Lecture 8B
MTH6102: Bayesian Statistical Methods

Eftychia Solea

Queen Mary University of London

2023

Today's agenda

Today's lecture

- Review
- Understand Markov Chain
- Understand Metropolis-Hastings
- Apply Metropolis-Hastings in Bayesian inference to generate samples from the posterior pdf.

Review: Monte Carlo integration

- The Monte Carlo integration refers to the theory and practice of approximating integrals using random samples.
- Monte Carlo integration methods are sometimes referred to as stochastic integration methods because they are based on random sampling.

Example

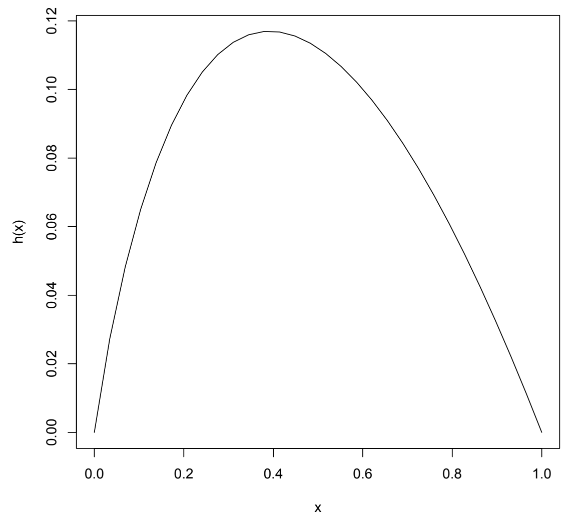
- Suppose that we wish to compute the following integral

$$I = \int_0^1 \frac{\sin(x(1-x))}{1+x+\sqrt{x}}\,dx.$$

- There does not appear to be any closed-form solution, so we can approximate the integral using Monte Carlo methods.

Example

- Plot of the function $h(x) = \frac{\sin(x(1-x))}{1+x+\sqrt{x}}$, $x \in [0, 1]$,



- Using a deterministic numerical integration method in R based on adaptive quadrature of functions, $I = 0.079$.

```
>h=function(x) sin(x*(1-x))/(1+x+sqrt(x))
>integrate(h,lower=0,upper=1)
>0.07852747 with absolute error < 9.2e-06
```

Example

- **Goal:** Compute the integral I using Monte Carlo integration

- The integral, I, can be written as the expectation of $h(X)$, where $X \sim U[0,1]$ and

$$h(x) = \frac{\sin(x(1-x))}{1+x+\sqrt{x}},\ x \in [0,1].$$

- Because,

$$I = \int_0^1 \frac{\sin(x(1-x))}{1+x+\sqrt{x}} \cdot 1\, dx = \int_0^1 h(x) f(x)\, dx = E(h(X)),$$

where $f(x) = 1, x \in [0,1]$ is the $U[0,1]$ density.

$$I = \mathbb{E}_f(h(X)) = \int h(x) f(x)\, dx$$

f does not need to be uniform but a well-known density

In this case, you generate $X_1, \ldots, X_n \sim f$

and the Monte Carlo estimator is

$$\frac{1}{N} \sum_{i=1}^{N} h(X_i)$$

Example

- Thus, we can generate IID observations $X_1, \ldots, X_N$ from $U[0,1]$, and estimate I by

$$\hat{I} = \frac{1}{N}\sum_{i=1}^{N} h(X_i) = \frac{1}{N}\sum_{i=1}^{N} \frac{\sin(X_i(1-X_i))}{1+X_i+\sqrt{X_i}}.$$

- $\hat{I}$ is the Monte Carlo integration estimator of $I = E(h(X))$, $X \sim U[0,1]$.

Example

- Then, the weak law of large numbers (WLLN) says

$$\hat{I} = \frac{1}{N}\sum_{i=1}^{N} h(X_i) \xrightarrow{P} I, \quad N \to \infty,$$

- As we use more samples N, $\hat{I}$ should get more and more accurate.

$$\hat{I} = \frac{1}{N}\left(h(X_1) + \cdots + h(X_N)\right)$$

Example

- In R

```
h=function(x) sin(x*(1-x))/(1+x+sqrt(x))
x=runif(1000)
mean(h(x))
```

- $\hat{I} = 0.081$, and it is the Monte Carlo estimate of the integral I.

