

Vòng retry chạy mãi

Request có hai trạng thái OK và LỖI. Về lâu dài, bao nhiêu % thời gian ở OK?



$$M = \begin{bmatrix} 0.9 & 0.2 \\ 0.1 & 0.8 \end{bmatrix}$$

cột = trạng thái HIỆN TẠI

$v_0 = (1, 0)$, lặp 200 lần

$\Rightarrow v = (0.6667, 0.3333)$

$M \cdot v - v = (0, 0)$

Đổi v_0 thành $(0, 1)$, $(0.5, 0.5)$, $(0.13, 0.87)$: vẫn ra $(2/3, 1/3)$.

$\text{eigvals}(M) = [1, 0.7]$.

Ba quan sát = cả bài 8

1. Có một vector mà $M \cdot v = v$ (đó là $\lambda = 1$).
2. Xuất phát từ đâu cũng trôi về nó.
3. $\lambda_2 = 0.7$ quyết định trôi NHANH cỡ nào.

Bài 7 dùng $[[0.9, 0.5], [0.1, 0.5]]$, $\pi = (5/6, 1/6)$. Ở đây cố ý đổi số: ra $(5/6, 1/6)$ ở đây là bạn đang chép, không tính.

Ma trận Markov: mỗi CỘT tổng 1

Vuông, mọi phần tử ≥ 0 , mỗi CỘT cộng lại = 1.

$M = [[0.9, 0.2], [0.1, 0.8]]$: cột 1: $0.9 + 0.1 = 1$, cột 2: $0.2 + 0.8 = 1$

Cột j = "đang ở j , bước sau đi đâu, xác suất bao nhiêu".

Cột 1: đang OK $\rightarrow$ 90% ở lại OK, 10% rơi sang LỖI.

Cột 2: đang LỖI $\rightarrow$ 20% hồi phục, 80% vẫn LỖI.

Cột tổng 1 vì đi đâu đó phải bằng 100%: không request nào bốc hơi.

Tổng luôn được bảo toàn

v là phân phối (≥ 0 , tổng 1) $\Rightarrow M \cdot v$ cũng là phân phối.

Lặp bao lâu tổng cũng = 1.0000000000: không trùng hợp, là tính chất cột.

Lệch tổng = bạn cài sai. Cách bắt bug rẻ nhất.

☆ Backend: một vòng retry chạy mãi. M = bảng xác suất chuyển trạng thái, v = phân bố mọi request; nhân M = chạy thêm một vòng.

Bẫy quy ước: nửa thế giới viết theo HÀNG ($v \cdot M$).

Bài này và Taubes dùng CỘT ($M \cdot v$). Đọc tài liệu khác: việc đầu tiên là kiểm $M.\text{sum}(\text{axis}=0)$ hay $M.\text{sum}(\text{axis}=1)$ bằng 1. Sai quy ước thì code vẫn chạy, mọi số sau đó đều sai.

Phân phối dùng LÀ eigenvector $\lambda = 1$

$$M \cdot \pi = \pi$$

Bài 7: $A \cdot v = \lambda \cdot v$. Đặt $\lambda = 1$: hai phương trình là MỘT.

Không "giống" eigenvector. Nó LÀ eigenvector, ứng với eigenvalue đúng bằng 1.

$$\begin{aligned} M \cdot (2/3, 1/3) &= (0.9 \cdot 2/3 + 0.2 \cdot 1/3, 0.1 \cdot 2/3 + 0.8 \cdot 1/3) \\ &= (0.6 + 0.0667, 0.0667 + 0.2667) = (2/3, 1/3) \checkmark \text{ đúng yên} \end{aligned}$$

Mẹo 2×2 (3B1B tập 15), trên giấy

$$m = \text{trace}/2 = 1.7/2 = 0.85, \quad p = \det = 0.9 \cdot 0.8 - 0.2 \cdot 0.1 = 0.7$$

