Lesson 4

Quiz, 8 questions

8/8 points (100%)

Congratulations! You passed!

Next Item

1. In which situation would we choose to use a Metropolis-Hastings (or any MCMC) sampler rather than straightforward Monte Carlo sampling?

1 / 1 points

- There is no easy way to simulate independent draws from the target distribution.

Correct
If we could, straightforward Monte Carlo sampling would be preferable.

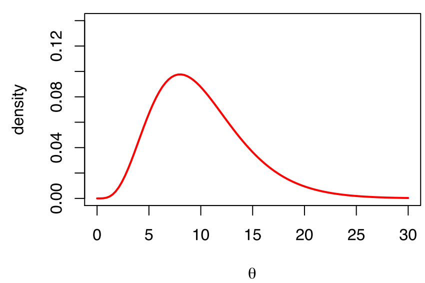
- The data (likelihood) come from a Markov chain.
- Monte Carlo estimation is easier than calculating the integral required to obtain the mean of the target distribution.
- The target distribution follows a Markov chain.

---

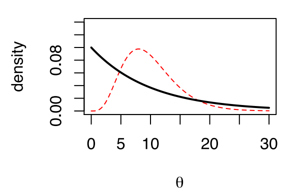
2. Which of the following candidate-generating distributions would be best for an independent Metropolis-Hastings algorithm to sample the target distribution whose PDF is shown below?

1 / 1 points

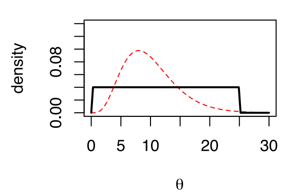
Note: In independent Metropolis-Hastings, the candidate-generating distribution q does not depend on the previous iteration of the chain.



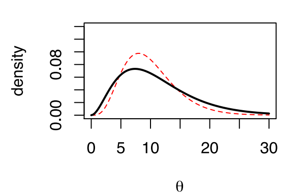
- $q = \mathrm{Exp}(0.1)$



- $q = \mathrm{Unif}(0.05, 25.0)$



- $q = \mathrm{Gamma}(3.0, 0.27)$

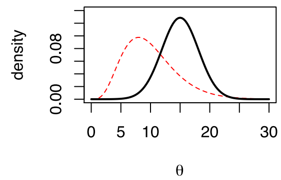


Lesson 4
Quiz, 8 questions

8/8 points (100%)

Correct
This candidate-generating distribution approximates the target distribution well, and even has slightly larger variance.

- $q = \mathrm{N}(15, 3.1^2)$



---

3. If we employed an independent Metropolis-Hastings algorithm (in which the candidate-generating distribution q does not depend on the previous iteration of the chain), what would happen if we skipped the acceptance ratio step and always accepted candidate draws?

1 / 1 points

- Each draw could be considered as a sample from the target distribution.
- The resulting sample would be a Monte Carlo simulation from q instead of from the target distribution.

Correct
Accepting all candidates just means we are simulating from the candidate-generating distribution. The acceptance step in the algorithm acts as a correction, so that the samples reflect the target distribution more than the candidate-generating distribution.

- The sampler would become more efficient because we are no longer discarding draws.
- The chain would explore the posterior distribution very slowly, requiring more samples.

---

4. If the target distribution $p(\theta) \propto g(\theta)$ is for a positive-valued random variable so that $p(\theta)$ contains the indicator function $I_{\theta>0}(\theta)$, what would happen if a random walk Metropolis sampler proposed the candidate $\theta^* = -0.3$?

1 / 1 points

- The candidate would be accepted with probability 1 because $g(\theta^*) = 0$, yielding an acceptance ratio $\alpha = \infty$.
- The candidate would be accepted with probability 1 because $g(\theta^*) = 0$, yielding an acceptance ratio $\alpha = 1$.
- The candidate would be accepted with probability 0.3 because $g(\theta^*) = |\theta^*|$, yielding an acceptance ratio $\alpha = 0.3$.
- The candidate would be rejected with probability 1 because $g(\theta^*) = 0$, yielding an acceptance ratio $\alpha = 0$.

Correct
This strategy usually works, but sometimes runs into problems. Another solution is to draw candidates for the logarithm of θ (which of course has a different target distribution that you must derive) using normal proposals.

---

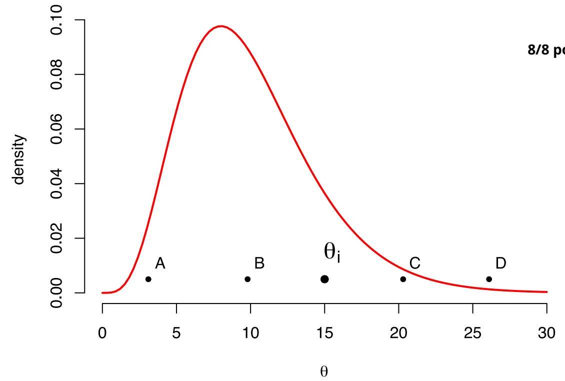
5. Suppose we use a random walk Metropolis sampler with normal proposals (centered on the current value of the chain) to sample from the target distribution whose PDF is shown below. The chain is currently at $\theta_i = 15.0$. Which of the other points, if used as a candidate θ^* for the next step, would yield the largest acceptance ratio α?

