Markov Chain Monte Carlo (MCMC) and Model Evaluation

August 15, 2017

UNIVERSITÉ DE SHERBROOKE

Linking Frequentist and Bayesian Statistics

How can we estimate model parameters and what does it imply?

Frequentist

Want to find the best model parameter(s) for the data at hand

Likelihood $\quad\quad P(\text{Data}|\text{Model})$

They are interested in **maximizing** the Likelihood

They need **data**

This can be done using

- Simulated annealing
- The Nelder-Mead Simplex
- Minimizing the sums of squares
- . . .

Linking Frequentist and Bayesian Statistics

How can we estimate model parameters and what does it imply?

Bayesian

Want to find how good the model parameter(s) are given some data

Posterior $$P(\text{Model}|\text{Data})$$

They are intered in the posterior distribution

They need **data** and **prior** information

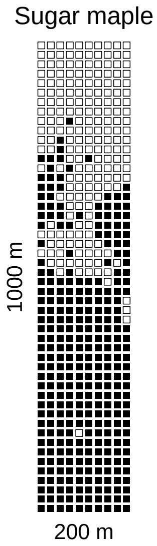
Recall that

$$\underbrace{P(\text{Model}|\text{Data})}_{\text{Posterior}} \propto \underbrace{P(\text{Data}|\text{Model})}_{\text{Likelihood}}\underbrace{P(\text{Model})}_{\text{Prior}}$$

Some ecological context

How important is elevation in defining sugar maple distribution on mont Sutton?

Mont Sutton



Definition of prior probability

The **prior probability** informes us about the probability of the model being true *before* the current data is considered.

Types of priors

“Uninformative”

These priors are meant to bring very little information about the model

Informative

These priors bring information about the model that is available

Conjugate

These priors have the same functional form (mathematically speaking) as the likelihood

"Uninformative priors"

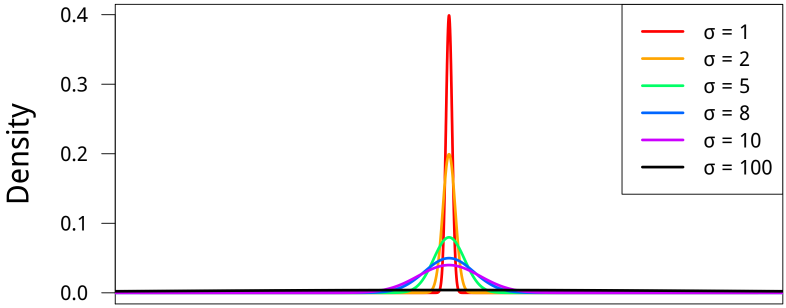
Example: If we have no idea of how elevation influence sugar maple

Gaussian distribution

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

with $\mu = 0$

σ = Large say 100



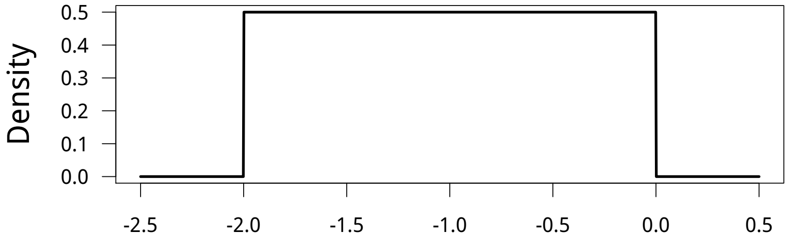
Informative priors

Example: If we know that

- There are less sugar maples the higher we go
- The influence of elevation on sugar maple cannot be more than two folds

Uniform distribution

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } x \in [a, b] \\ 0 & \text{otherwise} \end{cases} \qquad \text{with} \qquad \begin{matrix} a > -2 \\ b < 0 \end{matrix}$$



Conjugate priors

These types of priors are convenient to use because

They are computationally faster to use

They can be interepreted as additional data

Why are they useful?

There is no need to write the likelihood down when using them. All that needs to be done is to sample them to obtain a parameter estimation.

What does it mean to be of the same *functional form*?

It means that both distribution have th same mathematical structure.

