Bayesian analysis of latent Gaussian models without MCMC

Håvard Rue[1]
Department of Mathematical Sciences
NTNU, Norway

Sept 2007

[1]With S.Martino/N.Chopin

Latent Gaussian models

Latent Gaussian models have often the following hierarchical structure

- Observed data $\mathbf{y}$, $y_i|x_i \sim \pi(y_i|x_i, \boldsymbol{\theta})$
- Latent Gaussian field $\mathbf{x} \sim \mathcal{N}(\cdot, \boldsymbol{\Sigma}(\boldsymbol{\theta}))$
- Hyperparameters $\boldsymbol{\theta}$
  - variability
  - length/strength of dependence
  - parameters in the likelihood

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

Task

Compute from

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

the posterior marginals:

$$\pi(x_i \mid \mathbf{y}), \qquad \text{for some or all } i$$

and/or

$$\pi(\theta_i \mid \mathbf{y}), \qquad \text{for some or all } i$$

Common approach: MCMC

It is common to use MCMC to estimate the marginals, but

- ...but vanilla schemes is not without serious problems
- ...better (block-)schemes are possible, but is often slow due to the Monte-Carlo error
- ...Monte-Carlo error is in most cases much larger than we would like it to be.

Our approach: approximate inference

- Can we compute (approximate) marginals directly without using MCMC?
- YES!
- Gain
  - **Huge speedup & accuracy**
  - **The ability to treat latent Gaussian models properly ;-)**

Main ideas (I)

Main ideas are simple and based on the identity

$$\pi(z) = \frac{\pi(x, z)}{\pi(x|z)} \qquad \text{leading to} \qquad \widetilde{\pi}(z) = \frac{\pi(x, z)}{\widetilde{\pi}(x|z)}$$

When $\widetilde{\pi}(x|z)$ is the Gaussian-approximation, this is the Laplace-approximation.

Main ideas (II)

Construct the approximations to

1. $\pi(\boldsymbol{\theta}|\mathbf{y})$
2. $\pi(x_i|\boldsymbol{\theta}, \mathbf{y})$

then we integrate

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

$$\pi(\theta_j|\mathbf{y}) = \int \pi(\boldsymbol{\theta}|\mathbf{y})\, d\boldsymbol{\theta}_{-j}$$

Characteristic features

- Dimension of the latent Gaussian field, n, is large, $10^2 - 10^5$.
- Dimension of the hyperparameters $\dim(\boldsymbol{\theta})$ is small, $1 - 5$, say.
- Dimension of the data $\dim(\mathbf{y})$ might vary, but is often non-Gaussian.

Examples

- *1D* Smoothing count data, general spline smoothing, semi-parametric regression
- *2D* Disease mapping, log-Gaussian Cox-processes, model-based geostatistics
- *3D* Time-series of images, spatio-temporal models.

Often, the latent Gaussian field has Markov properties, ie its a GMRF.

GMRFs: def

A *Gaussian Markov random field (GMRF)*, $\mathbf{x} = (x_1, \ldots, x_n)^T$, is a normal distributed random vector with additional Markov properties

$$x_i \perp x_j \mid \mathbf{x}_{-ij} \quad \Longleftrightarrow \quad Q_{ij} = 0$$

where $\mathbf{Q}$ is the precision matrix (inverse covariance)

GMRFs: computational properties

- Due to Markov properties $\mathbf{Q}$ is a (very) sparse matrix, often only $\mathcal{O}(n)$ non-zero terms
- "Computing" with GMRFs involves *sparse matrices*
  - Factorising $\mathbf{Q}$ into $\mathbf{L}\mathbf{L}^T$
  - Solving $\mathbf{L}\mathbf{u} = \mathbf{v}$ and $\mathbf{L}^T\mathbf{u} = \mathbf{v}$

  Using numerical methods for sparse (SPD) matrices:

Case	Factorisation cost
Time	$\mathcal{O}(n)$
Spatial	$\mathcal{O}(n^{3/2})$
Time×Space	$\mathcal{O}(n^2)$

The GMRF-approximation

$$
\begin{aligned}
\pi(\mathbf{x} \mid \mathbf{y}) &\propto \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} - \sum_i \exp(x_i)\right) \\
&\approx \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T(\mathbf{Q} + \mathrm{diag}(c_i))(\mathbf{x}-\boldsymbol{\mu})\right) = \widetilde{\pi}(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})
\end{aligned}
$$

Constructed as follows:

- Locate the mode $\mathbf{x}^*$
- Expand to second order

Markov and computational properties are preserved

The Laplace approximation

The *Laplace approximation* for $\pi(\boldsymbol{\theta}|\mathbf{y})$ is

$$\begin{aligned}
\pi(\boldsymbol{\theta} \mid \mathbf{y}) &= \frac{\pi(\mathbf{x}, \theta|\mathbf{y})}{\pi(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})} \qquad (\text{any } \mathbf{x}) \\
&\approx \left. \frac{\pi(\mathbf{x}, \theta|\mathbf{y})}{\widetilde{\pi}(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})} \right|_{\mathbf{x}=\mathbf{x}^*(\boldsymbol{\theta})} = \widetilde{\pi}(\boldsymbol{\theta}|\mathbf{y}) \qquad (1)
\end{aligned}$$

Remarks

The Laplace approximation

$$\widetilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$$

turn out to be accurate: $\mathbf{x}|\mathbf{y},\boldsymbol{\theta}$ appears *almost Gaussian* in most cases, as

- $\mathbf{x}$ is *a priori* Gaussian.
- $\mathbf{y}$ is typically not very informative.
- Observational model is usually 'well-behaved'.