1 / 1 points

Lesson 4

Quiz, 8 questions

8/8 points (100%)



- [ ] A) $\theta^* = 3.1$
- [x] B) $\theta^* = 9.8$

Correct
B is the only point with a target density value (close to 0.09) higher than that of θ_i (close to 0.04).

Since this is a random walk Metropolis sampler with symmetric proposal distribution, the expression for calculating the acceptance ratio for iteration $i+1$ is $\alpha = g(\theta^*)/g(\theta_i)$. In this case α would be close to 2, whereas for A, C, and D, we have $\alpha < 1$. If point B were proposed, it would be accepted in this case.

- [ ] C) $\theta^* = 20.3$
- [ ] D) $\theta^* = 26.1$

---

1 / 1 points

6. Suppose you are using a random walk Metropolis sampler with normal proposals. After sampling the chain for 1000 iterations, you notice that the acceptance rate for the candidate draws is only 0.02. Which corrective action is most likely to help you approach a better acceptance rate (between 0.23 and 0.50)?

- [x] Decrease the variance of the normal proposal distribution q.

Correct
A low acceptance rate in a random walk Metropolis sampler usually indicates that the candidate-generating distribution is too wide and is proposing draws too far away from most of the target mass.

- [ ] Replace the normal proposal distribution with a uniform proposal distribution centered on the previous value and variance equal to that of the old normal proposal distribution.
- [ ] Fix the mean of the normal proposal distribution at the last accepted candidate's value. Use the new mean for all future proposals.
- [ ] Increase the variance of the normal proposal distribution q.

---

1 / 1 points

7. Suppose we use a random walk Metropolis sampler to sample from the target distribution $p(\theta) \propto g(\theta)$ and propose candidates θ^* using the $\mathrm{Unif}(\theta_{i-1} - \epsilon, \theta_{i-1} + \epsilon)$ distribution where ϵ is some positive number and θ_{i-1} is the previous iteration's value of the chain. What is the correct expression for calculating the acceptance ratio α in this scenario?

Hint: Notice that the $\mathrm{Unif}(\theta_{i-1} - \epsilon, \theta_{i-1} + \epsilon)$ distribution is centered on the previous value and is symmetric (since the PDF is flat and extends the same distance ϵ on either side).

- [ ] $\alpha = \frac{\mathrm{Unif}(\theta_{i-1} \mid \theta^* - \epsilon, \theta^* + \epsilon)}{\mathrm{Unif}(\theta^* \mid \theta_{i-1} - \epsilon, \theta_{i-1} + \epsilon)}$ where $\mathrm{Unif}(\theta \mid a, b)$ represents the PDF of a $\mathrm{Unif}(a, b)$ evaluated at θ.
- [ ] $\alpha = \frac{g(\theta_{i-1})}{g(\theta^*)}$
- [x] $\alpha = \frac{g(\theta^*)}{g(\theta_{i-1})}$

Correct
Since the proposal distribution is centered on the previous value and is symmetric, evaluations of q drop from the calculation of α.

- [ ] $\alpha = \frac{\mathrm{Unif}(\theta^* \mid \theta_{i-1} - \epsilon, \theta_{i-1} + \epsilon)}{\mathrm{Unif}(\theta_{i-1} \mid \theta^* - \epsilon, \theta^* + \epsilon)}$ where $\mathrm{Unif}(\theta \mid a, b)$ represents the PDF of a $\mathrm{Unif}(a, b)$ evaluated at θ.

Lesson 4

Quiz, 8 questions

8/8 points (100%)

1 / 1
points

8. The following code completes one iteration of an algorithm to simulate a chain whose stationary distribution is $p(\theta) \propto g(\theta)$. Which algorithm is employed here?

```
# draw candidate
  theta_cand = rnorm(n=1, mean=0.0, sd=10.0)

# evaluate log of g with the candidate
  lg_cand = lg(theta=theta_cand)

# evaluate log of g at the current value
  lg_now = lg(theta=theta_now)

# evaluate log of q at candidate
  lq_cand = dnorm(theta_cand, mean=0.0, sd=10.0, log=TRUE)

# evaluate log of q at the current value
  lq_now = dnorm(theta_now, mean=0.0, sd=10.0, log=TRUE)

# calculate the acceptance ratio
  lalpha = lg_cand + lq_now - lg_now - lq_cand
  alpha = exp(lalpha)

# draw a uniform variable which will be less than alpha with probability
    min(1, alpha)
  u = runif(1)

  if (u < alpha) { # then accept the candidate
    theta_now = theta_cand
    accpt = accpt + 1 # to keep track of acceptance
  }
```

- (selected) Independent Metropolis-Hastings (*q* does not condition on the previous value of the chain) with normal proposal

Correct
Candidates are always drawn from the same $\mathrm{N}(0, 10^2)$ distribution.

- Independent Metropolis-Hastings (*q* does not condition on the previous value of the chain) with uniform proposal
- Random walk Metropolis with normal proposal
- Random walk Metropolis with uniform proposal