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

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

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

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

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

ï¿œ

Review: Common proposal types

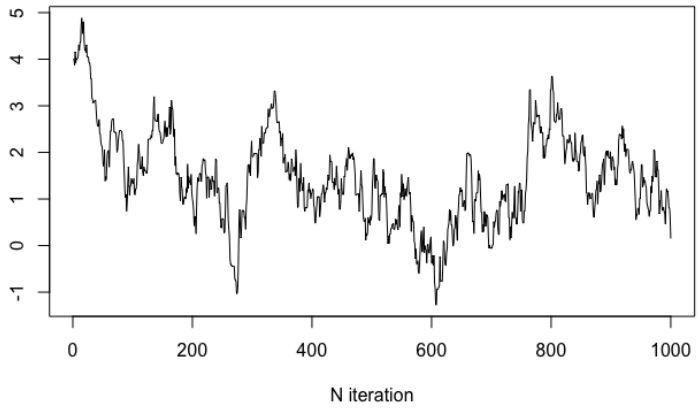
- Independent proposals: $Q(y|x) = q(y)$
  - usually not a good alternative (alone)
- Random walk proposals: $Q(y|x) = N(y|x, \sigma^2 I)$
  - is used a lot
  - includes a tuning parameter: σ
- Gibbs updates: $Q(y^j|x^j, \mathbf{x}^{-j}) = f(x^j|\mathbf{x}^{-j})$
  - is used a lot
  - the proposal density is the full conditional
  - no tuning paramter
  - acceptance rate 1
  - can be conbined with MH update

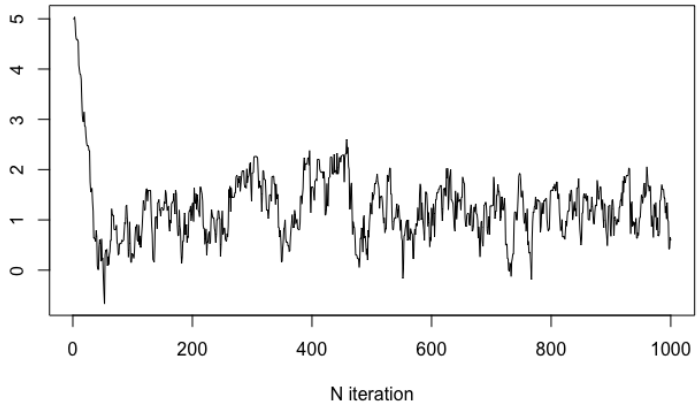
Review: Convergence diagnostic

Has the MC converged?

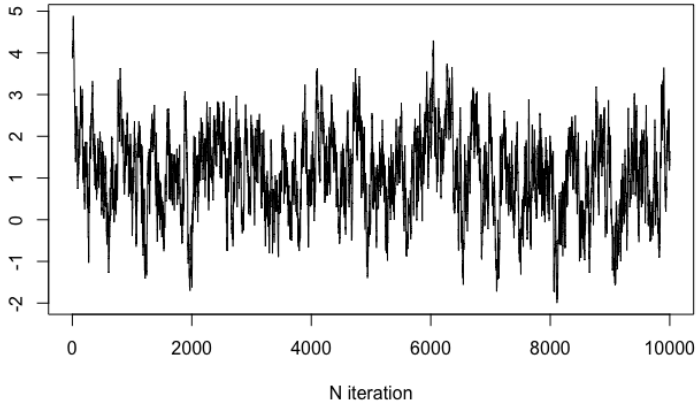
- Formal convergence diagnostics exists
  - some based on a single Markov chain run
  - some based on several Markov chain runs
- Standard way to assess convergence is to look at the traceplot
- If some properties of the target distribution is known: use it to check convergence!
- All convergence diagnostics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