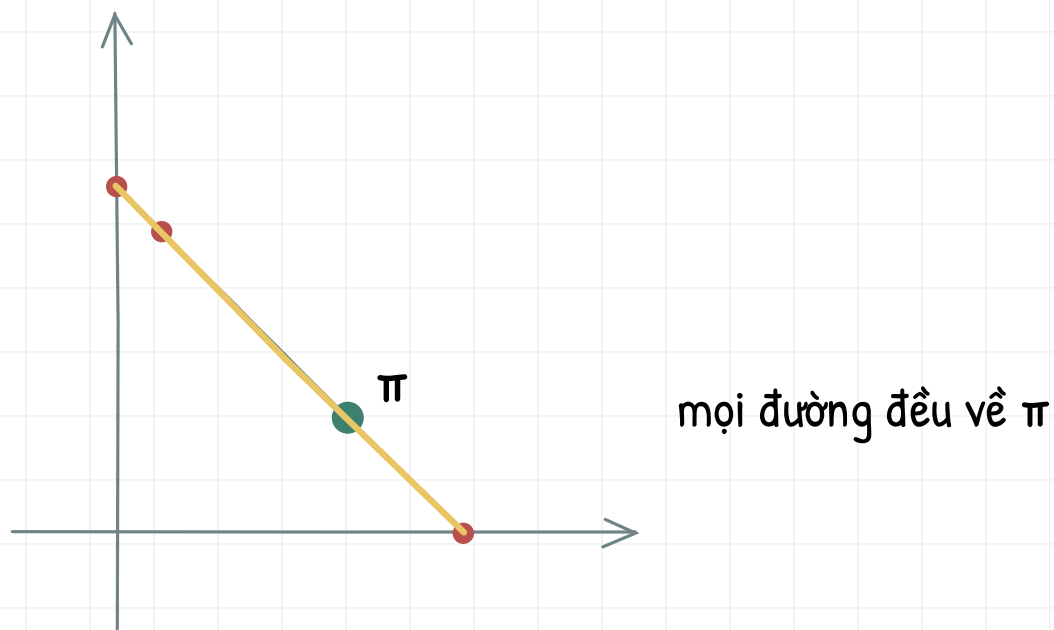
$$\lambda = m \pm \sqrt{m^2 - p} = 0.85 \pm \sqrt{0.0225} = 0.85 \pm 0.15 \Rightarrow 1.0 \text{ và } 0.7 \checkmark$$

"Dùng" nghĩa là gì

Không phải hệ thống ngừng chạy: request vẫn nhảy OK $\leftrightarrow$ LỖI liên tục.

Cái đúng yên là TỈ LỆ tổng thể: cứ 3 request thì 2 đang OK, mãi mãi.

Như cache đã ấm: entry vào ra không ngừng, hit rate đúng im một con số.



Vì sao $\lambda = 1$ LUÔN tồn tại

Với ma trận thường không có gì đảm bảo. Với Markov thì luôn luôn:

1. $\mathbf{1} = (1, \dots, 1)$. "Mỗi cột tổng 1" viết bằng ma trận: $\mathbf{1}^T \cdot \mathbf{M} = \mathbf{1}^T$
2. Chuyển vị: $\mathbf{M}^T \cdot \mathbf{1} = \mathbf{1} \Rightarrow \mathbf{M}^T$ có eigenvector $\mathbf{1}$ với $\lambda = 1$ (hiển nhiên).
3. $\det(\mathbf{M} - \lambda \mathbf{I}) = \det(\mathbf{M}^T - \lambda \mathbf{I}) \Rightarrow$ cùng đa thức đặc trưng
 $\Rightarrow$ cùng tập eigenvalue $\Rightarrow \mathbf{M}$ cũng có $\lambda = 1$. ■

$\lambda = 1 \Rightarrow \det(\mathbf{M} - \mathbf{I}) = 0 \Rightarrow (\mathbf{M} - \mathbf{I})\mathbf{v} = 0$ có nghiệm $\neq 0$: phân phối dừng tồn tại.

Kiểm trên Markov ngẫu nhiên $3 \times 3, 5 \times 5, 8 \times 8$: $|\det(\mathbf{M} - \mathbf{I})| \leq 4e^{-17}$, $\max|\lambda| = 1$.

Điểm thứ hai, cũng miễn phí

Mọi eigenvalue của ma trận Markov có $|\lambda| \leq 1$.

$\mathbf{M}$ bảo toàn tổng, không sinh số âm $\Rightarrow$ không thể phóng to một phân phối.

Nếu có $|\lambda| > 1$ thì lặp mãi tổng nổ ra vô cực: mâu thuẫn.

$\lambda = 1$ là eigenvalue LỚN NHẤT, và luôn có mặt.

Đó là lý do phép lặp có đích để đi tới, và PageRank không bao giờ "không có đáp án".

Đồ thị A—B—C—D của bài này lại có $|\lambda_2| = 1$: quả bom nổ ở tờ 9.