Bayesian inference

- Quite often a quantity of interest in statistics may be expressed as an integral that we wish to evaluate.
- For instance, in Bayesian analysis, one is often interested in the posterior mean of a particular continuous parameter θ

$$\hat{\theta}_B = \int \theta \, p(\theta \mid y) \, d\theta,$$

- or in the posterior mean of transformed parameters $\psi = g(\theta)$

$$\hat{\psi}_B = \int g(\theta) \, p(\theta \mid y) \, d\theta,$$

- or in the posterior probability,

$$P(\theta \leq c) = \int_{-\infty}^{c} p(\theta \mid y) \, d\theta = \int_{-\infty}^{\infty} I(\theta \leq c) \, p(\theta \mid y) \, d\theta = E(I(\theta \leq c)).$$

Board example: Poisson data/Gamma prior

- Suppose we observe iid data $y_1, \ldots, y_n$ from Poisson(λ).
- Let λ have the gamma(α, β) prior distribution, the conjugate prior distribution for the Poisson likelihood.

1. What is the posterior density, $p(\lambda \mid y)$ of λ?
2. How would you estimate the posterior probability $P(\lambda < c)$ by Monte Carlo integration, for some $c > 0$?

Solution

(1) From the ex-sheet 5 (Problem 3) and ex. sheet 6 (Problem 2), the posterior density, $p(\lambda|y)$, of λ is

$$p(\lambda|y) \sim \text{Gamma}\left(a + \sum_{i=1}^{n} y_i,\ \beta + n\right), \quad \lambda > 0$$

(2) We want to compute $P(\lambda < c)$ using Monte Carlo Integration.

First, note that

$$P(\lambda < c) = \int_0^\infty I(\lambda < c)\, p(\lambda|y)\, d\lambda = \mathbb{E}(\underbrace{I(\lambda < c)}_{h(\lambda)}) = I$$

We generate an IID sample $\Lambda^{(1)}, \dots, \Lambda^{(N)}$ from $\text{Gamma}\left(a + \sum^{n} y_i, n + \beta\right)$

then

$$\hat{I} = \frac{1}{N} \sum_{i=1}^{N} I(\Lambda^{(i)} < c) = \frac{1}{N} \sum_{i=1}^{N} h(\Lambda^{(i)})$$

Board question: Binomial data/flat prior

- Let $k \sim \text{binom}(n, q)$.
- Assume flat prior on q.
- Let $n = 860$ and $k = 441$
- R code below

```
a=1
b=1
n=860
k=441
N=10000
beta.post.sample=rbeta(N, shape1=a+k,shape2=b+n-k)
gamma.sample=log((beta.post.sample/(1-beta.post.sample)))
mean(gamma.sample)
c(quantile(gamma.sample,0.025),quantile(gamma.sample,0.975))
```

- The first command will contain a sample of 10^4 observations from the posterior density, $\text{beta}(1+x, 1+n-x)$, say $q^{(1)}, \ldots, q^{(N)}$

The second will contain a sample from $h(q) = \log\left(\frac{q}{1-q}\right)$ (log of odds)

$\{h(q_1), \ldots, h(q_N)\}$ = gamma.sample

- The estimator $\hat{\theta}$ is the sample mean of $h(q_1), \ldots, h(q_N)$:

$$\hat{\theta} = \frac{1}{N} \sum_{i=1}^{N} h(q_i) = \frac{1}{N} \sum_{i=1}^{N} \log\left(\frac{q_i}{1-q_i}\right)$$

This is the Monte Carlo estimator of θ.

θ is the posterior mean of $h(q)$

$$\theta = \mathbb{E}\left(h(q)\right) = \mathbb{E}\left(\log \frac{q}{1-q}\right) = \int \log\left(\frac{q}{1-q}\right) p(q|x)\, dq$$

$p(q|x) \sim \text{beta}(1+x, 1+n-x)$.

Board question: Binomial data/flat prior

- When this code has run, what will `beta.post.sample` contain? What will `gamma.sample` contain?
- Describe the estimator $\hat{\theta}$ for a quantity θ (which you should also determine) that would be obtained by the following R commands

```
gamma.sample=log((beta.post.sample/(1-beta.post.sample)))
mean(gamma.sample)
```

- In statistical terms, what quantity will the last line of code output?
- See also, **Question 3, final exam Jan 2023**

Markov Chain Monte Carlo (MCMC)

- Monte Carlo integration estimates $E_f[h(X)]$ by directly sampling iid samples from the pdf f or from the posterior pdf in Bayesian inference

$$\hat{I} = \frac{1}{N}\sum_{i=1}^{N} h(X_i), \quad X_1, \ldots, X_n \text{ iid} \sim f.$$

- **Question:** But what if we cannot sample directly from f?
  - f is not analytically tractable.
  - Then, simple Monte Carlo integration cannot be used.