Binomial distribution	**Beta distribution**
$\theta^{a}(1-\theta)^{b}$	$\theta^{\alpha-1}(1-\theta)^{\beta-1}$

https://en.wikipedia.org/wiki/Conjugate_prior

Rejection Sampling

It is designed so that we can sample from any distribution

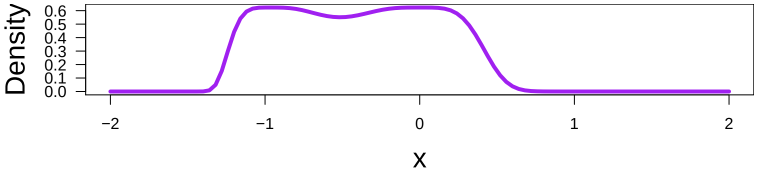
Step 1 Define an easy to sample candidate distribution (c(x))

Uniform distribution

$$\begin{cases} \frac{1}{b-a} & \text{for } x \in [a, b] \\ 0 & \text{otherwise} \end{cases}$$

Step 2 Define the target (hard) distribution (t(x))

$$e^{-(x^2+\sin(2x))^4-0.473}$$



Step 3 Reject candidates with a probability proportional to the difference between the two distributions

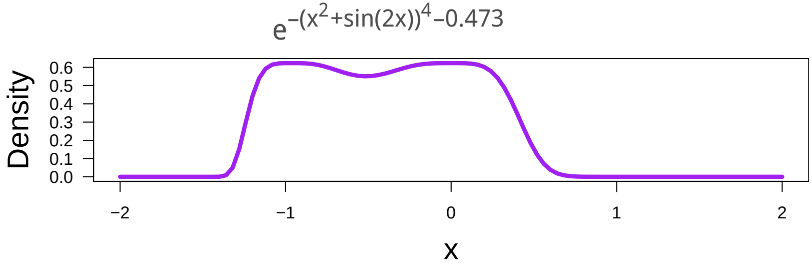
Define M such that $Mc(x) \geq t(x)$ for all x

Algorithm in pseudocode

```
REPEAT
   sample y from cand(x)
   Calculate acceptance probability p = target(y) / (M*cand(y))
   Draw a value U a random uniform distribution (0,1)
   IF U < p
      accept y
   ELSE
      reject y
UNTIL y is accepted
```

Write your own rejection sampling algorithm

Problem Sample from the following target distribution



Hint

Use `dunif` and `runif` as candidate distribution with a range that covers a little more than the span of the target distribution

Why are Markov Chain Monte Carlo (MCMC) useful?

Typical reasons to favour MCMC

- It is flexible
- It can be applied to complex models such as models with multiple levels of hierarchy
- It can be implemented from scratch (we will do it today !)

In practice, there is no reason to write an MCMC when the likelihood *can* be solved analytically

Other alternatives to MCMC

- Hamiltonian (hybrid) Monte Carlo
- Laplace approximation
- Integrated nested Laplace approximation
- . . .

Properties Markov Chain Monte Carlo (MCMC)

Similarity with simulated annealing

New parameter values are chosen sequentially but randomly

There are many ways to choose and accept new parameter values

Difference with simulated annealing

The main goal of MCMC is to sample the posterior distribution **not** to find “the best” value

No “temperature” is defined

It does not impossible to get stuck in a loop while iterating

Assumptions of MCMC

All potential parameter combinations can be reached from all other parameter combination

After enough iterations the chain will converges to a stationary distribution

The Metropolis-Hasting Algorithm

Theory

Usefulness

- It is well designed to approach univariate problems
- It is useful to sample Bayesian posterior distribution

Properties

- Each iteration generates a sample from the target distribution
- Samples are dependent on one another (they are autocorrelated)... So, effective sample size is smaller than the chain length

Defining the important parts

Number of steps (N) to run the MCMC

It has to be large.

Starting value (θ)

It should roughly describe the distribution to be estimated

Target distribution ($f(\theta)$)

It is the distribution of value we aim at estimating

Jumping distribution ($j(\theta_{cand}|\theta_{t-1})$)

Many choices are possible but it must allow for positive recurrence

The normal distribution ($\mu = \theta_{cand}$, $\sigma^2 = 1$) is a good starting point

$$\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Defining the important parts (conitnued)

Acceptance probability

$$r = \frac{f(\theta_{cand})j(\theta_{t-1}|\theta_{cand})}{f(\theta_{t-1})j(\theta_{cand}|\theta_{t-1})}$$

Note that for any *symmetric* distribution, such as the uniform or normal distribution, the acceptance probability becomes

$$r = \frac{f(\theta_{cand})}{f(\theta_{t-1})}$$

This is known as the **Metropolis algorithm**

Algorithm in pseudocode

```
for t in 1 to N
   sample theta_c from j(theta_c|theta[t-1])
   set r_c = r(theta_c, theta[t-1])
   sample U from uniform(0, 1)
   IF U < r_c
      theta[t] = theta_c
   ELSE
      theta[t] = theta[t-1]
```

How many step is enough. . .

