# A few Bayesian lectures for the Uninitiated

—

# Part 3: MCMC


Carsten F. Dormann

Biometry & Environmental System Analysis, University of Freiburg


16. Februar 2022

# Bayes?!

Remember Bayes' Theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{1}$$

# Bayes?!

Remember Bayes' Theorem:

$$\boxed{P(A|B) = \frac{P(B|A)P(A)}{P(B)}} \tag{1}$$

Or, in the terms of fitting a model:

$$\boxed{P(\theta|\text{data}) = \frac{P(\text{data}|\theta)P(\theta)}{P(\text{data})}} \tag{2}$$

# Bayes?!

Remember Bayes' Theorem:

$$\boxed{P(A|B) = \frac{P(B|A)P(A)}{P(B)}} \tag{1}$$

Or, in the terms of fitting a model:

$$\boxed{P(\theta|\text{data}) = \frac{P(\text{data}|\theta)P(\theta)}{P(\text{data})}} \tag{2}$$

Likelihood and prior are fine now, but what is that strange denominator, $P(\text{data})$?

# The Bayes denominator

$P(\text{data})$ is independent of the model parameters $\theta$, so it is constant for a given data set.

# The Bayes denominator

$P(\text{data})$ is independent of the model parameters $\theta$, so it is constant for a given data set.
But one cannot compute the probability of data without making assumptions about the probability distribution, and a distribution has parameters!

# The Bayes denominator

$P(\text{data})$ is independent of the model parameters $\theta$, so it is constant for a given data set.

But one cannot compute the probability of data without making assumptions about the probability distribution, and a distribution has parameters!

So, we have to put the parameters back in:

$$P(\text{data}) = \int_{\Omega} P(\text{data}|\theta)P(\theta)d\theta$$

# The Bayes denominator

$P(\text{data})$ is independent of the model parameters $\theta$, so it is constant for a given data set.
But one cannot compute the probability of data without making assumptions about the probability distribution, and a distribution has parameters!
So, we have to put the parameters back in:

$$P(\text{data}) = \int_{\Omega} P(\text{data}|\theta)P(\theta)d\theta$$

This means:
We compute the probability of the data by integrating over all values of $\theta$ (aka "integrating out" or "marginalising over" $\theta$).
(The $\Omega$ is the parameter space, i.e. all possible value combinations for multidimensional $\theta$.)

# The trick (I hope I got that right)

- An integral is just a sum over very many small values.

# The trick (I hope I got that right)

- An integral is just a sum over very many small values.
- The sum is just the mean times $N$: $\sum P_i(x) = N \cdot \overline{P(x)}$

# The trick (I hope I got that right)

- An integral is just a sum over very many small values.
- The sum is just the mean times $N$: $\sum P_i(x) = N \cdot \overline{P(x)}$
- The integral/sum needs to be 1, since $P(X)$ is a probability distribution.

# The trick (I hope I got that right)

- An integral is just a sum over very many small values.
- The sum is just the mean times $N$: $\sum P_i(x) = N \cdot \overline{P(x)}$
- The integral/sum needs to be 1, since $P(X)$ is a probability distribution.
- As long as we sample $X$ proportional to $P(X)$, we can simply compute the mean of **x** and voilà, we have our integral:

$$\sum_{i \in \text{small steps from min to max } x} P(x_i) = \sum_{i \propto P(x)} x_i = \bar{x}$$

# The trick (I hope I got that right)

- An integral is just a sum over very many small values.
- The sum is just the mean times $N$: $\sum P_i(x) = N \cdot \overline{P(x)}$
- The integral/sum needs to be 1, since $P(X)$ is a probability distribution.
- As long as we sample $X$ proportional to $P(X)$, we can simply compute the mean of **x** and voilà, we have our integral:

$$\sum_{i \in \text{small steps from min to max } x} P(x_i) = \sum_{i \propto P(x)} x_i = \bar{x}$$

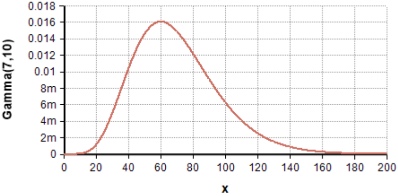
Problem: How to sample $X$ proportional to $P(X)$?

# MCMC: sampling $X$ proportional to $P(X)$!

Solution: Markov chain Monte Carlo (MCMC)

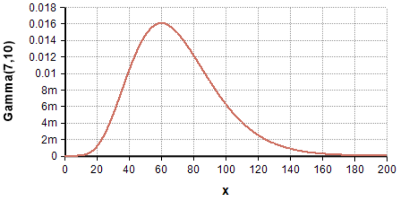
# MCMC: sampling $X$ proportional to $P(X)$!

Solution: Markov chain Monte Carlo (MCMC)



# MCMC: sampling $X$ proportional to $P(X)$!

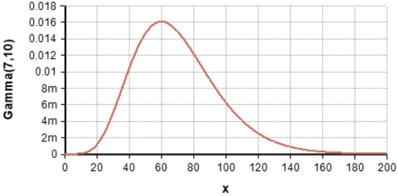
Solution: Markov chain Monte Carlo (MCMC)




Imagine drawing random values of $X$, say $x = 1.2, 20, 62$ and 140. Are these probable values?

