DATA 581

Modeling and Simulation II

Lecture 6: Markov Chain Models

What We Discuss Today

- **Recap on Markov Chains**
- **Probability calculation in MC models**
- **Classification of Markov Chains**
- **Law of Large Numbers for Markov Chains**
- **An Industrial Example**

Recap on Markov Chains

Discrete time stochastic process $X_1, X_2, \ldots, X_n, \ldots$ where X_n is the value at time n, is a Markov Chain such that

- **There is a state space S, such that each X_n is in S.**

- **Is memoryless (Markov Property)**

$$P(X_{n+1} = j_{n+1} | X_n = j_n, X_{n-1} = j_{n-1}, X_{n-2} = j_{n-2}, \ldots)$$

$$= P(X_{n+1} = j_{n+1} | X_n = j_n)$$

  where $j_n, j_{n-1}, j_{n-2}, \ldots$ are elements of S.

- **Probability of transition from state i to j for all i and j specified by transition probability matrix P with entites p_{ij};**

$$p_{ij} = Pr(X_n = j | X_{n-1} = i)$$

Probability Calculation

- $x^{\{n\}}$ **is called the** n**th state vector of the Markov chain and specifies the probability distribution of** X_n**.**

$$x^{\{n\}} = [P(X_n = 1)\ P(X_n = 2)\ \ldots\ P(X_n = s)]$$

where $S = \{1, 2, \ldots, s\}$ **is the sample space.**

- **The sum of the entries of** $x^{\{n\}}$ **must always be one.**

- $x^{\{0\}}$ **denotes the distribution of the *initial state*** X_0**.**
  - **For the mouse example, if the mouse starts in compartment 1,** $x^{\{0\}} = [1, 0, 0, 0]$**.**
  - **If the mouse starts in a randomly selected compartment,** $x^{\{0\}} = [.25, .25, .25, .25]$**.**

Probability Calculation

Using Chapman-Kolmogorov Equations, we can calculate the n-step probabilities.

Theorem I:

$P(X_n = j | X_0 = i)$ **is the** (i, j)**th element of the matrix** P^n**.**

We can calculate the n-th sate vector of the markov chain at time n, knowing the distribution of the initial state of the Markov chain.

Theorem II:

$$x^{\{n\}} = x^{\{0\}} P^n$$

Example

Consider a three-states markov chain with the following transition matrix.

$$P = \begin{bmatrix} 0 & 0.4 & 0.6 \\ 0.5 & 0 & 0.5 \\ 0.25 & 0 & 0.75 \end{bmatrix}$$

Example - $x^{\{0\}}$, $x^{\{1\}}$ and $x^{\{2\}}$ with random starting point

```
P <- matrix(c(0, 0.4, 0.6, 0.5, 0, 0.5, 0.25, 0, 0.75), nrow = 3,
                                        byrow = TRUE)
x0 <- rep(1/3, 3)  # random starting point
x0

## [1] 0.3333333 0.3333333 0.3333333

x1 <- x0%*%P  # distribution after 1 transition
x1

##      [,1]      [,2]      [,3]
## [1,] 0.25 0.1333333 0.6166667

x2 <- x0%*%(P%*%P) # distribution after 2 transitions
x2

##           [,1] [,2]      [,3]
## [1,] 0.2208333  0.1 0.6791667
```

Probability Distribution of X_n When n is Large

```
x0 <- rep(1/3, 3)
P2 <- P%*%P
P4 <- P2%*%P2 # 4th power of P
P8 <- P4%*%P4 # 8th power of P
P16 <- P8%*%P8 # 16th power of P
x16 <- x0%*%P16 # distribution after 16 transitions
x16
```

```
##           [,1]       [,2]      [,3]
## [1,] 0.2173913 0.08695657 0.6956522
```

```
x32 <- x16%*%P16 # distribution after 32 transitions
x32
```

```
##           [,1]       [,2]      [,3]
## [1,] 0.2173913 0.08695652 0.6956522
```

The distribution of X_n no longer seems to depend on n.

We have found called the invariant or equilibrium or stationary distribution of the Markov chain.

Question : Given a transition matrix P,

- **Does a stationary distribtution always exist?**
- **If stationary distribtution is unique, in the long run does MC converge to a specific distribution no matter what the starting state is?**

In this case, stationary distribtution is often called the limiting distribution.

Limiting Distribution

Question: **Does Every Markov Chain Have a Limiting Distribution?**

No, but Markov chains with *regular* transition matrices do.

