Marcov Chain

E. F. Haghish

29 May 2016

Topics

- Markov chain
- Markov chain Monte Carlo
- Hastings-Metropolis sampler
- Gibbs sampler

Definition of Markov Chain

- Markov chain is a stochastic process (random process) named after **Andrey Markov**
- It represents evolution of a system of random variables over time
- Marcov chain is indexed by time $t \geq 0$
- t can be discrete or continuous
- In discrete time Markov chain, t will be a nonnagative integer
- The goal is to generate a chain by simulation

Definition of Markov Chain

- Markov chain undergoes transitions from one state to another on a **state space**
- A **state space** is the set of values which a process can take
- Markov chain is **memoryless** i.e. the probability distribution of the next state depends only on the current state and not on the sequence of events that preceded it.
- **Markov property** defines a **serial dependence** only between adjacent periods (as in a "chain"). In fact, it describes a system of chain linked events.

Definition of Markov Chain

The sequance $\{X_t \mid t \geq 0\}$ is a Markov chain if for all pairs of states $(i, j), t \geq 0$

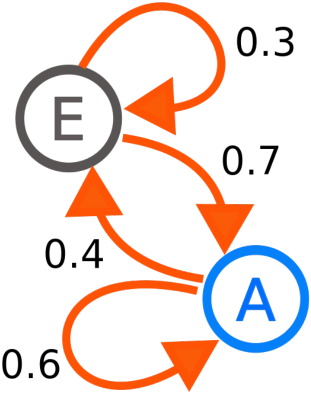
$$P(X_{t+1} = j \mid X_0 = i_0, X_1 = i_1, ..., X_{t-1} = i_{t-1}, X_t = i) =$$

$$P(X_{t+1} = j \mid X_t = i)$$

- When the state space is finite, the transition probability $P(X_{t+1} = j \mid X_t = i)$ can be represented as $\mathbb{P} = (p_{ij})$ and is called the **transition probability**
- The **transition probability** is the probability of moving from one state to another

Example of a 2-state system

- A system where it's **state space** is only **A** and **E**
- At each state, the system might retain it's state or change to the other
- At any state, the system can move the other state **within 1 step**



Properties

- Markov chain us **irreducible** if all states communicate with all other states i.e. given that the chain is in state i, there is a positive probability that the chain can enter state j in finite time, for all pairs of states (i, j).
- Since the process must be in some state after it leaves states i, the transition probabilities satisfy

$$\sum_{j=1}^{N} P_{ij} = 1$$

Properties

- For an irreducible Markov chain, π_j denotes the long-run proportion of time that the process is in state j. The π_j is called **stationary probability**

$$\sum_{j=1}^{N} \pi_j = 1$$

- Having the stationary probability of π_i, we can calculate the stationary probability π_j which is *the proportion of time in which the Markov chain* **has just entered** state j.

$$\pi_j = \sum_{i=0}^{\infty} \pi_i P_{ij}$$

Properties

- For any function h on the state space, with probability 1, it can be shown that:

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} h(X_i) = \sum_{j=1}^{N} \pi_j h(j)$$

Stochastic matrix

- And the transition probabilities can be used to form a **stochastic matrix** (also probability matrix, transition matrix, Markov matrix).
- The stichastic matrix describes the transitions of a Markov chain, where entry is a nonnegative real number representing a probability ($P_{ij} \geq 0$)

$$\mathbb{P} = \begin{bmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,j} & \cdots \\ P_{2,1} & P_{2,2} & \cdots & P_{2,j} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \\ P_{j,1} & P_{j,2} & \cdots & P_{j,j} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{bmatrix}$$

- P_{ij} is the probability of a transition from state i to state j
- The i^{th} row is the conditional probability distribution $P(X_{n+1} = j \mid X_n = i), j = 1, ..., N$ of a transition from state i to j
- Each row sum to 1 (**row stochastic** or **right stochastic** matrix)

Aperiodic Markov chain

- A state i has period k if any return to state i must occur in multiples of k time steps
- The **period** of state i is the *greatest common divisor* of the lengths of paths starting and ending at i.
- An **aperiodic** and irreducible Markoc chain is aperiodic if all states are aperiodic ($k = 1$)
- In an **aperiodic** Markov chain, there is always a positive probability for the system to get to state i in one step.
- In an irreducible Markov chain is **aperiodic** if for some $n \geq 0$ and some state j

$$P\{X_n = j \mid X_0 = j\} > 0 \quad \text{and} \quad P\{X_{n+1} = j \mid X_0 = j\} > 0$$

