Alternatives to MCMC based inference for latent Gaussian models

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

August 2006

[1]With Sara Martino and Jo Eidsvik (parts)

Outline I

Outline II

Latent Gaussian models

We will consider the following class of models

1. Observed data $\mathbf{y} = \{y_i : i \in \mathcal{I}\}$ where $m = |\mathcal{I}|$

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

2. Latent Gaussian field $\mathbf{x} = (x_1, \ldots, x_n)^T$

$$\pi(\mathbf{x} \mid \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x};\ \boldsymbol{\mu},\ \boldsymbol{\Sigma}(\boldsymbol{\theta}))$$

3. Hyperparameters $\boldsymbol{\theta}$

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

4. Possible linear constraints: $\mathbf{A}\mathbf{x} = \mathbf{e}$

Latent Gaussian models

We will consider the following class of models

1. Observed data $\mathbf{y} = \{y_i : i \in \mathcal{I}\}$ where $m = |\mathcal{I}|$

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

2. Latent Gaussian field $\mathbf{x} = (x_1, \ldots, x_n)^T$

$$\pi(\mathbf{x} \mid \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x};\ \boldsymbol{\mu},\ \boldsymbol{\Sigma}(\boldsymbol{\theta}))$$

3. Hyperparameters $\boldsymbol{\theta}$

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

4. Possible linear constraints: $\mathbf{Ax} = \mathbf{e}$

Latent Gaussian models

We will consider the following class of models

1. Observed data $\mathbf{y} = \{y_i : i \in \mathcal{I}\}$ where $m = |\mathcal{I}|$

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

2. Latent Gaussian field $\mathbf{x} = (x_1, \ldots, x_n)^T$

$$\pi(\mathbf{x} \mid \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x};\ \boldsymbol{\mu},\ \boldsymbol{\Sigma}(\boldsymbol{\theta}))$$

3. Hyperparameters $\boldsymbol{\theta}$

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

4. Possible linear constraints: $\mathbf{Ax} = \mathbf{e}$

Latent Gaussian models

We will consider the following class of models

1. Observed data $\mathbf{y} = \{y_i : i \in \mathcal{I}\}$ where $m = |\mathcal{I}|$

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

2. Latent Gaussian field $\mathbf{x} = (x_1, \ldots, x_n)^T$

$$\pi(\mathbf{x} \mid \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x};\ \boldsymbol{\mu},\ \boldsymbol{\Sigma}(\boldsymbol{\theta}))$$

3. Hyperparameters $\boldsymbol{\theta}$

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

4. Possible linear constraints: $\mathbf{A}\mathbf{x} = \mathbf{e}$

Latent Gaussian models

Characteristic features

- Dimension of the latent Gaussian field, n, is large, $10^2 - 10^5$.
- 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.

Exceptions exists, but we do not consider these.

Latent Gaussian models

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.

Exceptions exists, but we do not consider these.

Latent Gaussian models

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.

Exceptions exists, but we do not consider these.

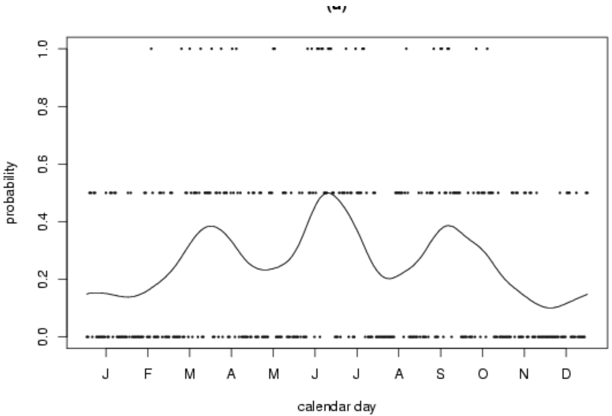
Latent Gaussian models

Characteristic features

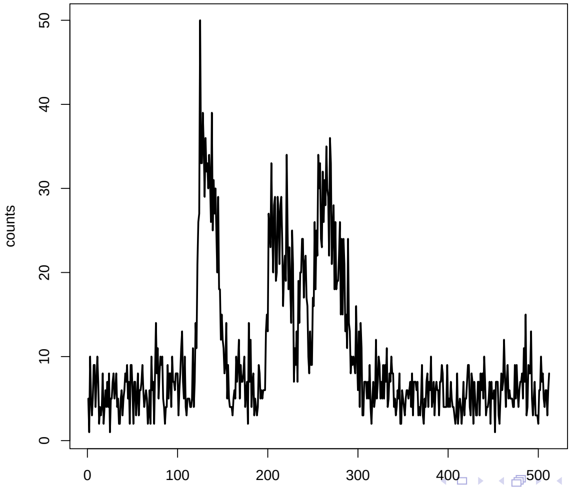
- Dimension of the latent Gaussian field, n, is large, $10^2 - 10^5$.
- Dimension of the hyperparameters $\dim(\boldsymbol{\theta})$ is small, $1 - 5$, say.
- Dimension of the data $\dim(\mathbf{y})$ might vary, but is often non-Gaussian.

Exceptions exists, but we do not consider these.

