Math 2250-1
Wednesday November 19
§6.3 Applications for powers of matrices, and diagonalization

Interacting species: One example is predator-prey models (think Foxes and Rabbits)

Write $\begin{bmatrix} F_k \\ R_k \end{bmatrix}$ for predator, prey population at time $k = 0, 1, 2, \ldots$ (weeks, years, etc.) months.

Assume a constant transition matrix describes how to get $\begin{bmatrix} F_{k+1} \\ R_{k+1} \end{bmatrix}$ from $\begin{bmatrix} F_k \\ R_k \end{bmatrix}$

e.g. $F_{k+1} = .4F_k + .3R_k$ with no rabbits, $F_{k+1} = .4F_k$ (exponential decay)

$R_{k+1} = -rF_k + 1.2R_k$ with no foxes, $R_{k+1} = 1.2R_k$ (exponential growth)

(r = "predation rate")

$$\begin{bmatrix} F_{k+1} \\ R_{k+1} \end{bmatrix} = \begin{bmatrix} .4 & .3 \\ -r & 1.2 \end{bmatrix} \begin{bmatrix} F_k \\ R_k \end{bmatrix}$$

so

$$\begin{bmatrix} F_k \\ R_k \end{bmatrix} = \begin{bmatrix} .4 & .3 \\ -r & 1.2 \end{bmatrix}^k \begin{bmatrix} F_0 \\ R_0 \end{bmatrix}$$

$$\vec{x}_k = A^k \vec{x}_0$$

eigenvalue analysis:

$$|A - \lambda I| = \begin{vmatrix} .4-\lambda & .3 \\ -r & 1.2-\lambda \end{vmatrix}$$

$$= (.4-\lambda)(1.2-\lambda) + .3r$$

$$= .48 - 1.6\lambda + \lambda^2 + .3r$$

$$= \lambda^2 - 1.6\lambda + (.48 + .3r)$$

roots $\lambda = \dfrac{1.6 \pm \sqrt{2.56 - 4(.48 + .3r)}}{2}$

$$= \frac{1.6 \pm \sqrt{.64 - 1.2r}}{2}$$

r determines λ_1, λ_2.

then $P^{-1}AP = D = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$

$\uparrow$ eigenbasis

$A = PDP^{-1}$

$A = P\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}P^{-1}$

$A^k = P\begin{bmatrix} \lambda_1^k & 0 \\ 0 & \lambda_2^k \end{bmatrix}P^{-1}$

- if $|\lambda_1|, |\lambda_2| < 1$, $A^k \to 0$ population dies out
- other cases

<u>Example 1</u> $r = .4$

$$\lambda = \frac{1.6 \pm \sqrt{.64 - .48}}{2} = \frac{1.6 \pm \sqrt{.16}}{2} = \frac{1.6 \pm .4}{2} = 1, .6$$

$\lambda = 1$: $\left[\begin{array}{cc|c} -.6 & .3 & 0 \\ -.4 & .2 & 0 \end{array}\right]$ $\vec{v}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$

$\lambda = .6$: $\left[\begin{array}{cc|c} -.2 & .3 & 0 \\ -.4 & .6 & 0 \end{array}\right]$ $\vec{v}_2 = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$.

$$P = \begin{bmatrix} 1 & 3 \\ 2 & 2 \end{bmatrix}$$

(equivalent arithmetic, but done differently ~~than text page 384~~):

Suppose $\begin{bmatrix} F_0 \\ R_0 \end{bmatrix}$ is initial population vector

We can write $\begin{bmatrix} F_0 \\ R_0 \end{bmatrix} = c_1\vec{v}_1 + c_2\vec{v}_2$ $\longleftarrow$ $\begin{bmatrix} F_0 \\ R_0 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 2 & 2 \end{bmatrix}\begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$