Then

$$\pi_j = lim_{n \to \infty} P\{X_n = j\}, \quad j = 1, ..., N$$

Time reversible Markov chain

- the Markov chain is **time reversible** if:

$$\pi_i P_{ij} = \pi_j P_{ji} \quad \textit{for } i \neq j, \quad \sum_{i=1}^{N} x_j = 1$$

- i.e. if the initial state is chosen according to the probabilities $\{\pi_j\}$, then starting at any point in time, the sequence of states **going backward in time** will also be a Markoc chain with transition probability P_{ij}.

Generating random numbers with Markov chain

- Suppose that we want to generate a random variable X with a probability mass function $P\{X = j = p_j,\ \ j = 1, ..., N\}$
- We could generate an irreducible aperiodic Markov chain with **limiting probability** p_j
- Then we could run the chain for n (a large number) steps to obtain the value of X_n
- If we wish to estimate the $E[h(X)]$ for any function, we can estimate it using $\sum_{j=1}^{N} \pi_j h(j)$
- Since the early states of the Markov chain **can be** strongly influenced by the initial state chosen, it is common in practice to disregard the first k states, *for some suitably chosen value of* k and estimate:

$$\frac{1}{n-k} \sum_{i=k+1}^{n} \pi_j h(X_i)$$

Markov Chain Monte Carlo (MCMC)

- Markov chain Monte Carlo (MCMC) methods are a general methodological framework introduced by **Metropolis** and **Hastings**.
- MCMC is an approach to **Monte Carlo integration**.

$$\overline{g} = \frac{1}{m} \sum i = 1^m g(x_i)$$

- However, MCMC approach to Monte Carlo is
  - Constructing a Markov chain with a stationary distribution
  - Run the chain for a sufficiently long time until the **chain converges** to the stationary distribution
- Therefore, the Mark ov Chain Monte Carlo methods estimate the integral using **Monte Carlo integration** abd the **Markov chain** provides a sampler that generates random observations from the target distribution.

Hastings-Metropolis Algorithm (H-M or M-H)

- The main idea is to generate a Markov chain such that the **stationary distribution is the target distribution**
- The algorithm must specify, for a given state X_n, how to generate the next state X_{t+1}
- The algorithm generates a candidate **link Y** for the chain (variable) from the proposal distribution. If the candidate link is accepted, the chain moves to state **Y** at the time $n+1$ and X_{n+1}. Otherwise, the chain stays in state X_n and time X_{n+1}.
- The choice of proposal distribution for Markov chain should be chosen so that the generated chain converges to the stationary distribution.
- The required conditions for Markov chain must be **irreducible**, **positive recurrence**, and **aperiodicity**

Example

- Suppose that we wish to calculate B

$$B = \sum_{j=1}^{m} b(j), \quad b(j) \geq 0, \quad j = 1, ..., m$$

- With a probability mass function

$$\pi(j) = \frac{b(j)}{B}, \quad j = 1, ..., m$$

- If calculating B for a large amount of m is difficult, and alternative solution would be to **simulate a sequence of random variables whose distributions converge** $\pi(j), j = 1, ..., m$ using a Markov chain that is easy to simulate and whose **limiting probabilities equal** π_j.

- let Q be an irreducible Markov transition probability matrix with $q(i,j)$ representing the row i, column j element of Q
- Then define Markov chain $\{X_n, n \geq 0\}$ as:
  - When $X_n = i$, a random variable X is generated in way that $P\{X = j\} = q(i,j), j = 1, ..., m$
  - If $X = j$, the X_{n+1} is set equal to j with probability $\alpha(i,j)$ and is set equal to i with probability $1 - \alpha(i,j)$

Summary of Hastings-Metropolis algorithm

1. Choose an irreducible Markov chain transition probability matrix Q with transition probabilities $q(i,j), i,j = 1, ..., m$ and choose some integer value k between 1 and m
2. Let $n = 0$ and $X_0 = k$
3. Generate a random variable X such that $P\{X = j\} = q(X_n, j)$ and generate a random number U from **Uniform(0,1)**
4. If $U < \alpha(i,j)$ which is calculated as:

$$U < \frac{[b(X)q(X, X_n)]}{[b(X_n)q(X_n, X)]}$$

Then Nex Step (NS) is X. Otherwise, next step would be X_n.