Note: $\widetilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$ itself does *not* look Gaussian. Thus, a Gaussian approximation of $(\boldsymbol{\theta},\mathbf{x})$ will be inaccurate.

Approximating $\pi(x_i|\mathbf{y}, \boldsymbol{\theta})$

This task is more challenging, since

- dimension of $\mathbf{x}$, n is large
- and there are potential n marginals to compute, or at least $\mathcal{O}(n)$.

An obvious simple and fast alternative, is to use the GMRF-approximation

$$\widetilde{\pi}(x_i|\boldsymbol{\theta}, \mathbf{y}) = \mathcal{N}(x_i;\ \mu(\boldsymbol{\theta}), \sigma^2(\boldsymbol{\theta}))$$

Laplace approximation of $\pi(x_i|\boldsymbol{\theta}, \mathbf{y})$

- The Laplace approximation:

$$\widetilde{\pi}(x_i \mid \mathbf{y}, \boldsymbol{\theta}) \approx \left. \frac{\pi(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y})}{\widetilde{\pi}(\mathbf{x}_{-i}|x_i, \mathbf{y}, \boldsymbol{\theta})} \right|_{\mathbf{x}_{-i}=\mathbf{x}^*_{-i}(x_i,\boldsymbol{\theta})}$$

- Again, approximation is very good, as $\mathbf{x}_{-i}|\mathbf{x}_i, \theta$ is 'almost Gaussian',
- but it is expensive. In order to get the n marginals:
  - perform n optimisations, and
  - n factorisations of $n-1 \times n-1$ matrices.

Can be solved.

Simplified Laplace Approximation

An series expansion of the LA for $\pi(x_i|\boldsymbol{\theta}, \mathbf{y})$:

- computational much faster: $\mathcal{O}(n \log n)$ for each i
- correct the Gaussian approximation for error in shift and skewness

$$\log \widetilde{\pi}(x_i|\boldsymbol{\theta}, \mathbf{y}) = -\frac{1}{2}x_i^2 + bx_i + \frac{1}{6}d\ x_i^3 + \cdots$$

- Fit a skew-Normal density

$$2\phi(x)\Phi(ax)$$

- sufficiently accurate for most applications

The integrated nested Laplace approximation (INLA) I

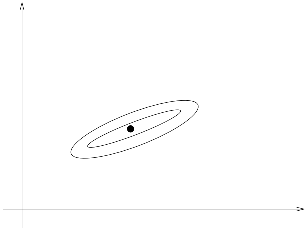
Step I Explore $\widetilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$

- 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 $\widetilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$

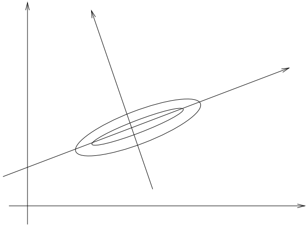
- Locate the mode
- Use the Hessian to construct new variables
- Grid-search
- Can be case-specific



The integrated nested Laplace approximation (INLA) I

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

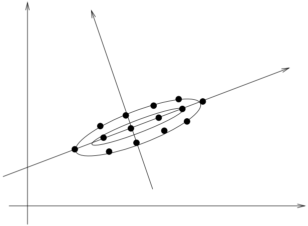
- Locate the mode
- Use the Hessian to construct new variables
- Grid-search
- Can be case-specific



The integrated nested Laplace approximation (INLA) I

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

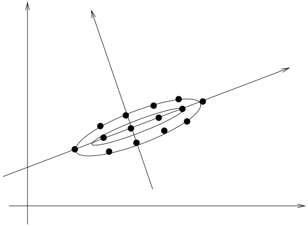
- Locate the mode
- Use the Hessian to construct new variables
- Grid-search
- Can be case-specific



The integrated nested Laplace approximation (INLA) I

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

- 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 $\boldsymbol{\theta}_j$

- For each i, evaluate the Laplace approximation for selected values of x_i
- Build a 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 $\boldsymbol{\theta}_j$

- For each i, sum out $\boldsymbol{\theta}$

$$\widetilde{\pi}(x_i \mid \mathbf{y}) \propto \sum_j \widetilde{\pi}(x_i \mid \mathbf{y}, \boldsymbol{\theta}_j) \times \widetilde{\pi}(\boldsymbol{\theta}_j \mid \mathbf{y})$$

- Build a log-spline corrected Gaussian

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

to represent $\widetilde{\pi}(x_i \mid \mathbf{y})$.

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 2: Estimate the error using Monte Carlo

$$\left\{ \frac{\widetilde{\pi}_u(\boldsymbol{\theta} \mid \mathbf{y})}{\pi(\boldsymbol{\theta} \mid \mathbf{y})} \right\}^{-1} \propto \mathrm{E}_{\widetilde{\pi}_G} \left[\exp \left\{ r(\mathbf{x}; \boldsymbol{\theta}, \mathbf{y}) \right\} \right]$$

where $r()$ is the sum of the log-likelihood minus the second order Taylor expansion.

How can we assess the error in the approximations?

