Approximate Bayesian inference for latent Gaussian models

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

December 4, 2009

[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 $\mathbf{x} \sim \mathcal{N}(\cdot, \boldsymbol{\Sigma}(\theta))$
- Hyperparameters θ
  - variability
  - length/strength of dependence
  - parameters in the likelihood

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

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 θ
  - variability
  - length/strength of dependence
  - parameters in the likelihood

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

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

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

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

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

Example: Generalised additive (mixed) models

$$g(\mu_i) = \sum_j f_j(z_{ji}) + \sum_k \beta_j \widetilde{z}_{ji} + \epsilon_i$$

where

- each $f_j(\cdot)$, is a smooth (random) function
- β_j is the linear effect of z_j

Observations $\{y_i\}$ from an exponential family with mean $\{\mu_i\}$

Examples

- *1D* Smoothing count data, general spline smoothing, semi-parametric regression, GLM(M), GAM(M), etc
- *2D* Disease mapping, log-Gaussian Cox-processes, model-based geostatistics, 1D-models with spatial effect(s)
- *3D* Time-series of images, spatio-temporal models.

Features

- Dimension of the latent Gaussian field, n, is large, $10^2 - 10^5$, but often Markov.
- 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, GLM(M), GAM(M), etc

2D Disease mapping, log-Gaussian Cox-processes, model-based geostatistics, 1D-models with spatial effect(s)

3D Time-series of images, spatio-temporal models.

Features

- Dimension of the latent Gaussian field, n, is large, $10^2 - 10^5$, but often Markov.
- 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, GLM(M), GAM(M), etc
- *2D* Disease mapping, log-Gaussian Cox-processes, model-based geostatistics, 1D-models with spatial effect(s)
- *3D* Time-series of images, spatio-temporal models.

Features

- Dimension of the latent Gaussian field, n, is large, $10^2 - 10^5$, but often Markov.
- Dimension of the hyperparameters $\dim(\theta)$ 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, GLM(M), GAM(M), etc
- *2D* Disease mapping, log-Gaussian Cox-processes, model-based geostatistics, 1D-models with spatial effect(s)
- *3D* Time-series of images, spatio-temporal models.

Features

- Dimension of the latent Gaussian field, n, is large, $10^2 - 10^5$, but often Markov.
- Dimension of the hyperparameters $\dim(\theta)$ 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, GLM(M), GAM(M), etc
- *2D* Disease mapping, log-Gaussian Cox-processes, model-based geostatistics, 1D-models with spatial effect(s)
- *3D* Time-series of images, spatio-temporal models.

Features

- Dimension of the latent Gaussian field, n, is large, $10^2 - 10^5$, but often Markov.
- 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, GLM(M), GAM(M), etc

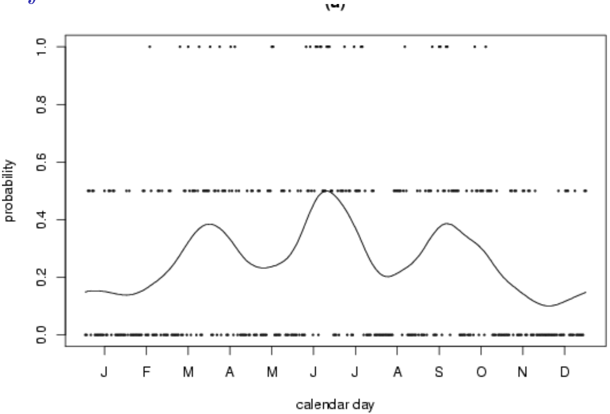
2D Disease mapping, log-Gaussian Cox-processes, model-based geostatistics, 1D-models with spatial effect(s)

3D Time-series of images, spatio-temporal models.

Features

- Dimension of the latent Gaussian field, n, is large, $10^2 - 10^5$, but often Markov.
- 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 of latent Gaussian models: 1D



Longitudinal mixed effects model: Epil-example from BUGS

Patient	y_1	y_2	y_3	y_4	Trt	Base	Age
1	5	3	3	3	0	11	31
2	3	5	3	3	0	11	30
3	2	4	0	5	0	6	25
4	4	4	1	4	0	8	36
....							
8	40	20	21	12	0	52	42
9	5	6	6	5	0	12	37
....							
59	1	4	3	2	1	12	37

Longitudinal mixed effects model: Epil-example from BUGS

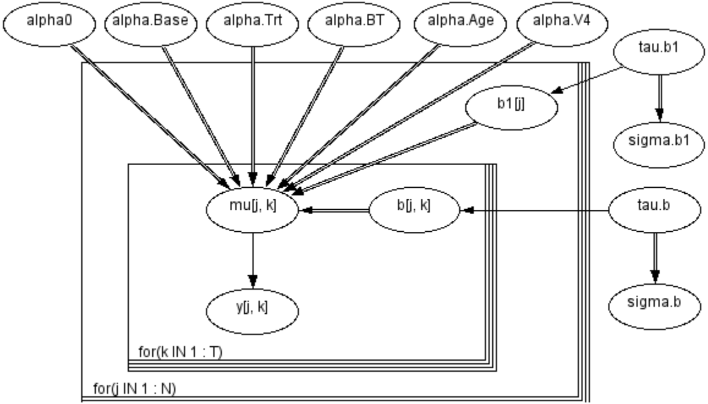
$$y_{jk} \sim \text{Poisson}(m_{jk})$$

$$\log m_{jk} = a_0 + a_{Base} \log(Base_j / 4) + a_{Trt} Trt_j + a_{BT} Trt_j \log(Base_j / 4) + a_{Age} Age_j + a_{V4} V_4 + b1_j + b_{jk}$$