Lặp mãi = một bộ LỌC

$$v_0 = c_1 u_1 + c_2 u_2 + \dots + c_n u_n \quad (u_i \text{ eigenvector}, \lambda_i \text{ eigenvalue})$$

$$M^t v_0 = c_1 \cdot \lambda_1^t \cdot u_1 + c_2 \cdot \lambda_2^t \cdot u_2 + \dots + c_n \cdot \lambda_n^t \cdot u_n$$

$$\lambda_1 = 1: 1^t = 1 \Rightarrow \text{thành phần } u_1 \text{ ĐÚNG YÊN}$$

$$|\lambda_i| < 1: \lambda_i^t \rightarrow 0 \Rightarrow \text{mọi thành phần khác CHẾT DẦN}$$

Mỗi vòng nhân thành phần i với λ_i : chỉ $\lambda = 1$ sống sót.

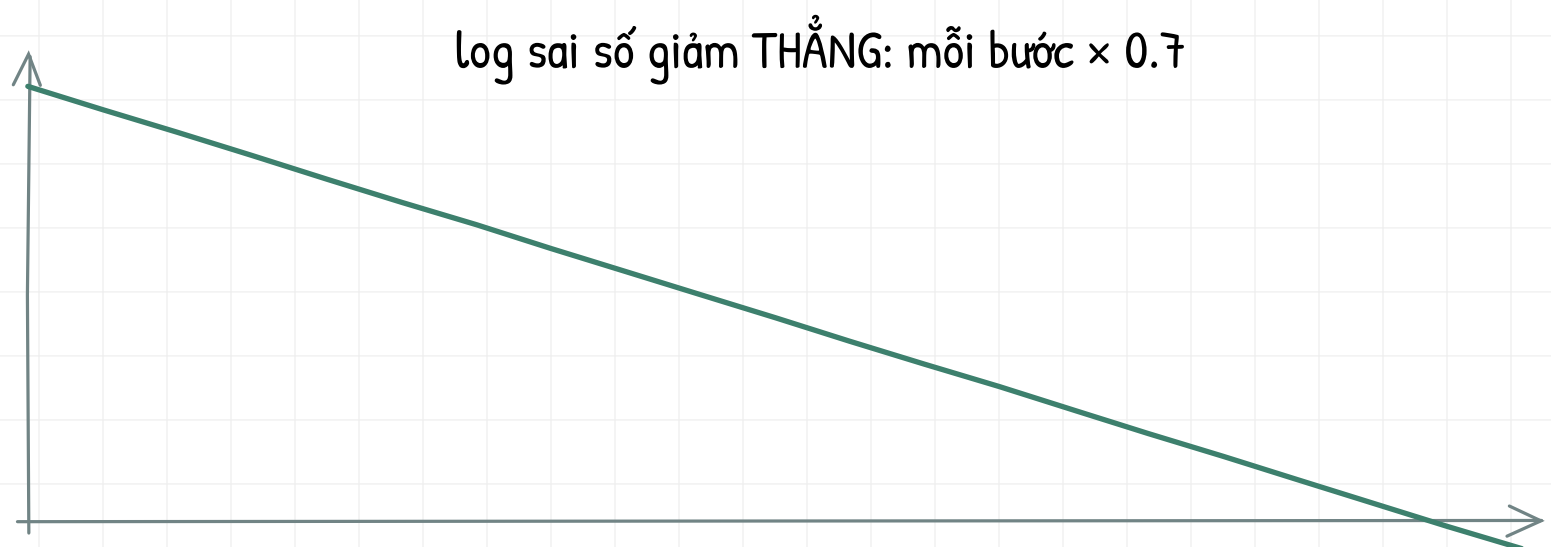
Bất kể xuất phát từ đâu: $(1,0)$, $(0,1)$, $(0.5,0.5)$, $(0.13,0.87)$ đều về $(2/3, 1/3)$.

Tốc độ hội tụ = $|\lambda_2|$

$$v_0 - \pi = (1/3) \cdot (1, -1) \Rightarrow \|v_t - \pi\|_1 = (2/3) \cdot 0.7^t$$

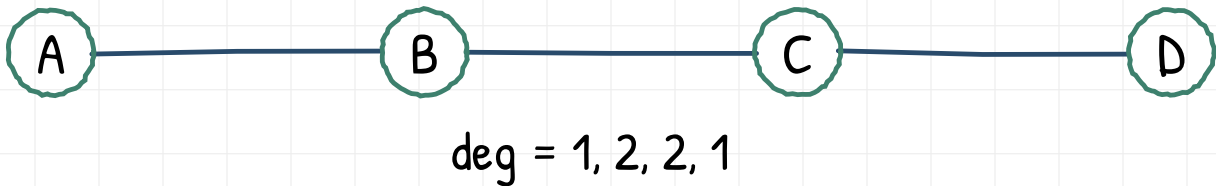
$$\text{bước } 37: (2/3) \cdot 0.7^{37} = 1.24e-06 \quad \text{chưa đạt}$$

$$\text{bước } 38: (2/3) \cdot 0.7^{38} = 8.66e-07 \quad \text{lần đầu } < 1e-6 \quad \checkmark$$



Số bước $\approx \log(\text{tol}) / \log|\lambda_2|$: không phụ thuộc kích thước ma trận,
chỉ phụ thuộc khoảng cách giữa λ_1 và λ_2 .

