link = edge ≃ good reference

linkedIN Facebook

- size and scale
- page importance (not content)
- connectivity/linkage

"page rank"

The WebGraph

legal docs patents

academia paper references

July 13, 2015

1 Web Graph: pages and links

?? product recomendat

The WebGraph describes the directed links between pages of the World Wide Web. A directed edge connects page X to page Y if there exists a hyperlink on page X, referring to page Y. Where X and Y are two vertices of giant WebGraph.

1.1 Modern Web : size and scale

disease/symptoms

Every day the World Wide Web grows by millions of electronic pages, adding to the billions of pages existing already. This staggering volume of information is loosely held together by billions of annotated connections, called hyperlinks.

people hierarchy in company subordination

CEO

Since the World Wide Web is so dynamic in nature we can only estimate the number of pages contained in WWW web-graph. The WWW web-graph is unstructured collection of documents growing and shrinking in size however incrementing total pages in sum total. Below is an estimate of google and bing document collection.

company deal/collab

macro-economic graph

money

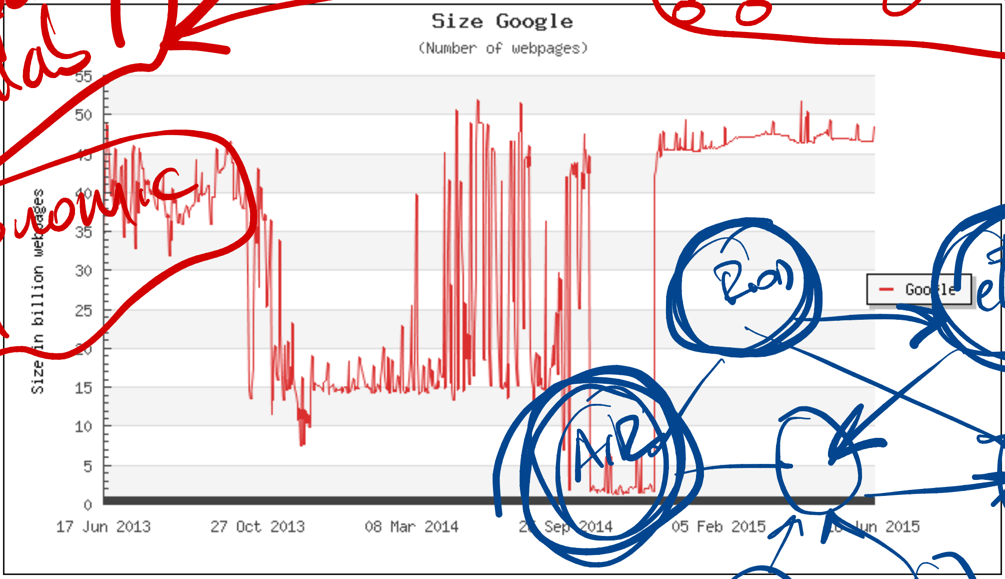


Figure 1: 2 year estimate of Google Web corpus

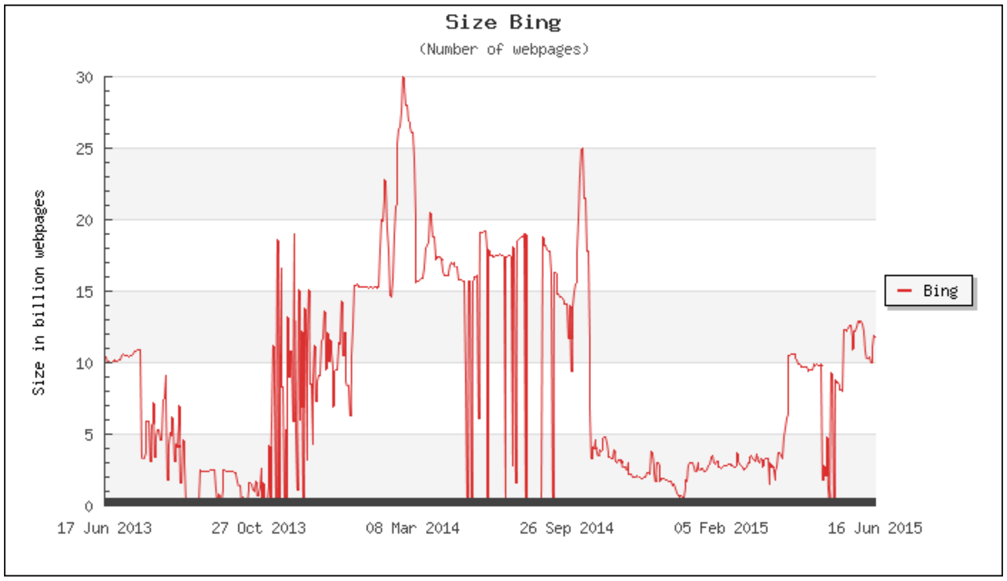
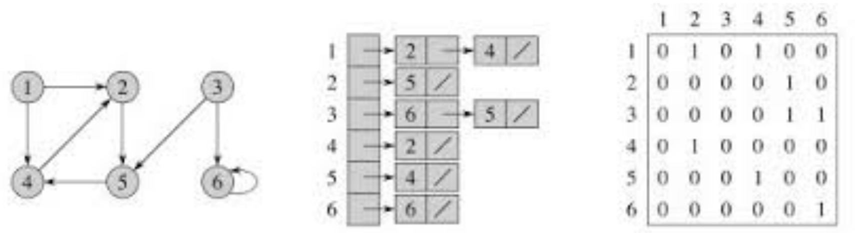


Figure 2: 2 year estimate of Bing Web corpus

1.2 Constructing the Web Graph

Essentially the web graph can be constructed by crawling the web. The in-links or out-links of web pages formulates the web-graph. For computation with graph the web-graph can be represented in either adjacency matrix or adjacency list form.