Definition: **P is said to be a regular if there exists some positive integer n such that all entries of P^n are greater than zero.**

```
P%*%P%*%P # 3rd power of P

##          [,1]  [,2]     [,3]
## [1,] 0.162500 0.140 0.697500
## [2,] 0.268750 0.050 0.681250
## [3,] 0.228125 0.075 0.696875
```

P is a regular transition matrix.

Limiting Probability Distribution

Theorem III: If P is a regular matrix, then there exists a vector q such that

$$\lim_{n\to\infty} P^n = \begin{bmatrix} \mathbf{q} \\ \vdots \\ \mathbf{q} \end{bmatrix}$$

The vector q is the limiting probability distribution.

Example

```
P32power <- P%*%P #P^2
P32power <- P32power%*%P32power #P^4
P32power <- P32power%*%P32power #P^8
P32power <- P32power%*%P32power #P^16
P32power <- P32power%*%P32power #P^32
P32power
```

```
##           [,1]       [,2]      [,3]
## [1,] 0.2173913 0.08695652 0.6956522
## [2,] 0.2173913 0.08695652 0.6956522
## [3,] 0.2173913 0.08695652 0.6956522
```

$\mathbf{q} = [0.2173, 0.08695, 0.6956]$

Theorem IV: If P is a regular matrix, then vector q is the unique solution to the equation

$$\mathbf{q} = \mathbf{q}P$$

whose entries sum to one.

The solution of the above equation is also called the stationary distribution.

```
q <- c(.2173913, .08695652, 0.6956522)
q%*%P  # test to see if equality holds here

##             [,1]       [,2]      [,3]
## [1,] 0.2173913 0.08695652 0.6956522
```

Note: If P is not regular, it is possible for $\mathrm{q} = \mathrm{q}P$ to hold for a *stationary* vector q but where q is not a limiting distribution.

Stationary Distribution vs Limiting Distribution

P for the mouse maze example has a stationary distribution

```
P_odor <- matrix(c(0, 1, 0, 0, 0.5, 0, 0.5, 0,
                   0, 0.25, 0, 0.75,0, 0, 1, 0), nrow = 4,
                   byrow = TRUE)   # P is the transition matrix
q <- c(.1, .2, .4, .3)
q%*%P_odor  # test to see if equality holds here

##      [,1] [,2] [,3] [,4]
## [1,]  0.1  0.2  0.4  0.3
```

However, it does not have a limiting distribution, as we see that the state vectors are not converging:

```
x0 <- c(1, 0, 0, 0) #inital state
P2 <- P_odor%*%P_odor
P4 <- P2%*%P2
P8 <- P4%*%P4
P16 <- P8%*%P8
P32<- P16 %*%P16
x32<- x0%*%P16 # probability distribution at time step 32
x32

##           [,1] [,2]      [,3] [,4]
## [1,] 0.2003129    0 0.7996871    0
```

```r
x33 <- x32%*%P_odor
x33  # probability distribution at time step 33

##      [,1]      [,2] [,3]      [,4]
## [1,]    0 0.4002346    0 0.5997654
```

Classification of Markov Chain States

The following discussion leads to a simple way of determining whether a transition matrix is regular or not.

Definition

State i *leads* to state j if there exists $n \geq 1$ such that $P_{ij}^{(n)} > 0$.

Proposition

The *leads to* relation is transitive. That is, if i leads to j and j leads to k, then i leads to k.

Definition

States i and j are said to *communicate* if i leads to j and j leads to i.

Definition

A *class* of states is defined as a subset of S in which any two members communicate.

Example:

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 0.5 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$\{1, 2\}$ is one class and $\{3\}$ is another class.

Definition. **S is said to be *irreducible* if it is a class. In other words, if all states in S communicate, S is said to be irreducible.**

Classification of Markov Chain States

Definition.

For any $i \in S$, the *period* of state i is defined to be the greatest common divisor of the set

$$\{n > 0 : P_{ii}^{(n)} > 0\}$$

Proposition. **If i and j communicate, then the periods of i and j are the same.**

Definition. **If the state space of a Markov chain is irreducible, then the period of the Markov chain is defined to be the common period of each state.**

In the Mouse Maze exampe, period is 2, since if the chain starts in State 1 it can never return to State 1 in an odd number of transitions, and all states communicate.

Definition.

If the period of a Markov chain is 1, the Markov chain is said to be aperiodic.