Tính tay: A—B—C—D, seed A, d = 1/2



$$v(x) = d \cdot \sum_{y \text{ kề } x} v(y) / \deg(y) + (1 - d) \cdot \text{teleport}(x)$$

Mỗi node chia đều điểm cho hàng xóm, giữ tỉ lệ d; phần 1 - d bơm mới vào seed.

$$\begin{aligned} v(A) &= \frac{1}{2} \cdot v(B) / 2 + \frac{1}{2} \\ v(B) &= \frac{1}{2} \cdot (v(A) / 1 + v(C) / 2) \\ v(C) &= \frac{1}{2} \cdot (v(B) / 2 + v(D) / 1) \\ v(D) &= \frac{1}{2} \cdot v(C) / 2 \end{aligned}$$

t	tử số / mẫu chung	thập phân
0	(1, 0, 0, 0) / 1	1, 0, 0, 0
1	(1, 1, 0, 0) / 2	0.5, 0.5, 0, 0
2	(5, 2, 1, 0) / 8	0.625, 0.25, 0.125, 0
3	(18, 11, 2, 1) / 32	0.5625, 0.3438, 0.0625, 0.03125
4	(75, 38, 13, 2) / 128	0.5859, 0.2969, 0.1016, 0.01562
5	(294, 163, 42, 13) / 512	0.5742, 0.3184, 0.08203, 0.02539
∞	(26, 14, 4, 1) / 45	0.5778, 0.3111, 0.08889, 0.02222

Tổng đúng bằng 1 ở mọi dòng: kiểm bug miễn phí.
v(A) dao động rồi thất lại về 0.5778 (vì λ₂ âm, tờ 10).
v(D) chỉ khác 0 từ bước 3: đúng số hop A → D. Điểm phải ĐI BỘ.

Giải ra (26, 14, 4, 1)/45

Đồ thị là đường thẳng: khử từ D ngược về A, khỏi giải 4×4.

