

INTRODUCTION

► Why link analysis?

- The web is **not** just a collection of documents – its hyperlinks are important!
- A link from page *A* to page *B* may indicate:
 - *A* is related to *B*, or
 - *A* is recommending, citing, voting for or endorsing *B*
- Links are either
 - referential – *click here and get back home*, or
 - Informational – *click here to get more detail*
- Links effect the ranking of web pages and thus have commercial value.

INTRODUCTION

- ▶ A technique that use the graph structure in order to determine the relative importance of the nodes (web pages). One of the biggest changes in our lives in the decade following the turn of the century was the availability of efficient and accurate Web search, through search engines such as Google. While Google was not the first search engine, it was the first able to defeat the **spammers** who had made search almost useless.
- ▶ The innovation provided by Google was a nontrivial technological advance, called "**PageRank**."
- ▶ When PageRank was established as an essential technique for a search engine, spammers invented ways to manipulate the PageRank of a Web page, often called **link spam**.
- ▶ That development led to the response of **TrustRank** and other techniques for preventing spammers from attacking PageRank.

HISTORY OF SEARCH ENGINES & SPAM

- ▶ **Spamdexing** is the deliberate manipulation of search engine indexes. It involves a number of methods, such as repeating unrelated phrases, to manipulate the relevance or prominence of resources indexed, in a manner inconsistent with the purpose of the indexing system

- ▶ **Doorway pages**

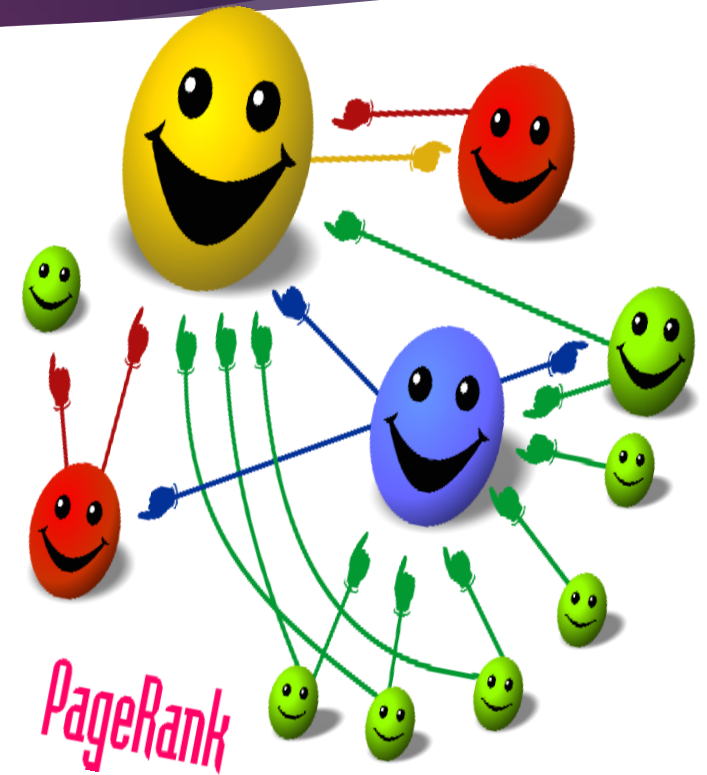
"Gateway" or doorway pages are low-quality web pages created with very little content, but are instead stuffed with very similar keywords and phrases.

- ▶ **Cloaking**

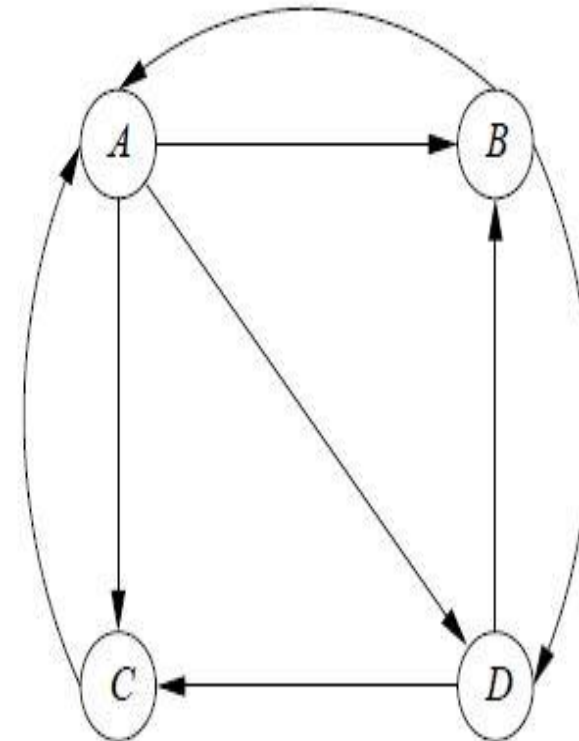
Cloaking refers to any of several means to serve a page to the search-engine spider that is different from that seen by human users. It can be an attempt to mislead search engines regarding the content on a particular web site.

PAGERANK

- ▶ PageRank is a function that assigns a real number to each page in the Web. The intent is that the higher the PageRank of a page, the more “important” it is. There is not one fixed algorithm for assignment of PageRank, and in fact, variations on the basic idea can alter the relative PageRank of any two pages



- Think of the Web as a directed graph, where pages are the nodes, and there is an arc from page p_1 to page p_2 if there are one or more links from p_1 to p_2 . Figure below is an example of a tiny version of the Web, where there are only four pages. Page **A** has links to each of the other three pages; page **B** has links to **A** and **D** only; page **C** has a link only to **A**, and page **D** has links to **B** and **C** only.

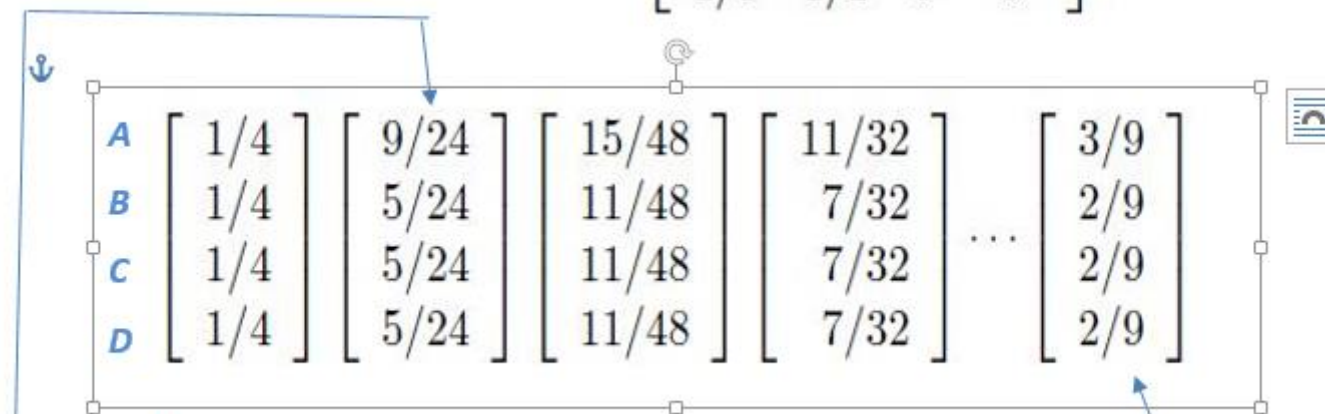


- ▶ Suppose a random surfer starts at page **A**. There are links to B, C, and D, so this surfer will next be at each of those pages with probability **1/3**, and has **zero** probability of being at A.
- ▶ A random surfer at **B** has, at the next step, probability **1/2** of being at A, **1/2** of being at D, and **0** of being at B or C.
- ▶ In general, we can define the transition (**stochastic**) **matrix** of the Web to describe what happens to random surfers after one step. This matrix **M** has **n rows and columns**, if there are **n pages**. The element M_{ij} in **row i** and **column j** has value $1/k$ if page j has k **arcs out**, and one of them is to page i. Otherwise, $M_{ij} = 0$.