P_{33}**(MC provided in example page 6) is aperiodic.**

Theorem V

An aperiodic irreducible Markov chain with a finite state space must have a regular transition matrix. $\rightsquigarrow$ **has a stationary/limiting distribution.**

P_{32} **is regular.** P **for the mouse maze example is not regular.**

Calculation of Stationary Vector

Question: How to calculate the stationary distribution?

The Stationary Vector q solves

$$\mathbf{q} = \mathbf{q}P$$

which can be re-written as

$$(P^\top - I)\pi = 0$$

where $\pi = \mathbf{q}^\top$.

```
P33<- matrix(c(0, 0.4, 0.6,
               0.5, 0, 0.5,
               0.25, 0, 0.75), nrow = 3,
               byrow = TRUE)
A <- t(P33) - diag(rep(1,3)) # P^T - I
solve(A, rep(0,3))    # solve A pi = 0

## Error in solve.default(A, rep(0, 3)):  Lapack routine dgesv:
system is exactly singular:  U[3,3] = 0
```

trouble! $P^\top - I$ is singular, so there are too many solutions. We need more equations.

Calculation of Stationary Vector

Since the Stationary distribution is a (discrete) probability distribution, its elements must sum to 1:

$$\pi_1 + \pi_2 + \pi_3 = 1, \qquad \textbf{so we include this equation:}$$

```
A <- rbind(A, rep(1,3))
RHS <- c(rep(0,3), 1)
qr.solve(A, RHS)   # no longer a square system

## [1] 0.21739130 0.08695652 0.69565217
```

What is Markov Chain Monte Carlo?

- **The most popular method for sampling from high-dimensional distributions is Markov chain Monte Carlo or MCMC**

- **Markov Chain Monte Carlo methods are a family of algorithms that uses Markov Chains to perform Monte Carlo estimate.**

  - ***Monte Carlo method: randomly sampling a probability distribution and approximating a desired quantity.***

  - ***Assumption: samples are independent***

MCMC allows us

- **Perform statistical inference for probability distributions where independent samples from the distribution cannot be drawn, or cannot be drawn easily.**
- **Random sampling of high-dimensional probability distributions that honors the probabilistic dependence between samples by constructing a Markov Chain that comprise the Monte Carlo sample.**

MCMC

Samples are drawn from the probability distribution by constructing a Markov Chain, where the next sample that is drawn from the probability distribution is dependent upon the last sample that was drawn.

The idea is that the chain will settle on (find equilibrium) on the desired quantity we are inferring.

The goal of the remaining slides is to explore the particular MCMC simulator, called the Metropolis-Hastings algorithm, in the context of Markov chain models and to provide some simple illustrative examples of its use.

In order to understand the Metropolis-Hastings algorithm, the concept of time-reversibility of a Markov chain is useful.

- ***In what follows, the state space can be infinite or finite.***

A Markov chain with transition matrix P and stationary distribution π such that $P(X_n = i) = \pi_i$ is said to be time-reversible if the Markov property holds for the chain when it is time reversed.

The joint probabilities of the form $P(X_{n+1} = j; X_n = i)$ are the same as $P(X_n = j; X_{n+1} = i)$.

Reversible Markov Chains

Theorem: A Markov Chain with transition matrix P is reversible if there exists a vector q such that

$$q_j P_{ji} = q_i P_{ij}.$$

- Observe what happens when we multiply such a vector q by P.
- The ith component of the vector qP is

$$\sum_{j=1}^{\infty} q_j P_{ji}$$

- Since the Markov chain is reversible, this is;

$$q_i \sum_{j=1}^{\infty} P_{ij} = q_i$$

- Therefore

$$\mathbf{q}P = \mathbf{q}.$$

- In other words, q is the stationary distribution for P.

Random Walk

An example of a Markov chain is a random walk in one dimension;

- **where the possible moves are 1, -1, chosen with equal probability,**
- **and the next point on the number line in the walk is only dependent upon the current position and the randomly chosen move.**

Example: Symmetric Random Walk with Reflecting Barrier. Suppose $S = \{0, \pm 1, \pm 2, \ldots, \pm k\}$ for some $k > 2$.

$$X_n = X_{n-1} + 2B_n - 1$$

- **where B_n is Bernoulli with parameter $p = .5$, independent of X_{n-1}.**
- **When $X_{n-1} = \pm k$, $X_n = k - 1$ (or $1 - k$).**
- **For $|j| < k$,**

$$P_{j,j+1} = P_{j,j-1} = 0.5.$$

- **otherwise,**

$$P_{k,k-1} = 1 = P_{-k,-k+1}.$$

Example: Symmetric Random Walk

Let's see if this Markov chain is time-reversible. Can we find a vector q that satisfies the condition in the theorem?