	1	2	3	4	5	6
1	0	1	0	1	0	0
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	1	0	0	0	0
5	0	0	0	1	0	0
6	0	0	0	0	0	1

Figure 3: Adjacency List and Matrix representation

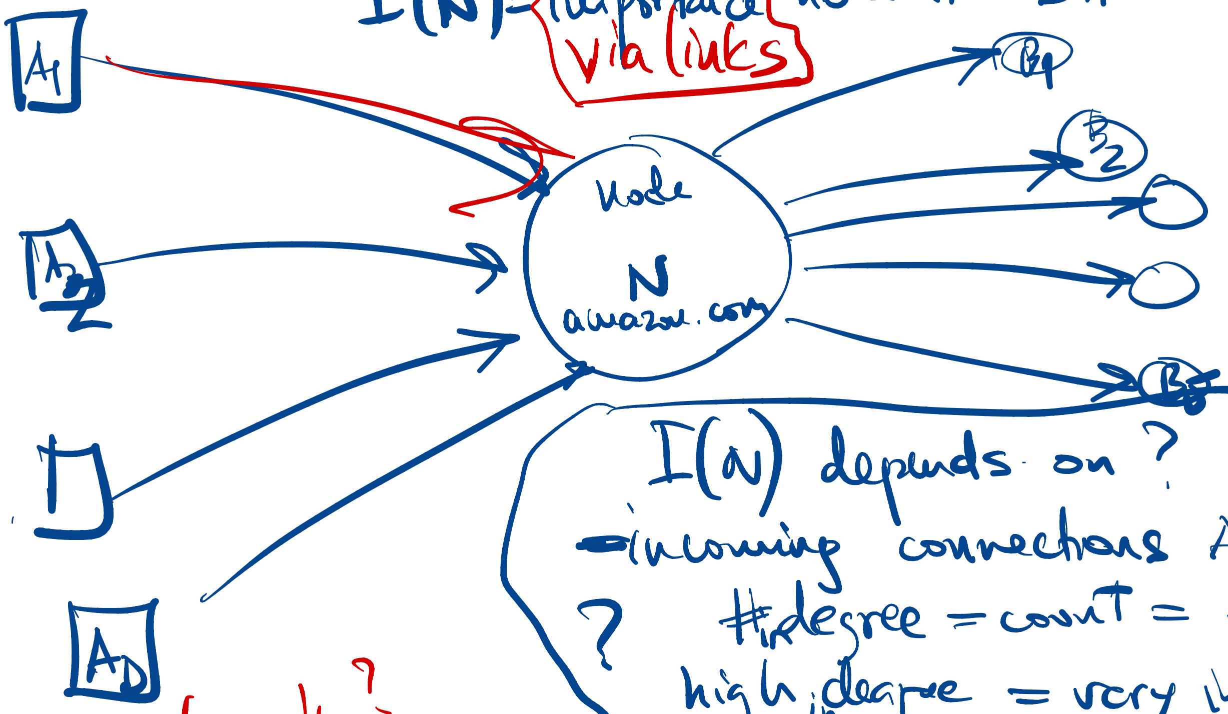
2 Graphs Recap

A graph is a way of specifying relationships among a collection of items. A graph consists of a set of objects, called nodes, with certain pairs of these objects connected by links called edges. Graphs can be divided in two categories

- Directed Graphs
- Undirected Graphs

A is the typical representation of a graph with little circles representing the nodes, and a line connecting each pair of nodes that are linked by an edge.

$I(N)$ = importance of node in Directed Graph? via links



$I(N)$ depends on ?

- incoming connections $A_i \to N$
  ? #indegree = count = e.x. 4
  high $_{in}$degree = very important

- outgoing connections #outdegree count = e.x. 5
  ? high outdegree = not important
  low outdegree = very important

Facebook? but not other graphs?

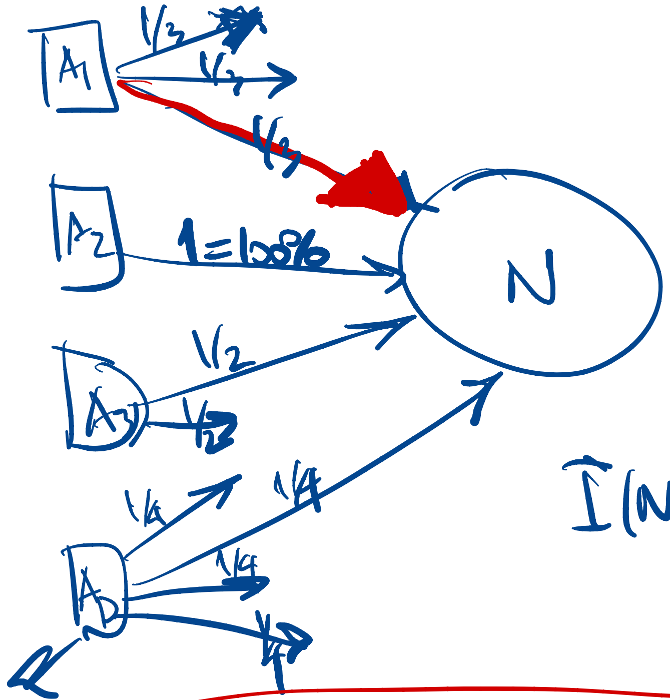
- ratio $\frac{\text{indegree}}{\text{outdegree}}$ = high $\Leftrightarrow$ important
  ? $\frac{\text{outdegree}}{\text{indegree}}$ = low $\Leftrightarrow$ not important?