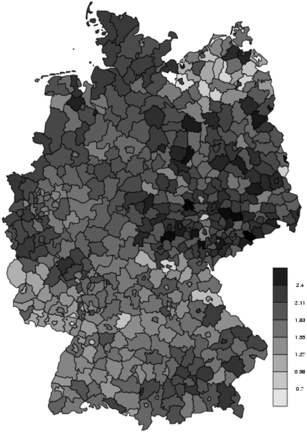
$$b1_j \sim \text{Normal}(0, t_{b1})$$

$$b_{jk} \sim \text{Normal}(0, t_b)$$

Longitudinal mixed effects model: Epil-example from BUGS

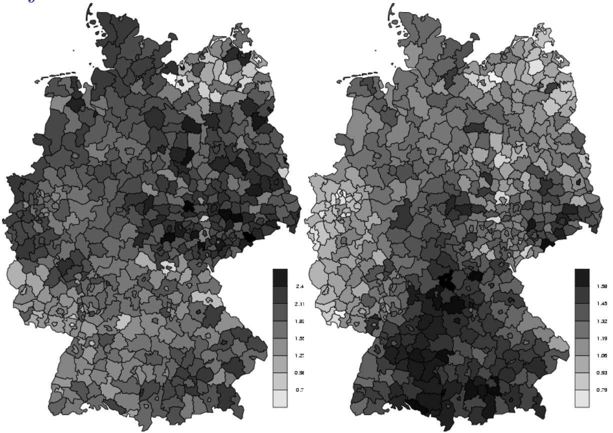


Examples of latent Gaussian models: 2D



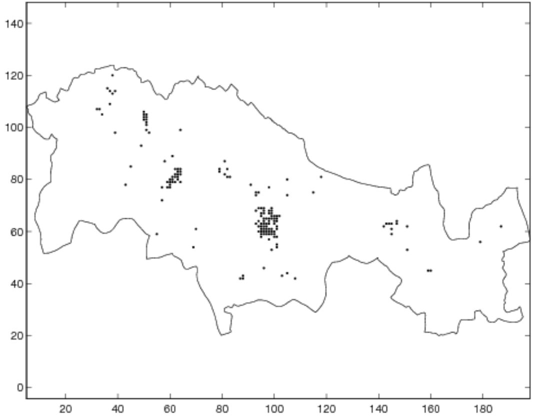
Disease mapping: Poisson data

Examples of latent Gaussian models: 2D



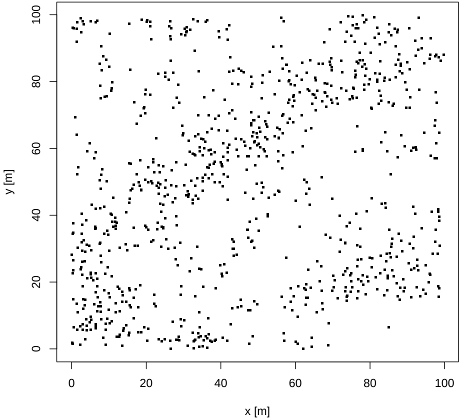
Joint disease mapping: Poisson data

Examples of latent Gaussian models: 2D



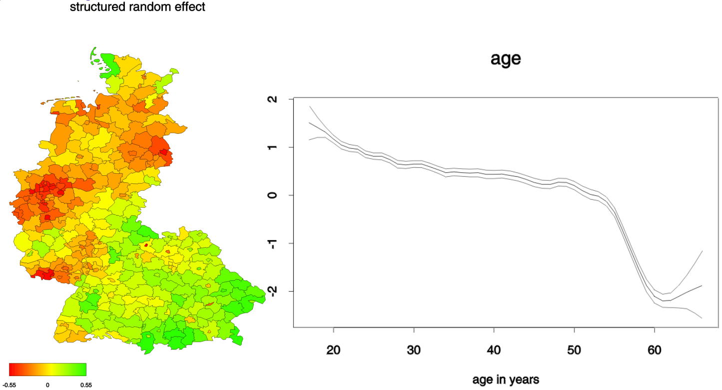
Spatial GLM with Binomial data

Examples of latent Gaussian models: 2D



Log-Gaussian Cox-process; Oaks-data

Examples of latent Gaussian models: 2D+



Spatial logit-model with semiparametric covariates

Tasks

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

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

Our approach: Approximate Bayesian Inference

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

Our approach: Approximate Bayesian Inference

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

Our approach: Approximate Bayesian 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 (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

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

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

Main ideas (II)

Construct the approximations to

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

then we integrate

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

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

Main ideas (II)

Construct the approximations to

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

then we integrate

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

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

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

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

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

Sparse matrices gives fast computations!

The GMRF-approximation

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

Constructed as follows:

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

Markov and computational properties are preserved

The GMRF-approximation

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

Constructed as follows:

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

Markov and computational properties are preserved

Part I

Some more background: The Laplace approximation

Outline I

Outline II

- High(er) number of hyperparameters
- Parallel computing using OpenMP
- Spatial GLMs

The Laplace approximation: The classic case

Compute and approximation to the integral

$$\int \exp(ng(x))\, dx$$

where n is the parameter going to ∞.

Let x_0 be the mode of $g(x)$ and assume $g(x_0) = 0$:

$$g(x) = \frac{1}{2} g''(x_0)(x - x_0)^2 + \cdots .$$

The Laplace approximation: The classic case

Compute and approximation to the integral

$$\int \exp(ng(x))\, dx$$

where n is the parameter going to ∞.