$$q_k P_{k,k-1} ? = q_{k-1} P_{k-1,k} ? \quad \textbf{Then}$$

$$q_k = q_{k-1} \times 0.5$$

and similarly,

$$q_{-k} = q_{1-k} \times 0.5.$$

For $|j| < k$**,**

$$q_j P_{j,j+1} = q_{j+1} P_{j+1,j} \quad \textbf{or}$$

$$0.5 q_j = 0.5 q_{j+1}.$$

Therefore, all q**'s other than the** q_k **and** q_{-k} **are equal, and have twice the value of** q_k **and** q_{-k}**. So, this Markov chain is time-reversible.**

Example: Symmetric Random Walk

Once we have shown that the Markov chain is time-reversible, we can use the associated q vector to find the stationary distribution.

$$q_3 = x, q_2 = 2x, q_1 = 2x, q_0 = 2x, q_{-1} = 2x, q_{-2} = 2x, q_{-3} = x$$

Also we know that sum of all entries of stationary distribution is 1. Therefore,

$$x + 2x + 2x + 2x + 2x + 2x + x = 1 \rightsquigarrow x = \frac{1}{12}$$

So the vector q is;

$$q_3 = q_{-3} = \frac{1}{12}$$

$$q_2 = q_1 = q_0 = q_{-1} = q_{-2} = \frac{1}{6}.$$

A Simulation Check on the Calculation

We can verify this by simulation as well.

```
#transition matrix for symmetric random walk k = 3
P <- matrix(c(0,1,0,0,0,0,0,
.5,0,.5,0,0,0,0,0,.5,0,.5,0,0,0,
0,0,.5,0,.5,0,0,0,0,0,0.5,0,.5,0,
0,0,0,0,.5,0,.5,0, 0, 0, 0, 0, 1, 0), nrow=7, byrow = TRUE)
Ntransitions <- 100000 # number of moves
location <- numeric(Ntransitions)#initializing
current.state <- 1 # initial stock
for (t in 1:Ntransitions) {
   current.state <- sample(1:7,
        size = 1, prob = P[current.state, ])
   location[t] <- current.state
}
pi <- table(location)/Ntransitions
pi

## location
##       1       2       3       4       5       6       7
## 0.08196 0.16490 0.16702 0.16906 0.16881 0.16604 0.08221
```

Other Time-Reversible Markov Chains

Suppose $\{\pi_i, i = 0, \pm 1, \pm 2, \ldots\}$ **is a set of positive real numbers with** $\sum_{i=-\infty}^{\infty} \pi_i = 1$**. (This is a probability distribution on the integers.)**

Set

$$P_{i,j} = \tfrac{1}{6} \min\left(\frac{\pi_j}{\pi_i}, 1\right), \text{ for } j = i-2, i-1, i+1, i+2$$

$$P_{i,j} = 0 \text{ for } |j - i| > 2,$$

$P_{i,i}$ **is set to ensure that the row sums of** P **are 1.**

To verify that the Markov chain is reversible, show that

$$\pi_i P_{i,i+2} = \pi_{i+2} P_{i+2,i}$$

and so on.*

*This equation follows by noting that the left hand side is $\frac{1}{6} \min(\pi_j, \pi_i)$ and the right hand side is $\frac{1}{6} \min(\pi_i, \pi_j)$.

Simulating from the Infinite State Markov Chain

Note: P **is an example of an infinite-state time-reversible Markov chain. Note that the stationary distribution vector has infinite length and has** i**th entry** π_i**.**

Question: How can we simulate from a infitite state markov chain?

We can simulate from a infinite state Markov chain by simply keeping track of the current state after each transition and updating the probability distribution for the next transition accordingly.

Simulating from the Infinite State Markov Chain

Example: **Suppose we define the probability distribution such that;**

- $\pi_i = k/(i+1)^4$ **for** $i > 0$
- $\pi_i = 0$ **for all** $i < 0$.
- k **is a constant that ensures that** $\sum_{i=1}^{\infty} \pi_i = 1$.

