Lecture 10 Review: MCMC Idea

Situation:

- Given a target distribution $f(x)$
- Want to generate samples from $f(x)$

Idea:

- construct a Markov chain $\{X_i\}_{i=1}^{\infty}$ so that $\lim_{i\to\infty} P(X_i = x) = f(x)$
- simulate the Markov chain for many iterations
- for m large enough $x_m, x_{m+1}, \ldots$ are (essentially) from $f(x)$

Review: How to construct the Markov chain

How to construct such a Markov chain? ($x \in \Omega$ discrete)

- Markov chain transition probabilities:

  $P(y|x) = P(X_{i+1} = y|X_i = x)$

- Need to have

$$f(y) = \sum_{x \in \Omega} f(x)P(y|x) \text{ for all } y \in \Omega$$

- Sufficient condition: Detailed balance condition

$$f(x)P(y|x) = f(y)P(x|y) \text{for all} x, y \in \Omega$$

Review: How to construct the Markov chain

Metropolis-Hastings setup for $P(y|x)$:

$$P(y|x) = Q(y|x)\alpha(y|x) \text{ when } y \neq x$$
$$P(x|x) = 1 - \sum_{y \neq x} Q(y|x)\alpha(y|x) \text{ when } y = x$$

where

$$\alpha(y|x) = \min\left\{1, \frac{f(y)}{f(x)}\frac{Q(x|y)}{Q(y|x)}\right\}$$

Review: Common proposal types

- Independent proposals: $Q(y|x) = q(y)$
  - usually not a good alternative (alone)
- Random walk proposals: $Q(y|x) = Q(x|y)$
  - used a lot
  - often includes tuning parameter for step size
- Gibbs updates: $Q(y^j|x^j, \mathrm{x}^{-j}) = f(x^j|\mathrm{x}^{-j})$
  - used a lot
  - the proposal density is the full conditional
  - no tuning parameter
  - acceptance rate 1
  - can be combined with MH update

Variance of the MCMC estimator

Recall: We want to estimate $\mu = \int g(x)\pi(x)\, dx$ with $\hat{\mu} = \frac{1}{n}\sum g(x_i)$ where $x_i \sim \pi(x)$.

In standard MC we have

$$x_1, x_2, \ldots, x_n \sim \pi(x), \text{ i.i.d.}$$

This gives

$$\mathrm{E}(\hat{\mu}) = \mu \text{ and } \mathrm{Var}(\hat{\mu}) = \frac{\mathrm{Var}(g(X))}{n}$$

We can estimate the variance $\mathrm{Var}(\hat{\mu})$ as

$$\widehat{\mathrm{Var}(\hat{\mu})} = \frac{\widehat{\mathrm{Var}(g(X))}}{n}$$

$$\widehat{\mathrm{Var}(g(X))} = \frac{1}{n-1}\sum (g(x_i) - \hat{\mu})^2$$

MCMC gives dependent samples, what is the variance then??

Autocorrelation

Let $x_1, \ldots, x_N$, where N is the number of samples, be our MCMC chain.

The lag k autocorrelation $\rho(k)$ is the correlation between every draw and its k-th lag. For N reasonably large

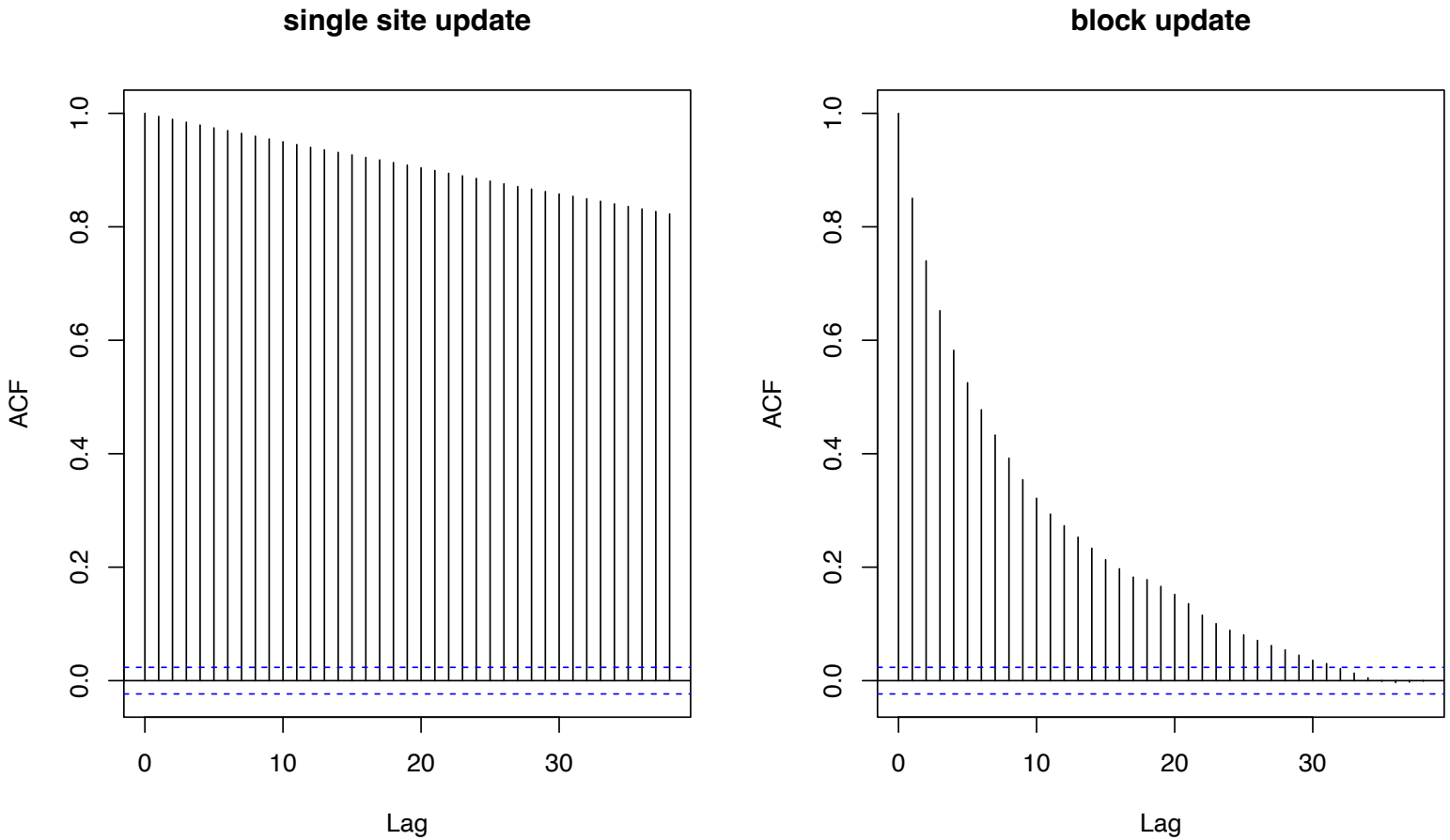
$$\rho(k) \approx \frac{\sum_{i=1}^{N-k}(x_i - \bar{x})(x_{i+k} - \bar{x})}{\sum_{i=1}^{N}(x_i - \bar{x})^2},$$

where $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ is the overall mean.

- With increasing lag k we expect lower autocorrelations.
- If autocorrelation is still relatively high for higher values of k, this indicates high degree of correlation between our draws and slow mixing.

Example: Korsbetningen

Autocorrelation function for N (after discarding the burn-in period)



Effective sample size

A useful measure to compare the performance of different MCMC samplers is the effective sample size (ESS) Kass et al. (1998) American Statistician 52, 93–100..

- The ESS is the estimated number of independent samples needed to obtain a parameter estimate with the same precision as the MCMC estimate based on N dependent samples.

$$\text{ESS} = \frac{N}{\tau}, \quad \tau = 1 + 2 \cdot \sum_{k=1}^{\infty} \rho(k),$$

where τ is the autocorrelation time and $\rho(k)$ the autocorrelation at lag k.

Estimate of ESS

$$\text{ESS} = \frac{N}{\tau}, \quad \tau = 1 + 2 \cdot \sum_{k=1}^{\infty} \rho(k),$$

Estimate τ as

$$\tau = 1 + 2 \cdot \sum_{k=1}^{m} \hat{\rho}(k)$$

where $\hat{\rho}(k)$ is the sample autocorrelation function at lag k, and m is choosen to fullfill some criteria.

Different criteria exists.

Example: Korsbetningen - Effective sample size (ESS)

```
> library(coda)
> nsamples
[1] 6000
> ## single site
> effectiveSize(as.mcmc(res1))
       N    theta
15.10473 11.50380
> ## block update
> effectiveSize(as.mcmc(res2))
       N    theta
357.7599 626.0842
>
```

The precision of the MCMC estimate of the posterior mean of N based on 8000 samples from a single site update is a good as taking 16 independent samples!