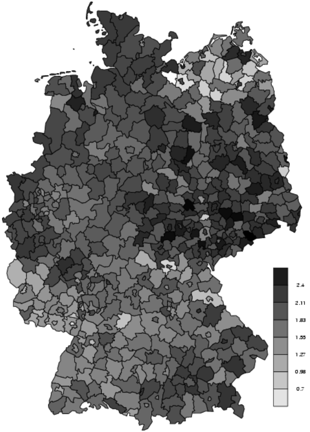
Examples of latent Gaussian models: 1D



Examples of latent Gaussian models: 1D

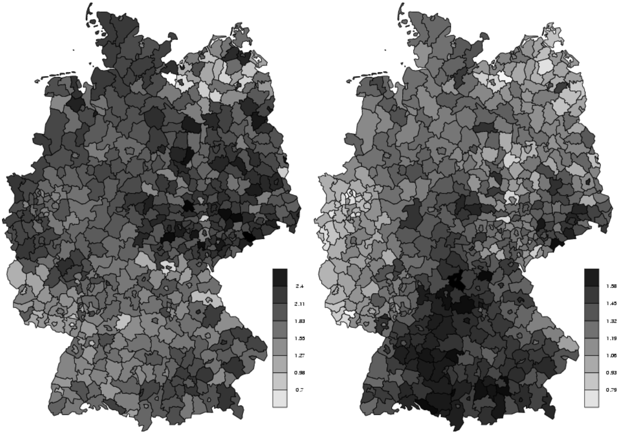


Examples of latent Gaussian models: 2D



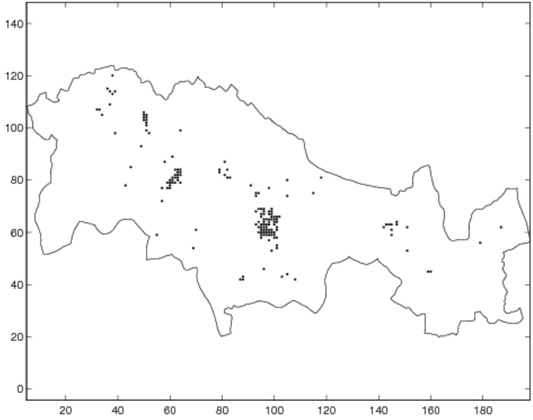
Disease mapping: Poisson data

Examples of latent Gaussian models: 2D



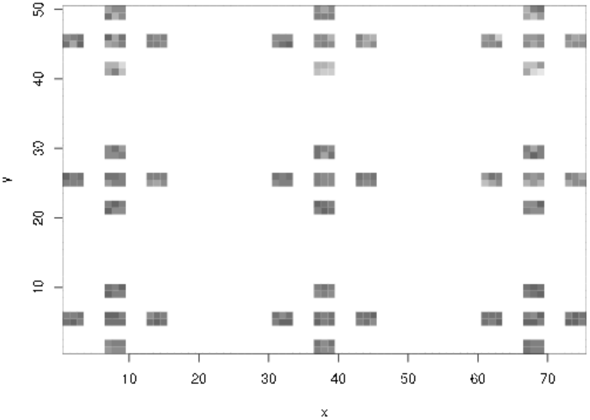
Joint disease mapping: Poisson data

Examples of latent Gaussian models: 2D



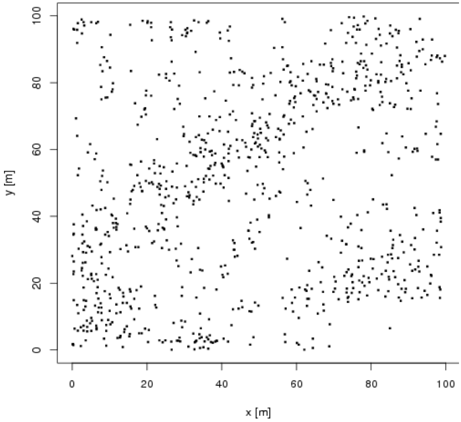
Spatial GLM with Binomial data

Examples of latent Gaussian models: 2D



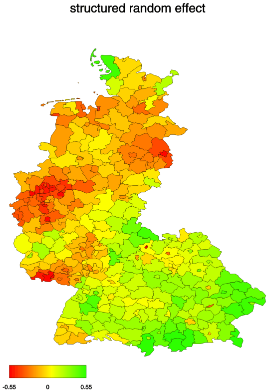
Log-Gaussian Cox-process; Weed-data

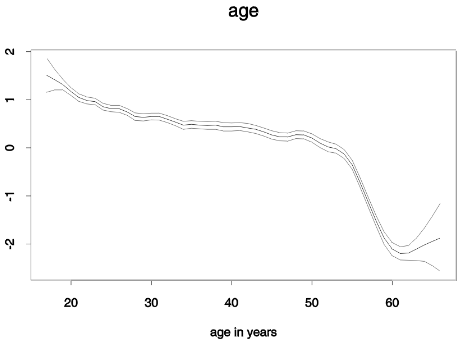
Examples of latent Gaussian models: 2D



Log-Gaussian Cox-process; Oaks-data

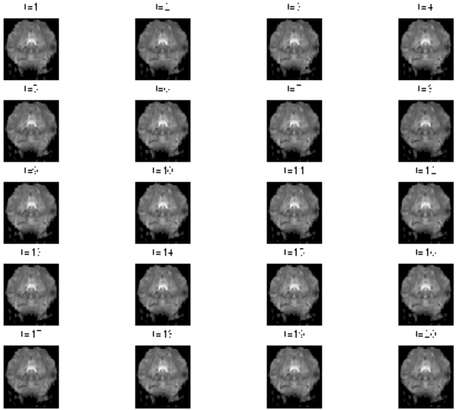
Examples of latent Gaussian models: 2D+





Spatial logit-model with semiparametric covariates

Examples of latent Gaussian models: 3D



Scans in time; fMRI

Characteristic precision/covariance structure

The Gaussian $\mathbf{x}$, is either

- Markov with a local neighbourhood, or
- stationary on a grid or torus

We will focus on Gaussians with local neighbourhood, ie $\mathbf{x}$ is a Gaussian Markov random fields (GMRF).

Discuss the stationary case later on.

Characteristic precision/covariance structure

The Gaussian $\mathbf{x}$, is either

- Markov with a local neighbourhood, or
- stationary on a grid or torus

We will focus on Gaussians with local neighbourhood, ie $\mathbf{x}$ is a Gaussian Markov random fields (GMRF).

Discuss the stationary case later on.

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	$\mathcal{O}(n^2)$

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	$\mathcal{O}(n^2)$

GMRFs: what can we do?

- Unconditional sampling and evaluation of the log density
- Conditional sampling and evaluation of the log density
  - condition on a subset
  - condition on linear hard constraints
  - condition on linear soft constraints
- Compute marginal variances with/without linear constraints

GMRFs: what can we do?

- Unconditional sampling and evaluation of the log density
- Conditional sampling and evaluation of the log density
  - condition on a subset
  - condition on linear hard constraints
  - condition on linear soft constraints
- Compute marginal variances with/without linear constraints

GMRFs: what can we do?

- Unconditional sampling and evaluation of the log density
- Conditional sampling and evaluation of the log density
  - condition on a subset
  - condition on linear hard constraints
  - condition on linear soft constraints
- Compute marginal variances with/without linear constraints

MCMC based inference

Posterior

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

Main problem concerning MCMC:

- The strong interaction between $\boldsymbol{\theta}$ and $\mathbf{x}$.
- Unless they are blocked, the convergence can be painfully slow.

MCMC based inference

Posterior

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

Main problem concerning MCMC:

- The strong interaction between $\boldsymbol{\theta}$ and $\mathbf{x}$.
- Unless they are blocked, the convergence can be painfully slow.

The GMRF-approximation

Build block-MCMC algorithms based on the *GMRF-approximation* $\widetilde{\pi}(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})$:

- A Gaussian approximation to $\pi(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})$ matching mode and curvature
- Markov and computational properties are preserved

Joint updates of $(\boldsymbol{\theta}, \mathbf{x})$ are needed.

The GMRF-approximation

Build block-MCMC algorithms based on the *GMRF-approximation* $\widetilde{\pi}(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})$:

- A Gaussian approximation to $\pi(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})$ matching mode and curvature
- Markov and computational properties are preserved

Joint updates of $(\boldsymbol{\theta}, \mathbf{x})$ are needed.