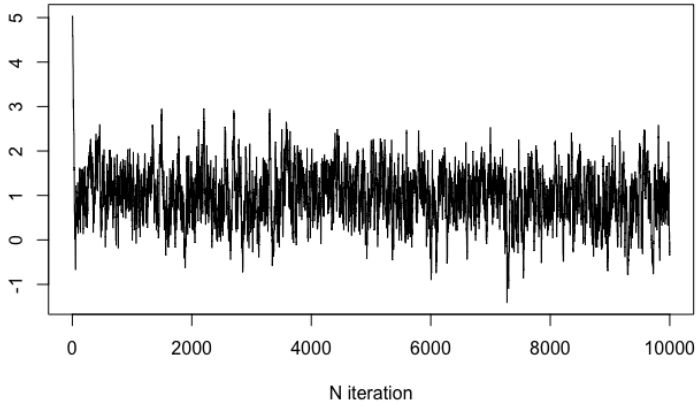
Has bivariate this MC converged?



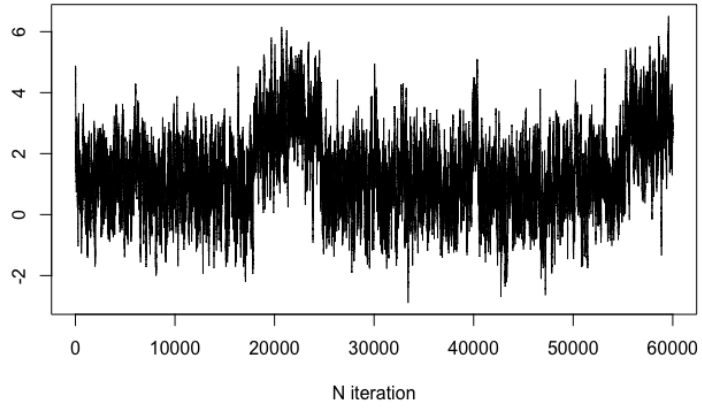


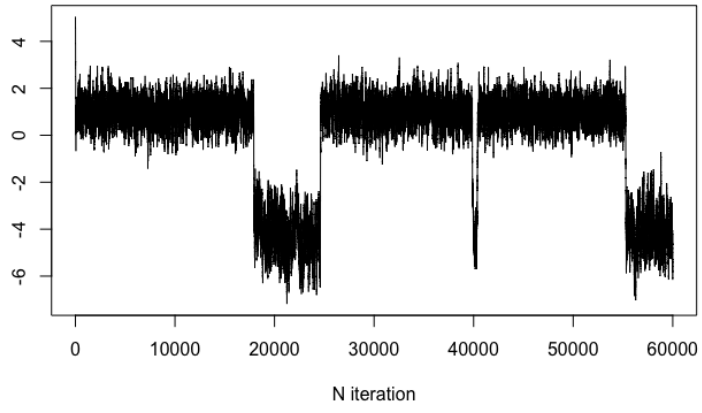
Has bivariate this MC converged?



