Math 108B	Professor: Padraic Bartlett

Lecture 4.5: Markov Processes

Week 5	*UCSB 2014*

On Friday of week 1, we asked the following question: given the matrix $A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$, is there a quick way to calculate large powers of this matrix? Our answer to this question turned out to be yes! We did this by writing

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} \frac{\varphi}{\sqrt{1+\varphi^2}} & -\frac{1}{\sqrt{1+\varphi^2}} \\ \frac{1}{\sqrt{1+\varphi^2}} & \frac{\varphi}{\sqrt{1+\varphi^2}} \end{bmatrix} \cdot \begin{bmatrix} \varphi & 0 \\ 0 & (-\frac{1}{\varphi}) \end{bmatrix} \cdot \begin{bmatrix} \frac{\varphi}{\sqrt{1+\varphi^2}} & \frac{1}{\sqrt{1+\varphi^2}} \\ -\frac{1}{\sqrt{1+\varphi^2}} & \frac{\varphi}{\sqrt{1+\varphi^2}} \end{bmatrix},$$

which is particularly nice because it writes A in the form UDU^{-1}, where D is a diagonal matrix (i.e. its only nonzero entries are on its diagonal.) This lets us find A^k very quickly, as it is just

$$UD\cancel{U^{-1}} \cdot \cancel{U}D\cancel{U^{-1}} \cdot \ldots \cdot \cancel{U}DU^{-1} = UD^kU^{-1},$$

which is very easy to calculate!

In general, finding large powers of various matrices is a task that mathematicians want to be able to do very quickly. One example of when we would want to calculate large powers of matrices is for studying Markov processes! Loosely speaking, a Markov process is a method for predicting some system or object over a period of time. It consists of the following pair of objects:

- States: Every Markov process has a number of states, that correspond to potential For example, suppose that our Markov chain was simulating the weather for Goleta on a given day; then our states would be things like "sunny," "cloudy," and "rainy."
- Links: For each state, the Markov process also records a number of **probabilities**, that denote the likelihood that when we advance one time step, our system changes from one state to another. Again, for example, suppose we're still making a Markov chain to simulate the weather. From the state "sunny," we would want to come up with probabilities that our weather stays sunny (say 80%), becomes cloudy (say 10%) or becomes rainy (say 10%) on the next day.

Given a Markov process, we often want to know about the **long-term** outcome of that process. We can simulate this by taking powers of appropriately-constructed matrices! Consider the following example:

Example. A gambler is in Vegas, and is currently betting on the following (very simple) event: flip a coin. If it comes up heads, you win twice what you bet; if it comes up tails, you lose what you bet. You can only bet $1 on each flip. Suppose that your gambler starts with

\$5, and will leave the table either when they're either broke or have doubled their money (\$10.) Also, because this is Vegas, suppose that the coin is imperceptibly biased towards the house — i.e. it comes up heads 49% of the time, and tails 51% of the time.

How often does the gambler walk away broke?

We can answer this problem with some linear algebra and with a Markov process! We start by describing the process: our **states** are the various amounts of money the gambler can ever have (i.e. the numbers $0 \ldots 11$), and the probabilities that tell us from a given state how likely we are to move from one state to another state. These probabilities are the following:

- From the state \$0: you return to state \$0 100% of the time (because you're broke.)
- From the states $\$n$, for n ranging from 1 to 9: you go to state $\$(n-1)$ 51% of the time, and to state $\$(n+1)$ 49% of the time.
- From the state \$10: you return to state \$10 100% of the time (because you've walked away from the game, as you've doubled your money.)

We can simulate this process by using a matrix! Specifically, given a Markov process with n states, label its states $1 \ldots n$. Use this labeling to construct the following $n \times n$ matrix: in each cell (i, j), put the probability that we will go to state i from state j. (If this reminds you of the PageRank algorithm from Wednesday of week 1, you're right!)

For the example we're looking at, our matrix is the following object:

$$A = \begin{bmatrix} 1 & .51 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & .51 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & .49 & 0 & .51 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & .49 & 0 & .51 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & .49 & 0 & .51 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & .49 & 0 & .51 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & .49 & 0 & .51 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & .49 & 0 & .51 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & .49 & 0 & 1 \end{bmatrix}$$

What can we do with this matrix? Well: we can use it to predict where our gambler will go! In particular, suppose that we know our gambler starts with 5 dollars. Then, we expect him at his next step to have either \$4 or \$6, with probabilities 51% and 49% respectively; and at the step after that to have either \$3, \$5 or \$7 with probabilties 26.01%, 49.98% and 24.01% respectively.

So: we can use our matrix to calculate this! In particular, if we plug in the vector $\vec{p} = (0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0)$ into our matrix above, we'll get$A\vec{p} = (0, 0, 0, 0, .51, 0, .49, 0, 0, 0, 0)$ back out, which we can interpret as the probability that our gambler is in state \$4 with likelihood 51% and in the state \$6 with likelihood 49%. Moreover, we can calculate that $A^2\vec{p} = (0, 0, 0, .2601, .4998, .2401, 0, 0, 0, 0)$, which we can again interpret as the statement that our gambler is in the states \$3, \$5 or \$7 with probabilties 26.01%, 49.98% and 24.01% respectively.

In general, suppose that at some state we have a probability vector $\vec{p} = (p_1, \ldots p_n)$ that denotes the likelihoods that our process is in one of the states $1 \ldots n$ respectively. Then the vector $A\vec{p}$ consists of precisely the likelihoods that our process is in the states $1 \ldots n$ after advancing our process one step!

Then, if we want to simulate what our gambler does according to the Markov process, we just need to use this matrix! In specific: we assumed that our gambler starts with $5 dollars. To simulate what happens to our gambler in the long run, we just need to find A^n for large powers of n, and look at $A^n \cdot (0,0,0,0,0,1,0,0,0,0,0)$. This will tell us where our gambler is at that time!

So: we want to calculate A^n for large powers of n. By grabbing a handy copy of Mathematica/Sage/a TI-92, we can calculate things like A^{100}, and see that $A^{100} \cdot \vec{p}$ is about

$$(.55, 0, 0, 0, 0, 0, 0, 0, 0, 0, .45).$$

In other words, in the long run, our gambler is broke 55% of the time and walked away with double their money 45% of the time.

But how does Mathematica do this? How can we do similar things? We were able to do this for our matrix $\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$; how do we do it in general? Roughly speaking, this is the goal of our course for the rest of the quarter — develop quick ways to manipulate matrices, and look at what these techniques let us do in other areas of mathematics.