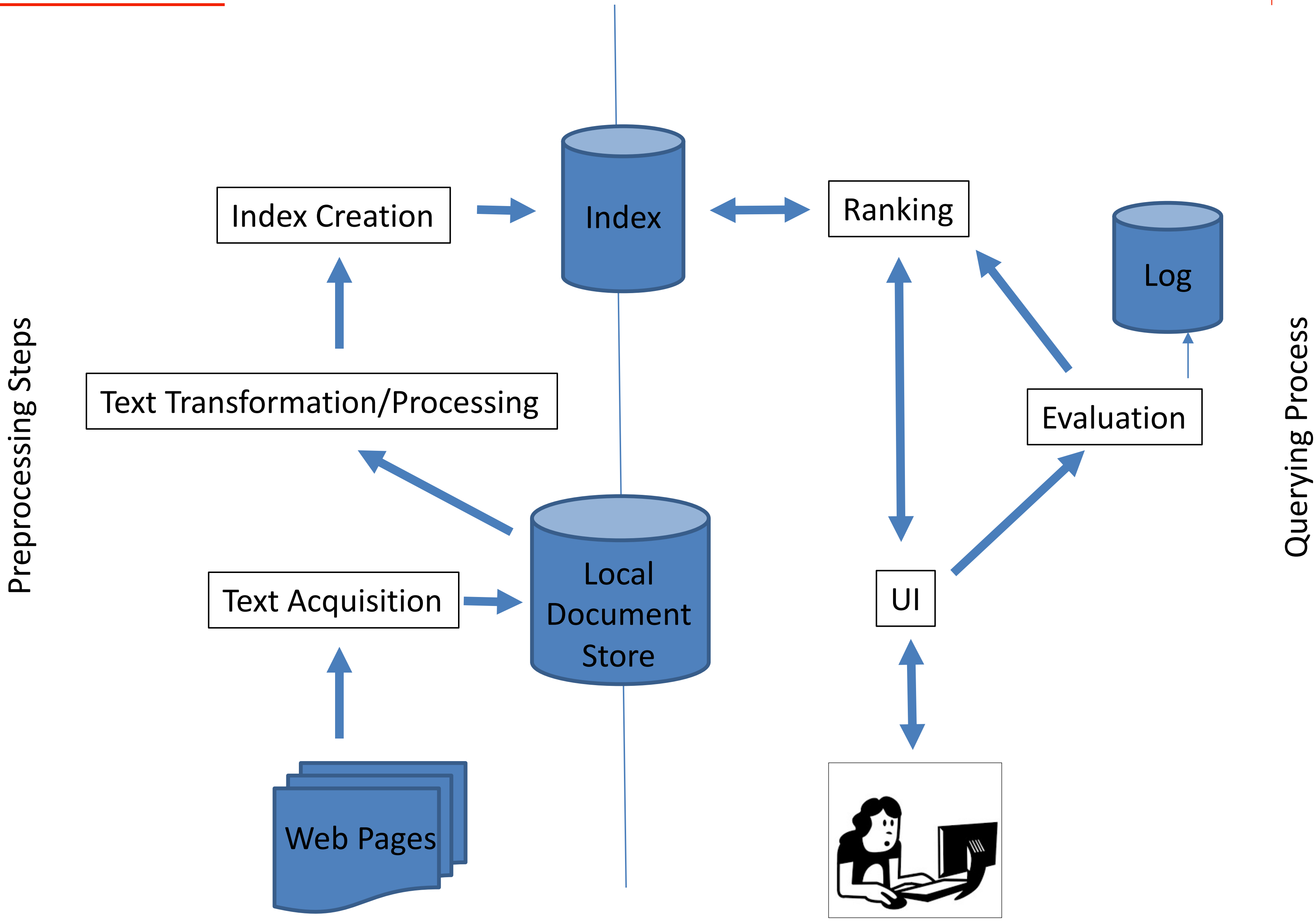
Informatics 225
Computer Science 221

Information Retrieval

Lecture 24

These course materials borrow, with permission, from those of Prof. Cristina Videira Lopes, Addison Wesley 2008, Chris Manning, Pandu Nayak, Hinrich Schütze, Heike Adel, Sascha Rothe, Jerome H. Friedman, Robert Tibshirani, and Trevor Hastie. Powerpoint theme by Prof. André van der Hoek.