Tool 3: Estimate the "effective" number of parameters

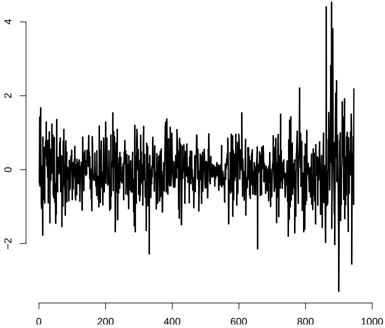
$$p_{\mathrm{D}}(\boldsymbol{\theta}) \approx \mathrm{Trace}\left\{\mathbf{Q}(\boldsymbol{\theta})\,\mathbf{Q}^{\star}(\boldsymbol{\theta})^{-1}\right\},$$

and compare this with the number of observations.

Low ratio is good.

Stochastic Volatility model

Frequently used to analyse the structure of the volatility in financial time series



Log of the daily difference of the pound-dollar exchange rate from October 1st, 1981, to June 28th, 1985.

Stochastic Volatility model

Simple model

$$x_t \mid x_1, \ldots, x_{t-1}, \tau, \phi \sim \mathcal{N}\left(\phi x_{t-1}, 1/\tau\right)$$

where $|\phi| < 1$ to ensure a stationary process.

Observations are taken to be

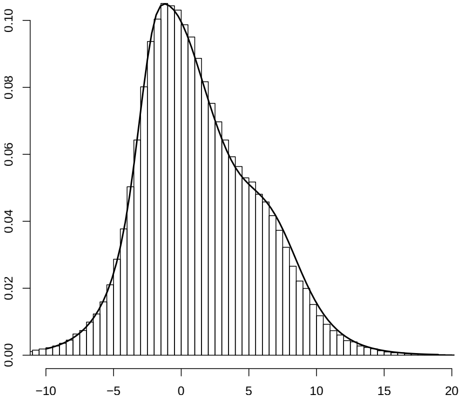
$$y_t \mid x_1, \ldots, x_t, \kappa \sim \mathcal{N}(0, \exp(x_t)/\kappa)$$

possibly Student-t distributed

Results

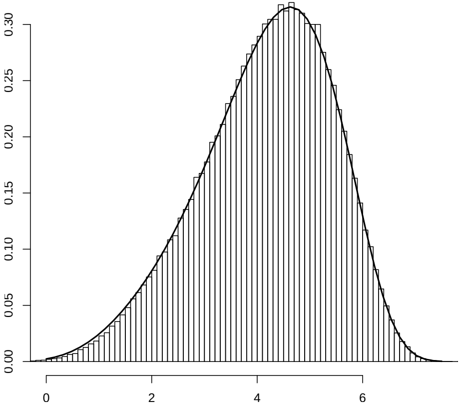
Using just the first 50 data-points only, which makes the problem much harder.

Results



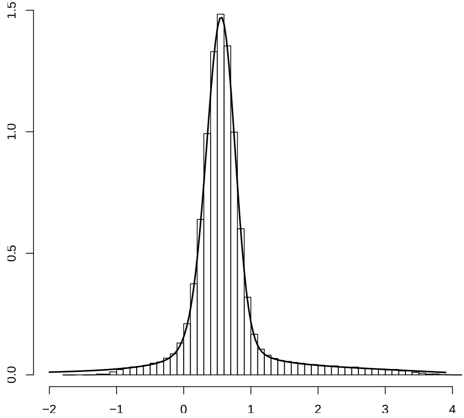
$$\nu = \mathrm{logit}(2\phi - 1)$$

Results



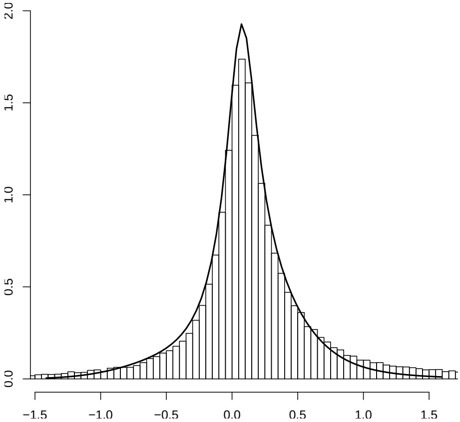
$\log(\kappa_x)$

Results



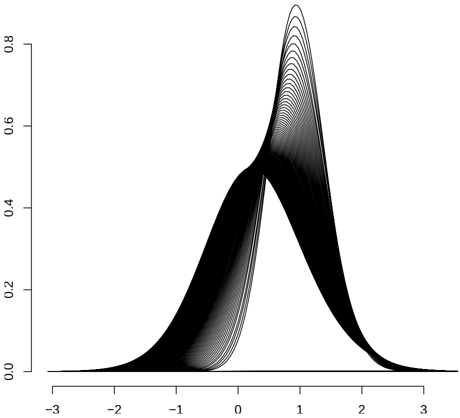
$\log(\kappa_\mathrm{y})$

Results



Node with max KLD

Results



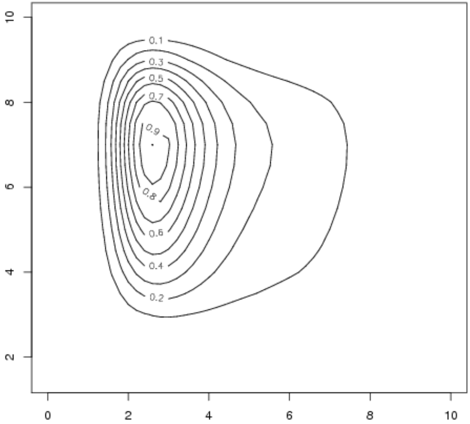
Predictions using the full data-set and Student-t distributed observations with stochastic dof.

