Bayesian statistics, simulation and software

Module 7: More Markov chain Monte Carlo methods

Jesper Møller and Ege Rubak

Department of Mathematical Sciences
Aalborg University

Reminder: The Bayesian idea

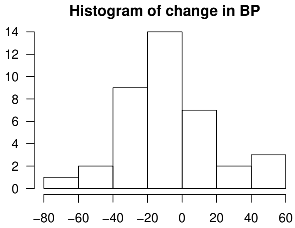
- **Data model**: Given parameter θ the data, X, is assumed to be distributed as
$$X \sim \pi(x|\theta).$$
- Parameter θ of interest is unknown.
- A priori knowledge is summarised in terms of **prior density**
$$\pi(\theta).$$
- Conditional on the observed data x, we obtain the **posterior density**
$$\pi(\theta|x) \propto \pi(x|\theta)\pi(\theta).$$
- All **conclusions** regarding θ are based on the posterior.

Example: Blood pressure

Setup: A group of $n = 38$ patients with high blood pressure are given a new drug.

Let x_i denote the change in blood pressure, $i = 1, \ldots, n$ ($x_i < 0$ is "good").

Data:



Data model:

$$x_1, \ldots, x_{38} \overset{iid}{\sim} \mathcal{N}(\mu, \tau).$$

Question: Is there a positive effect of the drug, that is, $\mu < 0$?

Bayesian analysis of blood pressure

Prior: (assume τ known)

$$\pi(\mu) \sim \mathcal{N}(\mu_0, \tau_0).$$

Posterior:

$$\pi(\mu|\mathbf{x}) \sim \mathcal{N}\left(\frac{n\tau\bar{x} + \tau_0\mu_0}{n\tau + \tau_0}, n\tau + \tau_0\right).$$

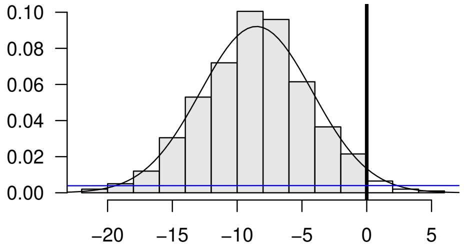
Question: What is the posterior probability that $\mu < 0$ (the (mean) effect of the drug is "good")? That is, calculate

$$P(\mu < 0|\mathbf{x}).$$

Answer: $P(\mu < 0|\mathbf{x}) = 0.975$ (using the normal posterior from above with a certain choice of the values of τ, μ_0, τ_0 – here, it is not important what these values are – we just want to discuss answers obtained by simulations).

Monte Carlo: Simulating an answer (imaging we couldn't do the calculation above)

- Notice that $P(\mu < 0|\mathbf{x}) = \mathbb{E}\left[\mathbb{1}[\mu < 0]\middle|\mathbf{x}\right]$.
- Simulate $\mu^{(1)}, \mu^{(2)}, \dots, \mu^{(1000)} \sim \mathcal{N}\left(\frac{n\tau\bar{x}+\tau_0\mu_0}{n\tau+\tau_0}, n\tau + \tau_0\right)$.
  Histogram:

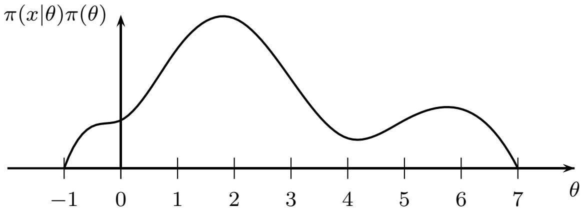


- Monte Carlo estimate of $P(\mu < 0|\mathbf{x})$:
  $\frac{1}{1000}\sum_{i=1}^{1000} \mathbb{1}[\mu^{(i)} < 0] = 0.981$ (close to the correct value 0.975).

New problem

- The posterior is $\pi(\theta|x) \propto \pi(x|\theta)\pi(\theta)$.
- Note that $\pi(x|\theta)\pi(\theta)$ is *not* a normalised density (unless $\pi(x) = 1$).

Assume the following is a plot of $\pi(x|\theta)\pi(\theta)$ vs θ:



Question: What is the probability $P(\theta > 5|x)$?

A "half answer": Simulations of $\pi(\theta|x)$ could answer this.

Problem: Often $\pi(\theta|x)$ is not (proportional to) a well-known density.

Solution: Use Markov chain Monte Carlo (MCMC) methods...

Markov chain: Mathematical definition

Definition: Markov chain

A sequence of random variables $(X^{(0)}, X^{(1)}, X^{(2)}, \ldots)$ with state space Ω is called a *time-homogeneous* Markov chain if for all events $A \subseteq \Omega$, all times $t = 0, 1, \ldots$ and all states $x^{(0)}, x^{(1)}, \ldots, x^{(t)} \in \Omega$, we have

$$\begin{aligned} &P(X^{(t+1)} \in A | X^{(0)} = x^{(0)}, X^{(1)} = x^{(1)}, \ldots, X^{(t)} = x^{(t)}) \\ &\quad = P(X^{(t+1)} \in A | X^{(t)} = x^{(t)}) \\ &\quad = P(x^{(t)}, A), \end{aligned}$$

where $P(x, A)$ is called the **transition kernel**.

