## Absorbing Markov chains (sections 11.1 and 11.2)

What is a Markov chain, really?

That is, what kind of mathematical object is it?

It's NOT a special kind of stochastic matrix (although we do use stochastic matrices to understand the behavior of Markov chains).

An *n*-state Markov chain is a probability measure on a space of (finite or infinite) sequences of states.

It's easiest to explain this in the case of an absorbing Markov chain.

Example: The Markov chain on the state-space {1,2,3,4} with transition matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

and initial state 2 is a random variable taking the value (2,1,1,1,1,1,...) with probability

$$p_{21}\, p_{11}\, p_{11}\, p_{11}\, p_{11} \ldots = 1/2,$$

the value (2,3,4,4,4,4,...) with probability

$$p_{23}\; p_{34}\; p_{34}\; p_{34}\; p_{34} \ldots = 1/4,$$

the value (2,3,2,1,1,1,...) with probability

$$p_{23}\; p_{32}\; p_{21}\, p_{11}\, p_{11} \ldots = 1/8,$$

the value (2,3,2,3,4,4,...) with probability

$$p_{23}\; p_{32}\; p_{23}\; p_{34}\; p_{44} \ldots = 1/16,$$

et c.

(and all other values have probability 0).

Note that (using the terminology introduced in the second lecture), things like (2,3,4,4,4,4,...) are not "states" but outcomes, or elements of the probability space.

We have an (infinite) probability space

$$\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4, \ldots\}$$

with

$\omega_1 = (2,1,1,1,1,1,...)$,

$\omega_2 = (2,3,4,4,4,4,...)$,

$\omega_3 = (2,3,2,1,1,1,...)$,

$\omega_4 = (2,3,2,3,4,4,...)$,

...

and a probability measure $m$ satisfying

$m(\omega_i) = 1/2^i$ for all $i$.

If you want, you can include the outcome (2,3,2,3,2,3,...), which has probability 0.

Or you can leave it out. Either way, we get a set of outcomes, and an assignment of probabilities to those outcomes such that every outcome has non-negative probability and the sum of the probabilities of all the outcomes equals 1, which is precisely what we mean by a probability space.

Sometimes it's more handy to index time starting from 0 rather than 1; that is, we use $(x_0, x_1, x_2, ...)$ instead of $(x_1, x_2, x_3, ...)$.

We formalize the notion of an $n$-state Markov chain by defining the set

$\Omega = \{(x_0, x_1, x_2, ...): 1 \le x_i \le n \text{ for all } i \ge 0\}$

and turning it into the probability space by giving it the probability measure

$m((x_0, x_1, x_2, ...)) = q_{x_0}\, p_{x_0 x_1}\, p_{x_1 x_2}\, p_{x_2 x_3} \ldots$

This prescription for building a probability space doesn't make sense for <u>all</u> Markov chains; e.g., if **P** is the 2-by-2 matrix

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

then for every $(x_0, x_1, x_2, ...)$ we get an infinite product with infinitely many factors equal to 1/2, which converges to 0. So each individual outcome $\omega$ has $m(\omega) = 0$! Later on I'll say a little bit about how we make sense of Markov chains in such cases.

But for <u>absorbing</u> Markov chains, it turns out we don't

have to worry about such things.

**Fact**: If the Markov chain associated with the matrix **P** is absorbing, then the sum of $m(\omega)$ over all $\omega$ in $\Omega$ is 1. So we have a well-defined discrete probability space.

(Note: For some sequences $\omega$ we have $m(\omega) = 0$ because one of the transitions in $\omega$ corresponds to an entry of the transition matrix that equals 0.

E.g., for our standard 4-state example, $m((2,3,2,2,1,1,1,...))$ because $p_{22} = 0$.

And for other sequences $\omega$ we have $m(\omega) = 0$ because the infinite product vanishes, even if none of the fac-tors vanish.

E.g., for our standard 4-state example, $m((2,3,2,3,2,3,...)) = (1/2)(1/2)(1/2)... = 0$.

But because every absorbing Markov chain has states $x$ with $p_{xx} = 1$, we have many outcomes $\omega$ with $m(\omega) > 0$.

And indeed if we sum $m(\omega)$ over all $\omega$ we get 1, as long as our Markov chain is an absorbing Markov chain.

This means that to compute the probability of some event $E \subseteq \Omega$ we just need to compute the (possibly infi - nite) sum $\sum_{\omega \text{ in } E} m(\omega)$.