Architecture



Link Analysis

Information Retrieval

These course materials borrow, with permission, from those of Prof. Cristina Videira Lopes, Addison Wesley 2008, Chris Manning, Pandu Nayak, Hinrich Schütze, Heike Adel, Sascha Rothe, Jerome H. Friedman, Robert Tibshirani, and Trevor Hastie

HUBS AND AUTHORITIES

Hyperlink-Induced Topic Search (HITS)

- In response to a query, instead of an ordered list of pages each meeting the query, find two sets of inter-related pages:
  - *Hub pages* are good lists of links on a subject.
    - e.g., “Bob’s list of cancer-related links.”
  - *Authority pages* occur recurrently on good hubs for the subject.

Hyperlink-Induced Topic Search (HITS)

- In response to a query, instead of an ordered list of pages each meeting the query, find two sets of inter-related pages:
  - *Hub pages* are good lists of links on a subject.
    - e.g., “Bob’s list of cancer-related links.”
  - *Authority pages* occur recurrently on good hubs for the subject.

Hyperlink-Induced Topic Search (HITS)

- In response to a query, instead of an ordered list of pages each meeting the query, find two sets of inter-related pages:
  - *Hub pages* are good lists of links on a subject.
    - e.g., “Bob’s list of cancer-related links.”
  - *Authority pages* occur recurrently on good hubs for the subject.

- Best suited for “broad topic” queries rather than for page-finding queries.

- Gets at a broader slice of common *opinion*.

Hubs and Authorities

- A good hub page for a topic *points* to many authoritative pages for that topic.

Hubs and Authorities

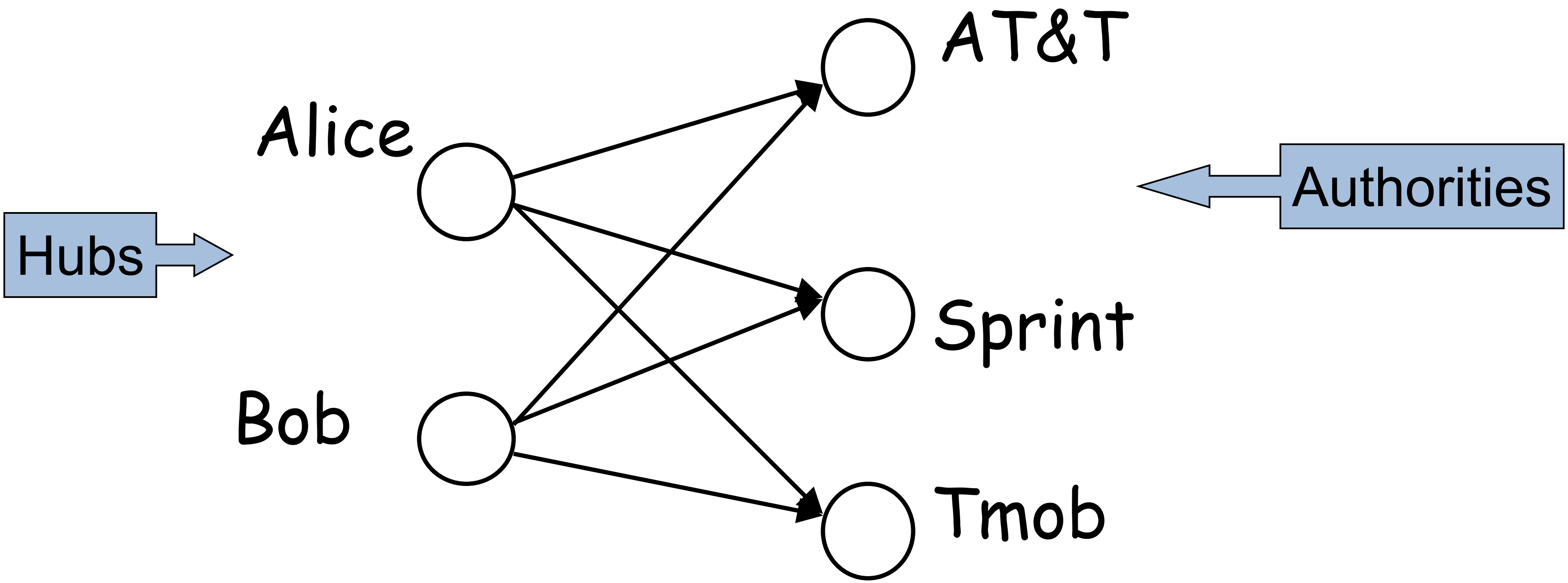
- A good hub page for a topic *points* to many authoritative pages for that topic.