$$M = \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$

- ▶ The power iteration is an eigenvalue algorithm: given a matrix \mathbf{M} , the algorithm will produce a number λ (the eigenvalue) and a nonzero vector \mathbf{v} (the eigenvector), such that $\mathbf{M}\mathbf{v} = \lambda\mathbf{v}$ To calculate the nodes probability.
- ▶ Suppose we apply, the process described above to the matrix M . Since there are four nodes, the initial vector \mathbf{v}_0 has four components, each $1/4$. The sequence of approximations to the limit that we get by multiplying at each step by M is:
- ▶ **$\text{Nodes}(N) = 4$**
- ▶ **$\text{Set } A, B, C, D \rightarrow 1 / N = 1 / 4 \text{ (initial vector)}$**

$$M = \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$



A	$\begin{bmatrix} 1/4 \end{bmatrix}$	$\begin{bmatrix} 9/24 \end{bmatrix}$	$\begin{bmatrix} 15/48 \end{bmatrix}$	$\begin{bmatrix} 11/32 \end{bmatrix}$	\dots	$\begin{bmatrix} 3/9 \end{bmatrix}$
B	$\begin{bmatrix} 1/4 \end{bmatrix}$	$\begin{bmatrix} 5/24 \end{bmatrix}$	$\begin{bmatrix} 11/48 \end{bmatrix}$	$\begin{bmatrix} 7/32 \end{bmatrix}$	\dots	$\begin{bmatrix} 2/9 \end{bmatrix}$
C	$\begin{bmatrix} 1/4 \end{bmatrix}$	$\begin{bmatrix} 5/24 \end{bmatrix}$	$\begin{bmatrix} 11/48 \end{bmatrix}$	$\begin{bmatrix} 7/32 \end{bmatrix}$	\dots	$\begin{bmatrix} 2/9 \end{bmatrix}$
D	$\begin{bmatrix} 1/4 \end{bmatrix}$	$\begin{bmatrix} 5/24 \end{bmatrix}$	$\begin{bmatrix} 11/48 \end{bmatrix}$	$\begin{bmatrix} 7/32 \end{bmatrix}$	\dots	$\begin{bmatrix} 2/9 \end{bmatrix}$

