Math 223, Lecture 27

Last time: diagonalization, idea of working in a basis of eigenvectors for a linear map

For example, analyzed "power method" for finding eigenvectors

HW: applied this idea to solve recurrence relations.

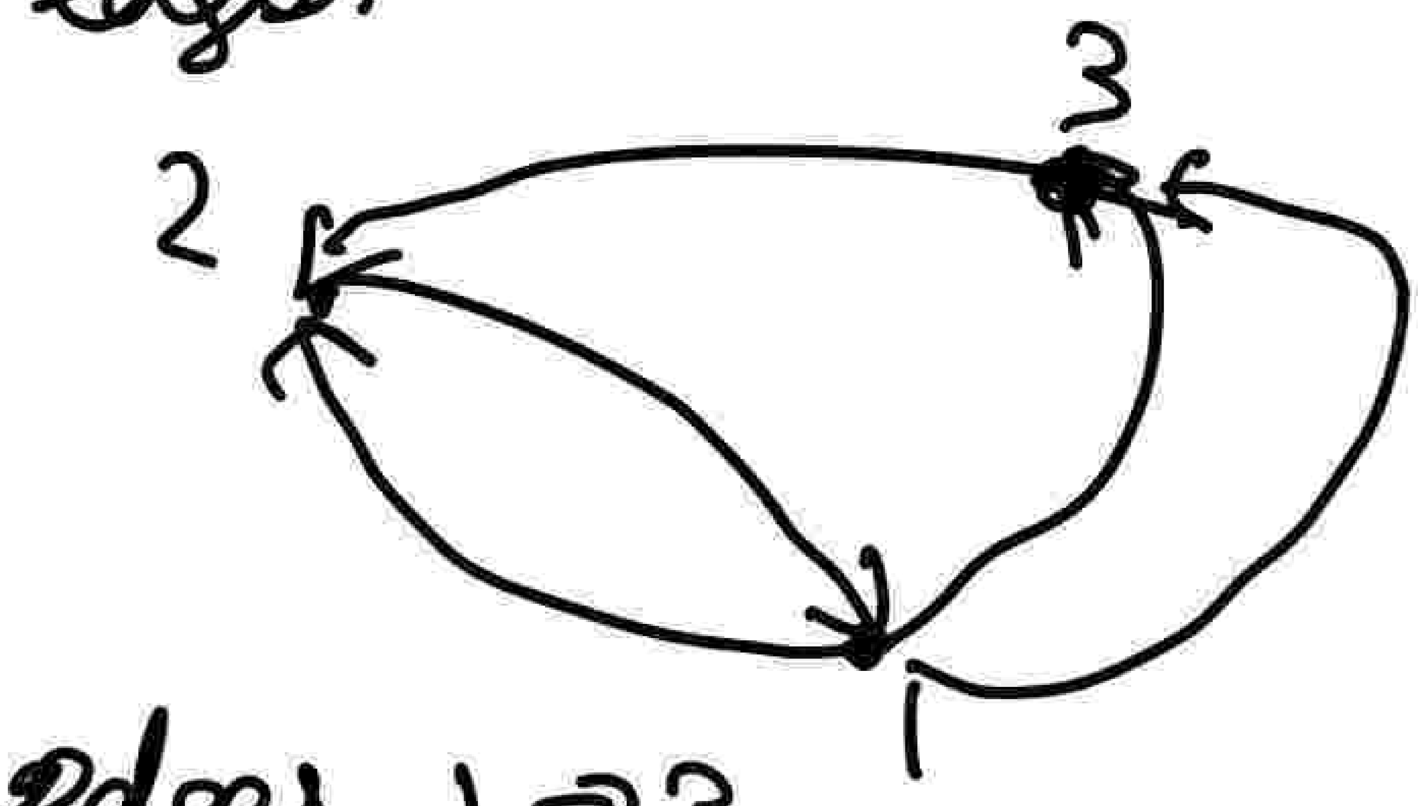
$$\left(F_n = \frac{\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n}{\sqrt{5}} \right)$$

Today: Applications to paths & walks on graphs.

(+ Google PageRank)

Def: A directed graph is a pair (V, E) where V is a set (of "vertices"), E is a set (of "edges") so that each edge connects two vertices (its head and tail).

