Lecture 24 - PageRank

Tuesday, April 21, 2026 4:41 PM

SLPT: You only need to ever buy one kind of pants.

Page Rank

"A modern classic"
- Gittens

PageRank: a centrality algorithm
↳ measure of importance

BITD: a lot of internet
BUT: no good way to find anything

search wasn't well established
→ keyword matching and similar naive techniques

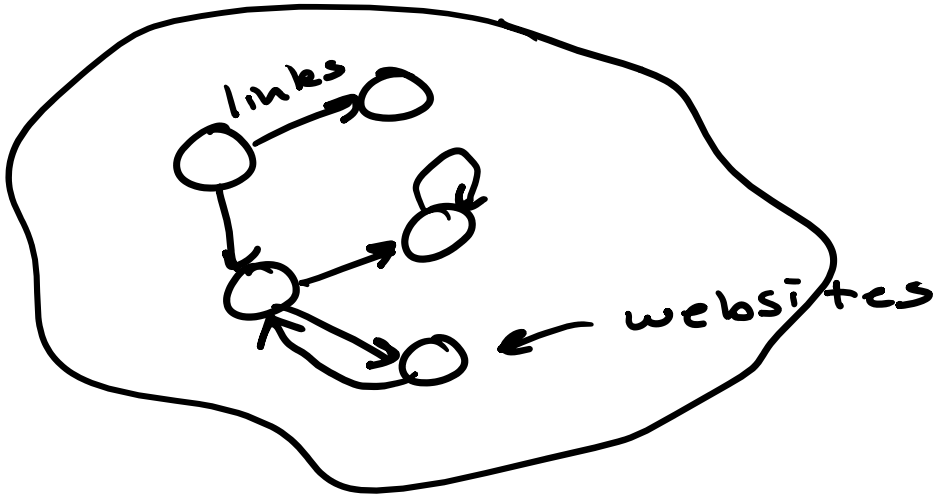
Obus: bad results
→ easy to spam

→ easy to spam

Google: we'll incorporated "trust" in our search

↳ reliable information

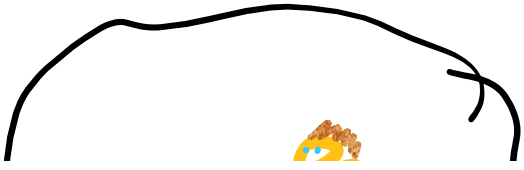
Let's consider the web graph



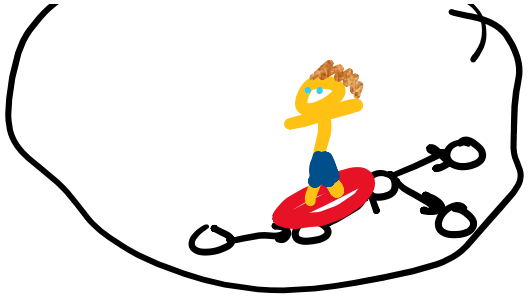
Trust: you have websites linking to you, and these website have their own trustworthiness

---

Random surfer model (walk)



→ random surfer,



→ random surfer, performing an infinite random walk

Internet

PageRank: probability that our random surfer is at same website at same time t

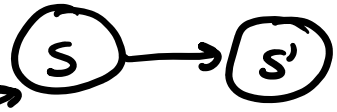
Note: What if the graph isn't strongly connected?

Issue 1: $d^+(v) = 0$, $d^-(v) = 0$

sink source

↳ have a surfer randomly jump to any website from a sink

Issue 2:



← sink SCC

source SCC

↳ randomly jump every k steps

OR jump each step with

OR jump each step with probability p

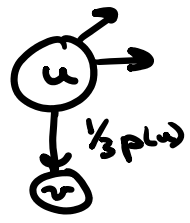
Graph Algorithmic Model

↳ vertex centric computations

PageRank(G):

initialize $p(v) = \frac{1}{|V(G)|}$ for all $v \in V(G)$

↖ PR vector



we iterate $p(v) = \sum_{u \in N^-(v)} \frac{p(u)}{d^+(u)}$

For sources/sinks:

sinks → add edges to all other $v \in V(G)$

For SCC sources/sinks:

Incorporate our

Incorporate our
damping factor

Linear Algebraic Model

(Graphs = matrices)

(Graphs = everything)

(matrices = everything)

Everything is linear algebra

Consider our adjacency matrix A

Consider the diagonal degree matrix D

↳ $D_{ij} = 0, \quad D_{ii} = \sum_{\substack{j \\ i \neq j}} A_{ij}$

Define the transitional probability matrix M

$M = D^{-1}A \rightarrow$ for in-edge transitions

$M = D^{-1}A$ → for in-edge trasitions we instead use M^T

$M^T = (D^{-1}A)^T$

Q: How can we use this to compute pageranks?

A: multiply a pagerank vector

$$p_0(v) = \frac{1}{|V(G)|} \quad \forall v \in V(G)$$

↖ PR iteration zero

$p_{i+1} = M^T p_i$ ← vector of all pageranks at iteration i

After $n \to \infty$ iterations

$p_\infty = M^T p_\infty$

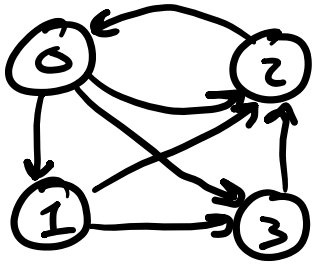
aka $\|p_{i+1} - p_i\| < \epsilon$

↳ recall for same matrix A

$Ax = \lambda x$ ← eigenvalue

↑ eigenvector

=> so Pageranks are just the eigenvector value of the transitional probability matrix with eigenvalue of one □



example PR computation using linear algebraic model

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$M = D^{-1}A$

$$D^{-1} = \begin{bmatrix} \frac{1}{3} & & \cdots & 0 \\ & \frac{1}{2} & & \vdots \\ \vdots & & 1 & \\ 0 & \cdots & & 1 \end{bmatrix} \quad M = \begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$M^T = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 1/2 & 0 & 1 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix} \quad p_0 = \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix}$$

$$p_1 = M^T p_0 = \begin{bmatrix} 1/4 \\ 1/12 \\ 1/12 + 1/8 + 1/4 = \frac{11}{24} \\ 1/12 + 1/8 = \frac{5}{24} \end{bmatrix} = \begin{bmatrix} \frac{6}{24} \\ \frac{2}{24} \\ \frac{11}{24} \\ \frac{5}{24} \end{bmatrix}$$

Competition Networks

vertices = competitors

edges = matches

<u>Orient</u>: add a direction to undirected edges

For PR on a competition network:

→ Orient edges such that they point to winners of matches

point to winners of matches

→ PR "flows" to winners

↳ Gives us a ranking of all the competitors

⇒ we can predict future match results

(who has the higher PR)

Modifications:

Can account for margins of victory

↳ add additional edges (or weights) so that a higher margin of victory gets a larger proportion of PR

Our example:

→ We have the 2025-2026 NFL season results

→ We can predict the outcome of the Super Bowl