Brin Stuck

Internet Search

crawlers gathering of information

indexer. parse & code raw info ⇒ produces a set of word occurrences for each webpage

⇒ forward index

Document	Words
1	the, cat
2	says, love

sorter: rearrange info by words ⇒ inverted index

word	document
the	1 3 7 ...
says	2 4 ----

searcher: uses inverted index ⇒ list of documents relevant to the key words

order of the list ⇐ relevance of the document to the query ← relative position, fontification, frequency of key words

PageRank of the webpage ← google use Markov chain to rank webpages

Brin Stuck

Internet Search

$G = (V, E)$ directed graph ← the collection of all web pages and links between them

V → webpages, E → links

$\tilde{G}$: add vertex 0. with edges to and fro all other vertices $v_1, v_2, \cdots v_N$

$b_{ij} = 1$ iff $\exists$ edge from i to j in $\tilde{G}$

$o(i)$: # outgoing edges from i. in $\tilde{G}$ ($o(i) > 0$ for all i)

$p \in (0,1)$: damping parameter

For $i = 0$. set $B_{ii} = 0$.

For. $i \neq j$ $i, j > 0$. set $B_{0i} = \frac{1}{N}$, $B_{i0} = \begin{cases} 1 & \text{if } o(i) = 1 \\ 1-p & \text{if } o(i) \neq 1 \end{cases}$

$$B_{ij} = \begin{cases} 0 & \text{if } b_{ij} = 0 \\ p/o(i) & \text{if } b_{ij} = 1 \end{cases}$$

what's the meaning of B →

B is stochastic and primitive by Corollary 3.3.3. B has a unique positive left eigenvector q with eigenvalue 1, whose entries add up to 1.

verify B is primitive

pair (B, q) is a Markov chain on the vertices of $\tilde{G}$

q: initial prob. distribution. B: transition prob. $Bq = q$

Google interprets q_i as the PageRank of webpage i. ?

why q_i serves as pagerank

For any initial prob. distribution q' on vertices of $\tilde{G}$.

the sequence $q'B^n$ converges exponentially to q. ?

thus one can find approx. for q by computing pB^n

(q': uniform distribution) ?

what's the exact algorithm.

$W_1, W_2, \cdots\cdot W_N$: web pages.

W_0: restart page

$C(i)$: total # of links contained in page i (outgoing degree)

P_{ij}: the prob. of entering page j given the current page is i $\quad \sum_j P_{ij} = 1$ for any i

d: $0 < d < 1$ how likely you will restart your navigation by not following links in pages

$P_{i0} = 1-d$: Every state $0 \le i \le N$ has prob. $1-d$ transiting to the restart state 0

$P_{0i} = d/N$ for $i \neq 0$: from restart state, the prob. of going to any page is equal

$\Rightarrow$ total prob of going to a real page is d

$P_{00} = P_{i0} = 1-d$: prob. of restart again

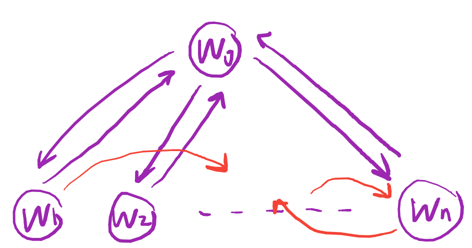
$P_{ij} = \frac{d}{C(i)} \cdot I_{\{i \text{ links to } j\}} = \begin{cases} 0 & \text{if } i \text{ does not link to } j \\ d/C(i) & \text{if } i \text{ links to } j \end{cases}$

⇑

For each page. besides prob. $1-d$ going to restart the rest prob. d is evenly divided among the $C(i)$ links contained in it

B is primitive :

$$\begin{bmatrix} 0 & 1-d & \cdots & 1 & \cdots & 1-d \\ 1/n & 0 & & 0 & & d/o(n) \\ \vdots & d/o(2) & \ddots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & & d/o(n) \\ 1/n & d/o(2) & & 0 & & 0 \end{bmatrix}$$



↑

$O(2) = 0$

Page Rank.

