

Link Analysis II

Alexander Schönhuth

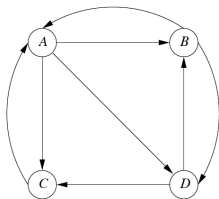


Bielefeld University
June 25, 2020

LEARNING GOALS TODAY

- ▶ PageRank Reality: Structure of the Web
- ▶ Topic-Sensitive PageRank: Classify Pages by Topics
- ▶ Link Spam and TrustRank: Fight Advanced Spammer Strategies
- ▶ Hubs and Authorities: Alternative, Non-PageRank Approach

PAGERANK: REMINDER



Hypothetical web with four pages

Adopted from mmds.org

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

Transition matrix for web from slides before

Adopted from mmds.org

PAGERANK: REMINDER

- ▶ Let p^0 be a start distribution for a “random web surfer” over all web pages
 - ▶ If n is number of web pages, then usually $p^0 = (1/n, \dots, 1/n) \in \mathbb{R}^n$
- ▶ Random steps of the surfer are reflected by repeated application of the web transition matrix M to the distribution over the pages of the surfer:

$$M^t p_0 \xrightarrow[t \rightarrow \infty]{} \bar{p}$$

- ▶ The limiting distribution \bar{p} , for which $M\bar{p} = \bar{p}$, expresses the importance of web pages

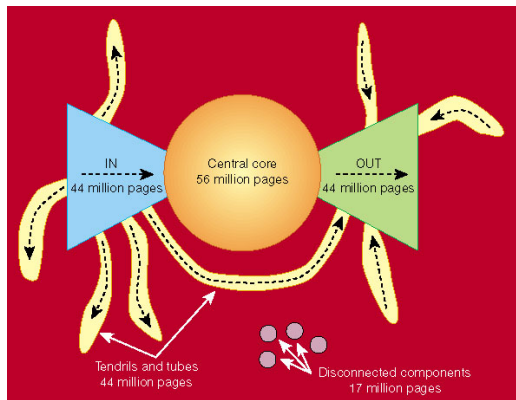
$$\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix}, \begin{bmatrix} 9/24 \\ 5/24 \\ 5/24 \\ 5/24 \end{bmatrix}, \begin{bmatrix} 15/48 \\ 11/48 \\ 11/48 \\ 11/48 \end{bmatrix}, \begin{bmatrix} 11/32 \\ 7/32 \\ 7/32 \\ 7/32 \end{bmatrix}, \dots, \begin{bmatrix} 3/9 \\ 2/9 \\ 2/9 \\ 2/9 \end{bmatrix}$$

Convergence to limiting distribution for four-node web graph

Adopted from mmds.org

PageRank Reality
Dead Ends and Spider Traps

STRUCTURE OF THE WEB



Bowtie picture of the web

Adopted from mmds.org

WEB BOWTIE: SUMMARY

- ▶ *Strongly connected component (SCC)*: core of the web
- ▶ *In-component (IC)*:
 - ▶ One can reach SCC from IC
 - ▶ but not return to IC once left
- ▶ *Out-component (OC)*:
 - ▶ Can be reached from SCC
 - ▶ but no longer be left
- ▶ *Tendrils*:
 - ▶ *First type*: reachable from IC, but can no longer be left
 - ▶ *Second type*: can reach OC, but cannot be returned to
- ▶ *Tubes*:
 - ▶ Can be reached from IC
 - ▶ Can only reach OC
- ▶ *Isolated components* are not reachable from and cannot reach other components

BOWTIE AND MARKOV CHAINS

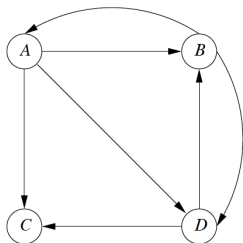
Issue: Limiting Distribution

- ▶ Random surfers will inevitably wind up in out-component
- ▶ Limiting distribution has probability 0 on IC and SCC
 - ☞ No page in IC or SCC of importance

PageRank Modification

- ▶ Avoid *dead ends*, single pages with no outlinks
- ▶ Avoid *spider traps*, sets of pages without dead ends, but no arcs out
- ▶ *Solution: Taxation*
 - ▶ Assume random surfer has small probability to leave the web
 - ▶ Instead, new surfer starts at random node of the web

DEAD ENDS



Web graph with dead end (node C)