$$A = (1/4 * 0) + (1/4 * 1/2) + (1/4 * 1) + (1/4 * 0)$$

$$= 1/8 + 1/4$$

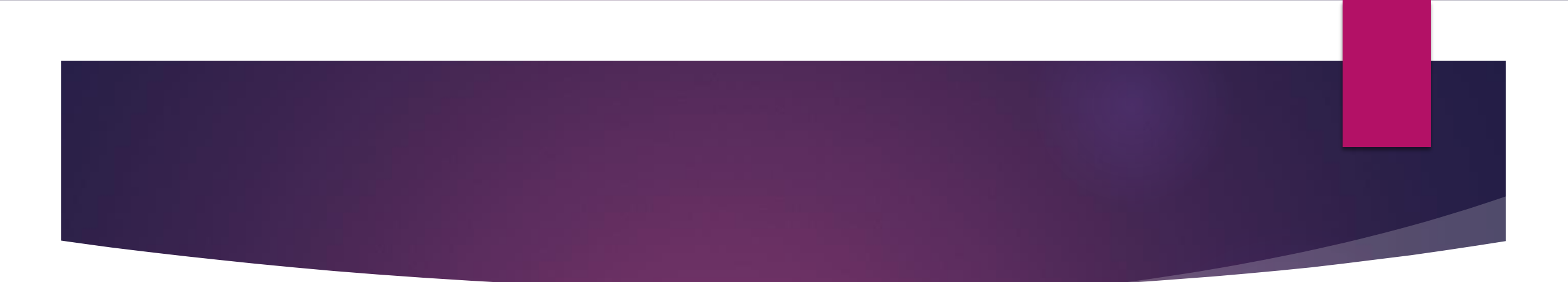
$$= 9/24$$

$$B = (1/4 * 1/3) + (1/4 * 0) + (1/4 * 0) + (1/4 * 1/2)$$

$$= 1/12 + 1/8$$

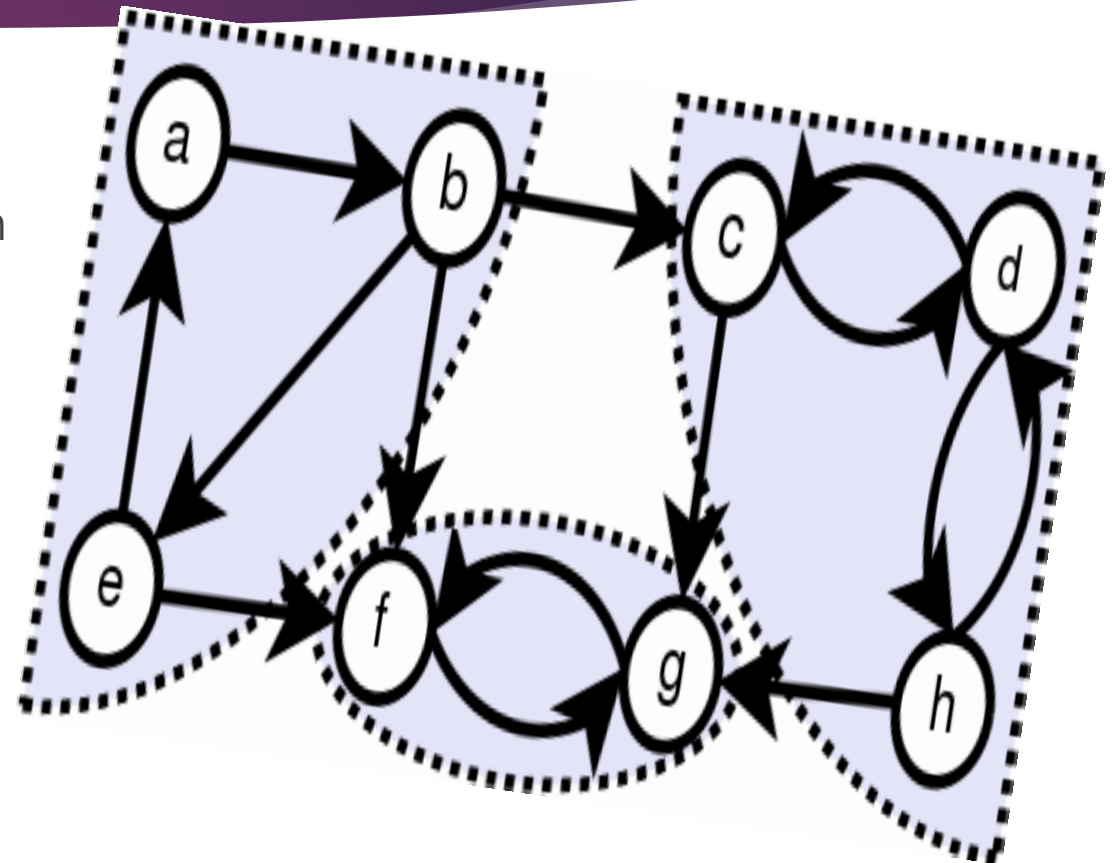
$$= 5/24$$

PageRank for A, B, C and D

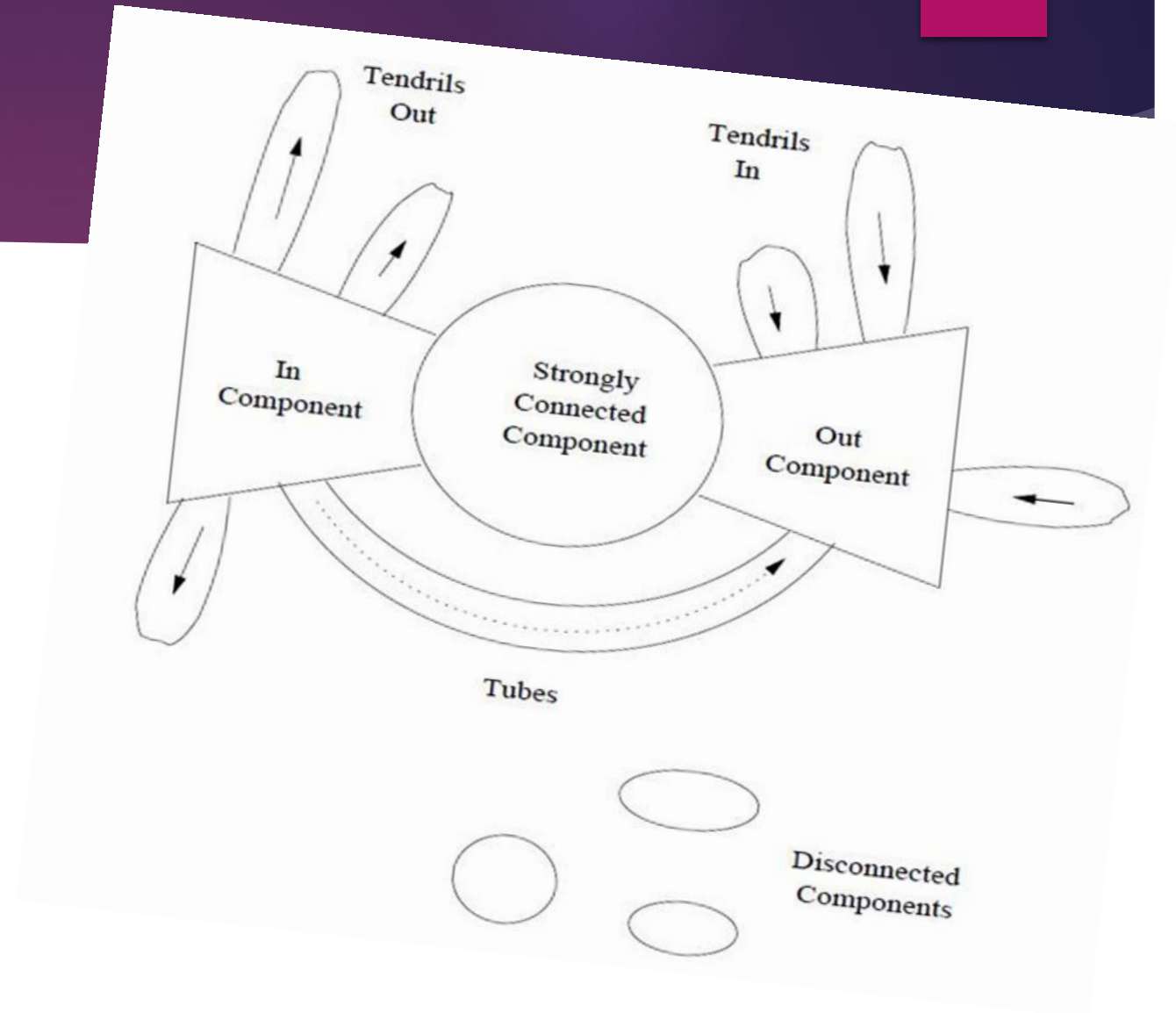
- 
- ▶ Notice that in this example, the probabilities for B, C, and D remain the same. It is easy to see that B and C must always have the same values at any iteration, because their rows in M are identical.
 - ▶ To show that their values are also the same as the value for D, an inductive proof works.
 - ▶ Given that the last three values of the limiting vector must be the same, it is easy to discover the limit of the above sequence.
 - ▶ The first row of M tells us that the probability of A must be $\frac{3}{2}$ the other probabilities, so the limit has the probability of A equal to $\frac{3}{9}$, or $\frac{1}{3}$, while the probability for the other three nodes is $\frac{2}{9}$.

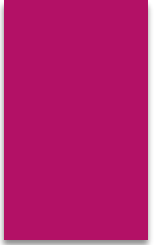
STRUCTURE OF THE WEB

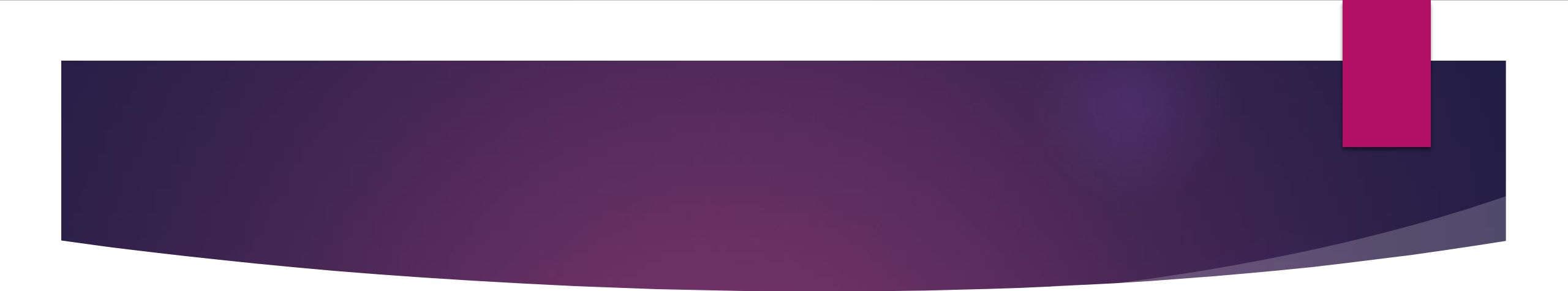
- It would be nice if the Web were strongly connected, when every node is reachable from every other nodes.
However, it is not, in practice.



- An early study of the Web found it to have the structure shown in figure below. There was a large strongly connected component (**SCC**), but there were several other portions that were almost as large.



- 
- ▶ 1. The **in-component**, consisting of pages that could reach the SCC by following links, but were not reachable from the SCC.
 - ▶ 2. The **out-component**, consisting of pages reachable from the SCC but unable to reach the SCC.
 - ▶ 3. **Tendrils**, which are of two types. Some tendrils consist of ***pages reachable from the in-component but not able to reach the in-component***. The other tendrils can ***reach the out-component, but are not reachable from the out-component***.
 - ▶ In addition, there were small numbers of pages found either in
 - ▶ (a) **Tubes**, which are pages reachable from the in-component and able to reach the out-component, but unable to reach the SCC or be reached from the SCC.
 - ▶ (b) **Isolated components** that are unreachable from the large components (the SCC, in- and out-components) and unable to reach those components.