Let x_0 be the mode of $g(x)$ and assume $g(x_0) = 0$:

$$g(x) = \frac{1}{2} g''(x_0)(x - x_0)^2 + \cdots .$$

The Laplace approximation: The classic case...

Then

$$\int \exp(ng(x))\, dx = \sqrt{\frac{2\pi}{n(-g''(x_0))}} + \cdots$$

- As $n \to \infty$, then the integrand gets more and more peaked.
- Error should tends to zero as $n \to \infty$
- Detailed analysis gives

$$\text{relative error}(n) = 1 + \mathcal{O}(1/n)$$

The Laplace approximation: The classic case...

Then

$$\int \exp(ng(x))\, dx = \sqrt{\frac{2\pi}{n(-g''(x_0))}} + \cdots$$

- As $n \to \infty$, then the integrand gets more and more peaked.
- Error should tends to zero as $n \to \infty$
- Detailed analysis gives

$$\text{relative error}(n) = 1 + \mathcal{O}(1/n)$$

The Laplace approximation: The classic case...

Then

$$\int \exp(ng(x))\, dx = \sqrt{\frac{2\pi}{n(-g''(x_0))}} + \cdots$$

- As $n \to \infty$, then the integrand gets more and more peaked.
- Error should tends to zero as $n \to \infty$
- Detailed analysis gives

$$\text{relative error}(n) = 1 + \mathcal{O}(1/n)$$

The Laplace approximation: The classic case...

Then

$$\int \exp(ng(x))\, dx = \sqrt{\frac{2\pi}{n(-g''(x_0))}} + \cdots$$

- As $n \to \infty$, then the integrand gets more and more peaked.
- Error should tends to zero as $n \to \infty$
- Detailed analysis gives

$$\text{relative error}(n) = 1 + \mathcal{O}(1/n)$$

Extension I

$$g_n(x) = \frac{1}{n}\sum_{i=1}^{n} g_i(x)$$

then the mode x_0 depends on n as well.

Extension II

$$\int \exp(ng(\mathbf{x}))\, d\mathbf{x}$$

and $\mathbf{x}$ is multivariate, then

$$\int \exp(ng(\mathbf{x}))\, d\mathbf{x} = \sqrt{\frac{(2\pi)^n}{n|-\mathbf{H}|}}$$

where $\mathbf{H}$ is the hessian (matrix) at the mode

$$H_{ij} = \frac{\partial^2}{\partial x_i \partial x_j} g(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}_0}$$

Extension II

$$\int \exp(ng(\mathbf{x}))\, d\mathbf{x}$$

and $\mathbf{x}$ is multivariate, then

$$\int \exp(ng(\mathbf{x}))\, d\mathbf{x} = \sqrt{\frac{(2\pi)^n}{n|-\mathbf{H}|}}$$

where $\mathbf{H}$ is the hessian (matrix) at the mode

$$H_{ij} = \frac{\partial^2}{\partial x_i \partial x_j} g(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}_0}$$

Computing marginals

- Our main issue is to compute marginals
- We can use the Laplace-approximation for this issue as well
- A more "statistical" derivation might be appropriate

Computing marginals

- Our main issue is to compute marginals
- We can use the Laplace-approximation for this issue as well
- A more “statistical” derivation might be appropriate

Computing marginals

- Our main issue is to compute marginals
- We can use the Laplace-approximation for this issue as well
- A more “statistical” derivation might be appropriate

Computing marginals...

Consider the general problem

- θ is hyper-parameter with prior $\pi(\theta)$
- x is latent with density $\pi(x|\theta)$
- y is observed with likelihood $\pi(y|x)$

then

$$\pi(\theta|y) = \frac{\pi(x,\theta|y)}{\pi(x|\theta,y)}$$

for any x!

Computing marginals...

Consider the general problem

- θ is hyper-parameter with prior $\pi(\theta)$
- x is latent with density $\pi(x|\theta)$
- y is observed with likelihood $\pi(y|x)$

then

$$\pi(\theta|y) = \frac{\pi(x, \theta|y)}{\pi(x|\theta, y)}$$

for any x!

Computing marginals...

Consider the general problem

- θ is hyper-parameter with prior $\pi(\theta)$
- x is latent with density $\pi(x|\theta)$
- y is observed with likelihood $\pi(y|x)$

then

$$\pi(\theta|y) = \frac{\pi(x,\theta|y)}{\pi(x|\theta,y)}$$

for any x!

Computing marginals...

Further,

$$\begin{aligned}
\pi(\theta|y) &= \frac{\pi(x,\theta|y)}{\pi(x|\theta,y)} \\
&\propto \frac{\pi(\theta)\,\pi(x|\theta)\,\pi(y|x)}{\pi(x|\theta,y)} \\
&\approx \left.\frac{\pi(\theta)\,\pi(x|\theta)\,\pi(y|x)}{\pi_G(x|\theta,y)}\right|_{x=x^*(\theta)}
\end{aligned}$$

where $\pi_G(x|\theta,y)$ is the Gaussian approximation of $\pi(x|\theta,y)$ and $x^*(\theta)$ is the mode.

Computing marginals...

Further,

$$
\begin{aligned}
\pi(\theta|y) &= \frac{\pi(x,\theta|y)}{\pi(x|\theta,y)} \\
&\propto \frac{\pi(\theta)\,\pi(x|\theta)\,\pi(y|x)}{\pi(x|\theta,y)} \\
&\approx \left.\frac{\pi(\theta)\,\pi(x|\theta)\,\pi(y|x)}{\pi_G(x|\theta,y)}\right|_{x=x^*(\theta)}
\end{aligned}
$$

where $\pi_G(x|\theta,y)$ is the Gaussian approximation of $\pi(x|\theta,y)$ and $x^*(\theta)$ is the mode.