Geweke diagnostics

The MCMC chain is divided into two windows

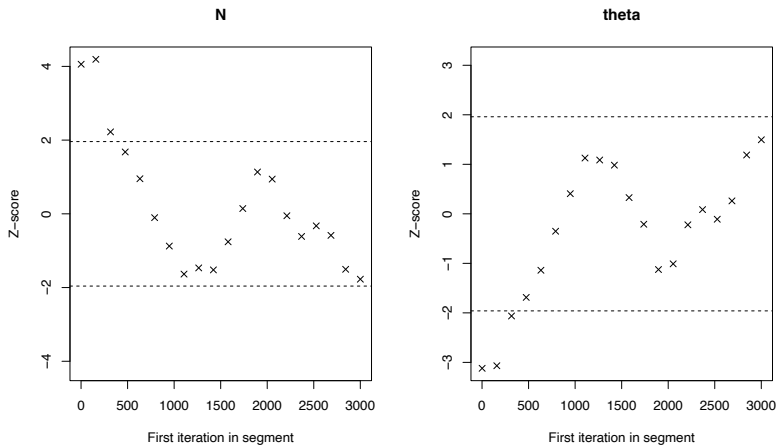
- the first x%, and
- the last y% of the iterates

(coda default: $x = 10$, $y = 50$). For both windows the mean is calculated.

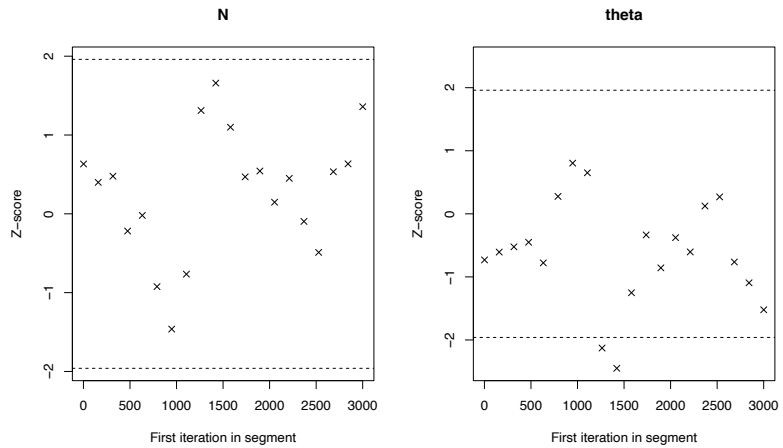
If the chain is stationary both values should be equal and Geweke's test statistic (z-score) follows an asymptotical standard normal distribution.

Example: Korsbetningen - Geweke plot

Single Site



Block Update



Further reading

There are several convergence diagnostics:

- some are based on a single Markov chain run
- some are based on several Markov chain runs

There are no guarantees!

For further reading see for example

- Gilks, W. R., Richardson, S. and Spiegelhalter, D.J. (1996) *Markov Chain Monte Carlo in Practice, Chapman & Hall, London*,

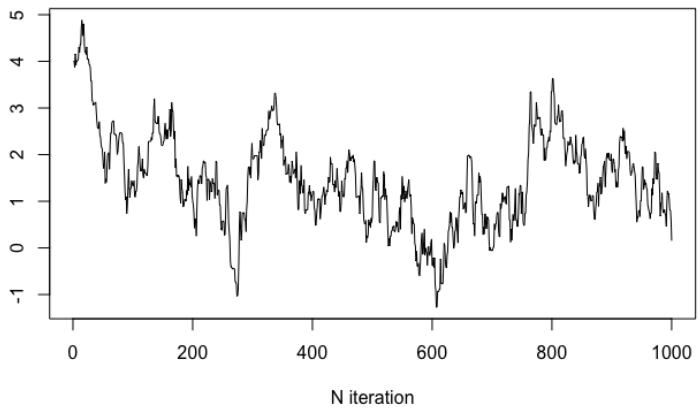
Different approaches are implemented in the

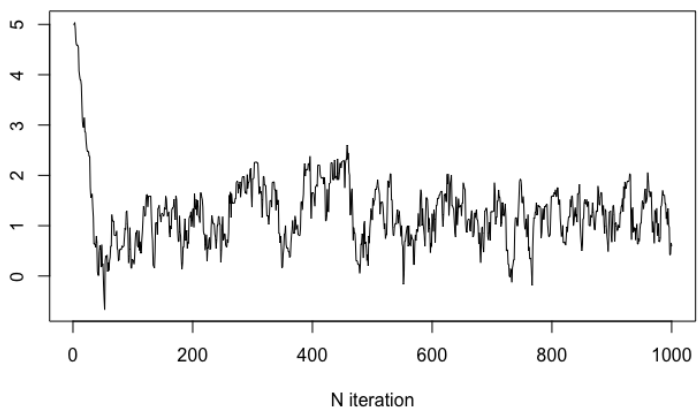
Review: Convergence diagnostic

Has the MC converged?

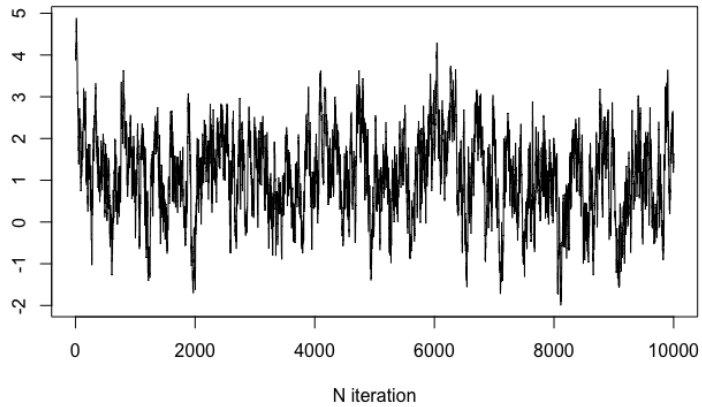
- Formal convergence diagnostics exists
  - some based on a single Markov chain run
  - some based on several Markov chain runs
- Standard way to assess convergence is to look at the traceplot
- If some properties of the target distribution is known: use it to check convergence!
- All convergence diagnostics can (and do) fail

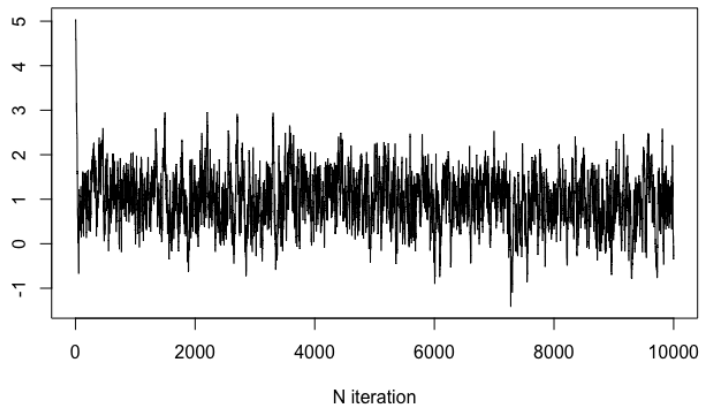
Has bivariate this MC converged?



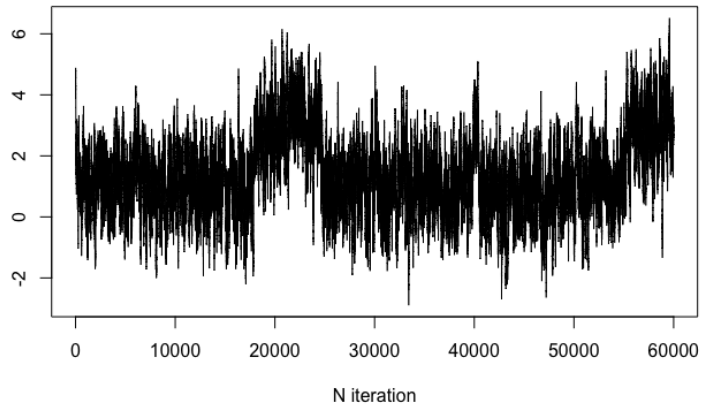


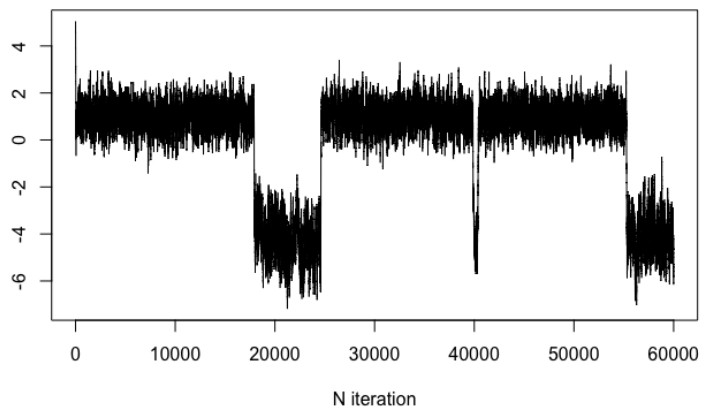
Has bivariate this MC converged?