MCMC can help when f is intractable

- Markov Chain Monte Carlo (MCMC) is a set of methods that can generate a sample with pdf f without having to sample from f directly.
- Thus, MCMC can be used to generate samples from complicated probability distributions.
- At the price, however, of yielding **dependent** observations that are **approximately** from f.

Markov Chain Monte Carlo (MCMC)

Goal: f is density very complicated
want to generate a sample from f.

- The general **idea** of Markov Chain Monte Carlo (MCMC) methods is to construct a sequence of RV $X_1, X_2 \ldots,$, called Markov chain, which (hopefully) converges to the distribution of interest f.
- However, $X_1, X_2 \ldots,$ is NOT independent any more.
- But it can still be used to estimate quantities (e.g, mean) because there is a WLLN for Markov chains.
- Under certain conditions,

$$\hat{I} = \frac{1}{N}\sum_{i=1}^{N} h(X_i) \xrightarrow{P} E[h(X)] = I, \quad \text{as } N \to \infty.$$

NOT IID ⟸ $X_1, \ldots, X_N$ (Markov chains)

What is a Markov Chain?

- **Definition (Markov Chain).** A Markov chain is a sequence $X_1, X_2, \ldots$ of random variables such that the probability distribution of X_i (pmf or pdf) only depends on the previous value X_{i-1}

$$p(X_i \mid X_1, X_2, \ldots X_{i-2}, X_{i-1}) = p(X_i \mid X_{i-1}).$$

past — present

- The process depends on the past only through the present.

Example: Random walk

- As an example of a Markov chain, suppose $X_1 = 1$, and for $i > 1$

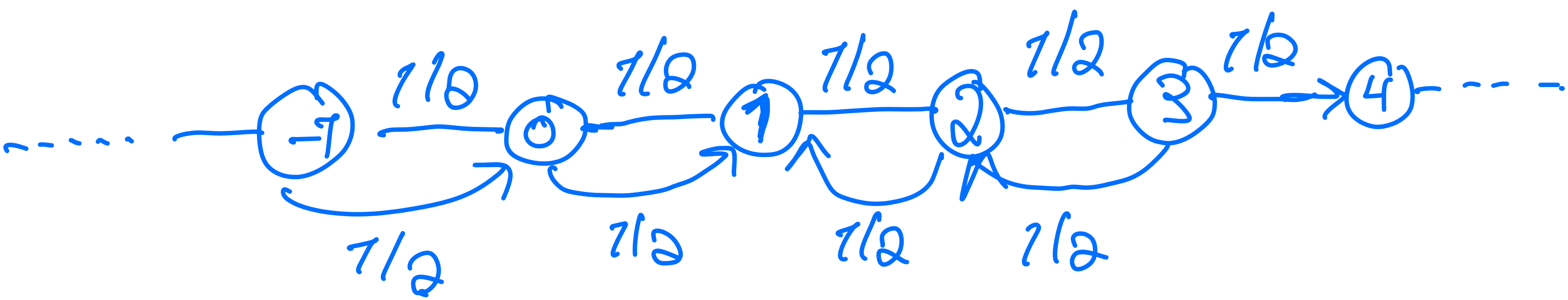
$$P(X_i = X_{i-1} + 1) = 1/2,$$
$$P(X_i = X_{i-1} - 1) = 1/2.$$

$i = 1$.

- This is a random walk starting at $X_1 = 1$.
- It can go off towards $-\infty$ or ∞ without limit.

Example: Random walk

- So you flip a coin move $+1$ steps if heads, move -1 steps if tails.
- At step i of this Markov chain, X_{i-1} is either increased or decreased by 1.
- Each possibility happens with probability $\frac{1}{2}$.



