

Thinking Globally: The Role of Big Data

Gavin Shaddick, Daniel Simpson & Karim Anaya-Izquierdo
University of Bath

23rd - 24th February 2016

Introduction

OUTLINE

Tuesday, February 23

- ▶ 10:00 - 10:30 Introduction
- ▶ 10:30 - 11:30 Scalable Spatial Modelling: Working with Big Data
- ▶ 11:30 - 12:00 Break
- ▶ 12:00 - 13:30 Scalable Spatial Modelling: Working with Big Data (Contd.)
- ▶ 13:30 - 15:00 Lunch
- ▶ 15:00 - 17:00 Computer Labs

OUTLINE

Wednesday, February 24

- ▶ 10:00 - 11:30 Regional / area based modelling
- ▶ 11:30 - 12:00 Break
- ▶ 12:00 - 13:30 Beyond space: modelling with multi-dimensional responses
- ▶ 13:30 - 15:00 Lunch
- ▶ 15:00 - 17:00 Computer Labs

COURSE TEXTBOOK

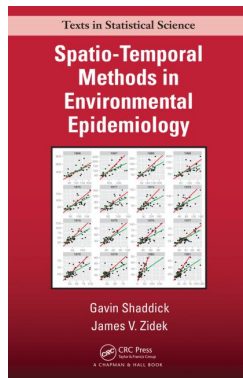
Title: Spatio-Temporal Methods in Environmental Epidemiology

Authors: Gavin Shaddick and Jim Zidek

Publisher: CRC Press

Resource Website:

<http://www.stat.ubc.ca/~gavin/STEPIDBookNewStyle/>



CONTACT INFORMATION

Dr. Gavin Shaddick, University of Bath

- ▶ Email: G.Shaddick@bath.ac.uk
- ▶ Webpage: <http://people.bath.ac.uk/masgs/>

Dr. Daniel Simpson, University of Bath

- ▶ Email: D.Simpson@bath.ac.uk

Dr. Karim Anaya-Izquierdo, University of Bath

- ▶ Email: K.Anaya-Izquierdo@bath.ac.uk
- ▶ Webpage: <http://people.bath.ac.uk/kai21/>

THE NEED FOR SPATIO-TEMPORAL MODELLING

- ▶ Spatial epidemiology is the description and analysis of geographical data, specifically health data in the form of counts of mortality or morbidity and factors that may explain variations in those counts over space.
- ▶ These may include demographic and environmental factors together with genetic, and infectious risk factors.
- ▶ It has a long history dating back to the mid-1800s when John Snow's map of cholera cases in London in 1854 provided an early example of geographical health analyses that aimed to identify possible causes of outbreaks of infectious diseases.

EXAMPLE: JOHN SNOW'S CHOLERA MAP

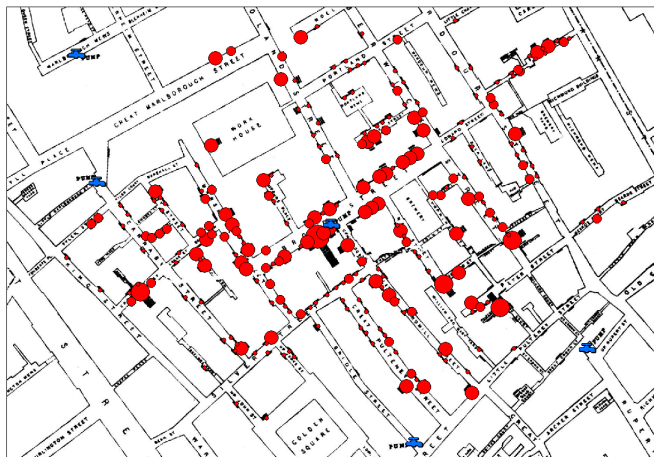


Figure: John Snow's map of cholera cases in London 1854. Red circles indicate locations of cholera cases and are scaled depending on the number of reported cholera cases. Purple taps indicate locations of water pumps.

THE NEED FOR SPATIO-TEMPORAL MODELLING AND THE ROLE OF BIG DATA

- ▶ Advances in statistical methodology together with the increasing availability of data recorded at very high spatial and temporal resolution has lead to great advances in spatial and, more recently, spatio-temporal epidemiology.
- ▶ These advances have been driven in part by increased awareness of the potential effects of environmental hazards and potential increases in the hazards themselves.

THE NEED FOR SPATIO-TEMPORAL MODELLING AND THE ROLE OF BIG DATA

- ▶ Over the past two decades, population predictions based on conventional demographic methods have forecast that the world's population will rise to about 9 billion in 2050, and then level off or decline.
- ▶ However, recent analyses using Bayesian methods have provided compelling evidence that such projections may vastly underestimate the world's future population and instead of the expected decline, population will continue to rise.
- ▶ Such an increase will greatly add to the anthropogenic contributions of environmental contamination and will require political, societal and economic solutions in order to adapt to increased risks to human health and welfare.

THE NEED FOR SPATIO-TEMPORAL MODELLING AND THE ROLE OF BIG DATA

- ▶ In order to assess and manage these risks there is a requirement for monitoring and modelling the associated environmental processes that will lead to an increase in a wide variety of adverse health outcomes.
- ▶ Addressing these issues will involve a multi-disciplinary approach and it is imperative that the uncertainties that will be associated with each of the components can be characterised and incorporated into statistical models used for assessing health risks.

EXAMPLE: GLOBAL MODELLING OF $PM_{2.5}$ USING MULTIPLE DATA SOURCES

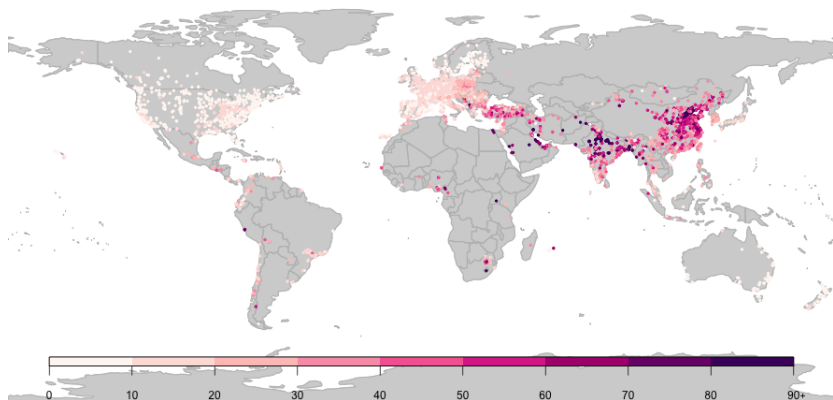


Figure: World map with ground monitor locations, coloured by the estimated level of $PM_{2.5}$

EXAMPLE: GLOBAL MODELLING OF $PM_{2.5}$ USING MULTIPLE DATA SOURCES

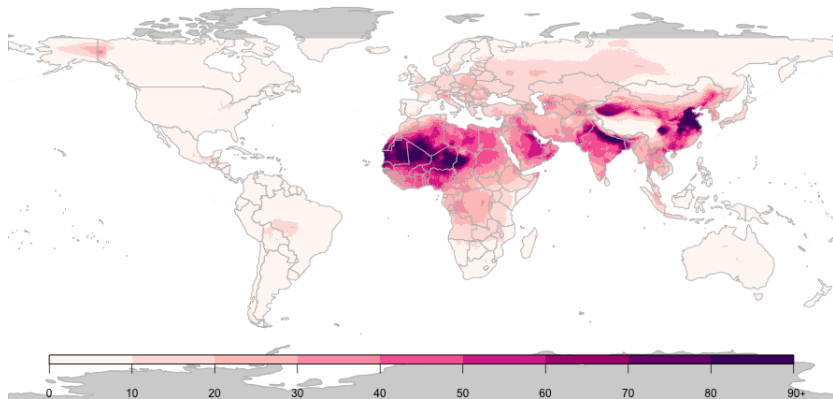


Figure: Global satellite remote sensing estimates of $PM_{2.5}$ for 2014.

EXAMPLE: GLOBAL MODELLING OF $PM_{2.5}$ USING MULTIPLE DATA SOURCES

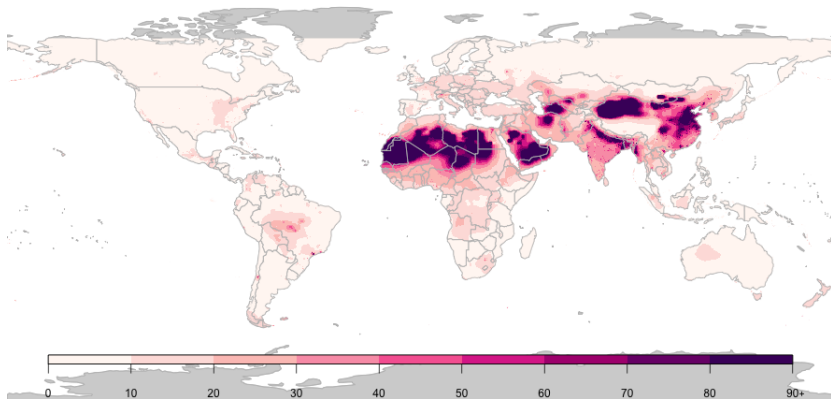


Figure: Global chemical transport model estimates of $PM_{2.5}$ for 2014.

Scalable Bayesian Modelling

INLA - INTEGRATED NESTED LAPLACE APPROXIMATIONS

Outline

- ▶ Describe the class of models INLA can be applied to.
- ▶ Look at simple examples in R-INLA.

TWO MAIN PARADIGMS FOR STATISTICAL ANALYSIS

- ▶ Let \mathbf{y} denote a set of observations, distributed according to a probability model $\pi(\mathbf{y}; \boldsymbol{\theta})$.
- ▶ Based on the observations, we want to estimate $\boldsymbol{\theta}$.

The classical approach:

$\boldsymbol{\theta}$ denotes **parameters** (unknown fixed numbers), estimated for example by maximum likelihood.

The Bayesian approach:

$\boldsymbol{\theta}$ denotes **random variables**, assigned a **prior** $\pi(\boldsymbol{\theta})$. Estimate $\boldsymbol{\theta}$ based on the **posterior**:

$$\pi(\boldsymbol{\theta} | \mathbf{y}) = \frac{\pi(\mathbf{y} | \boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\pi(\mathbf{y})} \propto \pi(\mathbf{y} | \boldsymbol{\theta})\pi(\boldsymbol{\theta}).$$

TWO MAIN PARADIGMS FOR STATISTICAL ANALYSIS

- ▶ Let \mathbf{y} denote a set of observations, distributed according to a probability model $\pi(\mathbf{y}; \boldsymbol{\theta})$.
- ▶ Based on the observations, we want to estimate $\boldsymbol{\theta}$.

The classical approach:

$\boldsymbol{\theta}$ denotes **parameters** (unknown fixed numbers), estimated for example by maximum likelihood.

The Bayesian approach:

$\boldsymbol{\theta}$ denotes **random variables**, assigned a **prior** $\pi(\boldsymbol{\theta})$. Estimate $\boldsymbol{\theta}$ based on the **posterior**:

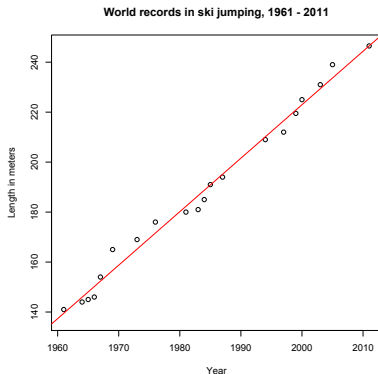
$$\pi(\boldsymbol{\theta} | \mathbf{y}) = \frac{\pi(\mathbf{y} | \boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\pi(\mathbf{y})} \propto \pi(\mathbf{y} | \boldsymbol{\theta})\pi(\boldsymbol{\theta}).$$

Example (Ski jumping records)

Assume a simple linear regression model with Gaussian observations

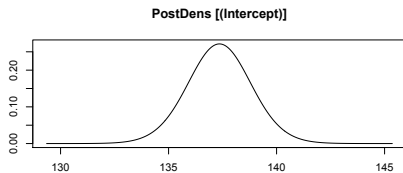
$\mathbf{y} = (y_1, \dots, y_n)$, where

$$E(y_i) = \alpha + \beta x_i, \quad \text{Var}(y_i) = \tau^{-1}, \quad i = 1, \dots, n$$

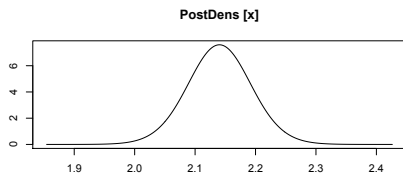


THE BAYESIAN APPROACH

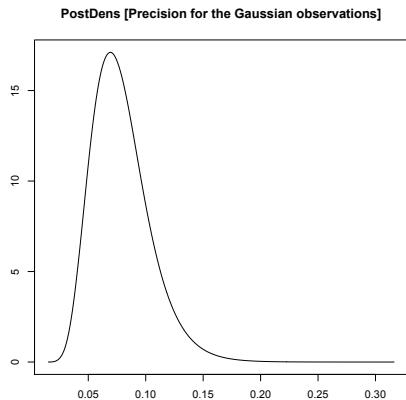
Assign priors to the parameters α , β and τ and calculate posteriors:



Mean = 137.354 SD = 1.508



Mean = 2.14 SD = 0.054



REAL-WORLD DATASETS ARE USUALLY MUCH MORE COMPLICATED!

Using a Bayesian framework:

- ▶ Build (hierarchical) models to account for potentially complicated dependency structures in the data.
- ▶ Attribute uncertainty to model parameters and latent variables using priors.

Two main challenges:

1. Need computationally efficient methods to calculate posteriors.
2. Select priors in a sensible way.

REAL-WORLD DATASETS ARE USUALLY MUCH MORE COMPLICATED!

Using a Bayesian framework:

- ▶ Build (hierarchical) models to account for potentially complicated dependency structures in the data.
- ▶ Attribute uncertainty to model parameters and latent variables using priors.

Two main challenges:

1. Need computationally efficient methods to calculate posteriors.
2. Select priors in a sensible way.

MCMC: MARKOV CHAIN MONTE CARLO METHODS

Based on **sampling**. Construct Markov chains with the target posterior as stationary distribution.

- ▶ Extensively used within Bayesian inference since the 1980's.
- ▶ Flexible and general, sometimes the only thing we can do!
- ▶ Available for specific models using e.g. BUGS, JAGS, BayesX.
- ▶ In general, not straightforward to implement. Slow, convergence issues, etc.

INLA: INTEGRATED NESTED LAPLACE APPROXIMATIONS

Introduced by Rue, Martino and Chopin (2009). Posteriors are estimated using numerical approximations. **No sampling** needed!

- ▶ Unified framework for analysing a general class of statistical models, named latent Gaussian models.
- ▶ Accurate and computationally superior to MCMC methods!
- ▶ Easily accessible using the R-interface R-INLA, see www.r-inla.org.

Reference:

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

WHAT IS A LATENT GAUSSIAN MODEL?

Classical multiple linear regression model

The mean μ of an n -dimensional observational vector \mathbf{y} is given by

$$\mu_i = \mathbb{E}(Y_i) = \alpha + \sum_{j=1}^{n_\beta} \beta_j z_{ji}, \quad i = 1, \dots, n$$

where

α : Intercept

β : Linear effects of covariates \mathbf{z}

ACCOUNT FOR NON-GAUSSIAN OBSERVATIONS

Generalized linear model (GLM)

The mean μ is linked to the linear predictor η_i :

$$\eta_i = g(\mu_i) = \alpha + \sum_{j=1}^{n_\beta} \beta_j z_{ji}, \quad i = 1, \dots, n$$

where $g(\cdot)$ is a link function and

α : Intercept

β : Linear effects of covariates \mathbf{z}

ACCOUNT FOR NON-LINEAR EFFECTS OF COVARIATES

Generalized additive model (GAM)

The mean μ is linked to the linear predictor η_i :

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

where $g(\cdot)$ is a link function and

- α : Intercept
- $\{f_k(\cdot)\}$: Non-linear smooth effects of covariates c_k

STRUCTURED ADDITIVE REGRESSION MODELS

GLM/GAM/GLMM/GAMM+++

The mean μ is linked to the linear predictor η_i :

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

where $g(\cdot)$ is a link function and

- α : Intercept
- β : Linear effects of covariates z
- $\{f_k(\cdot)\}$: Non-linear smooth effects of covariates c_k
- ϵ : Iid random effects

LATENT GAUSSIAN MODELS

- ▶ Collect all parameters (random variables) in the linear predictor in a **latent field**

$$\mathbf{x} = \{\alpha, \beta, \{f_k(\cdot)\}, \eta\}.$$

- ▶ A latent Gaussian model is obtained by assigning Gaussian priors to all elements of \mathbf{x} .
- ▶ Very flexible due to many different forms of the unknown functions $\{f_k(\cdot)\}$:
 - ▶ Include temporally and/or spatially indexed covariates.
- ▶ **Hyperparameters** account for variability and length/strength of dependence.

LATENT GAUSSIAN MODELS

- ▶ Collect all parameters (random variables) in the linear predictor in a **latent field**

$$\mathbf{x} = \{\alpha, \beta, \{f_k(\cdot)\}, \eta\}.$$

- ▶ A latent Gaussian model is obtained by assigning Gaussian priors to all elements of \mathbf{x} .
- ▶ Very flexible due to many different forms of the unknown functions $\{f_k(\cdot)\}$:
 - ▶ Include temporally and/or spatially indexed covariates.
- ▶ **Hyperparameters** account for variability and length/strength of dependence.

LATENT GAUSSIAN MODELS

- ▶ Collect all parameters (random variables) in the linear predictor in a **latent field**

$$\mathbf{x} = \{\alpha, \beta, \{f_k(\cdot)\}, \eta\}.$$

- ▶ A latent Gaussian model is obtained by assigning Gaussian priors to all elements of \mathbf{x} .
- ▶ Very flexible due to many different forms of the unknown functions $\{f_k(\cdot)\}$:
 - ▶ Include temporally and/or spatially indexed covariates.
- ▶ **Hyperparameters** account for variability and length/strength of dependence.

SOME EXAMPLES OF LATENT GAUSSIAN MODELS

- ▶ Generalized linear and additive (mixed) models
- ▶ Semiparametric regression
- ▶ Disease mapping
- ▶ Survival analysis
- ▶ Log-Gaussian Cox-processes
- ▶ Geostatistical models
- ▶ Spatial and spatio-temporal models
- ▶ Stochastic volatility
- ▶ Dynamic linear models
- ▶ State-space models
- ▶ +++

UNIFIED FRAMEWORK: A THREE-STAGE HIERARCHICAL MODEL

1. Observations: y

2. Latent field: x

3. Hyperparameters: θ

UNIFIED FRAMEWORK: A THREE-STAGE HIERARCHICAL MODEL

1. Observations: y

Assumed **conditionally independent** given x and θ_1 :

2. Latent field: x

Assumed to be a **GMRF** with a sparse precision matrix $Q(\theta_2)$:

3. Hyperparameters: $\theta = (\theta_1, \theta_2)$

Precision parameters of the Gaussian priors:

UNIFIED FRAMEWORK: A THREE-STAGE HIERARCHICAL MODEL

1. Observations: \mathbf{y}

Assumed **conditionally independent** given \mathbf{x} and $\boldsymbol{\theta}_1$:

$$\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}_1 \sim \prod_{i=1}^n \pi(y_i \mid x_i, \boldsymbol{\theta}_1).$$

2. Latent field: \mathbf{x}

Assumed to be a **GMRF** with a sparse precision matrix $\mathbf{Q}(\boldsymbol{\theta}_2)$:

$$\mathbf{x} \mid \boldsymbol{\theta}_2 \sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{\theta}_2), \mathbf{Q}^{-1}(\boldsymbol{\theta}_2)).$$

3. Hyperparameters: $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$

Precision parameters of the Gaussian priors:

$$\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta}).$$

MODEL SUMMARY

The joint posterior for the latent field and hyperparameters:

$$\begin{aligned}\pi(\mathbf{x}, \boldsymbol{\theta} \mid \mathbf{y}) &\propto \pi(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})\pi(\mathbf{x}, \boldsymbol{\theta}) \\ &\propto \prod_{i=1}^n \pi(y_i \mid x_i, \boldsymbol{\theta})\pi(\mathbf{x} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta})\end{aligned}$$

Remarks:

- ▶ $m = \dim(\boldsymbol{\theta})$ is often quite small, like $m \leq 6$.
- ▶ $n = \dim(\mathbf{x})$ is often large, typically $n = 10^2 - 10^6$.

TARGET DENSITIES ARE GIVEN AS HIGH-DIMENSIONAL INTEGRALS

We want to estimate:

- ▶ The marginals of all components of the latent field:

- ▶ The marginals of all the hyperparameters:

TARGET DENSITIES ARE GIVEN AS HIGH-DIMENSIONAL INTEGRALS

We want to estimate:

- ▶ The marginals of all components of the latent field:

$$\begin{aligned}\pi(x_i | \mathbf{y}) &= \int \int \pi(\mathbf{x}, \boldsymbol{\theta} | \mathbf{y}) dx_{-i} d\boldsymbol{\theta} \\ &= \int \pi(x_i | \boldsymbol{\theta}, \mathbf{y}) \pi(\boldsymbol{\theta} | \mathbf{y}) d\boldsymbol{\theta}, \quad i = 1, \dots, n.\end{aligned}$$

- ▶ The marginals of all the hyperparameters:

$$\begin{aligned}\pi(\theta_j | \mathbf{y}) &= \int \int \pi(\mathbf{x}, \boldsymbol{\theta} | \mathbf{y}) dx d\boldsymbol{\theta}_{-j} \\ &= \int \pi(\boldsymbol{\theta} | \mathbf{y}) d\boldsymbol{\theta}_{-j}, \quad j = 1, \dots, m.\end{aligned}$$

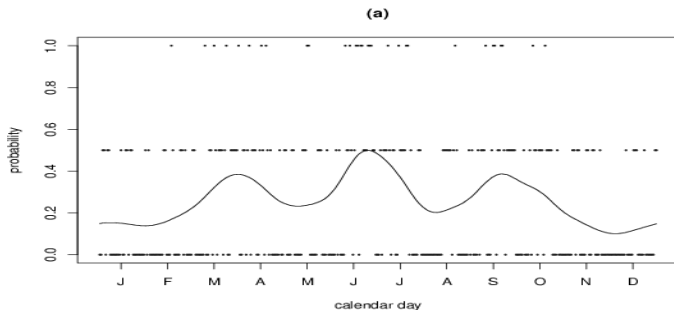
AN MCMC CASE-STUDY

- ▶ Study a seemingly trivial hierarchical model
 - ▶ Latent temporal Gaussian, with
 - ▶ Binary observations
- ▶ Develop a “standard” MCMC-algorithm for inference
 - ▶ Auxiliary variables
 - ▶ (Conjugate) single-site updates
- ▶ ..and study empirically its properties.

AUXILIARY AIMS

- ▶ Give a “historical” development of the ideas in INLA
- ▶ Show how to make good proposal distributions for latent Gaussian models
- ▶ Remind you not to make bad Gibbs samplers

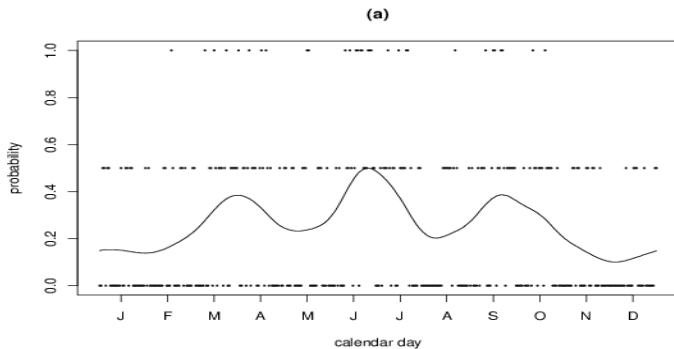
TOKYO RAINFALL DATA



Stage 1 Binomial data

$$y_i \sim \begin{cases} \text{Binomial}(2, p(x_i)) \\ \text{Binomial}(1, p(x_i)) \end{cases}$$

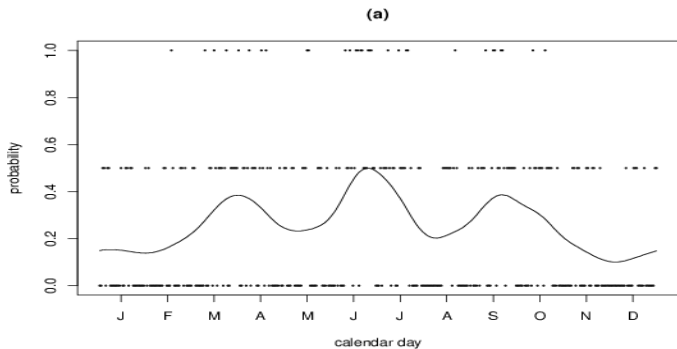
TOKYO RAINFALL DATA



Stage 2 Assume a smooth latent x ,

$$x \sim RW2(\kappa), \quad \text{logit}(p_i) = x_i$$

TOKYO RAINFALL DATA



Stage 3 $\text{Gamma}(\alpha, \beta)$ -prior on κ

MODEL SUMMARY

$$\pi(\mathbf{x} \mid \kappa) \pi(\kappa) \prod_i \pi(y_i \mid x_i)$$

where

- ▶ $\mathbf{x} \mid \kappa$ is Gaussian (Markov) with dimension 366
- ▶ κ is Gamma
- ▶ $y_i \mid x_i$ is Binomial with $p(x_i)$

CONSTRUCTION OF NICE FULL CONDITIONALS

A popular approach is to introduce auxiliary variables w , so that

$$x \mid \text{the rest}$$

is Gaussian.

EXAMPLE: BINARY REGRESSION

GMRF x and Bernoulli data

$$\begin{aligned}y_i &\sim \mathcal{B}(g^{-1}(x_i)) \\g(p) &= \Phi(p) \quad \text{probit link}\end{aligned}$$

Equivalent representation using auxiliary variables w

$$\begin{aligned}\epsilon_i &\stackrel{\text{iid}}{\sim} \mathcal{N}(0, 1) \\w_i &= x_i + \epsilon_i \\y_i &= \begin{cases} 1 & \text{if } w_i > 0 \\ 0 & \text{otherwise.} \end{cases}\end{aligned}$$

for the probit-link.

SINGLE-SITE GIBBS SAMPLING

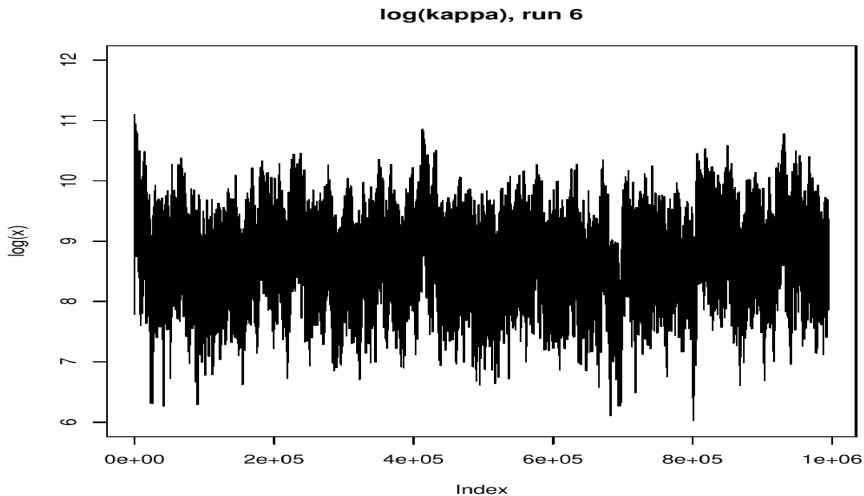
Auxiliary variables can be introduced for the logit-link¹, to achieve this sampler:

- ▶ $\kappa \sim \Gamma(\cdot, \cdot)$
- ▶ for each i
 - ▶ $x_i \sim \mathcal{N}(\cdot, \cdot)$
- ▶ for each i
 - ▶ $w_i \sim \mathcal{W}(\cdot)$

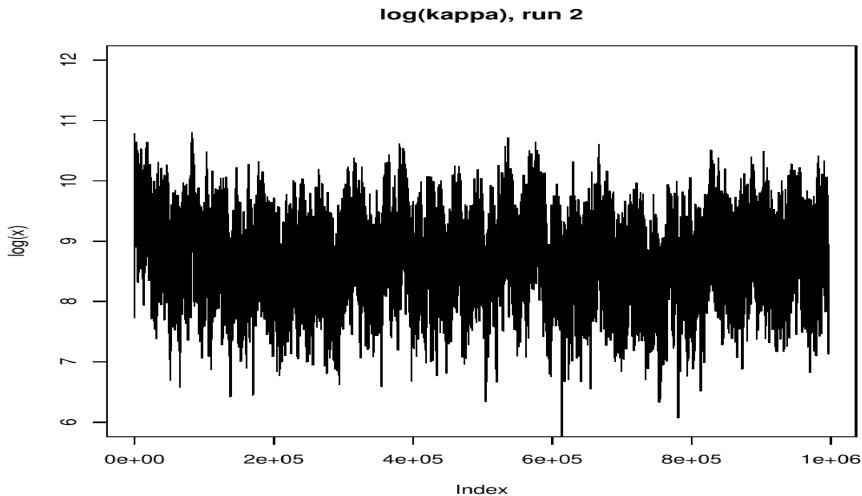
It is fully automatic; no tuning!!!