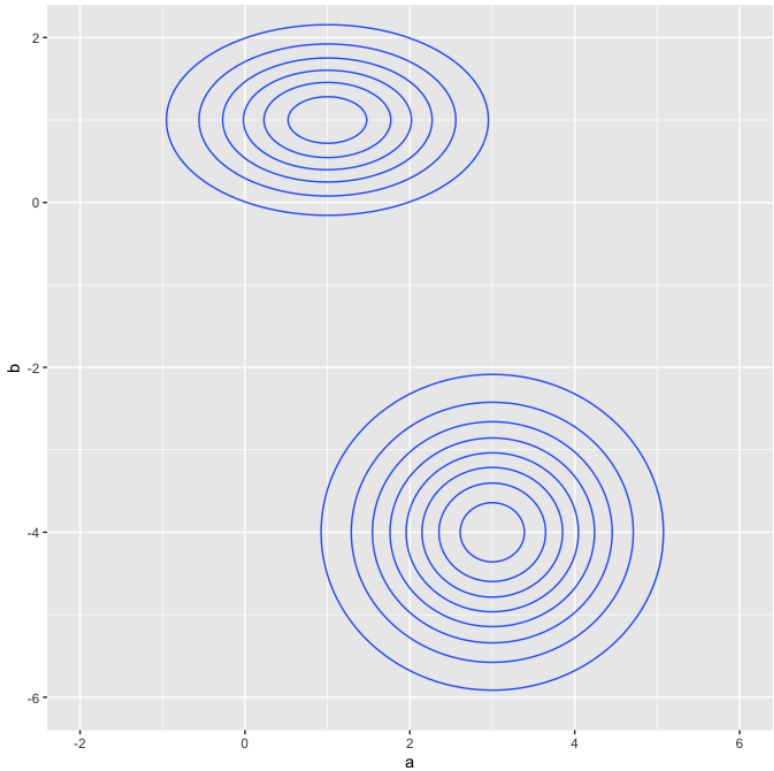


Has bivariate this MC converged?





Has bivariate this MC converged?



This is how the distribution looks like.

Used a RW proposal $\mathcal{N}(0, 0.3^2 I)$

Summary

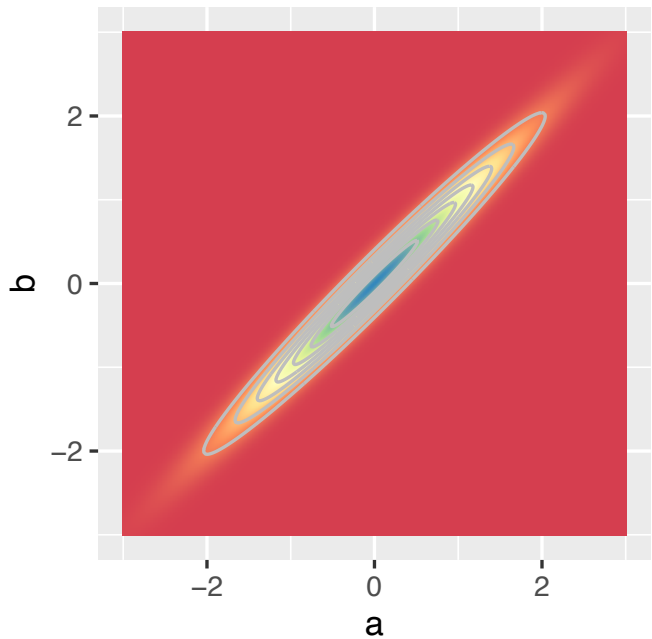
- Diagnostics cannot guarantee that chain has converged
- Can indicate that it has not converged

Solutions?

- Run longer and thin output
- Reparametrize model
- "Block" correlated variables together
  - Joint update might be more efficient however for some parameter combination the acceptance rate can be very slow!
- integrate out variables
- ...

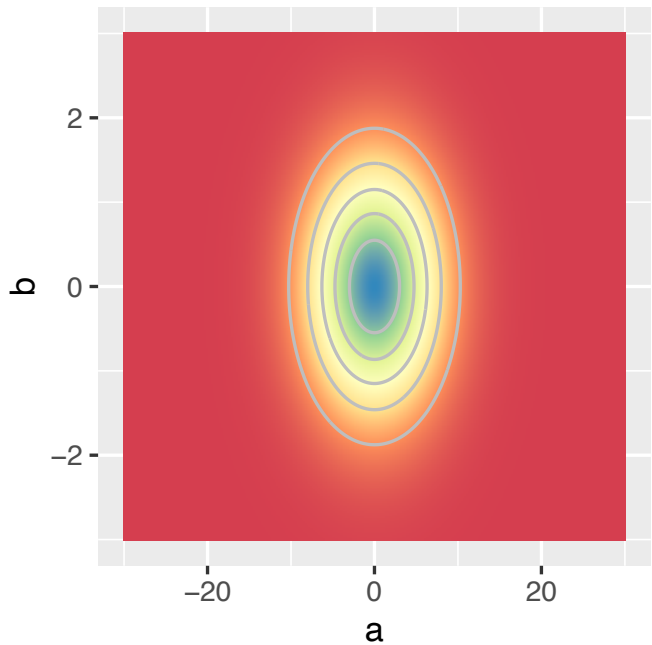
Typical MCMC problems

- Properties of $f(x)$ that may make MCMC difficult
  - strong dependency between variables



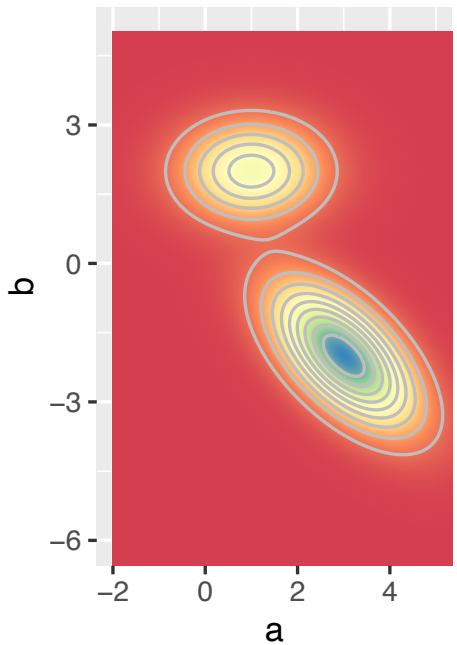
Typical MCMC problems

- Properties of $f(x)$ that may make MCMC difficult
  - strong dependency between variables
  - different scales on different variables



Typical MCMC problems

- Properties of $f(x)$ that may make MCMC difficult
  - strong dependency between variables
  - different scales on different variables
  - several modes



Typical MCMC problems

- Properties of $f(x)$ that may make MCMC difficult
  - strong dependency between variables
  - different scales on different variables
  - several modes
- In toy examples: this is not a problem
  - we know how $f(x)$ looks like
- In real problems: this may be difficult
  - we have a formula for $f(x)$
  - we don't know how $f(x)$ looks like

MCMC

- As computing power has increased, MCMC and Bayesian statistics has become increasingly accessible
- Rise in MCMC software (e.g. BUGS, WinBUGS, OpenBUGS, JAGS, Stan)
- MCMC is very general and can be applied to "any" model

MCMC

- As computing power has increased, MCMC and Bayesian statistics has become increasingly accessible
- Rise in MCMC software (e.g. BUGS, WinBUGS, OpenBUGS, JAGS, Stan)
- MCMC is very general and can be applied to "any" model
- However:
  - Even if in theory MCMC can provide (nearly) exact inference given perfect convergence and MC error $\rightarrow 0$, in practice this must be balanced with model complexity and running time
  - This is particularly an issue for problems characterised by large data or very complex structure (e.g. hierarchical models)
  - Testing model sensitivity to the prior and doing model validation can take too long to be practical

What is INLA?

Integrated Nested Laplace Approximation

The short answer:

INLA is a fast method to do Bayesian inference with latent Gaussian models and `R-INLA` *is an R-package that implements this method with a flexible and simple interface*

A (much) longer answer can be found in:

Rue, Martino, and Chopin (2009) "Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations." *Journal of the royal statistical society*: Series B. 319-392

Ingredients of INLA

- Latent Gaussian Models
  - Class of models where INLA can be applied
- Gaussian Markov Random Fields (GMRFs)
  - Sparse matrix computations
- Laplace Approximation
  - Method of approximating posterior

Integrated Nested Laplace Approximation (INLA)

What is it?

Why?

When can it be applied?

How does it work?

How do we use it?

Integrated Nested Laplace Approximation (INLA)

What is it? A numerical method to do fast approximate bayesian inference

Why?

When can it be applied?

How does it work?

How do we use it?

Integrated Nested Laplace Approximation (INLA)

What is it? A numerical method to do fast approximate bayesian inference

Why? MCMC takes too long to converge

When can it be applied?

How does it work?

How do we use it?