- A good authority page for a topic is *pointed* to by many good hubs for that topic.

Hubs and Authorities

- A good hub page for a topic *points* to many authoritative pages for that topic.
- A good authority page for a topic is *pointed* to by many good hubs for that topic.
- Circular definition - will turn this into an iterative computation.

The hope



Mobile telecom companies

High-level scheme

- Extract from the web a base set of pages that *could* be good hubs or authorities.

- From these, identify a small set of top hub and authority pages;

  →iterative algorithm.

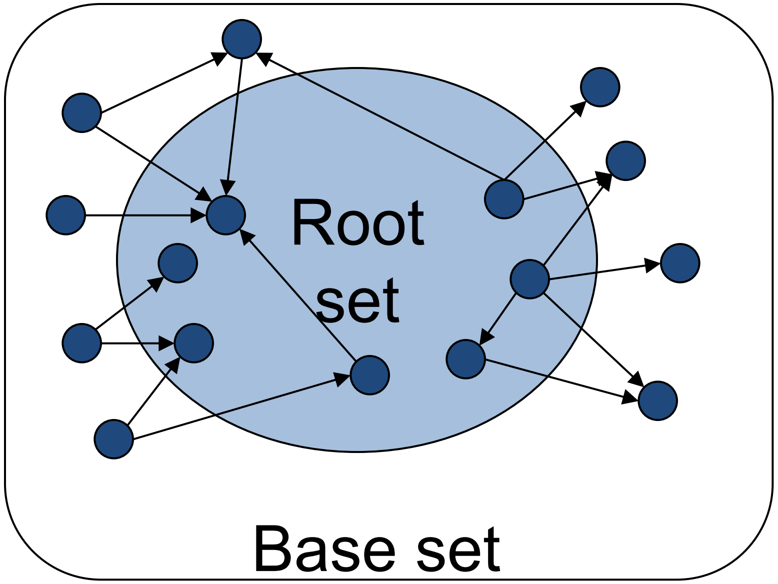
Creating the Base set

- Given text query (say ***browser***), use a text index to get all pages containing ***browser.***
  - Call this the root set of pages.

Creating the Base set

- Given text query (say ***browser***), use a text index to get all pages containing ***browser.***
  - Call this the root set of pages.
- Add in any page that either
  - points to a page in the root set, or
  - is pointed to by a page in the root set.
- Call this the base set.

Visualization



Get in-links (and out-links) from a ***connectivity server***

Connectivity Server

- Support for fast queries on the web graph
  - Which URLs point to a given URL?
  - Which URLs does a given URL point to?

Connectivity Server

- Support for fast queries on the web graph
  - Which URLs point to a given URL?
  - Which URLs does a given URL point to?

- Stores mappings in memory from
  - URL to outlinks, URL to inlinks
  - Use sparse properties of the graph (do not build the entire matrix!)