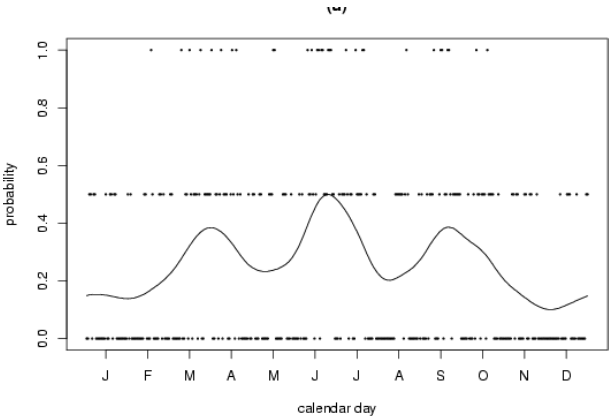
The GMRF-approximation

Build block-MCMC algorithms based on the *GMRF-approximation* $\widetilde{\pi}(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})$:

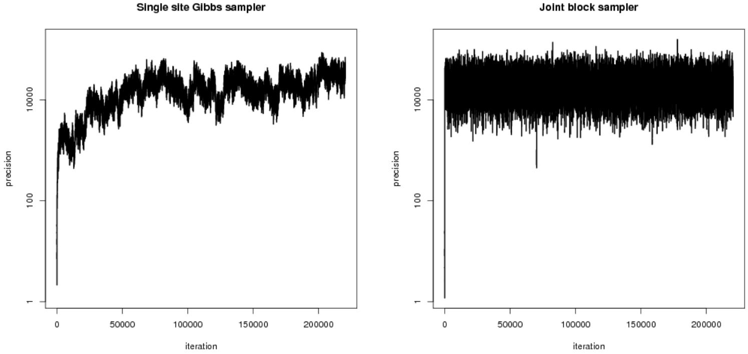
- A Gaussian approximation to $\pi(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y})$ matching mode and curvature
- Markov and computational properties are preserved

Joint updates of $(\boldsymbol{\theta}, \mathbf{x})$ are needed.

Example



Example



Convergence for single-site sampling can be very slow and the uncertainty can be seriously underestimated.

Approximate inference

In most cases the task for the inference, is to compute

- Posterior marginals for x_i

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

- Sometimes, also posterior marginals for θ_j

$$\pi(\theta \mid \mathbf{y})$$

Approximate inference:

- Can we use the GMRF-approximation to estimate these directly without any MCMC?
- Can we gain robustness, accuracy and speed?

Approximate inference

In most cases the task for the inference, is to compute

- Posterior marginals for x_i

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

- Sometimes, also posterior marginals for θ_j

$$\pi(\theta \mid \mathbf{y})$$

Approximate inference:

- Can we use the GMRF-approximation to estimate these directly without any MCMC?
- Can we gain robustness, accuracy and speed?

Approximate inference

In most cases the task for the inference, is to compute

- Posterior marginals for x_i

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

- Sometimes, also posterior marginals for θ_j

$$\pi(\theta \mid \mathbf{y})$$

Approximate inference:

- Can we use the GMRF-approximation to estimate these directly without any MCMC?
- Can we gain robustness, accuracy and speed?

The Laplace approximation

Let $\widetilde{\pi}(\mathbf{x} \mid \mathbf{y}, \boldsymbol{\theta})$ denote the family of GMRF-approximations indexed by $\boldsymbol{\theta}$ and constructed at modes $\mathbf{x}^* = \mathbf{x}^*(\boldsymbol{\theta})$.

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

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

The Laplace approximation

Let $\widetilde{\pi}(\mathbf{x} \mid \mathbf{y}, \boldsymbol{\theta})$ denote the family of GMRF-approximations indexed by $\boldsymbol{\theta}$ and constructed at modes $\mathbf{x}^* = \mathbf{x}^*(\boldsymbol{\theta})$.

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

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

Remarks

The approximation

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

turn out to be very good, since

- $\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}$ is *essentially* Gaussian, since $\mathbf{x}$ is Gaussian.
- The error is *relative* and $\mathcal{O}(m^{-3/2})$ in a $m^{-1/2}$ neighbourhood after renormalisation (Tierney and Kadane, 1986).

Remarks

The approximation

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

turn out to be very good, since

- $\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}$ is *essentially* Gaussian, since $\mathbf{x}$ is Gaussian.
- The error is *relative* and $\mathcal{O}(m^{-3/2})$ in a $m^{-1/2}$ neighbourhood after renormalisation (Tierney and Kadane, 1986).

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 alternative is to use the GMRF-approximation.

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 alternative is to use the GMRF-approximation.

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 alternative is to use the GMRF-approximation.

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

- Let $\widetilde{\pi}(\mathbf{x}_{-i}|x_i, \mathbf{y}, \boldsymbol{\theta})$ be the family of GMRF-approximations indexed by $(x_i, \boldsymbol{\theta})$ and constructed at the mode $\mathbf{x}^*_{-i} = \mathbf{x}^*_{-i}(x_i, \boldsymbol{\theta})$.
- The Laplace approximation is then

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

- Again, it's essentially Gaussian
- However, a such approach is not "practical" for large n, unless...

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

- Let $\widetilde{\pi}(\mathbf{x}_{-i}|x_i, \mathbf{y}, \boldsymbol{\theta})$ be the family of GMRF-approximations indexed by $(x_i, \boldsymbol{\theta})$ and constructed at the mode $\mathbf{x}^*_{-i} = \mathbf{x}^*_{-i}(x_i, \boldsymbol{\theta})$.
- The Laplace approximation is then

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

- Again, it's essentially Gaussian
- However, a such approach is not "practical" for large n, unless...

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

- Let $\widetilde{\pi}(\mathbf{x}_{-i}|x_i, \mathbf{y}, \boldsymbol{\theta})$ be the family of GMRF-approximations indexed by $(x_i, \boldsymbol{\theta})$ and constructed at the mode $\mathbf{x}^*_{-i} = \mathbf{x}^*_{-i}(x_i, \boldsymbol{\theta})$.
- The Laplace approximation is then

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

- Again, it's essentially Gaussian
- However, a such approach is not "practical" for large n, unless...

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

- Let $\widetilde{\pi}(\mathbf{x}_{-i}|x_i, \mathbf{y}, \boldsymbol{\theta})$ be the family of GMRF-approximations indexed by $(x_i, \boldsymbol{\theta})$ and constructed at the mode $\mathbf{x}^*_{-i} = \mathbf{x}^*_{-i}(x_i, \boldsymbol{\theta})$.
- The Laplace approximation is then

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

- Again, it's essentially Gaussian
- However, a such approach is not "practical" for large n, unless...

Practicalities: Overview

...we cut the costs!

- Reduce the size n to involving only the "important" neighbours in some sense
- Remove the optimisation step in the GMRF-approximation $\widetilde{\pi}(\mathbf{x}_{-i}|x_i, \mathbf{y}, \boldsymbol{\theta})$

Practicalities: Overview

...we cut the costs!

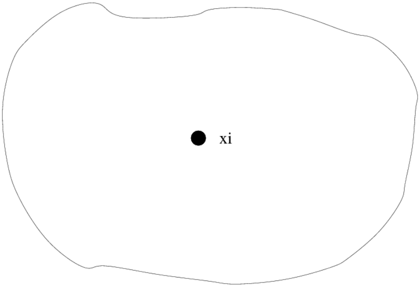
- Reduce the size n to involving only the “important” neighbours in some sense
- Remove the optimisation step in the GMRF-approximation $\widetilde{\pi}(\mathbf{x}_{-i}|x_i, \mathbf{y}, \theta)$

Practicalities: Overview

...we cut the costs!