Adopted from mmds.org

- ▶ Dead end = columns of all zeroes in the web transition matrix M
- ▶ M then is *substochastic* (= column sums at most 1)
- ▶ $M^i v$ yields vector with zeroes for certain components
- ▶ Dead ends *drain out* the web

DEAD ENDS

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

Transition matrix for web with dead end (node C)

Adopted from `mmds.org`

$$\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix}, \begin{bmatrix} 3/24 \\ 5/24 \\ 5/24 \\ 5/24 \end{bmatrix}, \begin{bmatrix} 5/48 \\ 7/48 \\ 7/48 \\ 7/48 \end{bmatrix}, \begin{bmatrix} 21/288 \\ 31/288 \\ 31/288 \\ 31/288 \end{bmatrix}, \dots, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Corresponding limiting distribution

Adopted from `mmds.org`

AVOIDING DEAD ENDS

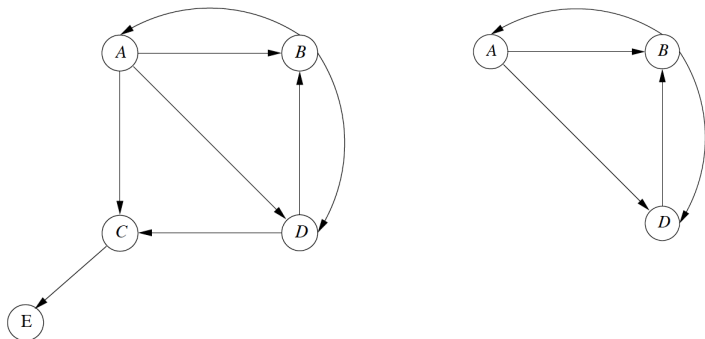
Dropping dead ends: Procedure

- ▶ Drop dead ends from graph, and corresponding edges
- ▶ Dropping dead ends may create more dead ends
- ▶ Keep dropping dead ends iteratively

Dropping dead ends: Consequences

- ▶ Removes parts of out-component, tendrils and tubes
- ▶ Leaves SCC and in-component

AVOIDING DEAD ENDS



Graph before (left) and after iterative removal of dead ends (right)

DROPPING DEAD ENDS: PAGERANK COMPUTATION

1. After iterative removal of dead ends, compute PageRank for remaining core nodes
2. Re-introduce nodes iteratively, in reverse order relative to their removal
3. PageRank for re-introduced node: sum over all predecessors, PageRank of predecessor p divided by the number of successors of p

DEAD ENDS

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

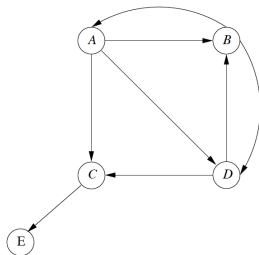
Transition matrix after removal of dead ends

$$\begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}, \begin{bmatrix} 1/6 \\ 3/6 \\ 2/6 \end{bmatrix}, \begin{bmatrix} 3/12 \\ 5/12 \\ 4/12 \end{bmatrix}, \begin{bmatrix} 5/24 \\ 11/24 \\ 8/24 \end{bmatrix}, \dots, \begin{bmatrix} 2/9 \\ 4/9 \\ 3/9 \end{bmatrix}$$

PageRank(A) = 2/9, PageRank(B) = 4/9, PageRank(D) = 3/9

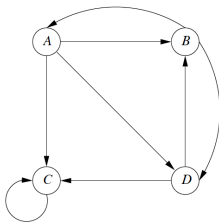
Adopted from mmds.org

DEAD ENDS: PAGERANK COMPUTATION



1. From core: $\text{PageRank}(A) = 2/9$, $\text{PageRank}(B) = 4/9$, $\text{PageRank}(D) = 3/9$
2. Re-introduce node C first:
$$\text{PageRank}(C) = 1/3 \times \text{PageRank}(A) + 1/2 \times \text{PageRank}(D) = \frac{13}{54}$$
3. Then re-introduce node E: $\text{PageRank}(E) = 1 \times \text{PageRank}(C) = \frac{13}{54}$

SPIDER TRAPS



Web graph with spider trap (set containing single node C)

Adopted from mmds.org

- ▶ (Small) group of nodes with no dead ends, but no arcs out
- ▶ Can appear intentionally or unintentionally
- ▶ “Soak up” all PageRank

SPIDER TRAPS

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

Transition matrix for web with single node spider trap (third column)