A rough procedure

Step 1 Perform a pilot run for a reduced number of steps (10 to 100) and measure the time it takes

Step 2 Decide on a number of steps to run the algorithm to obtain a result in a reasonable amount of time

Step 3 Run the algorithm again !

Step 4 Study the chain visually

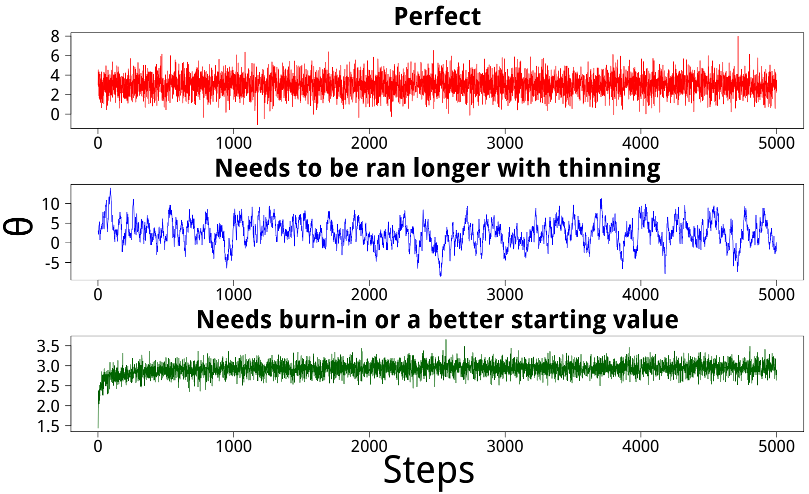
A more statistical way - The Raftery-Lewis diagnostic

It relies on a pilot run to estimate the number of steps to be carried out

It is implemented in the `raftery.diag` function of the `coda` R package

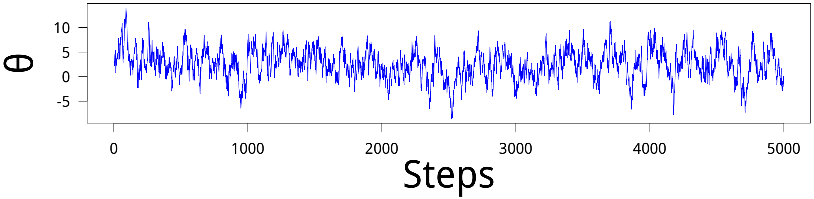
Good Practices

Trace plot

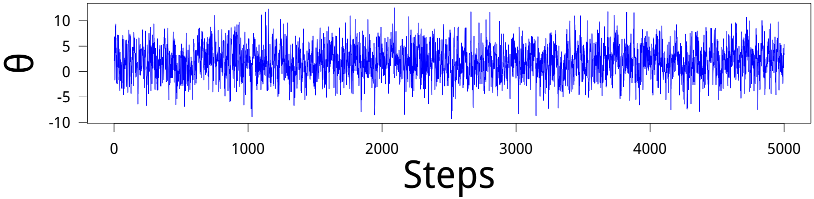


Good Practices

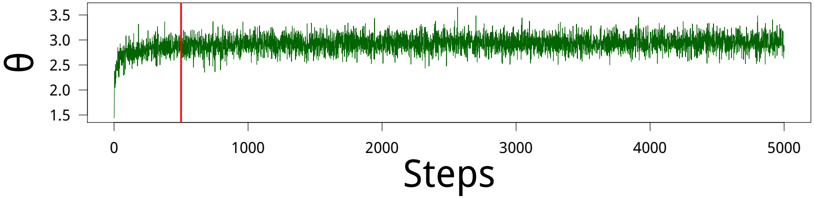
Thinning is essentially **subsampling**



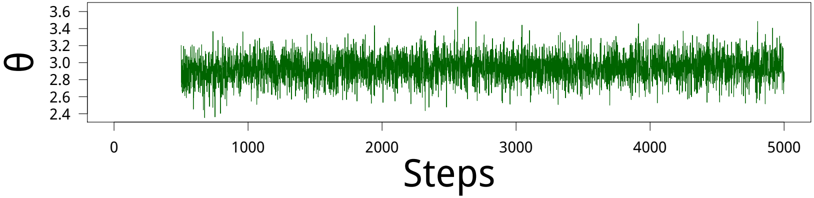
If we ran the same MCMC as above but instead for 50000 steps and we save θ at every 10 steps, we obtain