Disease mapping: The BYM-model

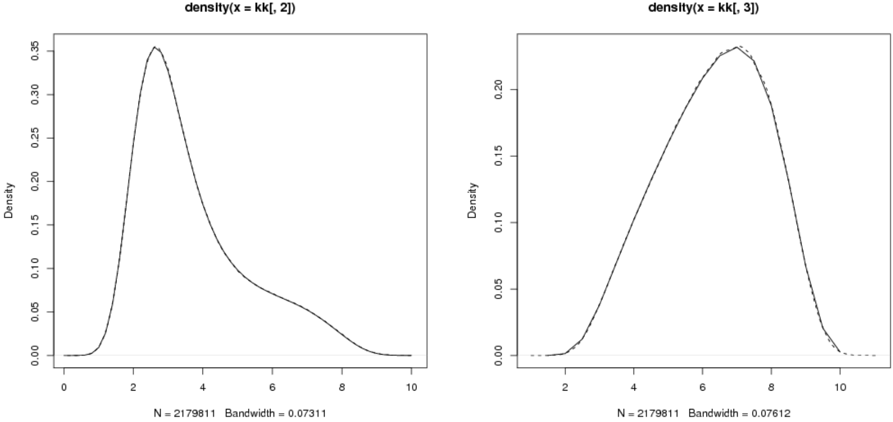
- Data $y_i \sim$ Poisson($E_i exp(\eta_i)$)
- Log-relative risk $\eta_i = u_i + v_i$
- Structured component $\mathbf{u}$
- Unstructured component $\mathbf{v}$
- Log-precisions $\log \kappa_u$ and $\log \kappa_v$

- A hard case: Insulin Dependent Diabetes Mellitus in 366 districts of Sardinia. Few counts.
- $\dim(\boldsymbol{\theta}) = 2$.

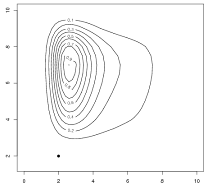
Marginals for $\boldsymbol{\theta}|\mathbf{y}$

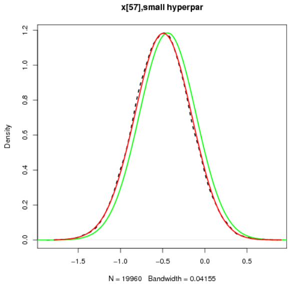


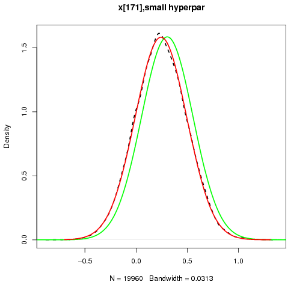
Marginals for $\boldsymbol{\theta}|\mathbf{y}$



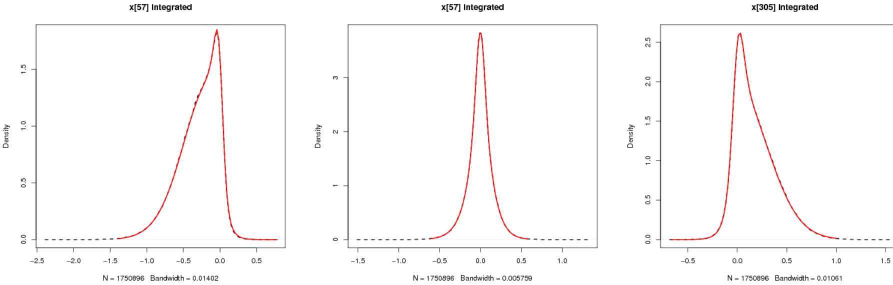
Marginals for $x_i|\boldsymbol{\theta}, \mathbf{y}$







Marginals for $\mathbf{x}_i|\mathbf{y}$



Semi-parametric ecological regression

Semi parametric ecological regression

$$\log(\eta_i) = \mu + u_i + v_i + f(c_i)$$

f is an unknown function of regional covariate c:

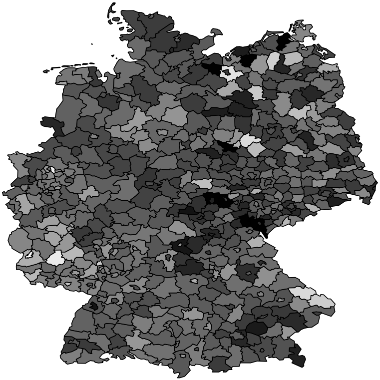
$$\pi(f) \propto \kappa^{(n-2)/2} \exp(-\frac{\kappa}{2} \sum_i (f_{i+1} - 2f_i + f_{i-1})^2)$$

Require $\sum_i u_i = 0$, to separate the spatial vrs the covariate effect.

$$\mathbf{x} = (\mu, \mathbf{u}, \boldsymbol{\eta}, \mathbf{f}) \mid \text{hyperparameters} \sim \text{GMRF} \qquad (2)$$