- 
- ▶ There are really two problems we need to avoid.
 - ▶ First is the dead end, a page that has no links out. Surfers reaching such a page disappear, and the result is that in the limit no page that can reach a dead end can have any PageRank at all.
 - ▶ **The second problem is groups of pages** that all have outlinks but they never link to any other pages. These structures are called ***spider traps***.
 - ▶ Both these problems are solved by a method called “**taxation**,” where we assume a random surfer has a finite probability of leaving the Web at any step, and new surfers are started at each page.



TOPIC SENSITIVE PageRank

DRAWBACKS OF EXISTING ALGORITHMS

- ▶ HITS Algorithm :
 - ▶ ☹ Expensive at runtime
 - ▶ ☹ Scores are calculated using subgraph of the entire Web graph
- ▶ PageRank Algorithm :
 - ▶ ☹ Query independent rank score
 - ▶ ☹ Random surfer model not appropriate in some situations
 - ▶ ☹ Prone to manipulations (Google bombs, link farms...)

What is Topic Sensitive PageRank ?

- ▶ Personalized PageRank (Taher H. Haveliwala, Stanford Univ, 2003)
- ▶ What was the personalization?
 - ▶ Instead of computing a single rank vector, why don't we compute a **set** of rank vectors, **one for each (basis) topic**?
- ▶ Two Steps
 - ▶ Pre-processing (Offline like PageRank)
 - ▶ Query processing

Steps of Topic Sensitive PageRank

- ▶ Pre-processing
 - ▶ Fixing topics for which PageRank vectors are needed. Open directories or ontologies can be used. Google uses query logs to identify important topics
 - ▶ Pick a teleport set for each vector and compute the TSPR vector for that topic.
 - ▶ Teleport set :
 - ▶ You don't need an exact sports page. Even a page which has a link to a Sports page is fine. You can just teleport from there!

Steps of Topic Sensitive PageRank

- ▶ Query Processing
 - ▶ Guess relevant set of topics from the query to the search engine.
 - ▶ Combine topic sensitive scores with respect to relevant topics to get final ranking of pages.

Topic-Sensitive PageRank Algorithm

- ▶ PageRank formula :
 - ▶ $r = \text{PR}(G)$
- ▶ Topic-Sensitive PageRank formula :
 - ▶ $r = \text{IPR}(G, \mathbf{v})$
- ▶ IPR stands for “Influenced” PageRank
- ▶ Input :
 - ▶ Web graph $G = (V, E)$
 - ▶ Influence vector is a vector of basis topics \mathbf{t}
- ▶ Output :
 - ▶ List of rank vectors \mathbf{r}
- ▶ It maps page i to :
 - ▶ page i importance, WRT topic t_i

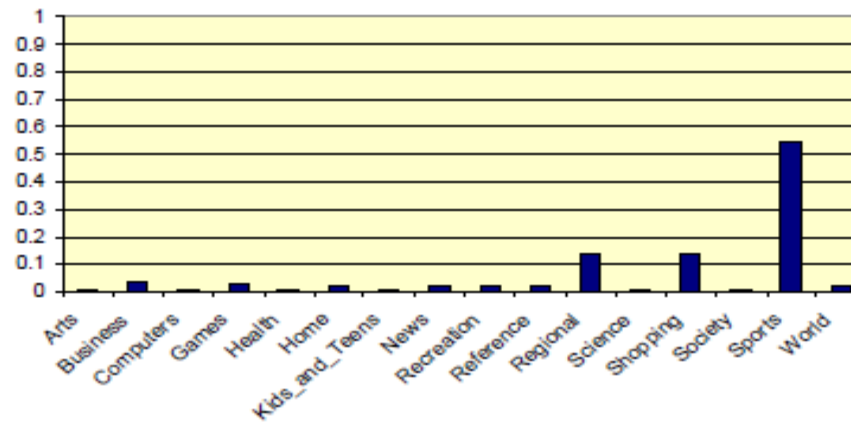
Topic-Sensitive PageRank Algorithm

- ▶ For the sake of simplicity, let's consider some page i and only 16 topics (categories) :
 - ▶ We can pick them from the first level of ODP
- ▶ Step 1 is performed once, offline, during Web crawl
- ▶ It uses the following iterative approach :

```
For each topic  $c_j \in v$ 
{
    // Part 1 : Calc  $v_j$ 
     $v_j[i] = 0$ ;
    if (  $i \in \text{pages}(c_j)$  ) {
         $v_j[i] = 1 / \text{num}(\text{pages}(c_j))$ 
    }
    // Part 2 : Calc  $r_j$ 
     $r_j[i] = \text{IPR}(W, v_j[i])$ ;
}
```

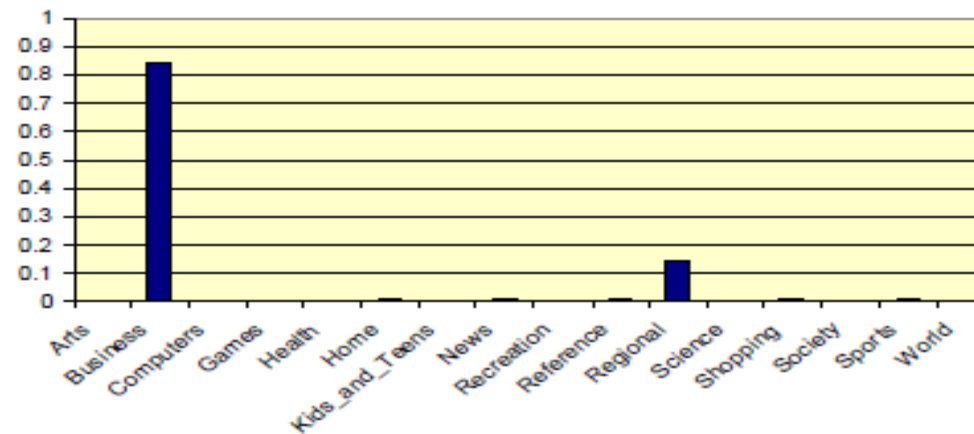
Topic-Sensitive PageRank Algorithm

- ▶ Step 2 assumes that we calculate some distribution of weights over the 16 topics in our basis
- ▶ Only the link structure of pages relevant to the query topic will be used to rank page i
- ▶ Example : Query is “golf”
- ▶ With no additional context, the distribution of topic weights we would use is :



Topic-Sensitive PageRank Algorithm

- ▶ If user issues queries about investment opportunities, a follow-up query on “golf” should be ranked differently, with the business-specific rank vector
- ▶ Example : Query is “golf”, but the previous query was “financial services investments”
- ▶ Distribution of topic weights we would use is :



Topic-Sensitive PageRank Algorithm

- ▶ At the end, calculate the composite PageRank score using the following formula :

$$s_d = \sum_j w_j r_j[d]$$

- ▶ Interpretation of the composite score :

$$\sum_j [w_j \cdot \text{IPR}(W, \mathbf{v}_j)] = \text{IPR}(W, \sum_j [w_j \cdot \mathbf{v}_j])$$