Google Search

$25 billion eigenvector

$A = (a_{tj})$ a_{tj}: # references from website W_j to W_t

normalize column: $a_{tj} \to a_{tj} / \sum_t a_{tj}$ $\Rightarrow A = [a_1, a_2, \cdots a_n]$

Define $B = [b_1, b_2, \cdots b_n]$ $\begin{cases} \text{if } a_j = 0. & b_j = [\frac{1}{n}, \cdots \frac{1}{n}]^T \\ \text{if } a_j \neq 0 & b_j = 0 \end{cases}$

$\Downarrow$

$A + B$ a stochastic matrix (every column adds to 1)

Markov chains

Perron Frobenius

Algorithm. eigenvector find.

COVID.

$\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$

1. Let $C := x(A+B) + (1-x) Q$ where Q random stochastic matrix x with $0 < x < 1$

$\Downarrow$

C is primitive when x is near 1. $\Rightarrow \rho(C) = 1 \in \Lambda(C)$

$\Downarrow$

Find Perron vector v. associated with 1. v_i gives ranking

Pick random vector u. $\lim_{k \to \infty} C^k u = w$ $\Rightarrow$ normalize $w \Rightarrow v$

2. Take matrix $O = (o_{tj})$ with $o_{tj} = \frac{1}{n}$ for all t, j.

$D = x(A+B) + (1-x) O$, $u = (\frac{1}{n}, \cdots \frac{1}{n})^T$

Let $u_1 = u$ and $u_{m+1} = D u_m = x(A+B) u_m + (1-x) O u_m$ $O u_m = u$ since u_m's are stochastic

$= xA u_m + xB u_m + (1-x) u$

precompute xA, xB

then $xA u_m$, $xB u_m = (a, \cdots a)^T$ with $a = \frac{x}{n(\sum_{tj} u_{tj})}$

Perron-Frobenius: $A > 0$. stochastic $\Rightarrow$ $\exists$ unique x s.t. $Ax = x$.

$A^k x_0 \to x$ $(k \to \infty)$ for any initial x_0

Markov Chain.

Google PageRank

Slides

Idea 1: x_k = # of links to page k

$\ddot{\frown}$: a link from an "important" page should carry more weight

Idea 2: x_k = Σ of important scores of pages linking to page k

$\ddot{\frown}$: a page is more important if it has more outgoing links

Idea 3: $x_k = \Sigma\, x_j / n_j$ $\Downarrow$

★ $j \to i$

n webpages

n_j : outgoing links from page j $\Rightarrow$ $A = (a_{ij})$ $\quad a_{ij} = \begin{cases} 1/n_j & \text{if page } j \text{ links to page } i \\ 0 & \text{otherwise} \end{cases}$

$\Downarrow$

A is stochastic $\quad A \geq 0$. $\quad Ax = x$ existence ?

Let $C = (C_{ij})$ $\quad C_{ij} = \frac{1}{n}$ for all i, j $\Rightarrow$ $B := 0.85A + 0.15C$ $\Rightarrow$ $\exists\, Bx = x$

★ Page Rank : a model of user behavior.

A random surfer is given a web page at random and keeps clicking on links.

never hitting "back" but eventually gets bored and starts on another random page.

$B = 0.85A + 0.15C$: surfer clicks a link on the current page with prob. 0.85.
opens up a random page with prob. 0.15

$\Rightarrow$ 选出所有符合基本条件的网页

如果没有ranking，用户要在它们之间随机浏览.