$\dim(\boldsymbol{\theta}) = 3$

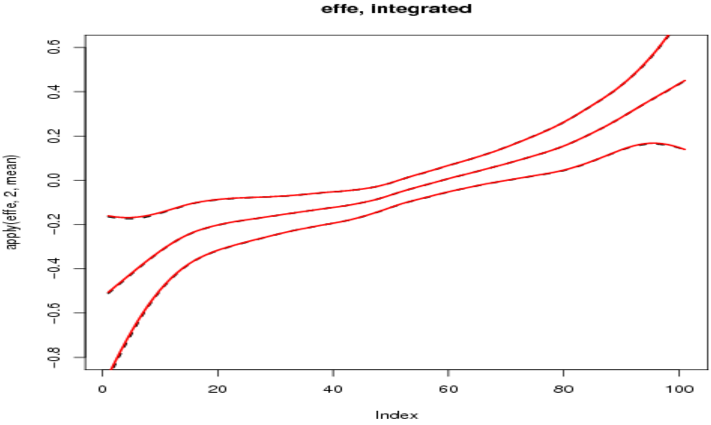
Example: Larynx cancer with smoking covariate



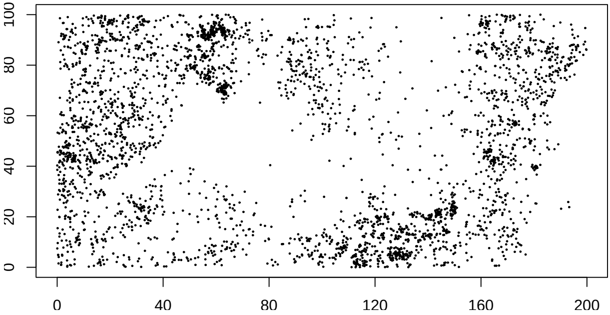
Larynx SMR

Smoking covariate

Example: Larynx cancer with smoking covariate

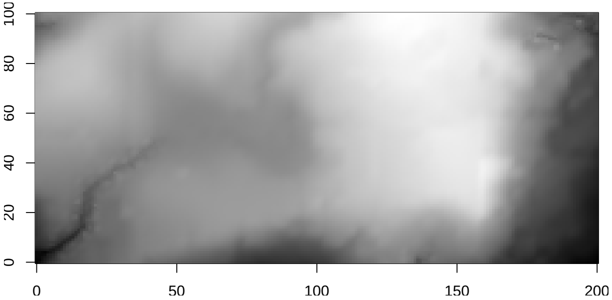


Log-Gaussian Cox process



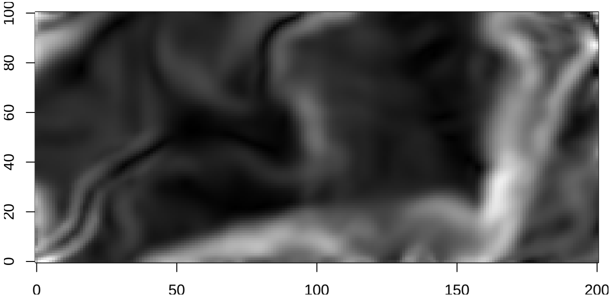
Locations of trees of a particular type: Data comes from a 50-hectare permanent tree plot which was established in 1980 in the tropical moist forest of Barro Colorado Island in Gatun Lake in central Panama.

Log-Gaussian Cox process



Covariate: altitude

Log-Gaussian Cox process



Covariate: norm of gradient

Model

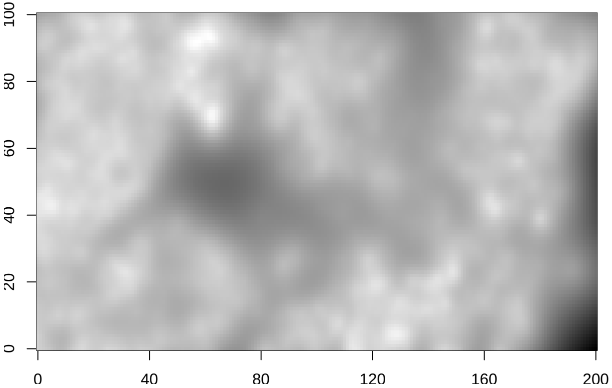
Model for log-density at each "pixel" in a 200×100 lattice

$$\eta_i = \beta_0 + \beta_1 c_{1i} + \beta_2 c_{2i} + u_i + v_i, \qquad \sum_i u_i = 0$$

The spatial term is an IGMRF

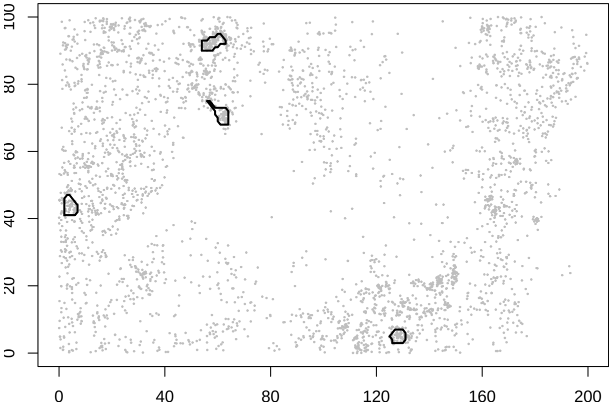
$$\begin{aligned}
\mathrm{E}(u_i \mid \mathbf{u}_{-i}) &= \frac{1}{20}\left(8 \begin{array}{ccccc} \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \bullet & \circ & \circ \\ \circ & \bullet & \circ & \bullet & \circ \\ \circ & \circ & \bullet & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ \end{array} - 2 \begin{array}{ccccc} \circ & \circ & \circ & \circ & \circ \\ \circ & \bullet & \circ & \bullet & \circ \\ \circ & \circ & \circ & \circ & \circ \\ \circ & \bullet & \circ & \bullet & \circ \\ \circ & \circ & \circ & \circ & \circ \end{array} - 1 \begin{array}{ccccc} \circ & \circ & \bullet & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ \\ \bullet & \circ & \circ & \circ & \bullet \\ \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & \bullet & \circ & \circ \end{array}\right) \\
\mathrm{Prec}(u_i \mid \mathbf{u}_{-i}) &= 20\kappa_{\mathbf{u}}
\end{aligned}$$

