Introduction to discrete Markov chains

John Novembre
University of Chicago
January 15, 2026

See here for a PDF version of this vignette.

Pre-requisites

An understanding of matrix multiplication and matrix powers.

Overview

Here we provide a quick introduction to discrete Markov Chains.

Definition of a Markov Chain

A Markov Chain is a discrete stochastic process with the *Markov property* : $P(X_t|X_{t-1},\ldots,X_1) = P(X_t|X_{t-1})$. It is fully determined by a probability transition matrix P which defines the transition probabilities ($P_{ij} = P(X_t = j|X_{t-1} = i)$ and an initial probability distribution specified by the vector x where $x_i = P(X_0 = i)$. The time-dependent random variable X_t is describing the state of our probabilistic system at time-step t.

Example: Gary's mood

In Sheldon Ross's Introduction to Probability Models, he has an example (4.3) of a Markov Chain for modeling Gary's mood. Gary alternates between 3 state: Cheery ($X = 1$), So-So ($X = 2$), or Glum ($X = 3$). Here we input the P matrix given by Ross and we input an arbitrary initial probability matrix.

```
# Define prob transition matrix
# (note matrix() takes vectors in column form so there is a transpose here to switch col's to ro
P=t(matrix(c(c(0.5,0.4,0.1),c(0.3,0.4,0.3),c(0.2,0.3,0.5)),nrow=3))
# Check sum across = 1
apply(P,1,sum)
# [1] 1 1 1

# Definte initial probability vector
x0=c(0.1,0.2,0.7)
# Check sums to 1
sum(x0)
# [1] 1
```

What are the expected probability states after one or two steps?

If initial prob distribution x_0 is 3×1 column vector, then $x_0^T P = x_1^T$. In R, the %*% operator automatically promotes a vector to the appropriate matrix to make the arguments conformable. In the case of multiplying a length 3 vector by a 3×3 matrix, it takes the vector to be a row-vector. This means our math can be written simply as:

```
# After one step
x0%*%P
#      [,1] [,2] [,3]
# [1,] 0.25 0.33 0.42
```

And after two time-steps:

```
## The two-step prob trans matrix
P%*%P
#      [,1] [,2] [,3]
# [1,] 0.39 0.39 0.22
# [2,] 0.33 0.37 0.30
# [3,] 0.29 0.35 0.36
## Multiplied by the initial state probability
x0%*%P%*%P
#       [,1]  [,2]  [,3]
# [1,] 0.308 0.358 0.334
```

What about an abitrary number of time steps?

To generalize to an arbitrary number of time steps into the future, we can compute a the matrix power. In R, this can be done easily with the package expm. Let's load the library and verify the second power is the same as we saw for P%*%P above.

```
# Load library
library(expm)
# Loading required package: Matrix
#
# Attaching package: 'expm'
# The following object is masked from 'package:Matrix':
#
#     expm
# Verify the second power is P%*%P
P%^%2
#      [,1] [,2] [,3]
# [1,] 0.39 0.39 0.22
# [2,] 0.33 0.37 0.30
# [3,] 0.29 0.35 0.36
```

And now let's push this by looking at the state of the chain after many steps, say 100. First let's look at the probability transition matrix...

```
P%^%100
#        [,1]  [,2]   [,3]
# [1,] 0.3387 0.371 0.2903
# [2,] 0.3387 0.371 0.2903
# [3,] 0.3387 0.371 0.2903
```

What do you notice about the rows? And let's see what this does for various different starting distributions:

```
c(1,0,0) %*%(P%^%100)
#        [,1]  [,2]   [,3]
# [1,] 0.3387 0.371 0.2903
c(0.2,0.5,0.3) %*%(P%^%100)
#        [,1]  [,2]   [,3]
# [1,] 0.3387 0.371 0.2903
```

Note that after a large number of steps the initial state does not matter any more, the probability of the chain being in any state j is independent of where we started. This is our first view of the *equilibrium distribuion* of a Markov Chain. These are also known as the *limiting probabilities of a Markov chain* or *stationary distribution*.