Burn-in is throwing away some iterations at the beginning of the MCMC run



After burn-in, we obtain

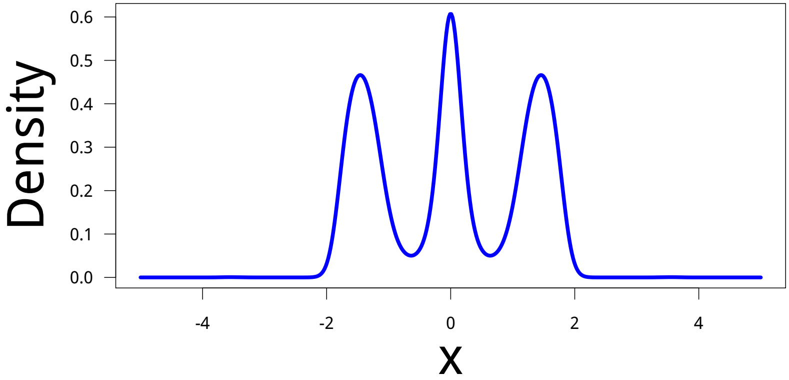


Exercice

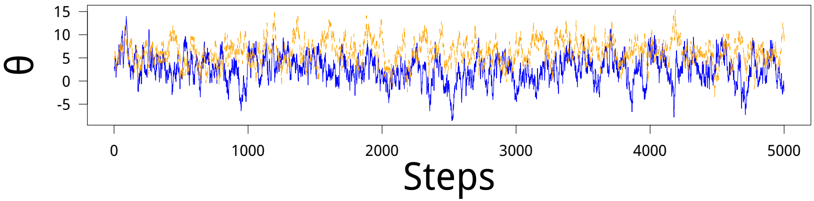
Write your own Metropolis-Hasting Algorithm

Problem Sample from the following target distribution

$$\frac{e^{-(x+\sin(3x))^2+0.195}}{2}$$

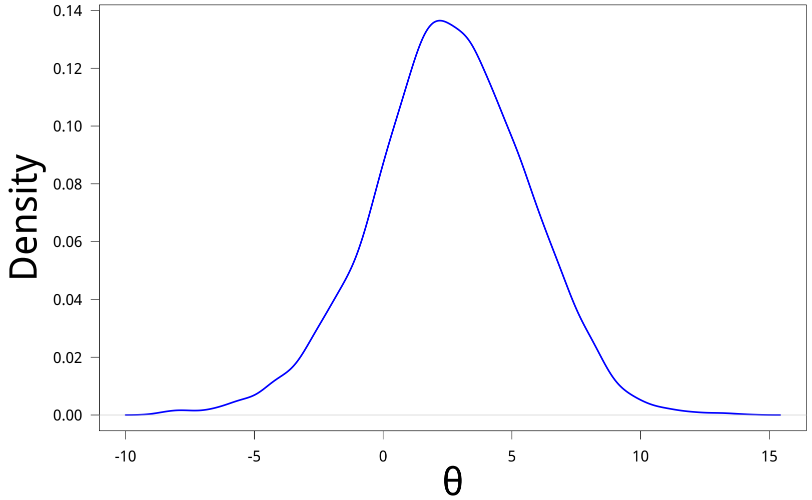


Rerun the estimation procedure multiple times with different starting values



Convergence

Density plot



Convergence

Geweke convergence diagnostics

It compares two sections of the same chain

Technically, it is a two sample t test of mean with unequal variance

$$Z = \frac{\bar{\theta}_A - \bar{\theta}_B}{\sqrt{\frac{S_A}{n_A} + \frac{S_B}{n_B}}}$$

where

$\bar{\theta}_A$ and $\bar{\theta}_B$ are the means of the chain section θ_A and θ_B,
n_A and n_B are the number of steps of the chain section θ_A and θ_B,

$$S_A = \frac{\sigma_A^2}{(1 - \sum \alpha_A)^2} \qquad S_B = \frac{\sigma_B^2}{(1 - \sum \alpha_B)^2}$$