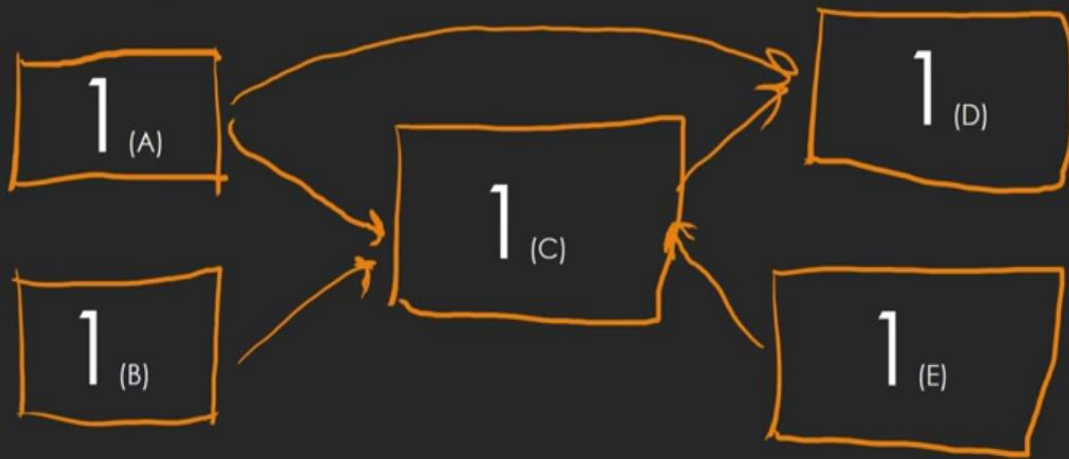
- ▶ Weighted sum of rank vectors itself forms a valid rank vector
- ▶ The final score can be used in conjunction with other scoring schemes

ADVANTAGES

- ▶ ☺ Query-specific rank score
- ▶ ☺ Fully automated
- ▶ ☺ Makes use of context
- ▶ ☺ Still inexpensive at runtime
- ▶ ☺ Developed with Web 3.0 Link Analysis and NLP in mind