outdegree? $\Leftarrow$

- # of times the node is visited (by a? ~~traversal?~~)
  webgraph — users (browsing the web)

- $\boxed{\text{importance of nodes} \to N \quad \text{e.x. } I(A_1)\ I(A_2), \ldots I(A_D) \Rightarrow \text{Weighting schema}}$

  ↳ recursive? definition $I(N)$ = function of $(I(A_1), \ldots I(A_D))$



$I(N)$ = importance = function of $I(A_1)\ I(A_2) \ldots I(A_d)$.

$I(N)$ = sum of importances coming from the links?

$$I(N) = \sum_{\text{incoming}} \text{importance } A_i \text{ sends to } N$$

$A_i \to N$

$$\boxed{I(N) = \sum_{A_i \to N} \frac{I(A_i)}{\text{outdegree}(A_i)}}$$ OR jump

incomplete yet

known jump

vectorial form

uniform ⟺ user follows links at random.

$\bar{I}$ = vector

$\bar{I} = \bar{I} \cdot$ matrix of outdegrees

Markov chain = probability chain $\rightarrow$ model prob of sequence
memory-less

states : B, M, S (many states)

transition

pr(B→M)	pr(B→S)	pr(B→B)
pr(M→M)	pr(M→S)	pr(M→B)
pr(S→M)	pr(S→S)	pr(S→B)

ex. starcraft
chess
- - -

$\boxed{\text{game}}$ = changing states

- manufacturing process

process = advancing to next stage

- weather

pattern = cycle of weather states

$$\begin{tikzcd} \text{state } \textcircled{B} \arrow[r] & \text{state } \textcircled{M} \end{tikzcd}$$

deterministic ⇒ Dectre

probabilistic => Marko

- other physics

- human situation

basic example Markov.   N = # of agents

3 pizza places B, M, S

everybody in the city (N = 2M) eats pizza every day.

$$\begin{tikzcd}
 & B \arrow[loop above, "0.7"] \arrow[dl, bend right, "0.2"'] \arrow[dr, bend left, "0.1"] & \\
M \arrow[ur, "0.3"] \arrow[rr, "0.1"] \arrow[loop below, "0.6"] & & S \arrow[ul, "0.3"'] \arrow[ll, bend left, "0.2"] \arrow[loop right, "0.5"]
\end{tikzcd}$$

$P(M|B) = P(B \to M)$ = probab that one eating pizza@B today will eat at @M tomorrow. "moving" from B to M

P = transition matrix = $\boxed{\text{fixed}}$

	B	M	S
B	.7	.2	.1
M	.3	.6	.1
S	.3	.2	.5

memory less: (no history)
$P(B \to M)$ only depends on being currently at B. (not path to get there)

$p(B \to M)$ = prob of one eating yesterday at B eats today at M 1 agent (follow)

= precentage of population moving from B to M. All N agents (follow)

Q: $\boxed{\text{today}}$ current distribution of agents is

$\pi(B)$ $\pi(M)$ $\pi(S)$ $\sum_i \pi(i) = 1$

then what is/calculate the next distr π_{next} $\boxed{\text{tomorrow}}$

$\pi_{next}(B)$ $\pi_{next}(M)$ $\pi_{next}(S)$ $? = f(\pi)$

tomorrow

$$\pi_{next}(B) = \underbrace{\pi(B) \cdot P(B \to B)}_{\text{from B}} + \underbrace{\overset{\text{today}}{\pi(M)} \cdot P(M \to B)}_{\text{from M}} + \underbrace{\pi(S) \cdot P(S \to B)}_{\text{from S}}$$

$\pi_{next}(M) = \pi(B)\cdot P(B\to M) + \pi(M)\cdot P(M\to M) + \pi(S)\cdot P(S\to M)$

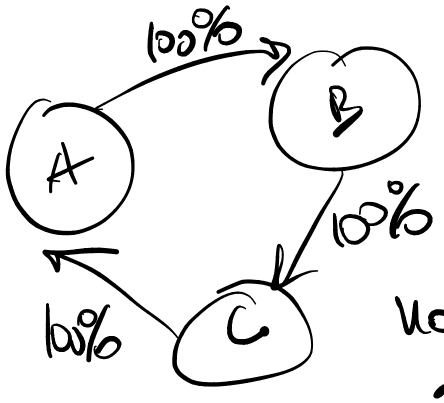
$\pi_{next}(S) = \pi(B)\cdot P(B\to S) + \pi(M)\cdot P(M\to S) + \pi(S)\cdot P(S\to S)$

stationary distribution π^*: distribution that converged to non-movable state dist for transition matrix P.

today $\pi^* \Longrightarrow$ tomorrow π^*

* most Markov Chains + trans probs P $\longrightarrow$ stationary dist π^*
* not all (periodic, ergodicity properties)

$$\begin{array}{c|ccc} & A & B & C \\ \hline A & 0 & 1 & 0 \\ B & 0 & 0 & 1 \\ C & 1 & 0 & 0 \end{array}$$



no station. dist !