Then $\begin{bmatrix} F_k \\ R_k \end{bmatrix} = A^k\left[c_1\vec{v}_1 + c_2\vec{v}_2\right] = c_1A^k\vec{v}_1 + c_2A^k\vec{v}_2$

$$= c_1 1^k \vec{v}_1 + c_2(.6)^k\vec{v}_2$$

$$\lim_{k\to\infty} ?_0 = c_1\vec{v}_1 = c_1\begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

<u>coexistence!</u> (twice as many rabbits as foxes)

Example 2 (r = .5) $\lambda = \frac{1.6 \pm \sqrt{.64 - 1.2r}}{2} = \frac{1.6 \pm \sqrt{.04}}{2} = \frac{1.6 \pm .2}{2} = .9, .7$

mutual extinction

Example 3 (r = .325) (yipes) you get $\lambda = 1.05, .55$

$A = \begin{bmatrix} .4 & .3 \\ -.325 & 1.2 \end{bmatrix}$

$\lambda = 1.05$

$$\left[\begin{array}{cc|c} -.650 & .3 & 0 \\ -.325 & .15 & 0 \end{array}\right]$$
$$\left[\begin{array}{cc|c} -325 & 150 & 0 \\ 0 & 0 & \end{array}\right]$$
$$\left[\begin{array}{cc|c} -13 & 6 & 0 \\ 0 & 0 & 0 \end{array}\right]$$

$\vec{v}_1 = \begin{bmatrix} 6 \\ 13 \end{bmatrix}$

$\lambda = .55$

$$\left[\begin{array}{cc|c} -.15 & .3 & 0 \\ -.325 & .65 & 0 \end{array}\right]$$
$$\left[\begin{array}{cc|c} -1 & 2 & 0 \\ 0 & 0 & 0 \end{array}\right]$$

$\vec{v}_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$

Write $\begin{bmatrix} F_0 \\ R_0 \end{bmatrix} = c_1\vec{v}_1 + c_2\vec{v}_2 = \begin{bmatrix} 6 & 2 \\ 13 & 1 \end{bmatrix}\begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$

Then $A^k \begin{bmatrix} F_0 \\ R_0 \end{bmatrix} = A^k(c_1\vec{v}_1 + c_2\vec{v}_2)$

$= c_1 A^k \vec{v}_1 + c_2 A^k \vec{v}_2 = c_1 (1.05)^k \vec{v}_1 + c_2(.55)^k \vec{v}_2$

∞ as $k \to \infty$; $\downarrow 0$ as $k \to \infty$

mutual population explosion

(in a ratio of 6:13)

(Better predator-prey modeling assumes the prey (Rabbits) grow logistically rather than exponentially, with no foxes around - see Chapter 9!)

Example 4 How google works?

try wikipedia (or a google search).

also, http://www.ams.org/featurecolumn/archive/pagerank.html :

A stochastic matrix (p 383 of our text) is an $n \times n$ matrix with all positive entries, such that the entries in each column add up to 1

e.g. $A = \begin{bmatrix} .1 & .8 & .8 \\ .8 & .1 & .1 \\ .1 & .1 & .1 \end{bmatrix}$

Perron's theorem says any Stochastic matrix has an eigenvalue $\lambda = 1$, $\lambda=1$ eigenspace dim = 1, with an eigenvector $\vec{v}$ with all positive entries. (You can scale $\vec{v}$ so that its entries add up to 1.) All other eigenvalues of a stochastic matrix have absolute value less than 1 (if they're real), or if they're complex, $\lambda = a + bi = \sqrt{a^2+b^2}\left(\frac{a}{\sqrt{a^2+b^2}} + \frac{ib}{\sqrt{a^2+b^2}}\right)$

$= \cos\theta \quad \sin\theta$

$= re^{i\theta}$ with $\underline{r < 1}$

so $\lambda^k = r^k e^{ik\theta}$

$= r^k(\cos k\theta + i \sin k\theta)$

thus, for all other λ's,

$\lambda^k \to 0$ as $k \to \infty$.