某些网页因被link的次数多，就会频繁地被浏览

page rank q : the prob. that the surfer will end up on that page

the fraction of time the random user spends on that page in the long run

Interpretation of Google Page Rank

Page ranks are proportional to the stationary prob. of the states in the Markov chain

Wander around the web pages randomly, after long time. the prob. of visiting any page at any time converges → q. not affected by the initial navigation

Set up

$w_1, w_2, \dots, w_N$: web pages.

w_0 : restart page

$C(i)$: total # of links contained in page i (outgoing degree)

P_{ij} : the prob. of entering page j given the current page is i $\quad \sum_j P_{ij} = 1$ for any i

d : $0<d<1$ how likely you will restart your navigation by not following links in pages

$P_{i0} = 1-d$: Every state $0 \le i \le N$ has prob. $1-d$ transiting to the restart state 0

$P_{0i} = d/N$ for $i \ne 0$: from restart state, the prob. of going to any page is equal

⇒ total prob of going to a real page is d

$P_{00} = P_{i0} = 1-d$: prob. of restart again

$$P_{ij} = \frac{d}{C(i)} \cdot I_{\{i \text{ links to } j\}} = \begin{cases} 0 & \text{if } i \text{ does not link to } j \\ d/C(i) & \text{if } i \text{ links to } j \end{cases}$$

For each page. besides prob. $1-d$ going to restart the rest prob. d is evenly divided among the $C(i)$ links contained in it

Let the stationary prob. of state i be π_i

then
$$\begin{cases} \pi_i = \sum_{j=0}^{N} \pi_j P_{j,i} \quad i = 0, 1, \dots N & (q = Aq) \\ \sum_{i=0}^{N} \pi_i = 1 & (q \text{ sums to } 1) \end{cases}$$

Specifically, $\pi_0 = \sum_{j=0}^{N} \pi_j P_{j,0} = \sum_{j=0}^{N} \pi_j (1-d) = 1-d$

$$\pi_i = \pi_0 P_{0i} + \sum_{j=1}^{N} \pi_j P_{j,i} = (1-d)\cdot \frac{d}{N} + \sum_{\substack{j \\ i \text{ linked to } j}} \pi_j \frac{d}{C(j)}$$

$\Downarrow \times \frac{N}{d}$ meaning ?

Page Rank: $PR(i) = \frac{N}{d}\pi_i = (1-d) + \sum_{\substack{j \\ i \text{ linked to } j}} PR(j) \frac{d}{C(j)}$ $\qquad q_i = A_{i,:} \cdot q = a_0 q_0 + \sum_{j=1}^{N} a_{ij} q_j$

Perron. $A \in M_n$. $A > 0$. then.

Matrix Analysis

① $\rho(A) > 0$. ② $\rho(A)$ is an algebraically simple eigenvalue of A.

③ $\exists$ unique $x \in \mathbb{R}^n$ s.t. $Ax = \rho(A)x$ and $\sum_i x_i = 1$. $x > 0$.

④ $\exists$ unique. $y \in \mathbb{R}^n$ s.t. $y^T A = \rho(A) y^T$ and $\sum_i x_i y_i = 1$. $y > 0$.

⑤ $|\lambda| < \rho(A)$ for every eigenvalue. $\lambda \neq \rho(A)$

⑥ $\left(\frac{1}{\rho(A)} A\right)^m \to xy^T$ as $m \to \infty$

Frobenius: $A \in M_n$. $A \geq 0$

① $\rho(A)$ is an eigenvalue of A. $\exists\ 0 \neq x \geq 0$ s.t. $Ax = \rho(A)x$

② If $\exists$ some $x > 0$ and $\lambda \geq 0$. s.t. either $Ax = \lambda x$ or $x^T A = \lambda x^T$, then $\lambda = \rho(A)$

Chapter 11

11.3 PageRank Algorithm

naive : rank sites by # incoming hyperlinks
(website should be more important if is visited more often.)

⇒ After sufficiently many steps. the website can be ranked by how many times they were visited.

