Markov Chains

Jonathan Navarrete

June 24, 2017

Introduction

Consider a (discrete) random process X_t that can take k-*states*. The process starts at one of these states and proceeds forward with some probability $p_{i,j}$, the probability of transitioning from state i to state j. The probabilities $p_{i,j}$ are called *transition probabilities*.

Now consider a chain with two states, *[0, 1]*. This chain takes $X_1, X_2, ...$ steps where at each step it transitions between the two states with some probability $p_{i,j}$. Therefore, when the chain is at state $i = 0$, the chain can transition to state $j = 1$ with some probability, or remain at $i = 0$.

The random variables $X_{t-1}, X_{t-2}, ...$ depend on the previous it state, the random variables are not independent. The random variables, albeit dependent, are only dependent on the previous iteration such that

$$P(X_t = x_t | X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, ...) = P(X_t = 1 | X_{t-1} = x_{t-1})$$

This conditional dependence on one-previous-step is a key property of **Markov Chains**, known as the **Markov Property**.

From the conditional probabilities, we can form a **transition matrix**

$$\mathbf{P}_{2\times2} = \begin{array}{c c} & \begin{array}{cc} 0 & 1 \end{array} \\ \begin{array}{c} 0 \\ 1 \end{array} & \begin{pmatrix} p_{i,i} & p_{i,j} \\ p_{j,j} & p_{j,j} \end{pmatrix} \end{array} = \begin{array}{c c} & \begin{array}{cc} 0 & 1 \end{array} \\ \begin{array}{c} 0 \\ 1 \end{array} & \begin{pmatrix} p_{0,0} & p_{0,1} \\ p_{1,0} & p_{1,1} \end{pmatrix} \end{array}$$

where the i^{th} row gives the conditional distribution of $X_t | X_{t-1}$, and each row's probabilities sum up to 1. If, for example, at from state $i = 1$, we have a probability of 0.4 to transition to $j = 1$, and from state $j = 1$ we have a probability of 0.55 of transitioning to state $i = 0$ for $n = 1, 2, ...$ iterations, then our matrix would be

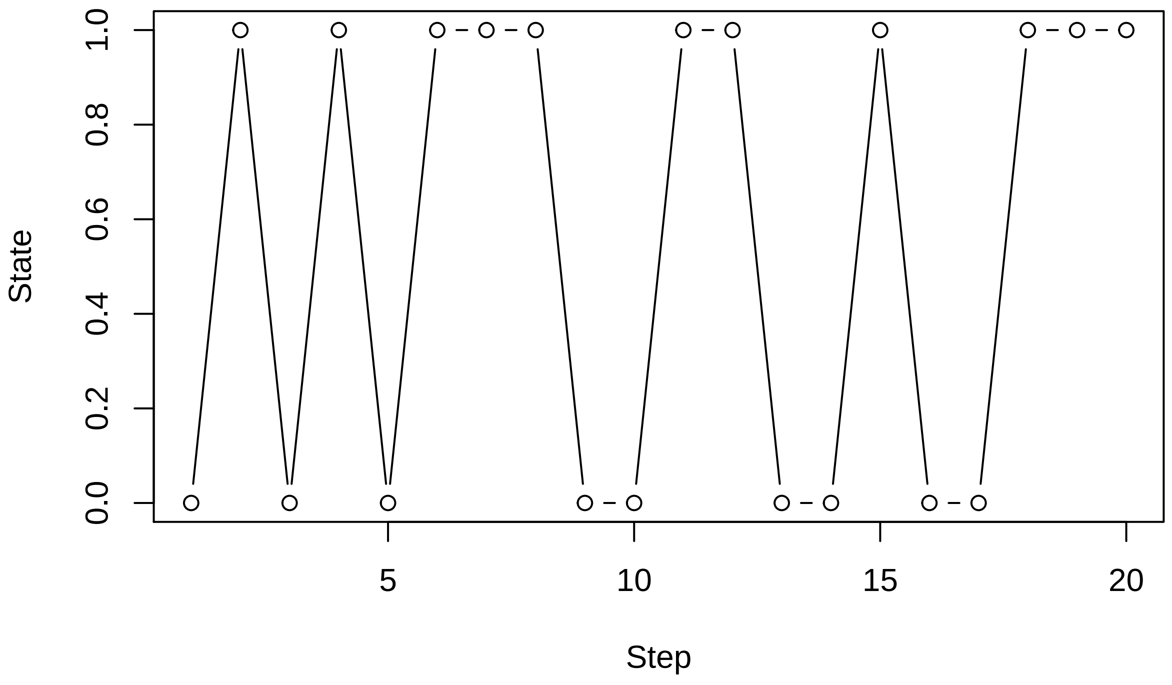
$$\mathbf{P}_{2\times2} = \begin{pmatrix} p_{0,0} & p_{0,1} \\ p_{1,0} & p_{1,1} \end{pmatrix} = \begin{pmatrix} 0.6 & 0.4 \\ 0.55 & 0.45 \end{pmatrix}$$