# MCMC: sampling $X$ proportional to $P(X)$!

Solution: Markov chain Monte Carlo (MCMC)




Imagine drawing random values of $X$, say $x = 1.2, 20, 62$ and 140. Are these probable values?

Enter the "Metropolis" algorithm.

In line with Stigler's Law of Eponymy, the "Metropolis" algorithm was invented by physicist Marshall Rosenbluth, after being posed the problem by Ed Teller, and first implemented by his wife Arianna Rosenbluth, but written up by Metropolis and these three, plus Ed Teller's wife Augusta, in 1953.

# MCMC: the Metropolis algorithm

1. Compute $P(x_1)$ for some initial value $x_1$ of $X$.

# MCMC: the Metropolis algorithm

1. Compute $P(x_1)$ for some initial value $x_1$ of $X$.
2. Draw a new value, $x_2$, randomly from a normal distribution around $x_1$.

# MCMC: the Metropolis algorithm

1. Compute $P(x_1)$ for some initial value $x_1$ of $X$.
2. Draw a new value, $x_2$, randomly from a normal distribution around $x_1$.
3. Compute $P(x_2)$, and from that $\alpha = \frac{P(x_2)}{P(x_1)}$.

# MCMC: the Metropolis algorithm

1. Compute $P(x_1)$ for some initial value $x_1$ of $X$.
2. Draw a new value, $x_2$, randomly from a normal distribution around $x_1$.
3. Compute $P(x_2)$, and from that $\alpha = \frac{P(x_2)}{P(x_1)}$.
4. Draw a random uniform number $u$ in $[0, 1]$. If $u < \alpha$, accept $x_2$ as new value, otherwise reject it and set $x_2 = x_1$.

# MCMC: the Metropolis algorithm

1. Compute $P(x_1)$ for some initial value $x_1$ of $X$.
2. Draw a new value, $x_2$, randomly from a normal distribution around $x_1$.
3. Compute $P(x_2)$, and from that $\alpha = \frac{P(x_2)}{P(x_1)}$.
4. Draw a random uniform number $u$ in $[0, 1]$. If $u < \alpha$, accept $x_2$ as new value, otherwise reject it and set $x_2 = x_1$.

Side notes:

Ad 2. The "proposal distribution", is typically $\mathcal{N}(x_1, \sigma)$. If asymmetric, some fudging is needed ("Metropolis-Hastings").

# MCMC: the Metropolis algorithm

1. Compute $P(x_1)$ for some initial value $x_1$ of $X$.
2. Draw a new value, $x_2$, randomly from a normal distribution around $x_1$.
3. Compute $P(x_2)$, and from that $\alpha = \frac{P(x_2)}{P(x_1)}$.
4. Draw a random uniform number $u$ in $[0, 1]$. If $u < \alpha$, accept $x_2$ as new value, otherwise reject it and set $x_2 = x_1$.

Side notes:

Ad 2. The "proposal distribution", is typically $\mathcal{N}(x_1, \sigma)$. If asymmetric, some fudging is needed ("Metropolis-Hastings").

Ad 3. If $P(x_2) > P(x_1)$, then $\alpha > 1$ and the proposal is always accepted.

# MCMC: the Metropolis algorithm

1. Compute $P(x_1)$ for some initial value $x_1$ of $X$.
2. Draw a new value, $x_2$, randomly from a normal distribution around $x_1$.
3. Compute $P(x_2)$, and from that $\alpha = \frac{P(x_2)}{P(x_1)}$.
4. Draw a random uniform number $u$ in $[0, 1]$. If $u < \alpha$, accept $x_2$ as new value, otherwise reject it and set $x_2 = x_1$.

Side notes:

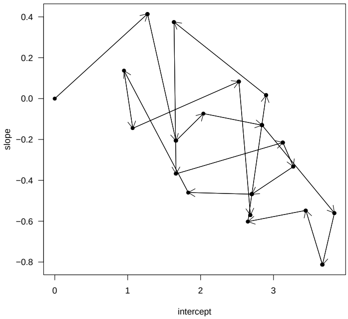
Ad 2. The "proposal distribution", is typically $\mathcal{N}(x_1, \sigma)$. If asymmetric, some fudging is needed ("Metropolis-Hastings").

Ad 3. If $P(x_2) > P(x_1)$, then $\alpha > 1$ and the proposal is always accepted.

Ad 4. Acceptance is thus more likely the more similar the likelihood of $x_1$ and $x_2$ are.