Computing marginals...

Further,

$$\begin{aligned}
\pi(\theta|y) &= \frac{\pi(x,\theta|y)}{\pi(x|\theta,y)} \\
&\propto \frac{\pi(\theta)\,\pi(x|\theta)\,\pi(y|x)}{\pi(x|\theta,y)} \\
&\approx \left.\frac{\pi(\theta)\,\pi(x|\theta)\,\pi(y|x)}{\pi_G(x|\theta,y)}\right|_{x=x^*(\theta)}
\end{aligned}$$

where $\pi_G(x|\theta, y)$ is the Gaussian approximation of $\pi(x|\theta, y)$ and $x^*(\theta)$ is the mode.

Computing marginals...

Error:

With n repeated measurements of the same x, then the error is

$$\widetilde{\pi}(\theta|y) = \pi(\theta|y)(1 + \mathcal{O}(n^{-3/2}))$$

after renormalisation.

Relative error is a very nice property!

Computing marginals...

Error:

With n repeated measurements of the same x, then the error is

$$\widetilde{\pi}(\theta|y) = \pi(\theta|y)(1 + \mathcal{O}(n^{-3/2}))$$

after renormalisation.

Relative error is a very nice property!

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.

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.

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.

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.

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

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

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

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

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.

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.

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

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

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

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

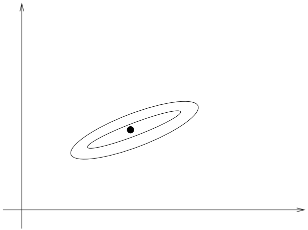
Step I Explore $\widetilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$

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

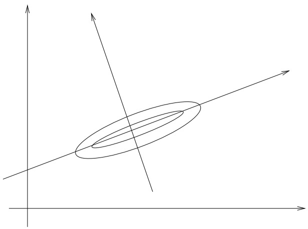
- Locate the mode
- Use the Hessian to construct new variables
- Grid-search
- Can be case-specific



The integrated nested Laplace approximation (INLA) I

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

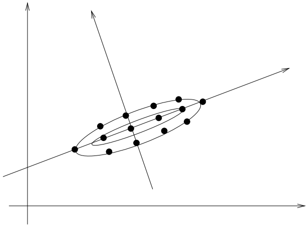
- Locate the mode
- Use the Hessian to construct new variables
- Grid-search
- Can be case-specific



The integrated nested Laplace approximation (INLA) I

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

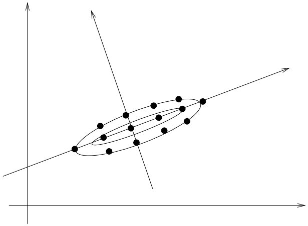
- Locate the mode
- Use the Hessian to construct new variables
- Grid-search
- Can be case-specific



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 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) II

Step II For each $\boldsymbol{\theta}_j$

- 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) II

Step II For each $\boldsymbol{\theta}_j$

- 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 $\boldsymbol{\theta}_j$

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

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

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

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 (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 (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 (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 $\boldsymbol{\theta}^*$
- ...BUT, adjust the standard deviation in each direction
- Then use numerical integration

Computing posterior marginals for θ_j (II)

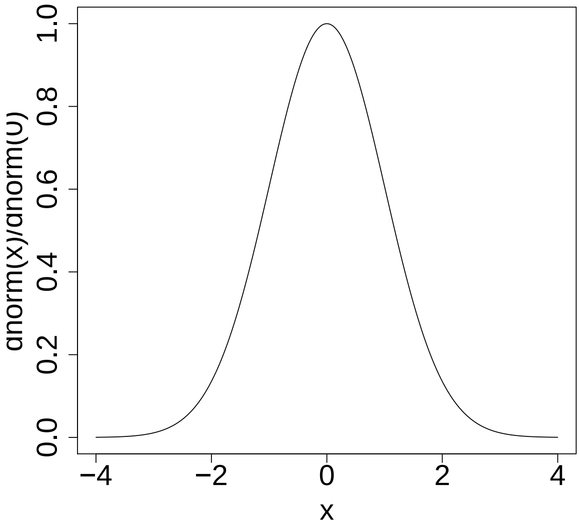
Practical approach (lower accuracy)

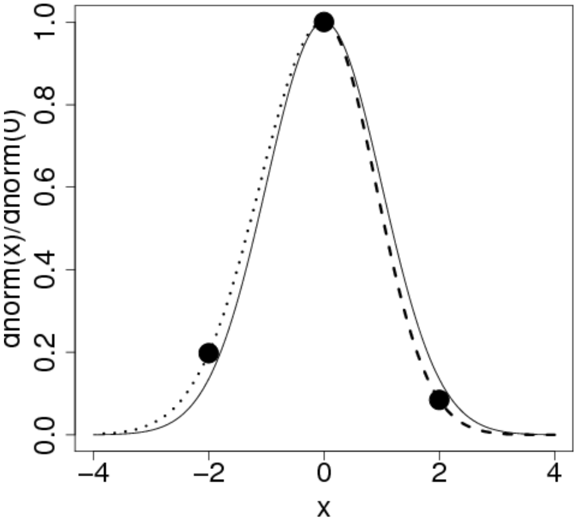
- Use the Gaussian approximation at the mode $\boldsymbol{\theta}^*$
- ...BUT, adjust the standard deviation in each direction
- Then use numerical integration

