the numerator and denominator of $\hat{\pi}_{LA}(u_i \mid \boldsymbol{\theta}, y)$
  - Corrects the Gaussian approximation for error in location and lack of skewness (fit a Skew-Normal density!).
  - Very much faster: $\mathscr{O}(n \log(n))$ for each i.

This is default option when using `INLA` but this choice can be modified.

Step 3: Numerical integration

Considering Step 1:

$$\tilde{\pi}\left(u_i|\mathrm{y}\right) = \int \tilde{\pi}\left(u_i|\boldsymbol{\theta}, \mathrm{y}\right)\tilde{\pi}(\boldsymbol{\theta}|\mathrm{y})d\boldsymbol{\theta}$$

and Step 2:

$$\tilde{\pi}\left(\theta_j|\mathrm{y}\right) = \int \tilde{\pi}(\boldsymbol{\theta}|\mathrm{y})d\boldsymbol{\theta}_{-j}$$

The numerical integration to approximate the marginals of the latent field:

$$\tilde{\pi}\left(u_i|\mathrm{y}\right) \approx \sum_k \tilde{\pi}\left(u_i|\theta_k, \mathrm{y}\right)\tilde{\pi}\left(\theta_k|\mathrm{y}\right)\Delta_k$$

where Δ_k is the weigth corresponding to θ_k.

INLA

See examples in R

Template Model Builder (TMB)

What is `TMB`?

- Template model builder (TMB) is an open source R package that enables quick implementation of complex nonlinear random effect (latent field) [Kristensen et al., 2015]
- It offers easy access to parallel computations and the user can defines the joint likelihood for the data and the random effects in a C++ template function, while all the other operations are done in `R` .

Why use TMB?

TMB can calculate first and second order derivatives of the likelihood function using ADMB ([Fournier et al., 2012]) or any objective function written in C++.

- The objective function (and its derivatives) can be called from R.
- The parameter estimation is done by nlminb() function.
- The user can use the Laplace approximation to obtain the marginal likelihood for a latent feld.
- Compute the standard deviations of any parameter, or derived parameter by the 'delta method'.

What is `tmbstan`?

- `tmbstan` facilitates linkage between `TMB` and `Stan` while adding a few top-level options ([Monnahan and Kristensen, 2018])
- `TMB` integrates random effects across Laplace appproximation while `Stan` integrates fixed effects with Hamiltonian Monte Carlo.

What is `tmbstan`?

- `tmbstan` facilitates linkage between `TMB` and `Stan` while adding a few top-level options ([Monnahan and Kristensen, 2018])
- `TMB` integrates random effects across Laplace appproximation while `Stan` integrates fixed effects with Hamiltonian Monte Carlo.

With `tmbstan` we can do the Bayesian inference and thus works with `Stan` tools automatically!!!

Framework

- Let $\boldsymbol{u}$ a vector of latent variables (latent field), $\boldsymbol{\theta}$ a vector of parameters and $g(\boldsymbol{u})$ is the joint loglikelihood.
- Assume a function $g(\boldsymbol{u}, \boldsymbol{\theta})$ is such that:

$$\hat{\boldsymbol{u}}(\boldsymbol{\theta}) = \underset{u}{\text{argmin}}\, g(\boldsymbol{u}, \boldsymbol{\theta})$$

and

$$\boldsymbol{H}(\boldsymbol{\theta}) = \frac{\partial^2}{\partial \boldsymbol{u}^2} g(\boldsymbol{u}, \boldsymbol{\theta})\Big|_{\boldsymbol{u}=\hat{\boldsymbol{u}(\boldsymbol{\theta})}}$$

The goal is to derive an efficient algorithm to maximize the Laplace approximation:

$$L^{\star}(\boldsymbol{\theta}) = |\det\{\boldsymbol{H}(\boldsymbol{\theta})\}|^{-1/2} exp[g\{\boldsymbol{u}(\boldsymbol{\theta})\}]$$

with respect to $\boldsymbol{\theta}$.

Hierarchical model

Let $\boldsymbol{y} = (y_1, ..., y_n)^T$ a vector of observations and $\boldsymbol{u} = (u_1, ..., u_q)^T$ a vector a latent variables (random effects) that have influence about $\boldsymbol{y}$. The conditional density of $\boldsymbol{y}$ given $\boldsymbol{u}$ is denoted by $f(\boldsymbol{y} \mid \boldsymbol{u})$ and the marginal density of $\boldsymbol{u}$ is denoted by $h(\boldsymbol{u})$.

The likelihood for $\boldsymbol{\theta}$ must be based on the marginal distribution of $\boldsymbol{y}$ (which is obtained by integrating out $\boldsymbol{u}$ from the joint density $f_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{u})h_{\boldsymbol{\theta}}(\boldsymbol{u})$. So, the marginal likelihood is:

$$L(\boldsymbol{\theta}) = \int f_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{u})h_{\boldsymbol{\theta}}(\boldsymbol{u})d\boldsymbol{u} = \int exp\{g(\boldsymbol{u}, \boldsymbol{\theta})\}d\boldsymbol{u}$$

where $g(\boldsymbol{u}, \boldsymbol{\theta}) = log\{f_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{u})\} + log\{h_{\boldsymbol{\theta}}(\boldsymbol{u})\}$

Hierarchical model

The Laplace approximation of the marginal loglikelihood is given as:

$$\ell^{\star}(\boldsymbol{\theta}) = -0.5 \ \log \ |\det\{\boldsymbol{H}(\boldsymbol{\theta})\}| + g\{\hat{\boldsymbol{u}}(\boldsymbol{\theta}, \boldsymbol{\theta})\}$$

where $\hat{\boldsymbol{u}}(\boldsymbol{\theta})$ and $\boldsymbol{H}(\boldsymbol{\theta})$ are tipically not available in closed-form.