Adopted from mmds.org

$$\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix}, \begin{bmatrix} 3/24 \\ 5/24 \\ 11/24 \\ 5/24 \end{bmatrix}, \begin{bmatrix} 5/48 \\ 7/48 \\ 29/48 \\ 7/48 \end{bmatrix}, \begin{bmatrix} 21/288 \\ 31/288 \\ 205/288 \\ 31/288 \end{bmatrix}, \dots, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Corresponding limiting distribution

Adopted from mmds.org

SPIDER TRAPS: TAXATION

- ▶ Allow the random surfer to get *teleported* to a random page
- ▶ *Notation:*
 - ▶ Let n be the total number of web pages
 - ▶ Let $\mathbf{e} := (1, \dots, 1)$ be the vector of length n with all entries one
 - ▶ Let β be a small constant; usually $0.8 \leq \beta \leq 0.9$
- ▶ *Taxation:* In each matrix-vector multiplication iteration, instead of just computing $\mathbf{v}' = M\mathbf{v}$, compute

$$\mathbf{v}' = \beta M\mathbf{v} + \frac{1}{n}(1 - \beta)\mathbf{e} \quad (1)$$

to obtain a new vector \mathbf{v}' from the actual one \mathbf{v}

SPIDER TRAPS: TAXATION

- ▶ *Taxation:* In each matrix-vector multiplication iteration, instead of just computing $\mathbf{v}' = M\mathbf{v}$, compute

$$\mathbf{v}' = \beta M\mathbf{v} + \frac{1}{n}(1 - \beta)\mathbf{e}$$

to obtain a new vector \mathbf{v}' from the actual one \mathbf{v}

- ▶ *Interpretation:*
 - ▶ With probability β , the surfer follows an out-link
 - ▶ With probability $1 - \beta$, the surfer get teleported to a random page
 - ▶ In dead ends, surfer disappears with probability β
 - ☞ when dead ends exist, sum of entries in \mathbf{v}' less than one

SPIDER TRAPS

$$\mathbf{v}' = \begin{bmatrix} 0 & 2/5 & 0 & 0 \\ 4/15 & 0 & 0 & 2/5 \\ 4/15 & 0 & 4/5 & 2/5 \\ 4/15 & 2/5 & 0 & 0 \end{bmatrix} \mathbf{v} + \begin{bmatrix} 1/20 \\ 1/20 \\ 1/20 \\ 1/20 \end{bmatrix}$$

Iteration with taxation, with spider trap (third column)

Adopted from mmds.org

$$\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix}, \begin{bmatrix} 9/60 \\ 13/60 \\ 25/60 \\ 13/60 \end{bmatrix}, \begin{bmatrix} 41/300 \\ 53/300 \\ 153/300 \\ 53/300 \end{bmatrix}, \begin{bmatrix} 543/4500 \\ 707/4500 \\ 2543/4500 \\ 707/4500 \end{bmatrix}, \dots, \begin{bmatrix} 15/148 \\ 19/148 \\ 95/148 \\ 19/148 \end{bmatrix}$$

Corresponding limiting distribution

Adopted from mmds.org

PAGERANK: EFFICIENT COMPUTATION

- ▶ PageRank virtually is matrix-vector multiplication
 - ▶ Consider MapReduce techniques (originally motivated by PageRank)
- ▶ *Caveats*, however:
 - ▶ Transition matrix M is very sparse; consider appropriate representation of M
 - ▶ To reduce communication cost, use combiners
 - ▶ Earlier striping technique not sufficient
- ▶ So, additional techniques necessary:

see <https://mmds.org>, section 5.2

Topic-Sensitive PageRank

TOPIC-SENSITIVE PAGERANK: MOTIVATION

- ▶ Different people have different interests, but ...
- ▶ ... different interests are expressed by identical terms
 - ▶ E.g. `jaguar` may refer to animal, automobile, operating system, game console
- ▶ *Ideally*: Each user has private PageRank vector that measures individual importance of pages
- ▶ *But*: It is not feasible to store a vector of length many billions for one billion users

TOPIC-SENSITIVE PAGERANK: BASIC IDEA

- ▶ Identify a (rather small) number of topics
- ▶ Compute topic specific PageRank vectors
 - ▶ Store topic vectors ...
 - ▶ ... instead of individual user vectors
 - ▶ There are much less topic vectors
 - ▶ *Example for useful topics:* See <https://www.dmoz.org> for 16 top-level categories
- ▶ Assign users to (weighted combination of) topic vectors
- ▶ *Drawback:* Looses accuracy
- ▶ *Benefit:* Saves massive amounts of space