- Reduce the size n to involving only the "important" neighbours in some sense
- Remove the optimisation step in the GMRF-approximation $\widetilde{\pi}(\mathbf{x}_{-i}|x_i, \mathbf{y}, \boldsymbol{\theta})$

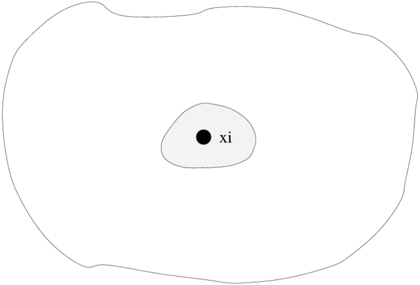
Practicalities: part I



- Compute the conditional mean in $\widetilde{\pi}(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$ when additionally condition on x_i. Rank 1 update.
- Classify using derivatives

$$\frac{d}{dx_i}\widetilde{\mathrm{E}}(x_j \mid \mathbf{y}, \boldsymbol{\theta}, x_i)$$

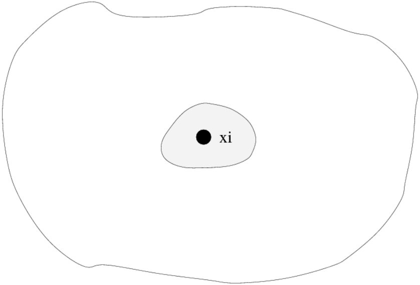
Practicalities: part I



- Compute the conditional mean in $\widetilde{\pi}(\mathbf{x}|\mathbf{y},\boldsymbol{\theta})$ when additionally condition on x_i. Rank 1 update.
- Classify using derivatives

$$\frac{d}{dx_i}\widetilde{\mathrm{E}}(x_j \mid \mathbf{y},\boldsymbol{\theta},x_i)$$

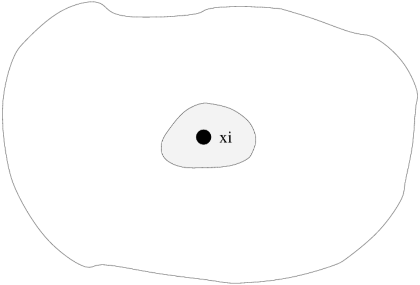
Practicalities: part I



- Compute the conditional mean in $\widetilde{\pi}(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$ when additionally condition on x_i. Rank 1 update.
- Classify using derivatives

$$\frac{d}{dx_i}\widetilde{\mathrm{E}}(x_j \mid \mathbf{y}, \boldsymbol{\theta}, x_i)$$

Practicalities: part I



- Compute the conditional mean in $\widetilde{\pi}(\mathbf{x}|\mathbf{y},\boldsymbol{\theta})$ when additionally condition on x_i. Rank 1 update.
- Classify using derivatives

$$\frac{d}{dx_i}\widetilde{\mathrm{E}}(x_j \mid \mathbf{y}, \boldsymbol{\theta}, x_i)$$

Practicalities: part II

- Construct the $\widetilde{\pi}(\mathbf{x}_{-i}|x_i, \mathbf{y}, \boldsymbol{\theta})$ at the same rank 1 adjusted conditional mean.
- This trick avoids optimisation and reduce CPU but not the computational complexity

Practicalities: part II

- Construct the $\widetilde{\pi}(\mathbf{x}_{-i}|x_i, \mathbf{y}, \boldsymbol{\theta})$ at the same rank 1 adjusted conditional mean.
- This trick avoids optimisation and reduce CPU but not the computational complexity

The integrated nested Laplace approximation (INLA) I

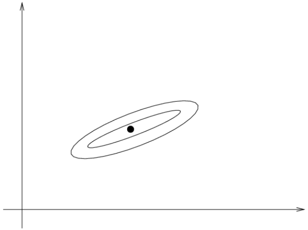
Step I Explore $\widetilde{\pi}(\boldsymbol{\theta}|\mathbf{y})$

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

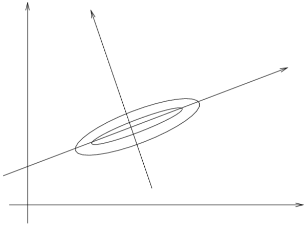
- Locate the mode
- Use the Hessian to construct new variables
- Grid-search
- Can be case-specific



The integrated nested Laplace approximation (INLA) I

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

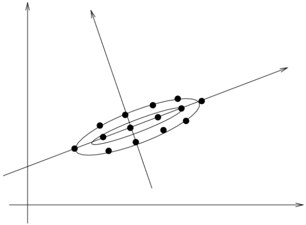
- Locate the mode
- Use the Hessian to construct new variables
- Grid-search
- Can be case-specific



The integrated nested Laplace approximation (INLA) I

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

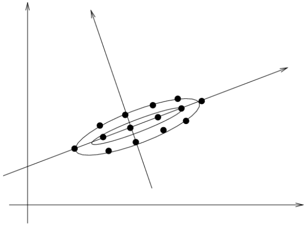
- Locate the mode
- Use the Hessian to construct new variables
- Grid-search
- Can be case-specific



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

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

Remarks

- The latent Gaussian makes the critical Gaussian approximations good, as they are "essentially" Gaussian
- Obtain *relative* error
- We obtain correct results in limits:
  - Strong smoothing: CLT type argument
  - Little smoothing: no dependence of x_i.

Remarks

- The latent Gaussian makes the critical Gaussian approximations good, as they are "essentially" Gaussian
- Obtain *relative* error
- We obtain correct results in limits:
  - Strong smoothing: CLT type argument
  - Little smoothing: no dependence of x_i.

Remarks

- The latent Gaussian makes the critical Gaussian approximations good, as they are “essentially” Gaussian
- Obtain *relative* error
- We obtain correct results in limits:
  - Strong smoothing: CLT type argument
  - Little smoothing: no dependence of x_i.

Computational complexity

Assume a spatial GMRF:

- Factorisation of **Q**: $\mathcal{O}(n^{3/2})$
- Compute the marginal for each i
  - Size of dependency $\mathcal{O}(1)$: cost $\mathcal{O}(1)$.
  - Size of dependency $\mathcal{O}(n)$: cost $\mathcal{O}(n^{3/2})$.
- Summing out θ: $\mathcal{O}(\exp(\dim(\theta)))$

Total cost is between $\mathcal{O}(n^{3/2})$ and $\mathcal{O}(n^{5/2})$, times $\mathcal{O}(\exp(\dim(\theta)))$

Computational complexity

Assume a spatial GMRF:

- Factorisation of $\mathbf{Q}$: $\mathcal{O}(n^{3/2})$
- Compute the marginal for each i
  - Size of dependency $\mathcal{O}(1)$: cost $\mathcal{O}(1)$.
  - Size of dependency $\mathcal{O}(n)$: cost $\mathcal{O}(n^{3/2})$.
- Summing out θ: $\mathcal{O}(\exp(\dim(\theta)))$

Total cost is between $\mathcal{O}(n^{3/2})$ and $\mathcal{O}(n^{5/2})$, times $\mathcal{O}(\exp(\dim(\theta)))$

Computational complexity

Assume a spatial GMRF:

- Factorisation of $\mathbf{Q}$: $\mathcal{O}(n^{3/2})$
- Compute the marginal for each i
  - Size of dependency $\mathcal{O}(1)$: cost $\mathcal{O}(1)$.
  - Size of dependency $\mathcal{O}(n)$: cost $\mathcal{O}(n^{3/2})$.
- Summing out $\boldsymbol{\theta}$: $\mathcal{O}(\exp(\dim(\boldsymbol{\theta})))$

