The Metropolis Algorithm

Ian Crandell

April 23rd, 2015

Outline

1. What is the Metropolis Algorithm?
2. Terminology
3. Intuition
4. Concepts
5. Example

What is the Metropolis Algorithm?

- The Metropolis Algorithm is a method for producing samples from arbitrary distributions.
- It is used primarily to sample from Bayesian posteriors which are difficult or impossible to sample from otherwise.
- Because it involves a likelihood ratio, it is not necessary to know the exact form of the desired distribution, only some distribution proportional to it.

Terminology

- The distribution from which we want to draw samples is called the *target distribution*. In this lecture we will use $f(x)$.
- The set of samples we draw will be denoted x_t for $t = 1, \ldots, T$.
- The entire sequence $\{x_t\}$ will be referred to as the chain.

Intuition

- The Metropolis algorithm is a Markov Chain Monte Carlo (MCMC) method. It works by producing a Markov Chain whose stationary distribution will eventually become that of the desired distribution.
- The chain starts in some arbitrary point in the function's domain and works by exploring nearby points. If it finds a point with a higher probability, it will move there.
- The chain can also move to positions of lower probability, but only sometimes.

Intuition

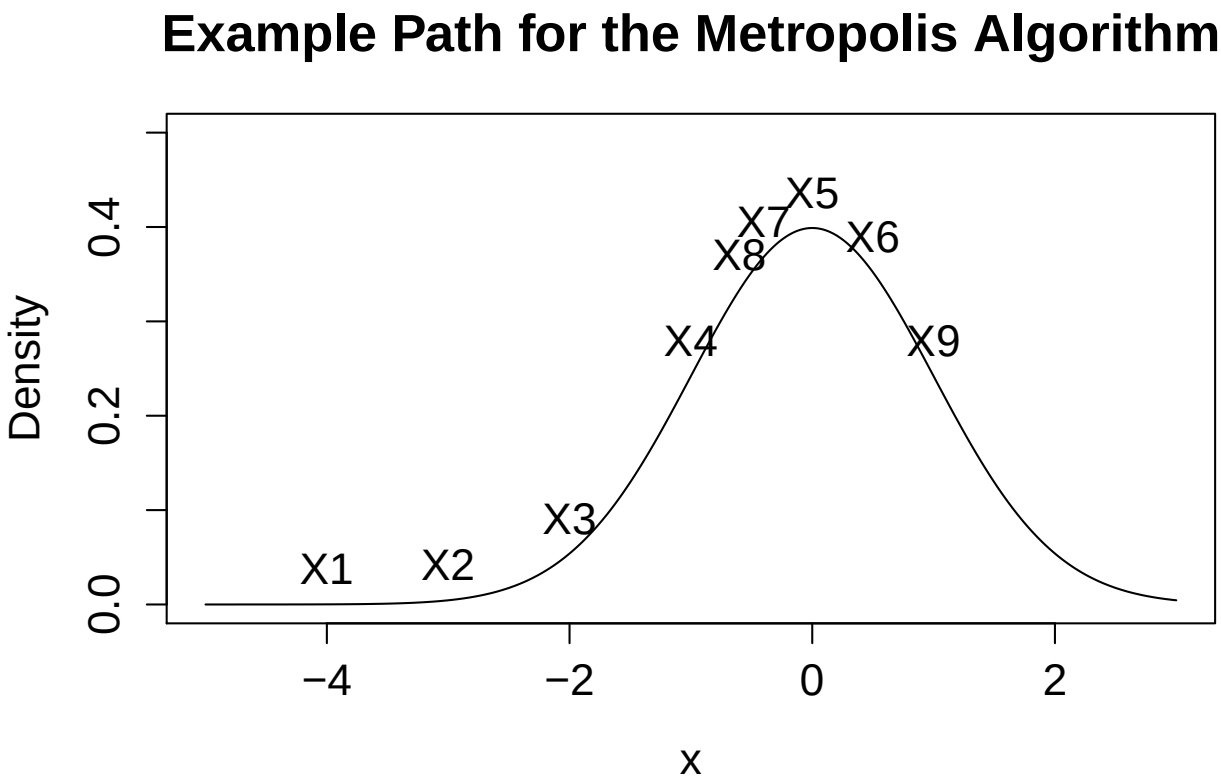

Figure: The Markov Chain can begin at any point, quite possibly far away from the region of interest. However, in a few steps the chain will find the high density region.