E.g., returning to our favorite example, to compute the probability of the event "The Markov chain eventually gets to state 4", we compute Prob($\{\omega_2, \omega_4, ...\}$ = $m(\omega_2)$ + $m(\omega_4)$ + ... =
1/4 + 1/16 + ... = 1/3.

For most absorbing Markov chains, the space of outc - omes has a much more complicated structure, so we can't answer questions like this by just summing geo- metric series. Instead, we use linear algebra.
For instance, returning to our favorite 4-state Markov chain, the probability the chain enters state 4 within

10 steps can be computed as the 2,4th entry of the matrix $\mathbf{P}^{10}$, and the probability that the chain eventually enters state 4 can be computed as the 2,4th entry of the matrix

$$\mathbf{P}^{\infty} := \lim_{n\to\infty} \mathbf{P}^{n}.$$

*Mathematica* caveat: don't confuse `P^n` with `MatrixPower[P,n]`!:

```
P = {{1, 0, 0, 0}, {1 / 2, 0, 1 / 2, 0}, {0, 1 / 2, 0, 1 / 2}, {0, 0, 0, 1}}

P^2

MatrixPower[P, 2]

N[MatrixPower[P, 10][[2, 4]]]

Limit[MatrixPower[P, m], m → ∞][[2, 4]]
```

But don't make the mistake of confusing these kinds of calculations with simulations, and don't mistake a Markov chain (a probability distribution on sequences of states) with the associated matrix.

The matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

is just a matrix, not a Markov chain; its entries determine a probability measure on the space of sequences of 1's, 2's, 3's and 4's, and this probability measure is what we mean by a Markov chain.

More generally, a probability space whose outcomes are sequences is called a **discrete-time stochastic process**.

(Note that a sequence $\omega = (x_0, x_1, x_2, ...)$ can also be thought of as a function whose domain is the set of non-negative integers. Later we'll see probability spaces whose outcomes are functions on $\{t \text{ in } \mathbb{R}: t \geq 0\}$ rather than on {0,1,2,...}; such a probability space is called a **continuous-time stochastic process**.)

When the elements of $\Omega$ are sequences of the form $(x_0, x_1, x_2, ...)$ (where $x_0, x_1, x_2, ...$ are elements of $\{1,2,3,...,n\}$), it's natural to define random variables $X_0, X_1, X_2, ...$ on $\Omega$. Remember that a random variable is just a function on $\Omega$. We define

$X_0(x_0, x_1, x_2, ...) = x_0$,

$X_1(x_0, x_1, x_2, ...) = x_1$,

$X_2(x_0, x_1, x_2, ...) = x_2$,
et c.

So a random element of $\Omega$ can (somewhat circularly!) be written as the sequence
$(X_0(\omega), X_1(\omega), X_2(\omega), ...)$

Note that we've been taking the $x$'s to be elements of $\{1,2,...,n\}$ where $n$ is the number of states. This is a good choice when the states ARE just the numbers $1,2,...,n$; but when the state $s_i$ is different from the number $i$, it's often handy to use $(s_{x_0}, s_{x_1}, s_{x_2}, ...)$ instead of $(x_0, x_1, x_2, ...)$.

Simulating a generic absorbing Markov chain

Simulating a Markov chain means generating a particular sequence $x_1, x_2, x_3, ...$ in accordance with the transition probabilities $p_{ij}$. More specifically, suppose you've already picked $x_1, x_2, ..., x_m$, and you chose $x_m = i$, and now it's time to choose $x_{m+1}$. Then your selection procedure must have the property that your chance of choosing $x_{m+1} = j$ should be $p_{ij}$, regardless of what $x_1, x_2, ..., x_{m-1}$ were.

That is, the probability of the compound event

$$X_1 = x_1,\ X_2 = x_2,\ X_3 = x_3,\ \ldots,\ X_m = x_m$$

should be of the form

$$q_{x_1}\, p_{x_1 x_2}\, p_{x_2 x_3}\, p_{x_3 x_4} \ldots p_{x_{m-1} x_m}$$

where $q_{x_1}$ is the probability $\text{Prob}(X_1 = x_1)$.

Let's write code to simulate one step of the Markov chain associated with an $n$-by-$n$ transition matrix **P**, starting from state $i$.

First let's solve the sub-problem of simulating a discrete random variable that takes its values on the set {1,2,...,$n$}, so that the probabilities of the outcomes 1,2,...,$n$ are precisely the entries of a specified probability vector **v**.