Metropolis-Hastings algorithm

- The Metropolis-Hastings algorithm is a type of Markov chain Monte Carlo (MCMC) that works as follows.
- Let $q(y|x)$ be a conditional density that we know how to sample from.
- $q(y|x)$ is called the proposal distribution.
- The Metropolis-Hastings algorithm creates a Markov Chain (dependent observations) $X_1, X_2, \ldots$ as follows.

Metropolis-Hastings algorithm

Choose X_1 arbitrarily. Suppose we have generated $X_1, \ldots, X_i$. To generate X_{i+1} do the following:

1. Generate a proposal or candidate random value $Y \sim q(y|X_i)$.
2. Evaluate $r \equiv r(X_i, Y)$ where

$$r(x, y) = \min\left\{\frac{f(y)}{f(x)}\frac{q(x|y)}{q(y|x)}, 1\right\}.$$

3. Generate $U \sim U(0,1)$. If $U \leq r$, set $X_{i+1} = Y$, otherwise set $X_{i+1} = X_i$.

Metropolis algorithm terminology

- q is the proposal distribution: we propose new rv Y using the conditional distribution $q(\cdot \mid X_i)$ that depends on X_i (not on the past).

- MH accepts Y with probability $r \equiv r(X_i, Y) = \min\left\{\frac{f(Y)}{f(X_i)}\frac{q(X_i|Y)}{q(Y|X_i)}, 1\right\}$, called the acceptance probability.

Metropolis algorithm terminology

- f is sometimes called the target distribution: this is what we are aiming for, i.e. we want to generate a sample with pdf f.

- In Bayesian inference, f would be the posterior distribution $p(\theta \mid y)$, and we want a sample of θ values from this posterior distribution.

Choosing events in computer code

Remarks:

- In general, to implement a random event that happens with probability r:
- Generate $u \sim$ Uniform$(0, 1)$;
- Event happens if $u \leq r$.
- If U is a random variable, with $U \sim$ Uniform$(0, 1)$, then U has cdf $F(r) = r$, so $P(U \leq r) = r$.

Metropolis-Hastings algorithm

Remarks:

- A common choice for $q(y|x)$ is $N(x, b^2)$ for some $b > 0$.
- This means that the proposal Y is a drawn from normal centered at the current value.
- By symmetry, $q(y|x) = q(x|y)$

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

Remarks:

- In the algorithm, f only appears in acceptance probability

$$r(X_i, Y) = \min\left\{1, \frac{f(Y)}{f(X_i)}\right\}.$$

- The acceptance probability **does not depend on the normalisation constant**, i.e. if $f(x) = cg(x)$, where $c > 0$ doesn't depend on x, then

$$r(X_i, Y) = \min\left\{1, \frac{g(Y)}{g(X_i)}\right\}.$$

- So we only need to know f up to a normalisation constant. Useful for Bayesian inference!

Output of the Metropolis-Hastings algorithm

1. The Metropolis-Hastings algorithm generates a dependent sequence of observations $X_1, X_2, \ldots$.

2. Since our procedure for generating X_{i+1} depends only on X_i, the conditional distribution of X_{i+1} given $X_1, \ldots, X_i$ depends only on X_i.

3. Hence, the sequence $X_1, X_2, \ldots$ is a Markov chain.

Output of the Metropolis-Hastings algorithm

- The chain $X_1, X_2, \ldots$ has the property that:
  if $X_{i-1} \sim f$, then $X_i \sim f$.

- f is the equilibrium distribution or stationary of the chain.

- However, we don't start with $X_1 \sim f$ (because if we could, we wouldn't need this algorithm).

- But for large enough i, if some technical conditions are met, then each $X_i \sim f$ approximately.

Output of the Metropolis-Hastings algorithm

1. In practice, we only generate $X_1, X_2, \ldots, X_N$ for some large N.

2. Under some conditions, the empirical distribution of $X_1, X_2, \ldots, X_N$ approximates f well if N is large.

3. Hence, we can approximate the integral $I = \int h(x) f(x)\, dx$ using the approximated $X_1, X_2, \ldots, X_N$, that is

$$\frac{1}{N} \sum_{i=1}^{N} h(X_i), \quad X_1, X_2, \ldots, X_N \sim f \text{ (approximately)},$$

and $X_1, X_2, \ldots, X_N$ generated by MH.

