Markov Chain Monte Carlo

Min-Te Sun, Ph.D.

1

Markov Chain Monte Carlo

- It is MCMC algorithms and software, along with fast computer hardware, that allow us to do Bayesian data analysis for realistic applications that would have been effectively impossible 30 years ago.
- For realistic applications that neither have appropriate conjugate prior nor can be solved by using a dense grid of points for prior distribution, MCMC methods produce accurate approximations to Bayesian posterior distributions.

2

Assumption of MCMC

1. The prior distribution is specified by a function that is easily evaluated.
   – If you specify a value for θ, then the value of *p*(θ) is easily determined.
2. The value of the likelihood function, *p*(*D*|θ), can be computed for any specified values of *D* and θ.

3

Why Monte Carlo

- This method produces an approximation of the posterior distribution, *p*(θ|*D*), in the form of a large number of θ values sampled from that distribution.
- This heap of representative θ values can be used to estimate the central tendency of the posterior, its highest density interval (HDI), etc.
- The posterior distribution is estimated by randomly generating a lot of values from it, and therefore, by analogy to the random events at games in a casino.

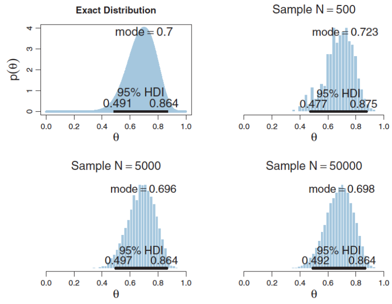
4

Approximating Distribution with a Large Sample

- Fundamental - The concept of representing a distribution by a large representative sample, such as polls and surveys.
- By randomly sampling a subset of people from a population, we estimate the underlying tendencies in the entire population.
  – The larger the sample, the better the estimation.
- The population from which we are sampling is a mathematically defined distribution, such as a posterior probability distribution.

5

An Example



6

A Simple Case of Metropolis Algorithm

- How can we sample a large number of representative values from a distribution?
  - Let’s ask a politician.
- Suppose an elected politician lives on a long chain of islands. He is constantly traveling from island to island, wanting to stay in the public eye. At the end of a grueling day of photo opportunities and fundraising, he has to decide whether to
  1. stay on the current Island,
  2. move to the adjacent island to the west, or
  3. move to the adjacent island to the east.

7

The Goal and Knowledge of Our Politician

- The goal – to visit all the islands proportionally to their relative population, so that the politician spends the most time on the most populated islands, and proportionally less time on the less populated islands.
- Assumptions:
  1. The politician has no idea what the total population of the island chain is.
  2. The politician doesn’t even know exactly how many islands there are!
  3. The politician can ask the mayor of the island they are on how many people are on the island.
  4. When the politician proposes to visit an adjacent island, he can ask the mayor of that adjacent island how many people are on that island.

8

A Working Heuristic

- First, he flips a (fair) coin to decide whether to propose the adjacent island to the east or the adjacent island to the west.
- If the proposed island has a larger population than the current island, then he definitely goes to the proposed island.
- Otherwise, if the proposed island has a smaller population than the current island, then he goes to the proposed island only probabilistically, to the extent that the proposed island has a population as big as the current island.
  - If the population of the proposed island is only half as big as the current island, the probability of going there is only 50%.

9

The Last Step of Heuristic

- Let us denote the population of the proposed island as P_{proposed}, and the population of the current island as P_{current}.
- The politician moves to the less populated island with probability $p_{\text{move}}=P_{\text{proposed}}/P_{\text{current}}$.
- The politician does this by spinning a fair spinner marked on its circumference with uniform values from zero to one.
  - If the pointed-to value is between zero and p_{move}, then he moves.

10

A Concrete Example (1/3)

- Suppose that there are seven islands in the chain, with relative populations as shown at the bottom of slide #14.
- The islands are indexed by the value θ, whereby the leftmost, western island is θ =1 and the rightmost, eastern island is θ =7. The relative populations of the islands increase linearly such that $P(\theta)=\theta$.
  - Notice that *P*( ) refers to the *relative* population of the island, not its absolute population and not its probability mass.