σ_A and σ_B are the variance of the chains sections θ_A and θ_B
α_A and α_B are autoregressive parameters of the chain sections θ_A and θ_B

It is implemented in the `geweke.diag` function of the `coda` R package

Convergence

Gelman-Rubin convergence diagnostics

It compares multiple chains

it is a corrected ratio of the pooled variance of all chains with the within variance of each chain

$$R = \sqrt{\frac{V}{W}}$$

Chains pooled variance

$$V = \frac{N-1}{N} W + \frac{1}{N} B$$

where

$$B = \frac{N}{M-1} \sum_{m=1}^{M} (\bar{\theta}_m - \bar{\theta})^2,$$

N is the length of each chain (it is assumed to be the same)
M is the number of chains
$\bar{\theta}_m$ is the mean chain m,
$\bar{\theta}$ is the mean of all chains.

Within chain variance

$$W = \frac{1}{M} \sum_{m=1}^{M} \sigma^2$$

It is implemented in the `gelman.diag` function of the `coda` R package.

Adaptive Metropolis-Hasting Algorithm

We adapt the standard deviation of the normal distribution during burn-in

$$\frac{1}{\sqrt{2\pi\sigma^2 A}} e^{-\frac{(x-\mu)^2}{2\sigma^2 A}}$$

where A is a tuning parameter

Adaptive Metropolis-Hasting Algorithm

Adaptative Algorithm in pseudocode

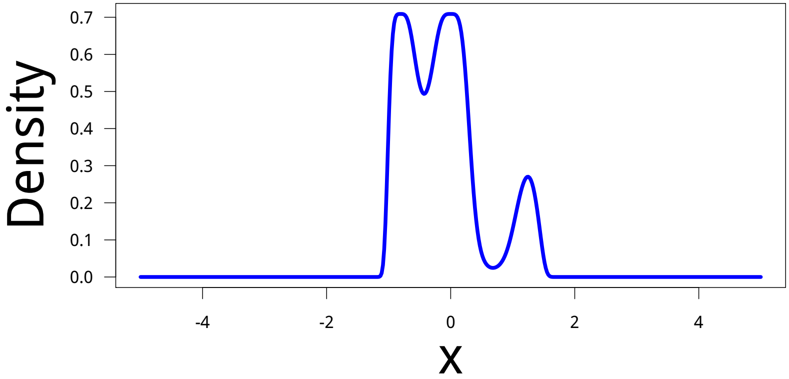
```
set A = 1
for t in 1 to N
   sample theta_c from j(theta_c|theta[t-1], A)
   set r_cand = r(theta_c, theta[t-1])
   sample U from uniform(0, 1)
   IF U < r_cand
      theta[t] = theta_c
      IF burnin
         A = A * 1.01
   ELSE
      theta[t] = theta[t-1]
      IF burnin
         A = A / 1.01
```

Exercice

Build your own adaptive Metropolis-Hasting algorithm

Problem Sample from the following target distribution

$$e^{-\left(x^2+\sin(3x)\right)^4-0.344}$$



Single component adaptive Metropolis-Hasting algorithm

Adaptative Algorithm in pseudocode

```
set A = repeat(1,nparam)
for t in 1 to N
   for i in 1 to nparam
      sample theta_c[i] from j(theta_c[i]|theta[i,t-1], A[i])
      set r_c = r(theta_c, theta[t-1])
      sample U from uniform(0, 1)
      IF U < r_c
         theta[t] = theta_c
         IF burning
            A[i] = A[i] * 1.01
      ELSE
         theta[t] = theta[t-1]
         IF burning
            A[i] = A[i] / 1.01
```

The SCAM is designed for models with multiple parameters

theta_cand is a vector, not a single value

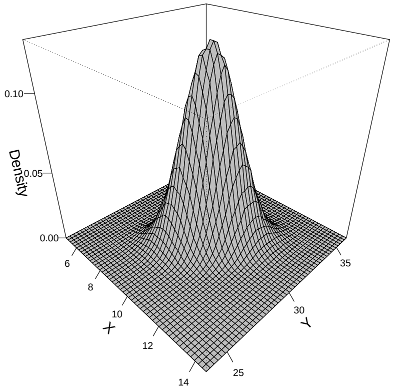
theta is a matrix, not a vector

Exercice

Build your own single component adaptive Metropolis-Hasting algorithm

Problem Sample from the following bivariate distribution

$$\frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}{\sqrt{|2\pi\boldsymbol{\Sigma}|}},$$



with

Mean $\boldsymbol{\mu} = [10, 30]$

Variance $\boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 2 \end{bmatrix}$

The probability density function of the bivariate normal distribution can be obtained through the `dmvnorm` function of the `mvtnorm` R package.