Integrated Nested Laplace Approximation (INLA)

What is it? A numerical method to do fast approximate bayesian inference

Why? MCMC takes too long to converge

When can it be applied? The (wide) class of Latent Gaussian Models

How does it work?

How do we use it?

Integrated Nested Laplace Approximation (INLA)

What is it? A numerical method to do fast approximate bayesian inference

Why? MCMC takes too long to converge

When can it be applied? The (wide) class of Latent Gaussian Models

How does it work? Uses GMRF and sparse matrix computations, Laplace approximation, numerical integration

How do we use it?

Integrated Nested Laplace Approximation (INLA)

What is it? A numerical method to do fast approximate bayesian inference

Why? MCMC takes too long to converge

When can it be applied? The (wide) class of Latent Gaussian Models

How does it work? Uses GMRF and sparse matrix computations, Laplace approximation, numerical integration

How do we use it? Already implemented in `R-INLA`

Latent Gaussian Models: a Unified framework

Observations: $\boldsymbol{y}$

Latent field: $\boldsymbol{x}$

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

Latent Gaussian Models: a Unified framework

Observations: $\boldsymbol{y}$ Assumed conditionally independent given $\boldsymbol{x}$ and $\boldsymbol{\theta}_1$

$$\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}_1 \sim \prod_i \pi(y_i|x_i, , \boldsymbol{\theta}).$$

Latent field: $\boldsymbol{x}$

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

Latent Gaussian Models: a Unified framework

Observations: $\boldsymbol{y}$ Assumed conditionally independent given $\boldsymbol{x}$ and $\boldsymbol{\theta}_1$

$$\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}_1 \sim \prod_i \pi(y_i|x_i, , \boldsymbol{\theta}).$$

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

$$\boldsymbol{x}|\boldsymbol{\theta}_1 \sim \mathcal{N}(0, \boldsymbol{Q}(\boldsymbol{\theta}_2)^{-1})$$

The latent field $\boldsymbol{x}$ can be large $(10^1 - 10^6)$

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

Latent Gaussian Models: a Unified framework

Observations: $\boldsymbol{y}$ Assumed conditionally independent given $\boldsymbol{x}$ and $\boldsymbol{\theta}_1$

$$\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}_1 \sim \prod_i \pi(y_i|x_i, , \boldsymbol{\theta}).$$

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

$$\boldsymbol{x}|\boldsymbol{\theta}_1 \sim \mathcal{N}(0, \boldsymbol{Q}(\boldsymbol{\theta}_2)^{-1})$$

The latent field $\boldsymbol{x}$ can be large $(10^1 - 10^6)$

Hyperparameters: $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ Precision parameters of the Gaussian field and parameters of the likelihood

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

The vector $\boldsymbol{\theta}$ is usually small (1-10)

Latent Gaussian models

A very general way of specifying the problem is by modelling the mean for the i-th unit by means of an additive linear predictor, defined on a suitable scale (e.g. logistic for binomial data)

$$\eta_i = \alpha + \sum_{l=1}^{L} f_l(u_{li}) + \sum_{k=1}^{K} \beta_k z_{ki} + \epsilon_i$$

where

- α is the intercept
- $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_K)$ quantify the effect of $\boldsymbol{x} = (x_1, \ldots, x_K)$ on the response
- $\boldsymbol{f} = (f_1, \ldots, f_L)$ is a set of functions defined in terms of some covariates $\boldsymbol{z} = (z_1, \ldots, z_K)$

And assume

$$\boldsymbol{x} = (\alpha, \boldsymbol{\beta}, \boldsymbol{f}) \sim \mathcal{N}(0, \boldsymbol{Q}(\theta)^{-1})$$

Many commonly used models can be written as LGM:

- Multiple regression
- Generalized linear model (GLM)
- Generalized additive model (GAM)
- Generalized additive/linear mixed model (GAMM, GLMM)

Many commonly used models can be written as LGM:

- Multiple regression

$$\eta_i = E(y_i) = \alpha + \sum_{k=1}^{K} \beta_k z_{ki}$$

  - α: Intercept
  - β: Linear effects of covariates z
- Generalized linear model (GLM)
- Generalized additive model (GAM)
- Generalized additive/linear mixed model (GAMM, GLMM)

Many commonly used models can be written as LGM:

- Multiple regression
- Generalized linear model (GLM)

$$\eta_i = g(\mu_i) = \alpha + \sum_{k=1}^{K} \beta_k z_{ki}$$

  - $g(\cdot)$: link function
  - α: Intercept
  - β: Linear effects of covariates z
- Generalized additive model (GAM)
- Generalized additive/linear mixed model (GAMM, GLMM)

Many commonly used models can be written as LGM:

- Multiple regression
- Generalized linear model (GLM)
- Generalized additive model (GAM)

$$\eta_i = g(\mu_i) = \alpha + \sum_{l=1}^{L} f_l(u_{li})$$

  - $g(\cdot)$: link function
  - α: Intercept
  - $\{f_l(\cdot)\}$: Non-linear smooth effects of covariates u_l
- Generalized additive/linear mixed model (GAMM, GLMM)

Many commonly used models can be written as LGM:

- Multiple regression
- Generalized linear model (GLM)
- Generalized additive model (GAM)
- Generalized additive/linear mixed model (GAMM, GLMM)

$$\eta_i = g(\mu_i) = \alpha + \sum_{l=1}^{L} f_l(u_{li})$$

  - $g(\cdot)$: link function
  - α: Intercept
  - β: Linear effects of covariates z
  - $\{f_l(\cdot)\}$: Non-linear smooth effects of covariates u_l

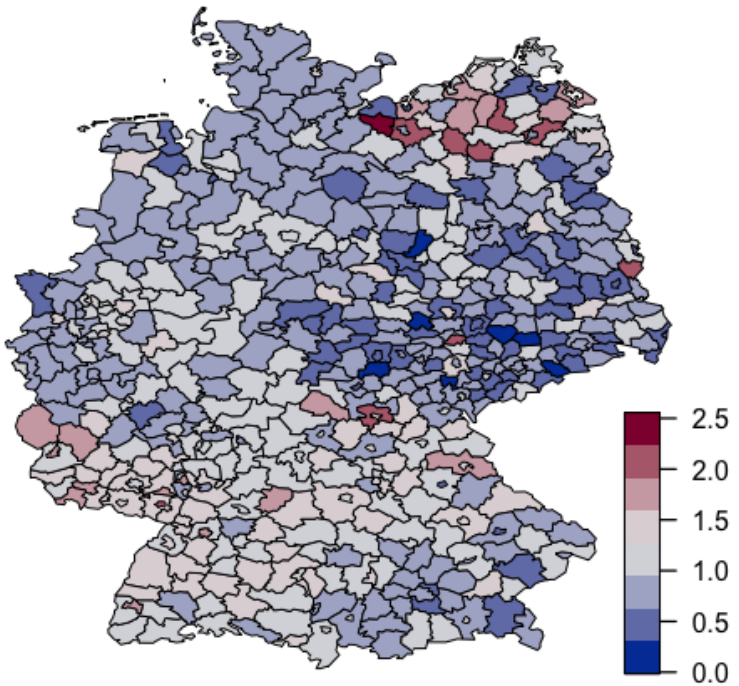
Some more example of LGM

- Disease Mapping
- Geostatistical models
- Survival models
- Stochastic volatility models
- Spatial and spatio-temporal models
- Spline smoothing
- +++

Example: Disease Mapping in Germany

We observed larynx cancer mortality counts for males in 544 district of Germany from 1986 to 1990 and want to make a model.

- y_i The count in disctrict i
- E_i An offset, expected number of cases in district i
- c_i A covariate (level of smoking consumption in district i)
- s_i Spatial location i (district)



Example: Disease Mapping in Germany

- Poisson likelihood

$$y_i|\eta_i \sim \text{Poisson}(E_i \exp(\eta_i))$$

- Laten Gaussian model

$$\eta_i = \mu + f_s(s_i) + f(c_i) + u_i$$

The latent field is $\boldsymbol{x} = \{\mu, (f_s(\cdot)), (f(\cdot)), u_1, \ldots, u_n\}$

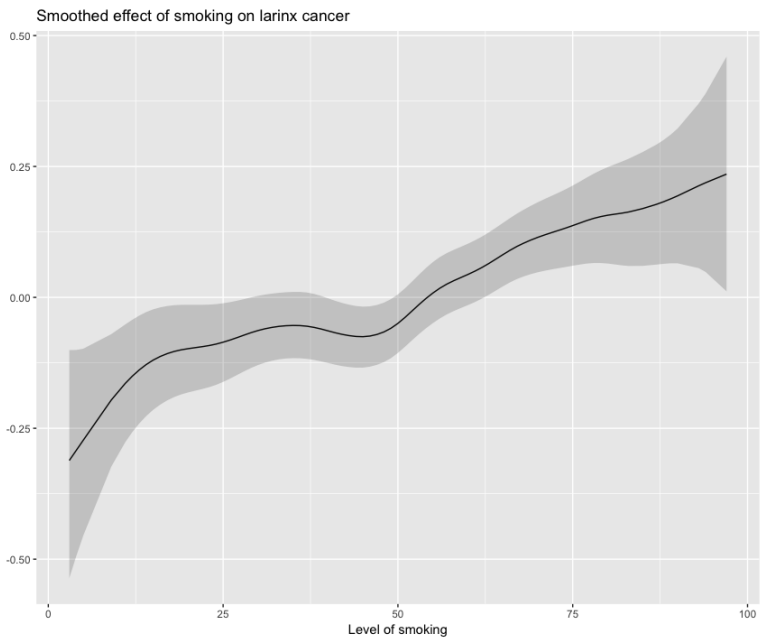
- Hyperparameters: τ_c , τ_f , τ_η : The precisions (inverse variances) of the covariate effect, spatial effect and unstructured effect, respectively.