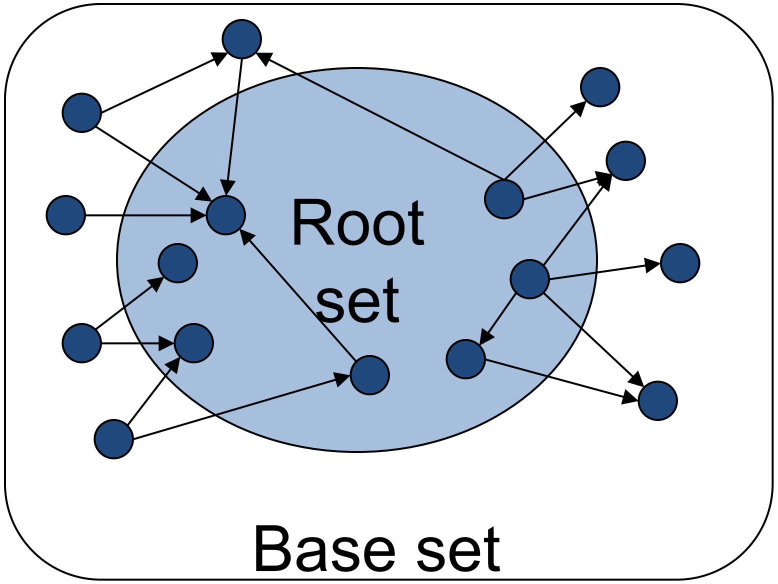
Connectivity Server

- Support for fast queries on the web graph
  - Which URLs point to a given URL?
  - Which URLs does a given URL point to?

- Stores mappings in memory from
  - URL to outlinks, URL to inlinks
  - Use sparse properties of the graph (do not build the entire matrix!)

- Applications
  - Crawl control
  - Web graph analysis
    - *Connectivity, crawl optimization*
  - Link analysis (HITS, PageRank, …)

Visualization



Get in-links (and out-links) from a ***connectivity server***

Distilling hubs and authorities

- Compute, for each page *x* in the base set:
  - a **hub score** ***h(x)***
  - and an **authority score** ***a(x).***

Distilling hubs and authorities

- Compute, for each page *x* in the base set:
  - a **hub score** ***h(x)***
  - and an **authority score** ***a(x).***

- Initialize: for all x, $h(x) \leftarrow 1$; $a(x) \leftarrow 1$;

Distilling hubs and authorities

- Compute, for each page *x* in the base set:
  - a **hub score** ***h(x)***
  - and an **authority score** ***a(x).***

- Initialize: for all x, $h(x) \leftarrow 1$; $a(x) \leftarrow 1$;

- Iteratively update all $h(x)$, $a(x)$;

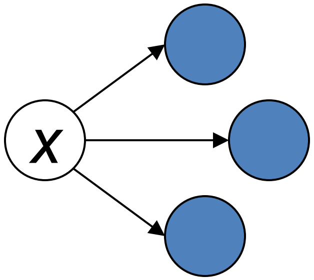
Distilling hubs and authorities

- Compute, for each page *x* in the base set:
  - a **hub score *h(x)***
  - and an **authority score *a(x)*.**
- Initialize: for all *x*, $h(x) \leftarrow 1$; $a(x) \leftarrow 1$;
- Iteratively update all *h(x), a(x)*;
- After a few iterations (or convergence)
  - output pages with highest *h()* scores as top hubs
  - highest *a()* scores as top authorities.

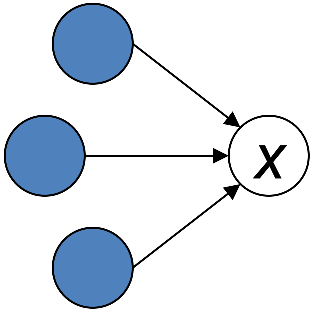
Iterative update

- Repeat the following updates of the **hub score h(x)** and the **authority score a(x)**, for all *x*:

$$h(x) \leftarrow \sum_{x \mapsto y} a(y)$$



$$a(x) \leftarrow \sum_{y \mapsto x} h(y)$$



Scaling

- To prevent the *h()* and *a()* values from getting too big, can scale down after each iteration.

Scaling

- To prevent the *h()* and *a()* values from getting too big, can scale down after each iteration.