$$v(D) = v(C)/4$$

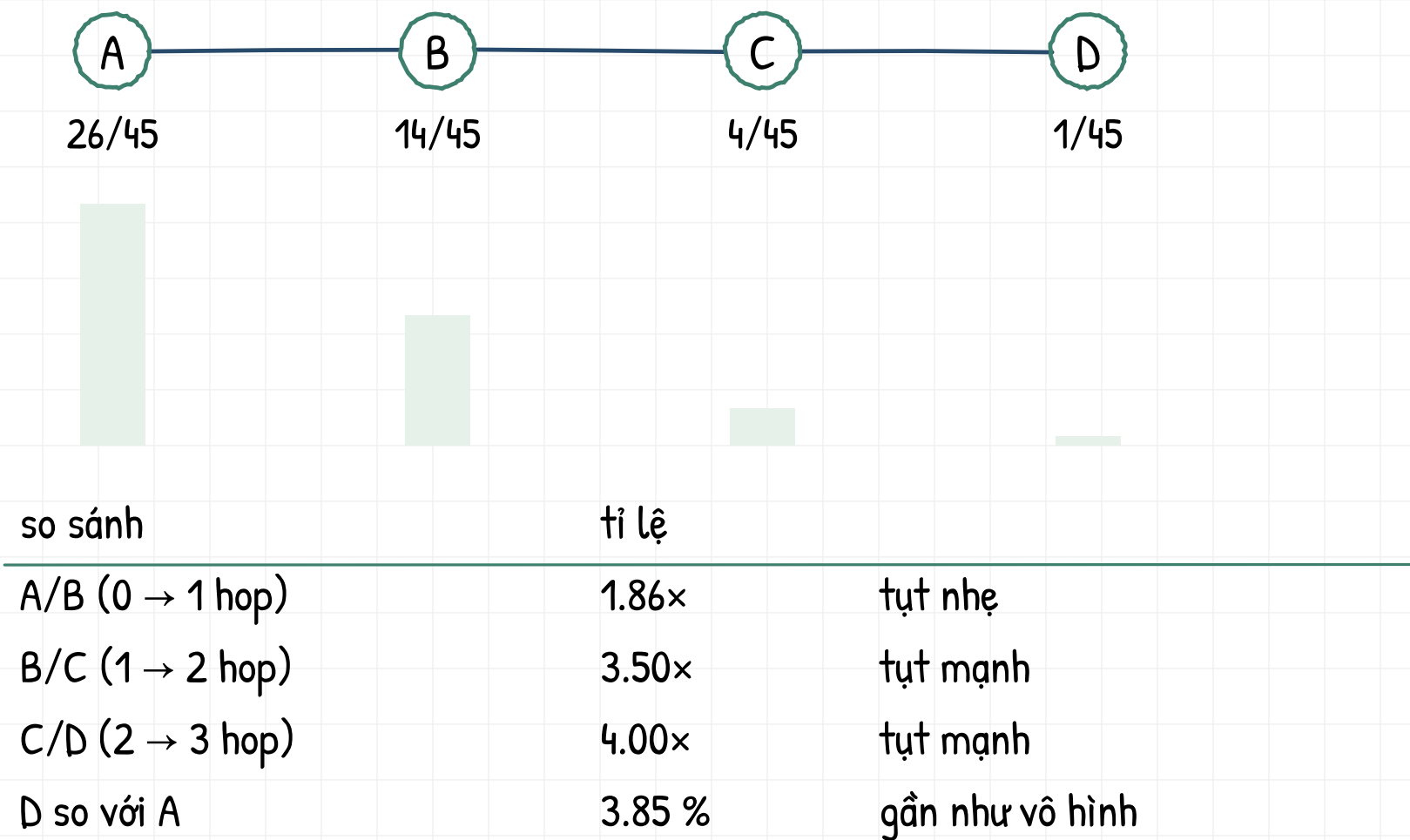
$$v(C) = v(B)/4 + v(C)/8 \Rightarrow v(C) = (2/7) \cdot v(B)$$

$$v(B) = v(A)/2 + v(B)/14 \Rightarrow v(B) = (7/13) \cdot v(A)$$

$$v(A) = v(B)/4 + \frac{1}{2} \Rightarrow v(A) = 26/45$$

$$\Rightarrow v(B) = 14/45, \quad v(C) = 4/45, \quad v(D) = 1/45$$

Kiểm: $(26 + 14 + 4 + 1)/45 = 1 \checkmark$ (không áp tổng ở đâu, nó tự ra)

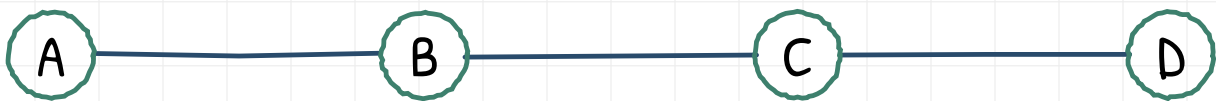


Hai lớp suy giảm chồng nhau: d↑ phạt theo độ dài đường đi, 1/deg phạt theo độ đông hàng xóm. d là núm chỉnh "nhìn xa bao nhiêu hop".

A' đếm đường đi độ dài l

$(A^2)_{ij} = \sum_k A_{ik} \cdot A_{kj}$: mỗi số hạng = 1 khi có $j \rightarrow k$ VÀ $k \rightarrow i$.

$(A^l)_{ij}$ = số đường đi độ dài l từ j tới i (quy nạp).



Từ A, độ dài 2: A-B-A, A-B-C $\Rightarrow$ cột A của $A^2 = (1, 0, 1, 0)$
Độ dài 3: A-B-A-B, A-B-C-B (2 đường tới B), A-B-C-D $\Rightarrow (0, 2, 0, 1)$
Phạm vi: "đường đi" cho phép lặp nút (A-B-A tính là một đường).

Nối vào bài này

Chuỗi PPR là tổng các đường đi: $v = (1 - d) \cdot (e_A + d \cdot M \cdot e_A + d^2 \cdot M^2 \cdot e_A + \dots)$
 $M = A \cdot D^{-1}$: mỗi bước rời y nhân thêm $1/\deg(y)$. $M^2 e_A = (\frac{1}{2}, 0, \frac{1}{2}, 0)$,
nhân $(1 - d)d^2 = 0.125 \Rightarrow (0.0625, 0, 0.0625, 0)$: đúng hàng $t = 2$.