Note that we can simulate from this Markov chain even without knowing k.

```
pi.fun <- function(i) {
    out <- 0
    if (i > 0) out <- 1/(i+1)^4
    out
}
```

Simulating from the Infinite State Markov Chain

```r
N <- 20000
X <- numeric(N)
current.state <- 50  # initialize the Markov chain
for (n in 1:N) {
      i <- current.state
      P <- c(min(pi.fun(i-2)/pi.fun(i), 1),
      min(pi.fun(i-1)/pi.fun(i), 1),
      min(pi.fun(i+1)/pi.fun(i), 1),
      min(pi.fun(i+2)/pi.fun(i), 1))/6
      P0 <- 1 - sum(P)
      P <- c(P[1:2], P0, P[3:4])
      transition <- sample(seq(-2,2,1), size = 1, prob = P)
      current.state <- current.state + transition
      X[n] <- current.state
}
```

Simulating from the Infinite State Markov Chain

The relative frequency distribution can be computed using the `table()` function after removing a sufficient number of the early observations.

```
n <-1000
observedDist <- table(X[-c(1:n)])
observedDist/(N-n)

##
##             1             2             3             4             5
## 0.7716842105 0.1507894737 0.0445263158 0.0175789474 0.0084210526 0.004
##             7             8             9            10
## 0.0009473684 0.0007368421 0.0001578947 0.0002105263
```

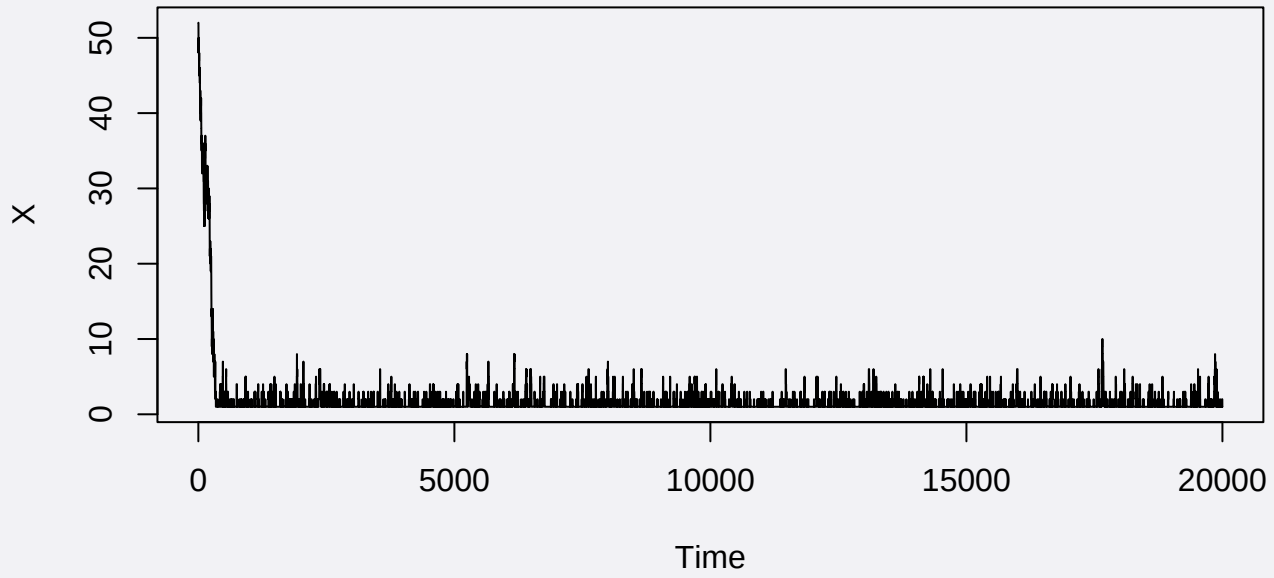
This distribution serves as an estimate of the distribution π.

Burn-In

Question: Why omit the first 1000 observations?

MCMC algorithms are sensitive to their starting point, and often require a warm-up phase or burn-in phase to move in towards a fruitful part of the search space, after which prior samples can be discarded and useful samples can be collected.

```
ts.plot(X)
```



Estimating k

$$\pi_2 = k/(2+1)^4 = k/81$$

so an estimate of k can be obtained by multiplying the observed probability of a 2 by 81:

```
k <- observedDist[2]/19000*81
k
```

```
##         2
## 12.21395
```

This procedure is one version of MCMC, developed by Metropolis and Hastings. The procedure is as follows.

