Section 5.6

Stochastic Matrices and PageRank

Stochastic Matrices

Definition

A square matrix A is **stochastic** if all of its entries are nonnegative, and the sum of the entries of each column is 1.

We say A is **positive** if all of its entries are positive.

These arise very commonly in modeling of probabalistic phenomena (Markov chains).

You'll be responsible for knowing basic facts about stochastic matrices, the Perron–Frobenius theorem, and PageRank, but we will not cover them in depth.

Stochastic Matrices

Example

Red Box has kiosks all over where you can rent movies. You can return them to any other kiosk. Let A be the matrix whose ij entry is the probability that a customer renting a movie from location j returns it to location i. For example, if there are three locations, maybe

$$A = \begin{pmatrix} .3 & .4 & .5 \\ .3 & .4 & .3 \\ .4 & .2 & .2 \end{pmatrix}.$$

30% probability a movie rented from location 3 gets returned to location 2

The columns sum to 1 because there is a 100% chance that the movie will get returned to *some* location. This is a positive stochastic matrix.

Note that, if $v = (x, y, z)$ represents the number of movies at the three locations, then (assuming the number of movies is large), Red Box will have approximately

$$Av = A\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} .3x + .4y + .5z \\ \boxed{.3x + .4y + .3z} \\ .4x + .2y + .2z \end{pmatrix}$$

"The number of movies returned to location 2 will be (on average):
30% of the movies from location 1;
40% of the movies from location 2;
30% of the movies from location 3"

movies in its three locations the next day. The *total number* of movies doesn't change because the columns sum to 1.

Stochastic Matrices and Difference Equations

If x_n, y_n, z_n are the numbers of movies in locations $1, 2, 3$, respectively, on day n, and $v_n = (x_n, y_n, z_n)$, then:

$$v_n = Av_{n-1} = A^2 v_{n-2} = \cdots = A^n v_0.$$

Recall: This is an example of a **difference equation**.

Red Box probably cares about what v_n is as n gets large: it tells them where the movies will end up *eventually*. This seems to involve computing A^n for large n, but as we will see, they actually only have to compute one eigenvector.

In general: A difference equation $v_{n+1} = Av_n$ is used to model a state change controlled by a matrix:

- v_n is the "state at time n",
- v_{n+1} is the "state at time $n + 1$", and
- $v_{n+1} = Av_n$ means that A is the "change of state matrix."

Eigenvalues of Stochastic Matrices

Fact: 1 is an eigenvalue of a stochastic matrix.

Why? If A is stochastic, then 1 is an eigenvalue of A^T:

$$\begin{pmatrix} .3 & .3 & .4 \\ .4 & .4 & .2 \\ .5 & .3 & .2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = 1 \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

Lemma

A and A^T have the same eigenvalues.

Proof: $\det(A - \lambda I) = \det\big((A - \lambda I)^T\big) = \det(A^T - \lambda I)$, so they have the same characteristic polynomial.

Note: This doesn't give a new procedure for finding an eigenvector with eigenvalue 1; it only shows one exists.

Eigenvalues of Stochastic Matrices

Continued

Fact: if λ is an eigenvalue of a stochastic matrix, then $|\lambda| \leq 1$. Hence 1 is the *largest* eigenvalue (in absolute value).

Why? If λ is an eigenvalue of A then it is an eigenvalue of A^T.

$$\text{eigenvector } v = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \qquad \lambda v = A^T v \implies \lambda x_j = \overset{j\text{th entry of } A^T v}{\boxed{\textstyle\sum_{i=1}^n a_{ij} x_i .}}$$

Choose x_j with the largest absolute value, so $|x_i| \leq |x_j|$ for all i.

$$|\lambda| \cdot |x_j| = \left| \sum_{i=1}^n a_{ij} x_i \right| \leq \sum_{i=1}^n \overset{\text{positive}}{a_{ij}} \cdot |x_i| \leq \sum_{i=1}^n a_{ij} \cdot \underset{\geq |x_i|}{|x_j|} = \overset{= \sum_i a_{ij}}{1} \cdot |x_j|,$$

so $|\lambda| \leq 1$.

Better fact: if $\lambda \neq 1$ is an eigenvalue of a *positive* stochastic matrix, then $|\lambda| < 1$.

Diagonalizable Stochastic Matrices

Example from §5.4

Let $A = \begin{pmatrix} 3/4 & 1/4 \\ 1/4 & 3/4 \end{pmatrix}$. This is a positive stochastic matrix.