- *Scaling factor doesn't really matter (as long as you are consistent):*
  - *we only care about the relative values of the scores.*

How many iterations?

- Claim: relative values of scores will converge after a few iterations:
  - in fact, suitably scaled, *h()* and *a()* scores settle into a steady state!
  - *You can demonstrate this…*
- In practice, ~5 iterations is enough to get you close to stability.

Japan Elementary Schools

Hubs

- schools
- LINK Page-13
- “ú–{,ÌŠw□Z
- □a‰„□¬Šw□Zƒz□[ƒ□ƒy□[ƒW
- 100 Schools Home Pages (English)
- K-12 from Japan 10/...rnet and Education)
- http://www...iglobe.ne.jp/~IKESAN
- ,l,f,j□¬Šw□Z,U”N,P‘g•¨Œê
- □ÒŠ—’¬—§□ÒŠ—“Œ□¬Šw□Z
- Koulutus ja oppilaitokset
- TOYODA HOMEPAGE
- Education
- Cay's Homepage(Japanese)
- –y“ì□¬Šw□Z,Ìƒz□[ƒ□ƒy□[ƒW
- UNIVERSITY
- ‰J—³□¬Šw□Z DRAGON97-TOP
- □Â‰ª□¬Šw□Z,T”N,P‘gƒz□[ƒ□ƒy□[ƒW
- ¶µ°é¼ÂÁ© ¥á¥Ë¥å¡¼ ¥á¥Ë¥å¡¼

Authorities

- The American School in Japan
- The Link Page
- ‰ª□è□s—§ˆä“c□¬Šw□Zƒz□[ƒ□ƒy□[ƒW
- Kids' Space
- ˆÀ□é□s—§ˆÀ□é□¼•”□¬Šw□Z
- ‹{□é‹³ˆç‘åŠw•□‘®□¬Šw□Z
- KEIMEI GAKUEN Home Page (Japanese)
- Shiranuma Home Page
- fuzoku-es.fukui-u.ac.jp
- welcome to Miasa E&J school
- □_“Þ□ìŒ§□E‰¡•l□s—§’†□ì□¼□¬Šw□Z,Ìƒy
- http://www...p/~m_maru/index.html
- fukui haruyama-es HomePage
- Torisu primary school
- goo
- Yakumo Elementary,Hokkaido,Japan
- FUZOKU Home Page
- Kamishibun Elementary School...

Things to note

- Pulled together good pages **regardless of language of page content.**

Things to note

- Pulled together good pages **regardless of language of page content.**

- Use *only* link analysis <u>after</u> base set assembled
  - **iterative scoring is query-independent.**

Things to note

- Pulled together good pages **regardless of language of page content.**

- Use *only* link analysis <u>after</u> base set assembled
  - **iterative scoring is query-independent.**

- Iterative computation <u>after</u> text index retrieval - significant overhead (but not at query-time!)

Hyper-link Induced Topic Search Issues

- Topic Drift
  - Off-topic pages can cause off-topic “authorities” to be returned
    - E.g., the neighborhood graph can be about a “super topic”

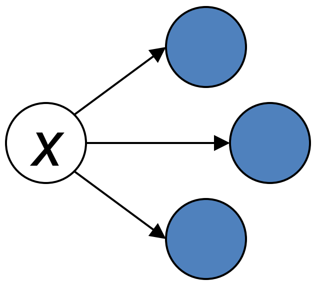
Hyper-link Induced Topic Search Issues

- Topic Drift
  - Off-topic pages can cause off-topic “authorities” to be returned
    - E.g., the neighborhood graph can be about a “super topic”
- Mutually Reinforcing Affiliates
  - Affiliated pages/sites can boost each others’ scores
    - *Linkage between affiliated pages is not a useful signal, so avoid considering them in the analysis*

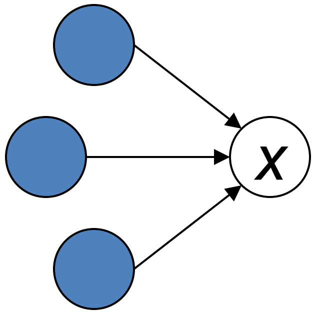
Overview!