To do this, we divide the interval [0,1] from left to right into subintervals of lengths $v_1$, $v_2$, $v_3$, ... and pick a random real number $r$ in [0,1]; our output should be the unique $i$ such that $r$ is in $v_i$. Note that the subintervals are

$[0, v_1]$, $[v_1, v_1+v_2]$, $[v_1+v_2, v_1+v_2+v_3]$, ..., so we can also describe the output as the smallest $i$ such that $v_1+v_2+\ldots+v_i$ exceeds r.

```
DRV[v_] := (* sample from the discrete random variable associated with the vector v *)
 Module[{i, sum, r},
  r = RandomReal[]; i = 1; sum = v[[1]];
  While[sum < r, i++; sum += v[[i]]];
  Return[i]]
(* If we wanted more efficiency,
we could use a bisection method to find the smallest such i. *)

Table[DRV[{1 / 2, 1 / 4, 1 / 4}], {n, 1, 100}]

Tally[%]

Tally[Table[DRV[{1 / 2, 1 / 4, 1 / 4}], {n, 1, 1000}]]
```

Now it's easy to write code to simulate a single step of the Markov chain with transition matrix **P**, starting from state *i*.

```
MarkovStep[P_, i_] := DRV[P[[i]]]
```

Using this we can simulate *n* steps of the Markov chain:

```
MarkovMultiStep[P_, i_, n_] := Module[{State = i, k},
  Print[State];
  For[k = 1, k ≤ n, k++,
   State = MarkovStep[P, State];
   Print[State]]]

(* Note that ";" takes precedence over "," in For[...] and If[...] statements! *)

P = {{1, 0, 0, 0}, {1 / 2, 0, 1 / 2, 0}, {0, 1 / 2, 0, 1 / 2}, {0, 0, 0, 1}}

MarkovMultiStep[P, 2, 10]
```

Now let's write code to simulate the chain starting from state until it enters a transient state.

```
MarkovAbsorb[P_, i_] := Module[{State = i, k},
  Print[State];
  For[k = 0, P[[State, State]] < 1, k++,
   State = MarkovStep[P, State];
   Print[State]]; Return[k]]

MarkovAbsorb[P, 2]

MarkovAbsorb[P_, i_] := Module[{State = i, k},
  For[k = 0, P[[State, State]] < 1, k++,
   State = MarkovStep[P, State]]; Return[k]]
```

Let's see what sorts of things we get if we run this process 1000 times.

```
Histogram[Table[MarkovAbsorb[P, 2], {n, 1, 1000}]]
```

For comparison, let's see what sorts of things we get if we sample from a geometric random variable (with parameter $p = 1/2$) 1000 times:

```
Histogram[Table[RandomInteger[GeometricDistribution[1 / 2]], {n, 1, 1000}]]
```

The histograms look similar, which makes sense, because the time it takes to get from state 2 to an absorbing state is a geometric random variable with parameter $p = 1/2$:

Canonical form (review)

We renumber the states so that the transient states come first.

Suppose the chain has $t$ transient states and $r$ absorbing states. Then we can write the canonical matrix in block-form as

$$\begin{pmatrix} \mathbf{Q} & \mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}$$

where

**Q** is a $t$-by-$t$ square matrix,

**R** is a non-zero $t$-by-$r$ matrix,

**0** is the all-zeroes $r$-by-$t$ matrix, and

**I** is the $r$-by-$r$ identity matrix.

We say such a transition matrix is in canonical form.

Number of visits and the fundamental matrix (review)

The matrix **I** - **Q** is invertible (where **I** here stands for the $t$-by-$t$ identity matrix), and its inverse can be written as the convergent infinite sum $\mathbf{N} = \mathbf{I} + \mathbf{Q} + \mathbf{Q}^2 + \ldots$. The matrix **N** is called the fundamental matrix for the absorbing Markov chain.

Claim: The $ij$-entry $n_{ij}$ of the matrix **N** is the expected number of times the chain is in state $s_j$, given that it starts in state $s_i$. The initial state is counted (as part of "the number of times...") if $i = j$.

Claim: Let $t_i$ be the expected number of steps before the chain is absorbed,
given that the chain starts in state $s_i$, and let **t** be the column vector whose $i$th entry is $t_i$. Then **t** = **N** **c**, where **c** is the column vector all of whose entries are 1.

Absorption probabilities

Let **B** = **N** **R** (with **N**, **R** as in the canonical form).

Claim: The $ij$-entry $b_{ij}$ of the matrix **B** is the probability that an absorbing chain started in the transient state $s_i$ will be absorbed in the absorbing state $s_j$ .

