The year: 2018
The event: Super Bowl 52 (Pats vs. Eagles)
The odds: New England a clear favorite
Except for one voice who stood up against the masses and said, "No"
That voice: The power of graphs

Review Centrality

↳ measure of "importance"

degree: number edges
(how far we can reach in a hop)
closeness: avg. shortest paths
(how far we can reach in n hops)
betweenness: ratio of shortest paths through v
(how much "flows" through)
eigenvector: how important are your neighbors

↳ PageRank

Back to the 90s

Good: cool new Internet
Bad: can't find much

Bad: can't find much

Ugly: spam and scams

Search Engines (Yahoo, Altavista)
↳ used basic keyword matching

Issue: spam or untrustworthy
sites could abuse these
simple search engines

Q: How can we define "trust"
on a graph?
(web graph)

It's impossible for Yahoo to
manually evaluate trust

However: "wisdom of the crowds"
↳ sum or average behavior
of a crowd can often
carry insight

* In this case, our "crowd" is
the internet as a whole

the internet as a whole

* They "vote" on trustworthiness via where they link

↳ spam still an issue, but more limited

Google:

Let's evaluate trustworthiness as part of a search

↳ PageRank algorithm

PageRank

A spectral centrality

↳ eigenvalues/eigenvectors of the adjacency matrix or a variant

Lots of applications

→ search, recommender systems, rankings, etc.

rankings, etc.

$\Rightarrow$ You're important/good or trusted if others link to you

1. Random surfer model

random surfer
$\downarrow$
infinite random walk on the web graph

big ol' Internet

PageRank: probability we are at page v at any point in time

Issue: sources / sinks
$d^-(v)=0$ $\quad d^+(v)=0$

$\hookrightarrow$ we randomly jump from a sink to any random page

Issue: SCC w/ no out edges

↳ damping factor: probability we randomly jump on a given step

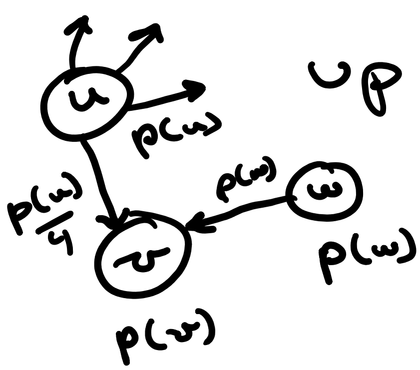
---

2. Vertex-centric model

(graph algorithm model)

init: all $p(v)$ initialized to $\frac{1}{|V(G)|}$

pagerank



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

sinks: can manually compute their contribution

OR

just add edges from sinks to all other vertices

SCC sinks: can do similar as above

γ = damping factor

update $p(v) = (1-\gamma) \sum \frac{p(u)}{} + \gamma$

γ = damping factor

update $p(v) = (1-\gamma)\sum \frac{p(u)}{d^+(u)} + \gamma$

---

3. Linear Algebraic Model

We have adjacency matrix A

diagonal degree matrix D

$\hookrightarrow d_{ii} = \sum_j a_{ij}$

$d_{ij} = 0$ if $i \neq j$

Consider transition prob. matrix M

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

For Pagerank, we want the transpose

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

We can use this matrix to compute PR as above

Initialize $P_0 = \begin{Bmatrix} \frac{1}{|V|} \\ \vdots \\ \frac{1}{|V|} \end{Bmatrix}$

Iterate $p_{i+1} = M p_i$

Until $\| p_{i+1} - p_i \| < \epsilon$ ← stopping criteria

Note: $A x = \lambda x$

eigenvector, eigenvalue

$p_\infty = M p_\infty$ ← power iteration

PR is the eigenvector of the transition prob. matrix with eigenvalue of 1

Personalized PageRank

To extend our surfer analogy

↳ To personalize it to a vertex v

* start from v
* randomly jump back to v

→ gives us a probability distribution

↳ gives us a probability distribution for our random surfer from v

Uses: personalized search, recommender systems, link prediction

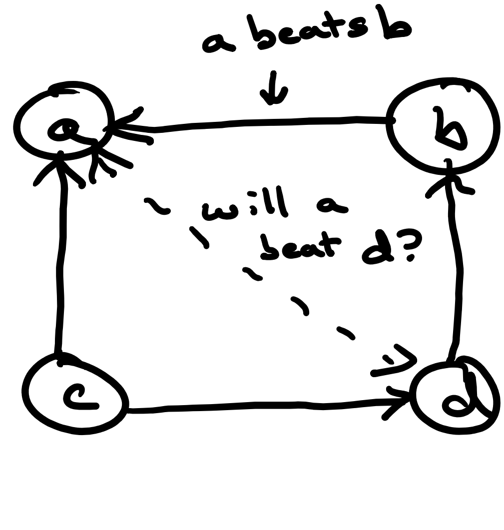
---

PageRank for prediction and ranking in competition networks

↳ vertices: competitors
edges: competition

consider players a, b, c, d

a plays b
b plays d
c plays d
c plays a



We can "orient" our competition network

↳ add direction to edges

↳ add direction to edges s.t. each edge points to the victor

This approach can use PageRank to evaluate and rank competitor quality and use that to predict outcomes

---

Our experiments

- Implent all three above models
- Use competition data (2025-2026 NFL season)
- Predict the outcome of the Super Bowl