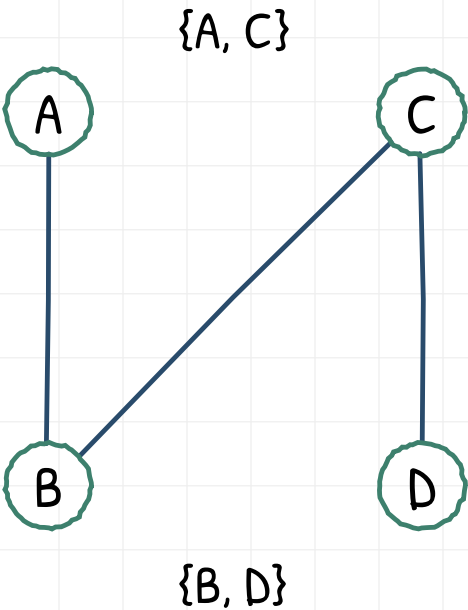
t	d^t	đóng góp vào v
0	1	(0.5, 0, 0, 0)
1	0.5	(0, 0.25, 0, 0)
2	0.25	(0.0625, 0, 0.0625, 0)
3	0.125	(0, 0.046875, 0, 0.015625)

Hai phía: mọi đường độ dài lẻ từ A kết thúc phía bên kia $\Rightarrow (A^l)[A,A] = 0$ với l lẻ.
Đường chéo $l = 2, 4, 6$: (1,2,2,1), (2,5,5,2), (5,13,13,5). Chu kỳ 2, thấy bằng đếm.

d = 1 hỏng kiểu 1: DAO ĐỘNG mãi

eigenvalues của $M = [1, -1, 0.5, -0.5]$: thành phần $\lambda = -1$ nhân $(-1)^t$, không teo.

t	v (d = 1, từ A)
0	(1, 0, 0, 0)
1	(0, 1, 0, 0)
2	(0.5, 0, 0.5, 0)
3	(0, 0.75, 0, 0.25)
4	(0.375, 0, 0.625, 0)
5	(0, 0.6875, 0, 0.3125)
6	(0.3438, 0, 0.6562, 0)
1000	(0.3333, 0, 0.6667, 0)
1001	(0, 0.6667, 0, 0.3333)



Hai phía: mọi cạnh nối phía này sang phía kia.
Đi một bước là chắc chắn đổi phía $\Rightarrow$ không bao giờ dừng.

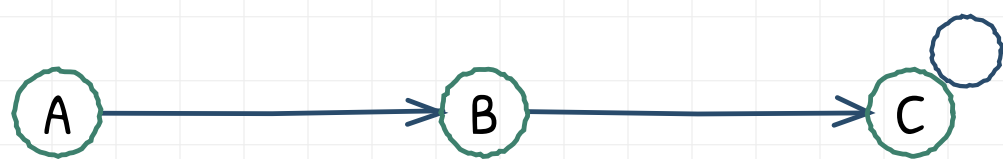
Hai phía $\Leftrightarrow$ không có chu trình độ dài LẼ.

Kiểm bằng BFS tô hai màu: tô được $\Rightarrow$ hai phía; gặp cạnh hai đầu cùng màu $\Rightarrow$ cạnh đó đóng một chu trình lẻ. Một assert rẻ trước khi tin số hội tụ.

đồ thị	eigenvalue của M	d = 1 sau 60 bước
A—B—C—D	-1, -0.5, 0.5, 1	dao động mãi
+ cạnh A—C	-0.73, -0.5, 0.23, 1	(0.25,0.25,0.375,0.125)

Một cạnh tạo tam giác là đủ để -1 biến mất; kết quả = bậc / 2|E|.

Kiểu 2: SPIDER TRAP, và cách vá



$v_0 = (1/3, 1/3, 1/3)$
 $v_1 = (0, 1/3, 2/3)$
 $v_2 = (0, 0, 1) \leftarrow$ xong, A và B = 0 vĩnh viễn

C là cái hố: vào được, không ra được. Mánh spam: cụm trang chỉ trỏ lẫn nhau.
 $d = 0.85$ (teleport đều): $(0.05, 0.0925, 0.8575)$. C vẫn cao nhất, A và B sống.