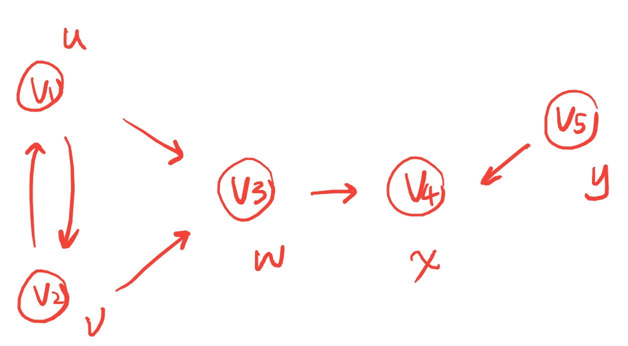
the walk is directed ⇒ random surfer can get stuck in sinks (nodes without out going edges)

Googl Matrix : Let W be random surfer matrix. $\alpha \in (0,1)$
R is a matrix with all entries $= \frac{1}{n}$

$$\Rightarrow M = \alpha \cdot W + (1-\alpha) \cdot R$$

in every step, with Prob. $1-\alpha$. the random surfer gets bored and surfs to a new random site.
α is tipically 0.85

eg.



intuition: V_4 should be more important than V_3

use transverse adjacent matrix

$$W = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad We = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 2 \\ 0 \end{bmatrix}$$