Thus, If A is stochastic, and is the transition matrix for a discrete dynamical system, then

$$\vec{x}_k = A^k \vec{x}_0 . \qquad \text{Express } \vec{x}_0 = \underset{\substack{\uparrow \\ \text{the } \lambda=1 \\ \text{eigenvector}}}{c_1\vec{v}_1} + \underbrace{c_2\vec{v}_2 + \dots + c_n\vec{v}_n}_{\text{the eigenvectors}}$$

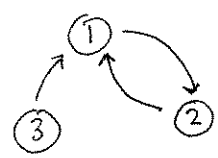
the eigenvectors (or "generalized" eigenvectors if A is not diagonalizable) with $|\lambda_i|<1$

$$\text{Then } A^k\vec{x}_0 = c_1\vec{v}_1 + c_2 \underbrace{A^k\vec{v}_2}_{\lambda_2^k\vec{v}_2 \to 0} + c_3 \underbrace{A^k\vec{v}_3}_{\to 0} + \dots \to 0 \quad \text{as } k\to\infty.$$

So the long-term dynamics are determined by the $\lambda=1$ eigenvector.
(Like in Tuesday's example.)

The google voting game:

Suppose that for a certain search topic there a 3 sites, and that they link (i.e. with hyperlinks) to each other as follows



Which site should be top ranked?

Initially give each site $1/3$ vote, so $\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$ is # of votes for sites 1, 2, 3.

to get from stage k to stage $k+1$ each site splits its current votes equally among the sites it links to:
So, in our case

votes for 1 $\longrightarrow$ $x_{k+1} = 0\,x_k + 1\,y_k + 1\,z_k$

" 2 $\longrightarrow$ $y_{k+1} = 1\,x_k + 0\,y_k + 0\,z_k$

" 3 $\longrightarrow$ $z_{k+1} = 0\,x_k + 0\,y_k + 0\,z_k$

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix}$$

Call this transition matrix B

how site 1 vote is split; how site 2 vote is split; how site 3 vote is split

- column entries always add to 1
- but not all entries are non-zero.

google fudge factor: if there are n sites, and for $\alpha = .15$, make everyone give $\frac{\alpha}{n}$ of their current vote to each site equally, and split the remaining fraction of their vote according to the original recipe.

$$\text{so} \quad \begin{bmatrix} x_{k+1} \\ y_{k+1} \\ z_{k+1} \end{bmatrix} = (1-\alpha)\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \frac{\alpha}{3}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} .1 & .8 & .8 \\ .8 & .1 & .1 \\ .1 & .1 & .1 \end{bmatrix} \quad (\text{for } \alpha = .3)$$

Stochastic!

With the google fudge factor

$$\begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} = \begin{bmatrix} .1 & .8 & .8 \\ .8 & .1 & .1 \\ .1 & .1 & .1 \end{bmatrix}^k \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \longrightarrow c_1 \vec{v}_1 \quad \text{as } k \to \infty$$

the $\lambda = 1$ eigenvector

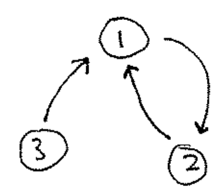
Since total vote is conserved, if $\vec{v}_1$ is the vector with positive entries so that their sum is 1, then since the original vote totalled 1, our answer will actually be $\vec{v}_1$. In fact, we could have apportioned the initial vote however we wanted — as long as the initial vote totalled 1, the limit is $\vec{v}_1$. (So if we initially gave all the vote to site j, which is an initial vote vector $\vec{e}_j = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ \vdots \end{bmatrix} \leftarrow$ entry j, the limit of $A^k e_j$ is $\vec{v}_1$, which is also the j^{th} column of the limit matrix!