TOPIC-SENSITIVE PAGERANK: COMPUTATION

Idea: Biased Random Walks

- ▶ Simulate random surfers that are to prefer pages adhering to particular topics
- ▶ Random surfers start at approved topic-specific pages only
- ▶ When surfing, they will preferably visit pages linked from topic-specific pages
- ▶ Such pages are likely to deal with topic as well
- ▶ When being re-introduced (to avoid dead ends, spider traps), surfers again start at approved pages

TOPIC-SENSITIVE PAGERANK: DEFINITION

- ▶ Let S be the *teleport set*, i.e. the pages that are approvedly topic-specific
- ▶ Let $n, \mathbf{v}, \mathbf{v}', M, \beta$ be as before
- ▶ Let $\mathbf{e}_S \in \{0, 1\}^n$ be a bit vector of length n such that

$$\mathbf{e}_S[i] = \begin{cases} 1 & \text{if } i\text{-th page belongs to } S \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

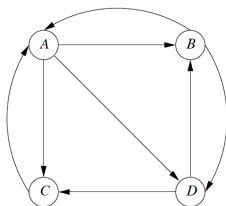
DEFINITION [TOPIC-SENSITIVE PAGERANK]

The *topic-sensitive PageRank* for S is the limit of the iteration

$$\mathbf{v}' = \beta M \mathbf{v} + (1 - \beta) \frac{\mathbf{e}_S}{|S|} \quad (3)$$

where $|S|$ is the cardinality (size) of S .

TOPIC-SENSITIVE PAGERANK: EXAMPLE



Example web graph

Adopted from mmds.org

$$\beta M = \begin{bmatrix} 0 & 2/5 & 4/5 & 0 \\ 4/15 & 0 & 0 & 2/5 \\ 4/15 & 0 & 0 & 2/5 \\ 4/15 & 2/5 & 0 & 0 \end{bmatrix}$$

Corresponding weighted web transition matrix

Adopted from mmds.org

TOPIC-SENSITIVE PAGERANK: EXAMPLE II

$$\mathbf{v}' = \begin{bmatrix} 0 & 2/5 & 4/5 & 0 \\ 4/15 & 0 & 0 & 2/5 \\ 4/15 & 0 & 0 & 2/5 \\ 4/15 & 2/5 & 0 & 0 \end{bmatrix} \mathbf{v} + \begin{bmatrix} 0 \\ 1/10 \\ 0 \\ 1/10 \end{bmatrix}$$

Topic sensitive PageRank computation iteration for teleport set {B,D}

Adopted from mmds.org

$$\begin{bmatrix} 0/2 \\ 1/2 \\ 0/2 \\ 1/2 \end{bmatrix}, \begin{bmatrix} 2/10 \\ 3/10 \\ 2/10 \\ 3/10 \end{bmatrix}, \begin{bmatrix} 42/150 \\ 41/150 \\ 26/150 \\ 41/150 \end{bmatrix}, \begin{bmatrix} 62/250 \\ 71/250 \\ 46/250 \\ 71/250 \end{bmatrix}, \dots, \begin{bmatrix} 54/210 \\ 59/210 \\ 38/210 \\ 59/210 \end{bmatrix}$$

Corresponding limiting distribution

Adopted from mmds.org

TOPIC-SENSITIVE PAGERANK: PRACTICAL CONSIDERATIONS

- ▶ Pick an appropriate set of topics
- ▶ For each topic selected, determine teleport set
- ▶ *Classifying documents by topic*
 - ▶ Has been studied in great detail
 - ▶ Topics are characterized by words relating to topic
 - ▶ Such words appear surprisingly often in topic-specific pages
 - ▶ Determine such words from pages known to relate to topic beforehand
 - ▶ Remember the TF.IDF measure (first lecture)

TOPIC-SENSITIVE PAGERANK: PRACTICAL CONSIDERATIONS

- ▶ When confronted with search query, decide on related topics
- ▶ *Determining user-specific topics:*
 - ▶ Allow user to choose from menu
 - ▶ Infer topics from words appearing in recent queries
 - ▶ Infer topics from information on user (bookmarks, stated interests in social media,...)
- ▶ Use corresponding topic-sensitive PageRank vectors for ranking responses