~~Theorem~~ * stationary unique distribution π^* can be achieved from any starting distribution

time t=0 π_0 = initial dist of agents to states

$$\begin{array}{cccc} & B & M & S \\ \pi_0 = & 1/3 & 1/3 & 1/3 \text{ (init)} \\ \text{OR } \pi_0 = & 1/2 & 1/8 & 3/8 \\ \text{OR } \pi_0 = & \frac{1}{10} & \frac{1}{10} & \frac{8}{10} \end{array}$$

time

t=1 π_1

t=2 π_2 . . . . . time n (large n)

⋮ $\pi_n \simeq \pi^*$

$\pi_{next}(B) = \pi(B)\cdot P(B \to B) + \pi(M)\cdot P(M \to B) + \pi(S)\cdot P(S \to B)$

$\vdots$

$$\boxed{\pi_{next} = \pi \underset{\text{mult.}}{\overset{\text{matrix}}{\times}} P}$$

$$\begin{bmatrix} \pi_{next}(B) & \pi_{next}(M) & \pi_{next}(S) \end{bmatrix}_{1\times 3} = \begin{bmatrix} \pi(B) & \pi(M) & \pi(S) \end{bmatrix}_{1\times 3} \cdot \begin{bmatrix} P \end{bmatrix}_{3\times 3}$$

$\pi_1 = \pi_0 \cdot P$

$\pi_2 = \pi_1 \cdot P = \pi_0 \cdot P \cdot P = \pi_0 \cdot P^2$

$\pi_3 = \pi_2 \cdot P = \pi_0 \cdot P^3$

$\vdots$

$$\boxed{\pi_n = \pi_0 \cdot P^n} \quad \forall n \geq 1$$

stationary distribution π^*

$$\boxed{\pi^* = \pi^* \cdot \boxed{P}}$$

$\to$ known

transition 1 step doesn't change the distribution.

Task: compute stationary dist π^*, given trans. matrix P.

3 Methods.

(I) iterative Matrix Mult

pick π_0 = uniform $1/3\ 1/3\ 1/3$

$\pi_1 = \pi_0 P,\ \pi_2 = \pi_1 . P \ldots \pi_n = \pi_{n-1} \cdot P$

(II) solve linear system

$\pi^* = \pi^* \cdot P$

3x3 linear system

π = var P = coef

until convergence criteria

$\pi_0 \to \pi_1 \ldots \to \pi_n \to \ldots \to \pi^*$

$|\pi_n - \pi_{n-1}| < \varepsilon$ threshold

Alg = ? efficient

$n=100 \quad \pi_{100} = \pi_0 \cdot P^{100}$

naively: $\times P$ 100 times.

better way: $P, P^2, P^4, P^8, P^{16}, P^{32}, P^{64}$ — 6 mult

repeated squaring mult

$P^2 \cdot P^2 \quad P^{16} \cdot P^{16}$

$100 = 64+32+4$

$\pi_{100} = \pi_0 \cdot P^{100} = \pi_0 \cdot \underbrace{P^{64} \cdot P^{32} \cdot P^4}_{3 \text{ mult}}$

total: 9 multiplic.

$B^* = \pi^*(B)$, var M^*, S^*

$B^* = B^* \times 0.7 + M^* \times 0.3 + S^* \times 0.3$

$M^* = B^* \times 0.2 + M^* \times 0.6 + S^* \times 0.2$

$S^* = B^* \times 0.1 + M^* \times 0.1 + S^* \times 0.5$

$\boxed{\begin{array}{l} \det |P| = 0 \\ \sum \text{row}(P) = 1 \end{array}}$?

to solve linear system +1 equation

$B^* + M^* + S^* = 1$

π^* = valid distribution

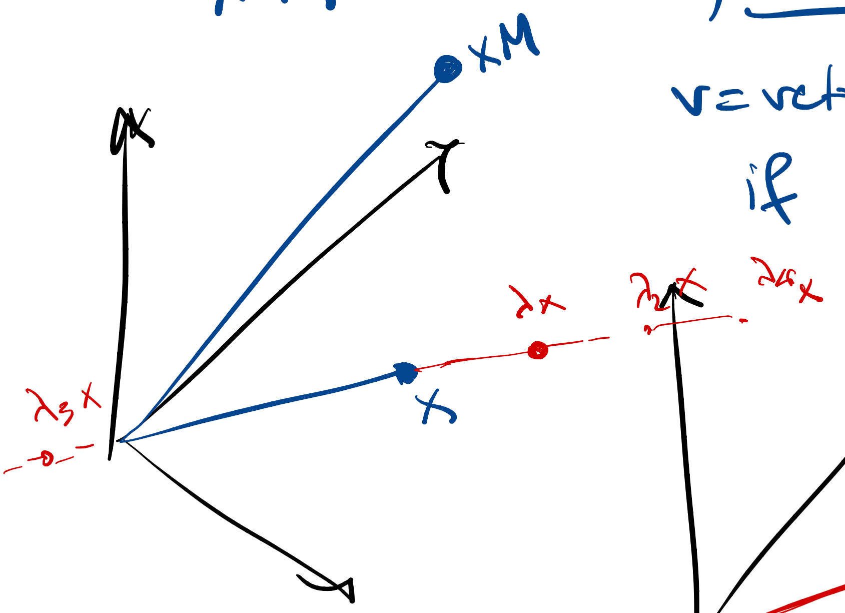
III eigenvectors/eigenvalue decomposition of P