The i^{th} entry of $\vec{v}_1$ is how much vote site i ends up with — the more this number is, the higher the google rank!

```
> with(linalg) :
> Digits := 5 :
> A := matrix(3, 3, [.1, .8, .8, .8, .1, .1, .1, .1, .1]);
  x := matrix(3, 1, [1/3., 1/3., 1/3.]);
```

$$A := \begin{bmatrix} 0.1 & 0.8 & 0.8 \\ 0.8 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \end{bmatrix}$$

$$x := \begin{bmatrix} 0.33333 \\ 0.33333 \\ 0.33333 \end{bmatrix}$$



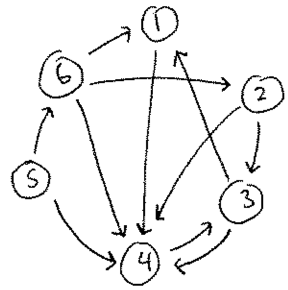

```
> evalm(A^40 &* x);
  evalm(A^40);
```

$$\begin{array}{ll} \begin{bmatrix} 0.47058 \\ 0.42942 \\ 0.099999 \end{bmatrix} & \begin{array}{l} \leftarrow \text{site 1 is tops} \\ \\ \leftarrow \text{site 3 is weak} \end{array} \end{array}$$

$$\begin{bmatrix} 0.47059 & 0.47059 & 0.47059 \\ 0.42941 & 0.42941 & 0.42941 \\ 0.10000 & 0.10000 & 0.10000 \end{bmatrix}$$

```
>
```

A bigger example



$$\begin{array}{cc} & \begin{array}{cccccc} \text{site } 1 & 2 & 3 & 4 & 5 & 6 \\ \downarrow & \downarrow & & & & \end{array} \\ B = & \begin{bmatrix} 0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & 1 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} & 0 & \frac{1}{2} & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \end{bmatrix} \end{array}$$

$\alpha = .15$

$$B := \begin{bmatrix} 0 & 0 & 0.5 & 0 & 0 & 0.33333 \\ 0 & 0 & 0 & 0 & 0 & 0.33333 \\ 0 & 0.5 & 0 & 1 & 0 & 0 \\ 1 & 0.5 & 0.5 & 0 & 0.5 & 0.33333 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 \end{bmatrix}$$

$$goog := \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

> alpha := .15 : n := 6 :
$A := evalm\left((1 - \text{alpha}) \cdot B + \frac{\text{alpha}}{n} \cdot goog\right);$

$$A := \begin{bmatrix} 0.025000 & 0.025000 & 0.45000 & 0.025000 & 0.025000 & 0.30833 \\ 0.025000 & 0.025000 & 0.025000 & 0.025000 & 0.025000 & 0.30833 \\ 0.025000 & 0.45000 & 0.025000 & 0.87500 & 0.025000 & 0.025000 \\ 0.87500 & 0.45000 & 0.45000 & 0.025000 & 0.45000 & 0.30833 \\ 0.025000 & 0.025000 & 0.025000 & 0.025000 & 0.025000 & 0.025000 \\ 0.025000 & 0.025000 & 0.025000 & 0.025000 & 0.45000 & 0.025000 \end{bmatrix}$$

> $evalm(A^{40});$

$$\begin{bmatrix} 0.18476 & 0.18476 & 0.18477 & 0.18476 & 0.18476 & 0.18477 \\ 0.035090 & 0.035090 & 0.035092 & 0.035091 & 0.035090 & 0.035092 \\ 0.35216 & 0.35216 & 0.35217 & 0.35217 & 0.35216 & 0.35217 \\ 0.36736 & 0.36735 & 0.36738 & 0.36736 & 0.36736 & 0.36738 \\ 0.025001 & 0.025001 & 0.025002 & 0.025001 & 0.025001 & 0.025002 \\ 0.035626 & 0.035626 & 0.035627 & 0.035626 & 0.035626 & 0.035627 \end{bmatrix}$$

← How do you google rank the 6 sites?

>

other applications:
any enclosed tank system!
other rankings?
(e.g. BCS football?)