Example: Disease Mapping in Germany

Posterior of interest

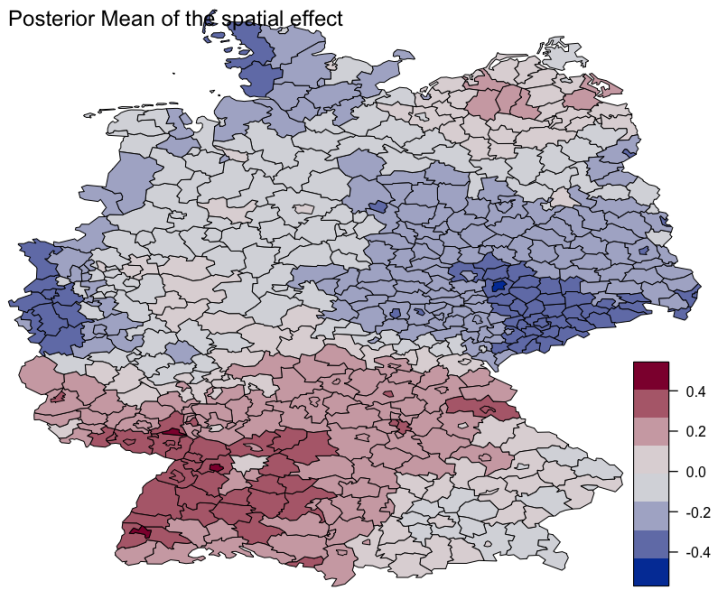
Effect of the covariate:

$$\pi(f(c_i)|\boldsymbol{y})$$



Structured spatial effect:

$$\pi(f_s(s_i)|\boldsymbol{y})$$



INLA computing scheme

From the posterior $\pi(\boldsymbol{\theta}, \boldsymbol{x}|\boldsymbol{y})$ we are mostly interested in

$$\pi(\theta_j|\boldsymbol{y}) \text{ and } \pi(x_i|\boldsymbol{y})$$

INLA computing scheme

From the posterior $\pi(\boldsymbol{\theta}, \boldsymbol{x}|\boldsymbol{y})$ we are mostly interested in

$$\pi(\theta_j|\boldsymbol{y}) \text{ and } \pi(x_i|\boldsymbol{y})$$

- Approximate $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ using Laplace approximation
  - Use numerical integration to approximate

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

  - This integral is not difficult to solve (dimension of $\boldsymbol{\theta}$ is small)

INLA computing scheme

From the posterior $\pi(\boldsymbol{\theta}, \boldsymbol{x}|\boldsymbol{y})$ we are mostly interested in

$$\pi(\theta_j|\boldsymbol{y}) \text{ and } \pi(x_i|\boldsymbol{y})$$

- Approximate $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ using Laplace approximation
  - Use numerical integration to approximate
$$\pi(\theta_j|\boldsymbol{y}) = \int \pi(\boldsymbol{\theta}|\boldsymbol{y}) d\boldsymbol{\theta}_{-j}$$
  - This integral is not difficult to solve (dimension of $\boldsymbol{\theta}$ is small)
- Approximate $\pi(x_i|\boldsymbol{\theta}, \boldsymbol{y})$ using Laplace approximation
  - Use numerical integration to approximate
$$\pi(x_i|\boldsymbol{y}) = \int \pi(x_i|\boldsymbol{\theta}, \boldsymbol{y}) \pi(\boldsymbol{\theta}|\boldsymbol{y}) d\boldsymbol{\theta}$$
  - This integral is not difficult to solve (dimension of $\boldsymbol{\theta}$ is small)

Smoothing noisy observations

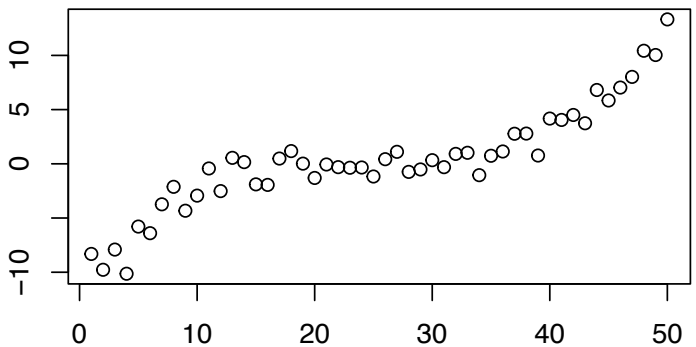
Assume

$$y_i = f(i) + \epsilon_i$$

where

$$\epsilon_i \sim \mathcal{N}(0, 1)$$

$f(i)$ smooth function of i



We have noisy observation, we want to recover the f function

Hierarchical Model

Data Gaussian Observations with known precision

$$y_i | x_i \sim \mathcal{N}(x_i, 1)$$

Latent Model : A Gaussian model for the smooth function (RW2 model)

$$\pi(\mathrm{x}|\theta) \propto \theta^{(n-2)/n} \exp\left\{ -\frac{\theta}{2} \sum_{i=2}^{n} (x_i - 2x_{i-1} + x_{i-2})^2 \right\}$$

Hyperparameter The precision of the smooth function θ. We assign a Gamma prior

$$\pi(\theta) \propto \theta^{a-1} \exp(-b\theta)$$

Posterior marginal for hyperparameter

We have that

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

so

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

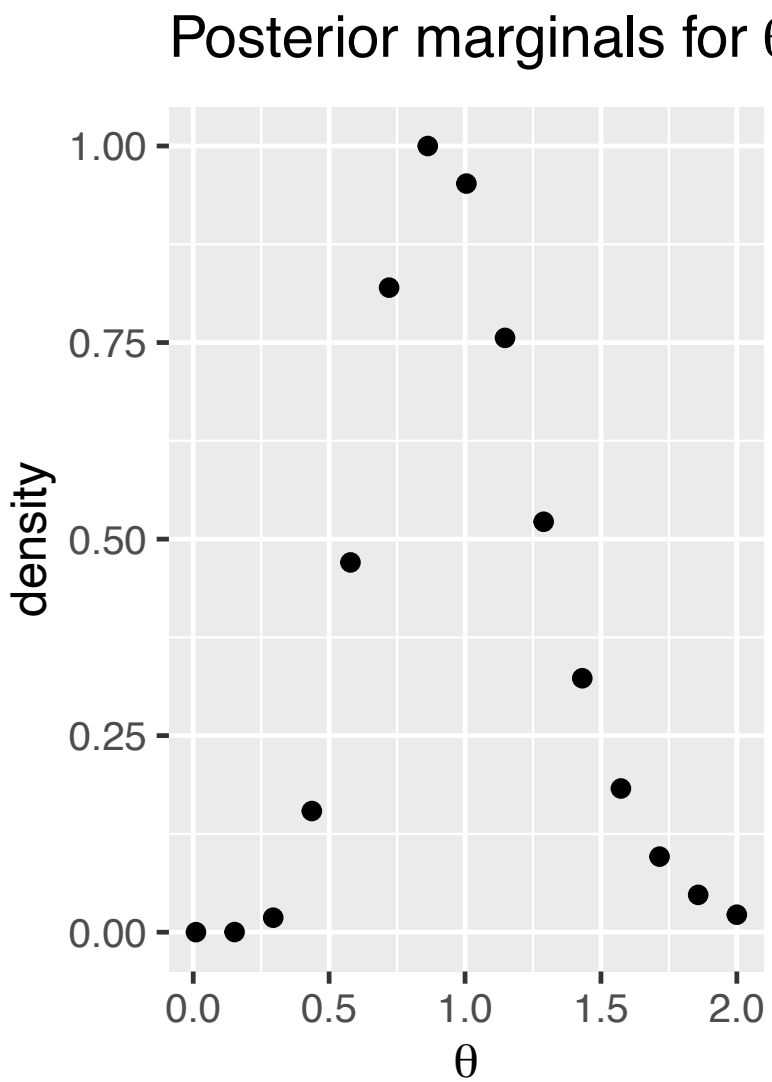
Since the likelihood is Gaussian, then $\pi(\mathsf{y}, \mathsf{x}|\theta)$ is also Gaussian. We have then:

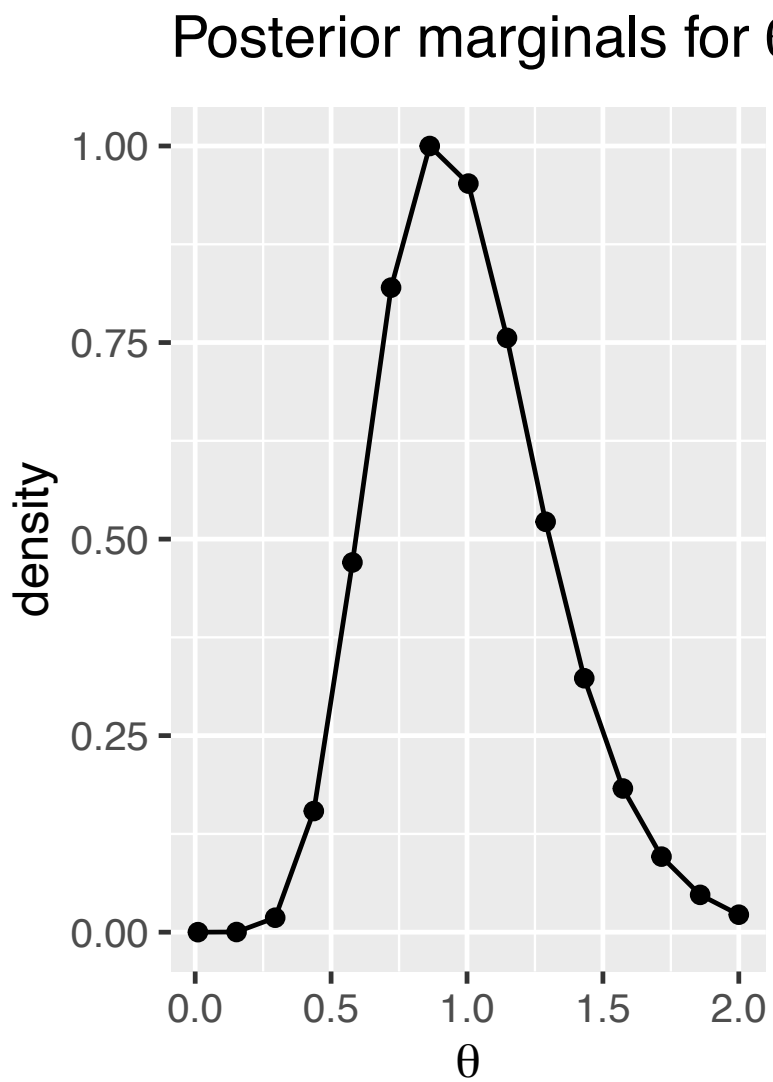
$$\pi(\theta|\mathsf{y}) \propto \frac{\overbrace{\pi(\mathsf{y}, \mathsf{x}|\theta)}^{\text{Gaussian}}\ \pi(\theta)}{\underbrace{\pi(\mathsf{x}|\theta, \mathsf{y})}_{\text{Gaussian}}}$$

This is valid for any x

Posterior marginal for the hyperparameter

Select a grid of points to represent the density $\pi(\theta|x)$





Posterior marginals for latent field

Again we have that

$$\mathsf{x}, \mathsf{y}|\theta \sim \mathsf{N}(\cdot, \cdot)$$

so also $\pi(x_i|\theta, \mathsf{y})$ is Gaussian!!

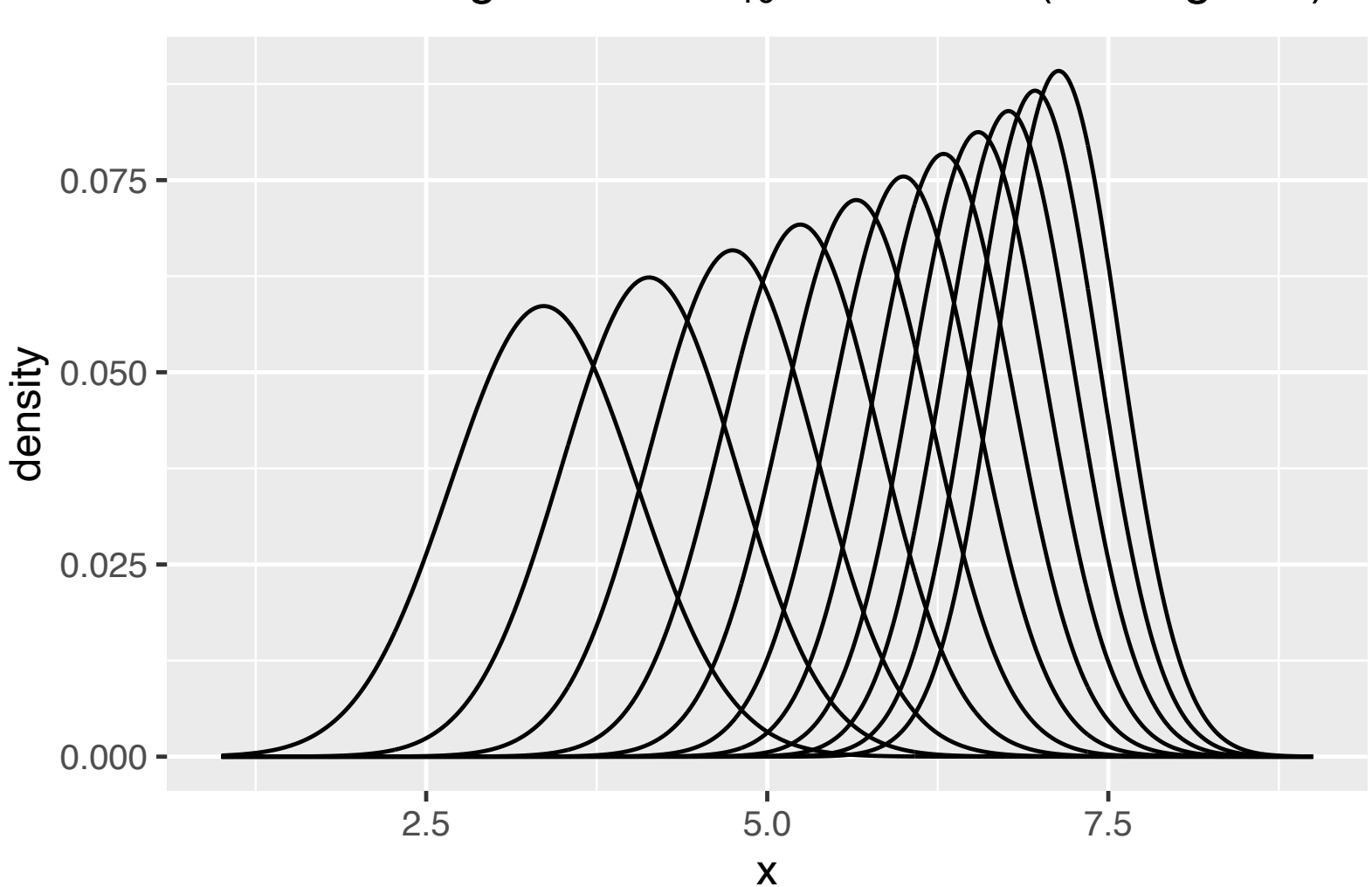
We compute

$$\begin{aligned} \pi(x_i|\mathsf{y}) &= \int \pi(x_i|\theta, \mathsf{y})\pi(\theta|\mathsf{y})d\theta \\ &\approx \sum_k \pi(x_i|\theta_k, \mathsf{y})\pi(\theta_k|\mathsf{y})\Delta_k \end{aligned}$$

where $\theta_k, k = 1, \ldots, K$ are the representative points of $\pi(\theta|\mathsf{y})$ and Δ_k are the corresponding weights