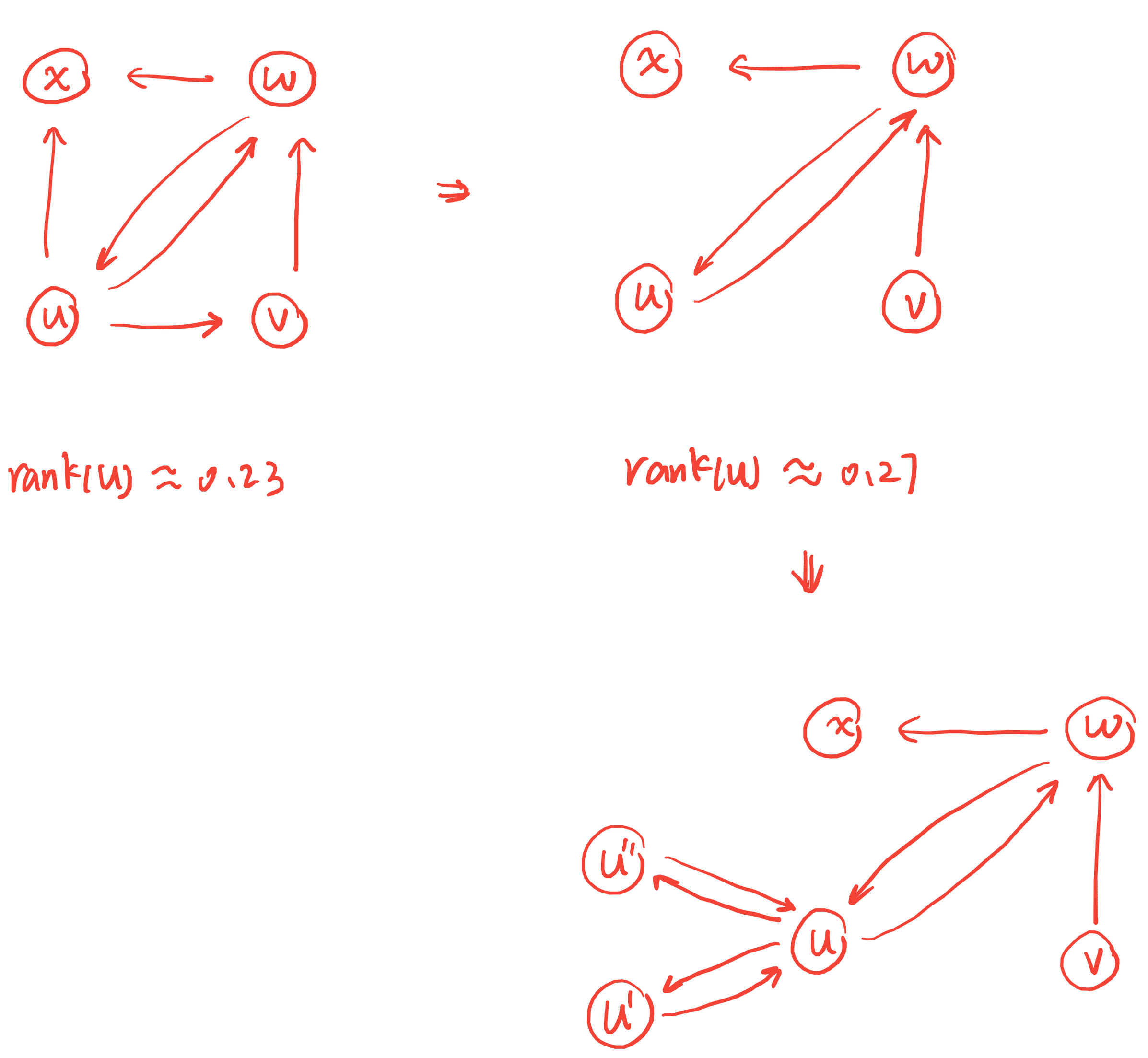
incoming links ⇒ $V_3 = V_4$

⇒ rank V_3 = rank V_4

$M = 0.85\,W + 0.15\,R$ ⇒ Page Rank

V_1	0.153846
V_2	0.153846
V_3	0.230769
V_4	0.384615
V_5	0.0769231

Attacks:



rank(u) ≈ 0.23 ⇒ rank(u) ≈ 0.27 ⇓ rank(u) ≈ 0.41

(discrete time) Markov chain: sequence of random variable $X_0, X_1, \cdots \in S$ that satisfies the Markov property

$$P[X_{t+1} = s_{t+1} \mid X_0 = s_0, X_1 = s_1, \cdots X_t = s_t] = P[X_{t+1} = s_{t+1} \mid X_t = s_t]$$

time homogeneous : $P[X_{t+1} = s_{t+1} \mid X_t = s_t]$ is independent of t

Applications of Perron Frobenius Thm:

I. The 3-point numerical PDE.

heat conduction problem: $u_t = \alpha u_{xx}$ $\quad a < x < b$

$\Rightarrow u^k = A^k u^0$ where $A = \begin{bmatrix} 1-2h & h & & \\ h & 1-2h & \ddots & \\ & \ddots & \ddots & h \\ & & h & 1-2h \end{bmatrix}$ $\quad h = \alpha \frac{\Delta t}{\Delta x^2}$

if $0 < h < 0.5$. $A \geq 0$. $\Rightarrow \rho(A) \in \Lambda(A)$ with $x \geq 0$. $Ax = \rho(A)x$

normalize $x \Rightarrow v. = x / \max_i \{|x_i|\}$ with $v_j = 1$.

$Av = \rho v \Rightarrow \rho = h v_{j-1} + (1-2h) + h v_{j+1} < 1 \Rightarrow u^k = A^k u^0$ converges.

II. Population Model

Algorithms for Perron Vector

For $A > 0$:

Generalized eigenvector : $v, (A-\lambda I)v, \cdots (A-\lambda I)^{k-1}v$ and $(A-\lambda I)^k v = 0$

linearly independent

generalize eigenspace: $\{v : (A-\lambda I)^p v = 0$ for some $p\}$

$$(A-\lambda I)^{k-1}v, \ (A-\lambda I)^{k-2}v, \cdots (A-\lambda I)v, \ v$$

$$(A-\lambda I)^{m-1}w, \ (A-\lambda I)^{m-2}w, \cdots (A-\lambda I)w, \ w$$

$$\vdots$$

$$A\cdot(A-\lambda I)^p = \lambda\cdot(A-\lambda I)^p + (A-\lambda I)^{p+1} \iff Aw_t = \lambda w_t + w_{t+1}$$

Choose a basis consisting of generalized eigenvectors $\{w_1, w_2, \cdots w_n\}$

$Aw_1 = w_1$, $Aw_t = \lambda_j w_t$ or $Aw_t = \lambda_j w_t + w_{t+1}$ with $|\lambda_j| < 1$

