ACM 106a: Lecture

Agenda

- Google PageRank algorithm
- Developing a formula for ranking web pages
- Interpretation
- Computing the score of each page

Google: background

- Mid nineties: many search engines. Often times not that effective
- Late nineties: Google goes online. Very effective search engine
- Seems to get what we are looking for
- At the heart of the engine: PageRank

Search engines

Three basic tasks:

1. Locate all the web pages with public access
2. Index all the web pages so that they can be searched efficiently (by key words or phrases)
3. Rate the importance of each page;

$$\text{query} \longrightarrow \text{returns most important pages first}$$

Many search engines & many ranking algorithms

PageRank

- Determined entirely by the link structure of the Web
- Does not involve any of the actual content of Web pages or of any individual query
- Given a query, finds the pages on the Web that match that query and lists those pages in the order of their PageRank.

Importance of PageRank

- Understanding PageRank influences web page design;

  how do we get listed first?

- Had a profound influence on the structure of the Internet

PageRank: basic idea

Internet is a directed graph with nodes and edges

- Nodes: pages; n pages indexed by $1 \le i \le n$
- Edges: hyperlinks; G is the n by n connectivity matrix

$$G_{i,j} = \begin{cases} 1, & \text{if there is a hyperlink from page } i \text{ to page } j, \\ 0, & \text{otherwise.} \end{cases}$$

Importance score of page i is x_i; x_i is nonnegative and $x_i > x_j$ means that page i is “more important” than page j

First ideas

Why not take as $\boldsymbol{x_i}$ the number of backlinks for page $\boldsymbol{i}$?

First objection: a link to page $\boldsymbol{i}$ should carry much more weight if it comes from an "important page." E.g. a link from CNN or Yahoo! should count more than a link from my webpage.

Modification: $\boldsymbol{L_i}$, set of webpages with a link to page $\boldsymbol{i}$

$$\boldsymbol{x_i = \sum_{j \in L_i} x_j}$$

Second objection: democracy! We do not want to have a page gaining overwhelming influence by simply linking to many pages.

Better idea

Define the self-referential scores as

$$x_i = \sum_{j \in L_i} x_j / n_j,$$

where n_j is the number of outgoing links from page j. A page has high rank if it has links to and from other pages with high rank.

Finding x is some sort of eigenvalue problem: since

$$x = Ax, \qquad A_{i,j} = G_{i,j} / n_j;$$

i.e. x is an eigenvector of A with eigenvalue 1.

But A may not have 1 as an eigenvalue...

Interpretation: Markov chain

- Surfing the Web, going from page to page by randomly choosing an outgoing link from one page to get to the next
- There can be problems:
  – lead to dead ends at pages with no outgoing links (dangling nodes)
  – cycles around cliques of interconnected pages
- Ignoring this problems, this random walk of the Web is a Markov chain
- The matrix $\boldsymbol{A}$ is the transition probability matrix of the Markov chain
- The score is the the limiting probability that the chain visits any particular page, x_i is the fraction of time the surfer spends in the long run, on page i of the web

Stochastic matrices

$\boldsymbol{A}$ is stochastic if all the entries are nonnegative and the columns of $\boldsymbol{A}$ sum to 1

Every stochastic matrix has 1 as an eigenvalue.

Why? $\boldsymbol{A}$ and $\boldsymbol{A^T}$ have the same eigenvalues. But

$$\boldsymbol{A^T} \begin{bmatrix} \mathbf{1} \\ \vdots \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ \vdots \\ \mathbf{1} \end{bmatrix}$$

so 1 is an eigenvalue of $\boldsymbol{A^T}$.

Nonunique rankings

What if there are no dangling nodes (so that $\boldsymbol{A}$ is stochastic) but the Web is such that there are two sets of pages which are disconnected from one another?

E.g. Starting from page i, and following hyperlinks, there are pages you will *never* see; i.e. the graph is disconnected

Then the eigenspace with eigenvalue 1 is at least of dimension 2. The score is ill-defined

The last idea

Define the transition probability matrix P

$$P_{i,j} = (1-\delta)A_{i,j} + \delta, \qquad P = (1-\delta)A + \delta\, \mathbf{1}\, \mathbf{1}^T.$$

Google sets $\delta = .15$.

Interpretation

- With probability $1-\delta$, surfer chooses a link at random
- With probability δ, surfer chooses a random page from anywhere on the Web (uniformly at random).

If $\delta = 0$, this is our previous idea. If $\delta = 1$, then all the webpages have the same score.

The Perron Frobenius Theorem

Assume no dangling node so that A is stochastic. Then

$$P = (1 - \delta)A + \delta\,\mathbf{1}\,\mathbf{1}^T$$

is also stochastic. Note that $P_{i,j} > 0$.

Theorem 1 (Perron Frobenius) *Consider any stochastic matrix obeying $P_{i,j} > 0$ for all pairs (i, j). Then the largest eigenvalue of P is equal to one and that the corresponding eigenvector, which satisfies the equation*

$$x = Px,$$

exists and is unique to within a scaling factor.

Note: More sophisticated results about the existence and uniqueness of the equilibrium measure of a Markov chain exist.

Normalize so that $\sum_i x_i = 1$, then this is the limiting probability distribution and the x_i's are the Google's PageRanks.

How to compute the largest eigenvector?

Big problem: n is well above 10 billion

Only real hope is the power method

Power method along with modification for speedup (shifts etc.):

- Pick $x^{(0)}$ and set $i = 0$
- Repeat
  - $x^{(i+1)} = Px^{(i)}/\|Px^{(i)}\|$

  until convergence

Rate of convergence depends on the eigenvalue gap, expected decrease is proportional to $|\lambda_2/\lambda_1|$ ($\lambda_1 = 1$ here)

$$\|x^{(i)} - x\| \leq O(|\lambda_2/\lambda_1|^i)\,\|x^{(0)} - x\|$$

Computed frequently. Can use yesterday's eigenvector as today's $x^{(0)}$.

Requires applying A (sparse) and $\mathbf{1}\,\mathbf{1}^T$ (cheap) many times. Still, this is an enormous computation (requires many computers, shared memory etc.)

Resources

1. K. Bryan and T. Leise, The $25,000,000,000 Eigenvector: The Linear Algebra behind Google. *SIAM Review* (2006).

2. C. Moler, The Worlds Largest Matrix Computation, (August 1, 2005).
   `http://www.mathworks.com/company/newsletters/news_notes/clevescorner/oct02_cleve.html`