1. **Given a distribution π, known up to a proportionality constant (k), find a time-reversible Markov chain with π as the stationary distribution vector.**
2. **Simulate from that Markov chain.**
3. **After simulating for a long enough period (burn-in), the observed values of the Markov chain follow the limiting stationary distribution, i.e. π.**

One way to do MCMC in R is with the `metrop()` function in *mcmc* package. The syntax for its use is

```
metrop(obj, initial, nbatch, blen = 1, nspac = 1,
    scale = 1, outfun, debug = FALSE, ...)
```

The main arguments are:

- `obj`**: an R function which evaluates the unnormalized posterior distribution or the result of a previous call to this function.**
- `initial`**: the initial state of the Markov chain.**
- `scale`**: controls the proposal step size in the random walk used for the Markov chain.**

Example:

Suppose N is Poisson distributed with mean 20, and given N, X is binomially distributed with parameters N and $p = 0.5$. N is not observed, but suppose $X = 5$.*

Use MCMC to simulate the distribution of N, given X.

- **The Poisson distribution for N is the *prior* distribution.**
- **The distribution of N, given $X = 5$, is called the *posterior* distribution.**

*This problem is somewhat artificial, but it can be viewed as arising from a coin-flipping experiment where the number of heads, X, has been recorded, but the number of coin flips was forgotten or not recorded in any case.

MCMC Application - Bayesian Statistics

```
posterior.fun <- function(i, x) {
    out <- 0
    if (i >= x) {
    out <- dpois(i, lambda = 20)*
            dbinom(x, size = i, prob = .5)
            }
    out
}
```

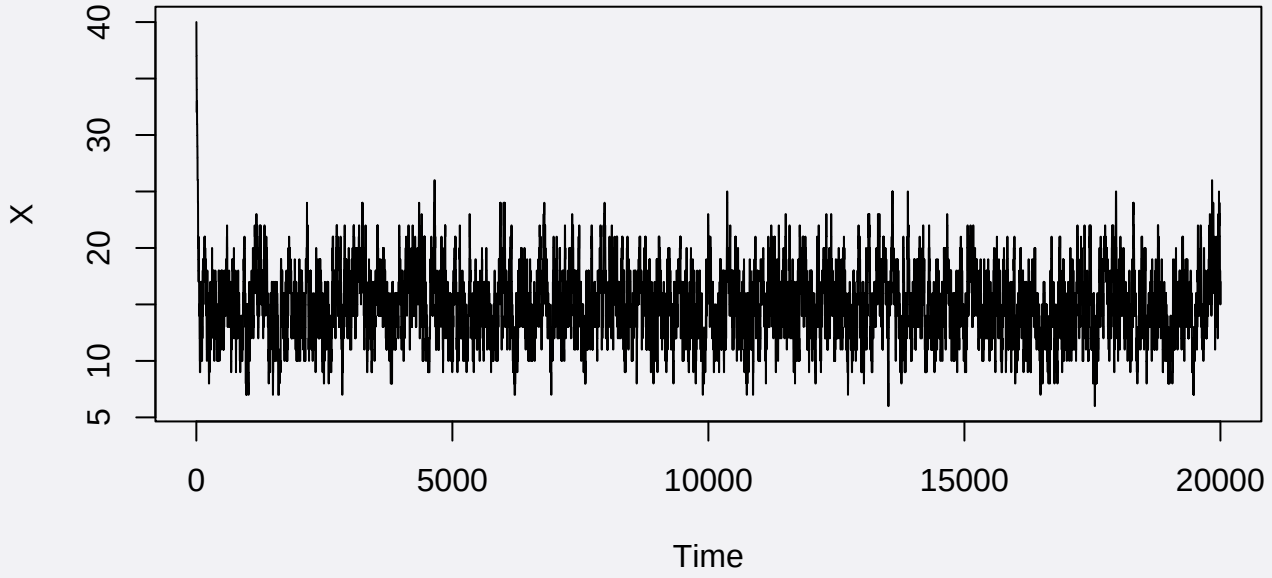
```
pi.fun <- function(i) {
    posterior.fun(i, x=5)
}
```

Simulating the Markov Chain

```
Ntransitions <- 20000
X <- numeric(Ntransitions)
current.state <- 40  # initialize the Markov chain
for (n in 1:Ntransitions) {
      i <- current.state
      P <- c(min(pi.fun(i-2)/pi.fun(i), 1),
                  min(pi.fun(i-1)/pi.fun(i), 1),
                  min(pi.fun(i+1)/pi.fun(i), 1),
                  min(pi.fun(i+2)/pi.fun(i), 1))/6
      P0 <- 1 - sum(P)
      P <- c(P[1:2], P0, P[3:4])
      transition <- sample(seq(-2,2,1), size = 1, prob = P)
      current.state <- current.state + transition
      X[n] <- current.state
}
observedDist <- table(X[-c(1:1000)])
```