This matrix is diagonalizable:

$$A = CDC^{-1} \quad \text{for} \quad C = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad D = \begin{pmatrix} 1 & 0 \\ 0 & 1/2 \end{pmatrix}.$$

Let $w_1 = \binom{1}{1}$ and $w_2 = \binom{1}{-1}$ be the columns of C.

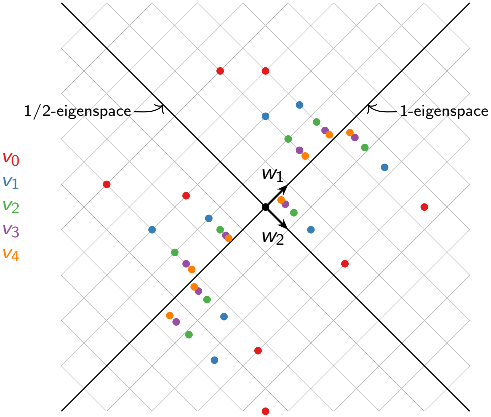
$$\begin{aligned} A(a_1w_1 + a_2w_2) &= a_1w_1 + \frac{1}{2}a_2w_2 \\ A^2(a_1w_1 + a_2w_2) &= a_1w_1 + \frac{1}{4}a_2w_2 \\ A^3(a_1w_1 + a_2w_2) &= a_1w_1 + \frac{1}{8}a_2w_2 \\ &\vdots \\ A^n(a_1w_1 + a_2w_2) &= a_1w_1 + \frac{1}{2^n}a_2w_2 \end{aligned}$$

When n is large, the second term disappears, so A^nx approaches a_1w_1, which is an *eigenvector with eigenvalue* 1 (assuming $a_1 \neq 0$). So all vectors get "sucked into the 1-eigenspace," which is spanned by $w_1 = \binom{1}{1}$.

Diagonalizable Stochastic Matrices

Example, continued

[interactive]



All vectors get “sucked into the 1-eigenspace.”

Diagonalizable Stochastic Matrices

The Red Box matrix $A = \begin{pmatrix} .3 & .4 & .5 \\ .3 & .4 & .3 \\ .4 & .2 & .2 \end{pmatrix}$ has characteristic polynomial

$$f(\lambda) = -\lambda^3 + 0.12\lambda - 0.02 = -(\lambda - 1)(\lambda + 0.2)(\lambda - 0.1).$$

So 1 is indeed the largest eigenvalue. Since A has 3 distinct eigenvalues, it is diagonalizable:

$$A = C \begin{pmatrix} 1 & 0 & 0 \\ 0 & .1 & 0 \\ 0 & 0 & -.2 \end{pmatrix} C^{-1} = CDC^{-1}.$$

Hence it is easy to compute the powers of A:

$$A^n = C \begin{pmatrix} 1 & 0 & 0 \\ 0 & (.1)^n & 0 \\ 0 & 0 & (-.2)^n \end{pmatrix} C^{-1} = CD^n C^{-1}.$$

Let w_1, w_2, w_3 be the columns of C, i.e. the eigenvectors of C with respective eigenvalues $1, .1, -.2$.

Diagonalizable Stochastic Matrices

Continued

Let $a_1 w_1 + a_2 w_2 + a_3 w_3$ be any vector in $\mathbf{R}^3$.

$$\begin{aligned}
A(a_1 w_1 + a_2 w_2 + a_3 w_3) &= a_1 w_1 + (.1) a_2 w_2 + (-.2) a_3 w_3 \\
A^2(a_1 w_1 + a_2 w_2 + a_3 w_3) &= a_1 w_1 + (.1)^2 a_2 w_2 + (-.2)^2 a_3 w_3 \\
A^3(a_1 w_1 + a_2 w_2 + a_3 w_3) &= a_1 w_1 + (.1)^3 a_2 w_2 + (-.2)^3 a_3 w_3 \\
&\vdots \\
A^n(a_1 w_1 + a_2 w_2 + a_3 w_3) &= a_1 w_1 + (.1)^n a_2 w_2 + (-.2)^n a_3 w_3
\end{aligned}$$

As n becomes large, this approaches $a_1 w_1$, which is an *eigenvector with eigenvalue* 1 (assuming $a_1 \neq 0$). So all vectors get "sucked into the 1-eigenspace," which (I computed) is spanned by

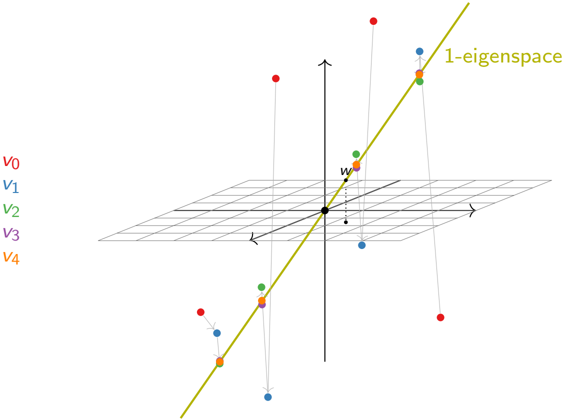
$$w = w_1 = \frac{1}{18}\begin{pmatrix} 7 \\ 6 \\ 5 \end{pmatrix}.$$

(We'll see in a moment why I chose that eigenvector.)