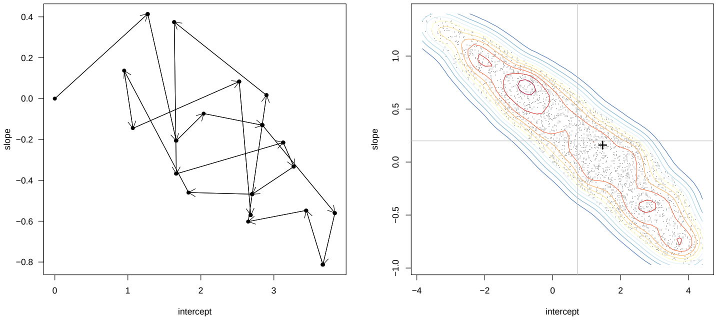
# MCMC: the Metropolis random walk

The above MCMC is an *autocorrelated* random walk in parameter space, spending more time at higher probabilities.



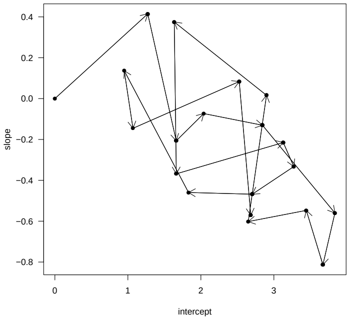
# MCMC: the Metropolis random walk

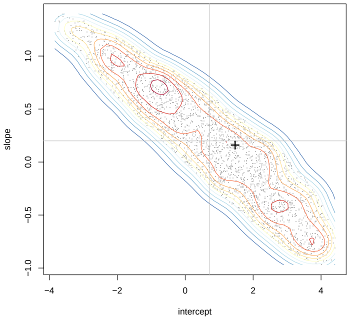
The above MCMC is an *autocorrelated* random walk in parameter space, spending more time at higher probabilities.



# MCMC: the Metropolis random walk

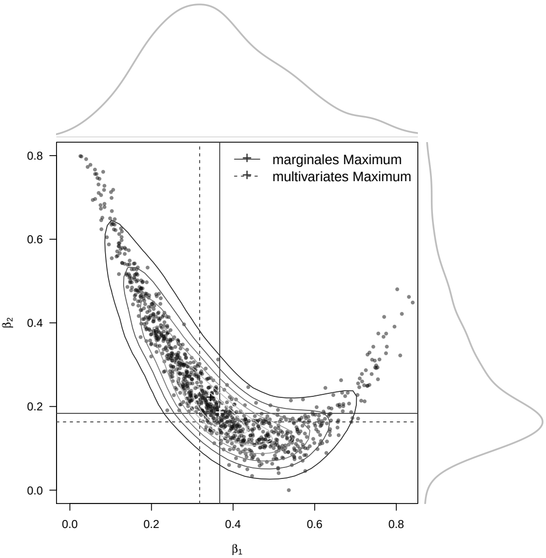
The above MCMC is an *autocorrelated* random walk in parameter space, spending more time at higher probabilities.




For many parameters, proposals are often drawn conditionally on other dimensions ("Gibbs sampling", hence JAGS).

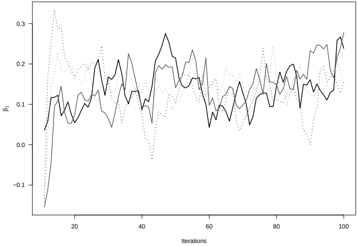
# MCMC: the “banana” problem

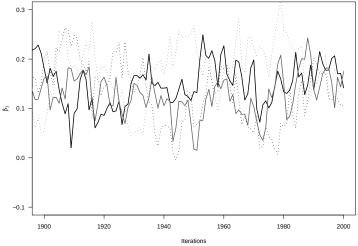



multivariate MAP $\neq$ marginal MAP

# MCMC: mixing and convergence

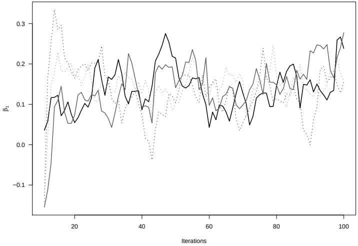
How many samples do we need to work out the shape of the posterior?

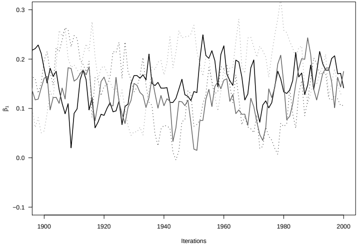


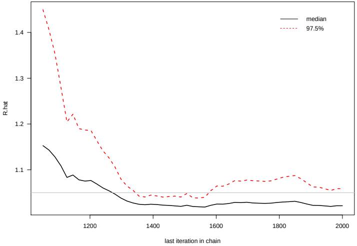



# MCMC: mixing and convergence

How many samples do we need to work out the shape of the posterior?



# MCMC: take-home-message

- Markov chain Monte Carlo is an ingenious way to sample proportional to an unknown distribution (e.g. the model's likelihood).
- It is slow and requires attention after fitting.
- MCMC is a must-know algorithm/concept in statistics, far beyond Bayes.
- If you want, you can use MCMC as robust-but-slow optimiser (if so, check modern extensions, such as DREAM, DEzs, SMC as implemented in **BayesianTools**).

**Next time:**
**Crazy little thing called *prior*?**