¹Held & Holmes (2006)

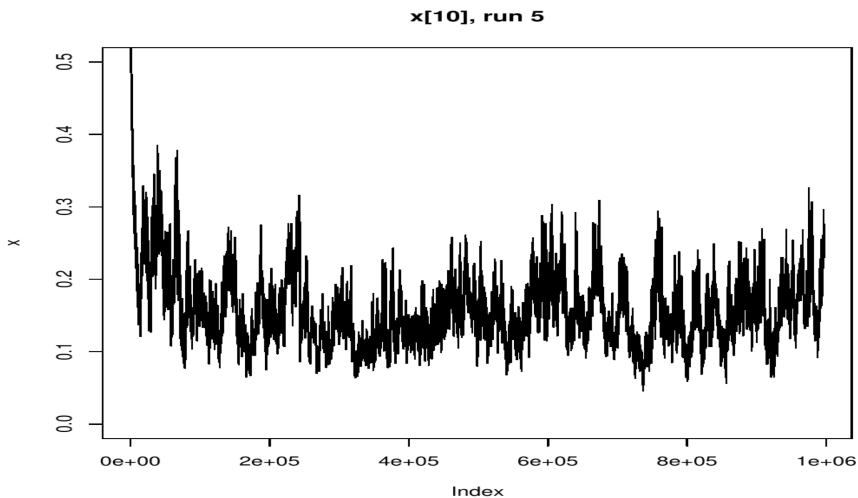
RESULTS: HYPER-PARAMETER $\log(\kappa)$



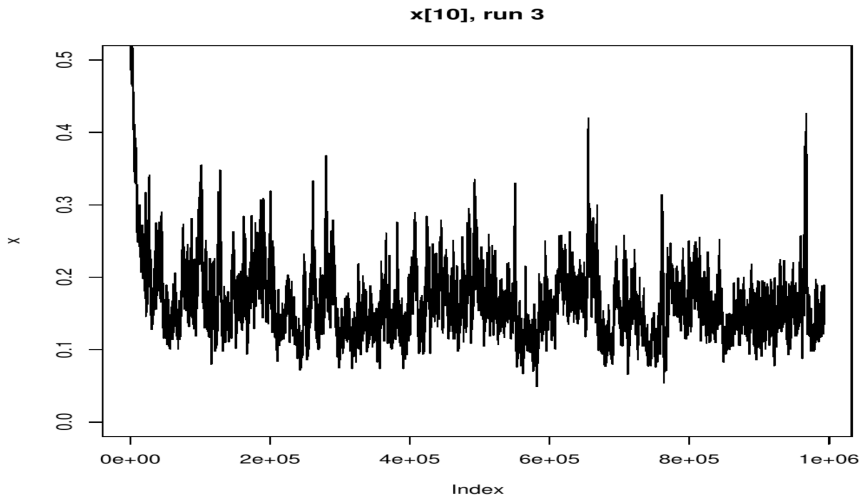
RESULTS: HYPER-PARAMETER $\log(\kappa)$



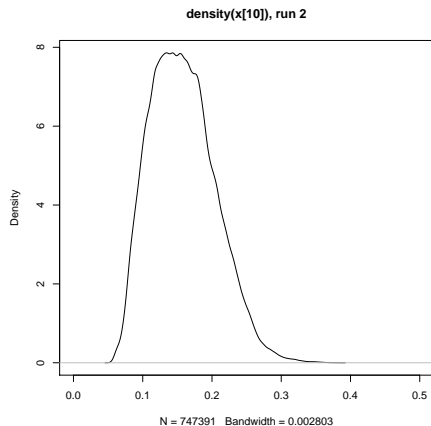
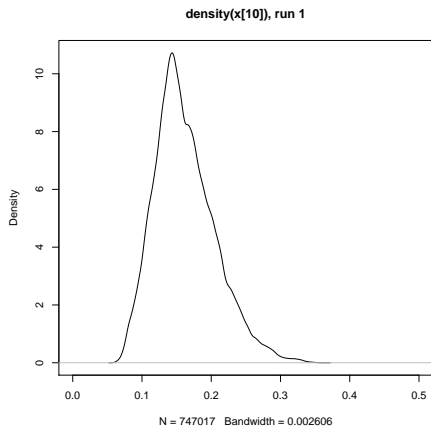
RESULTS: LATENT NODE x_{10}



RESULTS: LATENT NODE x_{10}



RESULTS: DENSITY FOR LATENT NODE x_{10}



DISCUSSION

Single-site sampler with auxiliary variables:

- ▶ Even *long runs* shows large variation
- ▶ “Long” range dependence
- ▶ *Very* slowly mixing

But:

- ▶ Easy to be “fooled” running shorter chains
- ▶ The variability can be underestimated.

WHAT IS CAUSING THE PROBLEM?

Two issues

1. Slow mixing within the latent field x
2. Slow mixing between the latent field x and θ .

Blocking is the “usual” approach to resolve such issues, if possible.

Note: blocking mainly helps within the block only.

STRATEGIES FOR BLOCKING

Slow mixing due to the latent field \mathbf{x} only:

- ▶ Block \mathbf{x}

Slow mixing due to the interaction between the latent field \mathbf{x} and $\boldsymbol{\theta}$:

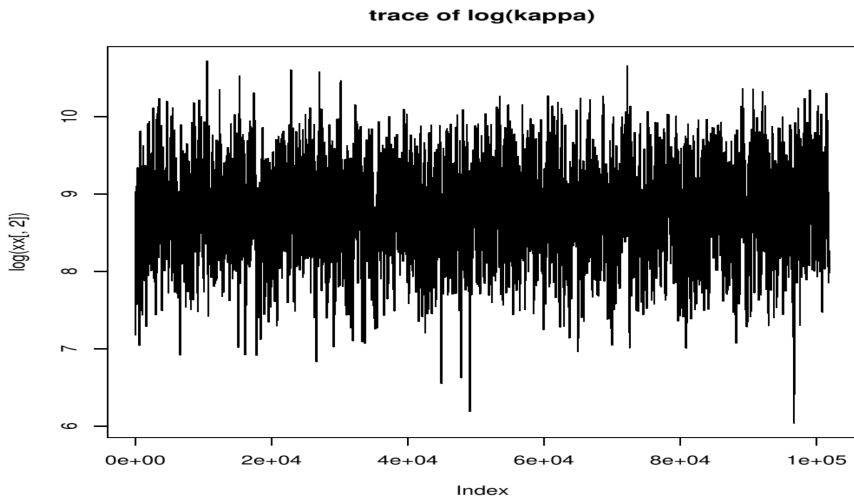
- ▶ Block $(\mathbf{x}, \boldsymbol{\theta})$.

In most cases: if you can do one, you can do both.

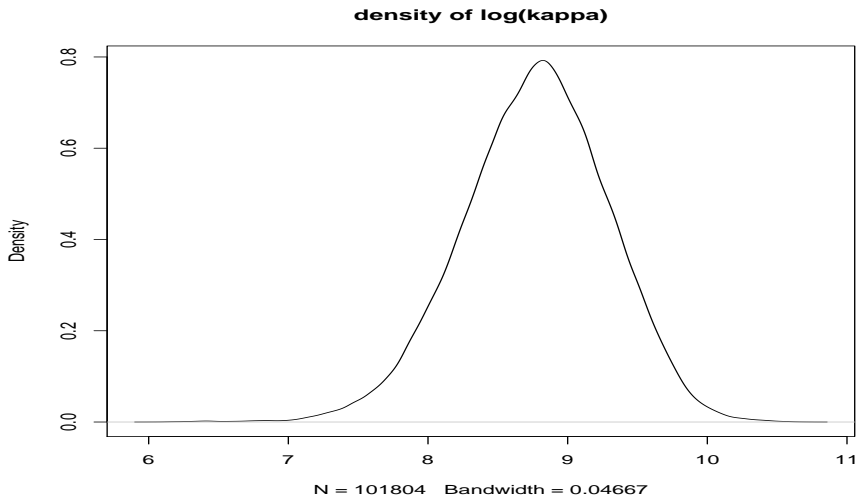
BLOCKING SCHEME I

- ▶ $\kappa \sim \Gamma(\cdot, \cdot)$
- ▶ $\mathbf{x} \sim \mathcal{N}(\cdot, \cdot)$
- ▶ $\mathbf{w} \sim \mathcal{W}(\cdot)$ (conditional independent)

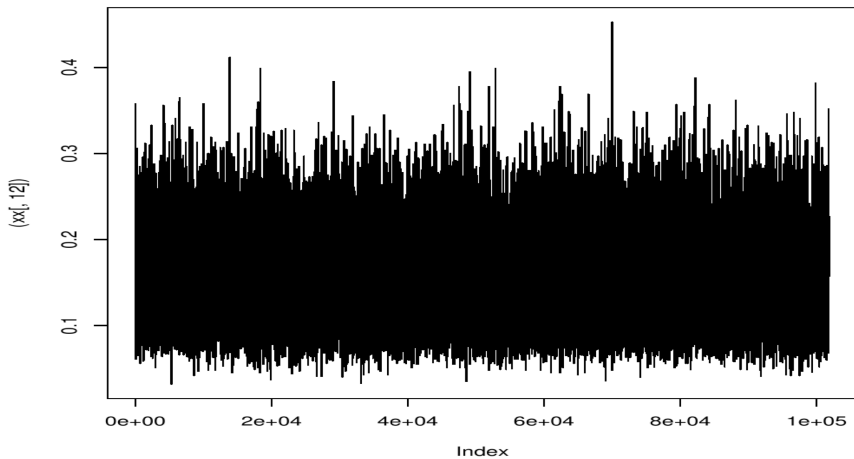
RESULTS



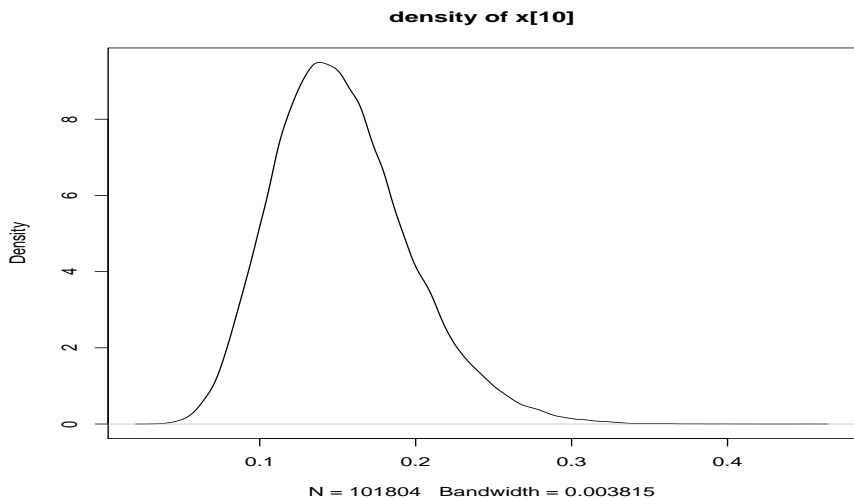
RESULTS



RESULTS



RESULTS



BLOCKING SCHEME II

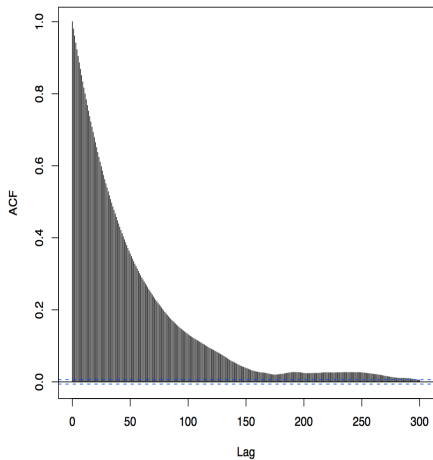
- ▶ Sample
 - ▶ $\kappa' \sim q(\kappa'; \kappa)$
 - ▶ $\mathbf{x}' | \kappa', \mathbf{y} \sim \mathcal{N}(\cdot, \cdot)$
 and then accept/reject (\mathbf{x}', κ') jointly
- ▶ $\mathbf{w} \sim \mathcal{W}(\cdot)$ (conditional independent)

Remarks

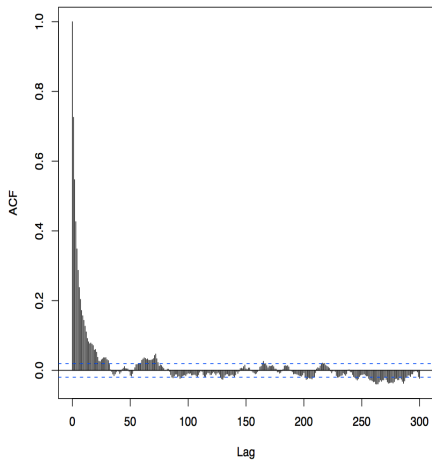
- ▶ If the normalising constant for $\mathbf{x} | \cdot$ is available, then this is an EASY FIX of scheme I.
- ▶ Usually makes a huge improvement
- ▶ Automatic “reparameterisation”
- ▶ Doubles the computational costs

RESULTS

ACF(log(kappa) scheme I)



ACF(log(kappa) scheme II)



REMOVING THE AUXILIARY VARIABLES

- ▶ The auxiliary variables makes the full conditional for \mathbf{x} Gaussian
- ▶ If we do not use them, the full conditional for \mathbf{x} looks like

$$\begin{aligned}\pi(\mathbf{x} \mid \dots) &\propto \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \sum_i \log(\pi(y_i|x_i))\right) \\ &\approx \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T(\mathbf{Q} + \text{diag}(\mathbf{c}))(\mathbf{x} - \boldsymbol{\mu})\right) \\ &= \pi_G(\mathbf{x} \mid \dots)\end{aligned}$$

- ▶ The Gaussian approximation is constructed by matching the
 - ▶ mode, and the
 - ▶ curvature at the mode.

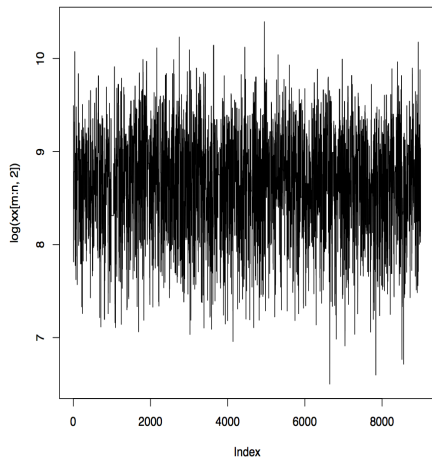
IMPROVED ONE-BLOCK SCHEME

- ▶ $\kappa' \sim q(\cdot; \kappa)$
- ▶ $\mathbf{x}' \sim \pi_G(\mathbf{x} \mid \kappa', \mathbf{y})$
- ▶ Accept/reject (\mathbf{x}', κ') jointly

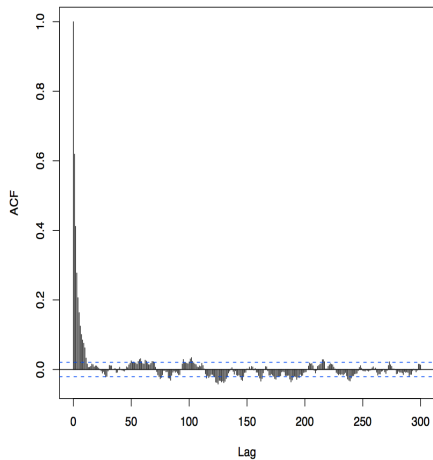
Note: $\pi_G(\cdot)$ is indexed by κ' , hence we need to compute one for each value of κ' .

RESULTS

Trace of $\log(\kappa)$



ACF($\log(\kappa)$)



INDEPENDENCE SAMPLER

We can construct an independence sampler, using $\pi_G(\cdot)$.
The Laplace-approximation for $\kappa|\mathbf{x}$:

$$\begin{aligned}\pi(\kappa | \mathbf{y}) &\propto \frac{\pi(\kappa) \pi(\mathbf{x}|\kappa) \pi(\mathbf{y}|\mathbf{x})}{\pi(\mathbf{x}|\kappa, \mathbf{y})} \\ &\approx \frac{\pi(\kappa) \pi(\mathbf{x}|\kappa) \pi(\mathbf{y}|\mathbf{x})}{\pi_G(\mathbf{x}|\kappa, \mathbf{y})} \Bigg|_{\mathbf{x}=\text{mode}(\kappa)}\end{aligned}$$

Hence, we do first

- ▶ Evaluate the Laplace-approximation at some “selected” points
- ▶ Build an interpolation log-spline
- ▶ Use this parametric model as $\tilde{\pi}(\kappa|\mathbf{y})$

INDEPENDENCE SAMPLER

- ▶ $\kappa' \sim \tilde{\pi}(\kappa|\mathbf{y})$
- ▶ $\mathbf{x}' \sim \pi_G(\mathbf{x}|\kappa', \mathbf{y})$
- ▶ Accept/reject (κ', \mathbf{x}') jointly

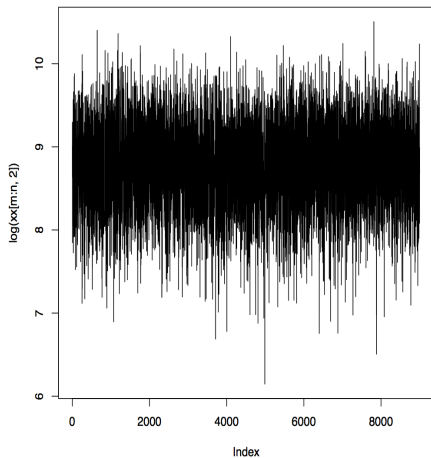
Note:

$$\text{Corr}(x(t+k), x(t)) \approx (1 - \alpha)^{|k|}$$

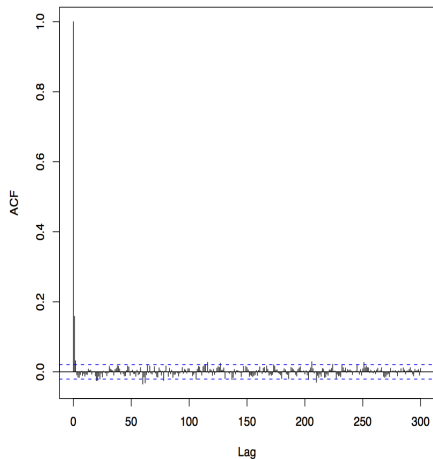
In this example, $\alpha = 0.83\dots$

RESULTS

Trace $\log(\kappa)$; independence sampler



ACF($\log(\kappa)$); independence sampler



CAN WE IMPROVE THIS SAMPLER?

- ▶ Yes, if we are interested in the posterior marginals for κ and $\{x_i\}$.
- ▶ The marginals for the Gaussian proposal $\pi_G(\mathbf{x}|\dots)$, are known analytically.
- ▶ Just use numerical integration!

DETERMINISTIC INFERENCE

Posterior marginal for κ :

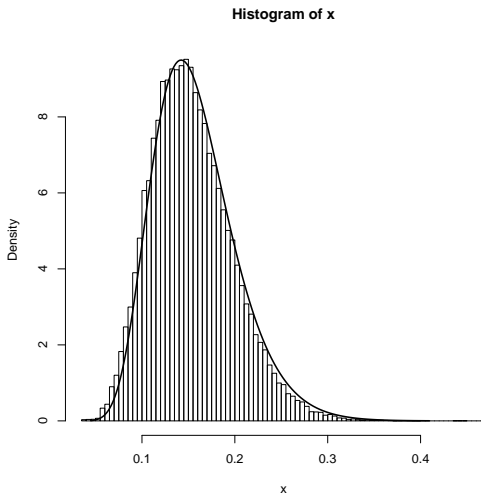
- ▶ Compute $\tilde{\pi}(\kappa|\mathbf{y})$

Posterior marginal for x_i :

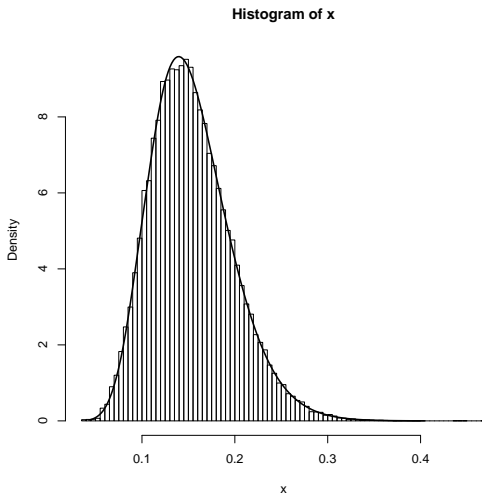
- ▶ Use numerical integration

$$\begin{aligned}\pi(x_i | \mathbf{y}) &= \int \pi(x_i | \mathbf{y}, \kappa) \pi(\kappa | \mathbf{y}) d\kappa \\ &\approx \sum_k \mathcal{N}(x_i; \mu_{\kappa_k}, \sigma^2(\kappa_k)) \times \tilde{\pi}(\kappa_k | \mathbf{y}) \times \Delta_k\end{aligned}$$

RESULTS: MIXTURE OF GAUSSIANS



RESULTS: IMPROVED....



WHAT CAN BE LEARNED FROM THIS EXERCISE?

For a relative simple model, we have implemented

- ▶ single-site with auxiliary variables (loong time; hours)
- ▶ various forms for blocking (long time; many minutes)
- ▶ independence sampler (long time; many minutes)
- ▶ approximate inference (nearly instant; one second)

WHAT CAN BE LEARNED FROM THIS EXERCISE? ...

Single-site Gibbs samplers don't work for when there's correlation.

This is completely unsurprising!

But they still get used. Which implies

- ▶ Most probably, the results would be not correct.
- ▶ They “accept” the long running-time.
- ▶ Trouble: such MCMC-schemes is not useful for routine analysis of similar data.

WHAT CAN BE LEARNED FROM THIS EXERCISE? ...

- ▶ In many cases, the situation is much worse in practice; this was a very simple model.
- ▶ Single-site MCMC is still the default choice for the non-expert user.
- ▶ Hierarchical models are popular, but they are difficult for MCMC.

Perhaps the development of models is not in sync with the development of inference? We cannot just wait for more powerful computers...

THE INTEGRATED NESTED LAPLACE APPROXIMATION (INLA) I

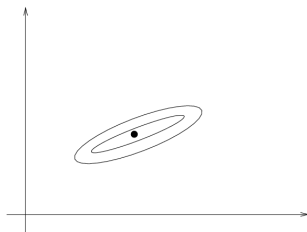
Step I Explore $\tilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$

- ▶ Locate the mode
- ▶ Use the Hessian to construct new variables
- ▶ Grid-search
- ▶ Can be case-specific

THE INTEGRATED NESTED LAPLACE APPROXIMATION (INLA) I

Step I Explore $\tilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$

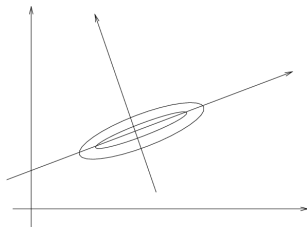
- ▶ Locate the mode
- ▶ Use the Hessian to construct new variables
- ▶ Grid-search
- ▶ Can be case-specific



THE INTEGRATED NESTED LAPLACE APPROXIMATION (INLA) I

Step I Explore $\tilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$

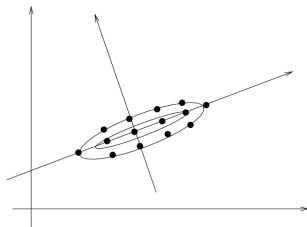
- ▶ Locate the mode
- ▶ Use the Hessian to construct new variables
- ▶ Grid-search
- ▶ Can be case-specific



THE INTEGRATED NESTED LAPLACE APPROXIMATION (INLA) I

Step I Explore $\tilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$

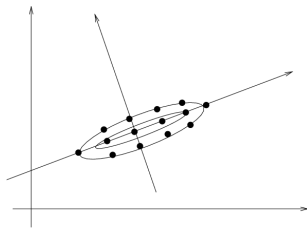
- ▶ Locate the mode
- ▶ Use the Hessian to construct new variables
- ▶ Grid-search
- ▶ Can be case-specific



THE INTEGRATED NESTED LAPLACE APPROXIMATION (INLA) I

Step I Explore $\tilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$

- ▶ Locate the mode
- ▶ Use the Hessian to construct new variables
- ▶ Grid-search
- ▶ Can be case-specific



THE INTEGRATED NESTED LAPLACE APPROXIMATION (INLA) II

Step II For each θ_j

- ▶ For each i , evaluate the Laplace approximation for selected values of x_i
- ▶ Build a Skew-Normal or log-spline corrected Gaussian

$$\mathcal{N}(x_i; \mu_i, \sigma_i^2) \times \exp(\text{spline})$$

to represent the conditional marginal density.

THE INTEGRATED NESTED LAPLACE APPROXIMATION (INLA) III

Step III Sum out θ_j

- ▶ For each i , sum out θ

$$\tilde{\pi}(x_i | \mathbf{y}) \propto \sum_j \tilde{\pi}(x_i | \mathbf{y}, \theta_j) \times \tilde{\pi}(\theta_j | \mathbf{y})$$

- ▶ Build a log-spline corrected Gaussian

$$\mathcal{N}(x_i; \mu_i, \sigma_i^2) \times \exp(\text{spline})$$

to represent $\tilde{\pi}(x_i | \mathbf{y})$.

COMPUTING POSTERIOR MARGINALS FOR θ_j (I)

Main idea

- ▶ Use the integration-points and build an interpolant
- ▶ Use numerical integration on that interpolant

COMPUTING POSTERIOR MARGINALS FOR θ_j (II)

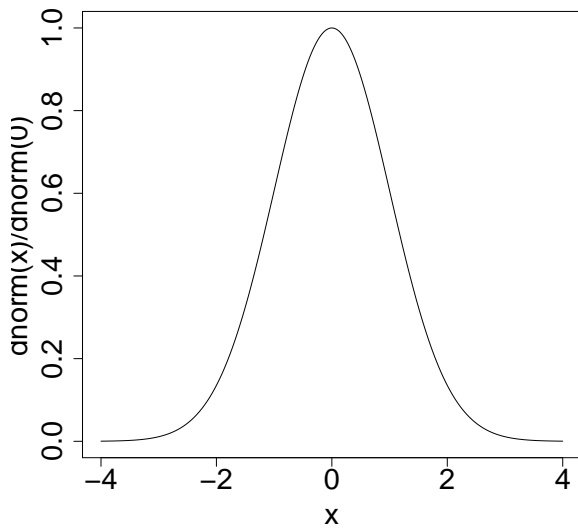
Practical approach (high accuracy)

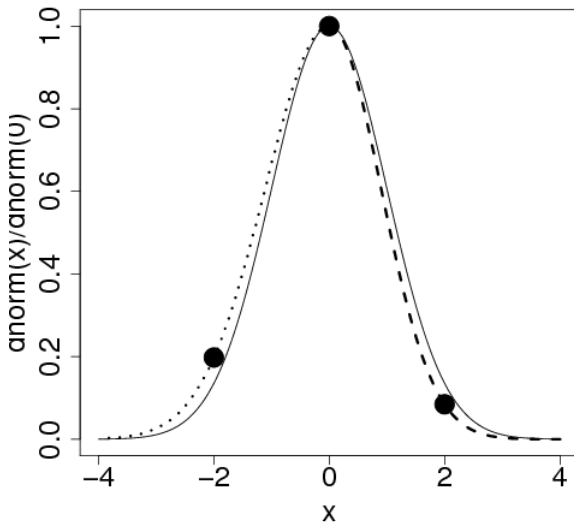
- ▶ Rerun using a fine integration grid
- ▶ Possibly with no rotation
- ▶ Just sum up at grid points, then interpolate

COMPUTING POSTERIOR MARGINALS FOR θ_j (II)

Practical approach (lower accuracy)

- ▶ Use the Gaussian approximation at the mode θ^*
- ▶ ...BUT, adjust the standard deviation in each direction
- ▶ Then use numerical integration





HOW CAN WE ASSESS THE ERROR IN THE APPROXIMATIONS?