Link Spam

LINK SPAM: INTRODUCTION

- ▶ Google rendered *term spam ineffective*
- ▶ Spammers developed *link spam* as a technique to artificially increase PageRank
- ▶ In the following, understand how to
 - ▶ create link spam
 - ▶ and how to fight it

SPAMMER VIEW OF WEB

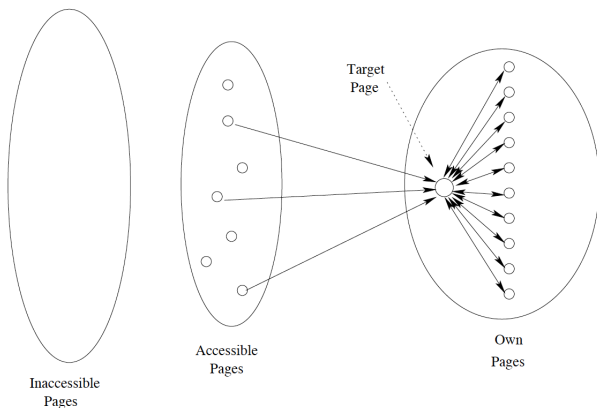
Types of pages

- ▶ *Inaccessible pages*: cannot be accessed by spammer; majority of pages
- ▶ *Accessible pages*: not owned, but can be accessed (manipulated)
 - ☞ Blogs, newspapers, forums allow leaving comments with links
- ▶ *Own pages*: owned and fully controlled by spammer

Spam farm

- ▶ Part of own pages with
 - ▶ *target page t* , for which maximum PageRank is to be achieved
 - ▶ *supporting pages m* , with links from and to t
- ▶ Note that without links from outside, spam farm would be useless

SPAMMER VIEW OF WEB



Spammer view: types of pages and spam farm

Adopted from mmds.org

SPAM FARM: ANALYSIS

- ▶ Let there be n web pages overall
- ▶ Let $\beta \in [0.8, 0.9]$ be the taxed fraction of PageRank
- ▶ Let there be a spam farm with target page t and m supporting pages
- ▶ Let $\text{In}(t)$ be all pages with a link to t ; $\text{PR}(p)$ be the PageRank for a page p ; $\text{Out}(p)$ be all successors of $p \in P$
- ▶ Let

$$x = \beta \sum_{p \in \text{In}(t)} \frac{\text{PR}(p)}{|\text{Out}(p)|}$$

be the PageRank provided to t by accessible pages

- ▶ Let $y = \text{PR}(t)$ be the unknown PageRank of t
- ▶ The PageRank of each supporting page is

$$\beta \frac{y}{m} + \frac{(1-\beta)}{n}$$

where $\beta \frac{y}{m}$ is due to t and $\frac{(1-\beta)}{n}$ is due to random teleporting

SPAM FARM: ANALYSIS

- ▶ Let $y = \text{PR}(t)$ be the unknown PageRank of t
- ▶ Let x be the PageRank provided to t by accessible pages
- ▶ Let $\beta \frac{y}{m} + \frac{(1-\beta)}{n}$ be the PageRank from each supporting page

Solving for y

1. We compute

$$y = x + \beta m \left(\frac{\beta y}{m} + \frac{1-\beta}{n} \right) = x + \beta^2 y + \beta(1-\beta) \frac{m}{n} \quad (4)$$

2. This yields

$$y = \frac{x}{1-\beta^2} + c \frac{m}{n} \quad (5)$$

where $c = \beta(1-\beta)/(1-\beta^2) = \beta/(1+\beta)$

Example: $\beta = 0.85$, so $1/(1-\beta^2) = 3.6$ and $c = 0.46$; spam farm has amplified external contribution to t by 360%; t also obtains 46% of the fraction m/n

COMBATING LINK SPAM

War on spam farms

- ▶ Search engines identify spam farm structures and eliminate pages from their index
- ▶ Spammers create alternative structures that raise PageRank of target pages
- ▶ Search engines in turn eliminate those structures, too
- ▶ ...
- ▶ Endless war between search engines and spammers

Systematic approaches

- ▶ *TrustRank*: Variation on topic-sensitive PageRank to lower score of spam pages
- ▶ *Spam mass*: Calculation that identifies pages likely to be spam
 - ✎ Eliminate such pages or lower their PageRank substantially