Concepts: Proposal Distribution

- The chain explores the space by moving around, but how does it decide where to go?
- This is determined by the *proposal distribution* $Q(x_{t+1}|x_t)$. This gives the density function of the next point in the chain. Note the dependence on the previous point alone, this makes it a Markov chain.
- A common choice is a normal distribution with variance σ^2. So,

$$Q(x_{t+1}|x_t) = N(x_t, \sigma^2).$$

We will use this in our example with $\sigma^2 = 2$.

Concepts: Acceptance Ratio

- After we propose a new point x^* from $Q(x_{t+1}|x_t)$ we have to decide if we will accept that point and move the chain along, or reject the point and sample from our current position again.
- Our action is determined by the *acceptance ratio*,

$$\alpha = \min\{1, \frac{f(x^*)}{f(x_t)}\}.$$

  Note that any proportionality constants will cancel in this ratio.
- Then, with probability α we set $x_{t+1} = x^*$ and with probability $1 - \alpha$ we set $x_{t+1} = x_t$.

Concepts: Burn in

- Since we can choose any arbitrary point, there's a chance the point we choose is nowhere near the target distribution.
- Eventually, the chain will get there, but that may take a few iterations.
- For this reason, you must discard the first several iterations of the algorithm. This is called *burn in*.
- Example: Suppose our target is a standard normal but we choose as our first point $x_0 = 1000$. It will take quite a while for us to get to the region of high probability.

Concepts: Convergence

- Once the chain has moved past burn in and is moving around the high probability region we say the chain has *converged*.
- Certain theoretical conditions guarantee convergence, but we won't go into those. We will assess convergence by examining a plot of the samples against time. This is called a *trace plot*.

Concepts: Algorithm

First, initialize x_0. Any value will do.

1. Propose x^* by sampling from $N(x_0, 2)$
2. Set
$$\alpha = \min\{1, \frac{f(x^*)}{f(x_0)}\}.$$
3. Sample u from a standard uniform distribution (uniform on $[0, 1]$).
4. If $u \leq \alpha$, set $x_1 = x^*$. If not, set $x_1 = x_0$.
5. Repeat these steps as many times as desired.

Example

Suppose we want to sample from the following distribution:

$$f(x) = x^2 e^{-2x} \qquad x > 0$$
$$f(x) = 0 \qquad x \leq 0$$

See code for details.

Example: Diagnostics

- Once we algorithm finishes running, we need to determine what the burn in period is an whether or not the chain has converged.
- This can be done graphically by examining the trace plot.
- Once the chain has burned in, the samples should have constant variance. It should resemble a *fuzzy caterpillar*.