Computing posterior marginals for θ_j (II)

Practical approach (lower accuracy)

- Use the Gaussian approximation at the mode $\boldsymbol{\theta}^*$
- ...BUT, adjust the standard deviation in each direction
- Then use numerical integration


dnorm(x)/dnorm(0)
0.0
0.2
0.4
0.6
0.8
1.0
−4
−2
0
2
4
x


1.0
0.8
0.6
0.4
0.2
0.0
dnorm(x)/dnorm(0)
−4
−2
0
2
4
x

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 as defined in the *Deviance Information Criteria*:

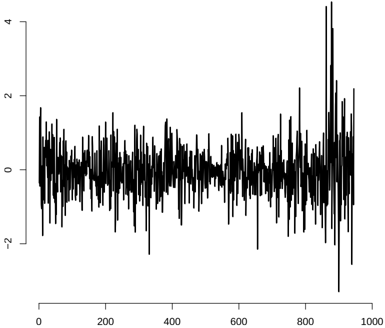
$$p_D(\boldsymbol{\theta}) = \overline{D}(\mathbf{x}; \boldsymbol{\theta}) - D(\overline{\mathbf{x}}; \boldsymbol{\theta})$$

and compare this with the number of observations.

Low ratio is good.

This criteria has theoretical justification.

Stochastic Volatility model



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

Stochastic Volatility model

Simple model

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

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

Observations are taken to be

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

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.

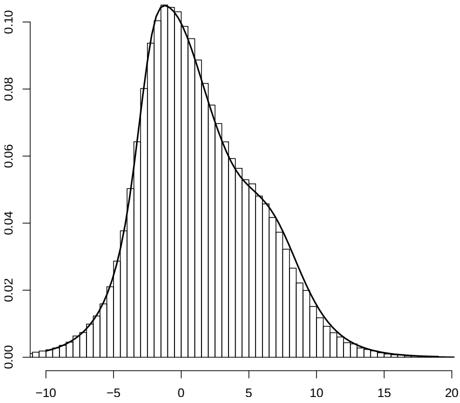
Observations are taken to be

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

Results

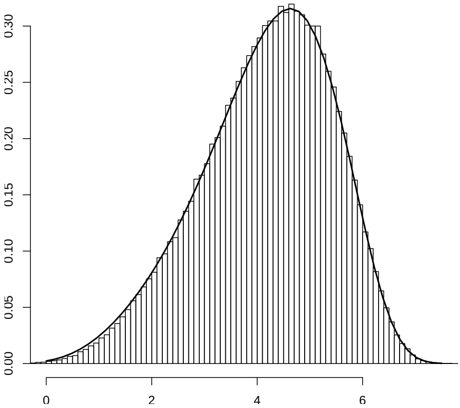
Using just the first 50 data-points only, which makes the problem much harder.

Results



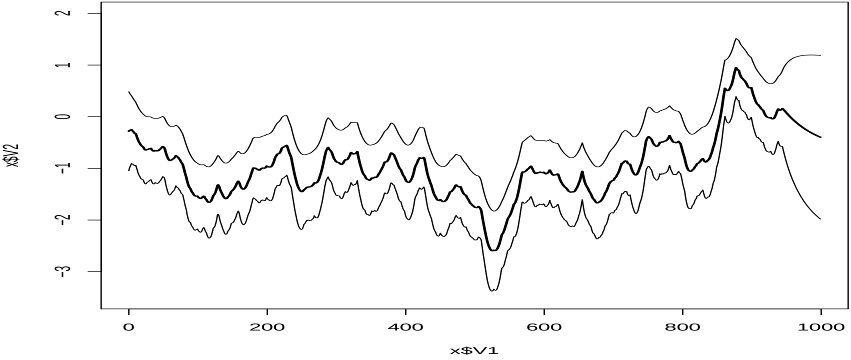
$\nu = \text{logit}(2\phi - 1)$

Results



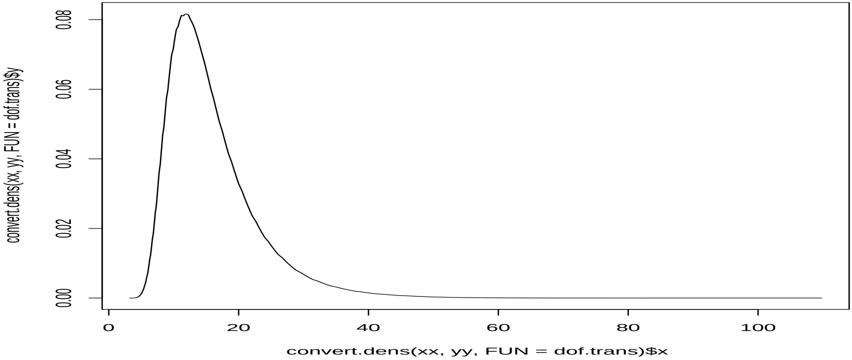
$\log(\kappa_x)$

Using the full dataset



Predictions for $\mu + x_{t+k}$

Student-t$_\nu$



Posterior marginal for ν.