Tool 1: Compare a sequence of improved approximations

1. Gaussian approximation
2. Simplified Laplace
3. Laplace

HOW CAN WE ASSESS THE ERROR IN THE APPROXIMATIONS?

Tool 3: Estimate the “effective” number of parameters as defined in the *Deviance Information Criteria*:

$$p_D(\boldsymbol{\theta}) = \bar{D}(\mathbf{x}; \boldsymbol{\theta}) - D(\bar{\mathbf{x}}; \boldsymbol{\theta})$$

and compare this with the number of observations.

Low ratio is good.

This criteria has theoretical justification.

IMPORTANT OBSERVATION

If $y|x, \theta$ is *Gaussian*, the
“approximation” is exact.

ANYONE CAN USE INLA!

The R-INLA project

- The home of the R-INLA project
- Contact us, stay updated, get help or report an error
- Discussion forum
- Download
- Examples and tutorials
 - Case Studies
 - Tutorials
 - Volume I
 - Volume II
- FAQ
- Getting started
- Help
- Internal use
- Models
 - Latent models
 - Likelihoods
 - Priors
 - Tools to manipulate models and likelihoods
- News
 - 10th Applied Statistics 2013 International conference, Slovenia
 - Bayes 2013: An introduction to INLA with a comparison to JAGS (hands on)
 - Bayesian Biostatistics short course: WinBUGS/SAS/INMUSC May 2012
 - Bayesian Computing with INLA & Spatial Modeling Using

Bayesian computing with INLA!

This site provides documentation to the [R-INLA package](#) which solves a large class of statistical models using the [INLA](#) approach.

[Here](#) is a short introduction describing the class of models which can be solved using R-INLA.

All [models](#) implemented in R-INLA are described in details, moreover a large series of worked out [examples](#) are provided and we hope that this will help the user to gain familiarity with the library. Recent changes in the code can be [viewed here](#).

Recent posts to the discussion group

Google Group



NYTT EMNE Hjelp

R-inla discussion group Delt offentlig

30 av 376 emner Om


Welcome to this discussion group about r-inla. Please ask your questions here in case you think they will be useful for others, otherwise send them to help@r-inla.org. You are of course free to comment on questions from others as well.


Best,
H


	Factor variable and random slope Av Eric Coker - 2 innlegg - 4 visninger	13. mars
	Hyperparameters for rw1 and SPOE models Av Ian Renner - 3 innlegg - 29 visninger	12. mars


Recent announcements

Recent Announcements

 **Open PhD-grants!** as attached. If you're interested, please contact us at hru@r-inla.org
Posted 3 Mar 2014 19:59 by Havard Rue

 **Spatial Modelling with INLA Workshop, 2-4 June, St Andrews, Scotland**
The link to the official page
Posted 23 Feb 2014 09:02 by Havard Rue

 **Short-course at the University of Girona, Spain April 24, 2014** Given by Gianluca Baio, details as attached.
Posted 29 Jan 2014 08:27 by Havard Rue

 **INLA-lectures in Florence, 28-29 Jan, 2014. The web-page is here.**
Posted 20 Jan 2014 10:03 by Havard Rue

GETTING STARTED WITH R-INLA

- ▶ Installation (NB: Not on CRAN)

```
> install.packages("INLA",  
  repos="http://www.math.ntnu.no/inla/R/testing")
```

- ▶ Load package and upgrade:

```
> library(INLA)  
> inla.upgrade(testing = TRUE)
```

- ▶ Help and examples at www.r-inla.org

BASIC STRUCTURE TO RUN A MODEL

- ▶ Define the **formula**, specifying non-linear functions using $f(\cdot)$, including the latent model and priors for hyperparameters:

```
> formula = y ~ 1 + z
           + f(c, model = "...",
             hyper = list(theta =
                           list(prior = "...", param = ...)))
```

- ▶ Call **inla(.)**, where you specify the relevant likelihood

```
> inla(formula, data=data.frame(...), family = "...")
```

IMPLEMENTED MODELS

Different likelihoods, latent models and (hyper)priors:

```
> names(inla.models())$likelihood)
```

```
> names(inla.models())$latent)
```

```
> names(inla.models())$prior)
```

Documentation (not complete):

```
> inla.doc("....")
```

EXAMPLE: LOGISTIC REGRESSION, 2×2 FACTORIAL DESIGN

Example (Seeds)

Consider the proportion of seeds that germinates on each of 21 plates. We have two seed types (x_1) and two root extracts (x_2).

```
> data(Seeds)
> head(Seeds)
   r  n x1 x2 plate
1 10 39  0  0     1
2 23 62  0  0     2
3 23 81  0  0     3
4 26 51  0  0     4
5 17 39  0  0     5
6  5  6  0  1     6
```

SUMMARY DATA SET

Number of seeds that germinated in each group:

		Seed types	
		$x_1 = 0$	$x_1 = 1$
Root extract	$x_2 = 0$	99/272	49/123
	$x_2 = 1$	201/295	75/141

STATISTICAL MODEL

- ▶ Assume that the number of seeds that germinate on plate i is binomial

$$r_i \sim \text{Binomial}(n_i, p_i), \quad i = 1, \dots, 21,$$

- ▶ Logistic regression model:

$$\text{logit}(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \alpha + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{1i} x_{2i} + \epsilon_i$$

where $\epsilon_i \sim N(0, \sigma_\epsilon^2)$ are iid.

Aim:

Estimate the main effects, β_1 and β_2 and a possible interaction effect β_3 .

STATISTICAL MODEL

- ▶ Assume that the number of seeds that germinate on plate i is binomial

$$r_i \sim \text{Binomial}(n_i, p_i), \quad i = 1, \dots, 21,$$

- ▶ Logistic regression model:

$$\text{logit}(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \alpha + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{1i} x_{2i} + \epsilon_i$$

where $\epsilon_i \sim N(0, \sigma_\epsilon^2)$ are iid.

Aim:

Estimate the main effects, β_1 and β_2 and a possible interaction effect β_3 .

USING R-INLA

```
> formula = r ~ x1 + x2 + x1*x2 + f(plate, model="iid")
> result = inla(formula, data = Seeds,
               family = "binomial",
               Ntrials = n,
               control.predictor =
                   list(compute = T, link=1),
               control.compute = list(dic = T))
```

Default priors

Default prior for fixed effects is

$$\beta \sim N(0, 1000).$$

Change using the `control.fixed` argument in the `inla`-call.

OUTPUT

```
> summary(result)
Call:
"inla(formula = formula, family = \"binomial\", data = Seeds, Ntrials = n)"

Time used:
  Pre-processing      Running inla  Post-processing      Total
      0.1354           0.0911           0.0347           0.2613

Fixed effects:
      mean      sd 0.025quant 0.5quant 0.975quant  kld
(Intercept) -0.5581 0.1261   -0.8076  -0.5573   -0.3130 0e+00
x1           0.1461 0.2233   -0.2933   0.1467    0.5823 0e+00
x2           1.3206 0.1776    0.9748   1.3197    1.6716 1e-04
x1:x2       -0.7793 0.3066   -1.3799  -0.7796   -0.1774 0e+00

Random effects:
Name      Model
plate    IID model

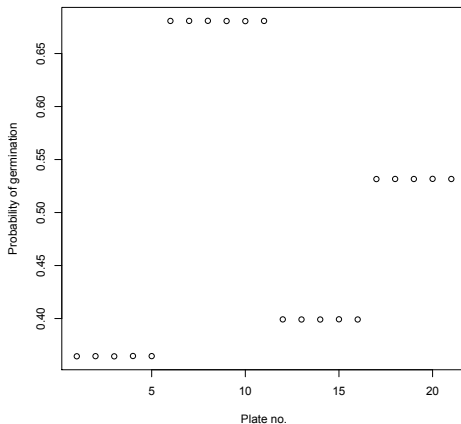
Model hyperparameters:
      mean      sd      0.025quant 0.5quant 0.975quant
Precision for plate 18413.03 18280.63 1217.90 13003.76 66486.29

Expected number of effective parameters(std dev): 4.014(0.0114)
Number of equivalent replicates : 5.231

Marginal Likelihood: -72.07
```

ESTIMATED GERMINATION PROBABILITIES

```
> result$summary.fitted.values$mean
```

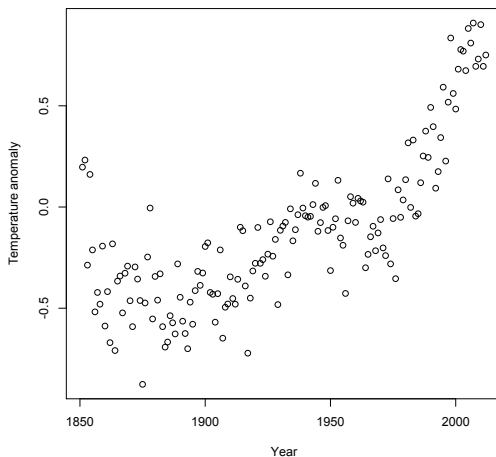


MORE IN THE PRACTICALS ...

```
> plot(result)
> result$summary.fixed
> result$summary.random
> result$summay.linear.predictor
> result$summay.fitted.values
> result$marginals.fixed
> result$marginals.hyperpar
> result$marginals.linear.predictor
> result$marginals.fitted.values
```

EXAMPLE: SEMIPARAMETRIC REGRESSION

Example (Annual global temperature anomalies)



ESTIMATING A SMOOTH NON-LINEAR TREND

- ▶ Assume the model

$$y_i = \alpha + f(x_i) + \epsilon_i, \quad i = 1, \dots, n,$$

where the errors are iid, $\epsilon_i \sim N(0, \sigma_\epsilon^2)$.

- ▶ Want to estimate the true underlying curve $f(\cdot)$.

R-CODE

- ▶ Define formula and run model

```
> formula = y ~ f(x, model = "rw2", hyper = ...)
```

```
> result = inla(formula, data = data.frame(y, x))
```

- ▶ The **default prior** for the hyperparameter of `rw2`:

```
hyper = list(prec =  
             list(prior = "loggamma",  
                 param = c(1, 0.00005)))
```

OUTPUT

- ▶ `> summary(result)`
`> plot(result)`

- ▶ **The mean effect of x :**

 - `> result$summary.random$x$mean`

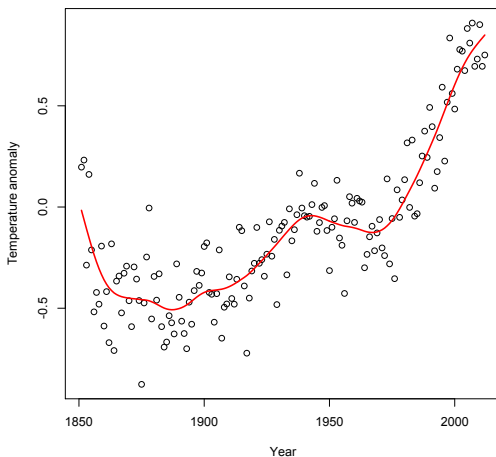
 - Note that this effect is constrained to sum to 0.

- ▶ **Resulting fitted curve**

 - `> result$summary.fitted.values$mean`

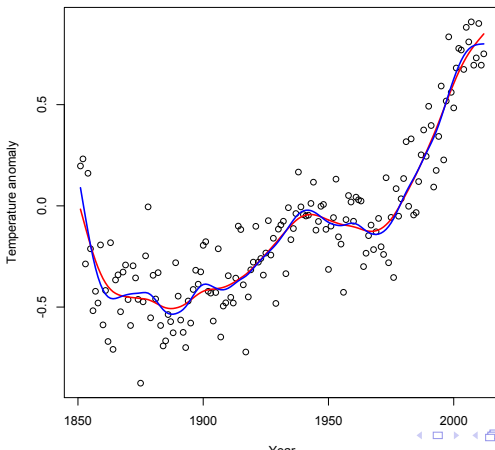
ESTIMATED FIT USING THE DEFAULT PRIOR

Example (Annual global temperature anomalies)



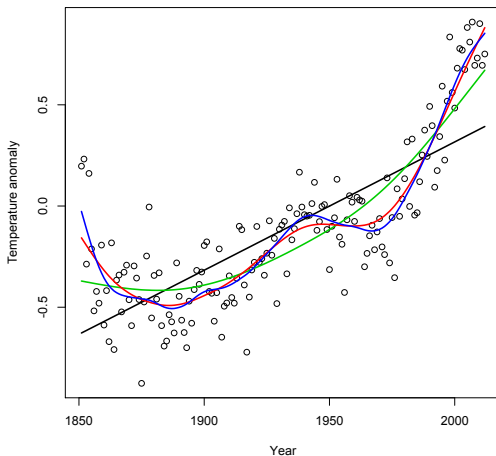
ESTIMATED FIT USING R-INLA COMPARED WITH SMOOTH.SPLINE

Example (Annual global temperature anomalies)



USING DIFFERENT PRIORS FOR THE PRECISION

Example (Annual global temperature anomalies)



DEFAULT PRIOR CHOICES IN R-INLA ARE NOT UNIVERSALLY GOOD

- ▶ Priors are an important part of Bayesian analysis
- ▶ There is no universally good way to specify priors for complex models
- ▶ The default priors in INLA are slightly sensible, but they should be used with caution
- ▶ We are in the process of incorporating a system of **Penalised Complexity priors** that perform better under complex hierarchical models.

Reference:

Daniel Simpson, Thiago Martins, Andrea Riebler, Geir-Arne Fuglstad, Håvard Rue, and Sigrunn Sørbye. *Penalising model complexity: A principled practical approach to constructing priors*. ArXiv:1403.4630.

SUMMARY

- ▶ INLA is used to analyse a broad class of statistical models, named latent Gaussian models.
- ▶ Unified computational framework with three levels:
 - Likelihood for the observations.
 - Latent field, model dependency structures.
 - Hyperparameters, tune smoothness.
- ▶ Efficient and accurate. Easily available using R-INLA.

Introduction to spatial statistics

STATISTICS IN SPACE!

Spatial data comes in essentially two different forms

- ▶ Point-referenced data
 - ▶ GPS tracking
 - ▶ Fixed measuring devices
 - ▶ “High resolution” satellites
- ▶ Region-based data
 - ▶ Census data
 - ▶ Plot data
 - ▶ Region-based counts
 - ▶ Historical data

LET'S THINK ABOUT DATA-GATHERING

A reasonably common way of getting spatial data is

- ▶ Break the area of interest up into smaller regions
- ▶ Get a team to survey the region
 - ▶ Completely
 - ▶ Partially
- ▶ NB: We are looking to build a joint statistical model for the process on the entire regions. Hence un-surveyed regions are not strictly “missing”, but rather part of the experimental design to be imputed based on the model.

How do we model this statistically?

NW ENGLAND



Fig 1. Leukaemia survival data: districts of Northwest England and locations of the observations.

HOW DO WE MODEL THIS?

Imagine we have animal counts in each region. We can model them as Poisson

$$y_i = Po(e^{\eta_i}).$$

How do we model the *linear predictor* η_i ?

- ▶ We could model the number of animals in each region independently
 - ▶ $\eta_i \sim N(\text{intercept} + (\text{covariates})_i, \sigma_i^2)$
 - ▶ Regional differences accounted through “random effect”
 - ▶ But... what if the distribution is inhomogeneous?
 - ▶ If there's an area where the animal is rare, we'll get lots of zero counts

HOW DO WE MODEL THIS?

Imagine we have animal counts in each region. We can model them as Poisson

$$y_i = Po(e^{\eta_i}).$$

How do we model the *linear predictor* η_i ?

- ▶ We could model the number of animals in each region independently
 - ▶ $\eta_i \sim N(\text{intercept} + (\text{covariates})_i, \sigma_i^2)$
 - ▶ Regional differences accounted through “random effect”
 - ▶ But... what if the distribution is inhomogeneous?
 - ▶ If there's an area where the animal is rare, we'll get lots of zero counts

HOW DO WE MODEL THIS?

Imagine we have animal counts in each region. We can model them as Poisson

$$y_i = Po(e^{\eta_i}).$$

How do we model the *linear predictor* η_i ?

- ▶ We could model the number of animals in each region independently
 - ▶ $\eta_i \sim N(\text{intercept} + (\text{covariates})_i, \sigma_i^2)$
 - ▶ Regional differences accounted through “random effect”
 - ▶ But... what if the distribution is inhomogeneous?
 - ▶ If there's an area where the animal is rare, we'll get lots of zero counts

HOW DO WE MODEL THIS?

Imagine we have animal counts in each region. We can model them as Poisson

$$y_i = Po(e^{\eta_i}).$$

How do we model the *linear predictor* η_i ?

- ▶ We could model the number of animals in each region independently
 - ▶ $\eta_i \sim N(\text{intercept} + (\text{covariates})_i, \sigma_i^2)$
 - ▶ Regional differences accounted through “random effect”
 - ▶ But... what if the distribution is inhomogeneous?
 - ▶ If there's an area where the animal is rare, we'll get lots of zero counts

HOW DO WE MODEL THIS?

Imagine we have animal counts in each region. We can model them as Poisson

$$y_i = Po(e^{\eta_i}).$$

How do we model the *linear predictor* η_i ?

- ▶ We could model the number of animals in each region independently
 - ▶ $\eta_i \sim N(\text{intercept} + (\text{covariates})_i, \sigma_i^2)$
 - ▶ Regional differences accounted through “random effect”
 - ▶ But... what if the distribution is inhomogeneous?
 - ▶ If there's an area where the animal is rare, we'll get lots of zero counts

HOW DO WE MODEL THIS?

Imagine we have animal counts in each region. We can model them as Poisson

$$y_i = Po(e^{\eta_i}).$$

How do we model the *linear predictor* η_i ?

- ▶ We could model some dependence across regions
 - ▶ “Nearby regions” should have similar counts
 - ▶ $\eta_i = \text{intercept} + (\text{covariates})_i + u_i$
 - ▶ Now the random effect $u_i \sim N(0, Q^{-1})$ is *correlated*
 - ▶ How should we do this?

HOW DO WE MODEL THIS?

Imagine we have animal counts in each region. We can model them as Poisson

$$y_i = Po(e^{\eta_i}).$$

How do we model the *linear predictor* η_i ?

- ▶ We could model some dependence across regions
 - ▶ “Nearby regions” should have similar counts
 - ▶ $\eta_i = \text{intercept} + (\text{covariates})_i + u_i$
 - ▶ Now the random effect $u_i \sim N(0, Q^{-1})$ is *correlated*
 - ▶ How should we do this?

HOW DO WE MODEL THIS?

Imagine we have animal counts in each region. We can model them as Poisson

$$y_i = Po(e^{\eta_i}).$$

How do we model the *linear predictor* η_i ?

- ▶ We could model some dependence across regions
 - ▶ “Nearby regions” should have similar counts
 - ▶ $\eta_i = \text{intercept} + (\text{covariates})_i + u_i$
 - ▶ Now the random effect $u_i \sim N(0, Q^{-1})$ is *correlated*
 - ▶ How should we do this?

HOW DO WE MODEL THIS?

Imagine we have animal counts in each region. We can model them as Poisson

$$y_i = Po(e^{\eta_i}).$$

How do we model the *linear predictor* η_i ?

- ▶ We could model some dependence across regions
 - ▶ “Nearby regions” should have similar counts
 - ▶ $\eta_i = \text{intercept} + (\text{covariates})_i + u_i$
 - ▶ Now the random effect $u_i \sim N(0, Q^{-1})$ is *correlated*
 - ▶ How should we do this?

HOW DO WE MODEL THIS?

Imagine we have animal counts in each region. We can model them as Poisson

$$y_i = Po(e^{\eta_i}).$$

How do we model the *linear predictor* η_i ?

- ▶ We could model some dependence across regions
 - ▶ “Nearby regions” should have similar counts
 - ▶ $\eta_i = \text{intercept} + (\text{covariates})_i + u_i$
 - ▶ Now the random effect $u_i \sim N(0, Q^{-1})$ is *correlated*
 - ▶ How should we do this?

MODELLING SPATIAL SIMILARITY

The easiest model of spatial similarity is the *Besag* model, which says that

$$x_i - x_j \sim N(0, \sigma^2)$$

if i and j are “neighbours”.

- ▶ This really does say nearby things are similar
- ▶ It says that the value at neighbouring sites is most probably not more than 3σ apart
- ▶ We need to choose neighbours.

EVERYBODY NEEDS GOOD NEIGHBOURS

How do we choose which points should be neighbours?

- ▶ Physical nearest points are often a good place to start
- ▶ Physical neighbours are not necessarily the best
- ▶ This is *modelling*, so you should consider your process
- ▶ Consider, for instance, the problem of Tromsø...

A THEORY DIVERSION: THE MARKOV PROPERTY

Models based on neighbourhood have a name in statistics: they are *Markovian models*

- ▶ Markovian models are specified entirely through “neighbourhood structures”
- ▶ It is easier to than specifying a full covariance
- ▶ For a first example, let's consider time

EXAMPLE: AR(1) PROCESS

$$x_t \mid x_{t-1} = \phi x_{t-1} + \epsilon_t, \quad t > 1, \epsilon_t \sim \mathcal{N}(0, \tau^{-1})$$
$$x_1 \sim \mathcal{N}\left(0, \frac{1}{1 - \phi^2}\right)$$

- ▶ The values at t is proportional to the value at t plus some extra variability
- ▶ ϕ is the *lag-one autocorrelation*
- ▶ ϵ_t is the innovation noise
- ▶ τ is the precision of the innovation
- ▶ The distribution for x_1 ensures the process is stationary.

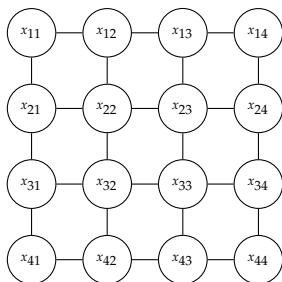
THE AR(1) PROCESS IN PICTURES

AR(1):



- ▶ The circles represent the values of x at individual time points
- ▶ There is a line between them if they are *conditionally dependent*

MARKOV IN SPACE!



- ▶ The model above is called a *first order conditional autoregressive model* or a CAR(1) model.
- ▶ Every node is conditionally dependent on its *four nearest neighbours*
- ▶ This is also called a *First Order Random Walk* or RW(1) model.

(INFORMAL) DEFINITION OF A GMRF

- ▶ A GMRF is a Gaussian distribution where the non-zero elements of the precision (inverse covariance) matrix are defined by the graph structure.
- ▶ In the previous example the precision matrix is tridiagonal since each variable is connected only to its predecessor and successor.



USES FOR THE SIMPLE 1-DIMENSIONAL PROCESSES IN R-INLA

- ▶ The AR(1) process can be used for time simple time effects
- ▶ A random walk (RW) process for “smooth effects”

$$x_i - x_{i-1} \sim N(0, \sigma^2)$$

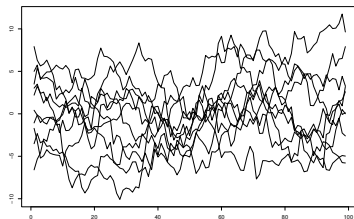
- ▶ A second-order random walk (RW2) for even “smoother” effects

$$(x_i - 2x_{i-1} + x_{i-2}) \sim N(0, \sigma^2)$$

RANDOM WALK

Can be used with a

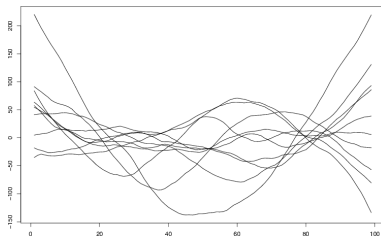
```
formula = Y ~ ... + f(covariate, model="rw1")
```



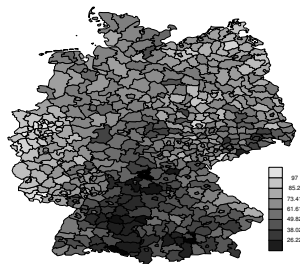
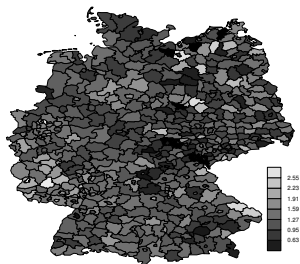
SECOND-ORDER RANDOM WALK

Can be used with a

```
formula = Y ~ ... + f(covariate, model="rw2")
```



LARYNX CANCER RELATIVE RISK



LARYNX CANCER RELATIVE RISK

Use a simple count model

$$y_i \sim \text{Poisson}(E_i e^{\nu_i}),$$

where the log-relative risk ν_i is modelled as

$$\nu_i = \text{Covariates} + \text{Spatial} + \text{Noise}.$$

In R-INLA

```
inla(formula = Y~...+f(region, model="besag",  
                        graph.file=g),  
      family="poisson", ...)
```

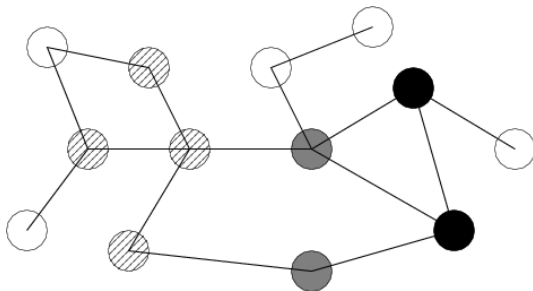
THE MARKOV PROPERTY ON A GRAPH

Let \mathbf{x} be a GMRF wrt $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

The global Markov property:

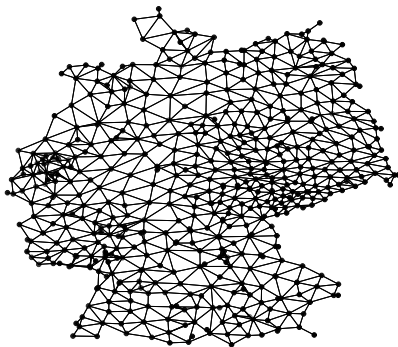
$$\mathbf{x}_A \perp \mathbf{x}_B \mid \mathbf{x}_C$$

for all disjoint sets A, B and C where C separates A and B , and A and B are non-empty.



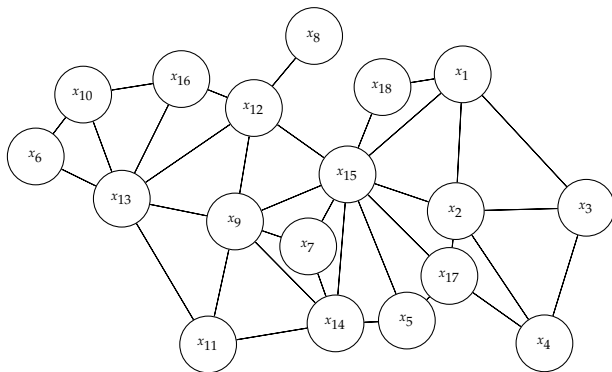
FULL GRAPH

Connecting all the neighbouring areas give the following graph



SUB GRAPH

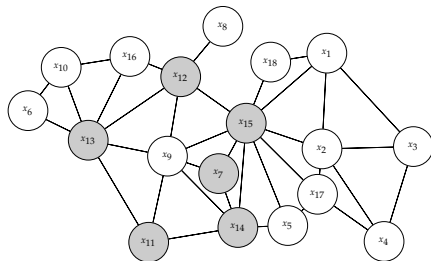
Let us focus on one small part of the graph



BESAG MODEL

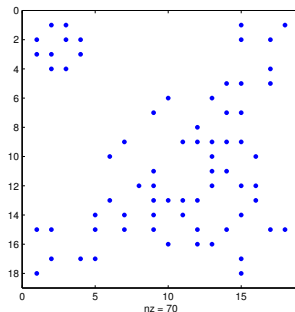
We apply a Besag model where each region conditionally has a Gaussian distribution with mean equal to the average of the neighbours and a precision proportional to the number of neighbours

$$x_9 | \mathbf{x}_{-9} \sim \mathcal{N} \left(\frac{1}{6} (x_7 + x_{11} + x_{12} + x_{13} + x_{14} + x_{15}), \frac{1}{6\tau} \right)$$



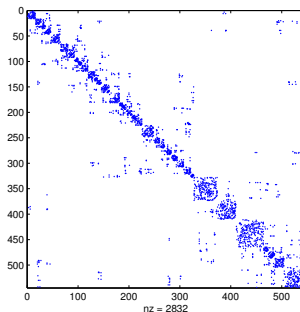
PRECISION MATRIX OF SUB GRAPH

The sub graph leads to a precision matrix with 21.6% non-zero elements.