Example: Metropolis-Hastings algorithm

- The Cauchy distribution has density

$$f(x) = \frac{1}{\pi(1 + x^2)}$$

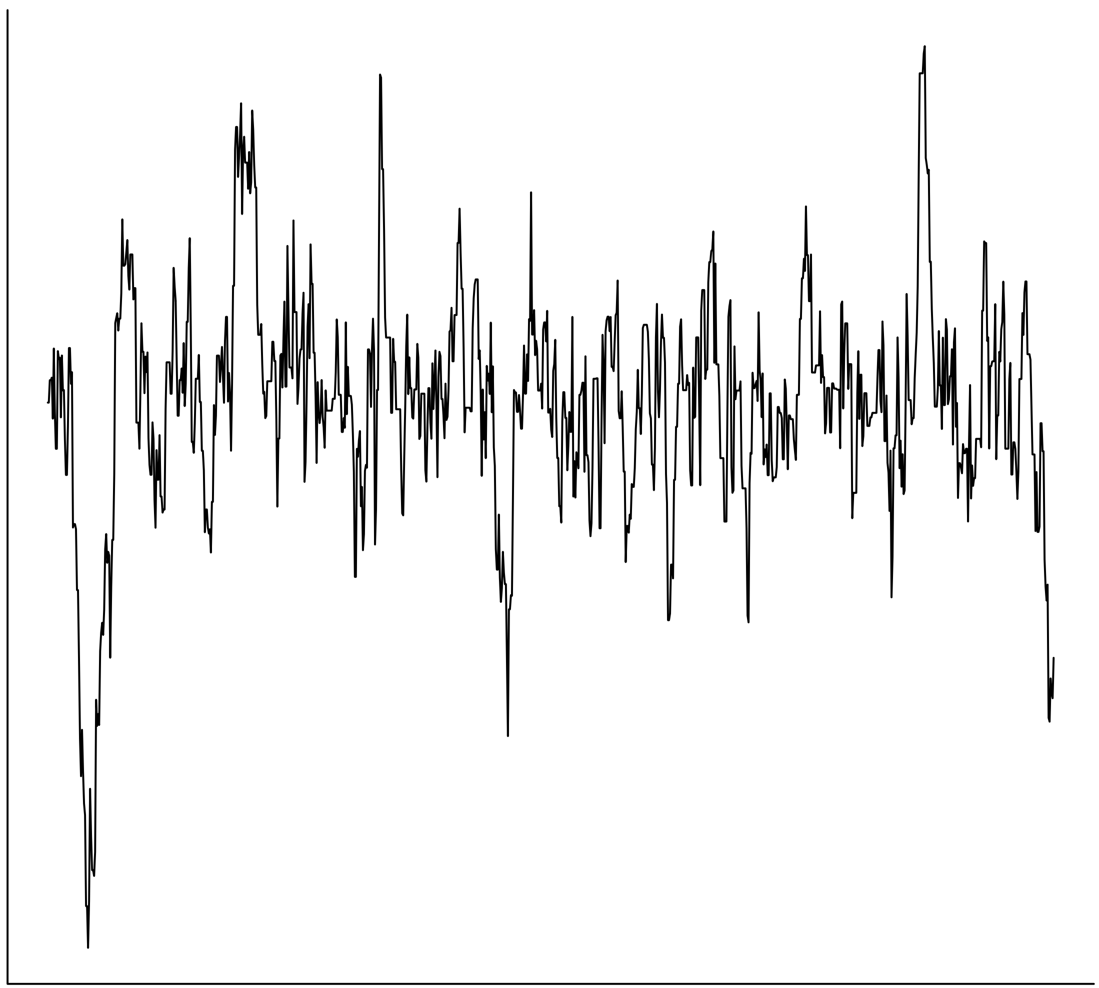
- Our goal is to simulate a Markov chain whose stationary distribution is f.
- Take $q(y|x)$ to be $N(x, b^2)$ for some $b > 0$.
- Then,

$$r(x, y) = \min\left\{\frac{1 + x^2}{1 + y^2}, 1\right\}.$$

- Let $r = r(X_i, Y)$. Generate $U \sim U(0, 1)$. If $U \leq r$, set $X_{i+1} = Y$, otherwise set $X_{i+1} = X_i$.