Then as a simulation exercise, we would have

```
## code goes here
m = 2000;
n = 1:m;

x = numeric(m);

x[1] = 0

alpha = 0.4;
beta = 0.55
# Simulation
for (i in 2:m){
    if (x[i-1]==0){
        x[i] = rbinom(1, 1, alpha)
    } else {
        x[i] = rbinom(1, 1, 1 - beta)
    }
}


y = cumsum(x)/n

# Running fractions of Long eruption# Results

y[m]
```

```
## [1] 0.435
```

```
# Fraction of Long eruptions among m. Same as: mean(a = sum(x[1:(m-1)]==0 & x[2:m]==1); a # No. of cycl

#m/a

# Average cycle length
plot(x[1:20], type="b", xlab="Step", ylab="State")
```



$$P_{m,n} = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m,1} & p_{m,2} & \cdots & p_{m,n} \end{pmatrix}$$

Here is another matrix

$$\begin{array}{c|ccc} & H & Y & D \\ \hline H & .8 & .2 & .0 \\ Y & .3 & .4 & .3 \\ D & .2 & .1 & .7 \end{array}$$

Another Example: 3-state chain

For this following example we'll be utilizing the `markovchain` library available from **CRAN**. Revolution Analytics (currently an extension of Microsoft) has a nice blog post and tutorial on the use of this package. See *Getting Started with Markov Chains*

Consider a basketball team currently participating in a tournament. As they proceed through their games, assume the team has three states: *win, lose, tie*. Say, at the start of the season, they win their first game, $X_0 = win$. From this point, they probabilities of winning, losing or drawing their next game. For our favorite team, say these probabilities are $P(X_t = win|X_{t-1} = win) = 0.35$, $P(X_t = lose|X_{t-1} = win) = 0.45$, $P(X_t = tie|X_{t-1} = win) = 0.2$. However, if they were to lose their first game, the probabilities change to $P(X_t = win|X_{t-1} = lose) = 0.60$, $P(X_t = lose|X_{t-1} = lose) = 0.30$, $P(X_t = tie|X_{t-1} = lose) = 0.1$. And if they were to draw their first game, then $P(X_t = win|X_{t-1} = tie) = 0.55$, $P(X_t = lose|X_{t-1} = tie) = 0.40$, $P(X_t = tie|X_{t-1} = tie) = 0.05$. Thus, these probabilities construct a transition matrix **P**.

$$P_{3\times3} = \begin{pmatrix} p_{win,win} & p_{win,lose} & p_{win,draw} \\ p_{lose,win} & p_{lose,lose} & p_{lose,draw} \\ p_{draw,win} & p_{draw,lose} & p_{draw,draw} \end{pmatrix} = \begin{pmatrix} 0.35 & 0.45 & 0.20 \\ 0.60 & 0.30 & 0.10 \\ 0.55 & 0.40 & 0.05 \end{pmatrix}$$

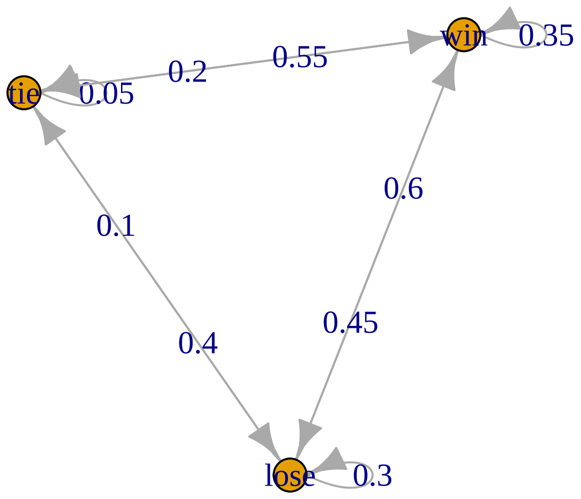
```r
library(markovchain)
```

```
## Package:  markovchain
## Version:  0.6.9.3
## Date:     2017-05-08
## BugReport: http://github.com/spedygiorgio/markovchain/issues
```

```r
## set up transition matrix
P <- matrix(data = c(0.35, 0.45, 0.20,
                     0.60, 0.30, 0.10,
                     0.55, 0.40, 0.05), byrow = TRUE,
            nrow = 3, ncol = 3, dimnames = list(c("win", "lose", "tie"), c("win", "lose", "tie")))

P <- new( "markovchain", transitionMatrix = P)

print(P)
```

```
##       win lose  tie
## win  0.35 0.45 0.20
## lose 0.60 0.30 0.10
## tie  0.55 0.40 0.05
```

```r
plot(P)
```