PRECISION MATRIX OF FULL GRAPH

The full graph leads to a precision matrix with 0.1% non-zero elements.



INTRINSIC GMRFs

- ▶ The Besag model is not proper
- ▶ There are linear combinations of the variables that have infinite variance or zero precision.
- ▶ This is not allowed in a proper distribution.
- ▶ In the Besag model it is caused by the fact that the conditional distributions give no information about the “mean”.

INTRINSIC GMRFs

- ▶ Distributions of this type (usually) become proper when one introduces observations
- ▶ **Identifiability issues:** for a Besag model with an intercept in the model introduce a constraint to stop the Besag from stealing the effect of the intercept.
- ▶ R-INLA uses $\sum_i x_i = 0$.

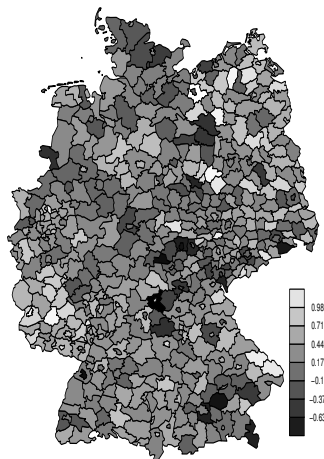
IT TURNS OUT THE BESAG MODEL DOESN'T FIT VERY WELL!

- ▶ The problem is that it only accounts for similarities between regions
- ▶ But it doesn't take into account that every region will have a little bit of individual spice
- ▶ The solution is to add an i.i.d. random effect in each region (a random intercept)
- ▶ This was the work of Besag, York and Mollié, so we call this the BYM model.

DISEASE MAPPING: THE BYM-MODEL

- ▶ Data $y_i \sim \text{Poisson}(E_i \exp(\eta_i))$
- ▶ Log-relative risk

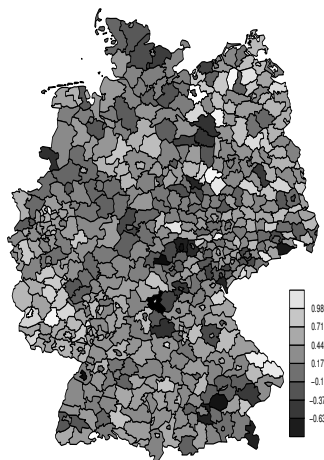
$$\eta_i = \mu + u_i + v_i + f(c_i)$$
- ▶ Structured/spatial component u
- ▶ Unstructured component v
- ▶ $f(c)$ is the non-linear effect of a covariate c .
- ▶ Precisions τ_u and τ_v ; smoothing parameter τ_f
- ▶ Common to use independent Gamma-priors



DISEASE MAPPING: THE BYM-MODEL

- ▶ Data $y_i \sim \text{Poisson}(E_i \exp(\eta_i))$
- ▶ Log-relative risk

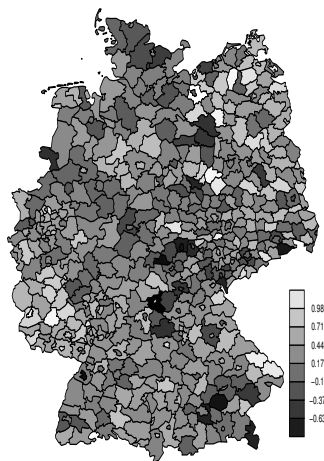
$$\eta_i = \mu + u_i + v_i + f(c_i)$$
- ▶ Structured/spatial component u
- ▶ Unstructured component v
- ▶ $f(c)$ is the non-linear effect of a covariate c .
- ▶ Precisions τ_u and τ_v ; smoothing parameter τ_f
- ▶ Common to use independent Gamma-priors



DISEASE MAPPING: THE BYM-MODEL

- ▶ Data $y_i \sim \text{Poisson}(E_i \exp(\eta_i))$
- ▶ Log-relative risk

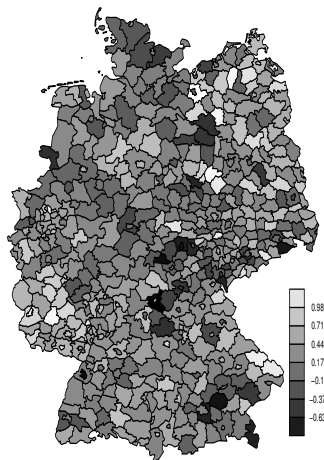
$$\eta_i = \mu + u_i + v_i + f(c_i)$$
- ▶ Structured/spatial component u
- ▶ Unstructured component v
- ▶ $f(c)$ is the non-linear effect of a covariate c .
- ▶ Precisions τ_u and τ_v ; smoothing parameter τ_f
- ▶ Common to use independent Gamma-priors



DISEASE MAPPING: THE BYM-MODEL

- ▶ Data $y_i \sim \text{Poisson}(E_i \exp(\eta_i))$
- ▶ Log-relative risk

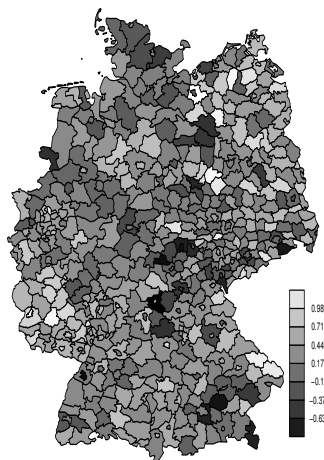
$$\eta_i = \mu + u_i + v_i + f(c_i)$$
- ▶ Structured/spatial component u
- ▶ Unstructured component v
- ▶ $f(c)$ is the non-linear effect of a covariate c .
- ▶ Precisions τ_u and τ_v ; smoothing parameter τ_f
- ▶ Common to use independent Gamma-priors



DISEASE MAPPING: THE BYM-MODEL

- ▶ Data $y_i \sim \text{Poisson}(E_i \exp(\eta_i))$
- ▶ Log-relative risk

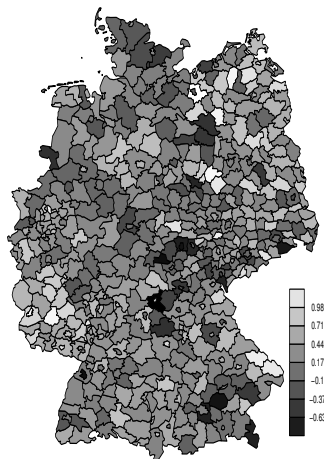
$$\eta_i = \mu + u_i + v_i + f(c_i)$$
- ▶ Structured/spatial component u
- ▶ Unstructured component v
- ▶ $f(c)$ is the non-linear effect of a covariate c .
- ▶ Precisions τ_u and τ_v ; smoothing parameter τ_f
- ▶ Common to use independent Gamma-priors



DISEASE MAPPING: THE BYM-MODEL

- ▶ Data $y_i \sim \text{Poisson}(E_i \exp(\eta_i))$
- ▶ Log-relative risk

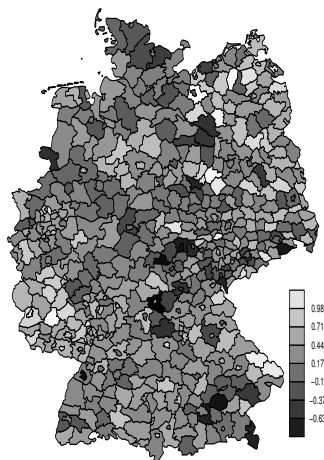
$$\eta_i = \mu + u_i + v_i + f(c_i)$$
- ▶ Structured/spatial component u
- ▶ Unstructured component v
- ▶ $f(c)$ is the non-linear effect of a covariate c .
- ▶ Precisions τ_u and τ_v ; smoothing parameter τ_f
- ▶ Common to use independent Gamma-priors



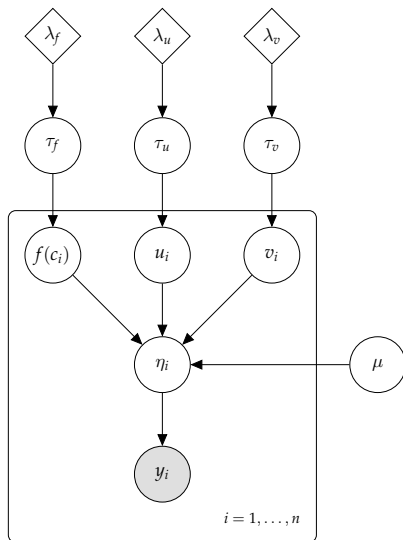
DISEASE MAPPING: THE BYM-MODEL

- ▶ Data $y_i \sim \text{Poisson}(E_i \exp(\eta_i))$
- ▶ Log-relative risk

$$\eta_i = \mu + u_i + v_i + f(c_i)$$
- ▶ Structured/spatial component u
- ▶ Unstructured component v
- ▶ $f(c)$ is the non-linear effect of a covariate c .
- ▶ Precisions τ_u and τ_v ; smoothing parameter τ_f
- ▶ Common to use independent Gamma-priors



COMPLICATED MODEL COMPONENTS



Does this make sense?

THINK OF THE VARIANCE

- ▶ The variance not explained by the covariate is modelled with u_i and v_i
- ▶ This amount of variance we can have is controlled by the independent precision parameters τ_u and τ_v
- ▶ This is ugly!
- ▶ It would be much easier to have one parameter controlling the scale of the random effect, and another controlling its makeup
- ▶ This is implemented as the `bym2` model in INLA

DISEASE MAPPING (II)

Rewrite the model as

$$\eta = \frac{1}{\sqrt{\tau}} \left(\sqrt{1-\gamma}v + \sqrt{\gamma}u \right)$$

- ▶ Marginal precisions τ .
- ▶ γ gives it interpretation: independence ($\gamma = 0$), maximal dependence ($\gamma = 1$)
- ▶ PC prior on γ depends on the graph!
- ▶ Parameters control different features. Use the PC priors (later!) for τ and γ separately.

DISEASE MAPPING (II)

Rewrite the model as

$$\eta = \frac{1}{\sqrt{\tau}} \left(\sqrt{1-\gamma}v + \sqrt{\gamma}u \right)$$

- ▶ Marginal precisions τ .
- ▶ γ gives it interpretation: independence ($\gamma = 0$), maximal dependence ($\gamma = 1$)
- ▶ PC prior on γ depends on the graph!
- ▶ Parameters control different features. Use the PC priors (later!) for τ and γ separately.

DISEASE MAPPING (II)

Rewrite the model as

$$\eta = \frac{1}{\sqrt{\tau}} \left(\sqrt{1-\gamma}v + \sqrt{\gamma}u \right)$$

- ▶ Marginal precisions τ .
- ▶ γ gives it interpretation: independence ($\gamma = 0$), maximal dependence ($\gamma = 1$)
- ▶ PC prior on γ depends on the graph!
- ▶ Parameters control different features. Use the PC priors (later!) for τ and γ separately.

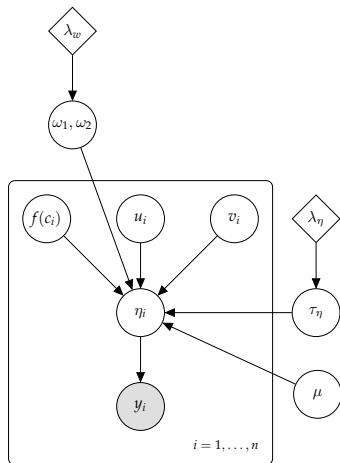
DISEASE MAPPING (II)

Rewrite the model as

$$\eta = \frac{1}{\sqrt{\tau}} \left(\sqrt{1-\gamma}v + \sqrt{\gamma}u \right)$$

- ▶ Marginal precisions τ .
- ▶ γ gives it interpretation: independence ($\gamma = 0$), maximal dependence ($\gamma = 1$)
- ▶ PC prior on γ depends on the graph!
- ▶ Parameters control different features. Use the PC priors (later!) for τ and γ separately.

BUILDING A BETTER BYM



This re-parameterisation in terms of "meaningful" parameters makes it easier to set priors and leads to more stable inference.

NEVER FORGET

Your model doesn't fit!

“All models are wrong, some models are useful” — George Box

BAYESIAN MODEL COMPARISON

- ▶ There is *no gold standard*
- ▶ It depends on what you want to do
- ▶ Basically two types
 - ▶ Ones that look at the posterior probability of the data under the model
 - ▶ Ones that look at how model the data fits the data
- ▶ The best hope is to have a model that represents data that wasn't used to fit it...

DEVIANE INFORMATION CRITERIA

Based on the *deviance*

$$D(\mathbf{x}; \boldsymbol{\theta}) = -2 \sum_i \log(y_i | x_i, \boldsymbol{\theta})$$

and

$$DIC = 2 \times \text{Mean} (D(\mathbf{x}; \boldsymbol{\theta})) - D(\text{Mean}(\mathbf{x}); \boldsymbol{\theta}^*)$$

This is quite easy to compute, but somewhat controversial

BAYESIAN CROSS-VALIDATION

Easy to compute using the INLA-approach

$$\pi(\mathbf{y}_i | \mathbf{y}_{-i}) = \int_{\boldsymbol{\theta}} \left\{ \int_{x_i} \pi(\mathbf{y}_i | x_i, \boldsymbol{\theta}) \pi(x_i | \mathbf{y}_{-i}, \boldsymbol{\theta}) dx_i \right\} \pi(\boldsymbol{\theta} | \mathbf{y}_{-i}) d\boldsymbol{\theta}$$

where

$$\pi(x_i | \mathbf{y}_{-i}, \boldsymbol{\theta}) \propto \frac{\pi(x_i | \mathbf{y}, \boldsymbol{\theta})}{\pi(\mathbf{y}_i | x_i, \boldsymbol{\theta})}$$

- ▶ If it is very small, this point may be an “outlier” under the model
- ▶ We can use this to define a score (bigger is better)

$$LCPO = \sum_i \log(\pi(y = y_i | \mathbf{y}_{-i}))$$

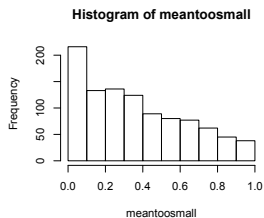
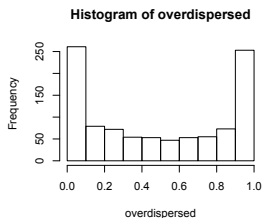
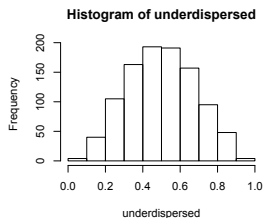
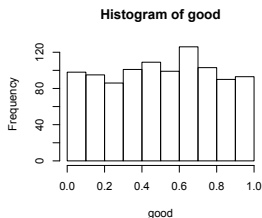
AUTOMATIC DETECTION OF “SURPRISING” OBSERVATIONS

Compute

$$pit_i = \text{Prob}(y_i^{\text{new}} \leq y_i \mid \mathbf{y}_{-i})$$

- ▶ pit_i shows how well the i th data point is predicted by the rest of the data
- ▶ If the model is true and the response is continuous, these PIT values are uniformly distributed
- ▶ We can use this to inspect the model fit

GOOD AND BAD PIT PLOTS



Areal/Regional Models

DISEASE MAPPING

Disease mapping has a long history in epidemiology, and may be defined as the estimation and presentation of summary measures of health outcomes.

The aims of disease mapping include

- ▶ simple description – a visual summary of geographical risk.,
- ▶ hypothesis generation,
- ▶ allocation of health care resources, assessment of inequalities, and
- ▶ estimation of background variability in underlying risk in order to place epidemiological studies in context.

DISEASE MAPPING

Aims:

- ▶ Provide estimates of risk by area to inform public health resource allocation.
- ▶ Give clues to etiology via informal examination of maps with exposure maps, components of spatial versus non-spatial residual variability may also provide clues to source of variability (e.g. environmental exposures usually have spatial structure). The formal examination is carried out via spatial regression.
- ▶ In general mapping is based on count data (which is more routinely available) – may also be carried out with point data but much less common (case-control studies are explicitly carried out to examine an exposure of interest, and cannot inform on risk without additional information).

DISEASE MAPPING: EXAMPLE

- ▶ Study on Lung and Brain cancer in the North-West of England as an illustration of smoothing techniques using hierarchical models.
- ▶ Two tumors were chosen to contrast mapping techniques for relatively non-rare (lung), and relatively rare (brain) cancers.
- ▶ The absence of information on smoking means that for lung cancer in particular the analysis should be viewed as illustrative only (since a large fraction of the residual variability would disappear if smoking information were included).
- ▶ Residual spatial dependence is induced by missing variables that are predictive of disease outcome (or data errors/model misspecification).

DISEASE MAPPING: EXAMPLE

Study details:

- ▶ Study period is 1981–1991. Incidence data by postcode, but the analysis is carried out at the ward level
- ▶ 144 wards in the study region
- ▶ For brain cancer the median number of cases per ward over the 11 year period is 6 with a range of 0 to 17
- ▶ For lung the median number is 20 with range 0–60
- ▶ “Expected counts” were based on ward-level populations from the 1991 census, by 5-year age bands and sex

DISEASE MAPPING: EXAMPLE

- ▶ The following figures show the SIRs together with the smoothed rates for lung and brain cancer, respectively.
- ▶ Notice that for lung the smoothed area-level relative risk estimates are not dramatically different from the raw versions – the large number of cases here mean that the raw SIRs are relatively stable.
- ▶ For brain we see a much greater smoothing of the estimates as compared to the raw relative risks

DISEASE MAPPING: EXAMPLE

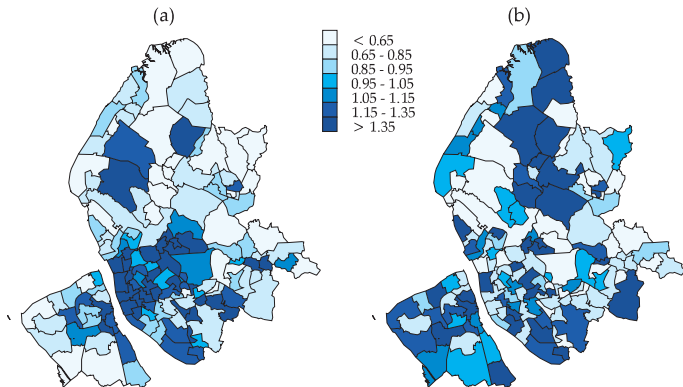


Figure: SIRs for (a) lung cancer, and (b) brain cancer.

DISEASE MAPPING: EXAMPLE

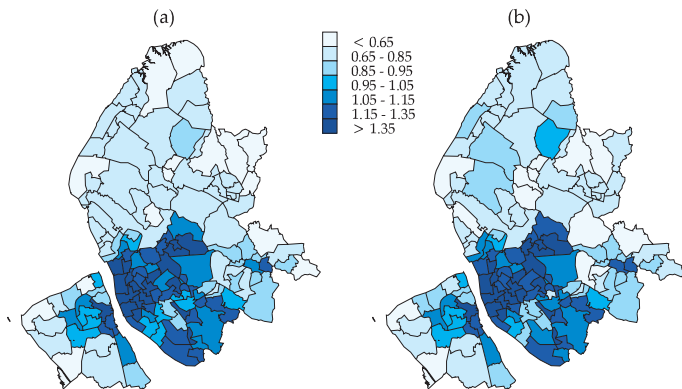


Figure: Smoothed SIRs for lung cancer under (a) a conditional spatial model, and (b) a marginal spatial model.

DISEASE MAPPING: EXAMPLE

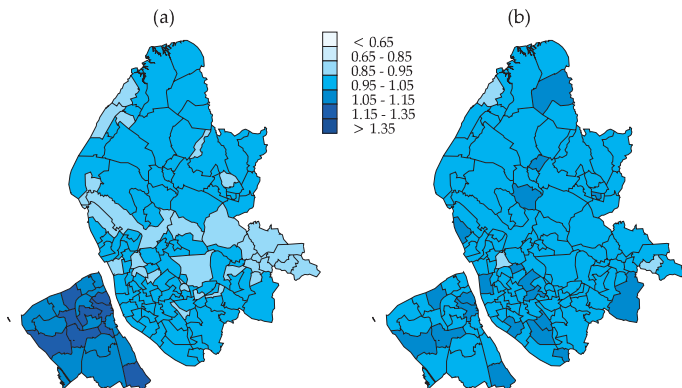


Figure: Smoothed SIRs for brain cancer under (a) a conditional spatial model, and (b) a marginal spatial model.

DISEASE MAPPING

There are difficulties with the mapping of raw estimates since, for small areas and rare diseases in particular, these estimates will be dominated by sampling variability.

For the model

$$Y_i \sim \text{Poisson}(E_i\theta_i)$$

the MLE is

$$\hat{\theta}_i = \text{SMR}_i = \frac{Y_i}{E_i}$$

with variance

$$\text{var}(\hat{\theta}_i) = \frac{\theta_i}{E_i}$$

so that areas with small E_i have high associated variance.

DISEASE MAPPING: EXAMPLE

Next figure shows the SMRs for the Scottish lip cancer data, and indicates a large spread with an increasing trend in the south-north direction.

The variance of the estimate is $\text{var}(\text{SMR}_i) = \text{SMR}_i/E_i$, which will be large if E_i is small.

For the Scottish data the expected numbers are highly variable. This variability suggests that there is a good chance that the extreme SMRs are based on small expected numbers (many of the large, sparsely-populated rural areas in the north have high SMRs).

A plot of SMRs versus the estimated standard errors clearly illustrates that the high SMRs have high associated standard error.

DISEASE MAPPING: EXAMPLE

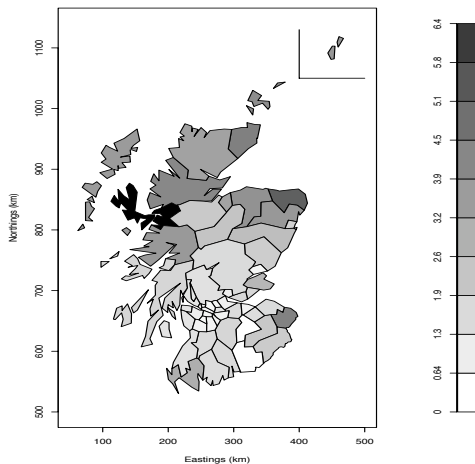


Figure: SMRs in 56 counties of Scotland.

SMOOTHING MODELS

- ▶ Variability in expected numbers led to methods being developed to *smooth* the SMRs using hierarchical/random effects models
- ▶ These models use the data from the totality of areas to provide more reliable estimates in each of the constituent areas.

POISSON-GAMMA MODEL WITHOUT COVARIATES

This two-stage model that offers analytic tractability and ease of estimation.

Assume the first stage likelihood is given by

$$Y_i | \theta_i, \beta \sim_{ind} \text{Poisson}(\mu E_i \theta_i),$$

where μ is the overall relative risk, and reflects differences between the reference rates and the rates in the study region.

At the second stage the random effects θ_i are assigned a distribution. We initially assume that across the map the deviations of the relative risks from the mean, μ , are modelled by

$$\theta_i | \alpha \sim_{iid} \text{Ga}(\alpha, \alpha),$$

a gamma distribution with mean 1, and variance $1/\alpha$.

POISSON-GAMMA MODEL WITHOUT COVARIATES

The advantage of this Poisson-gamma formulation is that the marginal distribution of $Y_i|\mu, \alpha$ (obtained by integrating out the random effects θ_i), is negative binomial.

Marginally, the mean and variance are given, respectively, by

$$\begin{aligned}E[Y_i|\mu, \alpha] &= E_i\mu \\ \text{var}(Y_i|\mu, \alpha) &= E[Y_i|\mu, \alpha](1 + E[Y_i|\mu, \alpha]/\alpha),\end{aligned}$$

so that the variance increases as a quadratic function of the mean, and the scale parameter α can accommodate different levels of “overdispersion”.

This form is substantively more reasonable than the naive Poisson model; it is important to consider excess-Poisson variability resulting from unmeasured confounders, data anomalies in numerator and denominator, and model misspecification.

POISSON-GAMMA MODEL WITH COVARIATES

With area-level covariates we have the model

$$Y_i | \theta_i, \beta \sim_{ind} \text{Poisson}(\mu_i E_i \theta_i),$$

where $\mu_i = \mu(\mathbf{x}_i, \beta)$ describes a regression model in area-level covariates \mathbf{x}_i . At the second stage the random effects θ_i are assigned a distribution. We assume that across the map the deviations of the relative risks from the mean, μ_i , are modelled by

$$\theta_i | \alpha \sim_{iid} \text{Ga}(\alpha, \alpha),$$

a gamma distribution with mean 1, and variance $1/\alpha$.

POISSON-LOGNORMAL MODEL

The Poisson-gamma model offers analytic tractability, but does not easily allow the incorporation of spatial random effects.

A Poisson-lognormal non-spatial random effect model is given by:

$$Y_i | \beta, V_i \sim_{ind} \text{Poisson}(E_i \mu_i e^{V_i}) \quad V_i \sim_{iid} N(0, \sigma_v^2)$$

where V_i are area-specific random effects that capture the residual or unexplained (log) relative risk of disease in area i , $i = 1, \dots, n$.

Whereas in the Poisson-Gamma model we have $\theta \sim \text{Ga}(\alpha, \alpha)$, here we have $\theta = e^{V_i} \sim \text{LogNormal}(0, \sigma^2)$.

REVIEW

- ▶ The aim is to provide stable relative risk estimates for area-level data.
- ▶ We have assumed that the relative risks arise from a common gamma/lognormal distribution, which allows smoothing towards a common value.
- ▶ The Poisson-Gamma model offers a useful exploratory option. For example, an empirical Bayes approach, estimates the parameters of the negative binomial model (β and α) and then combines the gamma distribution with the data to obtain the empirical Bayes posterior distribution for the relative risks.
- ▶ Poisson-lognormal model does not give a marginal distribution of known form, but does naturally lead to the addition of spatial random effects.
- ▶ Poisson-lognormal marginal variance is of the same quadratic form as the negative binomial.

SPATIAL REGRESSION

Aims:

- ▶ Examination of the association between disease outcome and explanatory variables, in a spatial setting
- ▶ Dependence is important but not the object of interest
- ▶ Conventional modeling approaches such as logistic regression for point data, and loglinear models for count data may be used though if there is significant residual variation methods must acknowledge this in order to obtain appropriate standard errors.
- ▶ Also included in this enterprise is the examination of risk with respect to a specific point or line putative source of pollution which may change over time.
- ▶ For count data in particular, the disease mapping models we describe may be extended to incorporate a regression component.

SPATIAL REGRESSION: EXAMPLE

Study about childhood asthma in Anchorage, Alaska. Study details:

- ▶ Data were collected on first grade children in Anchorage, with questionnaires being sent to the parents of children in 13 school districts
- ▶ Data on 905 children, with 885 aged 5–7 years. There were 804 children without asthma, the remainder being cases.
- ▶ The exposure of interest is exposure to pollution from traffic.
- ▶ Traffic counts were recorded at roads throughout the study region and a 50m buffer was created at the nearest intersection to the child's residential address and within this buffer traffic counts were aggregated (for confidentiality reasons the exact residential locations were not asked for in the survey).

AUTOREGRESSIVE ANALOG; THE CAR APPROACH

Space unlike time not ordered. Conditional autoregressive approach (CAR) is one way of emulating the AR model. Let:

- ▶ $D = \{s_1, \dots, s_m\}$ be the lattice
- ▶ $X(s_i, t)$ be a response of interest
- ▶ \mathbf{X}_i be all responses but $X(s_i, t)$
- ▶ $N(s_i)$ be s_i neighbourhood

The CAR model:

$$X(s_i, t) \sim N(\mu_i, \sigma_i^2), \text{ for all } i$$

with

$$E(X(s_i, t) | \mathbf{X}_i) = \sum_{s_j \in N(s_i)} c_{ij} X(s_j, t), \quad \text{Var}(X(s_i, t) | \mathbf{X}_i) = \tau_i^2$$

THE CAR APPROACH

Does CAR necessarily determine a joint distribution

$$[X(s_i, t), \dots, X(s_m, t)]?$$

Answer: Yes under reasonable conditions.

CAR IN A PROCESS MODEL

The following hierarchical model induces a CAR structure.

