

# MSA101/MVE187 2020 Lecture 6.1

## MCMC Examples. Random walk. Independent proposal.

Petter Mostad

Chalmers University

September 12, 2020

# The Metropolis-Hastings algorithm

Given a probability density  $f$  that we want to simulate from. Construct a *proposal function*  $q(y | x)$  which for every  $x$  gives a probability density for a proposed new value  $y$ . The algorithm starts with a choice of an initial value  $x^{(0)}$  for  $x$ , and then simulates  $x^{(t)}$  given  $x^{(t-1)}$ . Specifically, given  $x^{(t)}$ ,

- ▶ Simulate a new value  $y$  according to  $q(y | x^{(t)})$ .
- ▶ Compute the acceptance probability

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

- ▶ Set

$$x^{(t+1)} = \begin{cases} y & \text{with probability } \rho(x^{(t)}, y) \\ x^{(t)} & \text{with probability } 1 - \rho(x^{(t)}, y) \end{cases}$$

## Toy example: Simulate from Beta(17, 13) distribution

- ▶ We first try a random-walk normal proposal function.
- ▶ Note the *trace plot*: The plot of the sequence of values  $x_0, x_1, x_2, \dots, x_N$ .
- ▶ Note the *acceptance rate*: The actual rate at which we are setting  $x^{(t+1)}$  equal to the value  $y$  proposed by the proposal function rather than setting  $x^{(t+1)} = x^{(t)}$ .

# Symmetric proposal functions

## Random walk Metropolis-Hastings

- ▶ For example, we use

$$q(y | x) = g(y - x), \text{ where } g(-x) = g(x) \text{ for all } x.$$

for some density function  $g$ : The proposal becomes symmetric around  $x$ .

- ▶ This means that  $q(y | x) = q(x | y)$  and the acceptance probability becomes

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

where  $f$  is the target density.

# Independent proposal functions

- ▶ A simple special case is when  $q(y | x)$  does not depend on  $x$ ; i.e. proposals are independently generated from  $q(y)$ .
- ▶ The generated values are however *not* independent: When the proposed value is not accepted, the new value in the chain is equal to the old.
- ▶ Note that, if the ratio  $f(x)/q(x)$  is unbounded, the chain can become stuck in such point where this ratio is too high. Then the convergence can be very bad.

## Apply to Beta(17, 13) toy example

- ▶ Try out a Uniform(0, 1) proposal function.
- ▶ Try out a Normal(0.57, 0.008) which has (approximately) same expectation and variance.
- ▶ For independent proposals, the proposal density should be as similar as possible to the target density.
- ▶ However, in thousands of dimensions, this may not be easy to achieve....

# MSA101/MVE187 2020 Lecture 6.2

## MCMC examples.

Petter Mostad

Chalmers University

September 13, 2020

# Bivariate example with random walk proposal

- ▶ As a toy example, we want to simulate from a density that is 0.5 on the squares  $[1, 2] \times [1, 2]$  and  $[3, 4] \times [3, 4]$  and zero everywhere else.
- ▶ As a first try, we use a proposal function  $(x, y) \mapsto (x + u_1, y + u_2)$  where  $u_1, u_2 \sim \text{Uniform}(-0.1, 0.1)$ .
- ▶ NOTE: The resulting Markov chain is not ergodic! So proposal function does not work.
- ▶ Second try:  $(x, y) \mapsto (x, y) + \epsilon$ , where  $\epsilon \sim \text{Normal}(0, \Sigma)$  for some covariance matrix  $\Sigma$ .
- ▶ The scaling of the size of the jumps can be very tricky to get right, to produce good convergence of the Markov chain.



## Example: A bimodal density

- ▶ A major difficulty for Metropolis Hastings may be to simulate from multimodal densities.
- ▶ As an example we explore simulating from the mixed density

$$\pi(p) = 0.5 \cdot \text{Beta}(p; 2, 20) + 0.5 \cdot \text{Beta}(p; 20, 2)$$

- ▶ A possibility in such cases is to mix proposing short jumps with occasionally proposing large jumps, of a size and direction that is tailored to the target density. (We hope to return to this point).

