Introduction to Markov Chains for MCMC

Jonathan Navarrete

Introduction

A (discrete) **Markov chain** is a random (stochastic) process $X_t : t \in T$ that can take *k*-states over time $t = 1, 2, \ldots$. The process starts at one of these states (i) and proceeds forward to another state (j) 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*.

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

Two-State Markov Chain

Consider an ordered stochastic process with *k* = 2 states (or outcomes), *[0, 1]*. This chain takes $X_1, X_2, \ldots$ steps where at each step it transitions between the two states with some probability $p_{i,j}$. When the chain is at state $i = 0$, the chain can transition to state $j = 1$ with some probability, $p_{0,1} = \alpha$, or remain at $j = 0$ with probability $p_{0,0} = 1 - \alpha$.

Likewise, if the chain is currently at state $i = 1$, the chain can transition to $j = 0$ with probability $p_{1,0} = \beta$ or remain at $j = 1$ with probability $p_{1,1} = 1 - \beta$.

Two-State Markov Chain

In a Markov chain, the random variables X_t depend on the previous it state X_{t-1}. Thus, the random variables are not independent; autocorrelation exists. The random variables, albeit dependent, are only dependent on the previous iteration such that the transition probabilities only depend on the most recent state

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

This conditional dependence on one-previous-step is known as the **Markov Property**.

Two-State Markov Chain

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

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

where the i^{th} row gives the conditional distribution of $X_t|X_{t-1}$, and each row's probabilities sum up to 1.

Example: 2-State Markov Chain

If from state $i = 0$, we have a probability of 0.4 to transition to $j = 1$, $P(X_t = 1|X_{t-1} = 1) = 0.4$. And from state $i = 1$ we have a probability of 0.55 of transitioning to state $j = 0$, $P(X_t = 1|X_{t-1} = 0) = 0.55$. Then for $n = 1, 2, \ldots$ iterations, then our matrix would be

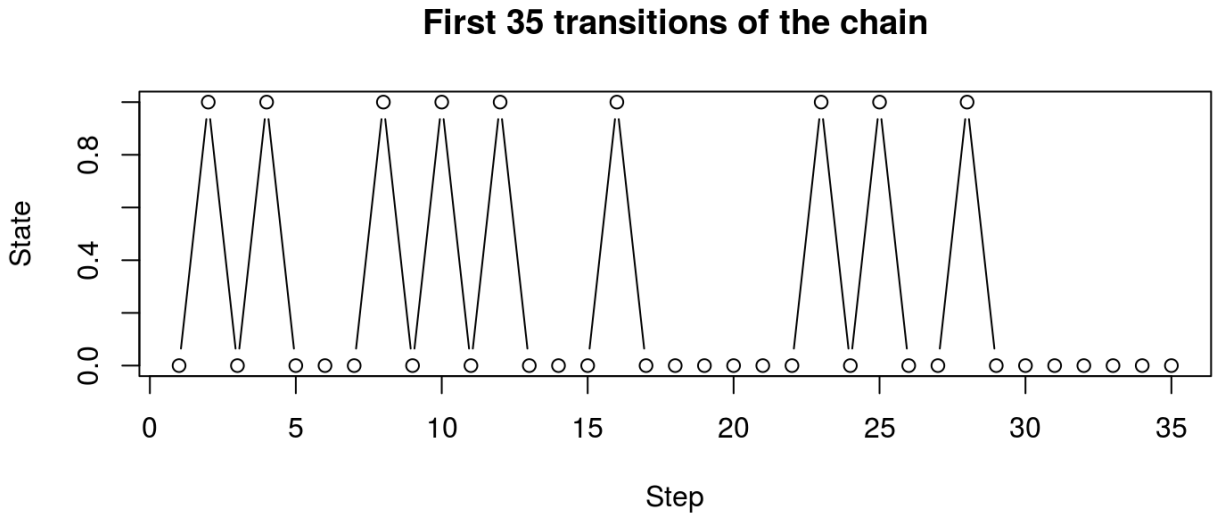
$$\mathbf{P}_{2\times 2} = \begin{pmatrix} p_{0,0} & p_{0,1} \\ p_{1,0} & p_{1,1} \end{pmatrix} = \begin{pmatrix} 0.6 & 0.4 \\ 0.55 & 0.45 \end{pmatrix}$$

Example: 2-State Markov Chain

```
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)
    }
}
```

Example: 2-State Markov Chain

```
## First 35 transitions of the chain
plot(x[1:35], type="b",
     main = "First 35 transitions of the chain",
     xlab="Step", ylab="State")
```



As can be seen in the first $t = 1, \ldots, 35$ transitions of the chain, the Markov chain does not always leave its current state. It transitions with a given transition probability.

Another Example: 3-state Markov chain

For this following example we'll be utilizing the `markovchain` library available from **CRAN**. Revolution Analytics (now an extension of Microsoft) has a nice blog post and tutorial on the use of this package. See *Getting Started with Markov Chains* (http://blog.revolutionanalytics.com/2016/01/getting-started-with-markov-chains.html)

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 have 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 tie 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**.