Hierarchical model

Example:

$$y \sim \beta_0 + \beta_1 x_1 +, ..., +\beta_k x_k + \text{random effects}$$

(a) Response variable
  - Poisson, Gaussian, Binomial, Gamma, etc

(b) Random effects
  - Non linear trends with respect to the space, time or covariates.
  - Structured (e.g., space or time dependence) or unstructured (e.g., measurement error).

TMB

See examples in R

Conclusions

Conclusions

Option1: INLA

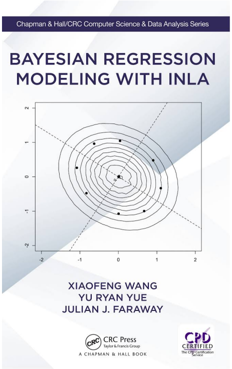
- TMB is a frequentist statistical platform that woks with AD (Automatic dierentiation) to obtain the first and second derivates of the a function.
- Stan ([Gelman et al., 2015], [Carpenter et al., 2017] is a probabilistic programming language that provides full Bayesian inference for continuous-variable models through Markov chain Monte Carlo method (Using the dynamic Hamiltonian Monte Carlo algorithm).

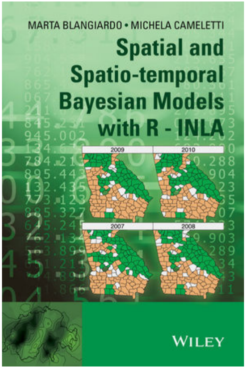
Option2: TMB and Stan

- TMB is a frequentist statistical platform that woks with AD (Automatic dierentiation) to obtain the first and second derivates of the a function.
- Stan ([Gelman et al., 2015], [Carpenter et al., 2017] is a probabilistic programming language that provides full Bayesian inference for continuous-variable models through Markov chain Monte Carlo method (Using the dynamic Hamiltonian Monte Carlo algorithm).

Additional

INLA





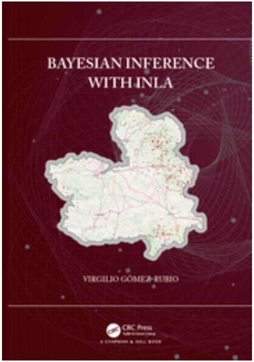


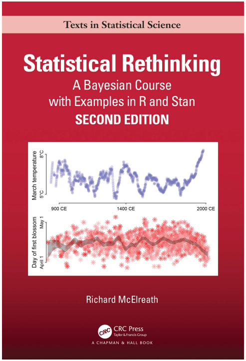
Figure: Some books to learn about INLA

Lectures

TMB

- see https://kaskr.github.io/adcomp/Introduction.html
- see https://kaskr.github.io/adcomp/_book/Introduction.html
- see https://arxiv.org/pdf/1509.00660.pdf

Stan



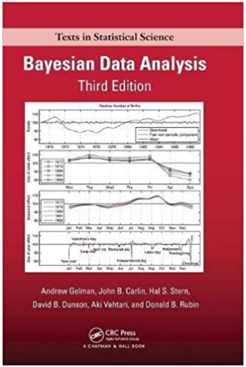


Figure: Some books to learn about Stan

References I

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

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M. A., Guo, J., Li, P., and Riddell, A. (2017).
Stan: a probabilistic programming language.
Grantee Submission, 76(1):1–32.

Cressie, N. A. (1993).
Spatial prediction and kriging.
Statistics for Spatial Data (Cressie NAC, ed). New York: John Wiley & Sons, pages 105–209.

References II

Fournier, D. A., Skaug, H. J., Ancheta, J., Ianelli, J., Magnusson, A., Maunder, M. N., Nielsen, A., and Sibert, J. (2012).
Ad model builder: using automatic differentiation for statistical inference of highly parameterized complex nonlinear models.
Optimization Methods and Software, 27(2):233–249.

Gelman, A., Lee, D., and Guo, J. (2015).
Stan: A probabilistic programming language for bayesian inference and optimization.
Journal of Educational and Behavioral Statistics, 40(5):530–543.

Kristensen, K., Nielsen, A., Berg, C. W., Skaug, H., and Bell, B. (2015).
Tmb: automatic differentiation and laplace approximation.
arXiv preprint arXiv:1509.00660.

References III

Lindgren, F., Rue, H., et al. (2015).
Bayesian spatial modelling with r-inla.
Journal of Statistical Software, 63(19):1–25.

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.

Monnahan, C. C. and Kristensen, K. (2018).
No-u-turn sampling for fast bayesian inference in admb and tmb: Introducing the adnuts and tmbstan r packages.
PloS one, 13(5):e0197954.

References IV

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 (statistical methodology), 71(2):319–392.