Epil-example from Win/Open-BUGS

Patient	y_1	y_2	y_3	y_4	Trt	Base	Age
1	5	3	3	3	0	11	31
2	3	5	3	3	0	11	30
3	2	4	0	5	0	6	25
4	4	4	1	4	0	8	36
....							
8	40	20	21	12	0	52	42
9	5	6	6	5	0	12	37
....							
59	1	4	3	2	1	12	37

Epil-example from Win/Open-BUGS

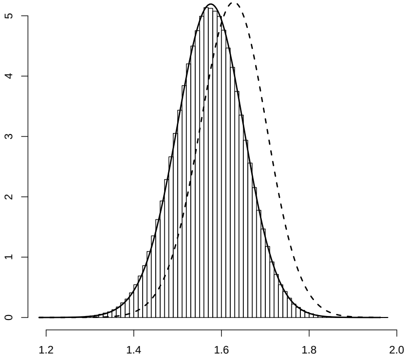
$$y_{jk} \sim \text{Poisson}(m_{jk})$$

$$\log m_{jk} = a_0 + a_{Base}\log(Base_j / 4) + a_{Trt}Trt_j + a_{BT}Trt_j\log(Base_j / 4) + a_{Age}Age_j + a_{V4}V_4 + b1_j + b_{jk}$$

$$b1_j \sim \text{Normal}(0, t_{b1})$$

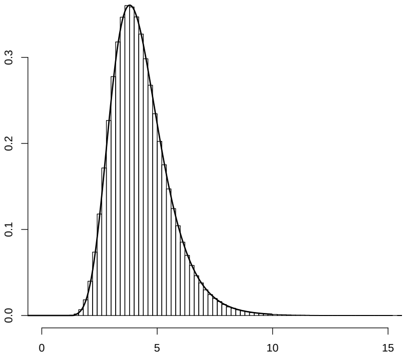
$$b_{jk} \sim \text{Normal}(0, t_b)$$

Epil-example from Win/Open-BUGS



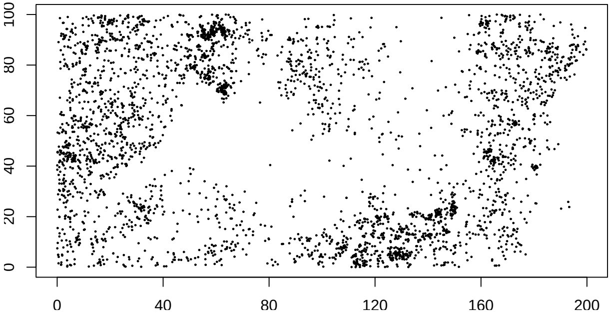
Marginals for a_0

Epil-example from Win/Open-BUGS



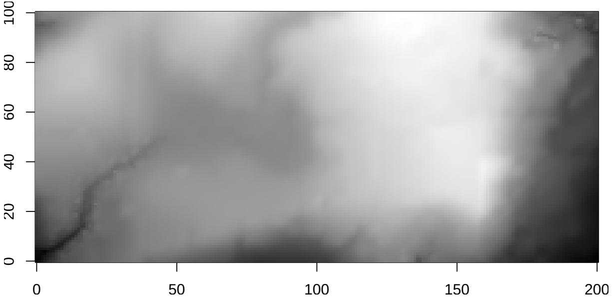
Marginals for τ_{b1}

Log-Gaussian Cox process



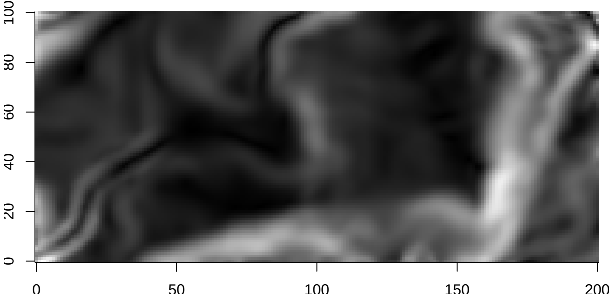
Locations of trees of a particular type: Data comes from a 50-hectare permanent tree plot which was established in 1980 in the tropical moist forest of Barro Colorado Island in Gatun Lake in central Panama.

Log-Gaussian Cox process



Covariate: altitude

Log-Gaussian Cox process



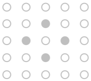
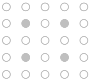
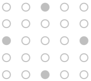
Covariate: norm of gradient

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

$$\mathrm{E}(u_i \mid \mathbf{u}_{-i}) = \frac{1}{20}\Big(8$$

$$-\,2$$

$$-\,1$$

$$\Big)$$

$$\mathrm{Prec}(u_i \mid \mathbf{u}_{-i}) = 20\kappa_{\mathbf{u}}$$

Model

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

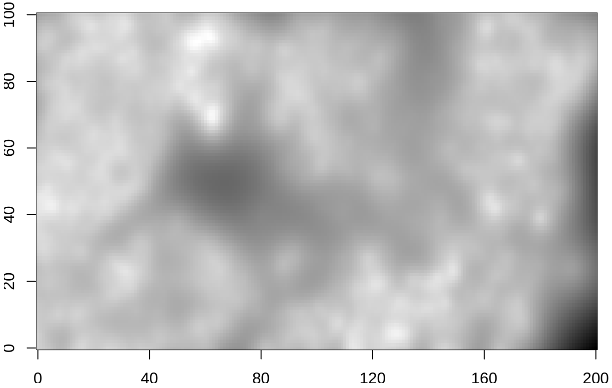
$$\eta_i = \beta_0 + \beta_1 c_{1i} + \beta_2 c_{2i} + u_i + v_i, \qquad \sum_i u_i = 0$$