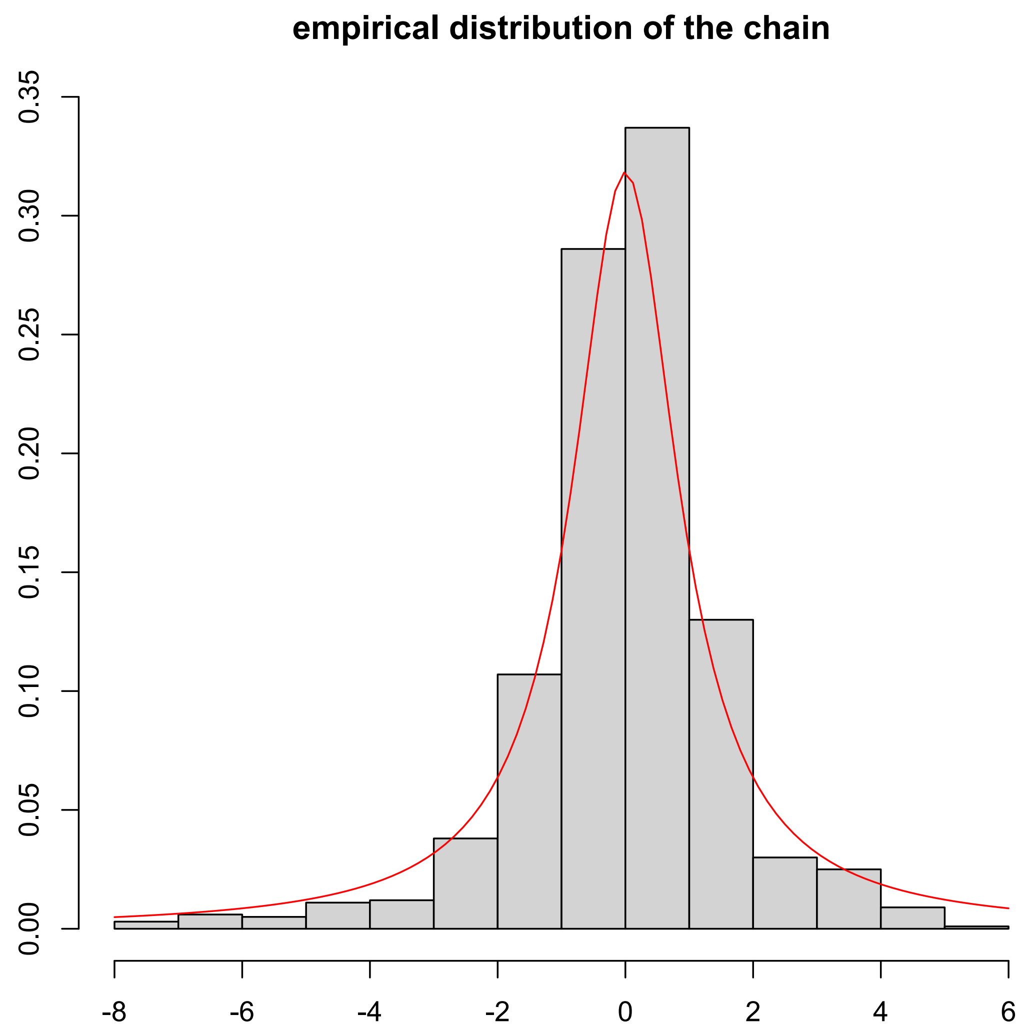
Example: Metropolis-Hastings algorithm

- Figure below shows the chains of length $N = 1000$ using $b = 1$



Example: Metropolis-Hastings algorithm

- Figure: Histogram of chains and the plot of the Cauchy density (red)
- The distribution of chain converges to the desired Cauchy distribution.



Relevance to Bayesian inference

- Let f be the posterior pdf, $p(\theta \mid y)$: this is the distribution we want to sample from.
- Let $q(\psi|\theta_i)$ be a pdf for the proposal ψ which is symmetric in ψ and θ, e.g., normal $N(\theta_i, b^2)$.
- The algorithm constructs a Markov chain $\theta_1, \theta_2, \ldots,$ where the θ_i are continuous rvs (in our applications).