Ma trận Google

$G = d \cdot M + (1 - d) \cdot \text{teleport} \cdot 1^T$

$d = 0.5$, seed A: $G = [[.5,.75,.5,.5],[.5,0,.25,0],[0,.25,0,.5],[0,0,.25,0]]$, cột tổng 1.

ma trận	eigenvalues	$ \lambda_2 $	hội tụ?
M ($d = 1$)	1, -1, 0.5, -0.5	1.0	không
G ($d = 0.5$)	1, -0.5, -0.25, 0.25	0.5	có

$\text{spec}(G) = \{1\} \cup \{ d \cdot \lambda : \lambda \in \text{spec}(M), \lambda \neq 1 \}$

Khớp ở $d = 0.5$ và 0.85 , teleport đều / dồn seed / ngẫu nhiên: không quan tâm teleport.
Damping BÓP mọi eigenvalue phụ xuống d lần: $|\lambda_2|$ từ 1 còn $d < 1$. Đó là thứ ép hội tụ.
 $\lambda = 1$ có sẵn miễn phí (tờ 4); d mua việc phép lặp TỐI ĐƯỢC đích.

Cái giá: d càng lớn chạy càng lâu

$|\lambda_2| = d \Rightarrow$ sai số giảm đúng d lần mỗi bước (từ bước 3: đúng 0.5000).

d	tol = 1e-6	tol = 1e-10
0.5	21	35
0.85	90	146
0.99	1444	2361

(lặp từ vector teleport, dừng khi $\|\Delta v\|_1 < \text{tol}$)

d = 0.99 tốn gấp 69 lần d = 0.5.

Như nói timeout: kết quả "đúng" hơn nhưng p99 latency nổ.

Google chọn d = 0.85: đủ xa 1 để nhanh, đủ gần 1 để cấu trúc link chi phối.

Quét d: "tầm nhìn xa"

d	v(A)	v(D)	D chiếm
0.1	0.9045	0.000228	0.02 %
0.3	0.7338	0.005312	0.53 %
0.5	0.5778	0.022222	2.22 %
0.7	0.4240	0.057483	5.75 %
0.85	0.3022	0.101284	10.13 %
0.95	0.2140	0.141947	14.19 %

d = 0.1: node cách 3 hop gần như vô hình. d = 0.95: chiếm 14 %. Chọn d theo đó.

Bỏ qua được: chứng minh Perron-Frobenius. Nhớ kết luận và rằng nó đến từ G dương.

PageRank và PPR: khác đúng một vector

	teleport	trả lời
PageRank	đều ($1/n, \dots, 1/n$)	quan trọng NÓI CHUNG
PPR	dồn vào seed	quan trọng VỚI seed này

PageRank (teleport đều) trên A—B—C—D:

d	A	B	C = B, D = A
0.5	0.2	0.3	đối xứng
0.85	0.1754	0.3246	
→ 1	1/6	1/3	= deg / 2 E

d → 1: PageRank vô hướng chỉ còn là BẬC. Giá trị nằm ở chỗ d CHƯA bằng 1.

PPR, d = 0.5, đổi seed:

seed	A	B	C	D
A	0.5778	0.3111	0.0889	0.0222
B	0.1556	0.6222	0.1778	0.0444
C	0.0444	0.1778	0.6222	0.1556
D	0.0222	0.0889	0.3111	0.5778

Bốn seed, bốn bảng khác hẳn, cùng đồ thị. Hàng D là gương lật của hàng A.
PageRank = ORDER BY popularity (cache toàn cục). PPR = ORDER BY relevance_to(:seed):
không cache toàn cục được ⇒ đắt hơn khi triển khai.

Trích đúng: PPR → Jeh & Widom, WWW 2003. Brin & Page 1998 chỉ nhắc ý tưởng trong MỘT đoạn; công thức nằm ở tech report. Topic-sensitive: Haveliwala 2003.

Nối vào AI

PageRank

Bảng xếp hạng = eigenvector $\lambda = 1$ của ma trận link và tỉ chiều.

Google không "tính điểm từng trang": tìm một eigenvector.

Power iteration

eig tốn $O(n^3)$, chết với n cỡ triệu.

Lặp chỉ tốn $O(\text{số cạnh})$ mỗi bước; số bước phụ thuộc d , không phụ thuộc n .

Damping

Chống spider trap và dao động trên đồ thị hai phía.

$d = 0.85 \rightarrow \sim 90$ bước cho $1e-6$.

Graph retrieval

HippoRAG chạy PPR trên đồ thị tri thức, seed = thực thể

trích từ câu hỏi, lấy top- k node làm ngữ cảnh. Đứng vòng lặp vừa cài.

Ma trận thưa

KG 50.000 node: ma trận dày $50.000^2 \times 8 \text{ byte} = 20 \text{ GB}$.

Lưu cạnh (CSR): 200.338 ô khác 0 = 0.008 % (networkx 3.6.1).