Gibbs Sampling

Properties

A specific case of the single component Metropolis-Hasting algorithm

It is designed to sample from a posterior with multiple parameters

$$p(\theta_1, \theta_2, \ldots, \theta_p | y, \mathbf{X})$$

For each step, the Gibb sampler cycles through the p parameters of θ where a sample is taken conditional on all other p – 1 parameters

$$f(\theta_i | y, \mathbf{X}, \theta_1, \ldots, \theta_i, \ldots, \theta_p)$$

Defining the (additional) important parts

Input data

This includes both the dependent and the independent variables

Define the joint posterior distributoin

$$f(\text{Model}, \text{Data})$$

Define the model (and thus the number of parameters) in the posterior

Define the conditional sampler for each parameter in terms of the joint posterior distribution f( )

$$C_i(\text{Model}_{t-1}, \text{Data})$$

Gibbs sampler `pseudocode` (in its simplest form)

```
for t in 1 to N
   for i in 1 to nparam
      draw theta[i,t] from C[i](theta[-i,t-1], y, X)
```

Exercice - Write your own Gibbs sampler

The question

Estimate posterior mean $\hat{\mu}$ and standard deviation $\hat{\sigma}$ of the following ten values

```
15    19.59 15.06 15.71 14.65
21.4  17.64 18.31 15.12 14.40
```

Prior specification

$$\mu \sim \mathcal{N}(\mu_0, \sigma_0^2) \text{ with } \mu_0 = 16, \sigma_0^2 = 0.4$$

$$\tau = \frac{1}{\sigma^2} \sim \mathcal{G}(\alpha_0, \beta_0) \text{ with } \alpha_0 = 1, \beta_0 = 3$$

Exercice - Write your own Gibbs sampler

A few guidelines

Recall that

$$\underbrace{\mathrm{P}(\mathrm{Model}|\mathrm{Data})}_{\text{Posterior}} \propto \underbrace{\mathrm{P}(\mathrm{Data}|\mathrm{Model})}_{\text{Likelihood}} \underbrace{\mathrm{P}(\mathrm{Model})}_{\text{Prior}}$$

$$\mathrm{P}(\boldsymbol{\theta}|\mathbf{Y}) \propto \mathrm{P}(\mathbf{Y}|\boldsymbol{\theta})\mathrm{P}(\boldsymbol{\theta})$$

Exercice - Write your own Gibbs sampler

A few guidelines

Define the different parts... Mathematically

Model

$$\boldsymbol{\theta} = (\hat{\mu}, \hat{\sigma})$$

Likelihood

$$P(\mathbf{Y}|\boldsymbol{\theta}) = \prod_{i=1}^{n} \frac{1}{\hat{\sigma}\sqrt{2\pi}} e^{-\frac{(Y_i-\hat{\mu})^2}{2\hat{\sigma}^2}}$$

Prior

$$P(\boldsymbol{\theta}) = (\hat{\mu}|\mu_0, \sigma_0)(\hat{\sigma}|\alpha_0, \beta_0)$$

$$= \frac{1}{\hat{\sigma}_0\sqrt{2\pi}} e^{-\frac{(\hat{\mu}-\mu_0)^2}{2\sigma_0^2}} \times \frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)} \hat{\tau}^{\alpha_0-1} e^{-\beta_0\hat{\tau}}$$

Exercice - Write your own Gibbs sampler

A few guidelines

Define the different parts... Mathematically

Joint posterior

$$P(\boldsymbol{\theta}|\mathbf{Y}) = \prod_{i=1}^{n} \frac{1}{\hat{\sigma}\sqrt{2\pi}} e^{-\frac{(Y_i-\hat{\mu})^2}{2\hat{\sigma}^2}} \times \frac{1}{\hat{\sigma}_0\sqrt{2\pi}} e^{-\frac{(\hat{\mu}-\mu_0)^2}{2\sigma_0^2}} \times \frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)} \hat{\tau}^{\alpha_0-1} e^{-\beta_0\hat{\tau}}$$