► **Measurement model:**

$$Y(s_i, t) \sim \text{ind Poi}(\exp [X(s_i, t)])$$

► **Process model:**

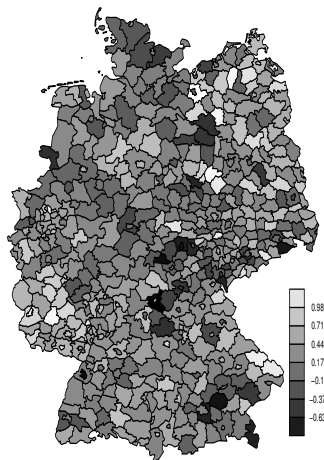
$$[\mathbf{X}|\boldsymbol{\beta}, \tau^2, \phi] = \text{Gau}(\mathbf{Z}\boldsymbol{\beta}, \Sigma[\tau^2, \phi])$$

where \mathbf{Z} represents site specific covariates or factors & $\Sigma[\tau^2, \phi]$ the CAR neighbourhood structure. **Parameter model:** $[\boldsymbol{\beta}, \tau^2, \phi]$

DISEASE MAPPING: THE BYM-MODEL

- ▶ Data $y_i \sim \text{Poisson}(E_i \exp(\eta_i))$
- ▶ Log-relative risk

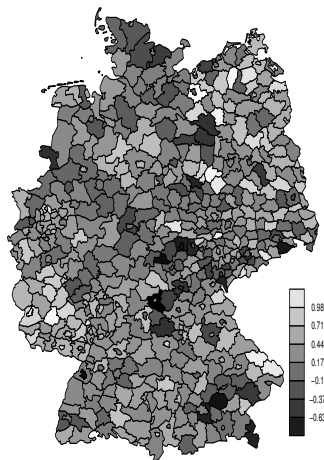
$$\eta_i = \mu + u_i + v_i + z' \beta$$
- ▶ Structured/spatial component u
- ▶ Unstructured component v
- ▶ z are relevant covariates
- ▶ Precisions τ_u and τ_v
- ▶ Common to use independent Gamma-priors



DISEASE MAPPING: THE BYM-MODEL

- ▶ Data $y_i \sim \text{Poisson}(E_i \exp(\eta_i))$
- ▶ Log-relative risk

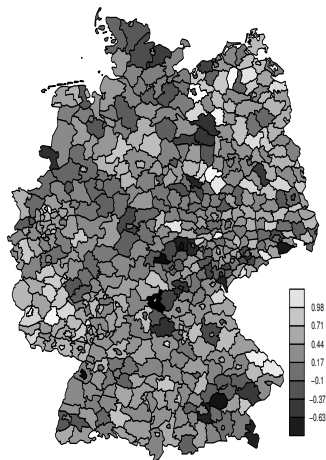
$$\eta_i = \mu + u_i + v_i + z' \beta$$
- ▶ Structured/spatial component u
- ▶ Unstructured component v
- ▶ z are relevant covariates
- ▶ Precisions τ_u and τ_v
- ▶ Common to use independent Gamma-priors



DISEASE MAPPING: THE BYM-MODEL

- ▶ Data $y_i \sim \text{Poisson}(E_i \exp(\eta_i))$
- ▶ Log-relative risk

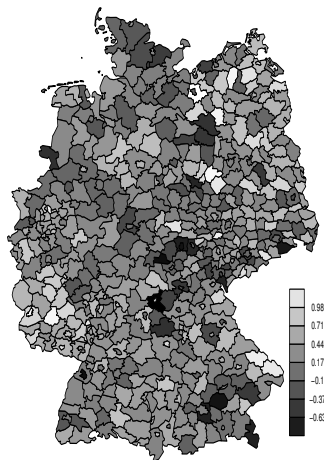
$$\eta_i = \mu + u_i + v_i + z' \beta$$
- ▶ Structured/spatial component u
- ▶ Unstructured component v
- ▶ z are relevant covariates
- ▶ Precisions τ_u and τ_v
- ▶ Common to use independent Gamma-priors



DISEASE MAPPING: THE BYM-MODEL

- ▶ Data $y_i \sim \text{Poisson}(E_i \exp(\eta_i))$
- ▶ Log-relative risk

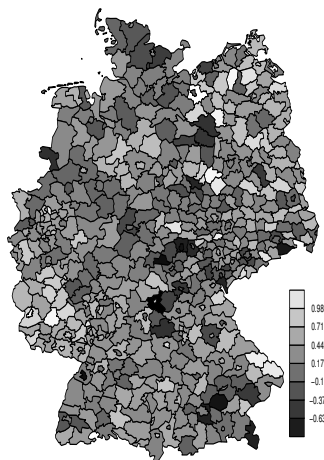
$$\eta_i = \mu + u_i + v_i + z' \beta$$
- ▶ Structured/spatial component u
- ▶ Unstructured component v
- ▶ z are relevant covariates
- ▶ Precisions τ_u and τ_v
- ▶ Common to use independent Gamma-priors



DISEASE MAPPING: THE BYM-MODEL

- ▶ Data $y_i \sim \text{Poisson}(E_i \exp(\eta_i))$
- ▶ Log-relative risk

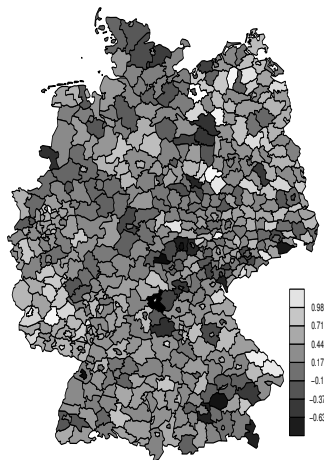
$$\eta_i = \mu + u_i + v_i + z' \beta$$
- ▶ Structured/spatial component u
- ▶ Unstructured component v
- ▶ z are relevant covariates
- ▶ Precisions τ_u and τ_v
- ▶ Common to use independent Gamma-priors



DISEASE MAPPING: THE BYM-MODEL

- ▶ Data $y_i \sim \text{Poisson}(E_i \exp(\eta_i))$
- ▶ Log-relative risk

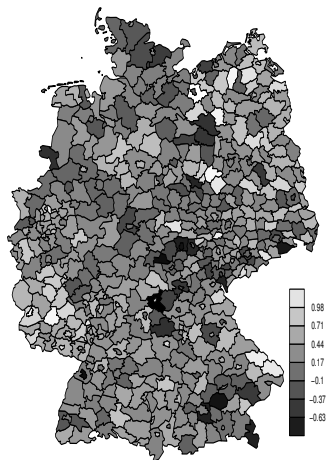
$$\eta_i = \mu + u_i + v_i + z' \beta$$
- ▶ Structured/spatial component u
- ▶ Unstructured component v
- ▶ z are relevant covariates
- ▶ Precisions τ_u and τ_v
- ▶ Common to use independent Gamma-priors



DISEASE MAPPING: THE BYM-MODEL

- ▶ Data $y_i \sim \text{Poisson}(E_i \exp(\eta_i))$
- ▶ Log-relative risk

$$\eta_i = \mu + u_i + v_i + z' \beta$$
- ▶ Structured/spatial component u
- ▶ Unstructured component v
- ▶ z are relevant covariates
- ▶ Precisions τ_u and τ_v
- ▶ Common to use independent Gamma-priors



A JOINT MODEL

- ▶ Assume that (U_1, \dots, U_n) arise from a zero mean multivariate normal distribution with variances $\text{var}(U_i) = \sigma_u^2$ and correlations $\text{corr}(U_i, U_j) = \exp(-\phi d_{ij}) = \rho^{d_{ij}}$ where d_{ij} is the distance between the centroids of areas i and j , and $\rho > 0$ is a parameter that determines the extent of the correlation.
- ▶ This model is *isotropic* since it assumes that the correlation is the same in all spatial directions. We refer to this as the *joint* model, since we have specified the joint distribution for \mathbf{U} .
- ▶ More generally the correlations can be modelled as $\text{corr}(U_i, U_j) = \exp(-(\phi d_{ij})^\kappa)$.

THE ICAR MODEL

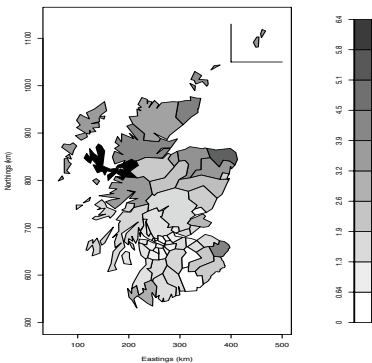
- ▶ A common model is to assign the spatial random effects an intrinsic conditional autoregressive (ICAR) prior.
- ▶ Under this specification it is assumed that

$$U_i | U_j, j \in \partial_i \sim N \left(\bar{U}_i, \frac{\omega_u^2}{m_i} \right),$$

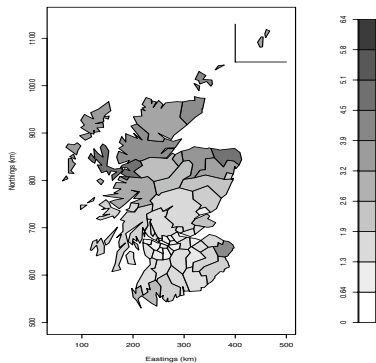
where ∂_i is the set of neighbors of area i , m_i is the number of neighbours, and \bar{U}_i is the mean of the spatial random effects of these neighbors.

- ▶ The parameter ω_u^2 is a conditional variance and its magnitude determines the amount of spatial variation.
- ▶ The variance parameters σ_v^2 and ω_u^2 are on different scales, σ_v is on the log odds scale while ω_u is on the log odds scale, *conditional* on $U_j, j \in \partial_i$; hence they are not comparable (in contrast to the joint model in which σ_u is on the same scale as σ_v).

- ▶ Notice that if ω_u^2 is “small” then although the residual is strongly dependent on the neighboring value the overall contribution to the residual relative risk is small.
- ▶ This is a little counterintuitive but stems from spatial models having two aspects, strength of dependence and total amount of spatial dependence, and in the ICAR model there is only a single parameter which controls both aspects.
- ▶ In the joint model the strength is determined by ρ and the total amount by σ_u^2 .
- ▶ A non-spatial random effect should always be included along with the ICAR random effect since this model cannot take a limiting form that allows non-spatial variability; in the joint model with U_i only, this is achieved as $\rho \rightarrow 0$. If the majority of the variability is non-spatial, inference for this model might incorrectly suggest that spatial dependence was present.



(a) SMR estimates



(b) Smoothed estimates

Figure: Raw and smoothed estimates in 56 counties of Scotland.

MARKOV RANDOM FIELD (MRF)

As before time t is fixed &

- ▶ $D = \{s_1, \dots, s_m\}$ be the lattice
- ▶ $X(s_i, t)$ be a response of interest
- ▶ \mathbf{X}_i be all responses but $X(s_i, t)$
- ▶ $N(s_i)$ be s_i neighbourhood

MRF models:

$$[X(s_i, t) | \{X(s_j, t), s_j \in N(s_i)\}] \text{ for all } i$$

MARKOV RANDOM FIELD (MRF)

When do the local MRF models determine

$$[X(s_1, t), \dots, X(s_m, t)]?$$

Hammersley - Clifford Theorem: Gives necessary and sufficient conditions involving the *Gibbs distributions*.

CONDITIONAL SPECIFICATION OF GMRFs

Consider the system of normal full conditionals that satisfies

$$\mathbb{E}(x_i | \mathbf{x}_{-i}) = - \sum_{j=1}^n \beta_{ij} x_j$$

and

$$\text{Prec}(x_i | \mathbf{x}_{-i}) = \kappa_i > 0.$$

Theorem

This full conditional specifies a multivariate Gaussian joint distributions $\mathbf{x} \sim N(\mathbf{0}, \mathbf{Q}^{-1})$ if and only if the matrix

$$Q_{ij} = \begin{cases} \kappa_i \beta_{ij}, & i \neq j, \\ \kappa_i, & i = j \end{cases}$$

is symmetric positive definite.

MARKOV RANDOM FIELDS - EXAMPLE

Example: Crown die back in birch trees.

Features:

- ▶ Single timepoint, t .
- ▶ $X(s_i, t)$ = probability a tree's crown dies back in region i with $m(s_i, t)$ trees in it.
- ▶ $Y(s_i, t)$ = # of trees with die back $\sim \text{Bin}(m(s_i, t), X(s_i, t))$.
- ▶ $N(s_i)$ = all regions within 48 km of i . Conditional on $N(s_i)$, $X(s_i, t)$ has beta distribution with parameters depending on responses in neighbours.
- ▶ parsimonious model but unclear how to include time

MARKOV RANDOM FIELDS: ASSESSMENT

PROS:

- ▶ elegant, simple mathematics + computational power
- ▶ may be useful component in hierarchical model

CONS:

- ▶ compatible joint distribution may not exist
- ▶ neighbours may be hard to specify
- ▶ a new site may not have neighbours for spatial prediction!
- ▶ conditional distributions may be hard to specify when “sites” are regions

NOTES ON AREAL DATA

Sometimes areal data can profitably be modelled as an aggregate of individual data.

- ▶ Can reflect greater uncertainty due to variation within areas
- ▶ Was used to explore the ecological effect and develop model that avoids it.

PRIOR CHOICE

For regression parameters $\beta = (\beta_0, \beta_1, \dots, \beta_J)$, an improper prior

$$p(\beta) \propto 1$$

may often be used, but in very circumstances such a choice may lead to an improper posterior.

If there are a large numbers of covariates, or high dependence amongst the elements of \mathbf{x} , then more informative priors will be beneficial.

HOW DO WE SET A PRIOR ON A PRECISION?

- ▶ Lots of “expert guidance” from the literature
- ▶ Some of it is saying how to set priors on the precision
- ▶ Some of it is setting priors on the precision for a specific problem
- ▶ Conjugate priors, reference priors, weakly informative priors, ...
- ▶ When will it end?

We only want this effect to be in the model if it is required to fit the data.

We don't want a prior that the data has to drag towards “no effect”!

HOW DO WE SET A PRIOR ON A PRECISION?

- ▶ Lots of “expert guidance” from the literature
- ▶ Some of it is saying how to set priors on the precision
- ▶ Some of it is setting priors on the precision for a specific problem
- ▶ Conjugate priors, reference priors, weakly informative priors, ...
- ▶ When will it end?

We only want this effect to be in the model if it is required to fit the data.

We don't want a prior that the data has to drag towards “no effect”!

HOW DO WE SET A PRIOR ON A PRECISION?

- ▶ Lots of “expert guidance” from the literature
- ▶ Some of it is saying how to set priors on the precision
- ▶ **Some of it is setting priors on the precision for a specific problem**
- ▶ Conjugate priors, reference priors, weakly informative priors, ...
- ▶ When will it end?

We only want this effect to be in the model if it is required to fit the data.

We don't want a prior that the data has to drag towards “no effect”!

HOW DO WE SET A PRIOR ON A PRECISION?

- ▶ Lots of “expert guidance” from the literature
- ▶ Some of it is saying how to set priors on the precision
- ▶ Some of it is setting priors on the precision for a specific problem
- ▶ Conjugate priors, reference priors, weakly informative priors, ...
- ▶ When will it end?

We only want this effect to be in the model if it is required to fit the data.

We don't want a prior that the data has to drag towards “no effect”!

HOW DO WE SET A PRIOR ON A PRECISION?

- ▶ Lots of “expert guidance” from the literature
- ▶ Some of it is saying how to set priors on the precision
- ▶ Some of it is setting priors on the precision for a specific problem
- ▶ Conjugate priors, reference priors, weakly informative priors, ...
- ▶ When will it end?

We only want this effect to be in the model if it is required to fit the data.

We don't want a prior that the data has to drag towards “no effect”!

HOW DO WE SET A PRIOR ON A PRECISION?

- ▶ Lots of “expert guidance” from the literature
- ▶ Some of it is saying how to set priors on the precision
- ▶ Some of it is setting priors on the precision for a specific problem
- ▶ Conjugate priors, reference priors, weakly informative priors, ...
- ▶ When will it end?

We only want this effect to be in the model if it is required to fit the data.

We don't want a prior that the data has to drag towards “no effect”!

HOW DO WE SET A PRIOR ON A PRECISION?

- ▶ Lots of “expert guidance” from the literature
- ▶ Some of it is saying how to set priors on the precision
- ▶ Some of it is setting priors on the precision for a specific problem
- ▶ Conjugate priors, reference priors, weakly informative priors, ...
- ▶ When will it end?

We only want this effect to be in the model if it is required to fit the data.

We don't want a prior that the data has to drag towards “no effect”!

BASIC INSTINCT

A base model

- ▶ We have a model component with distribution $\pi(\mathbf{x} \mid \xi)$
- ▶ ξ is a **flexibility parameter**,
- ▶ $\xi = 0$ indexes the **base model**
- ▶ The base model is the **simplest model**

Idea: Build a prior that has a mode at the base model. The posterior only concentrates on $\xi > 0$ if the data requires the more complex model.

SOME EXAMPLES

Case	Parameter	ξ	Base
Student-t	ν (dof)	$\xi = 1/\nu$	$\xi = 0$ (Gaussian)
IID	τ (precision)	$\xi = 1/\tau$	$\xi = 0$ (no random effect)
IGMRFs	τ (precision)	$\xi = 1/\tau$	$\xi = 0$ (const, linear, plane)
AR(1)	ρ (correlation)	$\xi = \rho$	$\xi = 0$ (no dep. in time)
		$\xi = \rho$	$\xi = 1$ (no changes in time)
FGN	H (Hurst param.)	$\xi = H$	$\xi = 0.5$ (White noise)
Correlation matrix	\mathbf{R}	$\xi = \mathbf{R}$	$\xi = \mathbf{I}$ (no correlation)

THE PLEASURE PRINCIPLE

To build a prior that knows about the base model, the idea of **Penalised Complexity (PC) Priors** is introduced:

- ▶ PC priors are an attempt to put together a set of principles that lead to a unique prior
- ▶ You can interrogate / criticise / modify the principles individually

PRINCIPLE I: OCCAM'S RAZOR

Prefer simplicity over complexity

Consider the more complex model

$$\pi(x|\xi), \quad \xi \geq 0$$

with base model $\pi(x|\xi = 0)$.

- ▶ The prior for $\xi \geq 0$ should penalise the complexity introduced by ξ
- ▶ The prior should be decaying with increasing measure by the complexity (the mode should be at the base model)

A prior will cause **overfitting/force complexity** if, loosely speaking,

$$\pi_\xi(\xi = 0) = 0$$

PRINCIPLE I: OCCAM'S RAZOR

Prefer simplicity over complexity

Consider the more complex model

$$\pi(x|\xi), \quad \xi \geq 0$$

with base model $\pi(x|\xi = 0)$.

- ▶ The prior for $\xi \geq 0$ should penalise the complexity introduced by ξ
- ▶ The prior should be decaying with increasing measure by the complexity (the mode should be at the base model)

A prior will cause **overfitting/force complexity** if, loosely speaking,

$$\pi_{\xi}(\xi = 0) = 0$$

PRINCIPLE I: OCCAM'S RAZOR

Prefer simplicity over complexity

Consider the more complex model

$$\pi(x|\xi), \quad \xi \geq 0$$

with base model $\pi(x|\xi = 0)$.

- ▶ The prior for $\xi \geq 0$ should penalise the complexity introduced by ξ
- ▶ The prior should be decaying with increasing measure by the complexity (the mode should be at the base model)

A prior will cause **overfitting/force complexity** if, loosely speaking,

$$\pi_{\xi}(\xi = 0) = 0$$

PRINCIPLE I: OCCAM'S RAZOR

Prefer simplicity over complexity

Consider the more complex model

$$\pi(x|\xi), \quad \xi \geq 0$$

with base model $\pi(x|\xi = 0)$.

- ▶ The prior for $\xi \geq 0$ should penalise the complexity introduced by ξ
- ▶ The prior should be decaying with increasing measure by the complexity (the mode should be at the base model)

A prior will cause **overfitting/force complexity** if, loosely speaking,

$$\pi_{\xi}(\xi = 0) = 0$$

PRINCIPLE I: OCCAM'S RAZOR

Prefer simplicity over complexity

Consider the more complex model

$$\pi(x|\xi), \quad \xi \geq 0$$

with base model $\pi(x|\xi = 0)$.

- ▶ The prior for $\xi \geq 0$ should penalise the complexity introduced by ξ
- ▶ The prior should be decaying with increasing measure by the complexity (the mode should be at the base model)

A prior will cause **overfitting/force complexity** if, loosely speaking,

$$\pi_\xi(\xi = 0) = 0$$

PRINCIPLE II: MEASURE OF COMPLEXITY

Use Kullback-Leibler discrepancy to measure the increased complexity introduced by $\xi > 0$,

$$\text{KLD}(f||g) = \int f(x) \log \left(\frac{f(x)}{g(x)} \right) dx$$

for flexible model f and base model g .

Gives a measure of the information lost when the base model is used to approximate the more flexible models

PRINCIPLE II: MEASURE OF COMPLEXITY

Use Kullback-Leibler discrepancy to measure the increased complexity introduced by $\xi > 0$,

$$\text{KLD}(f||g) = \int f(x) \log \left(\frac{f(x)}{g(x)} \right) dx$$

for flexible model f and base model g .

Gives a measure of the information lost when the base model is used to approximate the more flexible models

PRINCIPLE III: CONSTANT RATE PENALISATION

Define

$$d(\xi) = \sqrt{2 \text{KLD}(\xi)}$$

as the (uni-directional) “distance” from flexible-model to the base model. Need the square-root to get the scale right.

Constant rate penalisation:

$$\pi(d) = \lambda \exp(-\lambda d), \quad \lambda > 0$$

with mode at $d = 0$

Invariance: OK

PRINCIPLE III: CONSTANT RATE PENALISATION

Define

$$d(\xi) = \sqrt{2 \text{KLD}(\xi)}$$

as the (uni-directional) “distance” from flexible-model to the base model. Need the square-root to get the scale right.

Constant rate penalisation:

$$\pi(d) = \lambda \exp(-\lambda d), \quad \lambda > 0$$

with mode at $d = 0$

Invariance: OK

PRINCIPLE III: CONSTANT RATE PENALISATION

Define

$$d(\xi) = \sqrt{2 \text{KLD}(\xi)}$$

as the (uni-directional) “distance” from flexible-model to the base model. Need the square-root to get the scale right.

Constant rate penalisation:

$$\pi(d) = \lambda \exp(-\lambda d), \quad \lambda > 0$$

with mode at $d = 0$

Invariance: OK

PRINCIPLE IV: USER-DEFINED SCALING

The rate λ is determined from knowledge of the *scale* or some interpretable property or impact, $Q(\xi)$ of ξ :

$$\Pr(Q(\xi) > U) = \alpha$$

- ▶ Problem dependent: must be!!!
- ▶ Can make the prior more informative or weakly informative this way

PRINCIPLE IV: USER-DEFINED SCALING

The rate λ is determined from knowledge of the *scale* or some interpretable property or impact, $Q(\xi)$ of ξ :

$$\Pr(Q(\xi) > U) = \alpha$$

- ▶ Problem dependent: must be!!!
- ▶ Can make the prior more informative or weakly informative this way

THE PRECISION OF A GAUSSIAN

PC prior for the precision τ when $\tau = \infty$ defines the base model

- ▶ “random effects”/iid-model
- ▶ The smoothing parameter in spline models
- ▶ etc...

Result Let $\pi_\tau(\tau)$ be a prior for $\tau > 0$ where $E(\tau) < \infty$, then $\pi_d(0) = 0$ and the prior overfits.

THE PRECISION OF A GAUSSIAN

PC prior for the precision τ when $\tau = \infty$ defines the base model

- ▶ “random effects”/iid-model
- ▶ The smoothing parameter in spline models
- ▶ etc...

Result Let $\pi_\tau(\tau)$ be a prior for $\tau > 0$ where $E(\tau) < \infty$, then $\pi_d(0) = 0$ and the prior overfits.

THE PRECISION CASE (II)

The resulting prior is a type-2 Gumbel

$$\pi(\tau) = \frac{\lambda}{2} \tau^{-3/2} \exp(-\lambda/\sqrt{\tau}), \quad \mathbb{E}(\tau) = \infty,$$

$\text{Prob}(\sigma > u) = \alpha$ gives

$$\lambda = -\frac{\ln(\alpha)}{u}$$

Alternative interpretation

$$\pi(\sigma) = \lambda \exp(-\lambda\sigma)$$

THE PRECISION CASE (II)

The resulting prior is a type-2 Gumbel

$$\pi(\tau) = \frac{\lambda}{2} \tau^{-3/2} \exp(-\lambda/\sqrt{\tau}), \quad \mathbb{E}(\tau) = \infty,$$

$\text{Prob}(\sigma > u) = \alpha$ gives

$$\lambda = -\frac{\ln(\alpha)}{u}$$

Alternative interpretation

$$\pi(\sigma) = \lambda \exp(-\lambda\sigma)$$

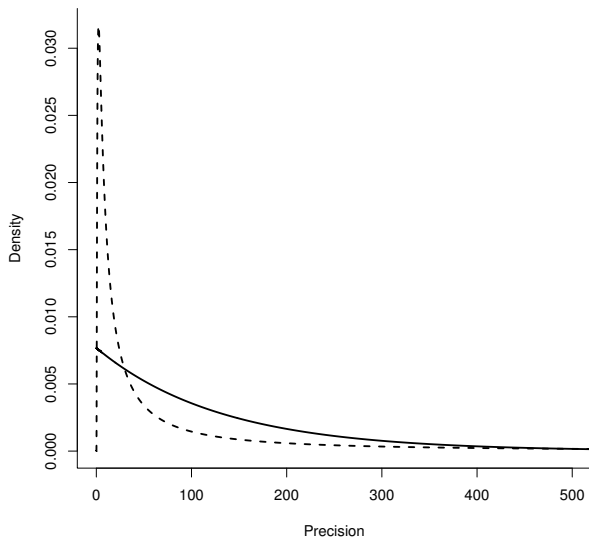
LINK WITH THE TRADITION

Other (good) priors for the precision are

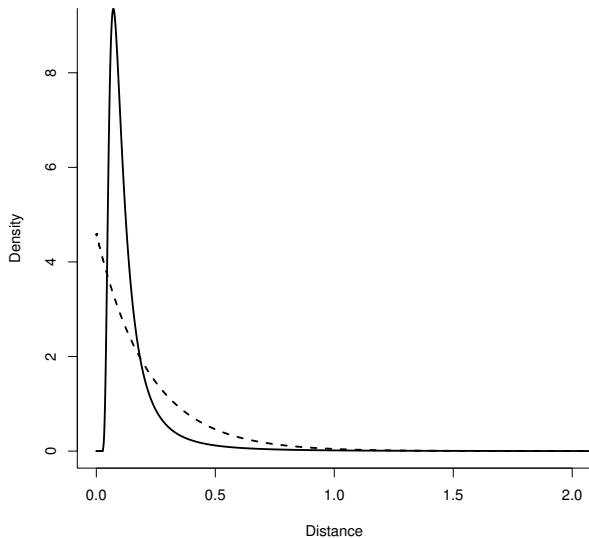
- ▶ A half-Gaussian on the standard deviation. (lighter tail than the PC prior)
- ▶ A half-Cauchy on the standard deviation. (heavier tail)
- ▶ A half-Student-t with more than 2 d.o.f. (heavier tail, similar risk properties)

The important thing here is that they all have a maximum at the base model. The tail behaviour is more “controversial”

COMPARISON WITH A SIMILAR GAMMA-PRIOR



COMPARISON WITH A SIMILAR GAMMA-PRIOR



PRIORS FOR THE BYM MODEL

- ▶ Data $y_i \sim \text{Poisson}(E_i \exp(\eta_i))$
- ▶ Log-relative risk $\eta_i = \mu + u_i + v_i + z' \beta$
- ▶ Structured/spatial component \mathbf{u}
- ▶ Unstructured component v
- ▶ z are relevant covariates
- ▶ Precisions τ_u and τ_v
- ▶ Assume a BYM model for $v_j + u_j$ where $v_j \stackrel{\text{iid}}{\sim} N(0, \tau_v^{-1})$ and $\mathbf{u} \sim N(0, \tau_u^{-1} \mathbf{Q}^+)$ is a Besag model.

THE STRUCTURED EFFECT

The structured difference in u between neighbouring regions is $N(0, \tau_u^{-1})$.

$$\pi(\mathbf{u}) \propto \tau_u^{(n-1)/2} \exp \left(-\frac{\tau_u}{2} \sum_{i \sim j} (u_i - u_j)^2 \right). \quad (1)$$

“ $i \sim j$ ” denotes the set of all *unordered* pairs of neighbours.