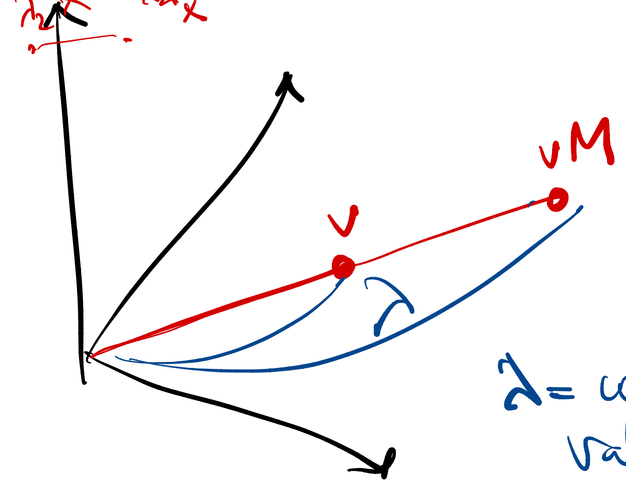
x = vector M = matrix

$x \cdot M$ = new vector, usually with different {direction than x / scale/size

v = vector $\boxed{\text{eigenvector for matrix } M}$

if $v \cdot M$ has the same direction as v (diff size) (same line)





$v \cdot M = v \cdot \lambda$

λ = scalar $\in \mathbb{R}$

λ = corresponding eigen value for v

$\underset{\text{vector}}{\pi^*} = \pi^* \cdot P \Longrightarrow \pi^*$ eigenvector of P

$\lambda = ?$ $\quad \pi^* \cdot 1 = \pi^* \cdot P \Rightarrow \lambda = 1$

<u>Method</u>: find the eigen-decomposition of P, select the eigenvector π^* corresponding to $\lambda = 1$ eigenvalue

* not very stable for $\left\{\begin{array}{l}\text{large matrices} \Longrightarrow \text{computation issues}\\ \text{sparse}\end{array}\right.$

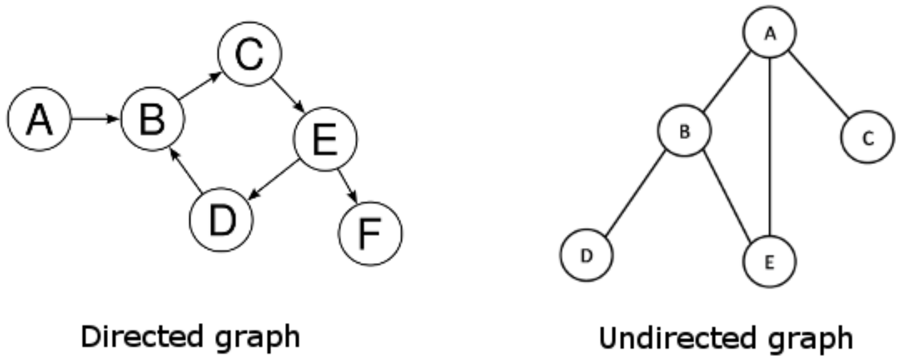


Figure 4: Graphs

In undirected graphs the relation between two nodes is symmetric; the edge simply connects them to each other. However in case of WebGraph we may want to express asymmetric relationships for example, that X points to Y but not neccesarily vice versa.

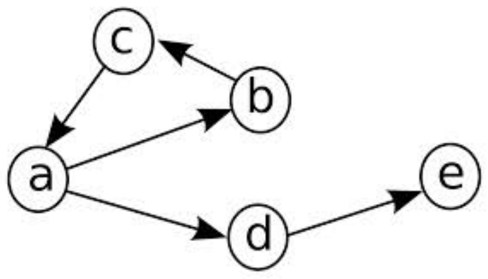


Figure 5: Directed Cyclic Graph

3 Graph Traversals

3.1 BFS traversal

Given a graph G = (V, E) and a distinguished source vertex S, breadth-first search systematically explores the edges of G to discover every vertex that is reachable from S. Breadth-first search is so named because it expands the frontier between discovered and undiscovered vertices uniformly across the breadth of the frontier. BFS traversal produces a breadth-first tree rooted at S that contains all reachable vertices.

3.2 DFS traversal

Depth-first search explores edges out of the most recently discovered vertex that still has unexplored edges leaving it. Once all of node's edges have been explored, the search backtracks to explore edges leaving the vertex from which was discovered. This process continues until we have discovered all the vertices that are reachable from the original source vertex.