5. move on $n = n + 1, hspace5mmX_n = NS$
6. Go to step 3 and generate another random variable

Hastings-Metropolis Example

Let's use Metropolis-Hastings sampler for generating a sample from a Rayleigh distribution and use Chisquare distribution as a proposal distribution. The Rayleigh distribution is defined as:

$$f(x) = \frac{x}{\sigma^2} e^{-x^2/(2\sigma^2)}, \qquad x \geq 0, \sigma > 0.$$

- Let's begin with writing a function for the Rayleigh distribution

```
f <- function(x, sigma) {
        if (any(x < 0)) return (0)
        stopifnot(sigma > 0)
        return((x / sigma^2) * exp(-x^2 / (2*sigma^2)))
    }
```

- setup the simulation

```
set.seed(2016)
m <- 10000  #simulation size
sigma <- 4  #for Rayleigh distribution
x <- numeric(m)
u <- runif(m) #standard random variable for testing Alpha
k <- 0      # counter for number of rejections
```

- define X_0

```
x[1] <- rchisq(1, df=1)
```

- Run the loop from the size of the simulation, but begin from the second value because we already generated X_0
- At each transition (round of the loop), the candidate point Y is generated from $\chi^2(\nu = X_i - 1)$
- And **for each** Y, the numerator and denominator of the $\alpha(i, j)$ value are calculated as `num` and `den` in the loop

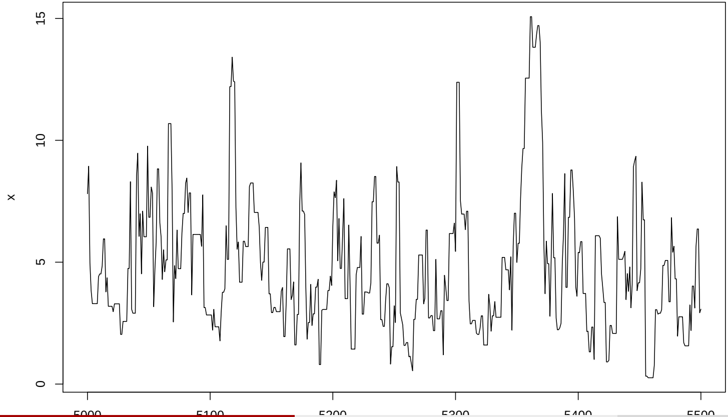
```r
for (i in 2:m) {
        xt <- x[i-1]      #DF of chisquare
        y <- rchisq(1, df = xt) #generate a candidate from chisquare

        # Calculate the alpha(i,j)
        num <- f(y, sigma) * dchisq(xt, df = y)
        den <- f(xt, sigma) * dchisq(y, df = xt)
        alpha <- num/den
        if (u[i] <= alpha) {
            x[i] <- y
        }
        else {
             x[i] <- xt
             k <- k+1
             }
}
print(k)
```

```
## [1] 4111
```

Visualizing "a part" of the simulated data

```
index <- 5000:5500
y1 <- x[index]
plot(index, y1, type="l", main="", ylab="x")
```



The Gibbs Sampler

- Proposed by Geman and Geman in 1984
- It is a special case of Hastings-Metropolis sampler
- The most popular Hastings-metropolis sampler
- Gibbs is often applied when the target is multivariate distribution
- The chain is generated by sampling from the **marginal distributions** of the target distribution
- Every generated candidate is accepted

The Gibbs Sampler

- Let $X = (X_1, ..., X_n)$ with a probability mass function $p(X)$
- Suppose we want to generate a random vector whose distribution is that of X. In other words, we want to generate a andom vector having mass function:

$$p(X) = C\, g(x)$$

- $g(X)$ is known but C (multiplicative constant) is not known

$$P\{X = x\} = P\{X_i = x \mid X_j,\ \ j \neq i\}$$

- The Gibbs sampler uses the Hastings-metropolis algorithm with states

Summary of the Gibbs algorithm

- Initialize $X(0)$ at the time $t = 0$
- For each iteration, indexed $t = 1, 2, ...$ repeat:
  1. Set $x_1 = X_1(t-1)$
  2. For each coordinate $j = 1, ..., d$
     1. Generate X_j^* from $f(X_j \mid x_{-j})$
     2. Update $x_j = X_j^*(t)$
  3. Set $X(t) = (X_1^*(t), ..., X_d^*(t))$ since every candidate is accepted
  4. Increment t