Plotting the Trace

```
ts.plot(X)
```



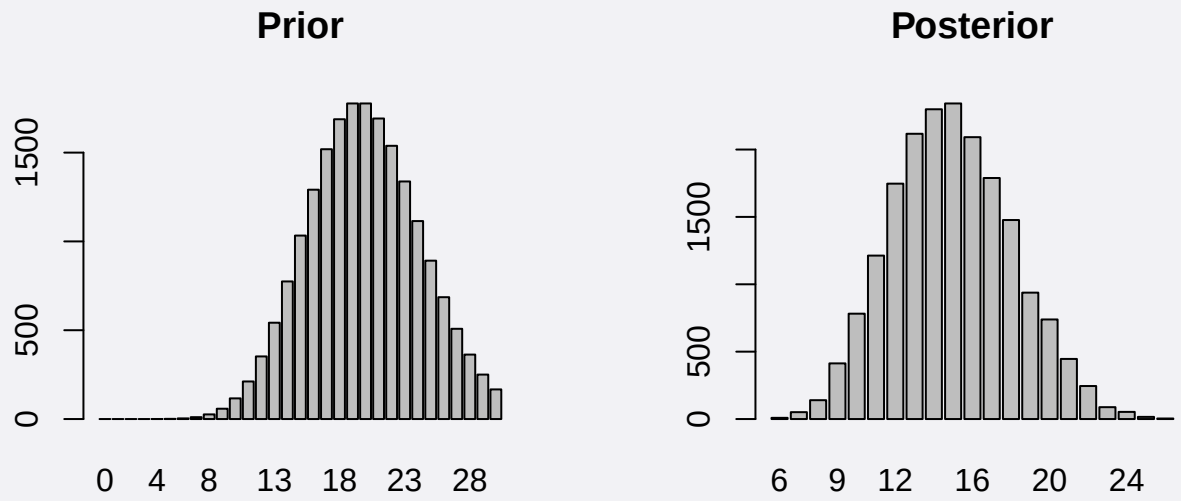
Posterior Distribution of N

```
options(width=50)
observedDist
```

```
##
##    6    7    8    9   10   11   12   13   14   15
##    9   51  140  413  782 1213 1747 2117 2299 2342
##   16   17   18   19   20   21   22   23   24   25
## 2092 1789 1477  938  739  446  245   88   53   16
##   26
##    4
```

Posterior Distribution of N

```
par(mfrow=c(1,2))
theoryDist <- 20000*dpois(0:30, lambda = 20)
names(theoryDist) <- 0:30
barplot(theoryDist, main = "Prior")
barplot(observedDist, main = "Posterior")
```



This is how the *data* $X = 5$ influences our *belief* (initially, Poisson(20)) about the distribution of the unknown value N.

What if our Prior Belief was Different?

e.g. $\lambda = 4$**:**

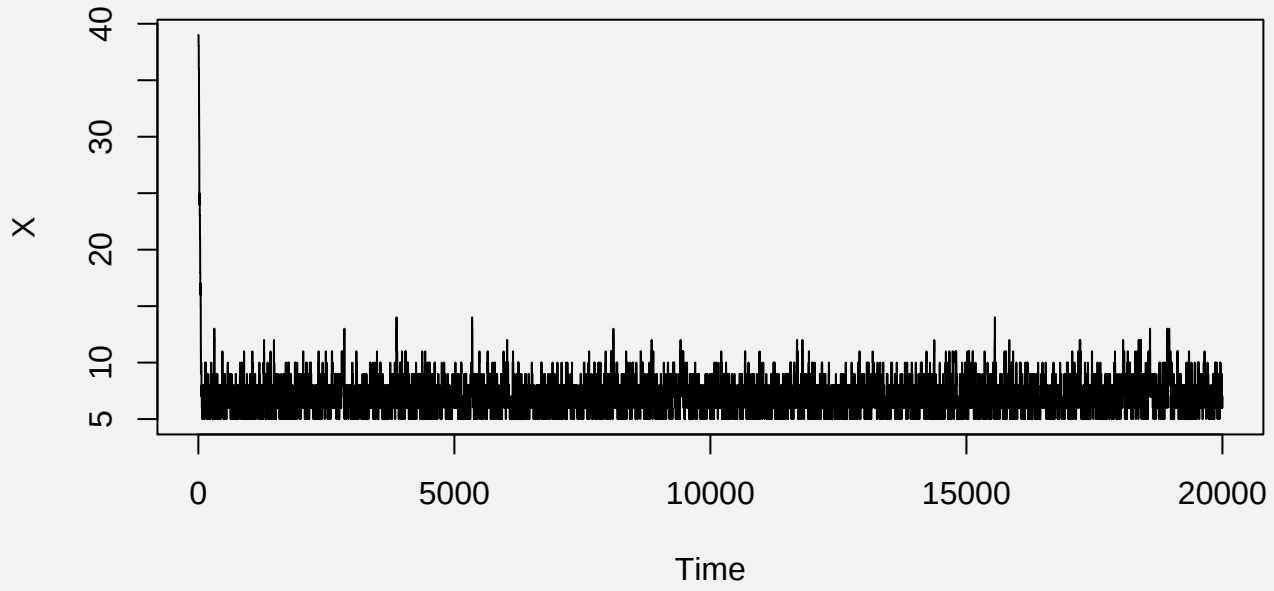
```
posterior.fun <- function(i, x) {
    out <- 0
    if (i >= x) out <- dpois(i, lambda = 4)*
            dbinom(x, size = i, prob = .5)
    out
}
```