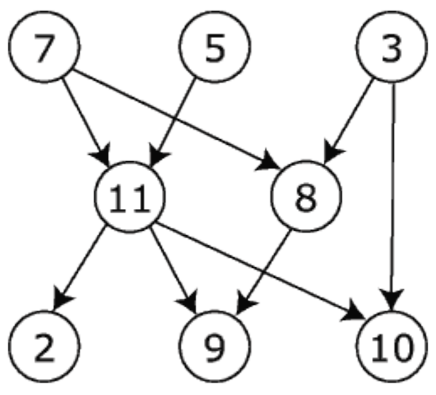


Figure 6: Directed Acyclic Graph

3.3 In/Out Degrees

The In/Out degree have a meaning when we are dealing with directed graph for example WebGraph. As the name suggest the IN degree corresponds to number of edges pointed towards a node/vertex in a graph. Similarly the OUT degree corresponds to number of edges going out from a node/vertiex in a graph.

3.4 Graph Cut

A cut is a partition of the vertices of a graph into two disjoint subsets. A cut C=(S,T) is a partition of V of a graph G=(V,E) into two subsets S and T. The cut-set of a cut C=(S,T) is the set (u,v) belongs to E, where u belongs to S and v belongs to T of edges that have one endpoint in S and the other endpoint in T.

4 Page Rank

Page Rank is procedure of calculating rank of web-page by solely deriving ranking by link structure of WebGraph. In layman term PageRank is a VOTE, by all the other pages on the Web, about how important a page is. A link to a page counts as a vote of support. If there is no link there is no support.

4.1 Markov Chains Introduction

A Markov chain is a discrete-time stochastic process: a process that occurs in a series of time-steps in each of which a random choice is made. A Markov chain consists of N states. Each web page will correspond to a state in the Markov chain we will formulate.

A Markov chain is characterized by an N * N transition probability matrix P each of whose entries is in the interval [0, 1]; the entries in each row of P add up to 1 i.e. each column containing destination node has probability as 1 / Out-links of source node.

4.1.1 Memoryless

In a Markov chain, the probability distribution of next states for a Markov chain depends only on the current state, and not on how the Markov chain arrived at the current state.

4.1.2 Ergodicity And Periodicity

For a Markov chain to be ergodic, two technical conditions are required of its states and the non-zero transition probabilities; these conditions are known as irreducibility and aperiodicity. Informally, the first

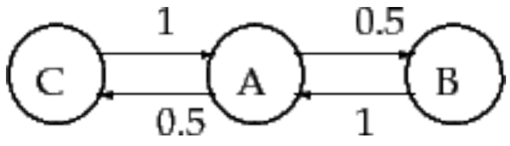


Figure 7: A Simple Markov Chain

$$\begin{pmatrix} 0 & 0.5 & 0.5 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Figure 8: Transition Probability Matrix of Markov Chain

ensures that there is a sequence of transitions of non-zero probability from any state to any other, while the latter ensures that the states are not partitioned into sets such that all state transitions occur cyclically from one set to another.

4.1.3 Applications

We can view a random surfer on the web graph as a Markov chain, with one state for each web page, and each transition probability representing the probability of moving from one web page to another. The teleport operation contributes to these transition probabilities.

4.2 Iterative Approach

Page Rank of say page A is as follows

PR(A) = (1-d) + d (PR(T1)/C(T1) + ... + PR(Tn)/C(Tn))

PR(Tn)
Each page has a notion of its own self-importance. Thats PR(T1) for the first page in the web all the way up to PR(Tn) for the last page.

C(Tn)
Each page spreads its vote out evenly amongst all of its outgoing links. The count, or number, of outgoing links for page 1 is C(T1), C(Tn) for page n, and so on for all pages.

PR(Tn)/C(Tn)
So if our page (page A) has a backlink from page n the share of the vote page A will get is PR(Tn)/C(Tn)

d
All these fractions of votes are added together but, to stop the other pages having too much influence, this total vote is damped down by multiplying it by 0.85 (the factor d)

(1 - d)
The (1 - d) bit at the beginning is a bit of probability math magic so the sum of all web pages' PageRanks will be one: it adds in the bit lost by the d. It also means that if a page has no links to it (no backlinks) even then it will still get a small PR of 0.15 (i.e. 1 - 0.85)

As we can see the page rank of page A is dependent on pages which point to *A*. Where *C(T1) ... C(TN)* represent the number of out links for the page *T1 to TN* i.e. pages which are in link for page *A*. So we calculate the page rank of page till the value converges i.e. difference between old and updated value is less than epsilon for all the pages.

Moreover since PageRank of each page is dependent on each other what should be initial value for pages so that we can start calculating PageRank values. It turns out that any positive value including zero works. It doesnt matter where you start your guess, once the PageRank calculations have settled down, the *normalized probability distribution* (the average PageRank for all pages) will be 1.0

4.3 The Stationary Distribution

4.3.1 Power Method

We first formulate the transition probability matrix from graph as shown below.