Results



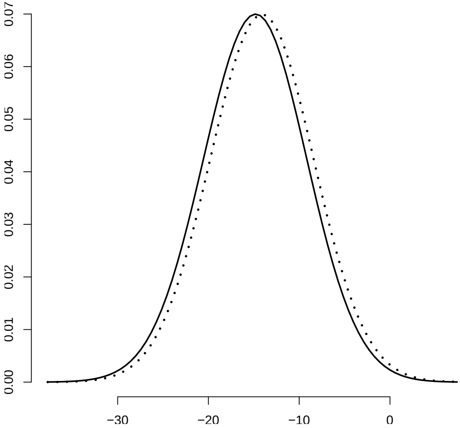
The posterior expectation of the spatial field

Results



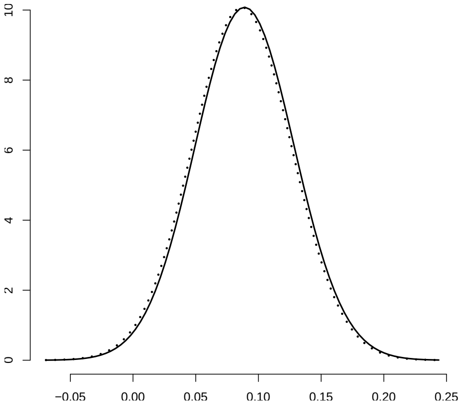
Locations with high KLD

Results



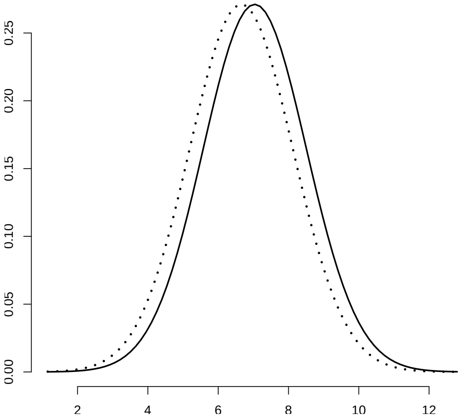
Overall constant β_0

Results



Effect of altitude

Results



Effect of norm of the gradient

Work in progress...

- The *inla*-program
- High(er) number of hyperparameters
- Model choice/selection
- Automatic detection of “surprising” observations
- Parallel computing using OpenMP.

The inla-program

- User-friendly access to the INLA for hierarchical GMRF-models
- No C-coding!
- Similar to the BayesX-program but without MCMC and GUI.
- Solve most problems with this; INLA-paper, GMRF-book, BayesX-examples...
- Soon to be released.

High(er) number of hyperparameters

Numerical (grid) integration is costly and costs at least

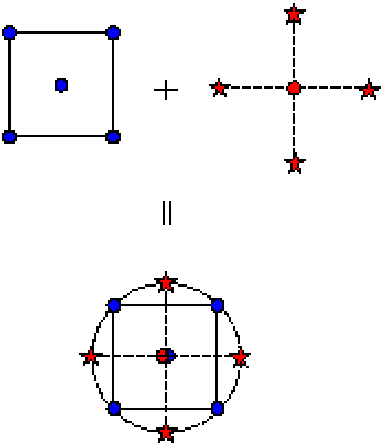
$$3^{\mathrm{dim}(\boldsymbol{\theta})}$$

Need another approach for “high-dimensional” hyperparameters.

Borrow ideas from experimental design...

www.wikipedia.org: *In statistics, a central composite design is an experimental design, useful in response surface methodology, for building a second order (quadratic) model for the response variable without needing to use a complete three-level factorial experiment.*

Idea



Number of integration points

Dimension	#Int.pts CCD	#Int.pts GRID: 3^{dim}
2	9	8
3	15	27
4	25	64
5	27	125
6	45	216
7	79	343
8	81	512
9	147	729
10	149	1000
14	285	2744
18	549	5832
22	1069	10648

Experience so far

- Works extremely well!
- Near identical results as expensive grid-integration
- This is really good news!
- The integration problems is well-behaved.

Model choice

Chose/compare various model is important but difficult

- Bayes factors (general available)
- Deviance information criterion (DIC) (hierarchical models)
- and all these ?IC...

We also need to be able to compute it within our INLA-framework.

Marginal likelihood

Marginal likelihood is the normalising constant for $\widetilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$,

$$\widetilde{\pi}(\mathbf{y}) = \int \left. \frac{\pi(\boldsymbol{\theta})\pi(\mathbf{x}|\theta)\pi(\mathbf{y}|\mathbf{x}, \theta)}{\widetilde{\pi}_{\mathrm{G}}(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})} \right|_{\mathbf{x}=\mathbf{x}^{\star}(\boldsymbol{\theta})} d\boldsymbol{\theta}. \tag{3}$$

I many hierarchical GMRF models the prior is intrinsic/improper, so this is difficult to use.