The spatial term is an IGMRF

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

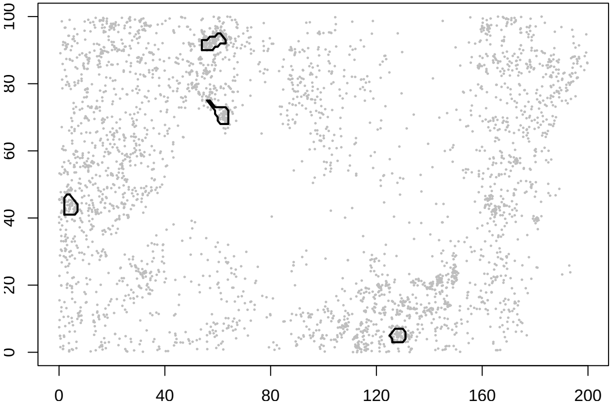
$$\mathrm{Prec}(u_i \mid \mathbf{u}_{-i}) = 20\kappa_{\mathbf{u}}$$

Results



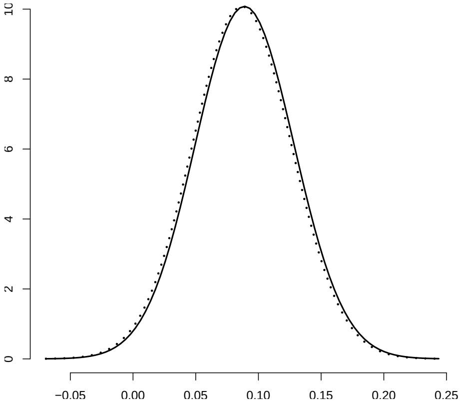
The posterior expectation of the spatial field

Results



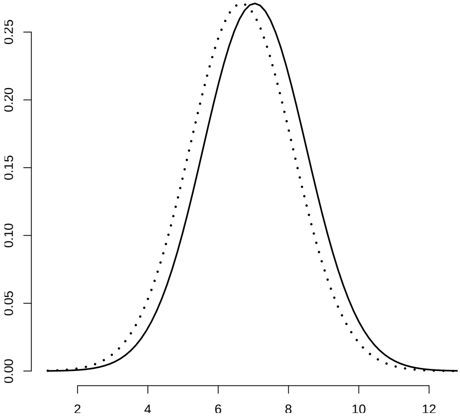
Locations with high KLD

Results



Effect of altitude

Results



Effect of norm of the gradient

Extensions

- Model choice/selection
- Automatic detection of "surprising" observations

Will not discuss

- High(er) number of hyperparameters
- Parallel computing using **OpenMP**
- Sensitivity Analysis

Extensions

- Model choice/selection
- Automatic detection of “surprising” observations

Will not discuss

- High(er) number of hyperparameters
- Parallel computing using **OpenMP**
- Sensitivity Analysis

Extensions

- Model choice/selection
- Automatic detection of "surprising" observations

Will not discuss

- High(er) number of hyperparameters
- Parallel computing using **OpenMP**
- Sensitivity Analysis

Model choice

Chose/compare various model is important but difficult

- Bayes factors (general available)
- Deviance information criterion (DIC) (hierarchical models)

Marginal likelihood

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

$$\widetilde{\pi}(\mathbf{y}) = \int \left. \frac{\pi(\boldsymbol{\theta})\pi(\mathbf{x}|\theta)\pi(\mathbf{y}|\mathbf{x},\theta)}{\widetilde{\pi}_{\mathsf{G}}(\mathbf{x}|\boldsymbol{\theta},\mathbf{y})} \right|_{\mathbf{x}=\mathbf{x}^{\star}(\boldsymbol{\theta})} d\boldsymbol{\theta}. \qquad (2)$$

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

Marginal likelihood

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

$$\widetilde{\pi}(\mathbf{y}) = \int \left. \frac{\pi(\boldsymbol{\theta})\pi(\mathbf{x}|\theta)\pi(\mathbf{y}|\mathbf{x},\theta)}{\widetilde{\pi}_{\mathsf{G}}(\mathbf{x}|\boldsymbol{\theta},\mathbf{y})} \right|_{\mathbf{x}=\mathbf{x}^{\star}(\boldsymbol{\theta})} d\boldsymbol{\theta}. \tag{2}$$

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

Deviance Information Criteria

Based on the *deviance*

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

and

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

This is quite easy to compute

Bayesian Cross-validation

Easy to compute using the INLA-approach

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

where

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

Require a one-dimensional integral for each i and $\boldsymbol{\theta}$.

Automatic detection of "surprising" observations

Compute

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

Look for unusual large or small values

Summary and discussion

- 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
- Extensions
  - Compare models (DIC/Bayes factors)
  - Cross-validation and "surprising" observations
  - High(er) number of hyperparameters
  - Sensitivity analysis

Summary and discussion

- 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
- Extensions
  - Compare models (DIC/Bayes factors)
  - Cross-validation and “surprising” observations
  - High(er) number of hyperparameters
  - Sensitivity analysis

Summary and discussion

- 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
- Extensions
  - Compare models (DIC/Bayes factors)
  - Cross-validation and "surprising" observations
  - High(er) number of hyperparameters
  - Sensitivity analysis

High(er) number of hyperparameters

Numerical (grid) integration is costly and costs at least

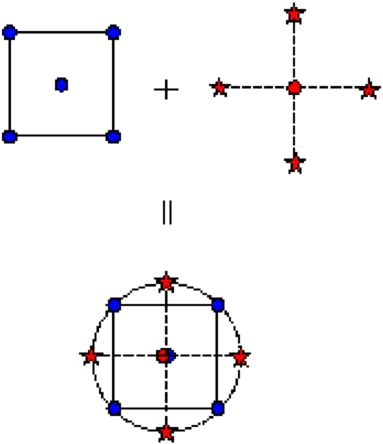
$$3^{\dim(\boldsymbol{\theta})}$$

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 quite well
- The integration problems is well-behaved.