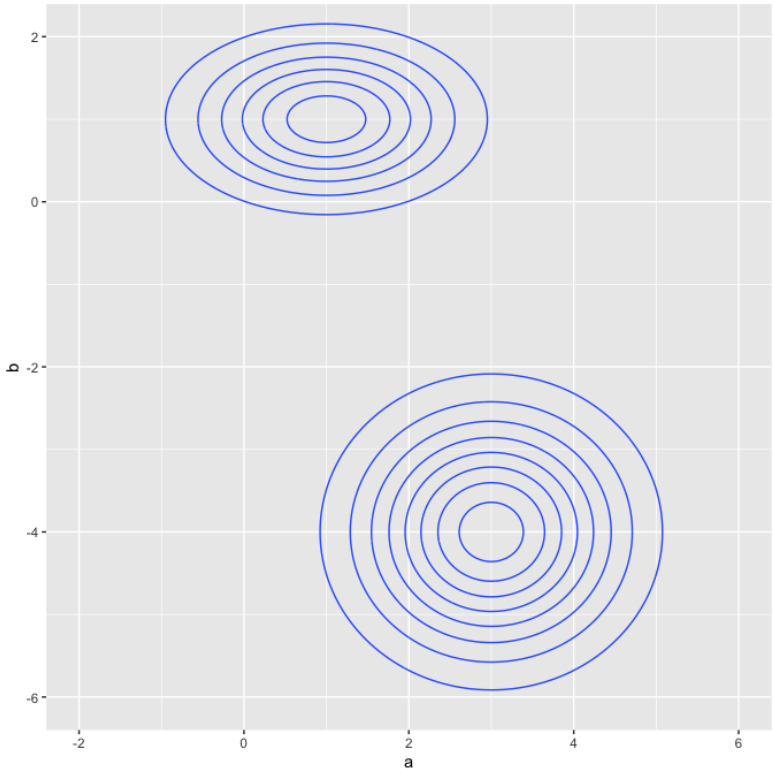


Has bivariate this MC converged?





Has bivariate this MC converged?



This is how the distribution looks like.

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

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

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

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

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

Autocorrelation

To examine dependencies of successive MCMC samples, the autocorrelation function can be used. Let $x_1, \ldots, x_N$, where N denotes the number of samples, denote our MCMC chain.

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

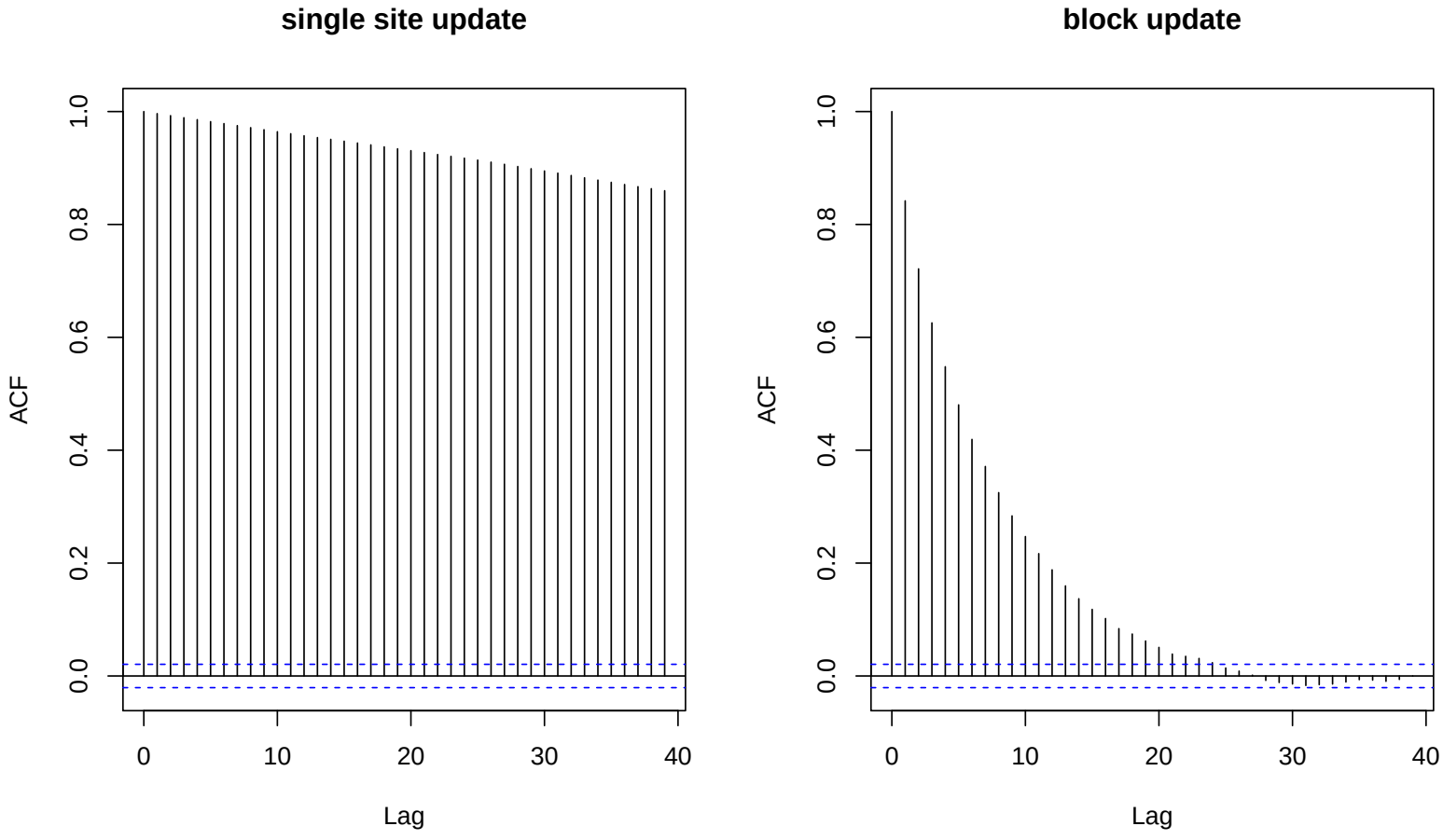
$$\rho(k) \approx \frac{\sum_{i=1}^{N-k}(x_i - \bar{x})(x_{i+k} - \bar{x})}{\sum_{i=1}^{N}(x_i - \bar{x})^2},$$