Example of a time-homogeneous Markov chain: A Gibbs sampler after each sweep.

Markov chain: A more useful definition for implementations

Definition: Markov chain

A sequence of random variables $(X^{(0)}, X^{(1)}, X^{(2)}, \dots)$ with state space Ω is called a *time-homogeneous* Markov chain if

$$X^{(t+1)} = \phi(X^{(t)}, U^{(t)}), \qquad t = 0, 1, \dots,$$

where ϕ is a deterministic function (the **"update function"**) and $U^{(0)}, U^{(1)}, \dots$ are IID (multivariate) RVs (the **"random bits"**).

In practice the update function and the random bits are given by your computer code!

Markov chain

Definition: n-step transition kernel

For any $t = 0, 1, \ldots$ and $n = 1, 2, \ldots$, the n-**step transition kernel** is

$$P^n(x, A) = P(X^{(t+n)} \in A | X^{(t)} = x).$$

That is, for $t = 0$ and conditional on $X^{(0)} = x$ we consider the distribution of

$X^{(1)} = \phi(x, U^{(0)}) \qquad$ ($n = 1$, i.e. one step),

$X^{(2)} = \phi(X^{(1)}, U^{(1)}) = \phi(\phi(x, U^{(0)}), U^{(1)}) \qquad$ ($n = 2$, i.e. two steps),

$X^{(3)} = \phi(X^{(2)}, U^{(2)}) = \phi(\phi(\phi(x, U^{(0)}), U^{(1)}), U^{(2)}) \quad$ ($n = 3$, i.e. three steps),

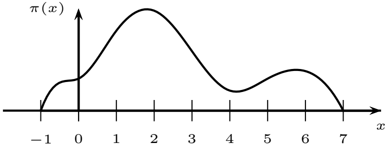
$\vdots$

Simulating more answers

The Gibbs sampler relied on generating samples from standard/tractable distributions for the full conditionals. What if we want to sample from a non-standard density?

Let $\pi(x)$ be a *target density*, i.e. a density we want to sample from (e.g. a posterior density).

As an example, we may want to generate a sample from this density:



Idea: Want to generate a Markov chain $(X^{(0)}, X^{(1)}, X^{(2)}, \ldots)$ so that $X^{(t)}$ (for sufficiently large t) is (approximately) a sample from the density $\pi(x)$.

Propose and accept/reject algorithm

Let $\pi(x)$ be our *target density*, and for each given x, let $q(x, y)$ be a density w.r.t. y, and let $a(x, y)$ be a probability (i.e. $0 \leq a(x, y) \leq 1$).

Propose and accept/reject algorithm for a homogeneous Markov chain

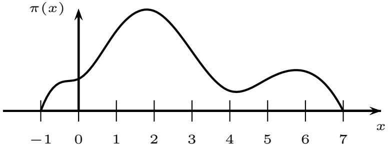
Choose initial value $X^{(0)} = x^{(0)}$.
For $t = 1, 2, \ldots, T$

1. Generate (a new) **proposal** $Y = y \sim q(X^{(t-1)}, y)$.
2. Accept proposal with probability $a(X^{(t-1)}, y)$ otherwise reject it. That is, generate (a new) $U = u \sim \text{Unif}(0, 1)$, and
3. if **acceptance**, i.e. $u \leq a(X^{(t-1)}, y)$: $X^{(t)} = y$;
4. else **reject**: $X^{(t)} = X^{(t-1)}$.

Main problem: How do we choose $q(x, y)$ and $a(x, y)$ so that (approximately/effectively) $\boldsymbol{X}^{(t)} \sim \pi(x)$ (for large t)? We will be flexible with the choice of proposal density $q(x, y)$, which then will determine the acceptance probability $a(x, y)$ as explained later.

Example of the random walk Metropolis Algorithm

Recall that we want to generate a sample from a distribution with this density:



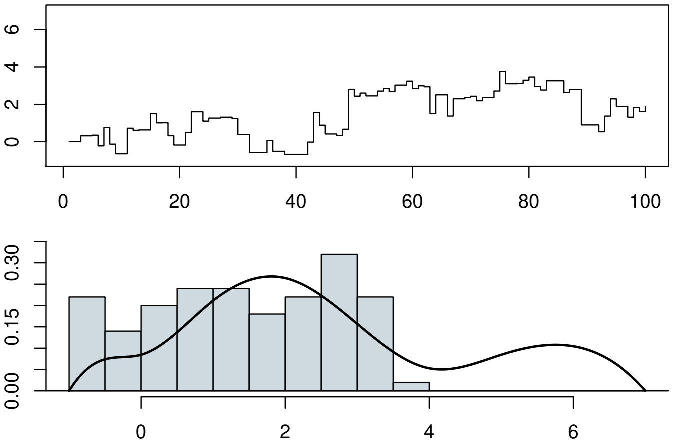
One possibility is a so-called **random walk Metropolis algorithm**, with a **normal proposal distribution** centred at the current value, and with a user-specified precision τ_p:

$$q(x, y) = \sqrt{\frac{\tau_p}{2\pi}} \exp\left(-\frac{1}{2}\tau_p(y - x)^2\right).$$