- ▶ This is the Besag model.
- ▶ It is rank deficient.
- ▶ How do we put a prior on τ_u ?
- ▶ **Big thing:** It will depend on the graph!

HOW TO SCALE?

- ▶ The precision is $\mathbf{Q}_u = \tau_u \mathbf{R}_u$
- ▶ The marginal variance of u_i is $\tau_u^{-1} [\mathbf{R}_u^{-1}]_{ii}$
- ▶ If we make the second term ≈ 1 , then τ_u is a precision parameter and our life is easier.
- ▶ Scale so that $\sigma_*^2 = 1$, where (f.ex)

$$\sigma_*^2 = \exp(\text{mean}(\log(\text{diag}(\mathbf{R}^-))))$$

- ▶ If we know the null-space of \mathbf{R} we can compute $\text{diag}(\mathbf{R}^-)$ using sparse matrix algebra.

This prior will then mean the same thing for every problem!

Correct scaling is implicit in the definition of the PC prior.

HOW TO SCALE?

- ▶ The precision is $\mathbf{Q}_u = \tau_u \mathbf{R}_u$
- ▶ The marginal variance of u_i is $\tau_u^{-1} [\mathbf{R}_u^{-1}]_{ii}$
- ▶ If we make the second term ≈ 1 , then τ_u is a precision parameter and our life is easier.
- ▶ Scale so that $\sigma_*^2 = 1$, where (f.ex)

$$\sigma_*^2 = \exp(\text{mean}(\log(\text{diag}(\mathbf{R}^-))))$$

- ▶ If we know the null-space of \mathbf{R} we can compute $\text{diag}(\mathbf{R}^-)$ using sparse matrix algebra.

This prior will then mean the same thing for every problem!

Correct scaling is implicit in the definition of the PC prior.

HOW TO SCALE?

- ▶ The precision is $\mathbf{Q}_u = \tau_u \mathbf{R}_u$
- ▶ The marginal variance of u_i is $\tau_u^{-1} [\mathbf{R}_u^{-1}]_{ii}$
- ▶ If we make the second term ≈ 1 , then τ_u is a precision parameter and our life is easier.
- ▶ Scale so that $\sigma_*^2 = 1$, where (f.ex)

$$\sigma_*^2 = \exp(\text{mean}(\log(\text{diag}(\mathbf{R}^-))))$$

- ▶ If we know the null-space of \mathbf{R} we can compute $\text{diag}(\mathbf{R}^-)$ using sparse matrix algebra.

This prior will then mean the same thing for every problem!

Correct scaling is implicit in the definition of the PC prior.

HOW TO SCALE?

- ▶ The precision is $\mathbf{Q}_u = \tau_u \mathbf{R}_u$
- ▶ The marginal variance of u_i is $\tau_u^{-1} [\mathbf{R}_u^{-1}]_{ii}$
- ▶ If we make the second term ≈ 1 , then τ_u is a precision parameter and our life is easier.
- ▶ Scale so that $\sigma_*^2 = 1$, where (f.ex)

$$\sigma_*^2 = \exp(\text{mean}(\log(\text{diag}(\mathbf{R}^-))))$$

- ▶ If we know the null-space of \mathbf{R} we can compute $\text{diag}(\mathbf{R}^-)$ using sparse matrix algebra.

This prior will then mean the same thing for every problem!

Correct scaling is implicit in the definition of the PC prior.

HOW TO SCALE?

- ▶ The precision is $\mathbf{Q}_u = \tau_u \mathbf{R}_u$
- ▶ The marginal variance of u_i is $\tau_u^{-1} [\mathbf{R}_u^{-1}]_{ii}$
- ▶ If we make the second term ≈ 1 , then τ_u is a precision parameter and our life is easier.
- ▶ Scale so that $\sigma_*^2 = 1$, where (f.ex)

$$\sigma_*^2 = \exp(\text{mean}(\log(\text{diag}(\mathbf{R}^-))))$$

- ▶ If we know the null-space of \mathbf{R} we can compute $\text{diag}(\mathbf{R}^-)$ using sparse matrix algebra.

This prior will then mean the same thing for every problem!

Correct scaling is implicit in the definition of the PC prior.

HOW TO SCALE?

- ▶ The precision is $\mathbf{Q}_u = \tau_u \mathbf{R}_u$
- ▶ The marginal variance of u_i is $\tau_u^{-1} [\mathbf{R}_u^{-1}]_{ii}$
- ▶ If we make the second term ≈ 1 , then τ_u is a precision parameter and our life is easier.
- ▶ Scale so that $\sigma_*^2 = 1$, where (f.ex)

$$\sigma_*^2 = \exp(\text{mean}(\log(\text{diag}(\mathbf{R}^-))))$$

- ▶ If we know the null-space of \mathbf{R} we can compute $\text{diag}(\mathbf{R}^-)$ using sparse matrix algebra.

This prior will then mean the same thing for every problem!

Correct scaling is implicit in the definition of the PC prior.

HOW TO SCALE?

- ▶ The precision is $\mathbf{Q}_u = \tau_u \mathbf{R}_u$
- ▶ The marginal variance of u_i is $\tau_u^{-1} [\mathbf{R}_u^{-1}]_{ii}$
- ▶ If we make the second term ≈ 1 , then τ_u is a precision parameter and our life is easier.
- ▶ Scale so that $\sigma_*^2 = 1$, where (f.ex)

$$\sigma_*^2 = \exp(\text{mean}(\log(\text{diag}(\mathbf{R}^-))))$$

- ▶ If we know the null-space of \mathbf{R} we can compute $\text{diag}(\mathbf{R}^-)$ using sparse matrix algebra.

This prior will then mean the same thing for every problem!

Correct scaling is implicit in the definition of the PC prior.

BUILDING A BETTER BYM

Base model = 0 \rightarrow iid \rightarrow dependence = more flexible model

Rewrite the model as

$$\eta = \frac{1}{\sqrt{\tau}} \left(\sqrt{1-\gamma}v^* + \sqrt{\gamma}u^* \right)$$

where \cdot^* is a unit-variance standardised model.

- ▶ Marginal precisions τ .
- ▶ γ gives it interpretation: independence ($\gamma = 0$), maximal dependence ($\gamma = 1$)
- ▶ PC prior on γ (base model $\gamma = 0$) depends on the graph!
- ▶ Parameters control different features. Use the PC priors for τ and γ separately.

BUILDING A BETTER BYM

Base model = 0 \rightarrow iid \rightarrow dependence = more flexible model

Rewrite the model as

$$\eta = \frac{1}{\sqrt{\tau}} \left(\sqrt{1-\gamma}v^* + \sqrt{\gamma}u^* \right)$$

where \cdot^* is a unit-variance standardised model.

- ▶ Marginal precisions τ .
- ▶ γ gives it interpretation: independence ($\gamma = 0$), maximal dependence ($\gamma = 1$)
- ▶ PC prior on γ (base model $\gamma = 0$) depends on the graph!
- ▶ Parameters control different features. Use the PC priors for τ and γ separately.

BUILDING A BETTER BYM

Base model = 0 \rightarrow iid \rightarrow dependence = more flexible model

Rewrite the model as

$$\eta = \frac{1}{\sqrt{\tau}} \left(\sqrt{1-\gamma}v^* + \sqrt{\gamma}u^* \right)$$

where \cdot^* is a unit-variance standardised model.

- ▶ Marginal precisions τ .
- ▶ γ gives it interpretation: independence ($\gamma = 0$), maximal dependence ($\gamma = 1$)
- ▶ PC prior on γ (base model $\gamma = 0$) depends on the graph!
- ▶ Parameters control different features. Use the PC priors for τ and γ separately.

BUILDING A BETTER BYM

Base model = 0 \rightarrow iid \rightarrow dependence = more flexible model

Rewrite the model as

$$\eta = \frac{1}{\sqrt{\tau}} \left(\sqrt{1-\gamma}v^* + \sqrt{\gamma}u^* \right)$$

where \cdot^* is a unit-variance standardised model.

- ▶ Marginal precisions τ .
- ▶ γ gives it interpretation: independence ($\gamma = 0$), maximal dependence ($\gamma = 1$)]
- ▶ PC prior on γ (base model $\gamma = 0$) depends on the graph!
- ▶ Parameters control different features. Use the PC priors for τ and γ separately.

BUILDING A BETTER BYM

Base model = 0 \rightarrow iid \rightarrow dependence = more flexible model

Rewrite the model as

$$\eta = \frac{1}{\sqrt{\tau}} \left(\sqrt{1-\gamma}v^* + \sqrt{\gamma}u^* \right)$$

where \cdot^* is a unit-variance standardised model.

- ▶ Marginal precisions τ .
- ▶ γ gives it interpretation: independence ($\gamma = 0$), maximal dependence ($\gamma = 1$)]
- ▶ PC prior on γ (base model $\gamma = 0$) depends on the graph!
- ▶ Parameters control different features. Use the PC priors for τ and γ separately.

CONCLUSIONS

- ▶ The aim of the PC prior project is to make priors that can find nothing when nothing is there
- ▶ The new BYM parameterisation gives a more interpretable way to look at the structure of the random effect
- ▶ The PC priors for this model satisfy a basic principle: **If something important in your model changes, the corresponding priors should also change**

CONCLUSIONS

- ▶ The aim of the PC prior project is to make priors that can find nothing when nothing is there
- ▶ The new BYM parameterisation gives a more interpretable way to look at the structure of the random effect
- ▶ The PC priors for this model satisfy a basic principle: **If something important in your model changes, the corresponding priors should also change**

CONCLUSIONS

- ▶ The aim of the PC prior project is to make priors that can find nothing when nothing is there
- ▶ The new BYM parameterisation gives a more interpretable way to look at the structure of the random effect
- ▶ The PC priors for this model satisfy a basic principle: **If something important in your model changes, the corresponding priors should also change**

OTHER APPLICATIONS

This example shows just a corner of the power of PC priors

- ▶ Splines
- ▶ Skew-Gaussian distributions
- ▶ Correlation matrices
- ▶ AR(p)
- ▶ Over-dispersion in Negative Binomials
- ▶ Hurst Parameters for fractional Brownian motion
- ▶ Degrees of freedom in a Student-t
- ▶ Parameters in Gaussian random fields (partially identifiable!)
- ▶ Non-stationary GRFs
- ▶ Correlated random effects
- ▶ Variances in multilevel models
- ▶ + + +

Continuous spatial models

LET'S TALK ABOUT COVARIATES

- ▶ There are two types of covariate:
 - ▶ Covariates that are only available at the observation (age, sex, species, etc).
 - ▶ Covariates that “exist” everywhere (time, temperature, precipitation, land use, etc.)
- ▶ The second of these is hard!
- ▶ We usually do not measure these everywhere!

THE PERILS AND PECCADILLOES OF PRE-PROCESSING

So what do we do?

- ▶ We have an important covariate and we need it's value everywhere
- ▶ We measure it at a (relatively) small number of places
- ▶ We need to construct and *interpolant*

This is a statistical question, so let's look at how to do it.

WHY IS THIS A STATISTICAL QUESTION

- ▶ The reconstructed covariate is now “incorrect”
- ▶ So we need to be able to quantify the uncertainty
- ▶ Plugging in the interpolated covariate as if it was exact *leads to bias!*

Tomorrow, we'll talk about how to incorporate the uncertainty into modelling, but for now let's just focus on constructing a good interpolant.

SPATIAL INTERPOLANTS

$$(\text{observed covariate})_i = (\text{true covariate at location } i) + (\text{error})_i$$

- ▶ We treat the *observed* covariates as being measured with error
- ▶ The errors are usually assumed to be independent and identically distributed (i.i.d.)
 - ▶ Usually, we take them to be Gaussian
 - ▶ If we think there may be outliers, we might use something else (e.g. a Student-T distribution)
 - ▶ The only change in R-INLA is in the `family` argument in the INLA call

SO HOW DOES THAT HELP US FILL IN THE COVARIATE?

$$(\text{observed covariate})_i = (\text{true covariate at location } i) + (\text{error})_i$$

or

$$y_i = x(s_i) + \epsilon_i$$

We need priors!

- ▶ We have chosen the error distribution to be $\epsilon_i \sim N(0, \sigma^2)$
 - ▶ A *zero mean* means that there is no systemic measurement error
 - ▶ A *common variance* means that everything was measured the same way
- ▶ Now we need a prior on the truth...

GAUSSIAN RANDOM FIELDS

If we have a process that is occurring everywhere in space, it is natural to try to model it using some sort of function.

- ▶ This is hard!
- ▶ We typically make our lives easier by making everything Gaussian.
- ▶ What makes a function Gaussian?

GAUSSIAN RANDOM FIELDS

If we are trying to model $x(s)$ what sort of things do we need?

- ▶ We don't ever observe a function *everywhere*.
- ▶ If \mathbf{x} is a vector of observations of $x(s)$ at different locations, we want this to be normally distributed:

$$\mathbf{x} = (x(s_1), \dots, x(s_p))^T \sim N(\mathbf{0}, \Sigma_{x(s_1), \dots, x(s_p)})$$

- ▶ This is actually quite tricky: the covariance matrix Σ will need to depend on the set of observation sites and always has to be positive definite.
- ▶ It turns out you can actually do this by setting $\Sigma_{ij} = c(s_i, s_j)$ for some *covariance function* $c(\cdot, \cdot)$.
- ▶ Not every function will ensure that Σ is positive definite!

GAUSSIAN RANDOM FIELDS

If we are trying to model $x(s)$ what sort of things do we need?

- ▶ We don't ever observe a function *everywhere*.
- ▶ If \mathbf{x} is a vector of observations of $x(s)$ at different locations, we want this to be normally distributed:

$$\mathbf{x} = (x(s_1), \dots, x(s_p))^T \sim N(\mathbf{0}, \Sigma_{x(s_1), \dots, x(s_p)})$$

- ▶ This is actually quite tricky: the covariance matrix Σ will need to depend on the set of observation sites and always has to be positive definite.
- ▶ It turns out you can actually do this by setting $\Sigma_{ij} = c(s_i, s_j)$ for some *covariance function* $c(\cdot, \cdot)$.
- ▶ Not every function will ensure that Σ is positive definite!

GAUSSIAN RANDOM FIELDS

If we are trying to model $x(s)$ what sort of things do we need?

- ▶ We don't ever observe a function *everywhere*.
- ▶ If \mathbf{x} is a vector of observations of $x(s)$ at different locations, we want this to be normally distributed:

$$\mathbf{x} = (x(s_1), \dots, x(s_p))^T \sim N(\mathbf{0}, \Sigma_{x(s_1), \dots, x(s_p)})$$

- ▶ This is actually quite tricky: the covariance matrix Σ will need to depend on the set of observation sites and always has to be positive definite.
- ▶ It turns out you can actually do this by setting $\Sigma_{ij} = c(s_i, s_j)$ for some *covariance function* $c(\cdot, \cdot)$.
- ▶ Not every function will ensure that Σ is positive definite!

GAUSSIAN RANDOM FIELDS

If we are trying to model $x(s)$ what sort of things do we need?

- ▶ We don't ever observe a function *everywhere*.
- ▶ If \mathbf{x} is a vector of observations of $x(s)$ at different locations, we want this to be normally distributed:

$$\mathbf{x} = (x(s_1), \dots, x(s_p))^T \sim N(\mathbf{0}, \Sigma_{x(s_1), \dots, x(s_p)})$$

- ▶ This is actually quite tricky: the covariance matrix Σ will need to depend on the set of observation sites and always has to be positive definite.
- ▶ It turns out you can actually do this by setting $\Sigma_{ij} = c(\mathbf{s}_i, \mathbf{s}_j)$ for some *covariance function* $c(\cdot, \cdot)$.
- ▶ Not every function will ensure that Σ is positive definite!

GAUSSIAN RANDOM FIELDS

If we are trying to model $x(s)$ what sort of things do we need?

- ▶ We don't ever observe a function *everywhere*.
- ▶ If \mathbf{x} is a vector of observations of $x(s)$ at different locations, we want this to be normally distributed:

$$\mathbf{x} = (x(s_1), \dots, x(s_p))^T \sim N(\mathbf{0}, \Sigma_{x(s_1), \dots, x(s_p)})$$

- ▶ This is actually quite tricky: the covariance matrix Σ will need to depend on the set of observation sites and always has to be positive definite.
- ▶ It turns out you can actually do this by setting $\Sigma_{ij} = c(\mathbf{s}_i, \mathbf{s}_j)$ for some *covariance function* $c(\cdot, \cdot)$.
- ▶ **Not every function will ensure that Σ is positive definite!**

A GOOD “FIRST MODEL”

Stationary random fields

A GRF is **stationary** if:

- ▶ has mean zero.
- ▶ the covariance between two points depends only on the distance and direction between those points.

It is **isotropic** if the covariance only depends on the *distance between the points*.

- ▶ Zero mean \rightarrow remove the mean
- ▶ Stationarity is a *mathematical assumption* and may have no bearing on reality
- ▶ But it makes lots of things easier.

A GOOD “FIRST MODEL”

Stationary random fields

A GRF is **stationary** if:

- ▶ has mean zero.
- ▶ the covariance between two points depends only on the distance and direction between those points.

It is **isotropic** if the covariance only depends on the *distance between the points*.

- ▶ Zero mean \rightarrow remove the mean
- ▶ Stationarity is a *mathematical assumption* and may have no bearing on reality
- ▶ But it makes lots of things easier.

A GOOD “FIRST MODEL”

Stationary random fields

A GRF is **stationary** if:

- ▶ has mean zero.
- ▶ the covariance between two points depends only on the distance and direction between those points.

It is **isotropic** if the covariance only depends on the *distance between the points*.

- ▶ Zero mean \rightarrow remove the mean
- ▶ Stationarity is a *mathematical assumption* and may have no bearing on reality
- ▶ But it makes lots of things easier.

A GOOD “FIRST MODEL”

Stationary random fields

A GRF is **stationary** if:

- ▶ has mean zero.
- ▶ the covariance between two points depends only on the distance and direction between those points.

It is **isotropic** if the covariance only depends on the *distance between the points*.

- ▶ Zero mean \rightarrow remove the mean
- ▶ Stationarity is a *mathematical assumption* and may have no bearing on reality
- ▶ But it makes lots of things easier.

THE THREE TYPICAL PARAMETERS FOR A GRF

- ▶ The variance (or precision) parameter:
 - ▶ This controls how wildly the function can deviate from its mean
- ▶ The range parameter
 - ▶ This controls the range over which the correlation between $x(s)$ and $x(s + h)$ is essentially zero
 - ▶ Often the “range” parameter is some transformation of this distance
- ▶ The smoothness parameter
 - ▶ Controls how differentiable the field is.
 - ▶ This essentially controls how similar nearby points are
 - ▶ Often not jointly identifiable with the range

For isotropic random fields, these parameters are constant.

THE THREE TYPICAL PARAMETERS FOR A GRF

- ▶ The variance (or precision) parameter:
 - ▶ This controls how wildly the function can deviate from its mean
- ▶ The range parameter
 - ▶ This controls the range over which the correlation between $x(\mathbf{s})$ and $x(\mathbf{s} + \mathbf{h})$ is essentially zero
 - ▶ Often the “range” parameter is some transformation of this distance
- ▶ The smoothness parameter
 - ▶ Controls how differentiable the field is.
 - ▶ This essentially controls how similar nearby points are
 - ▶ Often not jointly identifiable with the range

For isotropic random fields, these parameters are constant.

THE THREE TYPICAL PARAMETERS FOR A GRF

- ▶ The variance (or precision) parameter:
 - ▶ This controls how wildly the function can deviate from its mean
- ▶ The range parameter
 - ▶ This controls the range over which the correlation between $x(s)$ and $x(s + h)$ is essentially zero
 - ▶ Often the “range” parameter is some transformation of this distance
- ▶ The smoothness parameter
 - ▶ Controls how differentiable the field is.
 - ▶ This essentially controls how similar nearby points are
 - ▶ Often not jointly identifiable with the range

For isotropic random fields, these parameters are constant.

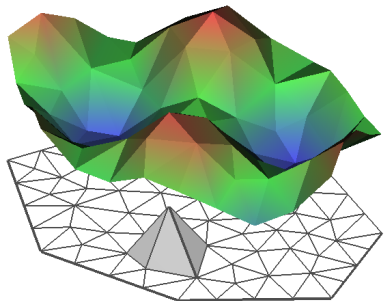
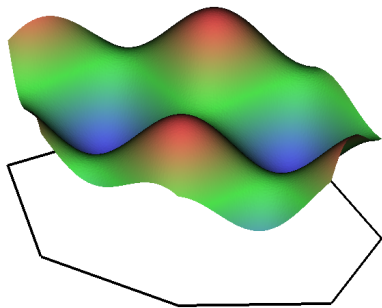
PRACTICAL CONSIDERATIONS

There are some practical problems with GRF models

- ▶ Random functions are hard to specify
- ▶ Random functions are hard to compute with
- ▶ Random functions require too much memory

We need to make a practical compromise.

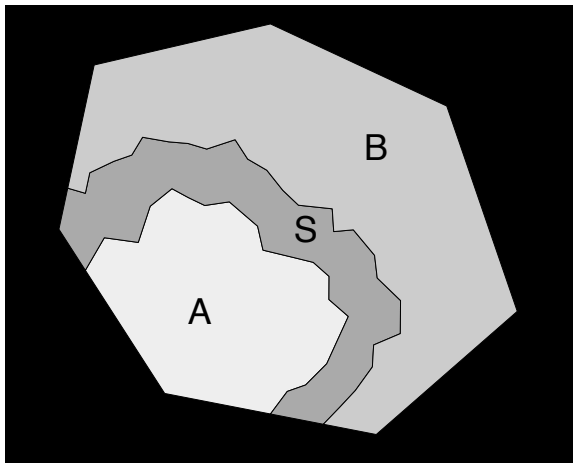
PRACTICAL CONSIDERATIONS



WHAT HAVE WE GIVEN UP?

- ▶ Pros:
 - ▶ Better computational properties
 - ▶ It's stable (i.e. hard to break)
- ▶ Cons:
 - ▶ We lose information inside the triangles
 - ▶ This will get taken into the "observation noise"

MARKOV IN SPACE!



SPDE MODELS

We call spatial Markov models defined on a mesh *SPDE models*.

SPDE* models have 3 parts

- ▶ A mesh
- ▶ A range parameter κ
- ▶ A precision parameter τ

SPDE=Stochastic Partial Differential Equation

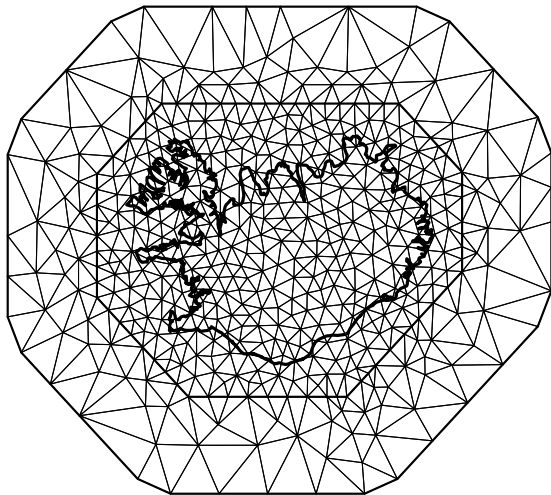
THE MESH

Meshes can be created using two different functions:

- ▶ `inla.mesh.create`: The workhorse function. An interface to the meshing code written by Finn Lindgren.
- ▶ `inla.mesh.2d`: A slightly more user friendly interface for creating practical meshes (we will focus on this one).

TYPICAL USE

```
mesh <- inla.mesh.2d(loc.domain=iceland,  
                    max.edge=c(40,800),  
                    offset=c(50,150),  
                    min.angle=25)
```



INLA.MESH.2D

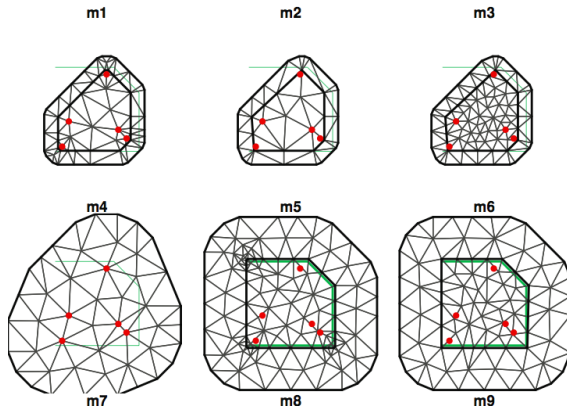
```
inla.mesh.2d(loc = NULL,  
             loc.domain = NULL,  
             offset = NULL,  
             n = NULL,  
             boundary = NULL,  
             interior = NULL,  
             max.edge,  
             min.angle = NULL,  
             cutoff = 1e-12,  
             plot.delay = NULL)
```

This function contains a mesh with two regions: the interior mesh, which is where the action happens; and the exterior mesh, which is designed to alleviate the boundary effects.

ARGUMENTS

- ▶ `loc`: Points to be included as vertices in the triangulation.
- ▶ `loc.domain`: Points not in the mesh, but that are used to define the internal mesh section (taken as the convex hull of these points).
- ▶ `offset=c(a,b)`: Distance from the points to the inner (outer) boundary. Negative numbers = relative distance.
- ▶ `boundary`: Prescribed boundary. (inla.mesh.segment type)
- ▶ `max.edge = c(a,b)`: Maximum triangle edge length in the inner (outer) segment.
- ▶ `min.angle = c(a,b)`: Minimum angle for the inner and outer segments (bigger angles are better, but harder to make)
- ▶ `cutoff`: Minimum distance between two distinct points.

GOOD AND BAD MESHES



BETWEEN THE MESH AND THE DATA

- ▶ So a good mesh probably doesn't have vertices at the data locations
- ▶ This means we need to have a way to get between values of the field at the vertices and the value of the field at the data points
- ▶ The trick is that the SPDE model is *linear* on the triangles, so the value of the field at any point is a weighted sum of the vertices of the triangle the point is in.
- ▶ In maths speak, we are observing Ax rather than x
- ▶ We call A the "A-matrix" or the "observation matrix"

MAKING OBSERVATION MATRICES IN INLA

When the observations don't occur at mesh points, we need some way to map between the latent field and the observation process.

- ▶ `inla.spde.make.A` constructs the matrix $A_{ij} = \phi_j(s_i)$ that maps a field defined on the mesh to the observation locations s_i .
- ▶ The function will also automatically deal with space-time models and replicates.
- ▶ A related function (`inla.mesh.projector`) builds an A-matrix for projecting onto a lattice. This is useful for plotting.

THE `INLA.SPDE.MAKE.A` CALL

```
inla.spde.make.A(mesh = NULL,  
                 loc = NULL,  
                 index = NULL,  
                 group = NULL,  
                 repl = 1L,  
                 n.mesh = NULL,  
                 n.group = max(group),  
                 n.repl = max(repl),  
                 group.mesh = NULL,  
                 group.method = c("nearest", "S0", "S1"),  
                 weights = NULL)
```

- ▶ The first two arguments are needed.
- ▶ `group` is needed to build space-time models
- ▶ The other arguments are fairly advanced!

OTHER MESH COMMANDS

- ▶ `inla.mesh.segment`: Constructs an object that can be given to `inla.mesh.create` as a boundary or interior segment
- ▶ `inla.mesh.boundary`: Extracts a boundary segment from a mesh.
- ▶ `inla.mesh.project` and `inla.mesh.projector`: Projects results from a mesh to a lattice. Useful for plotting.
- ▶ `inla.mesh.basis`: Constructs a B-spline basis of a given degree on a mesh.
- ▶ `inla.mesh.query`: Extracts information about the topology of the mesh (advanced!)

CONSTRUCTING SPDE MODELS

For historical reasons there are two different SPDE classes (`spde1` and `spde2`)

- ▶ `spde1` is the “classic” SPDE model!
- ▶ The `spde2` class is more flexible and defines non-stationarity in a more natural way.
- ▶ The primary difference between the two models is in the prior specification.
- ▶ At some point there will probably be an `spde3` class: We are interested in backwards-compatibility!
- ▶ For “stationary” models, these are fairly much the same (up to prior specification)

THE SPDE1 CALL

```
inla.spde.create(mesh,  
  model = c("matern", "imatern", "matern.osc"),  
  param = NULL)
```

- ▶ “imatern” is the intrinsic model ($\kappa^2 = 0$).
- ▶ “matern.osc” is an oscillating Matérn model.
- ▶ param is a list that contains alpha (1 or 2) and stuff about non-stationarity.