Exercice - Write your own Gibbs sampler

A few guidelines

How to sample each parameter independently

Mean ($\hat{\mu}$)

$$P(\hat{\mu}|\hat{\sigma}, \mathbf{Y}) \propto \prod_{i=1}^{n} \frac{1}{\hat{\sigma}\sqrt{2\pi}} e^{-\frac{(Y_i-\hat{\mu})^2}{2\hat{\sigma}^2}} \times \frac{1}{\hat{\sigma}_0\sqrt{2\pi}} e^{-\frac{(\hat{\mu}-\mu_0)^2}{2\sigma_0^2}}$$

Standard deviation ($\hat{\sigma}$)

$$P(\hat{\sigma}|\hat{\mu}, \mathbf{Y}) \propto \prod_{i=1}^{n} \frac{1}{\hat{\sigma}\sqrt{2\pi}} e^{-\frac{(Y_i-\hat{\mu})^2}{2\hat{\sigma}^2}} \times \frac{\beta_0^{\alpha_0}}{\Gamma(\alpha_0)} \hat{\tau}^{\alpha_0-1} e^{-\beta_0\hat{\tau}}$$

Exercice - Write your own Gibbs sampler

A few guidelines

How to sample each parameter independently

It is essential to use log-likelihood instead of the likehood when implementing the Gibb sampler

Mean ($\hat{\mu}$)

$$\log(P(\hat{\mu}|\hat{\sigma}, \mathbf{Y})) \propto \sum_{i=1}^{n} \log\left(\frac{1}{\hat{\sigma}\sqrt{2\pi}} e^{-\frac{(Y_i - \hat{\mu})^2}{2\hat{\sigma}^2}}\right) - \frac{(\hat{\mu} - \mu_0)^2}{2\sigma_0^2}$$

Standard deviation ($\hat{\sigma}$)

$$\log(P(\hat{\sigma}|\hat{\mu}, \mathbf{Y})) \propto \sum_{i=1}^{n} \log\left(\frac{1}{\hat{\sigma}\sqrt{2\pi}} e^{-\frac{(Y_i - \hat{\mu})^2}{2\hat{\sigma}^2}}\right) + \log(\tau)(\alpha_0 - 1) - \beta_0\hat{\tau}$$

This is one way. . . , there is another using conjugate prior

Exercice - Write your own Gibbs sampler

A few guidelines

How to sample parameters using conjugate priors

When using conjugate prior, we **do not** need to define the likelihood; to obtain the posterior we simply need to sample from the right conjugate prior distribution.

Important points to consider

- “Special” priors need to be define
- This gives less flexibility to the choice of priors we can choose
- It is computationally faster

Exercice - Write your own Gibbs sampler

A few guidelines

How to sample parameters using conjugate priors

At this point, Wikipedia can be **very** useful

https://en.wikipedia.org/wiki/Conjugate_prior

It tells us that

$$\hat{\mu} \sim \mathcal{N}\left(\frac{\frac{\mu_0}{\sigma_0^2} + \frac{\sum_{i=1}^{n} \mathbf{Y}_i}{\sigma^2}}{\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2}}\right)$$

$$\hat{\tau} \sim \mathcal{G}\left(\alpha + \frac{n}{2}, \beta + \frac{\sum_{i=1}^{n} (\mathbf{Y}_i - \hat{\mu})}{2}\right)$$

If we sample $\hat{\mu}$ and $\hat{\tau}$ from the previous distributions we should be able to obtain the “true” value.

At this point, the choice of priors may affect the result. Try different ones !!

Bonus - An ecological problem

How important is elevation in defining sugar maple distribution on mont Sutton?

Model

$$y = \beta_0 + \beta_1 x + \varepsilon$$

y	is the abundance (or presence-absence. . . you decide!) of sugar maple
x	is elevation
β_0	is an intercept
β_1	is the importance of elevation
ε	is the model residuals

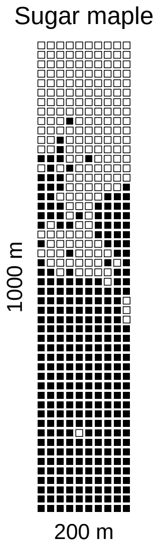
Things to think about

What data you choose to analyse

The type of regression performed

The way you write the likelihood

The priors (what do you know or don't know)


Sugar maple
1000 m
200 m