TRUSTRANK

- ▶ *TrustRank* is like topic-sensitive PageRank where the “topic” are pages believed to be “trustworthy”
 - ▶ Inaccessible pages belong to the topic
 - ▶ Accessible pages like blogs or newspapers are only borderline trustworthy
- ▶ Choosing trustworthy pages:
 1. Human picked pages, or pages of highest PageRank (not achievable by link spam)
 2. Pick pages trustworthy by domain, such as .edu, .ac.uk, .gov and so on

SPAM MASS

DEFINITION [SPAM MASS]

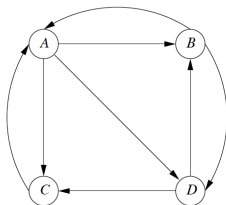
- ▶ For a page p , let $r(p)$ and $t(p)$ be its PageRank and its TrustRank
- ▶ The *spam mass* of p is defined to be

$$\frac{(r(p) - t(p))}{r(p)}$$

EXPLANATION

- ▶ Negative or small spam mass indicates that p is not spam
- ▶ Spam mass close to 1 indicates that p is likely to be spam

SPAM MASS: EXAMPLE



Example web graph; B and D are trusted pages

Adopted from mmds.org

Node	PageRank	TrustRank	Spam Mass
<i>A</i>	3/9	54/210	0.229
<i>B</i>	2/9	59/210	-0.264
<i>C</i>	2/9	38/210	0.186
<i>D</i>	2/9	59/210	-0.264

Corresponding page rank, trust rank and spam mass

Adopted from mmds.org

Hubs and Authorities

HUBS AND AUTHORITIES: INTRODUCTION

- ▶ The hubs-and-authorities algorithm, also called *HITS* (*hyperlink-induced topic search*), is an alternative to PageRank
- ▶ *Similarities:*
 - ▶ Quantifies importance of pages
 - ▶ Involved fixedpoint computation by iterative matrix-vector multiplication
- ▶ *Differences:*
 - ▶ Divides pages into hubs and authorities
 - ▶ Not a preprocessing step: ranks importance of responses to query

HITS: INTUITION

- ▶ Importance is twofold
- ▶ *Authorities* are pages deemed to be valuable because they provide information on a topic
 - ▶ E.g. course website at university
- ▶ *Hubs* are pages deemed to be valuable because of providing directions about topics
 - ▶ E.g. department directory providing links to all course websites
- ▶ Mutually recursive definition:
 - ▶ *Good hub* links to good authorities
 - ▶ *Good authority* is linked to by good hubs

HUBBINESS AND AUTHORITY: DEFINITION

DEFINITION [HUBBINESS, AUTHORITY]

- ▶ Let the number of webpages be n
- ▶ Let $\mathbf{h} \in \mathbb{R}^n$, $\mathbf{a} \in \mathbb{R}^n$ be two vectors where
 - ▶ \mathbf{h}_i quantifies the goodness of page i as a hub
 - ▶ \mathbf{a}_i quantifies the goodness of page i as an authority
- ▶ \mathbf{h}_i is also referred to as *hubbiness* of page i

REMARK

- ▶ Values of \mathbf{h} , \mathbf{a} are generally scaled such that
 - ▶ *either* the largest component is 1
 - ▶ *or* the sum of components is 1
 - ▶ In the following, first option will be used here

LINK MATRIX: DEFINITION

DEFINITION [LINK MATRIX]

- ▶ Let the number of webpages be n
- ▶ The *link matrix* $L \in \{0, 1\}^{n \times n}$ of the Web is defined by

$$L_{ij} = \begin{cases} 1 & \text{there is a link from page } i \text{ to page } j \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

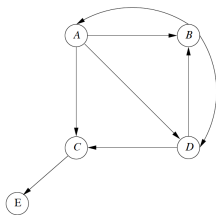
- ▶ Its transpose L^T is defined by $L_{ij}^T = L_{ji}$, that $L_{ij}^T = 1$ if there is a link from the j -th to the i -th page, and zero otherwise

REMARK

- ▶ L^T is similar to the PageRank web matrix M insofar as

$$L_{ij}^T \neq 0 \quad \text{if and only if} \quad M_{ij} \neq 0$$

LINK MATRIX: EXAMPLE



Example web graph

Adopted from mmds.org

$$L = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad L^T = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Corresponding link matrix and its transpose