Link Spam : Pagerank

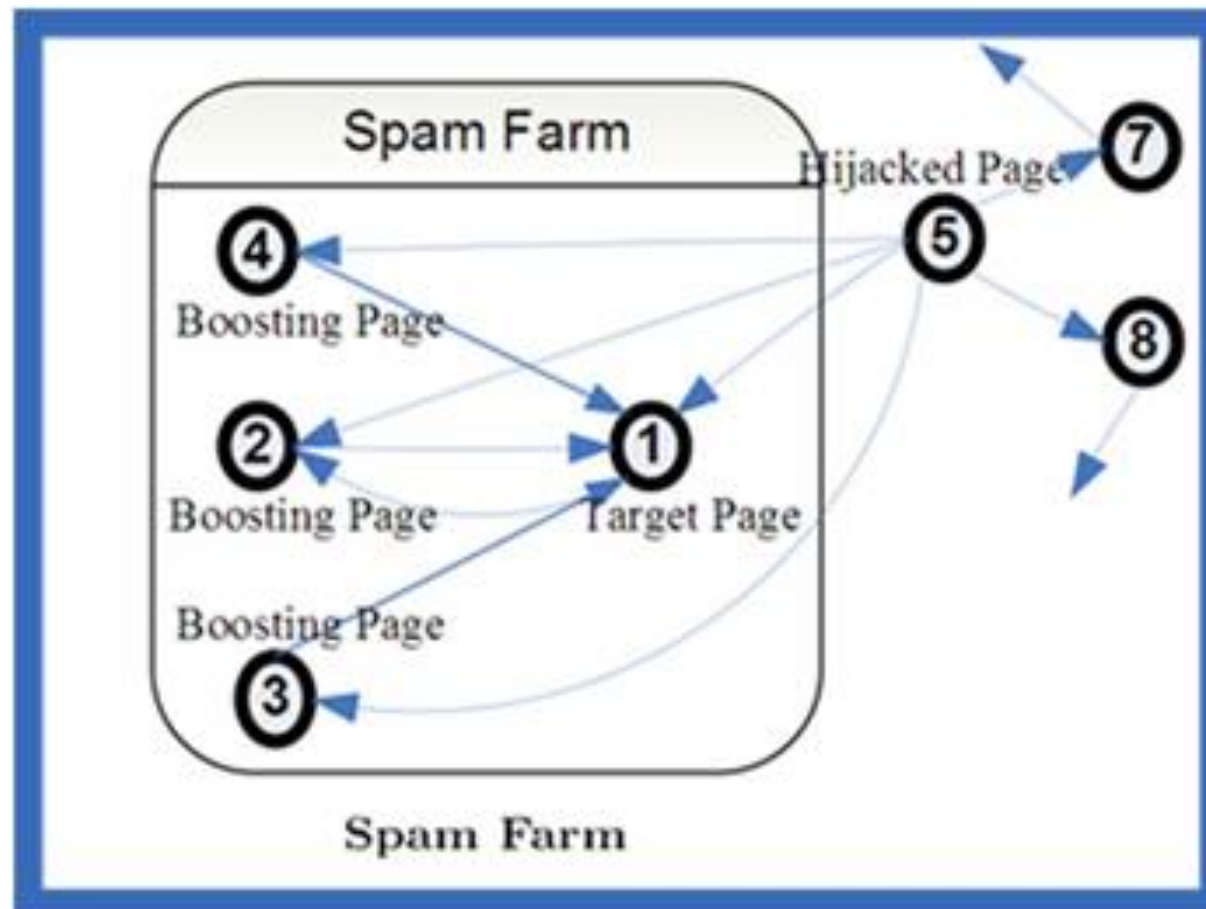
Lets Give Each Page 1 Vote



Next Lets Allow the PageRank to Flow



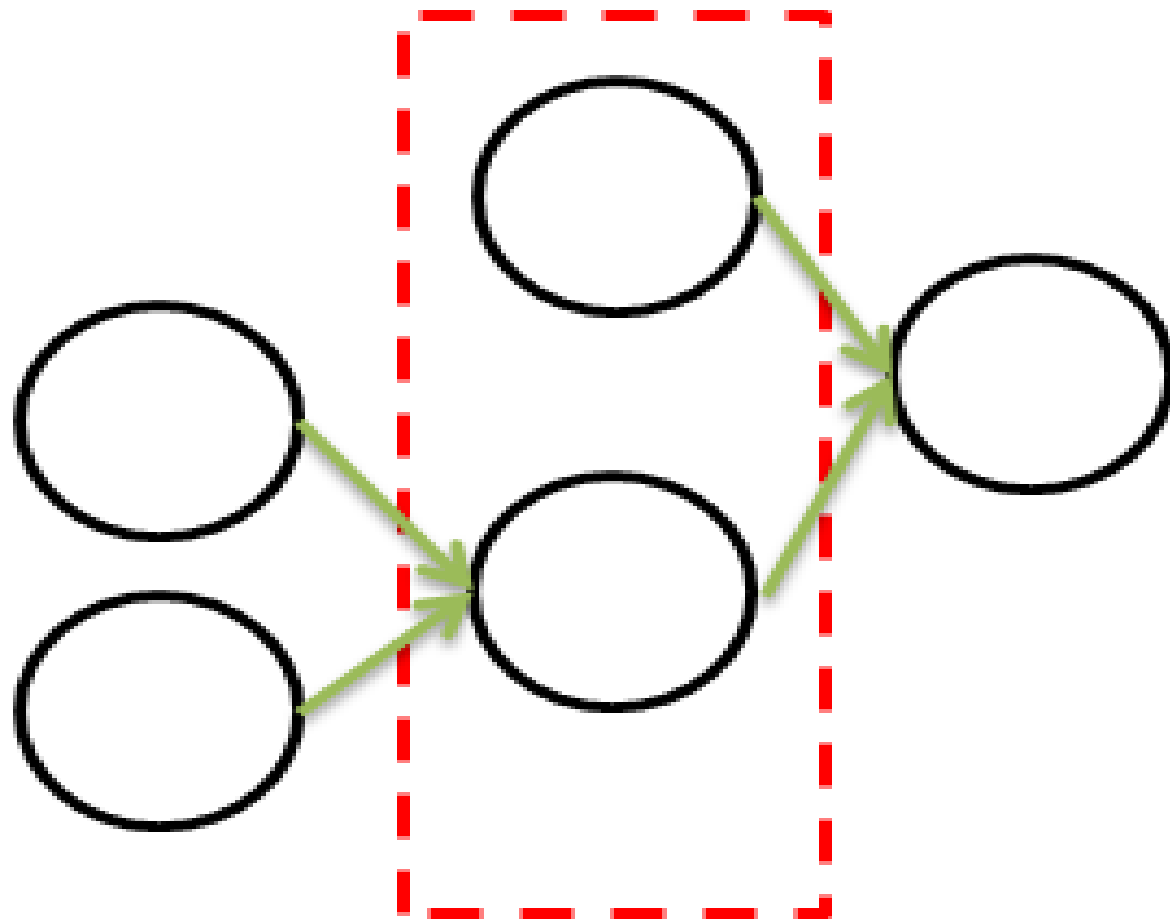
Link Spam : Spam Farm



Combating techniques

- ▶ Truncated PageRank
- ▶ Supporters
- ▶ Type of links
- ▶ Link Threshold

Truncated PageRank

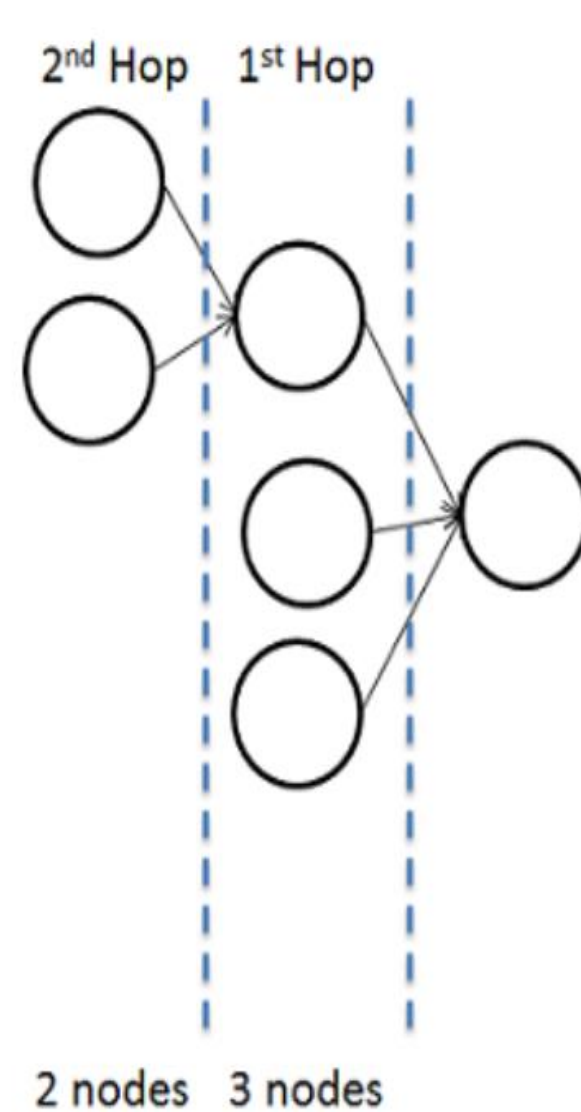
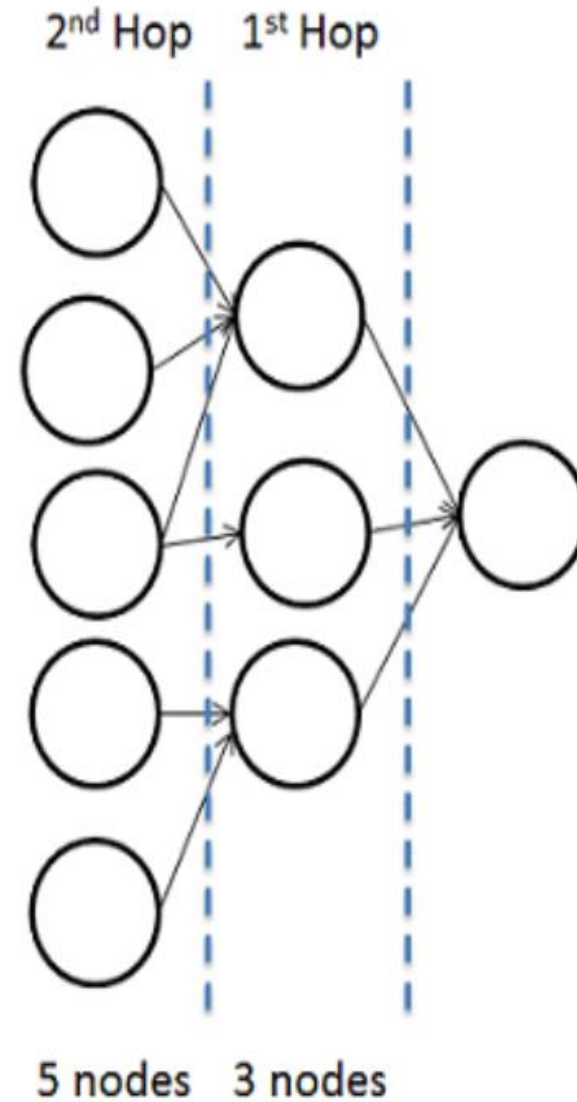