Diagonalizable Stochastic Matrices

Picture

Start with a vector v_0 (the number of movies on the first day), let $v_1 = Av_0$ (the number of movies on the second day), let $v_2 = Av_1$, etc.



We see that v_n approaches an eigenvector with eigenvalue 1 as n gets large: all vectors get “sucked into the 1-eigenspace.” [interactive]

Diagonalizable Stochastic Matrices

Interpretation

If A is the Red Box matrix, and v_n is the vector representing the number of movies in the three locations on day n, then

$$v_{n+1} = Av_n.$$

For any starting distribution v_0 of videos in red boxes, after enough days, the distribution v ($= v_n$ for n large) is an eigenvector with eigenvalue 1:

$$Av = v.$$

In other words, eventually each kiosk has the same number of movies, every day.

Moreover, we know exactly what v is: it is the multiple of $w \sim (0.39, 0.33, 0.28)$ that represents the same number of videos as in v_0. (Remember the total number of videos never changes.)

Presumably, Red Box really does have to do this kind of analysis to determine how many videos to put in each box.

Perron–Frobenius Theorem

Definition

A *steady state* for a stochastic matrix A is an eigenvector w with eigenvalue 1, such that all entries are *positive* and sum to 1.

Perron–Frobenius Theorem

If A is a positive stochastic matrix, then it admits a unique steady state vector w, which spans the 1-eigenspace.

Moreover, for any vector v_0 with entries summing to some number c, the iterates $v_1 = Av_0$, $v_2 = Av_1$, ..., $v_n = Av_{n-1}$, ..., approach cw as n gets large.

Translation: The Perron–Frobenius Theorem says the following:

- The 1-eigenspace of a positive stochastic matrix A is a line.
- To compute the steady state, find any 1-eigenvector (as usual), then divide by the sum of the entries; the resulting vector w has entries that sum to 1, and are *automatically* positive.
- Think of w as a vector of steady state *percentages*: if the movies are distributed according to these percentages today, then they'll be in the same distribution tomorrow.
- The sum c of the entries of v_0 is the total number of movies; eventually, the movies arrange themselves according to the steady state percentage, i.e., $v_n \to cw$.

Steady State

Red Box example

Consider the Red Box matrix $A = \begin{pmatrix} .3 & .4 & .5 \\ .3 & .4 & .3 \\ .4 & .2 & .2 \end{pmatrix}$.

I computed $\mathrm{Nul}(A - I)$ and found that

$$w' = \begin{pmatrix} 7 \\ 6 \\ 5 \end{pmatrix}$$

is an eigenvector with eigenvalue 1.

To get a steady state, I divided by $18 = 7 + 6 + 5$ to get

$$w = \frac{1}{18} \begin{pmatrix} 7 \\ 6 \\ 5 \end{pmatrix} \sim (0.39, 0.33, 0.28).$$

This says that eventually, 39% of the movies will be in location 1, 33% will be in location 2, and 28% will be in location 3, every day.

So if you start with 100 total movies, eventually you'll have $100w = (39, 33, 28)$ movies in the three locations, every day.

The Perron–Frobenius Theorem says that our analysis of the Red Box matrix works for *any* positive stochastic matrix—whether or not it is diagonalizable!

Google's PageRank

Internet searching in the 90's was a pain. Yahoo or AltaVista would scan pages for your search text, and just list the results with the most occurrences of those words.

Not surprisingly, the more unsavory websites soon learned that by putting the words "Alanis Morissette" a million times in their pages, they could show up first every time an angsty teenager tried to find *Jagged Little Pill* on Napster.

Larry Page and Sergey Brin invented a way to rank pages by *importance*. They founded Google based on their algorithm.

Here's how it works. (roughly)

Reference:

`http://www.math.cornell.edu/~mec/Winter2009/RalucaRemus/Lecture3/lecture3.html`

The Importance Rule

Each webpage has an associated importance, or **rank**. This is a positive number.

