Markov Chain Monte Carlo

Metropolis-Hastings

Sachin Shanbhag

Department of Scientific Computing
Florida State University,
Tallahassee, FL 32306.

Introduction

Hastings generalized the original Metropolis algorithm to allow for non-symmetric proposal moves.

Detailed balance requires that the "net traffic" between any two points x_c and x_n in the state space be zero.

$$\begin{aligned} W(x_c \to x_n)\pi(x_c) &= W(x_n \to x_c)\pi(x_n), \\ W_{nc}\pi_c &= W_{cn}\pi_n, \\ p_{nc}a_{nc}\pi_c &= p_{cn}a_{cn}\pi_n \end{aligned} \tag{1}$$

Where transition probability $W = pa$ has been decomposed into the

- proposal p, and
- acceptance a probabilities.

Metropolis Review

In Metropolis MCMC,

- proposal moves are symmetric

$$p_{nc} = p_{cn},$$

which from detailed balance implies:

$$\begin{aligned} p_{nc}a_{nc}\pi_c &= p_{cn}a_{cn}\pi_n && (2)\\ a_{nc}\pi_c &= a_{cn}\pi_n. \\ \frac{a_{nc}}{a_{cn}} &= \frac{\pi_n}{\pi_c} && (3) \end{aligned}$$

- The Metropolis criterion suggests,

$$a_{nc} = \min\left\{1, \frac{\pi_n}{\pi_c}\right\} \tag{4}$$

Metropolis-Hastings

In Metropolis-Hastings,

- the proposal moves can be asymmetric,

$$p_{nc} \neq p_{cn}.$$

  This leads to a slight alteration in the acceptance criterion

- Metropolis-Hastings criterion

$$\frac{a_{nc}}{a_{cn}} = \frac{p_{cn}\pi_n}{p_{nc}\pi_c} = r_H \tag{5}$$

  The term r_H is often called the Hastings' ratio.

  This leads to the criterion:

$$a_{nc} = \min\left\{1, \frac{p_{cn}\pi_n}{p_{nc}\pi_c}\right\} \tag{6}$$

Example: LogNormal Distribution

Consider a *wide and asymmetric* probability distribution like a log-normal distribution:

$$\pi(x; \mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}}\, e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}}, \quad x > 0 \tag{7}$$

In this example, we set $\mu = 2.0$, and $\sigma = 1.0$.

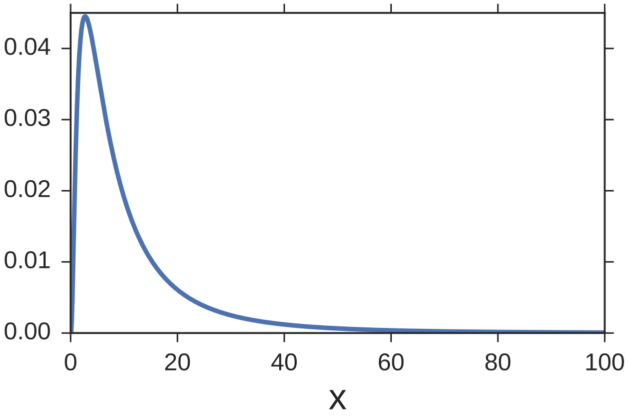
For the lognormal distribution it is known that the mean

$$\langle x \rangle = e^{\mu + (\sigma^2/2)} = e^{2.5} = 12.18,$$

and the variance is

$$\langle x^2 \rangle - \langle x \rangle^2 = (e^{\sigma^2} - 1)e^{2\mu + \sigma^2} = (e - 1)e^5 \approx 255.$$

LogNormal Distribution

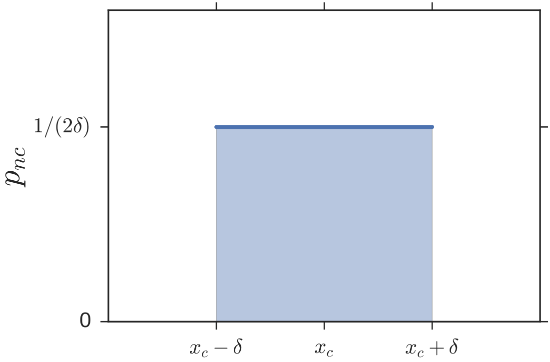


Let's write a simple Metropolis MCMC code with the proposal:

$$x_n \sim p_{nc} = U[x_c - \delta, x_c + \delta].$$

Metropolis Proposal

$$x_n \sim p_{nc} = U[x_c - \delta, x_c + \delta].$$

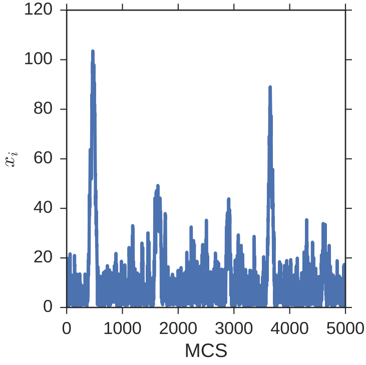


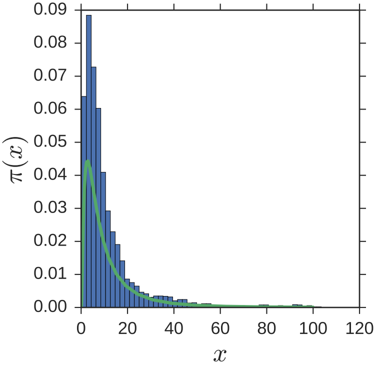
Proposal and Acceptance

```python
def metro_proposal(x, delta):
    newx = np.random.uniform(x-delta, x+delta)
    return newx

def metro_accept(xn, fc, f):
    acc  = False
    fn   = f(xn)
    ratio = f(xn)/fc

    if ratio > 1.:
        acc = True
    elif np.random.rand() < ratio:
        acc = True
    else:
        fn = fc
    return acc, fn
```

$\delta = 2.0$

acceptance ratio $= 0.91$

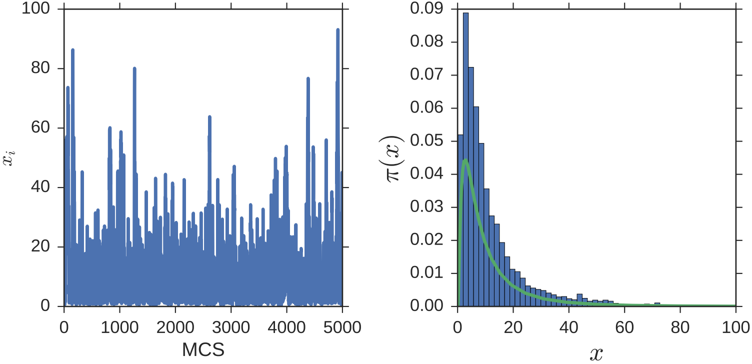




$\bar{x} = 10.59$, $s_x^2 = 82.73$

$\delta = 5.0$

acceptance ratio $= 0.78$



$\bar{x} = 10.30$, $s_x^2 = 96.8$

Comments

- As δ increases, acceptance ratio goes down
- As δ increases, the long tail is sampled better; this is evident in the larger variance
- In this example, I used a good initial state
- The computed mean $\bar{x}$ is a simple mean.
- Could use block averaging to find the error bounds on $\bar{x}$ and Gelman-Rubin metric to test convergence.
- Metropolis MCMC seems to do a reasonable job, although the error bars on the mean are somewhat large.

Metropolis-Hastings

In Metropolis MCMC the large x regime is sometimes not well-explored leading to an potential underestimation of $\langle x \rangle$.

In principle, a long enough Metropolis MCMC simulation will overcome this deficiency.

However, this seems to be a structural problem with the proposed moves. To jump from $x = 5$ to $x = 100$, we have to take a lot of small hops of size δ.

The shape of $\pi(x)$ makes this even harder, since moves that veer too far away from the peak are less likely to be accepted.

One possibility is to propose trials with larger moves.

Metropolis-Hastings

Thus, instead of *additive* random steps,

$$x_n \sim p_{nc} = U[x_c - \delta, x_c + \delta],$$

we could propose multiplicative random steps,

$$x_n = \beta x_c,$$

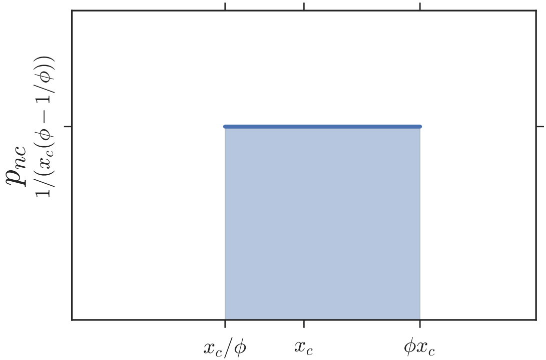
where $\beta \sim U[1/\phi, \phi]$, with $\phi = 1.5$.