11

A Concrete Example (2/3)

- The middle panel of slide #14 shows one possible trajectory taken by the politician.
- Each day corresponds to one-time increment, indicated on the vertical axis.
- On the first day (t=1), the politician happens to start on the middle island in the chain, hence $\theta_{\text{current}}=4$.
- To decide where to go on the second day, he flips a coin to propose moving either one position left or one position right. In this case, the coin proposed moving right, hence θ_{proposed} =5. Because the relative population at the proposed position is greater than the relative population at the current position (i.e., $P(5) > P(4)$), the proposed move is accepted.

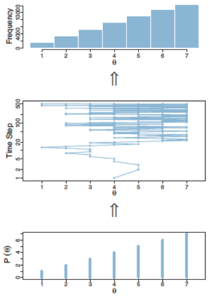
12

A Concrete Example (3/3)

- Consider the next day, when *t* =2 and θ = 5. The coin flip proposes moving to the left. The probability of accepting this proposal is $p_{\text{move}} = P(\theta_{\text{proposed}})/P(\theta_{\text{current}})$=4/5=0.80. The politician then spins a fair spinner that has a circumference marked from 0 to 1, which happens to come up with a value greater than 0.80. Therefore, the politician rejects the proposed move and stays at the current island.
- The middle panel of slide #14 shows the trajectory for the first 500 steps in this random walk across the islands.
- The upper panel of slide #14 shows a histogram of the frequencies with which each position is visited during this junket.

13

The Result of Random Walks



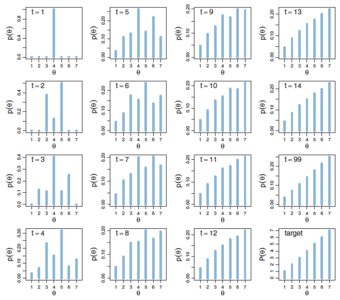
14

Probability of Being θ at Time t

- The figures in slide #16 shows the probability of being in each position as a function of time.
  - At time *t*=1, the politician starts at θ =4. This starting position is indicated in the upper-left panel of slide #16, labeled *t*=1, by the fact that there is 100% probability of being at θ =4.
  - The probability of being θ at time t gets expanded as time goes by.

15

The Results



16

Thoughts After Observation

- The graphs of slide # 16 show the *probability* that the moving politician is at each value of θ.
- But remember, at any given time step, the politician is at only one particular position, as shown in slide #14.
- When we have a long record of where the traveler has been, we can approximate the target probability at each value of θ by simply counting the relative number times that the traveler visited that value.

17

Summary of Algorithm for Moving from One Position to Another

- Let the current position be θ_{current}. We then propose to move one position right or one position left. The specific proposal is determined by flipping a coin, which can result in 50% heads (move right) or 50% tails (move left).
  - The range of possible proposed moves, and the probability of proposing each, is called the *proposal distribution*.
  - In the present algorithm, the proposal distribution is very simple: It has only two values with 50-50 probabilities.

18

Deciding Whether to Accept a Proposed Move

- The acceptance decision is based on the value of the target distribution at the proposed position, relative to the value of the target distribution at our current position.
  - Specifically, if the target distribution is greater at the proposed position than at our current position, then we definitely accept the proposed move: We always move higher if we can.
  - On the other hand, if the target position is less at the proposed position than at our current position, we accept the move probabilistically: We move to the proposed position with probability p_{move} =$P(\theta_{\text{proposed}})/P(\theta_{\text{current}})$, where $P(\theta)$ is the value of the target distribution at θ.
- The probability of moving to the proposed position:

$$p_{\text{move}} = \min\left(\frac{P(\theta_{\text{proposed}})}{P(\theta_{\text{current}})}, 1\right)$$

19

Three “Must” in Random-Walk Process

1. We must be able to generate a random value from the proposal distribution, to create θ_{proposed}.
2. We must be able to evaluate the target distribution at any proposed position, to compute $P(\theta_{\text{proposed}})/P(\theta_{\text{current}})$.
3. We must be able to generate a random value from a uniform distribution, to accept or reject the proposal according to p_{move}.

- By being able to do those three things, we are able to *indirectly* generate random samples from the target distribution.
- This technique is profoundly useful when the target distribution $P(\theta)$ is a posterior proportional to $p(D|\theta)p(\theta)$.
  - Merely evaluating $p(D|\theta)p(\theta)$ without normalizing it by $p(D)$

20

Mathematics Behind Why MCMC Works – Concept

- Consider two adjacent positions and the probabilities of moving from one to the other. We’ll see that the relative transition probabilities, between adjacent positions, exactly match the relative values of the target distribution. Extrapolate that result across all the positions, and you can see that, in the long run, each position will be visited proportionally to its target value.