Pseudo-likelihood

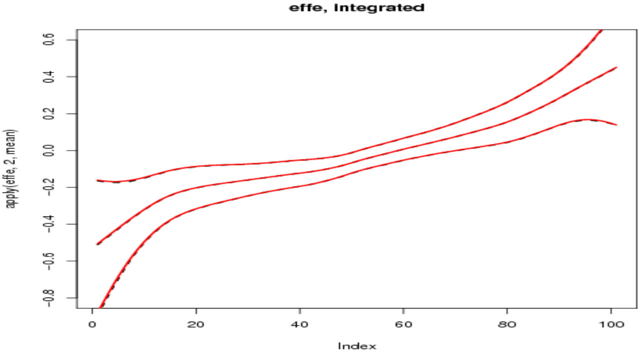
$$PL = \prod_i \pi(y_i \mid \mathbf{y}_{-i})$$

and

$$\text{Pseudo-BF} = \frac{\prod_i \pi(y_i \mid \mathbf{y}_{-i}, M_1)}{\prod_i \pi(y_i \mid \mathbf{y}_{-i}, M_2)}$$

- well defined for intrinsic/improper priors
- similar interpretation
- connects to (Bayesian) cross-validation/leave-one-out procedure
- can be used to detect "surprising" observations
- +++

Example



Will a linear effect be sufficient?

Automatic detection of "surprising" observations

- Leave-one-out idea:

$$\pi(y_i \mid \mathbf{y}_{-i})$$

- Needs proper scaling due to the latent x_i, or the computing of

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

Parallel computing using OpenMP

Why?

- Speed (primary)
- Ability to run larger models (secondary)

Why are so few doing this?

- (Seemingly) difficult
- Better to wait more than to code more
- Lack of local parallel machines.

Result

The Gain/Pain-ratio is simply to low!

But there is *hope*, due to

- new trends in computing
- including parallel tools into mainstream compilers

Trends in computing

Once upon a time, chip makers made computer chips faster every year by increasing their processing speeds. But lately, the microprocessor industry has run into some fundamental limits to those speeds.

Trends in computing

The latest solution: Design chips with multiple processor cores.

Trends in computing

The result: Today's big-brained chips that can do more processing than ever before, if the software is modified to take advantage of their design.

Parallel machines are now everywhere...

Toshiba bærbar PC
SATA20017S

6 995,-

Kjøp

Kraftig bærbar PC med Intel Pentium Dual-Core Prosessor og 160GB harddisk.

Satellite A200-17S er en bærbar PC med 15.4” Widescreen, med et lekkert blått design med sølv og sort! Intel Dual Core prosessor, innebyggetWiFi (802.11b/g), webkamera og DVD-brenner .

Spesifikasjoner »

How to make use of multicore machines?

May 13, 2007: GCC 4.2 Release Series

OpenMP *is now supported for the C, C++ and Fortran compilers.*

OpenMP: coding

- Easy way to parallelize code
- Start with a serial version
- Parallel parts of the code when you have time
- Will still run on a serial machine
- Very little interference with the code itself, mainly compiler directives

Example from GMRFLib

```
#pragma omp parallel for private(i)
for (i = 0; i < n; i++) {
        GMRFLib_2order_approx(NULL, &bb[i], &cc[i], d[i],
                    mode[i], i,
                    mode, loglFunc, loglFunc_arg,
                    &(blockupdate_par->step_len));
        cc[i] = MAX(0.0, cc[i]);
}
```

Spatial GLMs

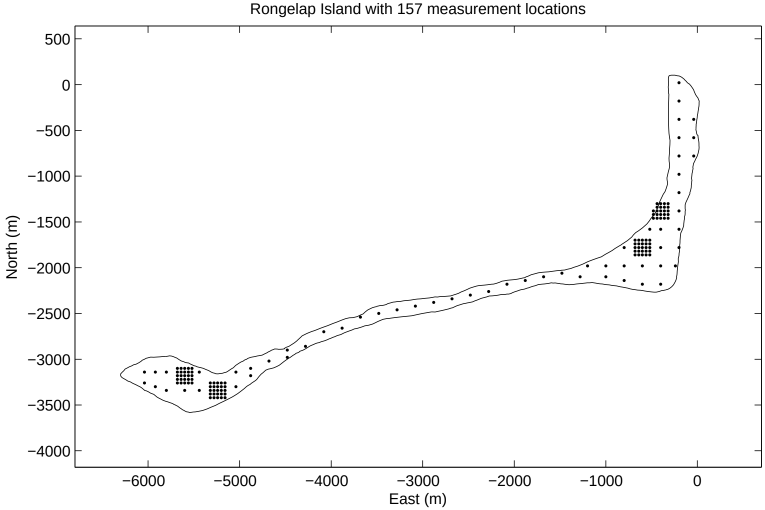
Model

- Stationary Gaussian field on a torus
- non-Gaussian observations
- n is huge: $n = 512^2$ or $n = 1024^2$
- number of observations, m, is small, a few hundred.