- Compute, for each page *x* in the base set:
  - a **hub score** ***h(x)***
  - and an **authority score** ***a(x).***
- Initialize: for all *x*,
  $h(x) \leftarrow 1;\ a(x) \leftarrow 1;$
- Iteratively update all
  $h(x),\ a(x);$
- After a few iterations (or convergence)
  - output pages with highest *h()* scores as top hubs
  - highest *a()* scores as top authorities.

$$h(x) \leftarrow \sum_{x \mapsto y} a(y)$$



$$a(x) \leftarrow \sum_{y \mapsto x} h(y)$$



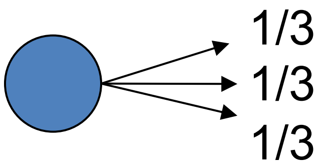
To prevent the *h()* and *a()* values from getting too big, can scale down after each iteration.

Scaling factor doesn’t really matter : only relative values are important

PAGE RANK

Pagerank scoring

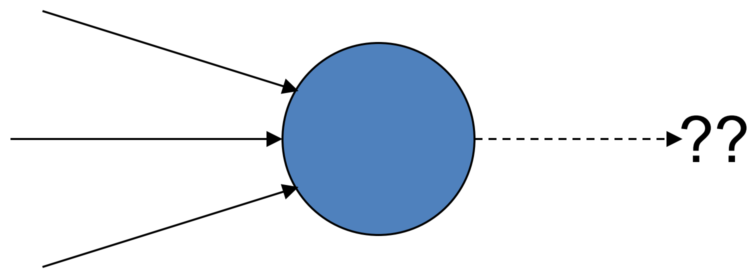
- Imagine a browser doing a random walk on web pages:
  - Start at a random page
  - At each step, go out of the current page along one of the links on that page, equiprobably



- "In the steady state" each page has a long-term visit rate - use this as the page's score.

Not quite enough

- The web is full of dead-ends.
  - Random walk can get stuck in dead-ends.
  - Makes no sense to talk about long-term visit rates.



Teleporting

- At a dead end, jump to a random web page.

- At any non-dead end, with probability 10%, jump to a random web page.
  - With remaining probability (90%), go out on a random link.
  - 10% - a parameter.

Result of teleporting

- Now cannot get stuck locally.

Result of teleporting

- Now cannot get stuck locally.

- There is a long-term rate at which any page is visited (not obvious).

Result of teleporting

- Now cannot get stuck locally.

- There is a long-term rate at which any page is visited (not obvious).
  - **The idea of page rank is that pages that are visited more by this random walker are more important**

PageRank

- Status of searching the web in November 1997: “only one of the top four commercial search engines finds itself (returns its own search page in response to its name in the top ten results).” Brin&Page, 1997

PageRank

- Status of searching the web in November 1997: "only one of the top four commercial search engines finds itself (returns its own search page in response to its name in the top ten results)." Brin&Page, 1997
  - "Our main goal is to improve the quality of web search engines." Brin&Page, 1997

PageRank

- Status of searching the web in November 1997: "only one of the top four commercial search engines finds itself (returns its own search page in response to its name in the top ten results)." Brin&Page, 1997
  - "Our main goal is to improve the quality of web search engines." Brin&Page, 1997
- *See the original Google and PageRank papers in Canvas*

Result of teleporting

- Now cannot get stuck locally.
- There is a long-term rate at which any page is visited (not obvious).
  - **The idea of page rank is that pages that are visited more by this random walker are more important**
- **How do we compute this visit rate?**

Computing PageRank

1. **Algebraic calculation (given in Google paper)**

2. Markov chain model

PageRank -- Algebraic

$$p_i = \boxed{\text{matemagic}}\; p_j$$

PageRank -- Algebraic

$$p_i = (1-d) + d\sum_{j=1}^{N}\left(\frac{L_{ij}}{c_j}\right)p_j$$

PageRank -- Algebraic

PageRank of page i $$p_i = (1-d) + d\sum_{j=1}^{N}\left(\frac{L_{ij}}{c_j}\right)p_j$$ PageRank of page j