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] 8000

> ## single site
> effectiveSize(as.mcmc(res1))

       N    theta
18.46377 13.65231

> ## block update
> effectiveSize(as.mcmc(res2))

       N    theta
564.3797 925.7764

>
```

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

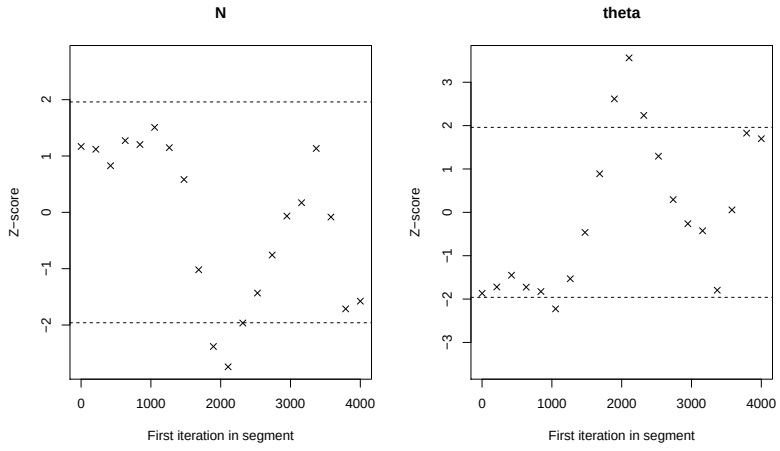
- the first x%, and
- the last y% of the iterates

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

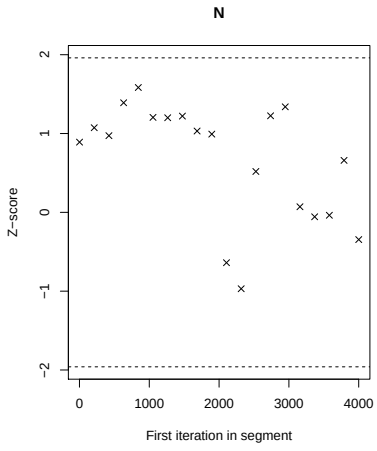
If the chain is stationary both values should be equal and Geweke's test statistic (z-score) follows an asymptotical standard normal distribution.

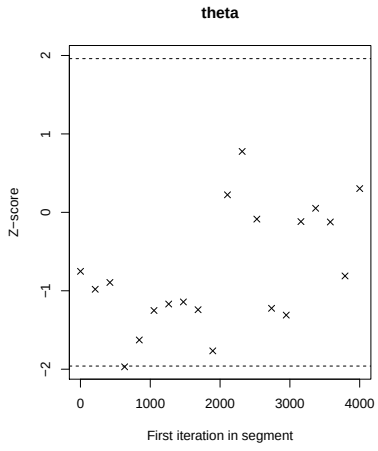
Example: Korsbetningen - Geweke plot

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

- R-package `coda`. (Plummer et al., 2006)

Summary

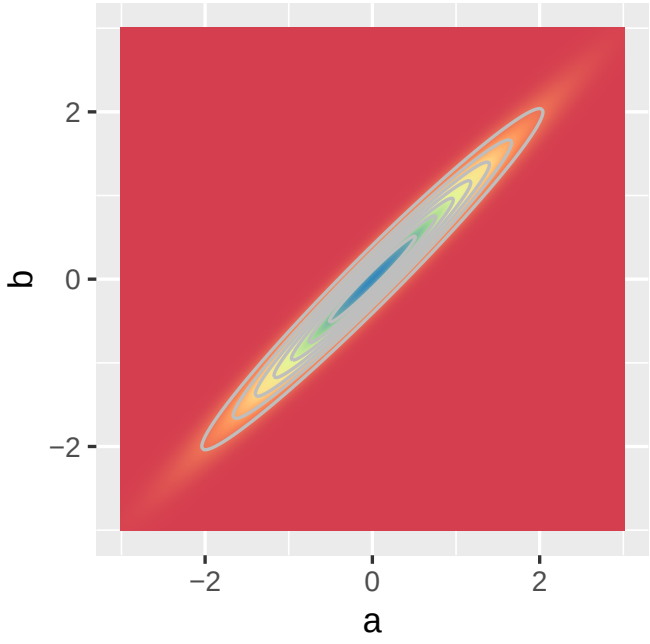
- Diagnostics cannot guarantee that chain has converged
- Can indicate that it has not converged

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
- ...

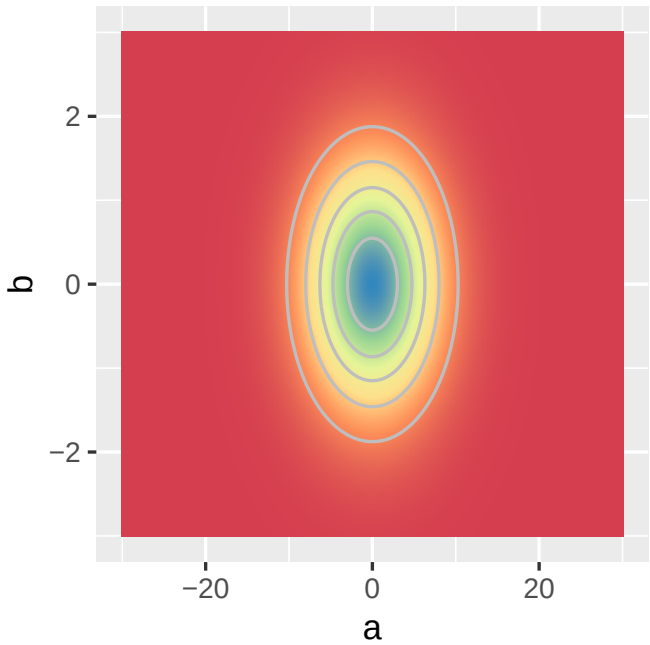
Typical MCMC problems

- Note: If you knows the solution, it is easy to solve a problem!
- Properties of $f(x)$ that may make MCMC difficult
  - strong dependency between variables



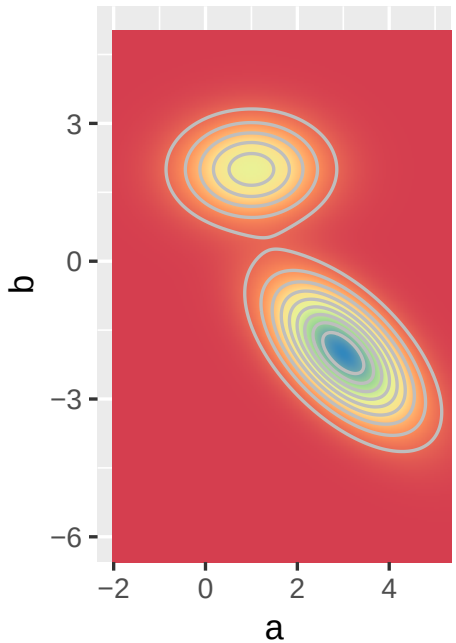
Typical MCMC problems

- Note: If you knows the solution, it is easy to solve a problem!
- Properties of $f(x)$ that may make MCMC difficult
  - strong dependency between variables
  - different scales on different variables



Typical MCMC problems

- Note: If you knows the solution, it is easy to solve a problem!
- Properties of $f(x)$ that may make MCMC difficult
  - strong dependency between variables
  - different scales on different variables
  - several modes



Typical MCMC problems