Relevance to Bayesian inference

- q is called the proposal distribution: it is used to generate the next possible point in the Markov chain.
- q is often taken as a normal distribution centred on the current point

$$\psi_i \sim N(\theta_i, b^2), \text{ for some } b > 0.$$

- The normal pdf is symmetric in θ and ψ, as required by the algorithm

$$q(\psi \mid \theta) = \frac{1}{\sqrt{2\pi}b} e^{-\dfrac{(\psi - \theta)^2}{2b^2}} = \frac{1}{\sqrt{2\pi}b} e^{-\dfrac{(\theta - \psi)^2}{2b^2}} = q(\theta \mid \psi).$$

Relevance to Bayesian inference

The algorithm constructs a Markov chain $\theta_1, \theta_2, \ldots$ as follows:

- Start with arbitrary θ_1.
- For each $i > 1$, generate ψ_i from distribution $q(\psi \mid \theta_i)$.
- Let
$$r = \min\left\{1, \frac{p(\psi \mid y)}{p(\theta_i \mid y)}\right\}$$
- Set
$$\theta_{i+1} = \begin{cases} \psi & \text{with probability } r \\ \theta_i & \text{with probability } 1 - r \end{cases}$$

Relevance to Bayesian inference

- In Bayesian inference, the posterior density is

$$p(\theta \mid y) \propto p(\theta)\, p(y \mid \theta)$$

- It's difficult to find the normalizing constant

$$\int p(\theta)\, p(y \mid \theta)\; d\theta$$

- We don't need to find this, we just put $g(\theta) = p(\theta)\, p(y \mid \theta)$, use g in the algorithm (where we have f), and we will get an approximate sample from $p(\theta \mid y)$.
- The Markov chain $\theta_1, \theta_2, \ldots$ is this sample.

Metropolis algorithm for Bayesian inference

Define $g(\theta) = p(\theta)\, p(y \mid \theta)$, the non-normalized posterior density.
Generate a Markov chain $\theta_1, \theta_2, \ldots$ as follows:

- Choose some $b > 0$.
- Start with θ_1, where $g(\theta_1) > 0$.
- For each $i > 1$:
  - Generate $\psi \sim N(\theta_i, b^2)$.
  - Let
$$r = \min\left\{1, \frac{g(\psi)}{g(\theta_i)}\right\}.$$
  - Set
$$\theta_{i+1} = \begin{cases} \psi & \text{with probability } r \\ \theta_i & \text{with probability } 1 - r \end{cases}$$

Working on the log scale

- We usually do the computations using the log of the posterior density.
- The likelihood is typically a product of many terms.

$$p(y \mid \theta) = \prod_{i=1}^{n} p(y_i \mid \theta)$$

- Due to finite accuracy of computers, if we multiply these together for a large dataset, the result is inaccurate.
- So calculate

$$\log\left(p(y \mid \theta)\right) = \sum_{i=1}^{n} \log\left(p(y_i \mid \theta)\right)$$

Using the log scale

- Define $\mathcal{L}(\theta) = \log\left(p(\theta)\, p(y \mid \theta)\right) = \log\left(p(\theta)\right) + \log\left(p(y \mid \theta)\right)$, the log of the posterior density (up to a constant).
- To work on the log scale, the part of the algorithm with the acceptance probability changes.
- Define

$$\delta = \min\left(0, \mathcal{L}(\psi) - \mathcal{L}(\theta_{i-1})\right)$$

- Generate $u \sim \text{Uniform}(0, 1)$
- Set

$$\theta_i = \begin{cases} \psi & \text{if } \log(u) \leq \delta \\ \theta_{i-1} & \text{otherwise} \end{cases}$$

Normal example with known variance

- $Y_1, \ldots, Y_n$ iid from $N(\theta, \sigma^2)$ where σ^2 is known.
- $\theta \sim N(\mu, \tau^2)$ with τ^2 known,
- Apply the Metropolis-Hastings algorithm to simulate from the posterior $p(\theta|y_1, \ldots, y_n)$ after observing $Y = y$

Metropolis-Hastings algorithm for Bayesian inference

- Metropolis-Hastings algorithm generates a dependent sequence $\theta^{(1)}, \ldots,$ of θ values.
- Under mild conditions, the empirical distribution of $\theta^{(i)}$, $i = 1, 2, \ldots$ will approximate well the posterior.
- We can view $\theta^{(i)}$, $i = 1, 2, \ldots$ as a sample from the posterior $p(\theta|y)$.
- Hence, we can approximate posterior means, quantiles and other posterior quantities of interest using $\{\theta^{(1)}, \ldots, \theta^{(N)}\}$ for large N.
- However, our approximation to these quantities will depend on how well our simulated sequence actually approximates $p(\theta \mid y)$.