PageRank -- Algebraic

Liij = 1 if page j points to i, and zero otherwise

PageRank of page i

$$p_i = (1-d) + d\sum_{j=1}^{N}\left(\frac{L_{ij}}{c_j}\right)p_j$$

PageRank of page j

$$\boxed{c_j = \sum_{i=1}^{N} L_{ij}}$$

How many pages are leaving page j

PageRank -- Algebraic

Liij = 1 if page j points to i, and zero otherwise

damping factor d

PageRank of page i

$$p_i = (1 - d) + d \sum_{j=1}^{N} \left(\frac{L_{ij}}{c_j}\right) p_j$$

PageRank of page j

$$\boxed{c_j = \sum_{i=1}^{N} L_{ij}}$$

How many pages are leaving page j

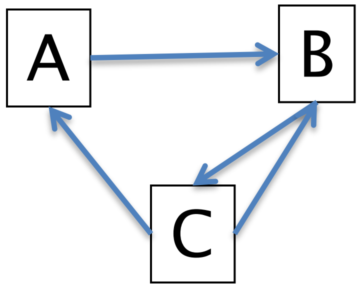
PageRank -- Algebraic

$$PR(A) = (1\text{-}d) + d \sum (PR(Ti)/C(Ti))$$

$$PR(A) = (1\text{-}d) + d\ (PR(T1)/C(T1) + \ldots + PR(Tn)/C(Tn))$$

Where
A is a page
d is a damping/teleport factor (usually 0.85)
T1...Tn are pages that link to A
PR(Ti) is the PageRank of Ti
C(Ti) is the number of outgoing links from Ti

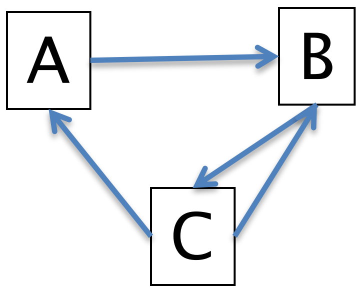
Example



PR(A) = 0.15 + 0.85 * PR(C)/2
PR(B) = 0.15 + 0.85 * (PR(A)/1 + PR(C)/2)
PR(C) = 0.15 + 0.85 * (PR(B)/1)

How do we start?!?
Guess: PR(p) = 1 for starters
Then, iterate

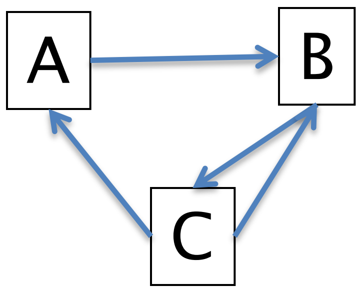
Example – Iterations



PR(A) = 0.15 + 0.85 * PR(C)/2
PR(B) = 0.15 + 0.85 * (PR(A)/1 + PR(C)/2)
PR(C) = 0.15 + 0.85 * (PR(B)/1)

Iter 1 $\left\{ \begin{array}{l} PR(A) = 0.15+0.85 * 1/2 = 0.575 \\ PR(B) = 0.15+0.85 * (1+1/2) = 1.425 \\ PR(C) = 0.15+0.85 * 1 = 1 \end{array} \right.$

Example – Iterations



PR(A) = 0.15 + 0.85 * PR(C)/2
PR(B) = 0.15 + 0.85 * (PR(A)/1 + PR(C)/2)
PR(C) = 0.15 + 0.85 * (PR(B)/1)

Iter 1
PR(A) = 0.15+0.85 * 1/2 = 0.575
PR(B) = 0.15+0.85 * (1+1/2) = 1.425
PR(C) = 0.15+0.85 * 1 = 1

Iter 2
PR(A) = 0.15+0.85 * 1/2 = 0.575
PR(B) = 0.15+0.85 * (0.575+1/2) = 1.06375
PR(C) = 0.15+0.85 * 1.425 = 1.36125

...