Total cost is between $\mathcal{O}(n^{3/2})$ and $\mathcal{O}(n^{5/2})$, times $\mathcal{O}(\exp(\dim(\boldsymbol{\theta})))$

Computational complexity

Assume a spatial GMRF:

- Factorisation of $\mathbf{Q}$: $\mathcal{O}(n^{3/2})$
- Compute the marginal for each i
  - Size of dependency $\mathcal{O}(1)$: cost $\mathcal{O}(1)$.
  - Size of dependency $\mathcal{O}(n)$: cost $\mathcal{O}(n^{3/2})$.
- Summing out $\boldsymbol{\theta}$: $\mathcal{O}(\exp(\dim(\boldsymbol{\theta})))$

Total cost is between $\mathcal{O}(n^{3/2})$ and $\mathcal{O}(n^{5/2})$, times $\mathcal{O}(\exp(\dim(\boldsymbol{\theta})))$

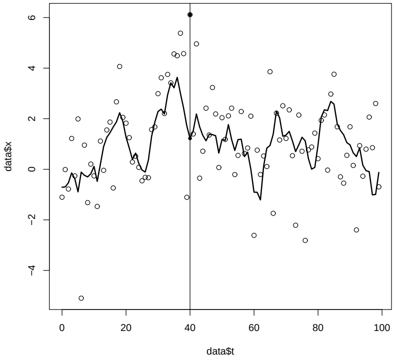
Examples: 1D

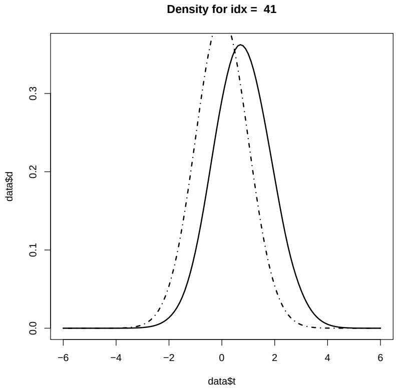
Toy example:

- AR1 model, $\phi = 0.9$, unit variance and common unknown mean.
- Additive noise of various types, or Bernoulli observations using logit

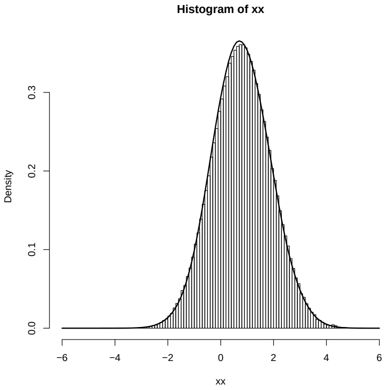
Fixed $\boldsymbol{\theta}$.

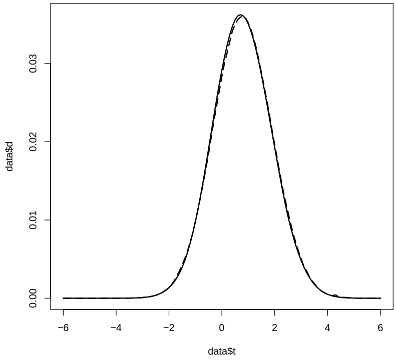
Student-t_3


data$x
data$t
6
4
2
0
−2
−4
0
20
40
60
80
100

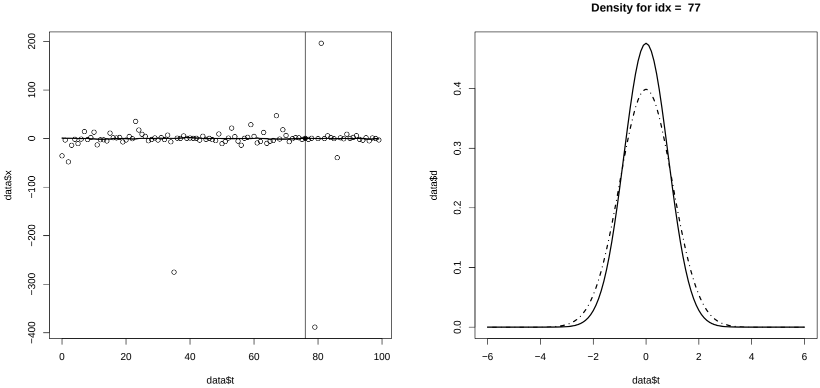

Density for idx = 41
data$d
data$t
0.0
0.1
0.2
0.3
−6
−4
−2
0
2
4
6

Student-t_3

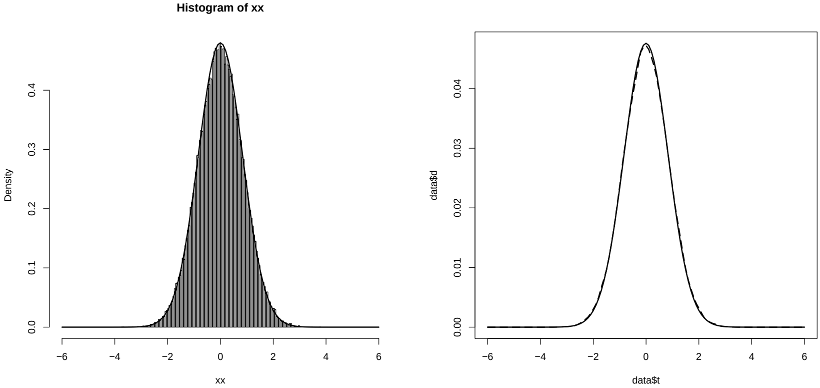

Histogram of xx
Density
0.0
0.1
0.2
0.3
−6
−4
−2
0
2
4
6
xx


data$d
0.00
0.01
0.02
0.03
−6
−4
−2
0
2
4
6
data$t

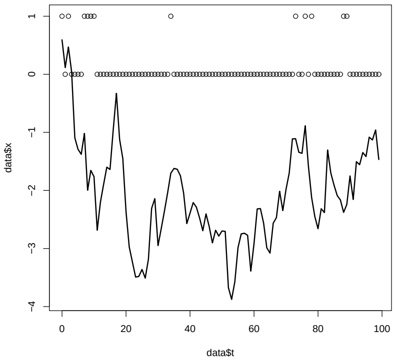
Cauchy with sum-to-zero constraint

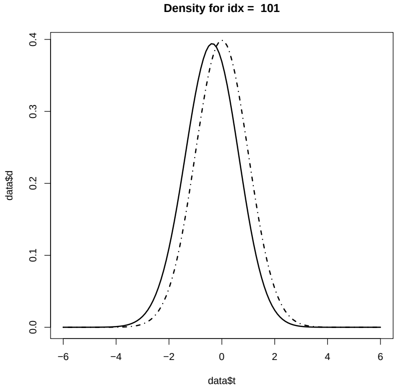


Cauchy with sum-to-zero constraint

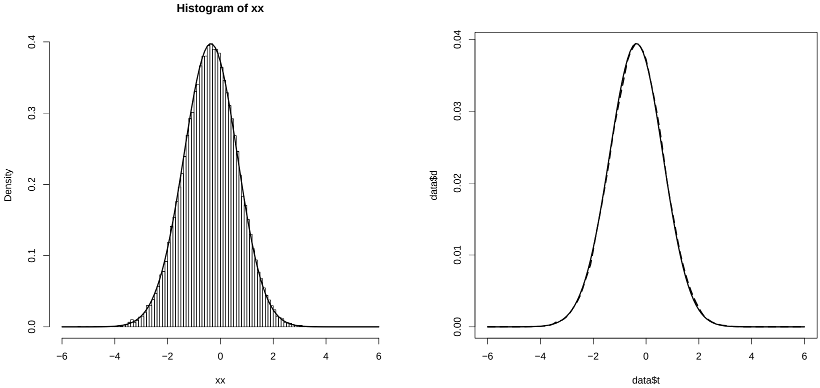


Bernoulli


data$x
1
0
−1
−2
−3
−4
0
20
40
60
80
100
data$t


Density for idx = 101
data$d
0.0
0.1
0.2
0.3
0.4
−6
−4
−2
0
2
4
6
data$t

Bernoulli


Histogram of xx
Density
0.0
0.1
0.2
0.3
0.4
−6
−4
−2
0
2
4
6
xx
data$d
0.00
0.01
0.02
0.03
0.04
−6
−4
−2
0
2
4
6
data$t