As in the 2-state example, our Markov chain's transition to X_t only depends on X_{t-1} and no further. So for the probability outcome of the current game, we would only need to look back at the most recent past game and not the entire season.

For a more thorough review of Markov chains, see *Chapter 11: Markov Chains*.

Ergodic Properties

In the set up of MCMC algorithms, Markov chains are constructed from a *transitional kernel K*, a conditional probability density such that $X_{t+1} \sim K(X_t, X_{t+1})$. A typical example is provided by the random walk process, formally defined as follows. As we continue with notes and definitions, note that this will be through the perspective of a discrete-time stochastic processes, $(X_t)_{t\in\mathbb{N}}$

Definition 6.1 Random Walk

A sequence of random variables (X_t) is a *random walk* if it satisfies

$$X_{t+1} = X_t + \epsilon_t,$$

where ϵ_t is generated independently of $X_t, X_{t-1}, \ldots$. If the distribution of the ϵ_t is symetric about zero, the sequence is called a symmetric random walk.

There are many examples of random walks, and random walks play a key role in many MCMC algorithms, particularly thosed based on the Metropolis-Hastings algorithm.

The chains encountered in MCMC settings enjoy a very strong stability proerty, namely a *stationary probability distribution* exists by construction; that is, a distribution f such that if $X_t \sim f$, then $X_{t+1} \sim f$, if the kernel K allows for free moves all over the state space. This freedom to move all around the state space is called *irreducibility*, and is essential for MCMC algorithms. Irreducibility states that for $n \in \mathbb{N}$ such that $P(X_n \in A|X_0) > 0$ for every A such that $\pi(A) > 0$.

The irreducibility property also ensures that most of the chains involved in MCMC algorithms are *recurrent*, that the number of times a chain visits an arbitrary set A in the state space is infinity; a chain is allowed to revisit any part of the state space always.

This latter point is quite important in the context of MCMC algorithms. Since most algorithms are started from some arbitrary point x_0, we are in effect starting the algorithm from a set of measure zero. Thus, ensuring that the chain converges for almost every starting point is not enough, and we need Harris reccurence (see Harris theorem)to guarantee convergence from every starting point.

For the most part, the Markov chains encountered in Markov chain Monte Carlo (MCMC) settings enjoy a very stsrong stability property. Indeed, a stationary probability distribution exists by construction for those chains; that is, there exists a probability distribution f such that if $X_t \sim f$, then $X_{t+1} \sim f$. Therefore, formally, the kernel and stationary distribution satisfy the equation

$$\int_{\mathcal{X}} K(x,y)f(x)dx = f(y)$$

The existence of a stationary distribution (or stationarity) imposes a preliminary constraint on K called *irreducibility* in the theory of Markov chains, in which the kernel K allows free movement all over the state-space. Thus, no matter the starting value of X_0, the sequence (X_t) has a positive probability of eventually reaching any region of the state-space. A sufficient condition is that $K(x,\cdot) > 0$ everywhere.

The *stationary distribution* is also a *limiting distribution* in the sense that the limiting distribution of X_{t+1} is f under the total variation norm, notwithstanding the initial value of X_0. Stronger forms of convergence are also encountered in MCMC settings, like geometric and *uniform* convergences. In a simulation setup, a most interesting consequence of this convergence property is that the empirical average

$$\frac{1}{T}\sum_{n=1}^{T} h(X_t) \to E_f[h(X)]$$

convergence to the expectation $E[h(X)]$ almost surely. When the chain is *reversible* (that is, when the transition kernel is symmetric), a Central Limit Theorem also holds for this average. In summary, the Law of Large Numbers that lies at the basis of previous *i.i.d.* Monte Carlo methods can also be applied in MCMC settings. This is often referred to the **Ergodic Theorem**.

This said, we can now begin delving into MCMC algorithms such as the Metropolis-Hastings algirithm or the Gibbs sampler, which by the ergodic theorem are almost always theretically convergent.