Removes the direct contribution
of the first levels of links

Counting Supporters

More Connected

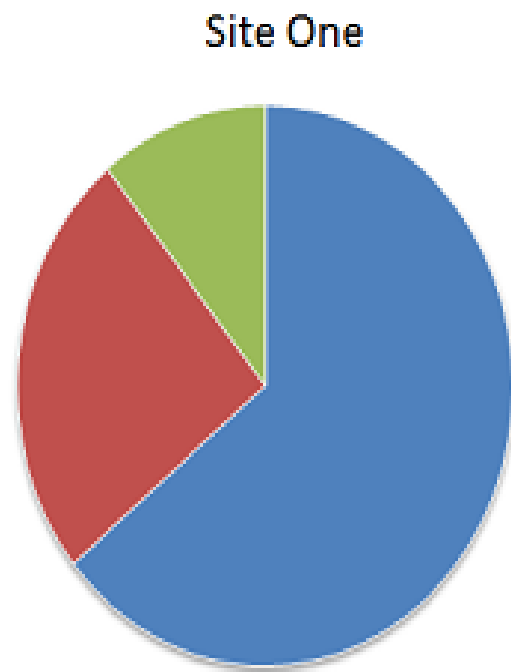
Less Connected



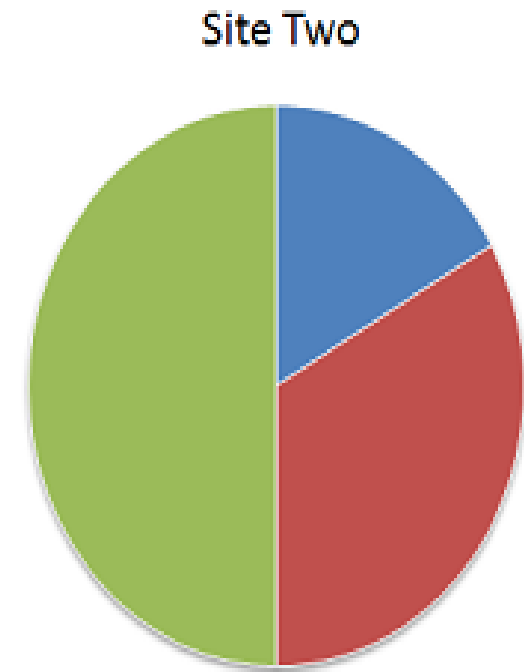


Types of links

Link threshold



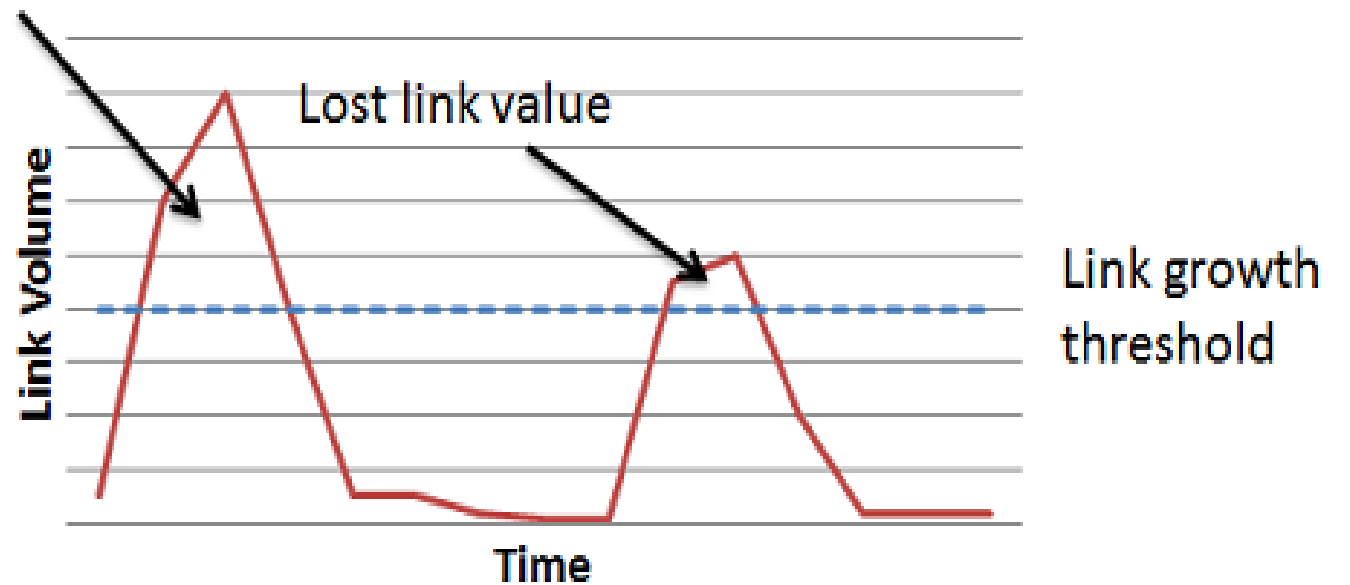
Owned
Accessible
Inaccessible



Owned
Accessible
Inaccessible

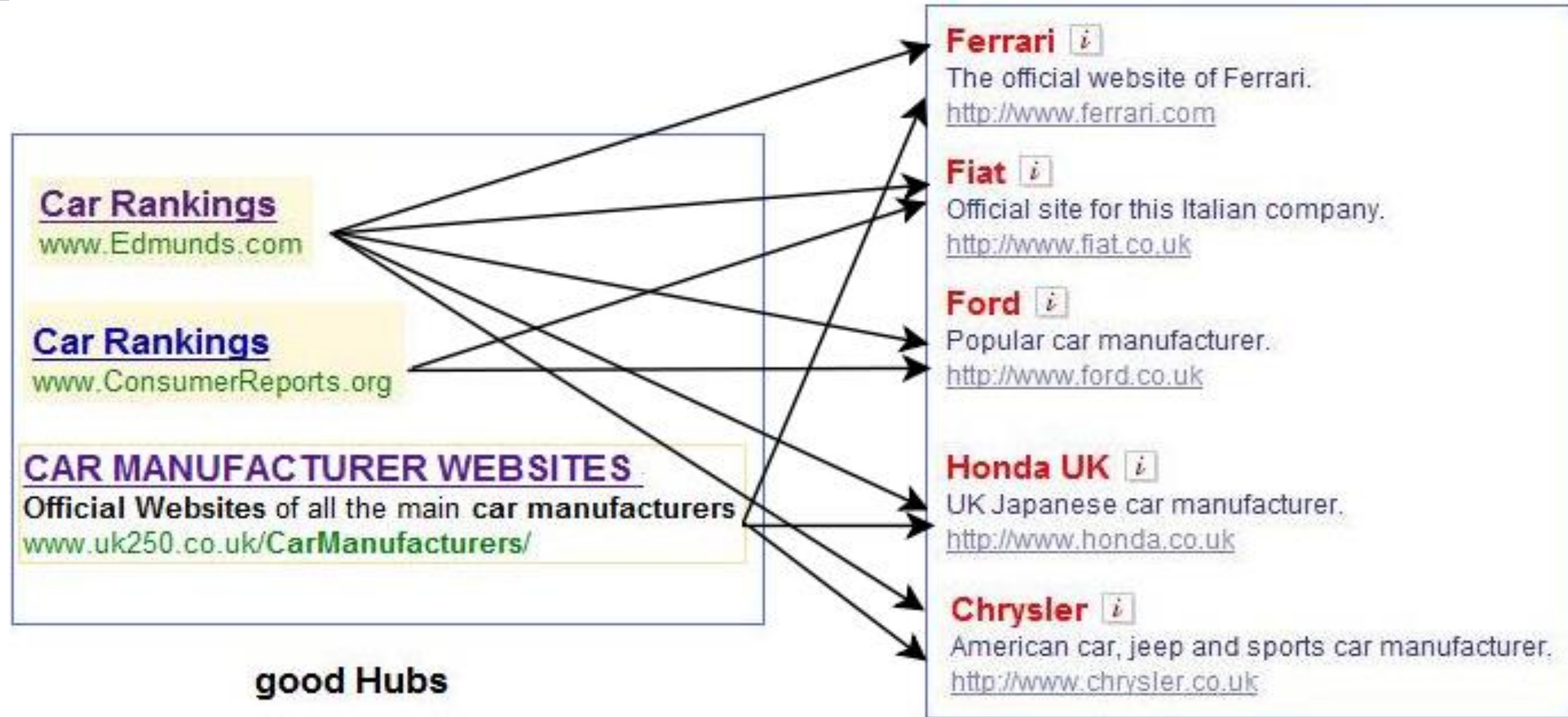
Lost link value

Link Threshold



Hubs & Authorities: Concept

- ▶ **AUTHORITY:** Page containing useful information
Ex: Home page of any valid website
- ▶ **HUB:** Pages that link to **AUTHORITIES**
Ex: List of 10/n- best websites for “xyz”



