Complex-Network Modelling and Inference

Lecture 9: Application: PageRank

Matthew Roughan
<matthew.roughan@adelaide.edu.au>
https://roughan.info/notes/Network_Modelling/

School of Mathematical Sciences,
University of Adelaide

January 14, 2025

Section 1

PageRank

Google PageRank

How does a search engine work [BP98]

- firstly crawl the web (spiders/robots)
  - read each page, and index terms
  - follow links, create graph of web
- order by relevence to the search criteria
  - search "blank verse" returns 690,000 entries
  - too many pages with equal "relevence"
  - easy for punters to "game" the system

Google PageRank

How does a search engine rank the web pages it finds?

- ideally
  - want to rate "quality" of page
  - want to understand what makes people go to a site
  - if a site is more popular, its likely it is more useful
- The original version of PageRank tries to do this
  - if a page has more links **to** it, it must be more interesting
  - if those links come from more "authoritative" sites, then all the better

Simplified Google Page-rank

- Start by giving all n pages equal rank $q_i = 1/n$
- Iterate

$$q_i \leftarrow \sum_{j:(j,i)\in E} q_j / k_j$$

  where k_j is the out-degree of web page j
- In essence, each page "votes" for the other pages by dividing its rank amongst the ones it points to.
- A higher ranked page conveys more rank to those it points to.

Simplified Google Page-rank

- One way to view the process is to consider a random walk on the graph of HTML pages
- Markov chain with equal probability of taking any out-link
- Sinks are treated by re-initializing at a random page.
- For a recurrent Markov Chain (a connected graph) PageRank obtains the equilibrium distribution or probability you will be at page i after a large number of clicks

Linear algebraic formulation

Iterative formulation of Google PageRank

$$\mathrm{q}^{(k+1)} = \mathrm{q}^{(k)} P$$

P is the probability transition matrix:

- P is just formed by normalizing the rows of the adjacency matrix A
- $p_{ij} = a_{ij} / \sum_j a_{ij}$
- Note
$$\mathrm{q}^{(k)} = \mathrm{q}^{(0)} P^k$$
- As $k \to \infty$ we care about P^k's limit

Linear algebraic formulation

Iterative formulation of Google PageRank

$$\mathrm{q}^{(k)} = \mathrm{q}^{(0)} P^k$$

P is the probability transition matrix:

- limiting behaviour of P^k depends on its eigenvalues

$$U^{-1} P U = D$$

where $P\mathrm{v}_k = \lambda_k \mathrm{v}_k$ and

$$U = \left[\begin{array}{c|c|c|c} \vdots & \vdots & & \vdots \\ \mathrm{v}_1 & \mathrm{v}_2 & \cdots & \mathrm{v}_n \\ \vdots & \vdots & & \vdots \end{array} \right] \text{ and } D = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{bmatrix}$$

Powers of matrices

We can compute P^k for a diagonalizable matrix P by

$$P^k = UD^kU^{-1}$$

where

$$D^k = \begin{bmatrix} \lambda_1^k & & & 0 \\ & \lambda_2^k & & \\ & & \ddots & \\ 0 & & & \lambda_n^k \end{bmatrix}$$

- $|\lambda_i| > 1$ then it grows
- $|\lambda_i| < 1$ then it decays
- $|\lambda_i| = 1$ then it remains stable

Perron-Frobenius theorem

- Perron-Frobenius theorem
  - non-negative matrix (entries ≥ 0) and irreducible
    - irreducible = associated graph is fully connected
  - Perron-Frobenius eigenvalue (spectral radius) is real value r such that $r \geq |\lambda_k|$
  - there exists a left eigenvector of r with non-negative entries
- Stochastic matrix P
  - has rows summing to 1, and ≥ 0
  - $r = 1$
  - P^k depends on the eigenvector of $r = 1$

Linear algebraic formulation

Standard equilibrium formulation of Markov chain

$$\pi = \pi P$$

P is the probability transition matrix:

- P is just formed by normalizing the rows of the adjacency matrix A
- π is the stationary distribution (equilibrium distribution)
- expresses balance of probability flows

$$\pi_j = \sum_i \pi_i P_{ij}$$

- Fast algorithms exist for computing eigenvectors

Damping Page-rank

- Above has some poor consequences, *e.g.*, a page that points to others, but has no links to it will be isolated, and so have rank zero.
- It can also be gamed (create a thousand self referential web pages).
- It needs some damping

$$q_i = \frac{1-d}{n} + d \sum_{j:(j,i)\in E} q_j / k_j$$

  - typical value of $d \sim 0.85$

HITS algorithm

- HITS by Jon Kleinberg separates authority from hubishness
- Hubs and authorities
  - hubs have lots of (authoritative) links into them
    - directory or encyclopedia
  - authorities have lots of (hubs) that link to them
    - actual information of value
- iteration
  - start with hubs and authority scores of 1
  - authority(i) $=$ sum of hub scores pointing to i
  - hub(i) $=$ sum of authority scores pointing to i
  - normalize by sums of squares for both scores

Further reading I

S. Brin and L. Page, *The anatomy of a large-scale hypertextual web search engine*, Seventh International World-Wide Web Conference (WWW 1998) (Brisbane, Australia), 1998.