$\Rightarrow \lim_{k\to\infty} A^k w_1 = w_1$. $\lim_{k\to\infty} A^k w_t = 0$ for all $t > 1$

$$A^2 w_t = A\cdot Aw_t = \lambda^2 w_t + \lambda w_{t+1} + w_{t+2}$$

$$\vdots$$

$$A^k w_t = \lambda^k w_t + \lambda^{k-1} w_{t+1} + \cdots + \lambda^q w_{t+p}$$

For any $x_0 = a_1w_1 + a_2w_2 + \cdots + a_nw_n$. $(a_1 \neq 0)$

$\lim_{k\to\infty} A^k x_0 = a_1 w_1 \Rightarrow$ normalize to get Perron vector w_1

$\Rightarrow$

$$\boxed{\begin{array}{l}\text{Compute Perron Vector:} \\ x^{(0)} \text{ arbitrary. } x^{(0)} > 0. \ \sum_i x_i^{(0)} = 1 \\ y^{(m+1)} = Ax^{(m)} ; \ x^{(m+1)} = \dfrac{y^{(m+1)}}{\sum_{i=1}^{n} y_i^{(m+1)}}\end{array}}$$

if A is stochastic. $Ax = \rho(A)x$

$$\Rightarrow \sum_{k=1}^{n} a_{ik} x_k = \rho(A) x_i$$

$$\sum_{i=1}^{n}\sum_{k=1}^{n} a_{ik}x_k = \sum_{i=1}^{n} \rho(A) x_i$$

$$\sum_{k=1}^{n} x_k \sum_{i=1}^{n} a_{ik} = \rho(A)\sum_{i=1}^{n} x_i$$

$$\sum_{k=1}^{n} x_k \cdot 1 = \rho(A)\sum_{i=1}^{n} x_i \Rightarrow \rho(A) = 1$$

Faster algorithm to find Perron vector?

if $\sum_{k=1}^{n} x_k = 1$. then. $\sum_{i=1}^{n} \underbrace{\sum_{k=1}^{n} a_{ik}x_k}_{(Ax)_i} = \sum_{k=1}^{n} x_k \sum_{i=1}^{n} a_{ik} = \sum_{k=1}^{n} x_k = 1$

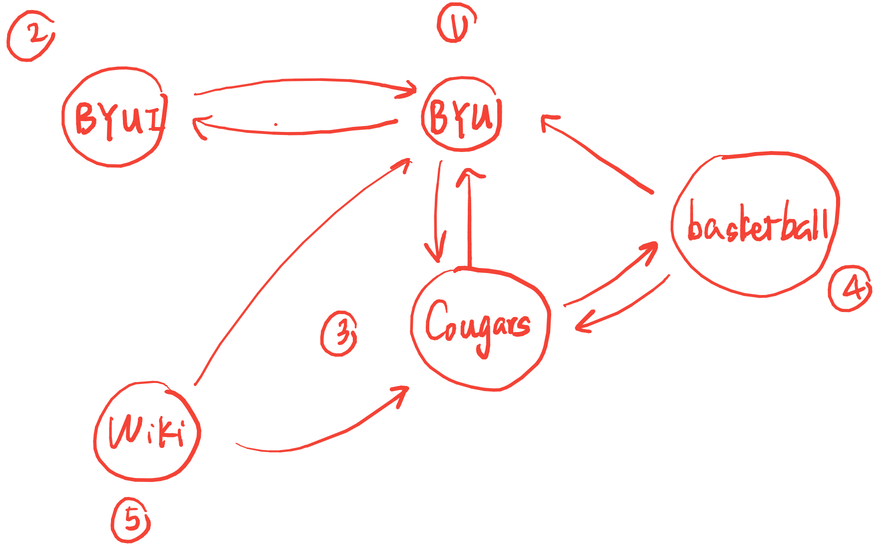
$$x^{(0)} = c_1 q_1 + c_2 q_2 + \cdots + c_n q_n.$$

$$\sum_{i=1}^{n} x_i^{(0)} = \sum_{i=1}^{n} (c_1 q_i^1 + c_2 q_i^2 + \cdots + c_n q_i^n) = c_1 \sum_{i=1}^{n} q_i^1 + c_2 \sum_{i=1}^{n} q_i^2 + \cdots + c_n \sum_{i=1}^{n} q_i^n$$