The Importance Rule

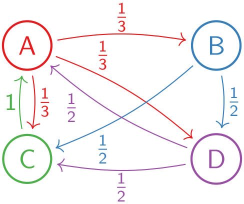
If page P links to n other pages $Q_1, Q_2, \ldots, Q_n$, then each Q_i should inherit $\frac{1}{n}$ of P's importance.

- So if a very important page links to your webpage, your webpage is considered important.
- And if a ton of unimportant pages link to your webpage, then it's still important.
- But if only one crappy site links to yours, your page isn't important.

Random surfer interpretation: a "random surfer" just sits at his computer all day, randomly clicking on links. The pages he spends the most time on should be the most important. This turns out to be equivalent to the rank.

The Importance Matrix

Consider the following Internet with only four pages. Links are indicated by arrows.



Page A has 3 links, so it passes $\frac{1}{3}$ of its importance to pages B, C, D.
Page B has 2 links, so it passes $\frac{1}{2}$ of its importance to pages C, D.
Page C has one link, so it passes all of its importance to page A.
Page D has 2 links, so it passes $\frac{1}{2}$ of its importance to pages A, C.

In terms of matrices, if $v = (a, b, c, d)$ is the vector containing the ranks a, b, c, d of the pages A, B, C, D, then

importance matrix: ij entry is importance page j passes to page i

$$\begin{pmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} c + \frac{1}{2}d \\ \frac{1}{3}a \\ \frac{1}{3}a + \frac{1}{2}b + \frac{1}{2}d \\ \frac{1}{3}a + \frac{1}{2}b \end{pmatrix} \overset{\text{Importance Rule}}{=} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}$$

The 25 Billion Dollar Eigenvector

Observations:

- The importance matrix is a stochastic matrix! The columns each contain $1/n$ ($n =$ number of links), n times.
- The rank vector is an eigenvector with eigenvalue 1!

Random surfer interpretation: If a random surfer has probability (a, b, c, d) to be on page A, B, C, D, respectively, then after clicking on a random link, the probability he'll be on each page is

$$\begin{pmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} c + \frac{1}{2}d \\ \frac{1}{3}a \\ \frac{1}{3}a + \frac{1}{2}b + \frac{1}{2}d \\ \frac{1}{3}a + \frac{1}{2}b \end{pmatrix}.$$

The rank vector is a *steady state* for the importance matrix: it's the probability vector (a, b, c, d) such that, after clicking on a random link, the random surfer will have the *same probability* of being on each page.

So, the important (high-ranked) pages are those where a random surfer will end up most often.

Note about Google Matrix for Spring 2020

Sometimes the internet can have problems (like disconnected parts or dangling pages) that prevent the importance matrix A from being positive-stochastic.

The next slide will discuss Page and Brin's solution to "fix" the importance matrix when it is not positive-stochastic, to get the Google matrix.

In the Spring 2020 semester, we will only test Google matrices where the importance matrix A is equal to the Google matrix (in the terminology of the next slide, this is the same as saying that the "damping factor" is $p = 0$).

The Google Matrix

Sometimes the internet can have problems (like disconnected parts or dangling pages) that prevent the importance matrix from being positive-stochastic. Here is Page and Brin's solution. First we fix the importance matrix A as follows: replace a column if zeros with a column of $1/N$s, where N is the number of pages.

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix} \quad \rightsquigarrow \quad A' = \begin{pmatrix} 0 & 0 & 1/3 \\ 0 & 0 & 1/3 \\ 1 & 1 & 1/3 \end{pmatrix}.$$

The **modified importance matrix** A' is always stochastic.

Now fix p in $[0,1)$, called the **damping factor**. (A typical value is $p = 0.15$.) The **Google Matrix** is

$$M = (1-p)\cdot A' + p \cdot B \quad \text{where} \quad B = \frac{1}{N}\begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix},$$

N is the total number of pages.

In the random surfer interpretation, this matrix M says: with probability p, our surfer will surf to a completely random page; otherwise, he'll click a random link. On a page with no links, he'll always surf to a completely random page.

The Google Matrix

Upshot

Lemma

The Google matrix is a positive stochastic matrix.

The PageRank vector is the steady state for the Google Matrix.

This exists and has positive entries by the Perron–Frobenius theorem. The hard part is calculating it: the Google matrix has 1 gazillion rows.

Summary

- **Stochastic** and **positive stochastic** matrices model probabilistic systems.
- We care about the long-term behavior of such a system. This is called the **steady state**. It tells us the eventual state of the system.
- The Perron–Frobenius theorem says that a positive stochastic matrix always has a unique steady state.
- If you can understand the RedBox example, then you understand almost everything.
- The Google matrix is an example of a positive stochastic matrix.
- The steady state of the Google matrix is the PageRank.