Example: Bayesian multiscale analysis of time series data

- Smoothing of time series data with noise and smoothing parameter κ
- Exploratory tool: Consider the family of smooths for all κ and display "significant" gradients

Example:

- Poisson count data

$$y_i \sim \text{Po}(\exp(x_i))$$

- Integrated Wiener process in continuous time-prior for $\mathbf{x}$ with precision κ

Example: Bayesian multiscale analysis of time series data

- Smoothing of time series data with noise and smoothing parameter κ
- Exploratory tool: Consider the family of smooths for all κ and display "significant" gradients

Example:

- Poisson count data

$$y_i \sim \text{Po}(\exp(x_i))$$

- Integrated Wiener process in continuous time-prior for $\mathbf{x}$ with precision κ

Example: Bayesian multiscale analysis of time series data

- Smoothing of time series data with noise and smoothing parameter κ
- Exploratory tool: Consider the family of smooths for all κ and display "significant" gradients

Example:

- Poisson count data

$$y_i \sim \text{Po}(\exp(x_i))$$

- Integrated Wiener process in continuous time-prior for $\mathbf{x}$ with precision κ

Example: Bayesian multiscale analysis of time series data

Significant positive/negative gradient for level κ:

$$\text{Prob}(\frac{d}{dt}x(t) > 0 \mid \mathbf{y}, \kappa) > 0.025$$

$$\text{Prob}(\frac{d}{dt}x(t) < 0 \mid \mathbf{y}, \kappa) > 0.025$$

- Write the integrated Wiener process as a GMRF by augmenting with the derivatives
- Access properties of the derivatives of $x(t)$ directly

Example: Bayesian multiscale analysis of time series data

Significant positive/negative gradient for level κ:

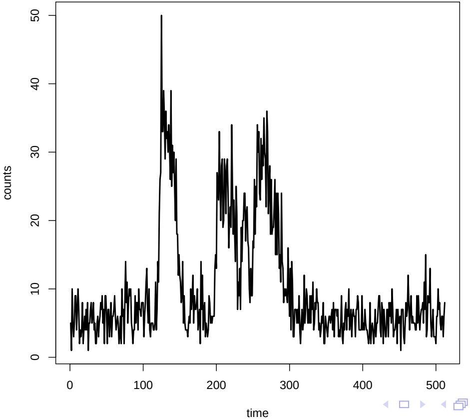
$$\text{Prob}\left(\frac{d}{dt}x(t) > 0 \mid \mathbf{y}, \kappa\right) > 0.025$$

$$\text{Prob}\left(\frac{d}{dt}x(t) < 0 \mid \mathbf{y}, \kappa\right) > 0.025$$

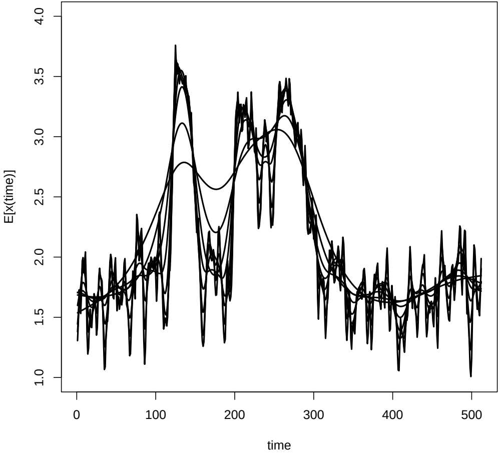
- Write the integrated Wiener process as a GMRF by augmenting with the derivatives
- Access properties of the derivatives of $x(t)$ directly

Example: Bayesian multiscale analysis of time series data

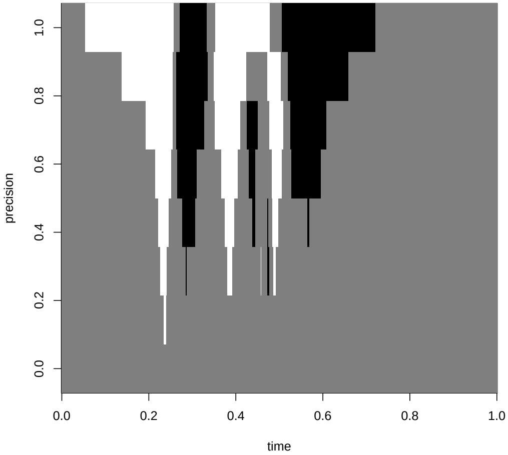
Gamma burst-signals from NASAs Compton Gamma Ray Observatory



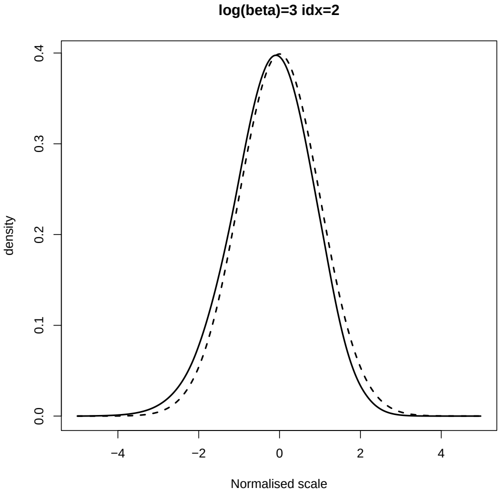
Example: Bayesian multiscale analysis of time series data



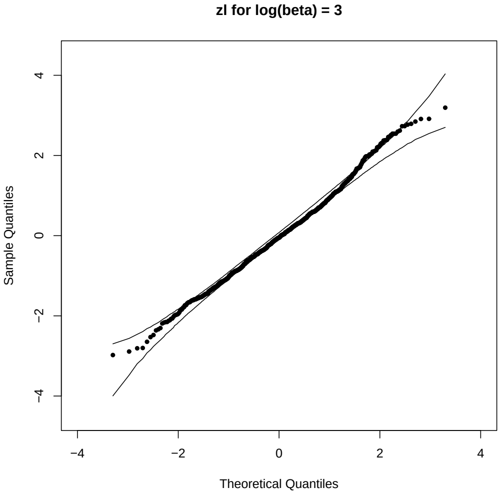
Example: Bayesian multiscale analysis of time series data