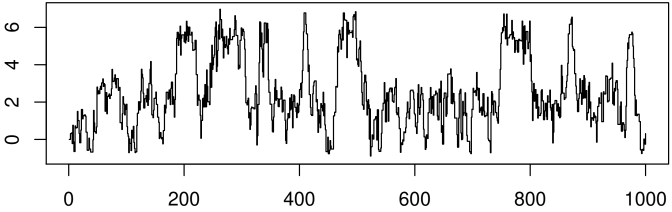
Then the **acceptance probability** becomes

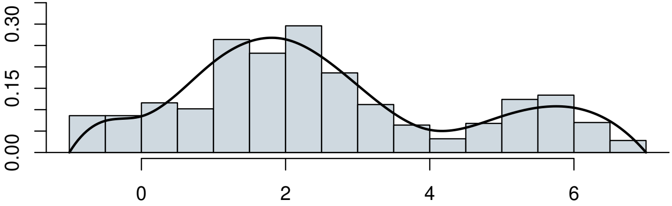
$$a(x, y) = \min\left\{1, \frac{\pi(y)}{\pi(x)}\right\}.$$

Results: 100 iterations

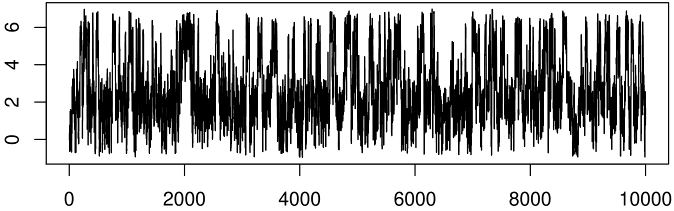


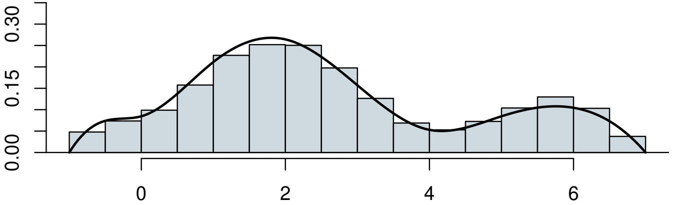
Results: 1000 iterations





Results: 10000 iterations





The Metropolis-Hastings algorithm

A very flexible/general MCMC algorithm is the **Metropolis-Hastings algorithm**: The user specifies a proposal kernel $q(x,y)$, and the algorithm then uses the correct acceptance probability $a(x,y)$:

Metropolis-Hastings algorithm

- Choose any *proposal kernel* $q(x,y)$ (i.e. $q(x,\cdot)$ is a density for any x).
- Define the *Hastings ratio*

$$H(x,y) = \frac{\pi(y)q(y,x)}{\pi(x)q(x,y)},$$

  where $H(x,y) = \infty$ if $\pi(x)q(x,y) = 0$.
- The acceptance probability is

$$a(x,y) = \min\{1, H(x,y)\}.$$

Remark: We need only to know $\pi(x) \propto \pi_0(x)$ (an **"unnormalized density"**). So useful when dealing with a posterior density $\pi(\theta|x) \propto \pi_0(\theta) = \pi(\theta|x)\pi(\theta)$.

The Metropolis algorithm

A special case of the MH-algorithm is when the proposal density is symmetric:

$$q(x,y) = q(y,x).$$

Then the Hastings ratio simplifies to

$$H(x,y) = \frac{\pi(y)q(y,x)}{\pi(x)q(x,y)} = \frac{\pi(y)}{\pi(x)}.$$

Example: The most common example is the random walk Metropolis algorithm with normal proposals, i.e. when the proposal is normally distributed with x as the mean value and τ_p as the (user-specified) precision:

$$q(x,y) = \sqrt{\frac{\tau_p}{2\pi}} \exp\left(-\frac{1}{2}\tau_p(y-x)^2\right).$$

Clearly, $q(x,y) = q(y,x)$.

The independent Metropolis-Hastings algorithm

Another special case is when the proposal density is independent of the current state x:

$$q(x, y) = q(y).$$

However, the Hastings ratio

$$H(x, y) = \frac{\pi(y)q(y, x)}{\pi(x)q(x, y)} = \frac{\pi(y)q(x)}{\pi(x)q(y)}$$

depends on both the current state x and the proposal y.

This independent Metropolis-Hastings algorithm is mainly of theoretical interest or when developing generic software.

In practice the acceptance probabilities may often be small, meaning that the Markov chain easily get stuck in the same state for a long time, and hence one may prefer to use another Metropolis-Hastings algorithm.