Proof 1: The probability in question is equal to
$\sum_{n\geq 0} \sum_k q_{ik}^{(n)} r_{kj} = \sum_k \sum_{n\geq 0} q_{ik}^{(n)} r_{kj} = \sum_k n_{ik} r_{kj}$ , which is $b_{ij}$. (Here $n$ denotes the number of steps the Markov

chain takes before it leaves the set of transient states, and k denotes the last transient state the chain visits.)

before it hits the absorbing state i, and n denotes the number of steps the chain takes

Proof 2: Let $c_{ij}$ be the probability that an absorbing chain started in the transient state $s_i$ will be absorbed in the absorbing state $s_j$. If we compute $c_{ij}$ in terms of the possibilities on the outcome of the first step, we have

$$c_{ij} = p_{ij} + \sum_k p_{ik}\, c_{kj} \ ,$$

where $k$ ranges over all the transient states. This is true for all $i,j$. This family of equations is equivalent to the single matrix equation $\mathbf{C} = \mathbf{R} + \mathbf{Q}\mathbf{C}$, which we can successively rewrite as $\mathbf{C} - \mathbf{Q}\mathbf{C} = \mathbf{R}$, $(\mathbf{I} - \mathbf{Q})\mathbf{C} = \mathbf{R}$, and $\mathbf{C} = (\mathbf{I} - \mathbf{Q})^{-1}\mathbf{R} = \mathbf{N}\mathbf{R} = \mathbf{B}$.

```
MatrixForm[FM.R]
```

This agrees with what we proved before ($P = 1/3$).

## Questions about the homework?

Hint for A:

See the discussion "Can we construct a Geometric( *p*) random variable from a Uniform(0,1) random variable?" from near the end of Lecture 2.

Hint for B:

```
P = {{.5, .5, 0}, {.25, .5, .25}, {0, .5, .5}}

N[MatrixPower[P, 1000000000]]

Limit[MatrixPower[P, k], k → ∞]

(* What's going on here? *)

P = {{1 / 2, 1 / 2, 0}, {1 / 4, 1 / 2, 1 / 4}, {0, 1 / 2, 1 / 2}}

(* Moral: Don't trust Mathematica's built-
  in ideas about handling limits in conjunction with decimal constants! *)
```

## *Mathematica* reimbursement

Please hand in your Expense Approval forms (for purchase of copies of *Mathematica*). Next Tuesday will be your last chance to do this if you want to reimbursed anytime soon.

Name of Person or Business To Be Reimbursed: <*your name*>
Date: <*today's date*>
Department: Mathematical Sciences
Remit To Address: <*your address*>
Purpose for Incurring the Expense: Purchase of soft - ware related to PI's research
Total: [make entries on separate lines for purchase price, sales tax, and total]
Signature (Person Incurring Expense): <your signature>

## Ergodic Markov chains (sections 11.3 and 11.5)

### Ergodicity

Definition 11.4: A Markov chain is called an ergodic chain if it is possible to go from every state to every state in a finite number of steps. (Note that no absorb -

ing Markov chain is ergodic, aside from the trivial absorbing Markov chains with zero or one absorbing states and no transient states.)

Example: Random walk on {1,2,3,4} with reflection on the ends (1→2 with probability 1 and 4→3 with probability 1) is ergodic.

```
Ref = {{0, 1, 0, 0}, {1 / 2, 0, 1 / 2, 0}, {0, 1 / 2, 0, 1 / 2}, {0, 0, 1, 0}};
MatrixForm[MatrixPower[Ref, 4]]
```

Note that for this example, it is possible to go from every state $s_i$ to every state $s_j$ in a finite number $N(i,j)$ of steps, but it is impossible to remove the dependence of $N(i,j)$ on $i$ and $j$; that is, there does not exist $N$ such that for all $i$ and $j$, it is possible to go from $s_i$ to $s_j$ in exactly $N$ steps. That is because $N(i,j)$ must have the same parity as $i$-$j$.

We say a Markov chain is regular if there exists an $N$ such that for all $i$ and $j$, it is possible to go from $s_i$ to $s_j$ in exactly $N$ steps; that is, the $N$th power of the transition matrix $\mathbf{P}$ has all its entries positive (see Definition 11.5). Note that every regular Markov chain is ergodic.

The reflecting random walk described above is ergodic but not regular. A slightly different reflecting random walk that is regular can be obtained from our non-regular example by changing the behavior at the ends so that reflection is only partial:

```
ParRef = {{1 / 2, 1 / 2, 0, 0}, {1 / 2, 0, 1 / 2, 0}, {0, 1 / 2, 0, 1 / 2}, {0, 0, 1 / 2, 1 / 2}};
MatrixForm[MatrixPower[ParRef, 3]]
```