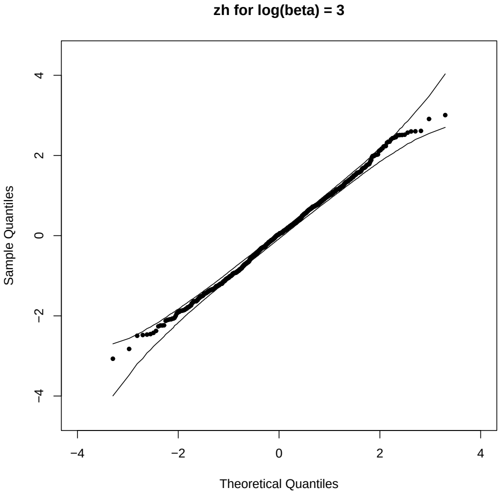
Example: Bayesian multiscale analysis of time series data



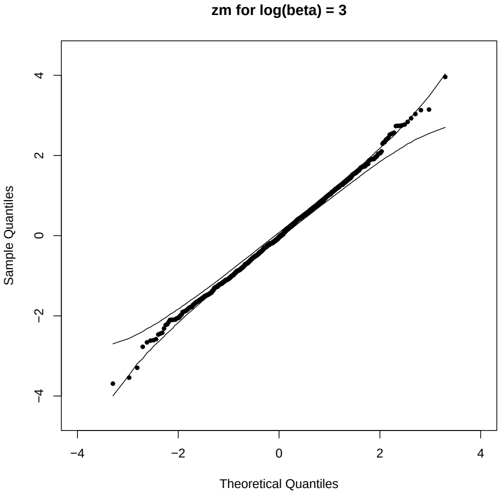
Example: Bayesian multiscale analysis of time series data



Example: Bayesian multiscale analysis of time series data



Example: Bayesian multiscale analysis of time series data



Disease mapping: The BYM-model

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

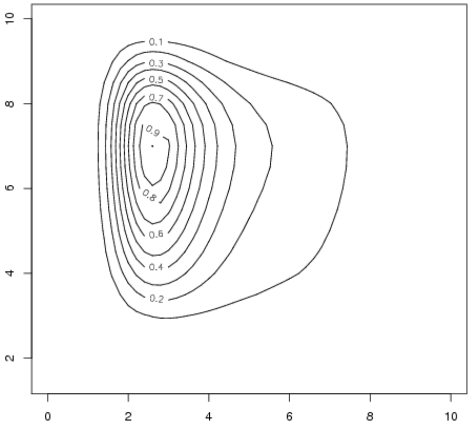
- A hard case: Insulin Dependent Diabetes Mellitus in 366 districts of Sardinia. Few counts.
- $\dim(\theta) = 2$.

Disease mapping: The BYM-model

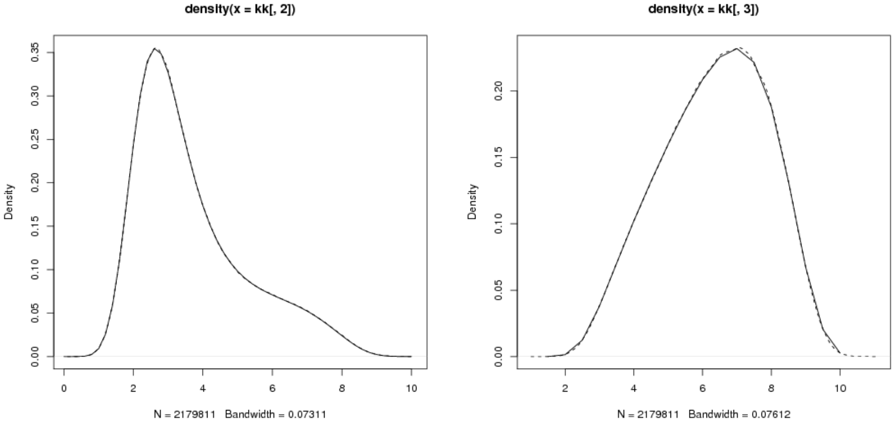
- Data $y_i \sim \text{Poisson}(E_i exp(\eta_i))$
- Log-relative risk $\eta_i = u_i + v_i$
- Structured component $\mathbf{u}$
- Unstructured component $\mathbf{v}$
- Log-precisions $\log \kappa_u$ and $\log \kappa_v$

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

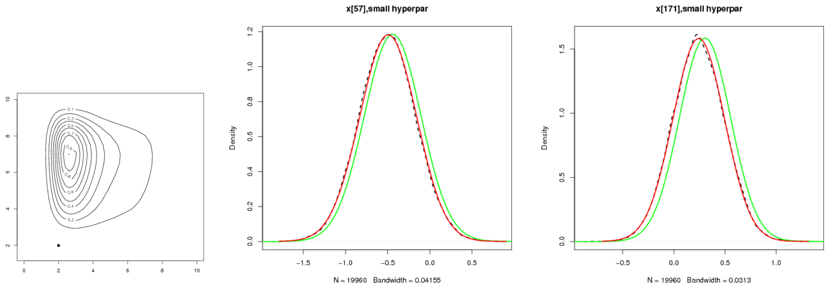
Marginals for $\theta|\mathbf{y}$



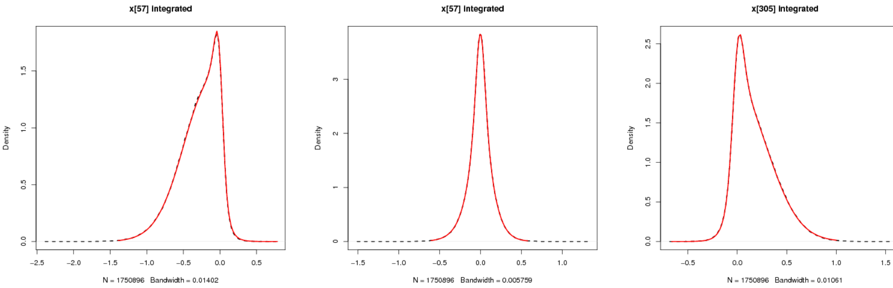
Marginals for $\theta|\mathbf{y}$



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:

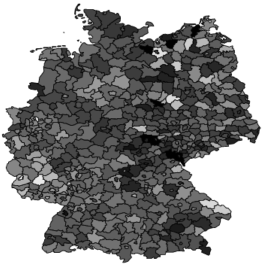
$$\pi(f) \propto \kappa^{(n-2)/2} \exp(-\frac{\kappa}{2} \sum_i (f_{i+1} - 2f_i + f_{i-1})^2)$$

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

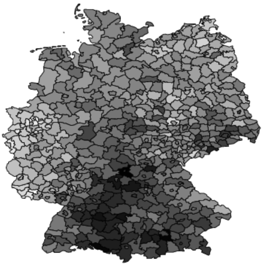
$$\mathbf{x} = (\mu, \mathbf{u}, \boldsymbol{\eta}, \mathbf{f}) \mid \text{hyperparameters} \sim \text{GMRF} \qquad (2)$$