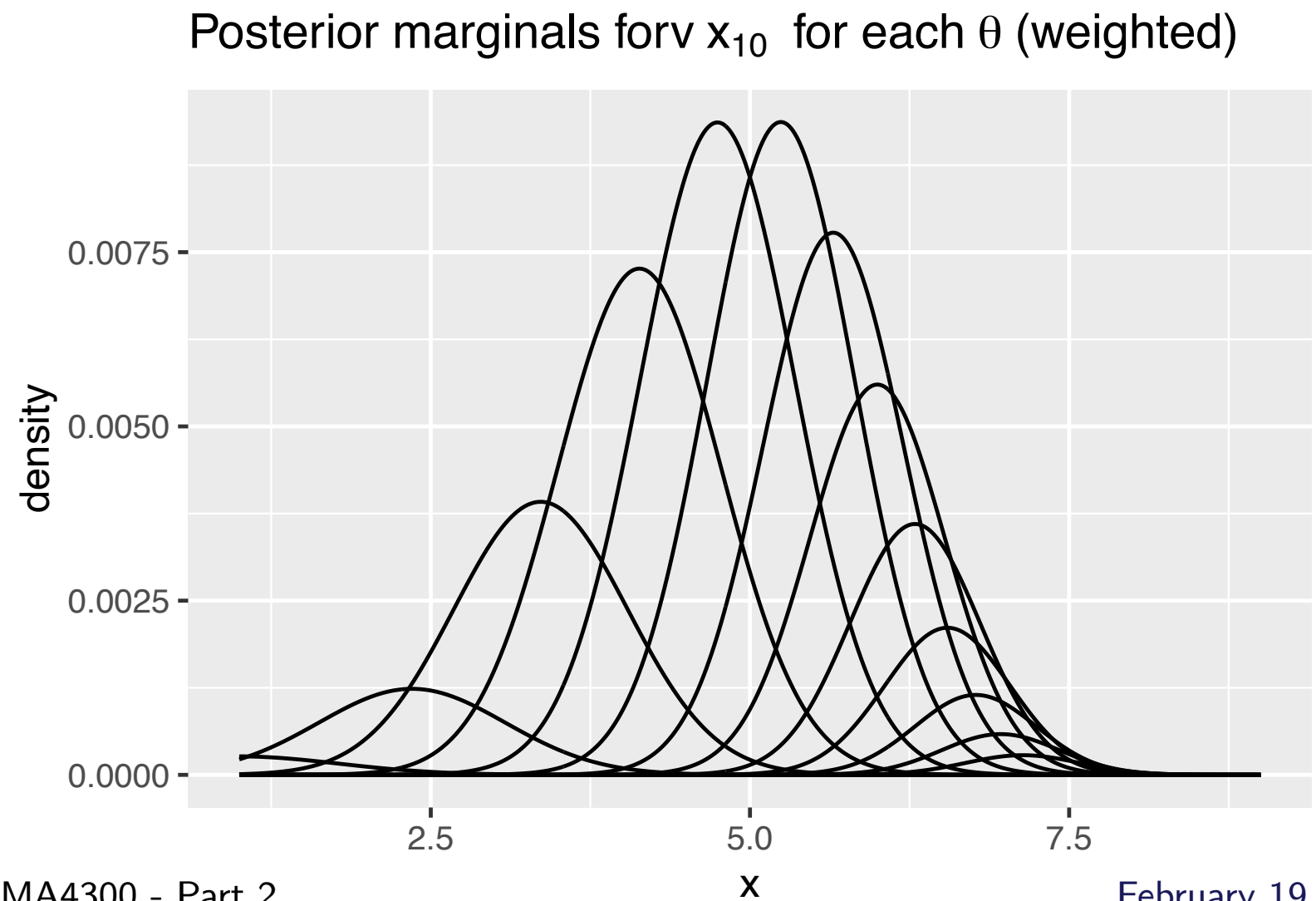
Posterior marginals for latent field

Compute the conditional posterior marginal for x_i given each θ_k



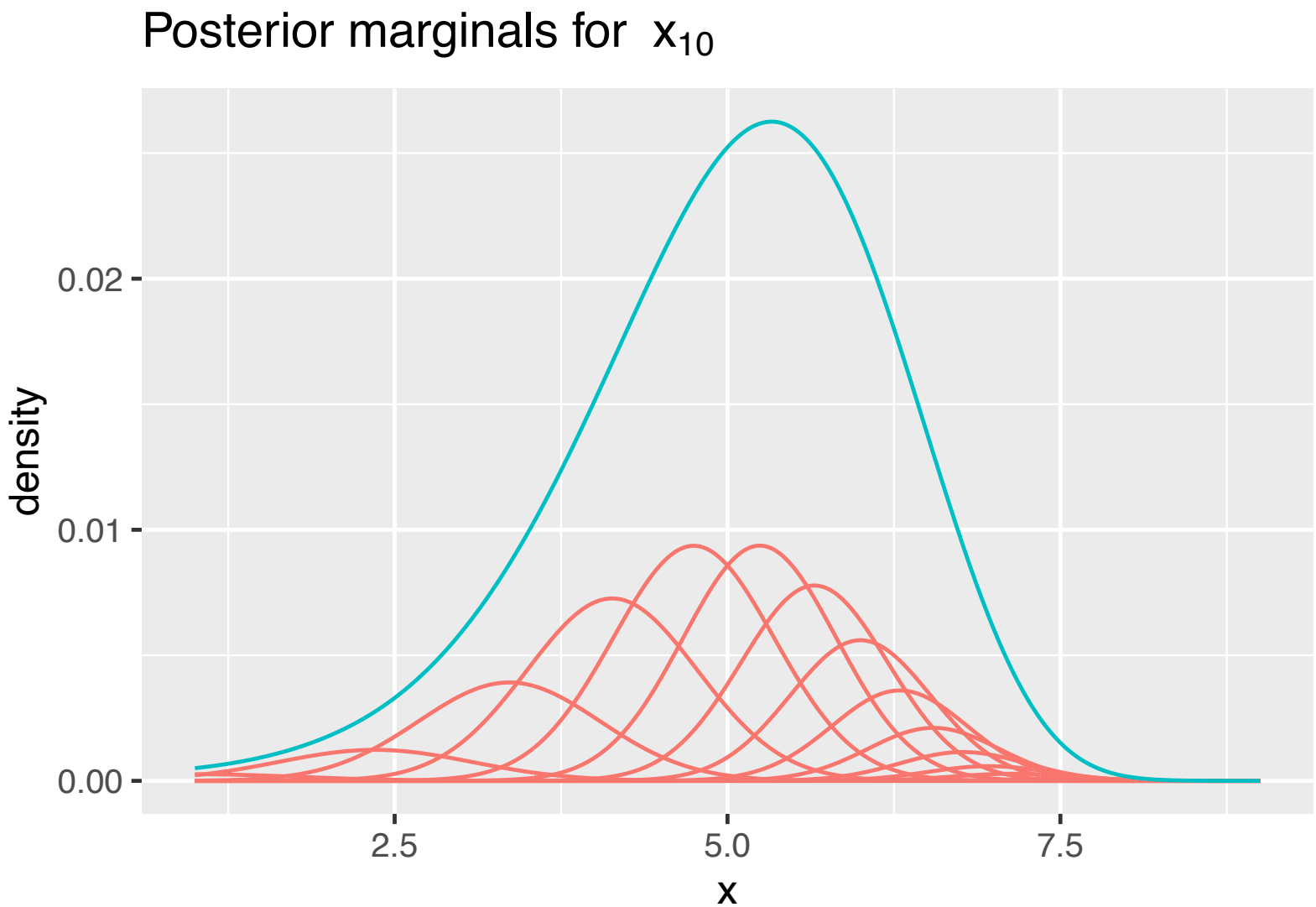
Posterior marginals for latent field

Weighted the conditional posterior marginal for $\pi(x_i|\theta_k, \mathrm{y})$ by $\pi(\theta_k|\mathrm{y})\acute{}_{\mathrm{k}}$