For $A \geq 0$:

Algorithm (Wiki): $b_{k+1} = \dfrac{Ab_k}{\|Ab_k\|}$

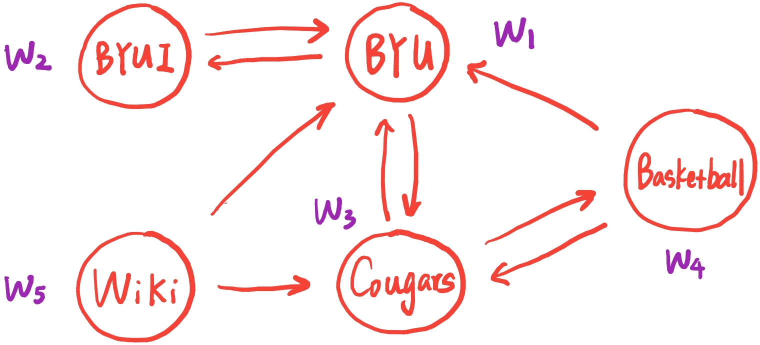
if $\rho(A) \in \Lambda(A)$ and b_0 has nozero component in w_1 direction $\Rightarrow$ a subsequence of $\{b_k\}$ converges to w_1



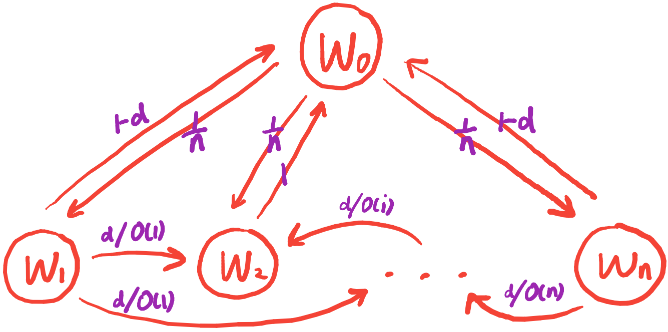
$$e = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \qquad A = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$Ae = \begin{bmatrix} 4 \\ 1 \\ 3 \\ 1 \\ 0 \end{bmatrix}$$ # incoming edges

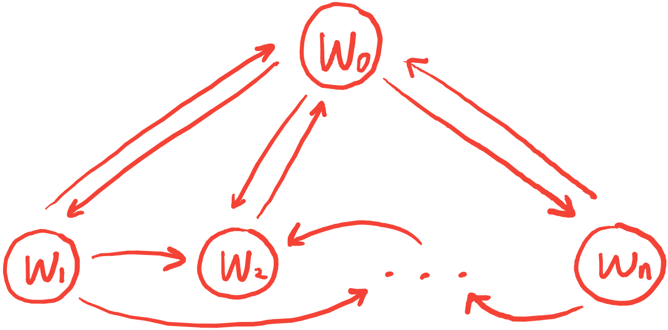
$$A^2 e = A \cdot Ae = \begin{bmatrix} 5 \\ 4 \\ 5 \\ 3 \\ 0 \end{bmatrix}$$ count "hidden" incoming edges



BYUI
BYU
W1
W2
Basketball
W3
W5
Wiki
Cougars
W4



W0
W1
W2
Wn
1-d
1/n
1/n
1
1/n
1-d
d/O(1)
d/O(1)
d/O(i)
d/O(n)
. . .



W_0
W_1
W_2
W_n
...