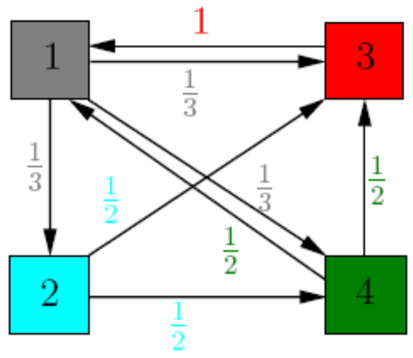


Figure 9: Directed Graph Example

$$\begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix}$$

Figure 10: Transition Probability Matrix

Suppose that initially the importance is uniformly distributed among the 4 nodes, each getting 1/4. Denote by v the initial rank vector, having all entries equal to 1/4. Each incoming link increases the importance of a web page, so at step 1, we update the rank of each page by adding to the current value the importance of the incoming links.

$$\mathbf{v} = \begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix}, \quad \mathbf{Av} = \begin{pmatrix} 0.37 \\ 0.08 \\ 0.33 \\ 0.20 \end{pmatrix}, \quad \mathbf{A}^2\mathbf{v} = \mathbf{A}(\mathbf{Av}) = \mathbf{A}\begin{pmatrix} 0.37 \\ 0.08 \\ 0.33 \\ 0.20 \end{pmatrix} = \begin{pmatrix} 0.43 \\ 0.12 \\ 0.27 \\ 0.16 \end{pmatrix}$$

$$\mathbf{A}^3\mathbf{v} = \begin{pmatrix} 0.35 \\ 0.14 \\ 0.29 \\ 0.20 \end{pmatrix}, \quad \mathbf{A}^4\mathbf{v} = \begin{pmatrix} 0.39 \\ 0.11 \\ 0.29 \\ 0.19 \end{pmatrix}, \quad \mathbf{A}^5\mathbf{v} = \begin{pmatrix} 0.39 \\ 0.13 \\ 0.28 \\ 0.19 \end{pmatrix}$$

$$\mathbf{A}^6\mathbf{v} = \begin{pmatrix} 0.38 \\ 0.13 \\ 0.29 \\ 0.19 \end{pmatrix}, \quad \mathbf{A}^7\mathbf{v} = \begin{pmatrix} 0.38 \\ 0.12 \\ 0.29 \\ 0.19 \end{pmatrix}, \quad \mathbf{A}^8\mathbf{v} = \begin{pmatrix} 0.38 \\ 0.12 \\ 0.29 \\ 0.19 \end{pmatrix}$$

Figure 11: PageRank Vector Calculation

This is the same as multiplying the matrix A with v. At step 1, the new importance vector is v1 = Av. We can iterate the process, thus at step 2, the updated importance vector is v2 = A(Av) = A**2v and this process goes on. We notice that after several iteration value of all node converges and we call result as PageRank vector of our web graph.

4.3.2 Eigen Vector Method

As noted from Markov chains we can represent our Web-Graph in link structure as stochastic matrix M.

The eigenvalues L and eigenvectors x of a matrix M satisfy the equation Mx = Lx where x != 0. We thus seek an eigenvector x with eigenvalue 1 for the matrix M. The matrix M for any Web-Graph must have 1 as an eigenvalue if the Web-Graph in question has no dangling nodes.

We compute the PageRank vector as
I = Identity Matrix of N*N
P = Teleportation Vector
s = 0.85
t = 1-s

PR = ***t*((I - s*M).I)*P***

4.4 Teleportation

Teleportation is an important step which helps in ensuring that random walker visits every node. The destination of a teleport operation is modeled as being chosen uniformly at random from all web pages. In graph of *N* nodes the teleport operation takes the surfer to each node with probability *1/N* including the present node.

Teleport operation is performed

1. When at a node with no out-links, the surfer invokes the teleport operation.
2. At any node that has outgoing links, the surfer invokes the teleport operation with probability A , that ranges as $0 < A < 1$.

4.5 Topic Specific Page Rank

To simulate topic specific Page Ranking we consider teleporting to a random web page chosen non-uniformly i.e. topic induced. By doing so, we are able to derive PageRank values tailored to particular interests.

Suppose our random surfer teleports to a random web page on the topic instead of teleporting to a uniformly chosen random web page. We will not focus on how we collect all web pages on the particular topic; in fact, we only need a non-zero subset S of topic-related web pages, so that the teleport operation is feasible.

Topic-specific PageRank.In this example we consider a user whose interests are 60% sports and 40% politics. If the teleportation probability is 10%, this user is modeled as teleporting 6% to sports pages and 4% to politics pages.

5 HITS (Hyperlink-Induced Topic Search)

The HITS, also known as Hub and Authority, is a procedure of finding the strongest authorities using user's query and Web-Graph.

5.1 Hubs and Authorities Definition

Good hubs are the web pages that pointing to good authoritative web pages. The strongest authorities are the web pages that contains the most relevant/important information about the query, and they are pointed by the strongest hubs.