- Note: If you knows the solution, it is easy to solve a problem!
- 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

- Since the advent of simulation-based techniques (notably MCMC), Bayesian computation has enjoyed incredible development
- This has certainly been helped by dedicated software (eg BUGS and then WinBUGS, OpenBUGS, JAGS, Stan)
- MCMC methods are very general and can effectively be applied to "any" model

MCMC

- Since the advent of simulation-based techniques (notably MCMC), Bayesian computation has enjoyed incredible development
- This has certainly been helped by dedicated software (eg BUGS and then WinBUGS, OpenBUGS, JAGS, Stan)
- MCMC methods are very general and can effectively be applied to "any" model
- However:
  - Even if in theory, MCMC can provide (nearly) exact inference, given perfect convergence and MC error $\to 0$, in practice, this has to be balanced with model complexity and running time
  - This is particularly an issue for problems characterised by large data or very complex structure (eg hierarchical models)
  - This is also a problem if one wants to, for example, test the sensitivity of the model to the prior choice

What it 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 Fieds
  - Sparse matrix computations
- Laplace Approximation
  -

Hierarchical Bayesian models

Hierarchical models are an extremely useful tool in Bayesian model building.

Three parts:

- Observation model $\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}$: Encodes information about observed data.
- The latent model $\boldsymbol{x}|\boldsymbol{\theta}$: The unobserved process.
- Hyperpriors for $\boldsymbol{\theta}$: Models for all of the parameters in the observation and latent processes.