## Example: Braking distance of cars

We are given data which, for  $i = 1, \dots, 50$  cars (in the 1920s) lists their speed  $x_i$  (in mph) and braking distance  $y_i$ . From this, we would like to predict the braking distance at speed 21 mph.

- ▶ First, decide on the type of predictions you need answered, for example: “If a car has speed 21 mph, what is the probability the breaking distance will be above 80 feet?”
- ▶ Next step: Explore the data and the context, and decide on a reasonable model.
- ▶ We come up with the following model:

$$y_i = \theta_1 + \theta_2 \cdot x_i + \theta_3 \cdot x_i^2 + \epsilon_i$$

where  $\epsilon_i \sim \text{Normal}(0, \theta_4^2)$ .

- ▶ For simplicity, we use a flat (improper) prior on the parameters  $\theta = (\theta_1, \theta_2, \theta_3, \theta_4)$ .

## Example: Braking distance of cars, cont.

- ▶ The posterior becomes *proportional to* the likelihood:

$$f(\theta) = \prod_{i=1}^{50} \text{Normal}(y_i; \theta_1 + \theta_2 \cdot x_i + \theta_3 \cdot x_i^2, \theta_4^2)$$

- ▶ We simulate using a Random Walk Metropolis Hastings MCMC.
- ▶ Note that, together with simulation of the parameters, we also simulate the breaking distance at speed 21 at these parameters, as this is what we want to predict. We can then use these values, with Monte Carlo integration, to answer or original question.

MSA101/MVE187 2020 Lecture 6.3  
MCMC  
Checking convergence. Burn-in etc.

Petter Mostad

Chalmers University

September 13, 2020

# Advantages with Metropolis Hastings

- ▶ Great flexibility: It will (in principle) work for any (posterior) density where the density function can be computed up to a constant.
- ▶ Great flexibility in the choice of proposal function  $q(x | y)$ .
- ▶ The algorithm is quite simple and can be easily programmed in many cases.

# Some problems with Metropolis Hastings

- ▶ (Small issue): You need to make sure your proposal function makes the Markov chain ergodic.
- ▶ (Large issue): Even if the Markov chain converges, it may converge *too slowly for practical use*.
- ▶ (Large issue): Even if very many proposal functions work in theory, it may be quite difficult to find ones that lead to reasonably fast convergence.
- ▶ (Large issue): It is almost always impossible to prove results about convergence, and it is quite often difficult to ascertain how well a chain has converged.
- ▶ This means that the accuracy of results is often unknown.

# Outputs to study convergence

As we generally cannot estimate the degree of convergence, we need to at least make sure we detect clear signs of non-convergence. For example using:

- ▶ Trace plots.
- ▶ Acceptance rates.
- ▶ Varying the starting point.

# Checking convergence

- ▶ An attempt on a systematic *test* for convergence is based on the following:
  - ▶ Start  $k$  independent chains at  $k$  independent starting points.
  - ▶ Generate the Markov chains in parallel.
  - ▶ If the chains have converged, the variance between the chains should correspond to the variance within the chains.
- ▶ Formal tests have been developed using this idea.
- ▶ An (old, but useful) R package directed towards analyzing convergence from MCMC output: coda.



- ▶ Values in the last part of the generated Markov chain will be closer in distribution to the target distribution than those in the first part.
- ▶ To improve the accuracy of the Monte Carlo integration, we throw away the first part, the “burn-in”.
- ▶ The size of the burn-in can be detected from plots, or from experience in similar simulations.

# Thinning

- ▶ The Markov chain sequence is a *dependent* sequence, *not* a random sample (even if each single value has a distribution close to the target distribution).
- ▶ The amount of *autocorrelation* can be studied in plots, e.g. with the R function `acf`.
- ▶ The amount of autocorrelation can then be reduced by using, e.g., only each 10th or 50th value in the chain.
- ▶ *Only a good idea* if you need an *approximate random sample*. For Monte Carlo integration, do not do thinning.