WHY DOES SPDE2 EXIST?

The problem with the `spde1` comes when specifying non-stationarity.

- ▶ Suppose we want to model $\tau(s) = \sum_{i=1}^k \theta_i^\tau b_i(s)$ for some basis functions $\{b_i(s)\}$. (Similar for $\kappa^2(s)$)
- ▶ The `spde1` model put i.i.d. log-normal priors on the θ_i .
- ▶ This is not a good idea: what if we want a smooth effect—should have a spline prior...
- ▶ We also penalise the (log) variance directly:

$$\log(\sigma^2) = \text{const.} - 2 \log(\kappa) - 2 \log(\tau)$$

- ▶ `spde2` fixes this by putting a multivariate normal prior on

$$\log(\boldsymbol{\tau}) = \mathbf{B}^\tau \boldsymbol{\theta}, \quad \log(\kappa^2) = \mathbf{B}^\kappa \boldsymbol{\theta}$$

with the same $\boldsymbol{\theta} \sim N(\boldsymbol{\mu}, \mathbf{Q}^{-1})$.

WHY DOES SPDE2 EXIST?

The problem with the `spde1` comes when specifying non-stationarity.

- ▶ Suppose we want to model $\tau(s) = \sum_{i=1}^k \theta_i^\tau b_i(s)$ for some basis functions $\{b_i(s)\}$. (Similar for $\kappa^2(s)$)
- ▶ The `spde1` model put i.i.d. log-normal priors on the θ_i .
- ▶ This is not a good idea: what if we want a smooth effect—should have a spline prior...
- ▶ We also penalise the (log) variance directly:

$$\log(\sigma^2) = \text{const.} - 2 \log(\kappa) - 2 \log(\tau)$$

- ▶ `spde2` fixes this by putting a multivariate normal prior on

$$\log(\tau) = \mathbf{B}^\tau \boldsymbol{\theta}, \quad \log(\kappa^2) = \mathbf{B}^\kappa \boldsymbol{\theta}$$

with the same $\boldsymbol{\theta} \sim N(\boldsymbol{\mu}, \mathbf{Q}^{-1})$.

WHY DOES SPDE2 EXIST?

The problem with the `spde1` comes when specifying non-stationarity.

- ▶ Suppose we want to model $\tau(s) = \sum_{i=1}^k \theta_i^\tau b_i(s)$ for some basis functions $\{b_i(s)\}$. (Similar for $\kappa^2(s)$)
- ▶ The `spde1` model put i.i.d. log-normal priors on the θ_i .
- ▶ This is not a good idea: what if we want a smooth effect—should have a spline prior...
- ▶ We also penalise the (log) variance directly:

$$\log(\sigma^2) = \text{const.} - 2 \log(\kappa) - 2 \log(\tau)$$

- ▶ `spde2` fixes this by putting a multivariate normal prior on

$$\log(\tau) = \mathbf{B}^\tau \boldsymbol{\theta}, \quad \log(\kappa^2) = \mathbf{B}^\kappa \boldsymbol{\theta}$$

with the same $\boldsymbol{\theta} \sim N(\boldsymbol{\mu}, \mathbf{Q}^{-1})$.

WHY DOES SPDE2 EXIST?

The problem with the `spde1` comes when specifying non-stationarity.

- ▶ Suppose we want to model $\tau(s) = \sum_{i=1}^k \theta_i^\tau b_i(s)$ for some basis functions $\{b_i(s)\}$. (Similar for $\kappa^2(s)$)
- ▶ The `spde1` model put i.i.d. log-normal priors on the θ_i .
- ▶ This is not a good idea: what if we want a smooth effect—should have a spline prior...
- ▶ We also penalise the (log) variance directly:

$$\log(\sigma^2) = \text{const.} - 2 \log(\kappa) - 2 \log(\tau)$$

- ▶ `spde2` fixes this by putting a multivariate normal prior on

$$\log(\tau) = B^\tau \theta, \quad \log(\kappa^2) = B^\kappa \theta$$

with the same $\theta \sim N(\mu, Q^{-1})$.

WHY DOES SPDE2 EXIST?

The problem with the `spde1` comes when specifying non-stationarity.

- ▶ Suppose we want to model $\tau(s) = \sum_{i=1}^k \theta_i^\tau b_i(s)$ for some basis functions $\{b_i(s)\}$. (Similar for $\kappa^2(s)$)
- ▶ The `spde1` model put i.i.d. log-normal priors on the θ_i .
- ▶ This is not a good idea: what if we want a smooth effect—should have a spline prior...
- ▶ We also penalise the (log) variance directly:

$$\log(\sigma^2) = \text{const.} - 2 \log(\kappa) - 2 \log(\tau)$$

- ▶ `spde2` fixes this by putting a multivariate normal prior on

$$\log(\boldsymbol{\tau}) = \mathbf{B}^\tau \boldsymbol{\theta}, \quad \log(\boldsymbol{\kappa}^2) = \mathbf{B}^\kappa \boldsymbol{\theta}$$

with the same $\boldsymbol{\theta} \sim N(\boldsymbol{\mu}, \mathbf{Q}^{-1})$.

THE SPDE2 CALL

```
inla.spde2.matern(mesh,  
                  alpha = 2,  
                  B.tau = matrix(c(0,1,0),1,3),  
                  B.kappa = matrix(c(0,0,1),1,3),  
                  prior.variance.nominal = 1,  
                  prior.range.nominal = NULL,  
                  prior.tau = NULL,  
                  prior.kappa = NULL,  
                  theta.prior.mean = NULL,  
                  theta.prior.prec = NULL,  
                  fractional.method = c("parsimonious", "null"))
```

ARGUMENTS

- ▶ `mesh`: An `inla.mesh` object. (Necessary)
- ▶ `alpha = 2`: The smoothness. Exact fields if it's an integer, approximate fields for non-integer α
- ▶ `B.tau`: The matrix B^τ use to define non-stationary $\tau(s)$
- ▶ `B.kappa`: As above, but for $\kappa^2(s)$
- ▶ `prior.variance.nominal`, `prior.range.nominal`: Helps the automatic prior know the scale of the variance and the range
- ▶ `prior.tau`, `prior.kappa`: Prior specification for τ and κ^2 . (not often used)
- ▶ `theta.prior.mean`, `theta.prior.prec`: Mean vector and precision matrix for θ prior.
- ▶ `fractional.method`: Method for constructing fractional α approximation

ARGUMENTS

- ▶ `mesh`: An `inla.mesh` object. (Necessary)
- ▶ `alpha = 2`: The smoothness. Exact fields if it's an integer, approximate fields for non-integer α
- ▶ `B.tau`: The matrix B^τ use to define non-stationary $\tau(s)$
- ▶ `B.kappa`: As above, but for $\kappa^2(s)$
- ▶ `prior.variance.nominal, prior.range.nominal`: Helps the automatic prior know the scale of the variance and the range
- ▶ `prior.tau, prior.kappa`: Prior specification for τ and κ^2 . (not often used)
- ▶ `theta.prior.mean, theta.prior.prec`: Mean vector and precision matrix for θ prior.
- ▶ `fractional.method`: Method for constructing fractional α approximation

ARGUMENTS

- ▶ `mesh`: An `inla.mesh` object. (Necessary)
- ▶ `alpha = 2`: The smoothness. Exact fields if it's an integer, approximate fields for non-integer α
- ▶ `B.tau`: The matrix B^τ use to define non-stationary $\tau(s)$
- ▶ `B.kappa`: As above, but for $\kappa^2(s)$
- ▶ `prior.variance.nominal, prior.range.nominal`: Helps the automatic prior know the scale of the variance and the range
- ▶ `prior.tau, prior.kappa`: Prior specification for τ and κ^2 . (not often used)
- ▶ `theta.prior.mean, theta.prior.prec`: Mean vector and precision matrix for θ prior.
- ▶ `fractional.method`: Method for constructing fractional α approximation

ARGUMENTS

- ▶ `mesh`: An `inla.mesh` object. (Necessary)
- ▶ `alpha = 2`: The smoothness. Exact fields if it's an integer, approximate fields for non-integer α
- ▶ `B.tau`: The matrix B^τ use to define non-stationary $\tau(s)$
- ▶ `B.kappa`: As above, but for $\kappa^2(s)$
- ▶ `prior.variance.nominal, prior.range.nominal`: Helps the automatic prior know the scale of the variance and the range
- ▶ `prior.tau, prior.kappa`: Prior specification for τ and κ^2 . (not often used)
- ▶ `theta.prior.mean, theta.prior.prec`: Mean vector and precision matrix for θ prior.
- ▶ `fractional.method`: Method for constructing fractional α approximation

ARGUMENTS

- ▶ `mesh`: An `inla.mesh` object. (Necessary)
- ▶ `alpha = 2`: The smoothness. Exact fields if it's an integer, approximate fields for non-integer α
- ▶ `B.tau`: The matrix B^τ use to define non-stationary $\tau(s)$
- ▶ `B.kappa`: As above, but for $\kappa^2(s)$
- ▶ `prior.variance.nominal, prior.range.nominal`: Helps the automatic prior know the scale of the variance and the range
- ▶ `prior.tau, prior.kappa`: Prior specification for τ and κ^2 . (not often used)
- ▶ `theta.prior.mean, theta.prior.prec`: Mean vector and precision matrix for θ prior.
- ▶ `fractional.method`: Method for constructing fractional α approximation

ARGUMENTS

- ▶ `mesh`: An `inla.mesh` object. (Necessary)
- ▶ `alpha = 2`: The smoothness. Exact fields if it's an integer, approximate fields for non-integer α
- ▶ `B.tau`: The matrix B^τ use to define non-stationary $\tau(s)$
- ▶ `B.kappa`: As above, but for $\kappa^2(s)$
- ▶ `prior.variance.nominal`, `prior.range.nominal`: Helps the automatic prior know the scale of the variance and the range
- ▶ `prior.tau`, `prior.kappa`: Prior specification for τ and κ^2 . (not often used)
- ▶ `theta.prior.mean`, `theta.prior.prec`: Mean vector and precision matrix for θ prior.
- ▶ `fractional.method`: Method for constructing fractional α approximation

ARGUMENTS

- ▶ `mesh`: An `inla.mesh` object. (Necessary)
- ▶ `alpha = 2`: The smoothness. Exact fields if it's an integer, approximate fields for non-integer α
- ▶ `B.tau`: The matrix B^τ use to define non-stationary $\tau(s)$
- ▶ `B.kappa`: As above, but for $\kappa^2(s)$
- ▶ `prior.variance.nominal`, `prior.range.nominal`: Helps the automatic prior know the scale of the variance and the range
- ▶ `prior.tau`, `prior.kappa`: Prior specification for τ and κ^2 . (not often used)
- ▶ `theta.prior.mean`, `theta.prior.prec`: Mean vector and precision matrix for θ prior.
- ▶ `fractional.method`: Method for constructing fractional α approximation

ARGUMENTS

- ▶ `mesh`: An `inla.mesh` object. (Necessary)
- ▶ `alpha = 2`: The smoothness. Exact fields if it's an integer, approximate fields for non-integer α
- ▶ `B.tau`: The matrix B^τ use to define non-stationary $\tau(s)$
- ▶ `B.kappa`: As above, but for $\kappa^2(s)$
- ▶ `prior.variance.nominal`, `prior.range.nominal`: Helps the automatic prior know the scale of the variance and the range
- ▶ `prior.tau`, `prior.kappa`: Prior specification for τ and κ^2 . (not often used)
- ▶ `theta.prior.mean`, `theta.prior.prec`: Mean vector and precision matrix for θ prior.
- ▶ `fractional.method`: Method for constructing fractional α approximation

A FEW MORE USEFUL COMMANDS

- ▶ `inla.spde.precision(spde, tau=..., kappa2=...)`— computes precision matrix. Less straightforward for `spde2` models
- ▶ `inla.qsample(n, Q, ...)`—Computes a sample and various other quantities needed for MCMC for precision matrix Q
- ▶ `inla.qreordering`—Computes a fill-in reducing reordering.
- ▶ `inla.qsolve`—Solve a linear system
- ▶ `inla.qinv(Q)`—Calculates the elements of the inverse corresponding to the non-zero elements of Q . Needed for computing derivatives of Gaussian likelihoods.

A FEW MORE USEFUL COMMANDS

- ▶ `inla.spde.precision(spde, tau=..., kappa2=...)`— computes precision matrix. Less straightforward for `spde2` models
- ▶ `inla.qsample(n, Q, ...)`—Computes a sample and various other quantities needed for MCMC for precision matrix Q
- ▶ `inla.qreordering`—Computes a fill-in reducing reordering.
- ▶ `inla.qsolve`—Solve a linear system
- ▶ `inla.qinv(Q)`—Calculates the elements of the inverse corresponding to the non-zero elements of Q . Needed for computing derivatives of Gaussian likelihoods.

A FEW MORE USEFUL COMMANDS

- ▶ `inla.spde.precision(spde, tau=..., kappa2=...)`— computes precision matrix. Less straightforward for `spde2` models
- ▶ `inla.qsample(n, Q, ...)`—Computes a sample and various other quantities needed for MCMC for precision matrix Q
- ▶ `inla.qreordering`—Computes a fill-in reducing reordering.
- ▶ `inla.qsolve`—Solve a linear system
- ▶ `inla.qinv(Q)`—Calculates the elements of the inverse corresponding to the non-zero elements of Q . Needed for computing derivatives of Gaussian likelihoods.

A FEW MORE USEFUL COMMANDS

- ▶ `inla.spde.precision(spde, tau=..., kappa2=...)`— computes precision matrix. Less straightforward for `spde2` models
- ▶ `inla.qsample(n, Q, ...)`—Computes a sample and various other quantities needed for MCMC for precision matrix Q
- ▶ `inla.qreordering`—Computes a fill-in reducing reordering.
- ▶ `inla.qsolve`—Solve a linear system
- ▶ `inla.qinv(Q)`—Calculates the elements of the inverse corresponding to the non-zero elements of Q . Needed for computing derivatives of Gaussian likelihoods.

A FEW MORE USEFUL COMMANDS

- ▶ `inla.spde.precision(spde, tau=..., kappa2=...)`—computes precision matrix. Less straightforward for `spde2` models
- ▶ `inla.qsample(n, Q, ...)`—Computes a sample and various other quantities needed for MCMC for precision matrix Q
- ▶ `inla.qreordering`—Computes a fill-in reducing reordering.
- ▶ `inla.qsolve`—Solve a linear system
- ▶ `inla.qinv(Q)`—Calculates the elements of the inverse corresponding to the non-zero elements of Q . Needed for computing derivatives of Gaussian likelihoods.

USEFUL FEATURES

- ▶ replicate and group
- ▶ more than one “family”
- ▶ copy
- ▶ linear combinations
- ▶ A matrix in the linear predictor
- ▶ values
- ▶ remote computing

FEATURE: REPLICATE

“replicate” generates iid replicates from the same $f()$ -model with the same hyperparameters.

If $x \mid \theta \sim \text{AR}(1)$, then `nrep=3`, makes

$$\mathbf{x} = (x_1, x_2, x_3)$$

with mutually independent x_i 's from $\text{AR}(1)$ with the same θ
Arguments

```
f(..., replicate = r [, nrep = nr ])
```

where replicate are integers 1, 2, ..., etc

EXAMPLE: REPLICATE

```
n=100
# x1 and x2 are the same ar1 process with
# different intercepts
x1 = arima.sim(n, model=list(ar=0.9)) + 1
x2 = arima.sim(n, model=list(ar=0.9)) - 1

y1 = rpois(n,exp(x1)) #poisson obs
y2 = rpois(n,exp(x2))
y = c(y1,y2)

i = rep(1:n,2) #indexing!
r = rep(1:2,each=n) #replicate no.
intercept = as.factor(r) #2 intercepts

formula = y ~ f(i, model="ar1", replicate=r) + intercept -1
result = inla(formula, family = "poisson",
              data = data.frame(y, i, r, intercept))
```

NAS IN INLA

What do NAs do?

- ▶ In the covariates, an NA is treated as a zero.
- ▶ In the random effect, NAs indicate that the effect does not contribute to the likelihood
- ▶ In the data, an NA indicates a location for prediction.

FEATURE: COPY

This feature fixes a limitation in the formula-formulation of the model

The model

$$\text{formula} = y \sim f(i, \dots) + \dots$$

Only allow ONE element from each sub-model, to contribute to the linear predictor for each observation.

Sometimes/Often this is not sufficient.

FEATURE: COPY

This feature fixes a limitation in the formula-formulation of the model

The model

$$\text{formula} = y \sim f(i, \dots) + \dots$$

Only allow ONE element from each sub-model, to contribute to the linear predictor for each observation.

Sometimes/Often this is not sufficient.

FEATURE: COPY

This feature fixes a limitation in the formula-formulation of the model

The model

$$\text{formula} = y \sim f(i, \dots) + \dots$$

Only allow ONE element from each sub-model, to contribute to the linear predictor for each observation.

Sometimes/Often this is not sufficient.

FEATURE: COPY

Suppose

$$\eta_i = u_i + u_{i+1} + \dots$$

Then we can code this as

```
formula = y ~ f(i, model="iid") +
          f(i.plus, copy="i") + ...
```

- ▶ The copy-feature, creates internally an additional sub-model which is ϵ -close to the target
- ▶ Many copies allowed, and copies of copies

FEATURE: COPY

Suppose

$$\eta_i = u_i + \beta u_{i+1} + \dots$$

Then we can code this as

```
formula = y ~ f(i, model="iid") +
  f(i.plus, copy="i",
    hyper = list(
      beta = list(fixed = FALSE))) + ...
```

FEATURE: COPY

Suppose that

$$\eta_i = a_i + b_i z_i + \dots$$

where

$$(a_i, b_i) \stackrel{\text{iid}}{\sim} \mathcal{N}_2(\mathbf{0}, \Sigma)$$

```
library(mvtnorm)
n = 100
Sigma = matrix(c(1, 0.8, 0.8, 1), 2, 2)
z = runif(n)
ab = rmvnorm(n, sigma = Sigma)
a = ab[, 1]
b = ab[, 2]
eta = a + b * z

y = eta + rnorm(n, sd=0.1)
i = 1:n
j = 1:n + n
formula = y ~ f(i, model="iid2d", n = 2*n) +
           f(j, z, copy="i") - 1
r = inla(formula, data = data.frame(y, i, j))
```

MULTIPLE LIKELIHOODS

In many situations, you need to combine data from different sources and need to be able to handle multiple likelihoods.

Examples:

- ▶ Joint modelling of longitudinal and event time data (Guo and Carlin, 2004)
- ▶ Preferential sampling (Diggle et al, 2010)
- ▶ “Marked” point processes
- ▶ Animal breeding modelling with multiple traits
- ▶ Combining data from multiple experiments

MULTIPLE LIKELIHOODS

In many situations, you need to combine data from different sources and need to be able to handle multiple likelihoods.

Examples:

- ▶ Joint modelling of longitudinal and event time data (Guo and Carlin, 2004)
- ▶ Preferential sampling (Diggle et al, 2010)
- ▶ “Marked” point processes
- ▶ Animal breeding modelling with multiple traits
- ▶ Combining data from multiple experiments

HOW TO DO THIS IN INLA

- ▶ Make response y a *matrix* rather than a vector.
 - > `Y = matrix(NA, N, 2)`
 - > `Y[1:n, 1] = y[1:n]`
 - > `Y[1:n + n, 2] = y[(n + 1):(2 * n)]`
- ▶ NAs are used to select components in the formula
 - > `cov1 = c(cov, rep(NA, n))`

```
n = 100
```

```
x1 = runif(n)
```

```
eta1 = 1 + x1
```

```
y1 = rbinom(n, size = 1, prob = exp(eta1)/(1+exp(eta1))) #binom
```

```
x2 = runif(n)
```

```
eta2 = 1 + x2
```

```
y2 = rpois(n, exp(eta2)) #poisson
```

```
Y = matrix(NA, 2*n, 2) # need the response variable as matrix
```

```
Y[1:n, 1] = y1 # binomial data
```

```
Y[1:n + n, 2] = y2 # poisson data
```

```
Ntrials = c(rep(1,n), rep(NA, n)) # required only for binomial
```

```
xx = c(x1, x2)
```

```
formula = Y ~ 1 + xx
```

```
result = inla(formula, data = list(Y = Y, xx = xx),
```

```
family = c("binomial", "poisson"),
```

```
Ntrials = Ntrials)
```

BACK TO COVARIATES

Consider a model with a linear predictor that looks like

$$\eta_i = \dots + \beta c(s_i) + \dots$$

where c is an unknown spatial covariate.

- ▶ c_i is unknown, but we have some measurements $\{c'_j\}$ at points $\{s'_j\}$
- ▶ We can model the true covariate field as above

$$\begin{aligned}c'_j | c(\cdot) &= c(s'_j) + \epsilon_j \\ c(\cdot) &\sim \text{SPDE model}\end{aligned}$$

JOINT MODELLING OF THE COVARIATE

We can then fit these models *at the same time!*

Likelihood:

$$y_i | \eta_i \sim \text{Any INLA likelihood with latent field } \eta$$

$$c'_j | c(\cdot) \sim N(\xi_j, \tau_c^{-1})$$

Latent field:

$$\eta_i = \dots + \beta c(s_i) + \dots$$

$$\xi_j = c(s'_j)$$

- ▶ We have *two likelihoods* (data and covariate)
- ▶ We use the covariate field $c(s)$ twice \rightarrow *copy*

SETTING UP THE LIKELIHOOD

We begin by putting the observations and the observed covariates *together as data*

```
> Y = matrix(NA, N, 2)
> Y[1:n, 1] = y
> Y[(n+1):(2*n), 2] = obs_covariate
```

SETTING UP THE FORMULA

We need to set up the formula carefully to separate out the two things. The trick is NAs in indices

```
> covariate_first_lik = c(1:spde$n.spde,
                          rep(NA, spde$n.spde))
> covariate_second_lik = c(rep(NA, spde$n.spde),
                           1:spde$n.spde)
```

The formula is then

```
> formula = Y ~ ... + f(covariate_first_lik, model=spde)
  + f(covariate_second_lik,
      copy=covariate_first_lik) + ...
```

THE INLA CALL

Finally, we need to make an `inla` call for this model.

```
> result = inla(formula, family = c("_____", "gaussian"),
  data = list(Y=Y,
  covariate_first_lik=covariate_first_lik,
  covariate_second_lik=covariate_second_lik),
  verbose=TRUE)
```

where `_____` is the data likelihood.

THIS MODEL IN PRACTICE

- ▶ Joint modelling the covariate adds 3 hyperparameters (range, precision, noise precision)
- ▶ This can be done any type of data (eg point patterns)
- ▶ If there is *misalignment*, it can get tricky
- ▶ In this case, you need A-matrices

ORGANISING DATA, LATENT FIELDS AND A MATRICES

Real life is hard!

- ▶ In complicated models, we will have multiple sources of data occurring in different places with different likelihood.
- ▶ The latent field may also be composed of sections defined at different resolutions (grid for a spatial covariate, mesh for random field, etc).
- ▶ So we need a function that takes these components and chains them together in a way that makes sense.
- ▶ (You can “roll your own” here, but I *really* don't recommend it!)

We are rescued by `inla.stack`!

THE `INLA.STACK` CALL

```
stack = inla.stack(data = list(...),  
                  A = list(...),  
                  effects = list(...),  
                  tag = NULL, ...)
```

- ▶ The trick here is lists!
- ▶ The first element of the `effects` list is mapped to the first element of the `data` list by the first element for the `A` list.
- ▶ Slightly more tricky when there are replicates and grouping (time!)
- ▶ The functions `inla.stack.data(stack)` and `inla.stack.A(stack)` are used to extract the `data.frame` and the `A-matrix` for use in the `inla` call.

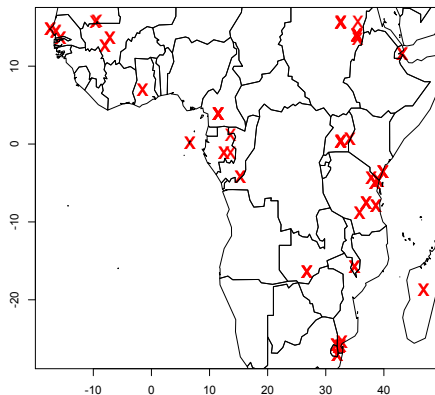
AN UNFORTUNATE FACT

At risk of disappointing you...

Just because you have data and a question, doesn't mean that the data can answer that question!

- ▶ The *best* statistics infers the answer to a question from data *specifically and carefully* collected to answer that question
- ▶ This is obviously not always possible, but we should do our best!
- ▶ For easy problems (differences of means, ANOVAs etc), there are well-known ways to do this
- ▶ In this session, we will have a look at some simple (and some practical) aspects of spatial experimental design

THE BAD NEWS: UNDESIGNED SPATIAL DATA MAY NOT ANSWER THE QUESTION

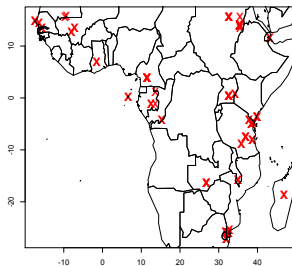


Question: Can we build a spatial map of sero-prevalence of a certain strain of malaria throughout Africa.

Answer: No.

WHAT WENT WRONG

- ▶ Data: $(n_{\text{test}}, n_{\text{present}})$
- ▶ Model: Binomial (low information!)
- ▶ Sampling locations are far apart
- ▶ Essentially uncorrelated!
- ▶ Low power, high uncertainty.



HOW THIS MANIFESTED

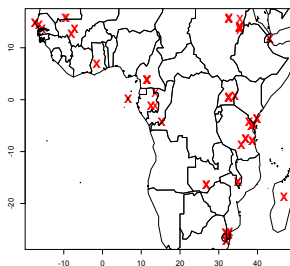
The Folk Theorem

If your computation breaks, the problem is usually your model.

- ▶ INLA assumes that there is enough information in your model to resolve all of the parameters
- ▶ If there isn't, it can break!
- ▶ That's what happened here!

WHAT DOES A GOOD SPATIAL DESIGN LOOK LIKE?

- ▶ Sampling locations cover region of interest (**needed for prediction**)
- ▶ Sampling locations are close enough together that there is correlation (**hard to know beforehand**)
- ▶ Sampling locations are clustered (**needed for parameter estimation**)



THIS CAN BE HARD!

Partial answer:— **Sequential design**

- ▶ Begin with an initial set of sampling locations
- ▶ Compute the posterior
- ▶ Add a new location in the **best** un-sampled location
- ▶ Best = “lowest variance”, “locally lowest variance”, “most valuable-of-information”

NB: The over-all design here is preferential!

Beyond spatial models

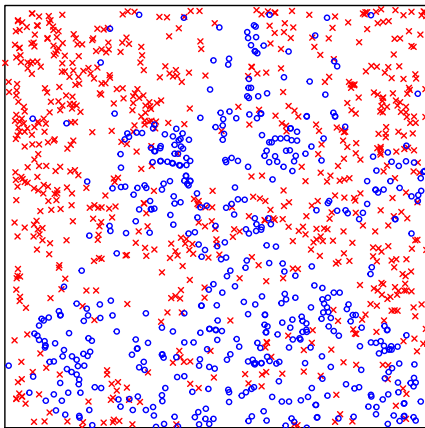
MULTISPECIES POINT PATTERNS

Think about trees

- ▶ Many species appear together
- ▶ We don't really think that these patterns are independent
- ▶ We can fit bivariate patterns and take a look at the correlation

MAPLE AND HICKORY

Hickories (x) and Maples (o)



FRAMEWORK

- ▶ Assume a **set** of spatial point patterns

$$\mathbf{x}_1, \dots, \mathbf{x}_T,$$

observed within bounded regions $\Omega_t \in \mathbb{R}^2$.

- ▶ Each pattern

$$\mathbf{x}_t = \{x_{t1}, \dots, x_{tn_t}\}.$$

is regarded as a realisation from a random spatial point process \mathbf{X}_t , where n_t is the number of points.

THE LOG-GAUSSIAN COX PROCESS

- ▶ Define random intensities

$$\Lambda_t(s) = \exp\{\eta_t(s)\}$$

where $\{\eta_t(s) : s \in \Omega_t \in \mathbb{R}^2\}$ is a Gaussian random field.