Example: Trace Plot

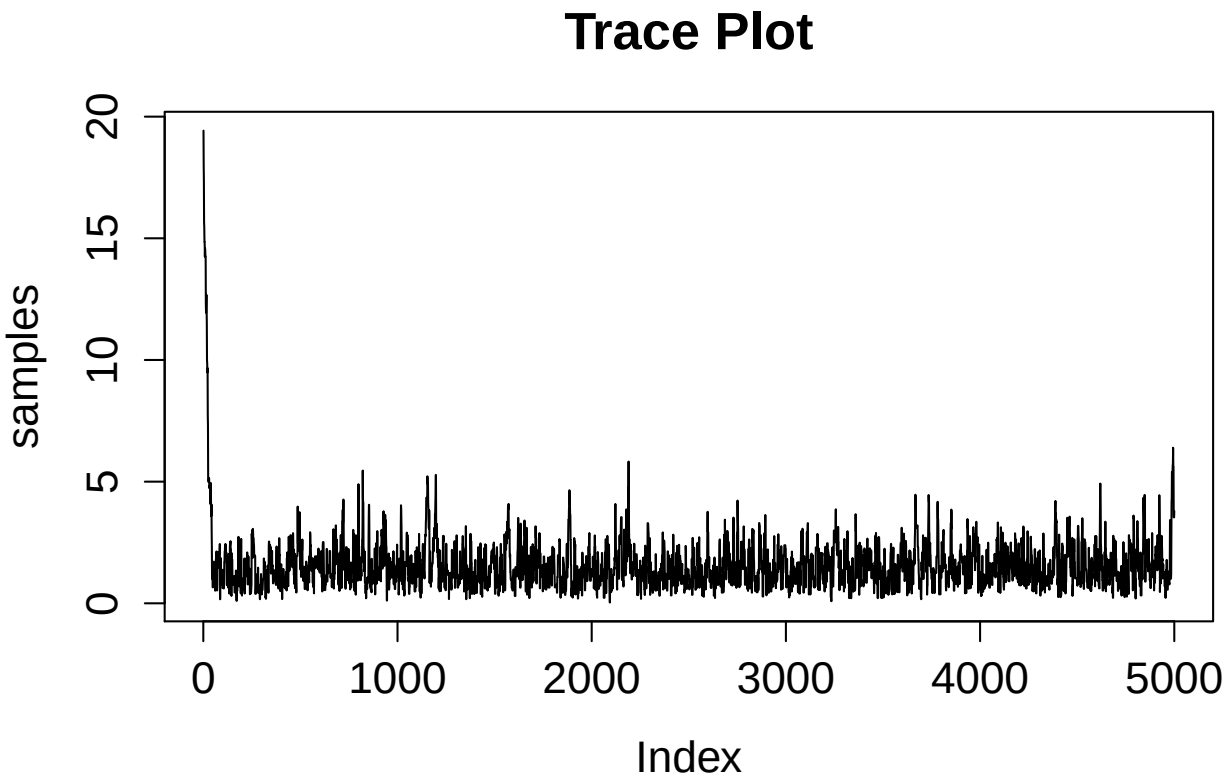

Figure: We see the trace plot starts at 20, where we set it, and quickly drops to about 1.5. We need to discard the first few samples, say 100, since those are not draws from the target distribution

Example: Trace Plot

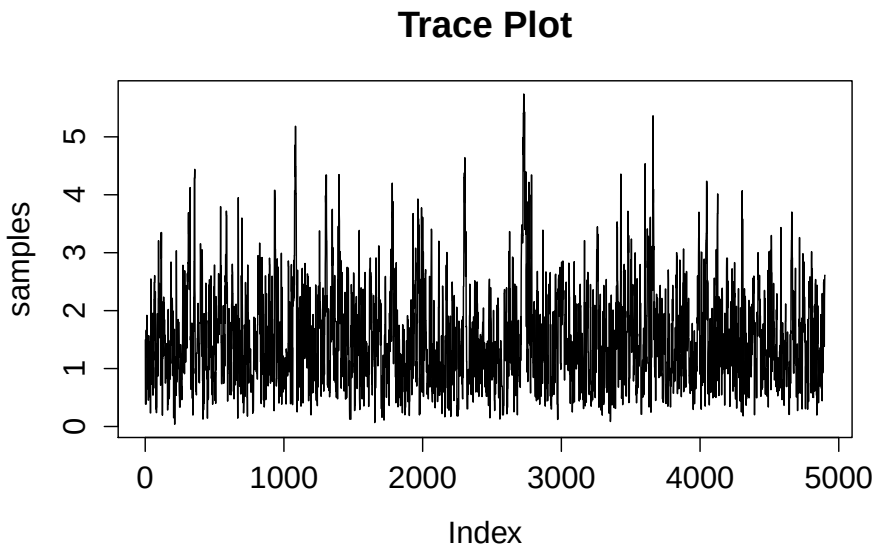

Figure: A converged trace plot

Figure: Note the resemblance.

Example Revealed

We sampled from this function:

$$f(x) = x^2 e^{-2x} \qquad x > 0$$
$$f(x) = 0 \qquad x \leq 0$$

But observe, this function is proportional to a Gamma(3, 2) density:

$$Gamma(x|\alpha = 3, \beta = 2) = \frac{2^3}{\Gamma(3)} x^2 e^{-2x} \qquad x > 0$$
$$Gamma(x|\alpha = 3, \beta = 2) = 0 \qquad x \leq 0$$