Pictorially, draw points for vertices, arrows for edges:



have two edges $1 \to 3$
edges $1 \to 2$, $2 \to 1$, $3 \to 2$

(also exist <u>undirected</u> graphs)

(The WWW is a directed graph: vertices are the <u>pages</u>, edges are the <u>hyperlinks</u>

<u>Question</u>: How many paths of length n are there in the graph?

(e.g. paths of length 1 = edges)

<u>Def</u>: The <u>adjacency matrix</u> of G is the matrix

$A = A_G$ where $A_{u,v}$ = # edges from u to v

$\in M_{V \times V}(\mathbb{R})$

<u>Claim:</u> # of paths of length n from u to v is $(A^n)_{u,v}$.

(aside: if $f: V \to \mathbb{R}$, $(A_G f)(u) = \sum_{u \to v} f(v)$)
so A_G is a linear map on $\mathbb{R}^V$.

<u>Pf:</u> for $n=0$ have a path of length 0 ~~for~~ from u to u, no such path from u to v if $u \neq v$.

So A^0 counts paths of length 0.

By induction: $\{$paths of length $n+1$ from u to $v\}$

$$= \bigcup_{\substack{\text{edges} \\ u \to w}} \left\{ \begin{array}{l} \text{path of length } n \\ \text{from } w \text{ to } v \end{array} \right\}$$

$$= (\#\text{paths of length } n+1) = \sum_{u \to w} \left(\#\text{ paths length } n \; w \to v\right)$$

$$= \sum_{w} \#\text{of edges}(u \to w) \cdot (\# \quad)$$

$$= \left(A_G \cdot \underset{\text{of length } n}{\text{Path counts}}\right)_{uv} = (A_G \cdot A_G^{n})_{uv}$$
$$= (A_G^{n+1})_{uv}.$$

$\blacksquare$

<u>Cor</u>: If λ is the largest eigenvalue of A_G then #paths $\sim c \cdot \lambda^n$.

(assume graph is irreducible: can get from any u to any v)

<u>Thm</u> (Perron-Frobenius) then A_G has a unique largest eigenvalue (in magnitude) λ, $\lambda > 0$, and its eigenvector has positive entries

Then if $A_G = SDS^{-1}$ can check that all entries of A_G^n involve λ^n.

$$(A_G)^n = SD^nS^{-1}$$

(Power method: dominant contribution is λ^n)

<u>Aside</u>: can show path counts satisfy recursion relation

Def A Markov chain is a directed graph + probability on each edge, s.t. for all u,

$$\sum_{u \to v} p(u \to v) = 1$$

sum over edges leaving u

(can model natural language: vertices are words, edges: prob next word is.)

Natural matrix: A_{uv} = prob of leaving u to v.

row vector $\pi = (\pi_u)_{u \in V}$ interpret as a prob distr. on V

Then $(\pi A)_v = \sum_u \pi_u A_{uv}$ = prob. of ending at v.

$\Rightarrow$ $(\pi \cdot A^n)_v$ = prob of being at v after n steps if at time 0 we are distributed a-la π.

$$\underset{\pi}{\begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{pmatrix}} \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \end{pmatrix} = \begin{pmatrix} \frac{1}{6}, & \frac{1}{4}, & \frac{7}{12} \end{pmatrix}$$

$M_{1,3}(\mathbb{R})$ $\qquad \begin{cases} M_{3,3}(\mathbb{R}) \end{cases}$ $\qquad M_{1,3}(\mathbb{R})$

col = prob of reaching v
row = " " leaving u

checks on right $\qquad A\begin{pmatrix}1\\ \vdots\\ 1\end{pmatrix} = \begin{bmatrix}1\\ \vdots\\ 1\end{bmatrix}$

(exit prob of u add to 1)

So $\lambda = 1$ is an eigenvalue with positive eigenvector $\Rightarrow$ it's the Perron-Frobenius eigenvalue

($\det A^T = \det A \Rightarrow p_{A^T} = p_A \Rightarrow$ same ev.

$\Rightarrow \lambda = 1$ is also the PF eigenvalue of A^T]

Interpretation: The positive eigenvector π_0 is the unique stationary distribution $\pi_0 A = \pi_0$

Power method: for any vector, $\underbrace{\pi A^n}_{\text{normalise}} \to \pi_0$

(no matter where we start, if we take many steps, we are roughly distributed by stationary distribution

Google idea: people link to useful pages so a random walk on the web graph will spend more time on useful pages

Pagerank of page u = value of stationary distribution at u