Thus, $x_n = \beta x_c \sim p_{nc}$, where,

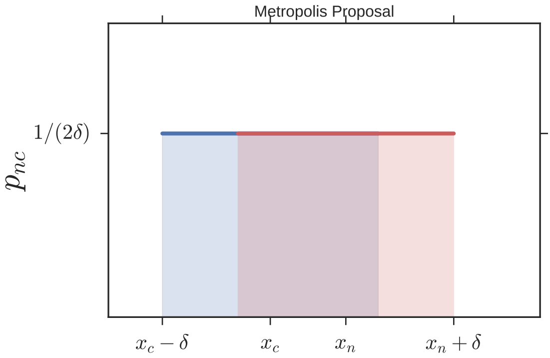
$$p_{nc} = \frac{1}{x_c\,(\phi - 1/\phi)}.$$

MH proposal

$$p_{nc} = \frac{1}{x_c\,(\phi - 1/\phi)}$$



Symmetric Proposals

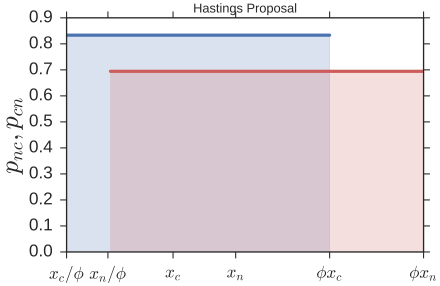


For "symmetric" Metropolis moves,

$$p_{nc} = p_{cn} = \frac{1}{2\delta}.$$

Asymmetric Proposals

$$p_{cn} = \frac{1}{x_n\left(\phi - 1/\phi\right)}, \quad \implies \frac{p_{nc}}{p_{cn}} = \frac{x_n}{x_c}.$$



p_{nc} = blue, p_{cn} = red.

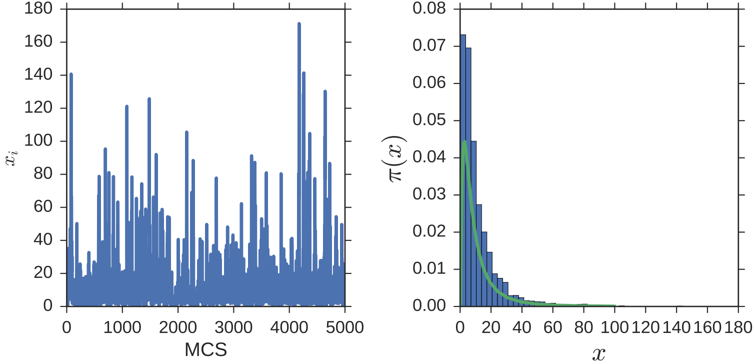
In the asymmetric (albeit uniform) proposal, the distribution becomes wider as x_c (or x_n) increases.

MH Code

```
def hastings_proposal(x, rho=1.5):
    beta = np.random.uniform(1./rho, rho)
    return beta * x

def hastings_accept(xn, xc, fc, f):
    """note needs xc in addition to fc"""
    acc = False
    fn  = f(xn)
    ratio = f(xn)/fc * xc/xn
    if ratio > 1.:
        acc = True
    elif np.random.rand() < ratio:
        acc = True
    else:
        fn = fc
    return acc, fn
```

$\rho = 1.5$

acceptance ratio $=$ 0.80



$\bar{x} =$ 12.52, $s_x^2 =$ 350.0

Summary

- When should you I Metropolis-Hastings over plain Metropolis?
  Typical use case is when the distribution $\pi(x)$ is:
  - truncated (e.g. $\pi(x < 0) = 0$)
  - skewed (asymmetric).
- Convergence diagnostics and block averaging can be applied to analyze the output of a Metropolis-Hastings sampler
- Hastings' 1970 paper generalized the Metropolis paper. Interestingly, he wrote only three peer-reviewed papers in his career, and supervised a single PhD student.