21

Now the Details

- Suppose we are at position θ. The probability of moving to θ + 1, denoted $p(\theta \rightarrow \theta + 1)$, is the probability of proposing that move times the probability of accepting it if proposed, which is

  $p(\theta \rightarrow \theta + 1)$=0.5 x min ($P(\theta + 1)/P(\theta)$, 1).
- On the other hand, if we are presently at position θ + 1, the probability of moving to θ is the probability of proposing that move times the probability of accepting it if proposed, which is $p(\theta + 1 \rightarrow \theta)$=0.5 x min ($P(\theta)/P(\theta + 1)$, 1).
- The ratio of the transition probabilities is =>

$$\frac{p(\theta \rightarrow \theta{+}1)}{p(\theta{+}1 \rightarrow \theta)} = \frac{0.5 \min\left(P(\theta{+}1)/P(\theta), 1\right)}{0.5 \min\left(P(\theta)/P(\theta{+}1), 1\right)}$$

$$= \begin{cases} \frac{1}{P(\theta)/P(\theta+1)} & \text{if } P(\theta+1) > P(\theta) \\ \frac{P(\theta+1)/P(\theta)}{1} & \text{if } P(\theta+1) < P(\theta) \end{cases}$$

$$= \frac{P(\theta{+}1)}{P(\theta)}$$

22

More Details: Probability of Transitioning from θ to Other Position

$$\begin{bmatrix} \ddots & p(\theta-2 \rightarrow \theta-1) & 0 & 0 & 0 \\ \ddots & p(\theta-1 \rightarrow \theta-1) & p(\theta-1 \rightarrow \theta) & 0 & 0 \\ 0 & p(\theta \rightarrow \theta-1) & p(\theta \rightarrow \theta) & p(\theta \rightarrow \theta+1) & 0 \\ 0 & 0 & p(\theta+1 \rightarrow \theta) & p(\theta+1 \rightarrow \theta+1) & \ddots \\ 0 & 0 & 0 & p(\theta+2 \rightarrow \theta+1) & \ddots \end{bmatrix}$$

23

The Matrix Previously Equals to …

$$\begin{bmatrix} \ddots & 0.5\min\left(\frac{P(\theta-1)}{P(\theta-2)}, 1\right) & 0 & 0 & 0 \\ \ddots & \begin{array}{c} 0.5\left[1-\min\left(\frac{P(\theta-2)}{P(\theta-1)}, 1\right)\right] \\ +\, 0.5\left[1-\min\left(\frac{P(\theta)}{P(\theta-1)}, 1\right)\right] \end{array} & 0.5\min\left(\frac{P(\theta)}{P(\theta-1)}, 1\right) & 0 & 0 \\ 0 & 0.5\min\left(\frac{P(\theta-1)}{P(\theta)}, 1\right) & \begin{array}{c} 0.5\left[1-\min\left(\frac{P(\theta-1)}{P(\theta)}, 1\right)\right] \\ +\, 0.5\left[1-\min\left(\frac{P(\theta+1)}{P(\theta)}, 1\right)\right] \end{array} & 0.5\min\left(\frac{P(\theta+1)}{P(\theta)}, 1\right) & 0 \\ 0 & 0 & 0.5\min\left(\frac{P(\theta)}{P(\theta+1)}, 1\right) & \begin{array}{c} 0.5\left[1-\min\left(\frac{P(\theta)}{P(\theta+1)}, 1\right)\right] \\ +\, 0.5\left[1-\min\left(\frac{P(\theta+2)}{P(\theta+1)}, 1\right)\right] \end{array} & \ddots \\ 0 & 0 & 0 & 0.5\min\left(\frac{P(\theta+1)}{P(\theta+2)}, 1\right) & \ddots \end{bmatrix}$$

24

Why Matrix?

- Putting the transition probabilities into a matrix is that we can then use matrix multiplication to get from any current location to the probability of the next locations.
- To use the transition matrix, we put the *current* location probabilities into a row vector, *w*.
  - For example, if at the current time, we are definitely in location θ =4, then *w* has 1.0 in its θ =4 component, and zeros everywhere else.

25

More Details (Cont.)