Simulating the Markov Chain

```
Ntransitions <- 20000
X <- numeric(Ntransitions)
current.state <- 40  # initialize the Markov chain
for (n in 1:Ntransitions) {
      i <- current.state
      P <- c(min(pi.fun(i-2)/pi.fun(i), 1),
                  min(pi.fun(i-1)/pi.fun(i), 1),
                  min(pi.fun(i+1)/pi.fun(i), 1),
                  min(pi.fun(i+2)/pi.fun(i), 1))/6
      P0 <- 1 - sum(P)
      P <- c(P[1:2], P0, P[3:4])
      transition <- sample(seq(-2,2,1), size = 1, prob = P)
      current.state <- current.state + transition
      X[n] <- current.state
}
observedDist <- table(X[-c(1:1000)])
```

Plotting the Trace

```
ts.plot(X)
```



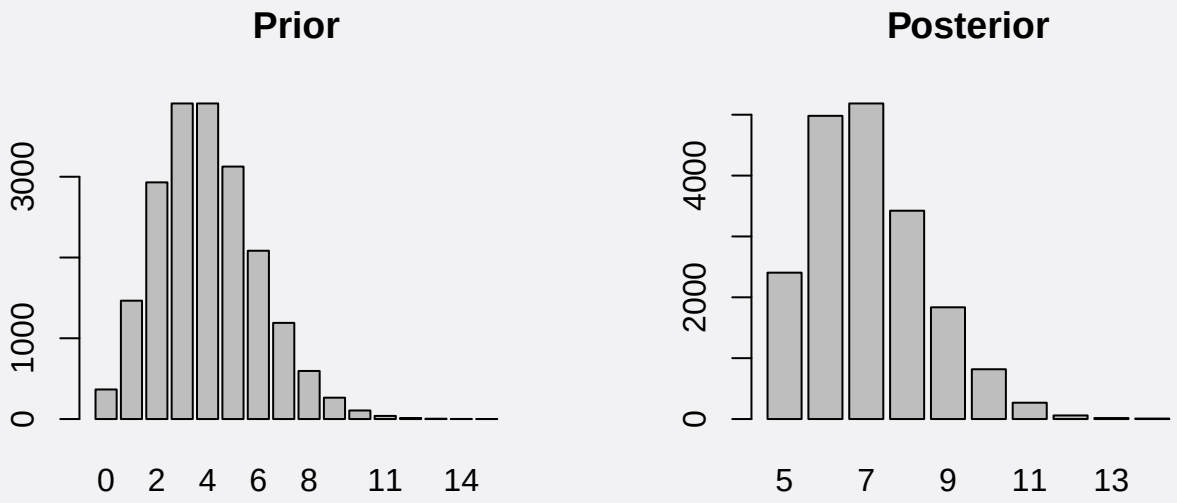
Posterior Distribution of N

```
options(width=50)
observedDist
```

```
##
##    5    6    7    8    9   10   11   12   13   14
## 2405 4982 5185 3423 1836  818  268   59   16    8
```

Posterior Distribution of N

```
par(mfrow=c(1,2))
theoryDist <- 20000*dpois(0:15, lambda = 4)
names(theoryDist) <- 0:15
barplot(theoryDist, main = "Prior")
barplot(observedDist, main = "Posterior")
```



This is how the *data* $X = 5$ influences our *belief* (initially, Poisson(4)) about the distribution of the unknown value N.