Attention

Mỗi hàng softmax tổng 1 = một bước lan truyền Markov

trên đồ thị đầy đủ các token; L lớp = lặp L lần.

Tự kiểm, rồi mới sang bài 9

1. Vì sao Markov LUÔN có $\lambda = 1$? Bước nào dùng $\det(A) = \det(A^T)$?
 2. Phân phối dừng và eigenvector $\lambda = 1$: một lý do chúng là CÙNG một vật.
 3. Vì sao từ BẤT KỲ phân phối nào cũng về cùng π ? (nhắc $\lambda_2, \lambda_i^+ \rightarrow 0$)
 4. $d = 1$ hỏng theo HAI kiểu. Eigenvalue nào là thủ phạm của kiểu nào?
 5. A—B—C—D hai phía ép M có eigenvalue nào? Thêm cạnh loại nào để phá, và điều kiện về độ dài chu trình là gì?
 6. Damping làm gì với PHỔ? Suy ra $d = 0.99$ chậm hơn $d = 0.5 \sim 69$ lần, không chạy code.
 7. $d \rightarrow 1$ thì PageRank = $\deg/2|E|$. Giá trị PageRank nằm ở khoảng d nào?
 8. Vì sao power iteration chứ không eig? Vì sao số bước không phụ thuộc n ?
- Câu 6 sâu nhất: một tham số, một tính chất phổ, một chi phí hợp làm một.

Bốn việc phải LÀM

1. Thêm cạnh A—D (vòng 4 node): PPR seed A, $d = 0.5$ ra bao nhiêu? Còn $\lambda = -1$?
2. Thêm cạnh A—C thay vì A—D: $\lambda = -1$ còn không? So độ dài chu trình hai cạnh.
3. Quét d từ 0.1 đến 0.95, vẽ $v(D)$ theo d (bảng ở tờ 11).
4. Tính tay v_5 từ $v_4 = (75, 38, 13, 2)/128$ bằng bốn phương trình, chỉ phân số.

Mang sang bài 9

Eigenvector chỉ có cho ma trận VUÔNG. Embedding 100000×768 thì sao?
SVD thay eigenvalue bằng giá trị suy biến, dùng được cho MỌI ma trận,
và cho biết nên rút chiều nào khi nén: nền của hạng thấp và LoRA.

Đáp nhanh

1. $1^T M = 1^T \Rightarrow M^T 1 = 1 \Rightarrow M^T$ có $\lambda = 1$; $\det(M - \lambda I) = \det(M^T - \lambda I)$ (bước 3)
 $\Rightarrow$ cùng eigenvalue $\Rightarrow M$ có $\lambda = 1$.
2. Cùng một phương trình: $M\pi = \pi$ là $Av = \lambda v$ với $\lambda = 1$.
3. $M^t v_0 = \sum c_i \lambda_i^t u_i$: thành phần $\lambda = 1$ giữ nguyên, $|\lambda_i| < 1$ thì $\lambda_i^t \rightarrow 0$;
 tốc độ $\sim |\lambda_2|^t$.
4. Dao động: $\lambda = -1$ (đồ thị hai phía). Spider trap: $\lambda = 1$ dồn hết vào một
 cụm hút (eigenvector $\lambda = 1$ tập trung ở hố).
5. Đi một bước luôn đổi phía $\Rightarrow \lambda = -1$. Thêm cạnh tạo chu trình LÊ (A—C tạo
 tam giác). A—D tạo chu trình 4 (chẵn) $\Rightarrow$ vẫn hai phía.
6. $|\lambda_2|$ từ 1 xuống d. Số bước $\approx \log \text{tol} / \log d$: $\log 0.5 / \log 0.99 \approx 69$.
7. Khoảng giữa, chưa sát 1: sát 1 chỉ còn bậc, sát 0 chỉ còn teleport (đều).
8. eig $O(n^3)$; lặp $O(\text{số cạnh})/\text{bước}$, số bước $\sim \log \text{tol} / \log d$, không có n.

Việc phải làm

1. $(7, 2, 1, 2)/12 = (0.5833, 0.1667, 0.0833, 0.1667)$. eigvals = $[1, 0, 0, -1]$:
 $\lambda = -1$ VẪN còn, vì vòng 4 node vẫn tô được hai màu.
2. eigvals = $[1, 0.2287, -0.5, -0.7287]$: -1 biến mất, $|\lambda_2| = 0.7287$.
3. $v(D)$: 0.000228, 0.005312, 0.022222, 0.057483, 0.101284, 0.141947.
4. $v_5 = (294, 163, 42, 13)/512$.