- To determine the probability of the locations at the next time step, we simply multiply *w* by *T*.
  - The probability to be at θ position is the θ component of wT is $\sum_r w_r T_{rc} = T_{\theta c}$
  - Repeating this process will generate the graphs in Slide #16.
- *Climactic Implication: When the vector of position probabilities is the target distribution, it stays that way on the next time step!*
  - *In other words, the position probabilities are stable at the target distribution.*

26

Verification of Equilibrium

- Suppose the current position probabilities are the target probabilities, i.e., *w* =[. . . , *P*(θ −1), *P*(θ), *P*(θ +1), . . .]/*Z*, where $Z = \sum_\theta P(\theta)$ is the normalizer for the target distribution.
- Consider the θ component of *wT*. We will demonstrate that the θ component of *wT* is the same as the θ component of *w*, for any component θ.

$$\begin{aligned}\sum_r w_r T_{r\theta} &= P(\theta-1)/Z \cdot 0.5\min\left(\frac{P(\theta)}{P(\theta-1)}, 1\right) \\ &+ P(\theta)/Z \cdot \left(0.5\left[1-\min\left(\frac{P(\theta-1)}{P(\theta)}, 1\right)\right] + 0.5\left[1-\min\left(\frac{P(\theta+1)}{P(\theta)}, 1\right)\right]\right) \\ &+ P(\theta+1)/Z \cdot 0.5\min\left(\frac{P(\theta)}{P(\theta+1)}, 1\right) \qquad (7.4)\end{aligned}$$

- Break into 4 cases to simplify this equation.

27

What If We Start at Different Place?

- No matter where we start, the distribution will naturally diffuse and explore other positions.
- It’s reasonable to think that the diffusion will settled into *some* stable state and we’ve just shown that the target distribution is *a* stable state.
- This algorithm is best known as Metropolis algorithm.

28

General Metropolis Algorithm

- Previously, we considered the simple case of
  (1) discrete positions,
  (2) on one dimension, and
  (3) with moves that proposed just one position left or right.
- The general algorithm applies to
  (1) continuous values,
  (2) on any number of dimensions, and
  (3) with more general proposal distributions.

29

General Metropolis Algorithm (Cont.)

- First, we have some target distribution, *P*(θ), over a multidimensional continuous parameter space from which we would like to generate representative sample values. We must be able to compute the value of *P*(θ) for any candidate value of θ. The distribution, *P*(θ), does not have to be normalized, however. It merely needs to be nonnegative. In typical applications, *P*(θ) is the unnormalized posterior distribution on θ,
  - it is the product of the likelihood and the prior.

30

General Metropolis Algorithm (Cont.)

- Sample values from the target distribution are generated by taking a random walk through the parameter space. The walk starts at some arbitrary point, specified by the user.
  - The starting point should be someplace where *P*(θ) is nonzero.
- The random walk progresses at each time step by proposing a move to a new position in parameter space and then deciding whether or not to accept the proposed move. Proposal distributions can take on many different forms, with the goal being to use a proposal distribution that efficiently explores the regions of the parameter space where *P*(θ) has most of its mass.
  - We must use a proposal distribution for which we have a quick way to generate random values!
  - For our purposes, we will consider the generic case in which the proposal distribution is normal, centered at the current position.

31

Metropolis Algorithm Applied to Bernoulli Likelihood and Beta Prior

- We conceive of the parameter dimension as a dense chain of infinitesimal islands
- We think of the (relative) population of each infinitesimal island as its (relative) posterior probability density.
- Instead of the proposed jump being only to immediately adjacent islands, the proposed jump can be to islands farther away from the current island.
  - use the normal distribution

32

Metropolis Algorithm for Coin Flips

- We flip a coin *N* times and observe *z* heads.
- We use a Bernoulli likelihood function, $p(z,N|\theta)=\theta^z(1-\theta)^{(N-z)}$.
- We start with a prior *p*(θ) = beta(θ|*a*, *b*).
- For the proposed jump in the Metropolis algorithm, we will use a normal distribution centered at zero with standard deviation (SD) denoted as σ.
  - the proposed jump denoted as Δθ ~ normal(μ=0,σ).
- Denote the current parameter value as θ_{cur} and the proposed parameter value as $\theta_{pro} = \theta_{cur} + \Delta\theta$.