Another Example: 3-state Markov chain

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

Another Example: 3-state Markov chain

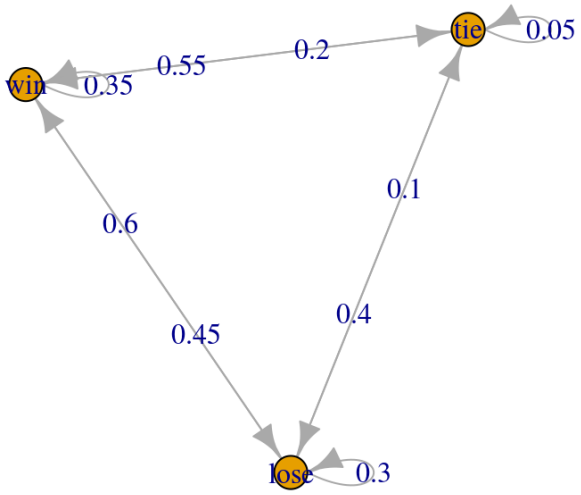
```
library(markovchain)
```

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

```
## 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)
```

Another Example: 3-state Markov chain

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



Limiting Probabilities

The *Chapman-Kolmogorov* equations provide a method for computing 1, 2, ..., *n*-step transition probabilities. The following can be used to calculate the *n*-step probabilities

$$P_{ij}^{n+m} = \sum_{x=0}^{\infty} P_{ix}^{n} P_{xj}^{m} \text{ for all } \ n, m \geq 0, \ \text{all } i, j$$

Let $\mathbf{P}^{(n)}$ denote the *n*-ste transition matrix, then by the *Chapman-Kolmogorov* equations

$$\mathbf{P}^{(n+m)} = \mathbf{P}^{(n)} \cdot \mathbf{P}^{(m)}$$

Thus, we can use simple matrix multiplication to evaluate the probability matrix for step *n*.

Limiting Probabilities

```
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")))

print(P) ## initial probabilities
```

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

```
P_2 = P %*% P
print(P_2)
```

```
##         win   lose    tie
## win  0.5025 0.3725 0.1250
## lose 0.4450 0.4000 0.1550
## tie  0.4600 0.3875 0.1525
```

Limiting Probabilities

```
P_4 = P_2 %*% P_2
print(P_4)
```

```
##            win      lose       tie
## win  0.4757688 0.3846188 0.1396125
## lose 0.4729125 0.3858250 0.1412625
## tie  0.4737375 0.3854438 0.1408188
```

```
P_8 = P_4 %*% P_4
print(P_8)
```

```
##            win      lose       tie
## win  0.4743866 0.3851979 0.1404155
## lose 0.4743798 0.3852007 0.1404195
## tie  0.4743818 0.3851999 0.1404183
```

```
P_16 = P_8 %*% P_8
print(P_16)
```

```
##            win      lose       tie
## win  0.4743833 0.3851992 0.1404175
## lose 0.4743833 0.3851992 0.1404175
## tie  0.4743833 0.3851992 0.1404175
```

```
P_20 = P_16 %*% P_4
print(P_20)
```

Limiting Probabilities

By step 20, it appears that we see a convergence in the *n*-step transition matrix. We've arrived at a limiting distribution for our 3-state Markov chain.

Not all Markov chains have limiting distributions, especially is there is not free movement across all states.

For a more thorough review of Markov chains, see *Chapter 11: Markov Chains* (http://www.dartmouth.edu/~chance/teaching_aids/books_articles/probability_book/Chapter11.pdf).

Ergodic Properties

For MCMC computations, only Markov chains with specific properties can be used. Markov chains are constructed from a *transition kernel K*, a conditional probability density such that $X_{t+1} \sim K(X_t, X_{t+1})$.

As we continue with notes and definitions, note that this will be through the perspective of a discrete-time stochastic processes, $(X_t)_{t \in \mathbb{N}}$

Ergodic Properties

The chains encountered in MCMC settings enjoy a very strong stability property, a *stationary probability distribution*. A stationary probability distribution exists if the kernel *K* allows for free movement all over the state space for all $X_1, X_2, \ldots, X_t$ transitions. 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.

Ergodic Properties

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, regardless of the starting value $X_0 = x_0$. Thus, as a result of these convergence properties, is that the empirical average

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

convergence to the expectation $E[h(X)]$ almost surely.

Another condition to be reviewed is the *detailed balance condition*, also known as *reversibility*.

$$f(x)K(y|x) = f(y)K(x|y)$$

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.

Ergodic Properties

Reversibility is important because it has the effect of balancing movement through the entire state space of a Markov Chain. When a Markov chain is reversible, $f(\cdot)$ is the unique, invariant, stationary distribution of that chain. Hence, if $f(\cdot)$ is of interest, we need only find the reversible Markov chain for which $f(\cdot)$ is the limiting distribution. This is why MCMC works!

These conditions together form what is often referred to the **Ergodic Theorem**. That said, we can now begin delving into MCMC algorithms such as the Metropolis-Hastings algorithm or the Gibbs samplers, which by the ergodic theorem are almost always theretically convergent.