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 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 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 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 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

Some more example of LGM

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

Unified framework

Observations: $\boldsymbol{y}$

Latent field: $\boldsymbol{x}$

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

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

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

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)

Main interest

The posterior distribution is given by

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

Main interest

The posterior distribution is given by

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

We are mainly interested in the posterior marginals

$$\pi(x_i|\boldsymbol{y}) = \int_{\boldsymbol{\theta}} \underbrace{\int_{\boldsymbol{x}_{-i}} \pi(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y}) d\boldsymbol{x}_{-i}}_{\pi(x_i, \boldsymbol{\theta}|\boldsymbol{y})} d\boldsymbol{\theta} = \int_{\boldsymbol{\theta}} \pi(x_i, \boldsymbol{\theta}|\boldsymbol{y}) d\boldsymbol{\theta} = \int_{\boldsymbol{\theta}} \pi(x_i|\boldsymbol{\theta}, \boldsymbol{y})\pi(\boldsymbol{\theta}|\boldsymbol{y}) d\boldsymbol{\theta}$$

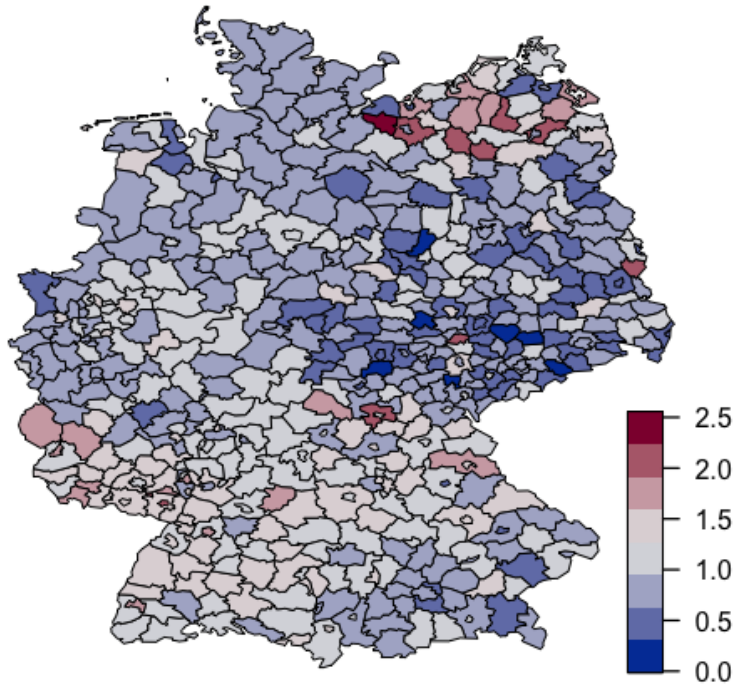
$$\pi(\theta_j|\boldsymbol{y}) = \int_{\boldsymbol{\theta}_{-j}} \underbrace{\int_{\boldsymbol{x}} \pi(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y}) d\boldsymbol{x}}_{\pi(\boldsymbol{\theta}|\boldsymbol{y})} d\boldsymbol{\theta}_{-j} = \int_{\boldsymbol{\theta}_{-j}} \pi(\boldsymbol{\theta}|\boldsymbol{y}) d\boldsymbol{\theta}_{-j}$$

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.

Information available:

- 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)



Level 1: The data

We have to decide on the likelihood of our data $\boldsymbol{y}$

- The responses are counts
- We choose a Poisson model

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

- η_i is a linear function of the latent components

Level 2: The Latent Model

The latent field $\boldsymbol{x}$ consists of two parts:

- One fixed effect: the intercept μ
- Three random effects:
  - The spatially structured effect f_s.
  - The unstructured effect u which accounts for non-observed variability
  - The unknown effect $f(c_i)$ of the exposure covariate which assumes value c_i for district i.

These are combined for each location to give a linear predictor

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

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

Level 3: The hyperparameters

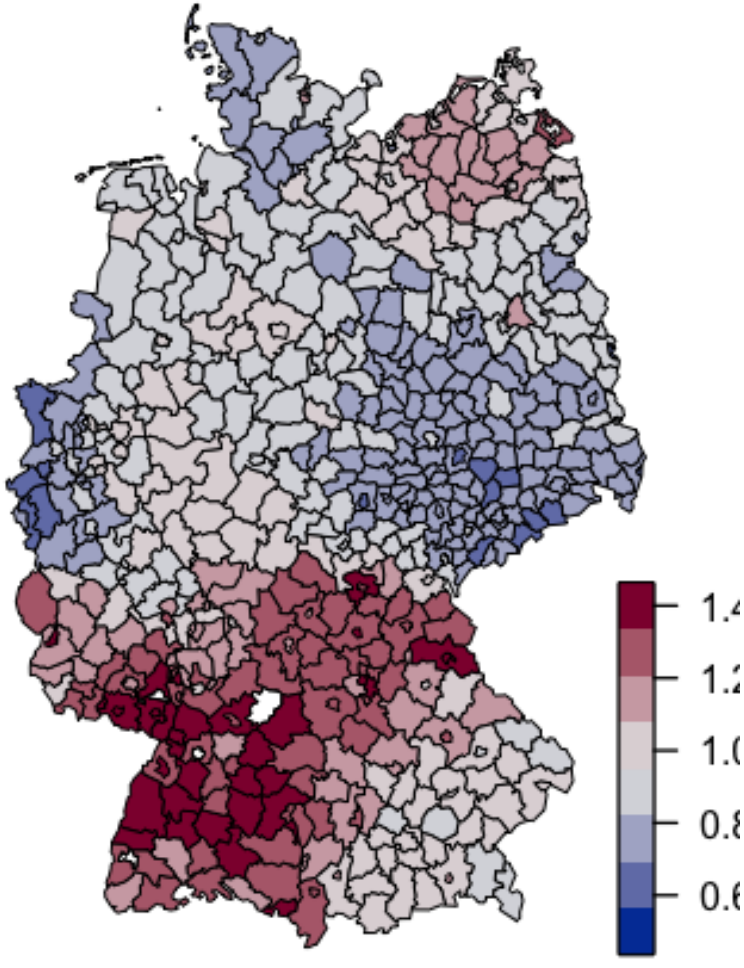
The structured and unstructured spatial effect as well as the smooth covariate effect will be each controlled by one parameter

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

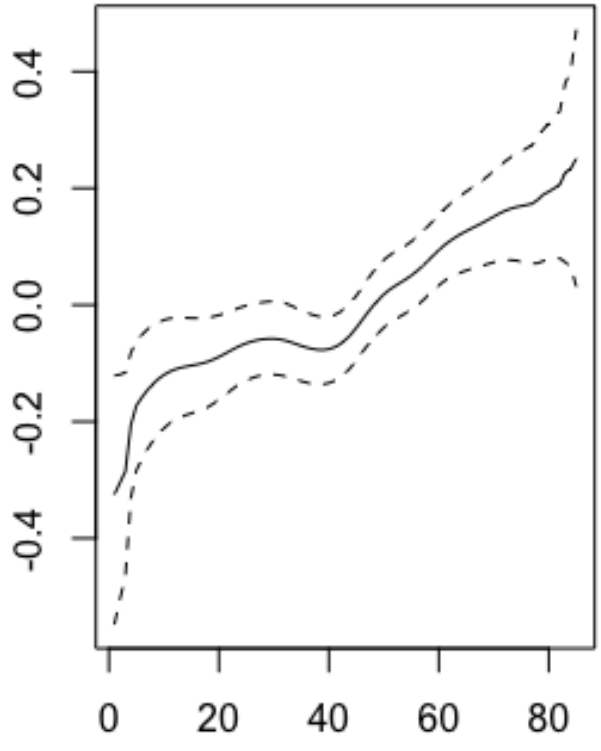
The hyperparameters are $\boldsymbol{\theta} = (\tau_c, \tau_f, \tau_\eta)$, and must be given a prior $\pi(\tau_c, \tau_f, \tau_\eta)$

What are we interested in?

Structured spatial effect $\exp(f_s(s_i))$



Covariate effect $\exp(f(c_i))$



INLA computing scheme

We want to approximate:

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

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

How INLA does it:

INLA computing scheme

We want to approximate:

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

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

How INLA does it:

- Approximate $\pi(\theta_j|\boldsymbol{y})$ as $\widetilde{\pi}(\theta_j|\boldsymbol{y})$