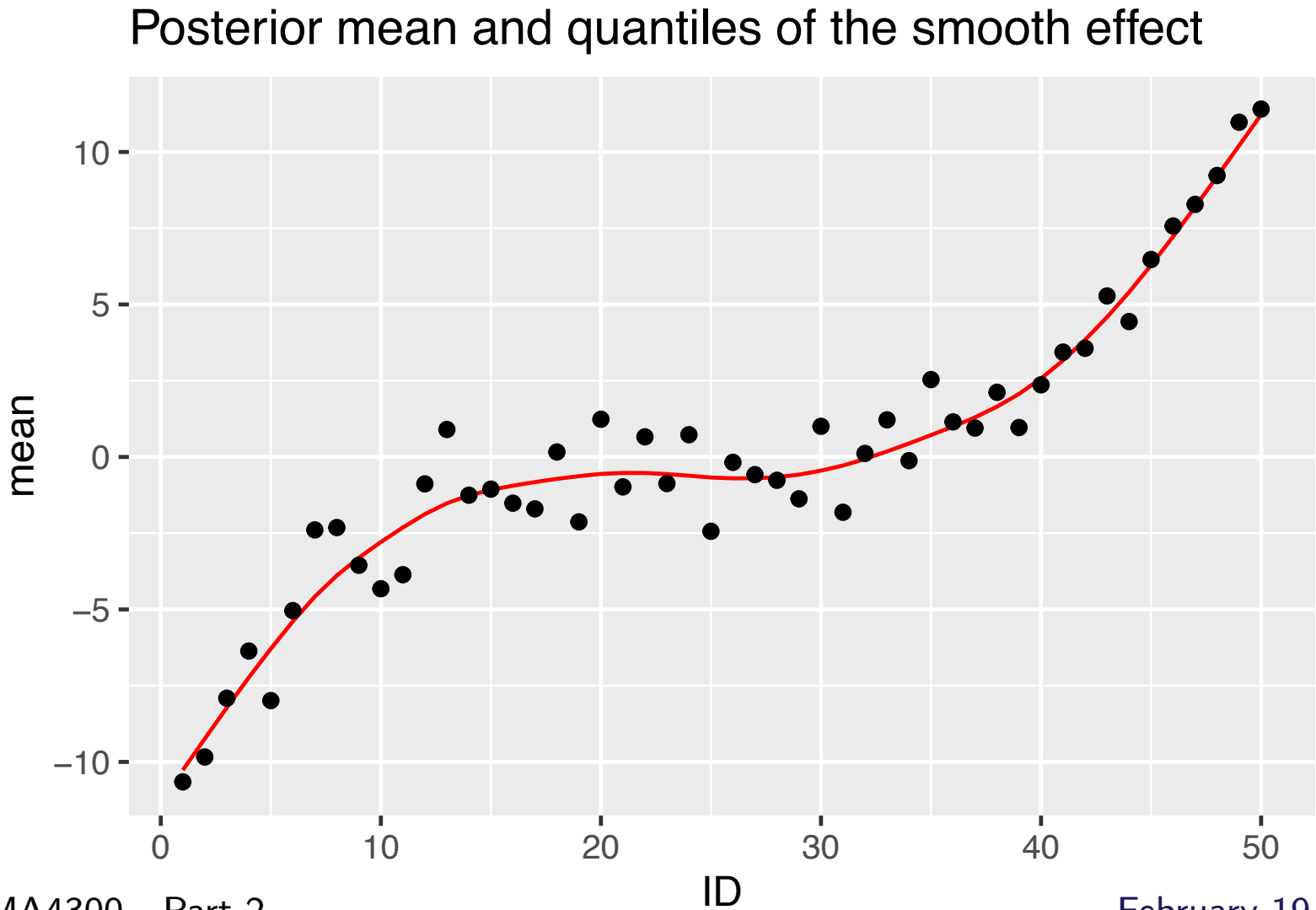
Posterior marginals for latent field

Sum to get the posterior marginal for $x_i|y$



Fitted Spline

The posterior marginals are used to calculate summary statistics, like means, variances and credible intervals:



Extending the method

This is the basic idea behind INLA. It is quite simple. However, we need to extend this basic idea so we can deal with

- More than one hyperparameter
- Non-Gaussian observations

Non-Gaussian Observations: Approximating $\pi(\boldsymbol{x}|\theta y)$

Let $\boldsymbol{x}$ denote a GMRF with precision matrix $\boldsymbol{Q}$ and mean $\boldsymbol{\mu}$.

Approximate

$$\pi(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y}) \propto \exp\left(-\frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \sum_{i=1}^{n} \log \pi(y_i|x_i)\right)$$

by using a second-order Taylor expansion of $\log \pi(y_i|x_i)$ around $\boldsymbol{\mu}_0$, say.

Recall

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2 = a + bx - \frac{1}{2}cx^2$$

with $b = f'(x_0) - f''(x_0)x_0$ and $c = -f''(x_0)$.

The GMRF approximation (II)

Thus,

$$\tilde{\pi}(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y}) \propto \exp\left(-\frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \sum_{i=1}^{n}(a_i + b_i x_i - 0.5 c_i x_i^2)\right)$$

$$\propto \exp\left(-\frac{1}{2}\boldsymbol{x}^T(\boldsymbol{Q} + \text{diag}(\boldsymbol{c}))\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x}\right)$$

to get a Gaussian approximation with precision matrix $\boldsymbol{Q} + \text{diag}(\boldsymbol{c})$ and mean given by the solution of $(\boldsymbol{Q} + \text{diag}(\boldsymbol{c}))\boldsymbol{\mu} = \boldsymbol{b}$. The canonical parameterization is

$$\mathcal{N}_C(\text{b}, \boldsymbol{Q} + \text{diag}(\boldsymbol{c}))$$

which corresponds to

$$\mathcal{N}((\boldsymbol{Q} + \text{diag}(\boldsymbol{c}))^{-1}\text{b}, (\boldsymbol{Q} + \text{diag}(\boldsymbol{c}))^{-1}).$$

The GMFR approximation - One dimensional example

Assume

$$y|\lambda \sim \text{Poisson}(\lambda) \text{ Likelihood}$$
$$\lambda = \exp(x) \text{ Likelihood}$$
$$x \sim \mathcal{N}(0, 1) \text{ Latent Model}$$

we have that

$$\pi(x|y) \propto \pi(y|x)\pi(x) \propto \exp\{-\frac{1}{2}x^2 + \underbrace{xy - exp(x)}_{\text{non-gaussian part}}\}$$

(Show R-code `Taylor_expansion.R`)

Non-Gaussian Observations

In many cases $\pi(\boldsymbol{x}|\boldsymbol{y}, \theta)$ is very close to a Gaussian distribution, and can be replaced with a Laplace approximation:

- This means that all the really hard, high-dimensional integrals with respect to the latent field are easy, and only the integrals with respect to the hyperparameters remain
- If the number of hyperparameters is low, these integrals can be done efficiently numerically

Limitations

- The dimension of the latent field x can be large ($10^2 - 10^6$)
- The dimension of the hyperparameters θ must be small (≤ 9)

In other words, each random effect can be big, but there cannot be too many random effects unless they share parameters.

Gaussian Markov Random Fields

A GMRF $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ is a random vector following a multivariate Gaussian distribution

$$\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{Q}^{-1}) \text{ where } \boldsymbol{Q}^{-1} = \boldsymbol{\Sigma}$$

and that is endowed with some Markov properties like

$$x_j \perp x_i | \boldsymbol{x}_{-ij}$$

where $\boldsymbol{x}_{-ij}$ indicates "all elements of $\boldsymbol{x}$ other than i and j"

The easiest example is a AR(1) model

Gaussian Markov Random Fields

If $\boldsymbol{\Sigma}$ is the covariance matrix of a Gaussian vector and $\boldsymbol{Q} = \boldsymbol{\Sigma}^{-1}$ is the precision matrix, we have that

$$x_i \perp x_j \iff \Sigma_{ij} = 0$$

and

$$x_i \perp x_j | \boldsymbol{x}_{-ij} \iff Q_{ij} = 0$$

Gaussian Markov Random Fields

If $\boldsymbol{\Sigma}$ is the covariance matrix of a Gaussian vector and $\boldsymbol{Q} = \boldsymbol{\Sigma}^{-1}$ is the precision matrix, we have that

$$x_i \perp x_j \iff \Sigma_{ij} = 0$$

and

$$x_i \perp x_j | \boldsymbol{x}_{-ij} \iff Q_{ij} = 0$$

GMRF have sparse precision matrices....this means it is "easy" to compute determinant and invert $\boldsymbol{Q}$

Bayesian computing

The posterior distribution is given by

$$\mathrm{p}(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y}) \propto \mathrm{p}(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})\, \mathrm{p}(\boldsymbol{x}|\boldsymbol{\theta})\, \mathrm{p}(\boldsymbol{\theta})$$

We are interested in the posterior marginal quantities like $\mathrm{p}(x_i|\boldsymbol{y})$ and $\mathrm{p}(\theta_j|\boldsymbol{y})$. This requires the evaluation of integrals of the form:

$$\mathrm{p}(x_i|\boldsymbol{y}) \propto \int_{\boldsymbol{x}_{-i}} \int_{\boldsymbol{\theta}} \mathrm{p}(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y}) d\boldsymbol{\theta} d\boldsymbol{x}_{-i}.$$

The computation of massively high dimensional integrals is at the core of Bayesian computing, especially for hierarchical models.

However, such high dimensional integrals might lead to long computation times for MCMC.

Approximate inference

Except for the (trivial) cases when everything can be computed exactly (maybe up to very small integration error), we can never do exact inference in this context.

- Integrated nested Laplace approximations (INLA) are a promising alternative to inference via MCMC in latent Gaussian models

  (Rue et al, 2009, JRSS-B).

- The methodology is particularly attractive if the latent Gaussian model is a Gaussian Markov random field (GMRF)

  (Rue and Held, 2005).

Gaussian Markov Random Field (GMRF)

Theorem

Let $\boldsymbol{x}$ be normal distributed with mean $\boldsymbol{\mu}$ and symmetric positive-definite (SPD) precision matrix $\boldsymbol{Q}$, i.e. $\boldsymbol{Q} > 0$, Then for $i \neq j$,

$$x_i \perp x_j | \boldsymbol{x}_{-ij} \iff Q_{ij} = 0$$

Gaussian Markov Random Field (GMRF) cont.

Definition

A random vector $\boldsymbol{x} = (x_1, \ldots, x_n)^\top \in \mathbb{R}^n$ is called a GMRF with respect to a labelled graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with mean $\boldsymbol{\mu}$ and precision matrix $\boldsymbol{Q} > 0$, iff its density has the form

$$\pi(\boldsymbol{x}) = (2\pi)^{-n/2} |\boldsymbol{Q}|^{1/2} \exp\left(-\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{Q} (\boldsymbol{x} - \boldsymbol{\mu}) \right)$$

and

$$Q_{ij} \neq 0 \iff \{i, j\} \in \mathcal{E}, \forall i \neq j$$

If $\boldsymbol{Q}$ is completely dense then $\mathcal{G}$ is fully connected. Thus, any normal distribution with a SPD covariance matrix is also a GMRF and vice versa.