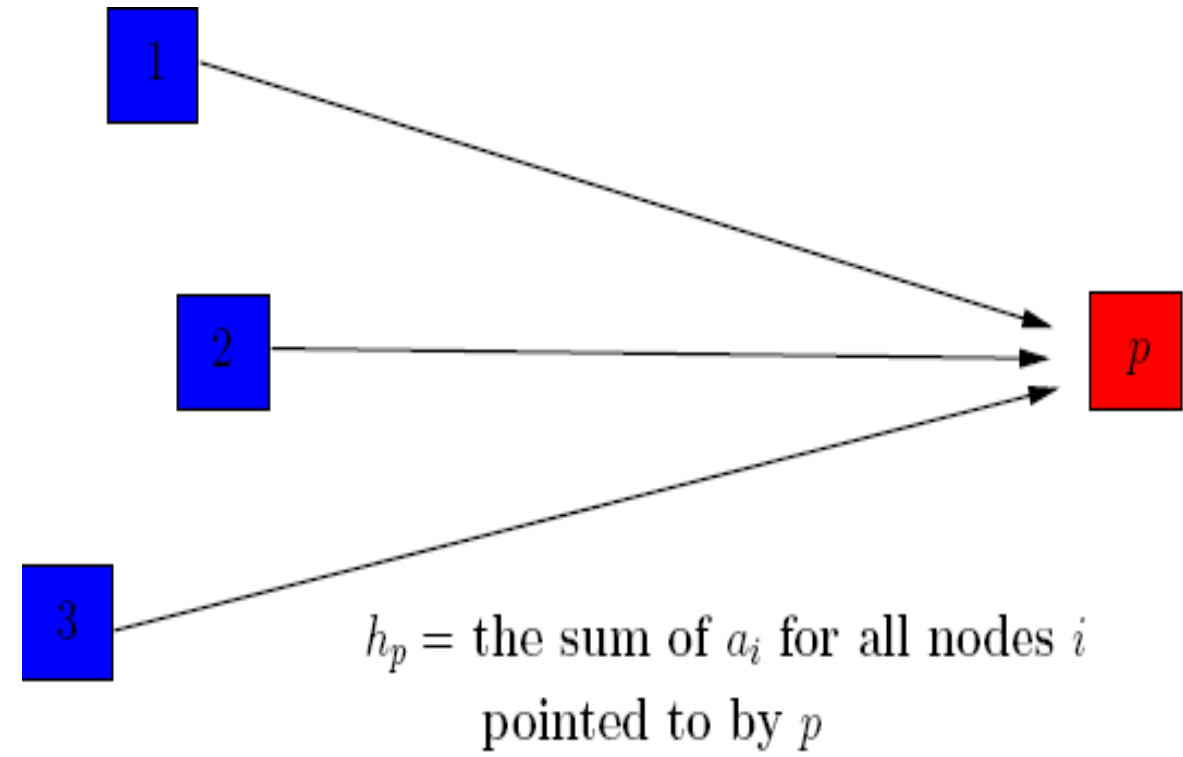
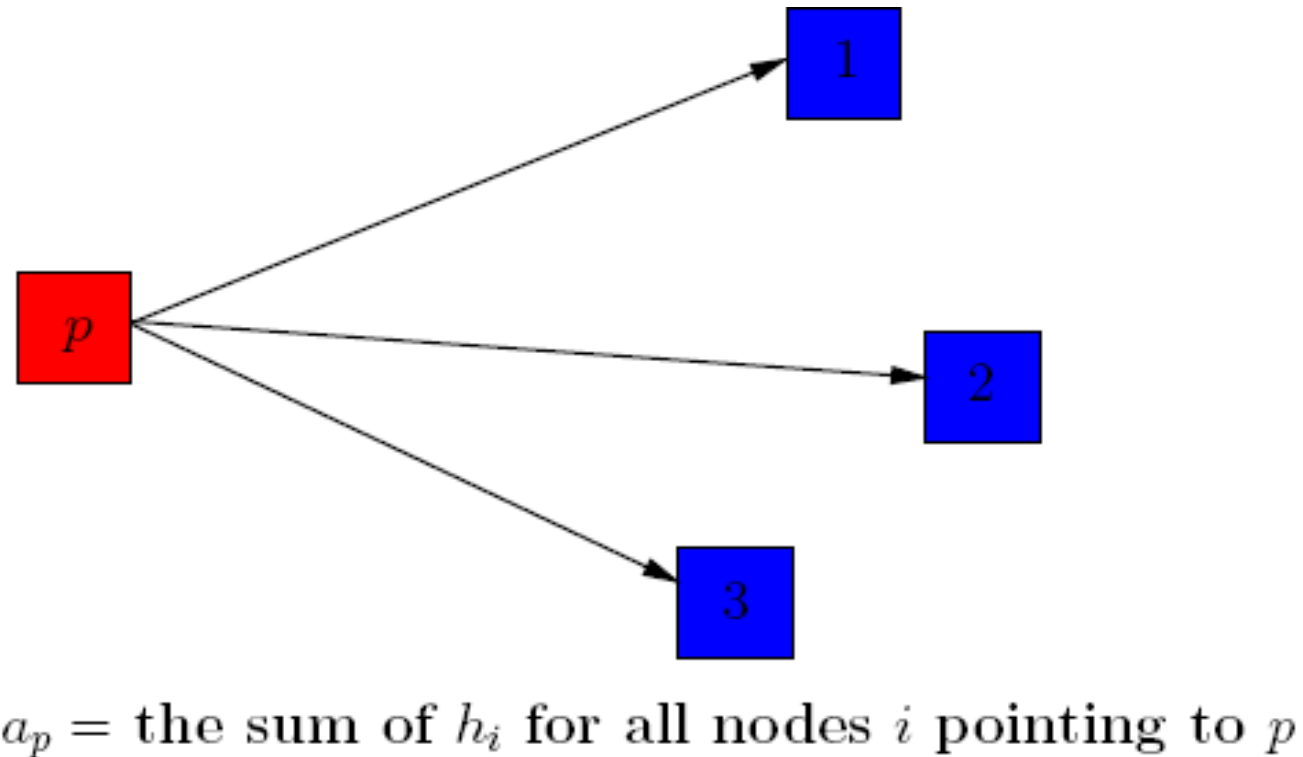
Query: **Top automobile makers**

Hubs & Authorities: HITS Algorithm

- ▶ **H**yperlink *href tags in html*
- ▶ **I**nduced *inductive programming*
- ▶ **T**opic *query entity*
- ▶ **S**earch *operation*

STEP 1 : The I operation

STEP 2 : The O operation



STEP 3 : Normalization

Repeat above steps for 5+ iterations until the authority and hub scores converge

	ADVANTAGE:	
	1.Topic based Search	

	DISADVANTAGE:	
	1.Query dependent 2. Cannot detect ads 3. Suffers from topic drift	