Adopted from mmds.org

HUBS AND AUTHORITIES: FORMAL RELATIONSHIP

- ▶ Good hub links to good authorities:

$$\mathbf{h}_i = \lambda \sum_{j=1}^n L_{ij} \mathbf{a}_j \quad \text{or, equivalently} \quad \mathbf{h} = \lambda L \mathbf{a} \quad (7)$$

where λ represents the necessary scaling of \mathbf{h}

- ▶ Good authority is linked to by good hubs:

$$\mathbf{a}_i = \mu \sum_{j=1}^n L_{ij}^T \mathbf{h}_j \quad \text{or, equivalently} \quad \mathbf{a} = \mu L^T \mathbf{h} \quad (8)$$

where μ represents the necessary scaling of \mathbf{a} .

HUBS AND AUTHORITIES: FORMAL RELATIONSHIP

- ▶ Substituting (8) into (7) yields:

$$\mathbf{h} = \lambda\mu LL^t \mathbf{h} \quad (9)$$

- ▶ Substituting (7) into (8) yields:

$$\mathbf{a} = \mu\lambda L^T L \mathbf{a} \quad (10)$$

- ▶ \mathbf{h}, \mathbf{a} can be determined by solving linear equations
- ▶ *However:* $LL^T, L^T L$ are not sufficiently sparse for their size to allow for solving corresponding linear equations
- ▶ *Solution:* HITS algorithm

THE HITS ALGORITHM

Initialization: Set $\mathbf{h}_i = 1$ for all i , that is $\mathbf{h} = (1, \dots, 1)$

Iteration:

1. Compute

$$\mathbf{a} = L^T \mathbf{h}$$

2. Scale such that largest component of \mathbf{a} is 1

3. Compute

$$\mathbf{h} = L \mathbf{a}$$

4. Scale such that largest component of \mathbf{h} is 1

5. Repeat until convergence

HITS ALGORITHM: EXAMPLE

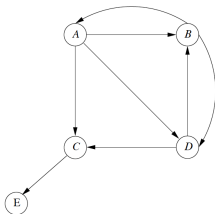
$$\begin{array}{ccccc} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 1 \\ 2 \\ 2 \\ 2 \\ 1 \end{bmatrix} & \begin{bmatrix} 1/2 \\ 1 \\ 1 \\ 1 \\ 1/2 \end{bmatrix} & \begin{bmatrix} 3 \\ 3/2 \\ 1/2 \\ 2 \\ 0 \end{bmatrix} & \begin{bmatrix} 1 \\ 1/2 \\ 1/6 \\ 2/3 \\ 0 \end{bmatrix} \\ \mathbf{h} & L^T \mathbf{h} & \mathbf{a} & L \mathbf{a} & \mathbf{h} \end{array}$$

$$\begin{array}{cccc} \begin{bmatrix} 1/2 \\ 5/3 \\ 5/3 \\ 3/2 \\ 1/6 \end{bmatrix} & \begin{bmatrix} 3/10 \\ 1 \\ 1 \\ 9/10 \\ 1/10 \end{bmatrix} & \begin{bmatrix} 29/10 \\ 6/5 \\ 1/10 \\ 2 \\ 0 \end{bmatrix} & \begin{bmatrix} 1 \\ 12/29 \\ 1/29 \\ 20/29 \\ 0 \end{bmatrix} \\ L^T \mathbf{h} & \mathbf{a} & L \mathbf{a} & \mathbf{h} \end{array}$$

First two iterations of HITS algorithm

Adopted from mmds.org

HITS ALGORITHM: EXAMPLE



A and D are good hubs, B and C are good authorities

Adopted from mmds.org

$$\mathbf{h} = \begin{bmatrix} 1 \\ 0.3583 \\ 0 \\ 0.7165 \\ 0 \end{bmatrix} \quad \mathbf{a} = \begin{bmatrix} 0.2087 \\ 1 \\ 1 \\ 0.7913 \\ 0 \end{bmatrix}$$

Limits of \mathbf{h} , \mathbf{a} on graph

Adopted from mmds.org

MATERIALS / OUTLOOK

- ▶ See *Mining of Massive Datasets*, chapter 4.1–4.4
- ▶ As usual, see <http://www.mmds.org/> in general for further resources
- ▶ Next lecture: “Mining Data Streams II / PageRank I”
 - ▶ See *Mining of Massive Datasets* 4.5–4.7; 5.1–5.2