INLA computing scheme

We want to approximate:

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

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

How INLA does it:

- Approximate $\pi(\theta_j|\boldsymbol{y})$ as $\widetilde{\pi}(\theta_j|\boldsymbol{y})$
- Approximate $\pi(x_i|\boldsymbol{\theta},\boldsymbol{y})$ as $\widetilde{\pi}(x_i|\boldsymbol{\theta},\boldsymbol{y})$

INLA computing scheme

We want to approximate:

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

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

How INLA does it:

- Approximate $\pi(\theta_j|\boldsymbol{y})$ as $\widetilde{\pi}(\theta_j|\boldsymbol{y})$
- Approximate $\pi(x_i|\boldsymbol{\theta}, \boldsymbol{y})$ as $\widetilde{\pi}(x_i|\boldsymbol{\theta}, \boldsymbol{y})$
- Use numerical integration (a finite sum) to compute
  - $$\widetilde{\pi}(x_i|\boldsymbol{y}) = \sum_k \widetilde{\pi}(x_i|\boldsymbol{\theta}_k, \boldsymbol{y})\ \widetilde{\pi}(\boldsymbol{\theta}_k|\boldsymbol{y})\ \Delta_k.$$
  - $$\widetilde{\pi}(\boldsymbol{\theta}_j|\boldsymbol{y})$$

Gaussian Markov Random Fields

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

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

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

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

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

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

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

The GMRF approximation

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

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

which corresponds to

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

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`)

The GMRF approximation