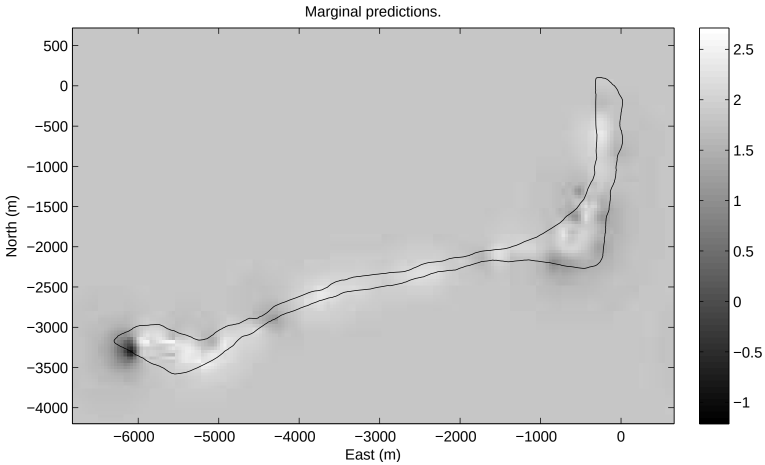
Solve using

- INLA, *but* the computational tools are now very different
  - Exploit the block Toeplitz structure using DFTs
  - and simply rank-m correct for the observations using soft constraints.

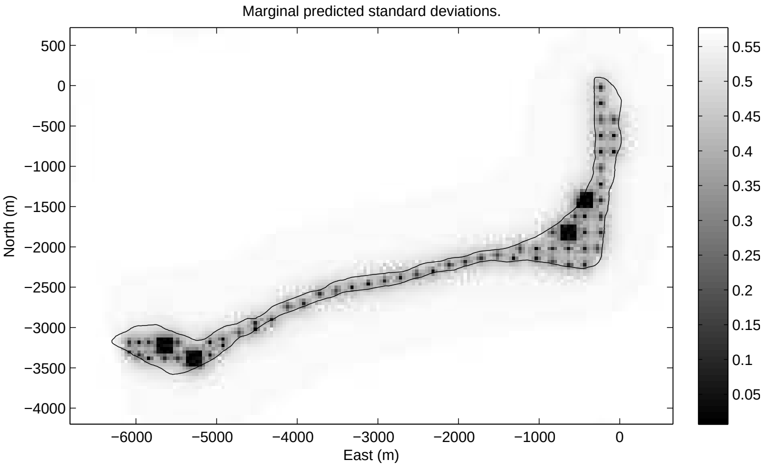
Example: Rongelap data



Example: Rongelap data, results



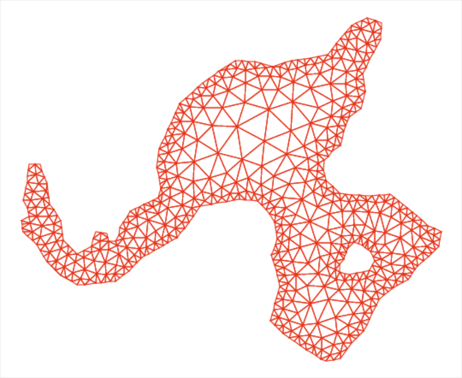
Example: Rongelap data, results



Constructing GMRF-approximations to Gaussian fields

Enlarge the class of models

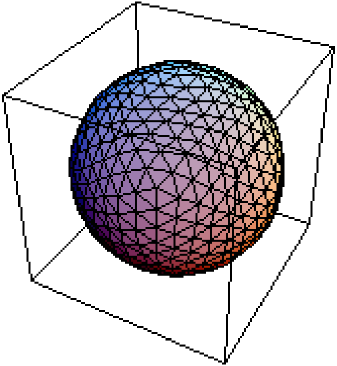
- Gaussian fields on triangulated regions
- Gaussian fields on triangulated manifolds (sphere)



Constructing GMRF-approximations to Gaussian fields

Enlarge the class of models

- Gaussian fields on triangulated regions
- Gaussian fields on triangulated manifolds (sphere)



SPDE-formulation

Starting point: stochastic partial differential equation

$$(\nabla - \kappa^2)^{\alpha/2} z(\mathbf{s}) = \epsilon(\mathbf{s}), \qquad \mathbf{s} \in \mathbb{R}^d$$

and $\epsilon(\mathbf{s})$ is Gaussian white noise.

Solutions has spectrum

$$(\|\boldsymbol{\omega}\|^2 + \kappa^2)^{-\alpha}$$

which is the Matérn-family.

Main idea

Use the Finite Element Method to define *weak* solutions to the SPDE for a class of test-functions.

Define the test-functions so that the solution is a GMRF.

Due to the fractional Laplacian we can only do this for $\alpha = 1, 2, \ldots$.

Summary and discussion (I)

- Latent Gaussian models are an important class of models with a wide range of applications
- The integrated nested Laplace-approximations works extremely well, way beyond my expectations!!!
  - Obtain in practice “exact” results
  - *Relative* error only
  - Computationally FAST even for large n
  - Take advantage of multicore architecture using OpenMP
- Extentions (in progress...)
  - Compare models
  - Locate “surprising” observations
  - High(er) number of hyperparameters

Summary and discussion (II)

- Generic methodology:
  - All examples use the same basic code: GMRFLib
  - Well suited for constructing (black-box) packages for inference: the **inla**-program.
- Conditions apply:
  - Marginals only. Bi- and tri-variate marginals are also OK.
  - Can always construct (artificial) counter-examples