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

Example: Larynx cancer with smoking covariate

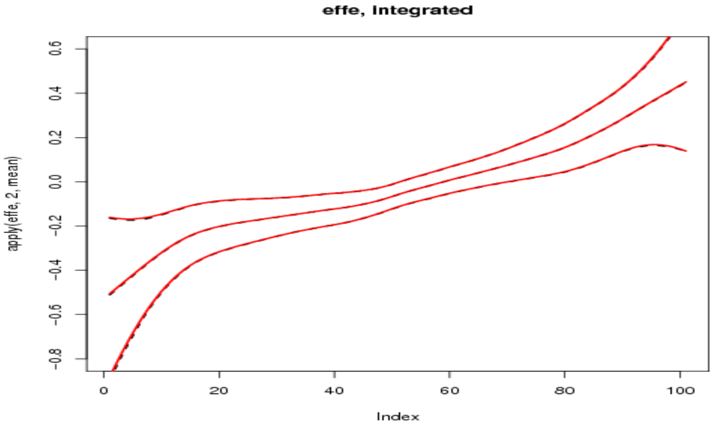


Larynx SMR



Smoking covariate

Example: Larynx cancer with smoking covariate



Example: 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: 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: 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: 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: Spatial GLMs

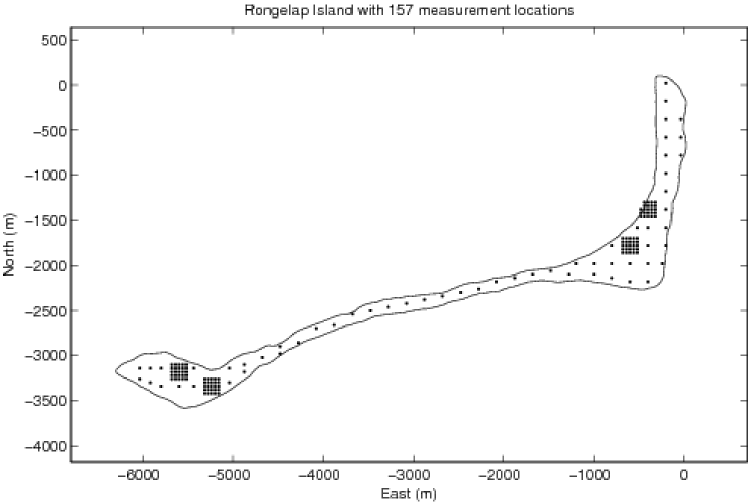
Model

- 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.

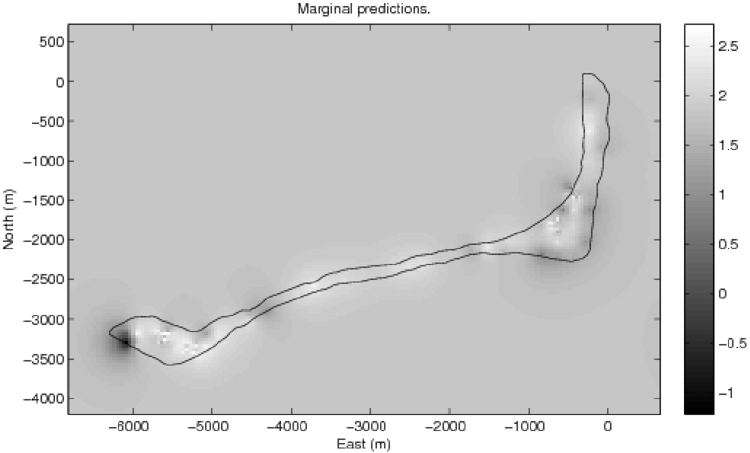
Solve using

- 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.

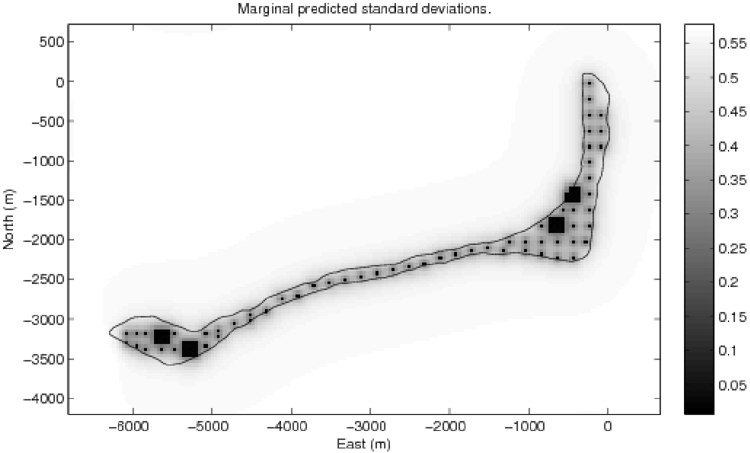
Example: Rongelap data



Example: Rongelap data, results



Example: Rongelap data, results



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

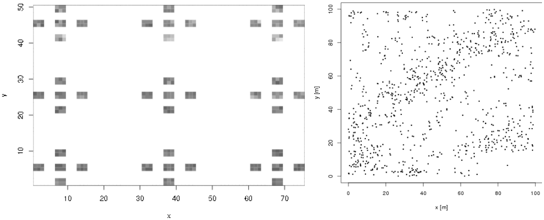
Spatial GLMs: Summary

- 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

Example: log-Gaussian Cox-processes



Again, excellent results!

Example: Space-time models

- Not yet
- We see no reasons for this not to go fine as well
- Only a change in the covariance-structure in the model

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
  - Obtain in practice "exact" results
  - *Relative* error only
- Computational convenient for large n
  - **GMRFs:** sparse matrix computations
  - **Stationary Gaussian fields:** DFT computations
  - **Fast:** minutes to compute all marginals
  - **Near instant:** use the GMRF-approximation. Error check.
  - **Parallel computing** excellent suited

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
  - Obtain in practice "exact" results
  - *Relative* error only
- Computational convenient for large n
  - **GMRFs:** sparse matrix computations
  - **Stationary Gaussian fields:** DFT computations
  - **Fast:** minutes to compute all marginals
  - **Near instant:** use the GMRF-approximation. Error check.
  - **Parallel computing** excellent suited

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
  - Obtain in practice "exact" results
  - *Relative* error only
- Computational convenient for large n
  - **GMRFs:** sparse matrix computations
  - **Stationary Gaussian fields:** DFT computations
  - **Fast:** minutes to compute all marginals
  - **Near instant:** use the GMRF-approximation. Error check.
  - **Parallel computing** excellent suited

Summary and discussion II

- Generic routines:
  - All GMRF-examples use the same library: less coding and less "errors"
  - Well suited for constructing (black-box) packages for inference
  - **Personal view:** do not use MCMC when INLA is appropriate
- Conditions apply
  - dim(θ) is not to high
  - Marginals only. Bi- and tri-variate marginals are also OK.
  - Can always construct counter-examples where INLA breaks down

Summary and discussion II

- Generic routines:
  - All GMRF-examples use the same library: less coding and less "errors"
  - Well suited for constructing (black-box) packages for inference
  - **Personal view:** do not use MCMC when INLA is appropriate
- Conditions apply
  - $\dim(\boldsymbol{\theta})$ is not to high
  - Marginals only. Bi- and tri-variate marginals are also OK.
  - Can always construct counter-examples where INLA breaks down