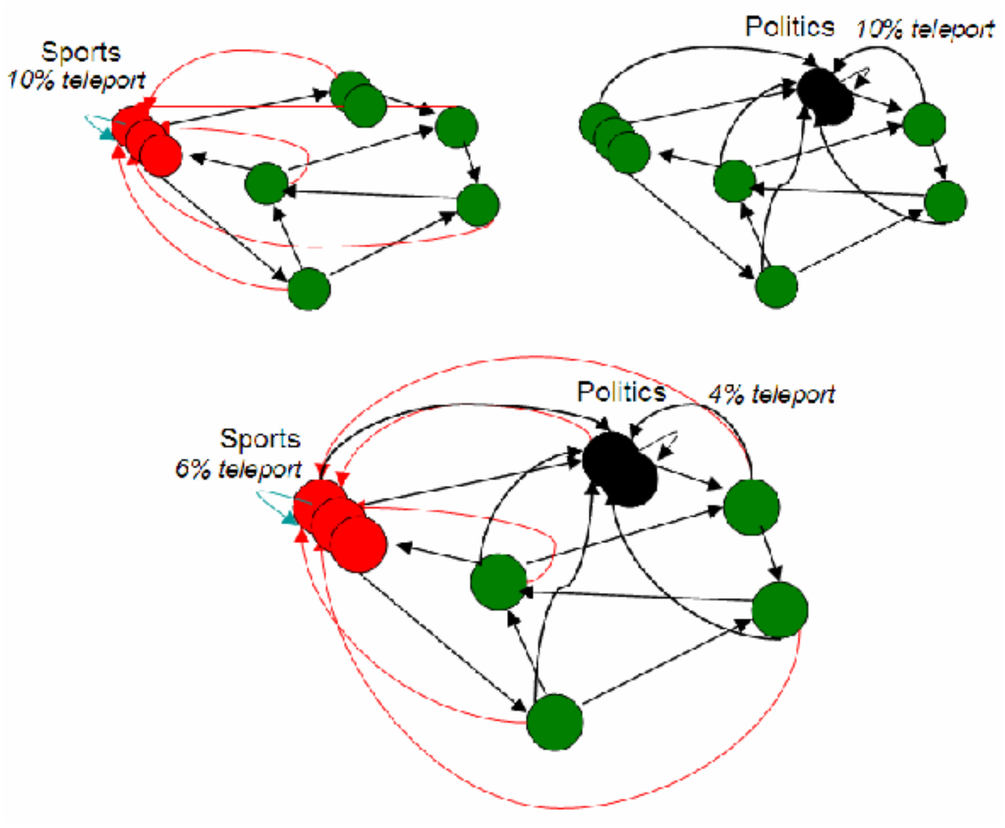


Figure 12: Topic Induced PageRank

5.2 HITS Procedure Overview

The HITS procedure is very similar to the PageRank procedure. The differences are that the HITS algorithm operates on a root set and every page in the root would need two scores (Hub and Authority scores). Therefore, your first step is to create a root set and then perform hub and authority scores update until they converge. The strongest authorities should have the highest authority scores and the strongest hubs should have the highest hub scores.

5.3 Base Set

The base set has three properties.

1. relatively small.
2. rich in relevant pages.
3. contains most of the strongest authorities.

The steps to obtain a base set:

1. Obtain top 1000 web pages as a root set using any IR function based on user's query, and add them to the base set
2. For each page p in the root set, obtain all pages that p points to and add them to the base set
3. For each page p in the root set, obtain a set of pages that pointing to p
   - If the size of the set is less than or equal to d, add all pages in the set to the base set

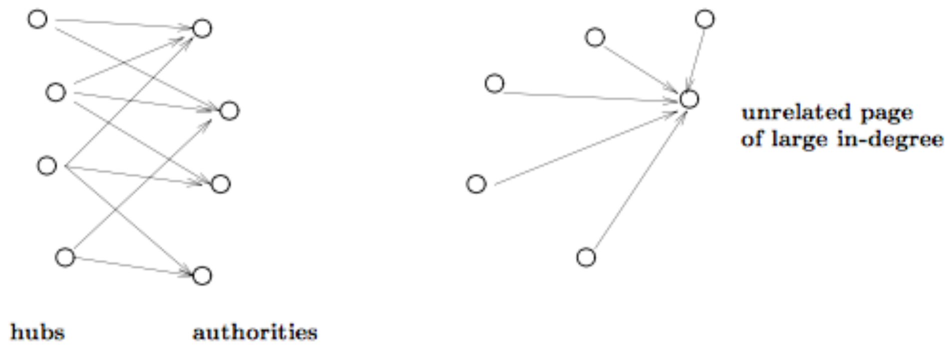


Figure 13: A densely linked set of hubs and authorities

  - If the size of the set is greater than d, add an arbitrary set of d pages from the set to the base set

4. Remove links (do not remove the pages) between pages that have the same domain name within the base set
    - e.g. *https://en.wikipedia.org/wiki/Google* has an outlink to *https://en.wikipedia.org/*, remove this link.

The first step ensures the size of the base set is small and it contains the most relavant web pages. The other two steps are trying to include the strongest authorities into the base set. The constant d can be 50 or others. The idea of d is trying to include more possibly strong hub pages into the base set while constraining the size of the base set. The last step is to remove in-domain links, because most of time, these links exist for navigation within the site.

5.4 Hubs and Authorities Update

For each web page, initialize its authority and hub scores to 1. Update both hub and authority scores until convergence.

- **Authorities Update:** Set each web page's authority score in the root set to the sum of the hub score of each web page that points to it
- **Hub Update:** Set each web pages's hub score in the root set to the sum of the authority score of each web page that it is pointing to

After every iteration, it is necessary to normalize the hub and authority scores.

- **Authority Score Norm:** Set each web page's authority score in the root set to $\sqrt{\sum_{p \in rootset} (p.authscore)^2}$
- **Hub Score Norm:** Set each web pages's hub score in the root set to $\sqrt{\sum_{p \in rootset} (p.hubscore)^2}$

6 Graph Visualization