MSA101/MVE187 2021 Lecture 6
MCMC. Random walk. Independent proposal. Convergence. Checking convergence. Burn-in. Smart proposals. Examples.

Petter Mostad

Chalmers University

September 15, 2021

Review: The Metropolis-Hastings algorithm

Given a probability density f that we want to simulate from. Construct a *proposal function* $q(y \mid x)$ 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 each $x^{(t)}$ based on $x^{(t-1)}$. Specifically, given $x^{(t)}$,

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

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

- Set

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

Overview of today

- Types of proposal functions.
- Example: Using MCMC for Bayesian linear regression.
- How to check convergence?
- Burn-in and thinning.

Random walk Metropolis-Hastings

- The proposal adds to the current value $x = x^{(t)}$ a random variable.
- Most often, the random variable has expectation zero and a symmetric distribution:

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

  for some density function g: The proposal becomes symmetric around x.
- This means that $q(y \mid x) = q(x \mid y)$ and the acceptance probability becomes

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

  where f is the target density.
- Very often g is a normal density.
- **Presentation break for example in R**.
- Note the *trace plot*: The plot of the sequence of values $x_0, x_1, x_2, \ldots x_N$.
- 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 $x^{(t+1)} = x^{(t)}$.

Independent proposal functions

- A simple special case is when $q(y \mid x)$ 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.
- **Presentation break for example in R** .
- Note that the method works well if the proposal distribution is close to the target distribution.
- Note for example 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.

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 $u_1, u_2 \sim \text{Uniform}(-0.1, 0.1)$.
- **Presentation break for a drawing**
- 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 Σ.
- **Presentation break for R computations**
- The scaling of the size of the jumps can be tricky to get right, to produce good convergence of the Markov chain.

Example: A bimodal density

- The Metropolis Hastings algorithm CAN simulate from multimodal densities. But results may be inaccurate for poor choices of the proposal distribution.
- As an example we explore simulating from the mixed density

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

- **Presentation break for R computations**
- 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 $i = 1, \ldots, 50$ 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.

- **Presentation break for plots**
- First step of analysis: Decide on Y_{data} and Y_{pred}: Done.
- 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 $\theta = (\theta_1, \theta_2, \theta_3, \theta_4)$, with $\theta_4 > 0$.

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.
- **Presentation break for R computations**
- 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.

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 by

- using trace plots.
- checking acceptance rates.
- varying the starting point $x^{(0)}$.

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 parallell.
  - 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.
- **Presentation break for illustration**
- An (old, but useful) R package directed towards analyzing convergence from MCMC output: `coda`.

Burn-in

- 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.
- **Presentation break for illustration**

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.
- **Presentation break for illustration**

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 $q(x \mid y)$.
- 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 results generally cannot be proven.