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.

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

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

OpenMP: running

- Just run the program and the run-time environment will take care of the rest.
- This includes how many CPU's that are used at the time.
- This will change during the execution of the program.

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]);
}
```

GMRFLib

- INLA-routines make quite good use of OpenMP
- and so does the **inla**-program.

Spatial GLMs (w/S.Martino/J.Eidsvik)

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.

Spatial GLMs (w/S.Martino/J.Eidsvik)

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.

Spatial GLMs (w/S.Martino/J.Eidsvik)

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.

Spatial GLMs (w/S.Martino/J.Eidsvik)

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.

Spatial GLMs (w/S.Martino/J.Eidsvik)

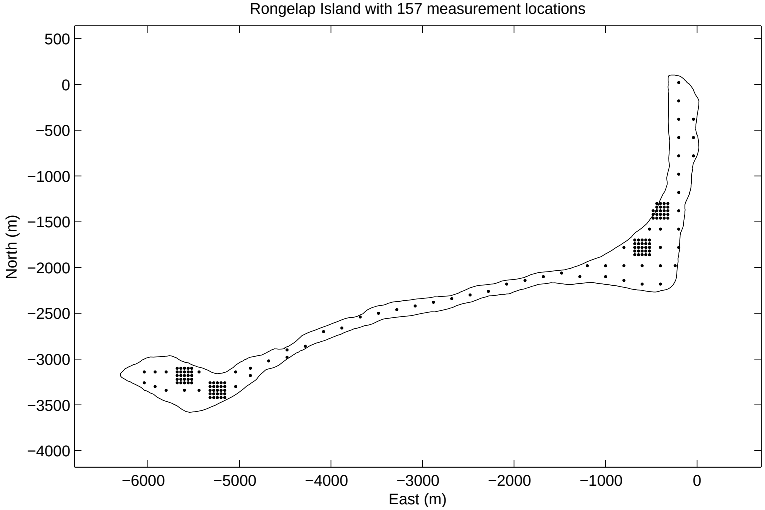
Model

- 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.

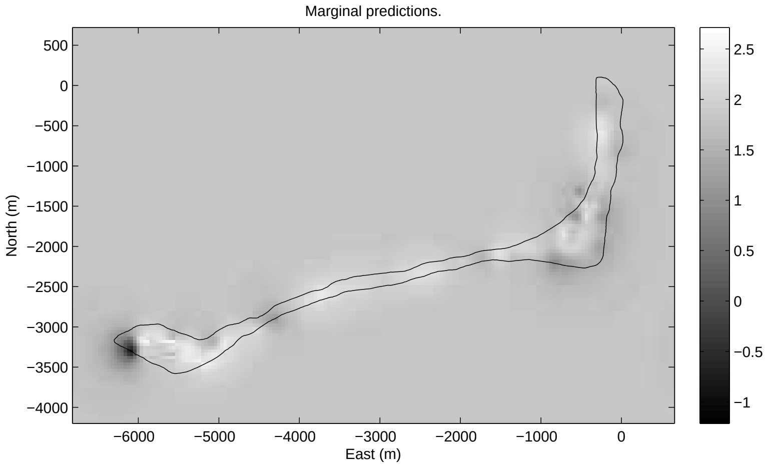
Solve using

- 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.

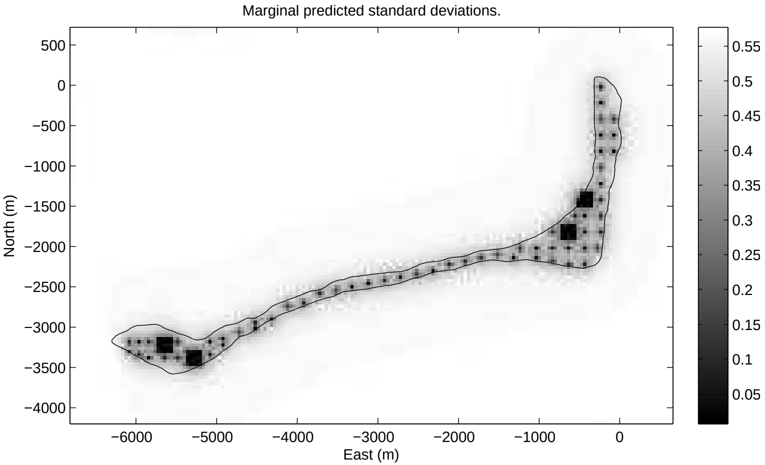
Example: Rongelap data



Example: Rongelap data, results



Example: Rongelap data, results



Spatial GLMs: Summary

- Main interest is to predict unobserved sites
- Gaussian approximations seems sufficient
- they are $\mathcal{O}(m)$-times faster to compute...
- Can also use GMRFs for large m using GMRF-proxies for Gaussian fields

Spatial GLMs: Summary

- Main interest is to predict unobserved sites
- Gaussian approximations seems sufficient
- they are $\mathcal{O}(m)$-times faster to compute...
- Can also use GMRFs for large m using GMRF-proxies for Gaussian fields

Spatial GLMs: Summary

- Main interest is to predict unobserved sites
- Gaussian approximations seems sufficient
- they are $\mathcal{O}(m)$-times faster to compute...
- Can also use GMRFs for large m using GMRF-proxies for Gaussian fields

Spatial GLMs: Summary

- Main interest is to predict unobserved sites
- Gaussian approximations seems sufficient
- they are $\mathcal{O}(m)$-times faster to compute...
- Can also use GMRFs for large m using GMRF-proxies for Gaussian fields