- ▶ Conditional on the random intensities

$$\mathbf{X}_t \mid \Lambda_t(s) \sim \text{Poisson}(\exp(\eta_t(s)))$$

THE LATTICE-BASED APPROACH

- ▶ Define

s_{ti} : Grid cell i in Ω_t

y_{ti} : Number of points in grid cell s_{ti} for pattern \mathbf{x}_t

η_{ti} : Representative value of the Gaussian field for pattern \mathbf{x}_t
in cell s_{ti} .

- ▶ Point patterns are assumed conditionally independent

$$y_{ti} | \eta_t(s_{ti}) \sim \text{Poisson}(|s_{ti}| \exp(\eta_t(s_{ti}))).$$

- ▶ **Special case:** $\Omega_t = \Omega$ for all t such that $s_{ti} = s_i$.

FITTING A JOINT MODEL

- ▶ Each point pattern might be too small to make sensible model.
- ▶ Fit **joint** model to several point patterns:

$$\eta_{ti} = \alpha_t + \sum_{j=1}^{n_\beta} \beta_j z_{tji} + \sum_{k=1}^{n_f} f_k(c_{tki}) + \epsilon_{ti}, \quad t = 1, \dots, T.$$

ESTIMATION BASED ON SEVERAL POINT PATTERNS

Use all of the point patterns to:

- ▶ Estimate fixed linear effects of covariates, that is the parameters $\beta_1, \dots, \beta_{n_\beta}$.
- ▶ Estimate non-linear random effects of covariates, that is the underlying smooth functions f_1, \dots, f_{n_f} .
- ▶ Account for dependencies/variation between different patterns.

In R-INLA:

The joint model is fitted just **stacking** the responses and covariate terms in vectors.

ESTIMATION BASED ON SEVERAL POINT PATTERNS

Use all of the point patterns to:

- ▶ Estimate fixed linear effects of covariates, that is the parameters $\beta_1, \dots, \beta_{n_\beta}$.
- ▶ Estimate non-linear random effects of covariates, that is the underlying smooth functions f_1, \dots, f_{n_f} .
- ▶ Account for dependencies/variation between different patterns.

In R-INLA:

The joint model is fitted just **stacking** the responses and covariate terms in vectors.

UNDERSTANDING MARKED POINT PATTERNS

distinguish

- a) different types of marks
- b) different roles of marks

a) is obvious

- ▶ qualitative marks (species, age-groups, infected vs. non-infected...)
- ▶ quantitative marks (size, age, chemical properties...)

b) is harder...

MARKED POINT PATTERNS

different roles of marks

- (i) models of the *pattern* that take the marks into account:
aim is to use marks to “explain” the pattern
- (ii) models of the *marks* in a point pattern:
aim is to model the marks – often along with the pattern (!)

UNDERSTANDING MARKED POINT PATTERNS

for qualitative marks

(i) models of the *pattern* that take the marks into account:

“superposition”

- ▶ consider several (sub-)patterns formed by different types of points
- ▶ different subpatterns have been generated by separate (but not necessarily independent) mechanisms

example: pattern formed by a multi-species plant community

(ii) models of the *marks* in a point pattern

“labelling”

- ▶ consider a single pattern with different (qualitative) characteristics
- ▶ some underlying mechanisms have lead to different qualitative properties of the points

example: pattern formed by a single species but individuals have been affected or not affected by a disease

UNDERSTANDING MARKED POINT PATTERNS

for quantitative marks

- (i) models of the *pattern* that take the marks into account:
- (ii) models of the *marks* in a point pattern

MODELS WITH GROUPS

Earlier we talked about replicated random effects, where we observed i.i.d. draws from the random effect distribution.

- ▶ Point patterns observed at different plots
- ▶ Annual rainfall observed during different years

But is this enough?

NO IT ISN'T!

In a lot of applications, the assumptions that the repeated random effects are *independent* is very restrictive.

- ▶ Monthly / daily rainfall data
- ▶ The results of nearby plots could be correlated

INLA provides the concept of a “group” that allows more complicated dependence structures

GROUP DEPENDENCE

Grouped random effects work as follows

- ▶ There is a *within group* correlation structure
 - ▶ Any INLA latent model (iid, ar1, bym, spde etc)
- ▶ There is also a *between group* correlation model
 - ▶ Not every model: "exchangeable" "ar1" "ar" "rw1" "rw2" "besag"

If $x_{g,i}$ is the i th element in group g , then

$$\text{Cov}(x_{g_1, i_1}, x_{g_2, i_2}) = (\text{cov between groups } g_1 \text{ and } g_2) \\ \times (\text{cov between elements } i_1 \text{ and } i_2)$$

THE KRONECKER STRUCTURE

Grouped models are a special case of “Kronecker models”

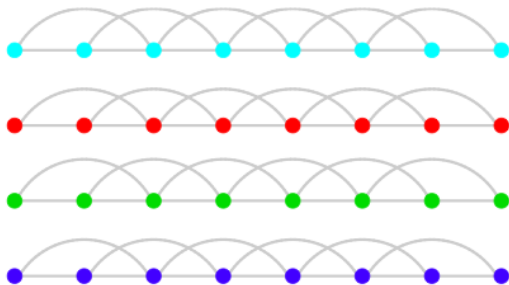
- ▶ These models have covariance matrices of the form $\Sigma_{\text{between group}} \otimes \Sigma_{\text{within group}}$
- ▶ We are working to implement the general structure (so you can group any models in INLA together)
- ▶ We’re going to look through some examples...

CORRELATED RANDOM EFFECTS

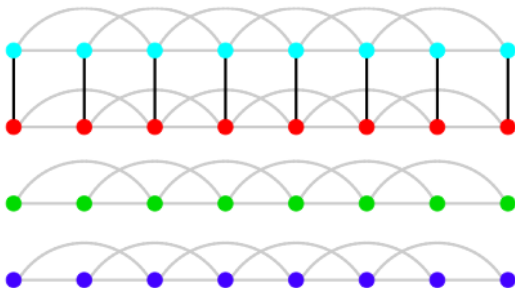
The simplest group model in INLA is the `exchangeable` model

- ▶ “Uniform correlation matrix”
- ▶ $\text{Corr}(\text{group } i, \text{group } j) = \rho, -1 < \rho < 1$
- ▶ This basically says that all of the groups are correlated in the same way
- ▶ This is all you need for *two* correlated effects
- ▶ Allows for some dependence in other cases.

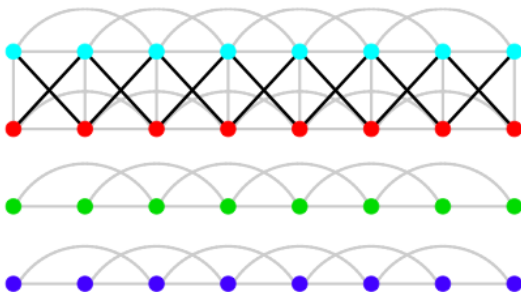
GRAPH FOR CORRELATED RW2



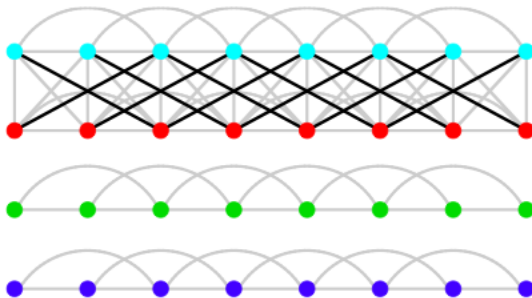
GRAPH FOR CORRELATED RW2



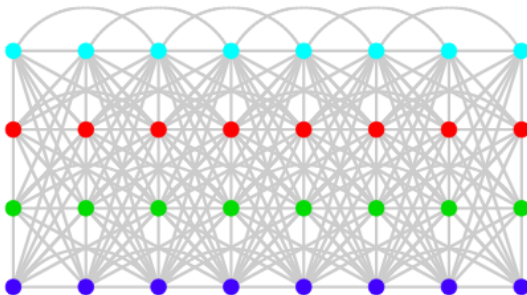
GRAPH FOR CORRELATED RW2



GRAPH FOR CORRELATED RW2



GRAPH FOR CORRELATED RW2



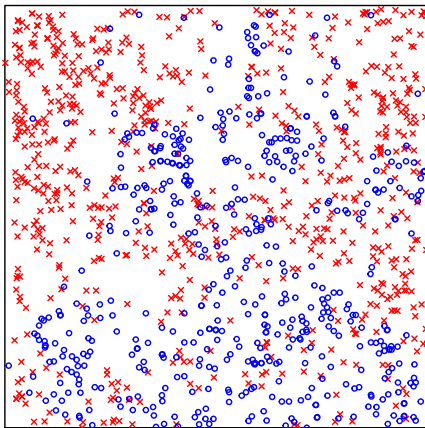
MULTISPECIES POINT PATTERNS

Think about trees

- ▶ Many species appear together
- ▶ We don't really think that these patterns are independent
- ▶ We can fit bivariate patterns and take a look at the correlation

MAPLE AND HICKORY

Hickories (x) and Maples (o)



THE LINEAR MODEL OF CO-REGIONALISATION (LMC)

The easiest way of modelling this is the LMC, which says

- ▶ Fit a common random effect for the two species
- ▶ For one species, add an independent random effect to “mop up” the extra structure

$$\eta_{\text{maple}} = (\text{common effect})$$

$$\eta_{\text{hickory}} = \beta(\text{common effect}) + (\text{extra hickory effect})$$

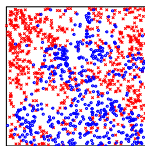
LMC IN INLA

```
#Make indices
common_maple = c(1:n, rep(NA, n))
common_hickory = c(rep(NA, n), 1:n)
extra_hickory = c(rep(NA, n), 1:n)

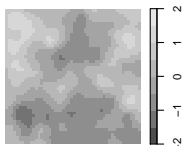
# Make formula

formula = y ~ ... + f(common_maple, model="rw2d")
          + f(common_hickory, copy="common_maple",
              hyper = list(beta=list(fixed=FALSE)))
          + f(extra_hickory, model="rw2")
```

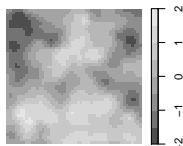
RESULTS



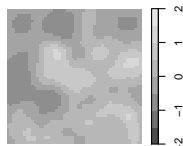
a)



b)



c)



d)

(b):

posterior mean for hickories, (c) post. mean for maples, (d) excess effect

THE GROUPED VERSION

The other option is to model the random effect for each species separately and let them be correlated.

- ▶ Advantage: A single parameter (ρ) that tells you about correlation
- ▶ Disadvantage: You don't get the pretty picture

```
#indices
effect = c(1:n,1:n)
group = rep(c(1,2), each=n)

#formula
formula = y~ ... + f(effect,model="rw2d",group=group,
                      control.group = list(model="exchangeable"))
```

RESULTS WITH SPDE MODEL

	range hickory	range maple	correlation	DIC
est	64	67	-0.69	-
group	70 (48, 98)	-	-0.63 (-0.77, -0.46)	5568.5
LMC	70 (42, 109)	110 (72, 178)	-0.79 (-0.95, -0.53)	5566.3

- ▶ Fitted using SPDE models (not `rw2d`)
- ▶ This allows for estimation of the correlation range for each parameter
- ▶ We see strong negative correlation
- ▶ In this case, the LMC fits better
- ▶ The better fit is attributed to the components having different correlation ranges for different species

SPATIOTEMPORAL MODELS

- ▶ Data frequently has a temporal component
- ▶ Easy fixes:
 - ▶ Treat them as independent (`replicate`)
 - ▶ Add a temporal random effect

$$\eta = \dots + f(\text{space}) + f(\text{time})$$

- ▶ Harder fix: Try to make space time models

SPATIOTEMPORAL MODELS

- ▶ Data frequently has a temporal component
- ▶ Easy fixes:
 - ▶ Treat them as independent (`replicate`)
 - ▶ Add a temporal random effect

$$\eta = \dots + f(\text{space}) + f(\text{time})$$

- ▶ Harder fix: Try to make space time models

There are two types of space-time models:

- ▶ Separable models:
 - ▶ Correlation between two points in space-time =
Corr in space \times Corr in time
 - ▶ This is easy to do and works well
 - ▶ Doesn't capture "spreading fronts"
- ▶ Non-separable models:
 - ▶ Anything that isn't separable!
 - ▶ Much more flexible
 - ▶ But harder to fit...
 - ▶ Not in INLA (yet...)

We're going to fit a separable model

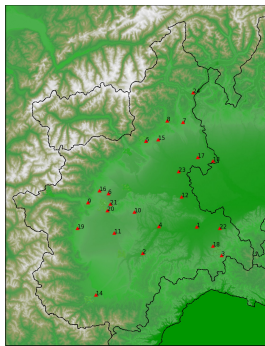
There are two types of space-time models:

- ▶ Separable models:
 - ▶ Correlation between two points in space-time =
Corr in space \times Corr in time
 - ▶ This is easy to do and works well
 - ▶ Doesn't capture "spreading fronts"
- ▶ Non-separable models:
 - ▶ Anything that isn't separable!
 - ▶ Much more flexible
 - ▶ But harder to fit...
 - ▶ Not in INLA (yet...)

We're going to fit a separable model

PM-10 CONCENTRATION IN PIEMONTE, ITALY

Everything that I'm talking about today is described in Cameletti *et al.* (2011) on `r-inla.org`. (It's a really good paper!)



PM10 concentration:

- ▶ 24 monitoring stations
- ▶ Daily data from 10/05 to 03/06

COVARIATES

- ▶ Daily mean wind speed ($WS, m/s$)
- ▶ Daily maximum mixing height ($HMIX, m$)
- ▶ Daily precipitation (P, mm)
- ▶ Daily mean temperature ($TEMP, K^\circ$)
- ▶ Daily emissions ($EMI, g/s$)
- ▶ Altitude (A, m) Coordinates ($UTMX$ and $UTMY, km$).

THE LATENT FIELD (STATE EQUATION)

We use an AR(1) structure

$$\boldsymbol{\xi}_t = a\boldsymbol{\xi}_{t-1} + \boldsymbol{\omega}_t,$$

where $a \in (0, 1)$ is a constant and

$$\boldsymbol{\omega}_t \stackrel{\text{i.i.d.}}{\sim} N(\mathbf{0}, \mathbf{Q}^{-1}),$$

is taken from a spatial SPDE model.

THE MEASUREMENT EQUATION

We take the measurement equation to be

$$\mathbf{y}_t = \mathbf{X}_t\boldsymbol{\beta} + \mathbf{A}\boldsymbol{\xi}_t + \boldsymbol{\epsilon}_t,$$

where \mathbf{X}_t is a matrix of covariates, $\boldsymbol{\beta}$ are the weights, \mathbf{A} picks out the appropriate values of $\boldsymbol{\xi}_t$ and

$$\boldsymbol{\epsilon}_t \stackrel{\text{i.i.d.}}{\sim} N(0, \sigma^2\mathbf{I}).$$

STEP 1: MAKE THE MESH

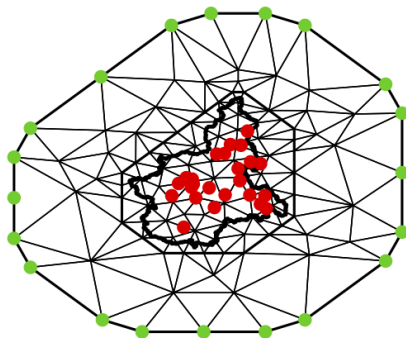
```
mesh =
  inla.mesh.2d(points =NULL,
              points.domain=borders,
              offset=c(10, 140),
              max.edge=c(40,1000),
              min.angle=21,
              cutoff=0,
              plot.delay=NULL
            )

boundary = inla.mesh.boundary(mesh)[[1]]

nmesh = mesh$n
#select (the rows of) the position of the stations
mesh.idx = 1:nmesh
```


A MESH

Constrained refined Delaunay triangulation



mesh

STEP 2: MAKE THE LATENT MODEL

In order to construct a kronecker product model in INLA, we use the (experimental) `group` feature

```
spde = inla.create.spde(mesh,model="matern")
```

```
formula = y ~ WS + HMIX + ...
          + intercept + f(field, model=spde
                          group =time,
                          control.group=list(model="ar
                                              )
```

- ▶ This tells INLA that the observations are grouped in a certain way.
- ▶ `control.group` contains the grouping model (only `ar1` and `exchangable`) as well as their prior specifications.
- ▶ NB: `intercept!`

STEP 3: MAKE AN A MATRIX

There are two ways to construct the A matrix: A for loop or an inbuilt function.

```
LocationMatrix = inla.spde.make.A(mesh = mesh,  
    loc =dataLoc, group=time, n.group=nT)
```

This locates the data points in each `group=time` level and stacks the corresponding local A matrices in an appropriate way.

STEP 4: ORGANISING THE DATA

We have a problem: we have the covariates at the data points, but the latent field only defined their through the A matrix.

We need to make sure that A only applies to the random effect.

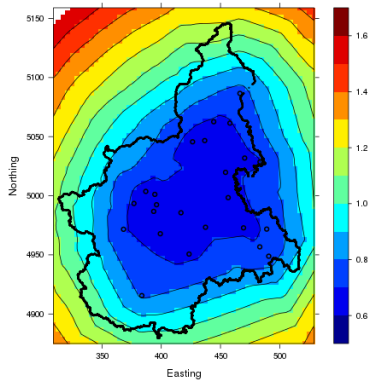
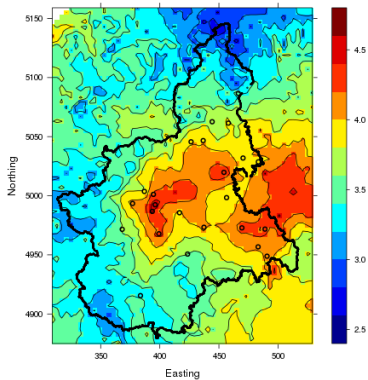
Solution: Padding by NAs.

STEP 5: ORGANISING THE DATA WITH `INLA.STACK`

We can now put everything together.

```
stack = inla.stack( data = dat,  
  A = list(1, LocationMatrix),  
  effects = list( list(WS = cov$WS,...),  
    c(inla.spde.make.index("mesh.idx",n.field=nme  
      n.group=T),  
      list(intercept=rep(1,mesh$n*nT)))  
  )  
)  
result = inla(formula, family = "gaussian",  
  data=inla.stack.data(stack).  
  control.predictor = list(A=inla.stack.A(stack)),  
  verbose=TRUE)
```

POSTERIOR MEAN PM10 CONCENTRATION FOR 30/01/2006 (LOG SCALE)



BUT DID WE ANSWER THE QUESTION?

- ▶ The question was not *fit a space-time surface*
- ▶ The limit value fixed by the European directive 2008/50/EC for PM_{10} is $50\mu g/m^3$. The daily mean concentration cannot exceed this value more than 35 days in a year.
- ▶ The question was “Does the PM-10 concentration exceed the EU-mandated maximum levels?”
- ▶ So can we get the answer to this question?

MULTIPLE COMPARISONS

- ▶ The easiest thing is to compute, for each point, the probability of exceeding the threshold
- ▶ We can do that with `inla.pmarginal`
- ▶ But this is bad...
- ▶ We want areas where *everything* exceeded the level... multiple comparisons
- ▶ These sets are called *excursion sets*

EXCURSIONS AND INLA

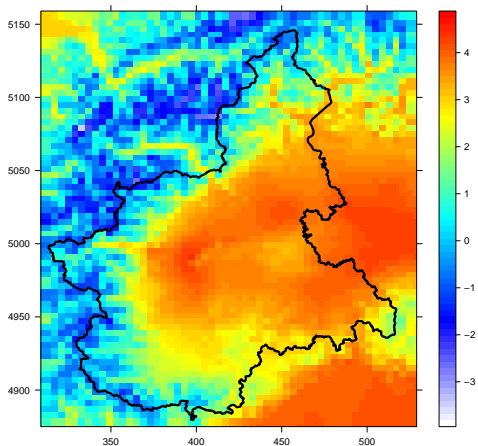
David Bolin (Chalmers) wrote an R package called `excursions` that works with INLA to solve this problem.

- ▶ It's pretty easy to use

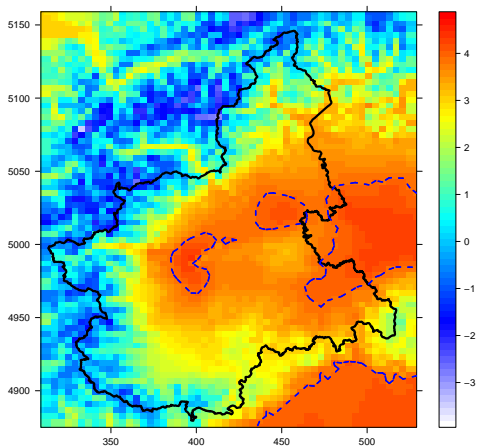
```
excursions.inla(result.inla, ind=indices, alpha=0.99,  
                u=0, method='QC', type='>')
```

- ▶ `result.inla` is the output from INLA
- ▶ You need to run INLA with the option `control.compute=list(config=TRUE)`
- ▶ `ind=indices` tells it which indices of the model you're interested in
- ▶ `u` and `alpha` are the level and the confidence
- ▶ `type=">"` says you want the set of things above level `u`
- ▶ `method='QC'` tells the function how to deal with the non-Gaussianity

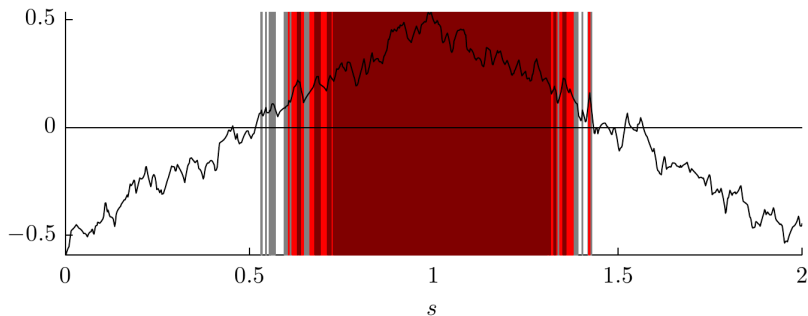
PM₁₀ IN PIEMONTE: WHERE IS PM₁₀ > 50?



PM₁₀ IN PIEMONTE: WHERE IS PM₁₀ > 50? UNCERTAINTY?



EXAMPLE 1: GAUSSIAN PROCESS WITH EXPONENTIAL COVARIANCE

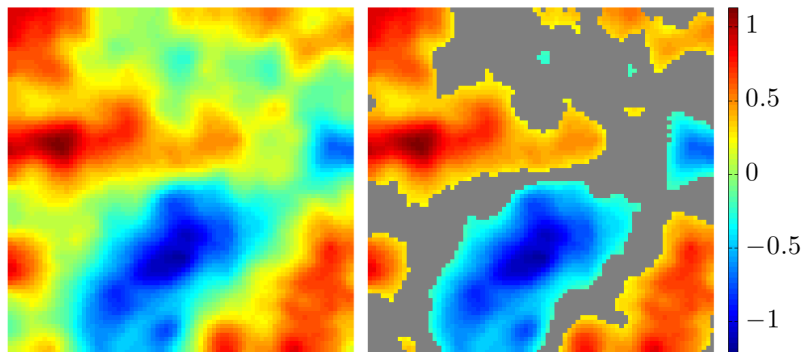


- ▶ Gaussian process with exponential covariance function.
- ▶ The 95% excursion set is shown in red.
- ▶ The grey area contains $\{s : \Pr(x(s) > 0) > 0.95\}$.
- ▶ The dark red set is the Bonferroni lower bound.
- ▶ The black curve is the kriging estimate of $x(s)$.

CONTOURS AND EXCURSIONS

- ▶ A contour curve of a reconstructed field can (almost) be found from the pointwise marginal distributions.
- ▶ But they are uncertain...
- ▶ The *uncertainty* depends on the full joint distribution.
- ▶ A credible contour region is a region where the field transitions from being clearly below, to being clearly above.
- ▶ This is the same problem as the excursion problem

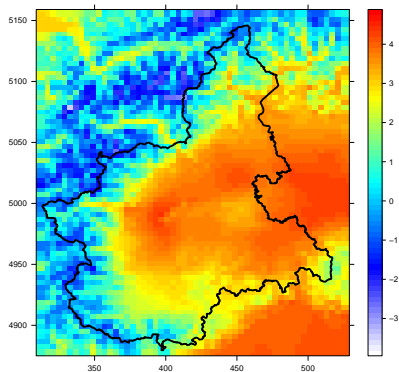
EXAMPLE 2: GAUSSIAN MATÉRN FIELD



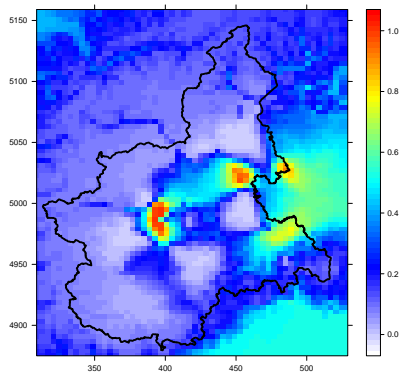
- ▶ Gaussian Matérn field measured under Gaussian noise.
- ▶ Left panel shows the kriging estimate,
- ▶ The grey block on the right is the 95% *contour* for the zero level
- ▶ i.e. The field is, with high probability, equal to zero somewhere in that region.

PM-10: JANUARY 30, 2006

Spatial reconstruction

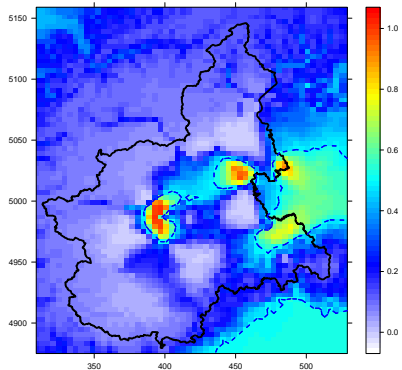
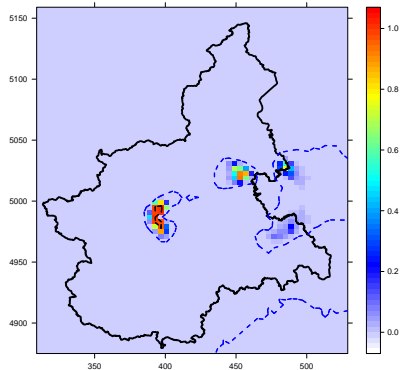


Marginal probabilities

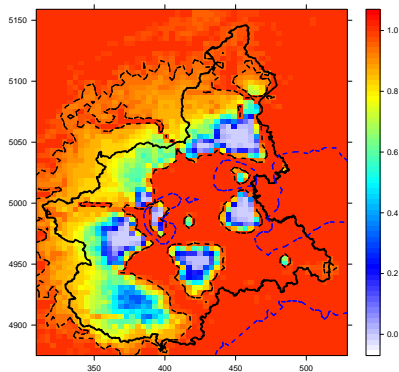
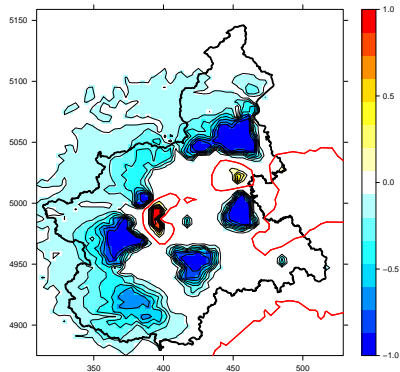


PM-10: JANUARY 30, 2006

Marginal probabilities

 $F_{50}^+(s)$ 

PM-10: JANUARY 30, 2006

Contour function $F_{50}^C(s)$ Signed avoidance $\pm F_{50}(s)$ 

Conclusions

THE KEY LESSONS OF THIS COURSE

- ▶ You should leverage the properties of the class of model you are using to improve the computation
- ▶ Bayesian computation does not need to be slow!
- ▶ MCMC is not the only horse in town (although it is the most flexible)
- ▶ You can buy computational flexibility and scaling by making Markov assumptions
- ▶ Priors are important!

BUC 4 AND ITT4: AN INVITATION TO VISIT BATH THIS SUMMER

BUC4: Bath, UK, 1st – 3rd June 2016

- ▶ ‘New frontiers: advanced modelling in space and time’
- ▶ Presented by Gavin Shaddick (Bath), Dan Simpson (Bath) and Jim Zidek (University of British Columbia)
- ▶ The third in the series on Big Data and Statistics in Environmental Research .

ITT4 (Integrated Think Tank): Bath, UK, 6th – 9th June 2016

- ▶ Industrial partners: AstraZeneca (pharmaceuticals) and the National Health Service (NHS)

Please come!