33

Details of General Metropolis Algorithm

1. Randomly generate a proposed jump, Δθ ~ normal(μ=0,σ) and denote the proposed value of the parameter as $\theta_{pro} = \theta_{cur} + \Delta\theta$.
2. Compute the probability of moving to the proposed value as below:

$$
\begin{aligned}
p_{\text{move}} &= \min\left(1, \frac{P(\theta_{\text{pro}})}{P(\theta_{\text{cur}})}\right) && \text{generic Metropolis form} \\
&= \min\left(1, \frac{p(D|\theta_{\text{pro}})p(\theta_{\text{pro}})}{p(D|\theta_{\text{cur}})p(\theta_{\text{cur}})}\right) && P \text{ is likelihood times prior} \\
&= \min\left(1, \frac{\text{Bernoulli}(z,N|\theta_{\text{pro}})\text{beta}(\theta_{\text{pro}}|a,b)}{\text{Bernoulli}(z,N|\theta_{\text{cur}})\text{beta}(\theta_{\text{cur}}|a,b)}\right) \\
& && \text{Bernoulli likelihood and beta prior} \\
&= \min\left(1, \frac{\theta_{\text{pro}}^{z}(1-\theta_{\text{pro}})^{(N-z)}\theta_{\text{pro}}^{(a-1)}(1-\theta_{\text{pro}})^{(b-1)}/B(a,b)}{\theta_{\text{cur}}^{z}(1-\theta_{\text{cur}})^{(N-z)}\theta_{\text{cur}}^{(a-1)}(1-\theta_{\text{cur}})^{(b-1)}/B(a,b)}\right) \\
& && \text{by Equations 6.2 and 6.3}
\end{aligned}
$$

34

Details of General Metropolis Algorithm (Cont.)

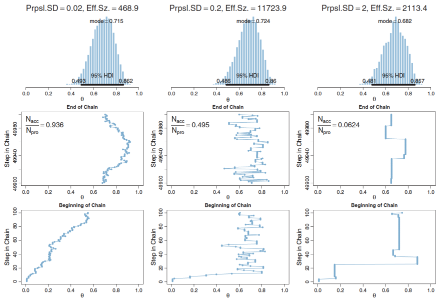
- Note that if the proposed value θ_{pro} (computed in previous slide) happens to land outside the permissible bounds of θ, the prior and/or likelihood is set to zero, hence p_{move} is zero.

3. Accept the proposed parameter value if a random value sampled froma [0, 1] uniform distribution is less than p_{move}, otherwise reject the proposed parameter value and tally the current value again.

- Repeat the above steps until it is judged that a sufficiently representative sample has been generated.

35

Examples (beta(θ|1, 1) prior, *N* = 20, *z* = 14, and θ starts at 0.01)



36

Thoughts After the Observation

- Regardless of the which proposal distribution in slide #36 is used, the Metropolis algorithm will eventually produce an accurate representation of the posterior distribution.
- The moderate proposal distribution will achieve a good approximation in fewer steps than either of the extreme proposal distributions.
- Suppose that our goal is to achieve an effective size of 10,000:
  - The proposal distribution in the middle column of slide #36 has achieved this goal.
  - For the proposal distribution in the left column, we would need to run the chain more than 20 times longer.
- Sophisticated implementations of the Metropolis algorithm have an automatic preliminary phase that adjusts the width of the proposal distribution so that the chain moves relatively efficiently.
  - A typical way to do this is to adjust the proposal distribution so that the acceptance ratio is a middling value such as 0.5.

37

Summary of Metropolis Algorithm Motivation

- Metropolis algorithm provides a high-resolution picture of the posterior distribution, even though in complex models we cannot explicitly solve the mathematical integral in Bayes' rule.
- We get a handle on the posterior distribution by generating a large sample of representative values.
  - The larger the sample, the more accurate is our approximation.
- Metropolis algorithm gets a sample of representative credible parameter values from the posterior distribution; it is not a resampling of data (there is a fixed data set).

38

Cleverness of Metropolis

- Representative parameter values can be randomly sampled from complicated posterior distributions without solving the integral in Bayes' rule, and by using only simple proposal distributions for which efficient random number generators already exist.
- All we need to decide whether to accept a proposed parameter value is the mathematical formulas for the likelihood function and prior distribution, and these formulas can be directly evaluated from their definitions.

39