Example -- Iterations

- Algorithm converges after N iterations
- Once converged, normalized probability distribution (average PageRank for all pages) will be 1
- You can gain some visual intuition : http://ed.ilogues.com/projects/2015/03/22/pagerank-visualization

Pagerank summary

- Preprocessing:
  - Given graph of links, build a (sparse) matrix of the links.
  - From it compute **page rank**
- One possibility to add PR in query processing (at query time):
  - Retrieve pages meeting query.
  - Rank them by their pagerank.
  - But this rank order is query-*independent* …

The reality

- Pagerank is used in google and other engines, but is hardly the full story of ranking
  - **It is just one among many criteria used in relevance scoring**
  - Many additional sophisticated features are used
  - Some address specific query classes
  - Machine learned ranking is used each time more

The reality

- Pagerank is used in google and other engines, but is hardly the full story of ranking
  - **It is just one among many criteria used in relevance scoring**
  - Many additional sophisticated features are used
  - Some address specific query classes
  - Machine learned ranking is used each time more

- **Given a query, a real-life search engine will compute a composite score for each web page that combines hundreds of features** (cosine similarity, term proximity, PageRank, HITS, etc.)

- Pagerank *alone* is still very useful for things like crawl policy

PageRank – Algebraic in matrix notation

$$p_i = (1-d) + d\sum_{j=1}^{N}(\frac{L_{ij}}{c_j})p_j$$

PageRank – Algebraic in matrix notation

$$p_i = (1-d) + d\sum_{j=1}^{N}\left(\frac{L_{ij}}{c_j}\right)p_j$$

$$\mathbf{p} = (1-d)\mathbf{e} + d \cdot \mathbf{L}\mathbf{D}_c^{-1}\mathbf{p}$$

Vector of N ones

Diagonal matrix with diagonal elements c_j

PageRank – Algebraic in matrix notation

$$p_i = (1-d) + d \sum_{j=1}^{N} \left(\frac{L_{ij}}{c_j}\right) p_j$$

$$\mathbf{p} = (1-d)\mathbf{e} + d \cdot \mathbf{L}\mathbf{D}_c^{-1}\mathbf{p} \qquad \boxed{\mathbf{e}^T\mathbf{p} = N}$$

If we adopt a normalization factor
such that the average pagerank is 1

PageRank – Algebraic in matrix notation

$$p_i = (1-d) + d \sum_{j=1}^{N} \left(\frac{L_{ij}}{c_j}\right) p_j$$

$$\mathbf{p} = (1-d)\mathbf{e} + d \cdot \mathbf{L}\mathbf{D}_c^{-1}\mathbf{p} \qquad \mathbf{e}^T\mathbf{p} = N$$

$$\begin{aligned}\mathbf{p} &= \left[(1-d)\mathbf{e}\mathbf{e}^T/N + d\mathbf{L}\mathbf{D}_c^{-1}\right]\mathbf{p} \\ &= \mathbf{A}\mathbf{p}\end{aligned}$$

PageRank – Algebraic in matrix notation

$$p_i = (1-d) + d \sum_{j=1}^{N} \left(\frac{L_{ij}}{c_j}\right) p_j$$

$$\mathbf{p} = (1-d)\mathbf{e} + d \cdot \mathbf{L}\mathbf{D}_c^{-1}\mathbf{p} \qquad \mathbf{e}^T\mathbf{p} = N$$

$$\begin{aligned} \mathbf{p} &= \left[(1-d)\mathbf{e}\mathbf{e}^T/N + d\mathbf{L}\mathbf{D}_c^{-1}\right]\mathbf{p} \\ &= \mathbf{A}\mathbf{p} \end{aligned}$$

$$\mathbf{p}_k \leftarrow \mathbf{A}\mathbf{p}_{k-1}; \quad \mathbf{p}_k \leftarrow N\frac{\mathbf{p}_k}{\mathbf{e}^T\mathbf{p}_k}.$$