Eigenvector Methods

in

Information Retrieval

Amy Langville

Carl Meyer

SAMSI Institute
Department of Mathematics
North Carolina State University

SIAM AN–New Orleans 7/12/2005

Outline

Part 1: Traditional IR

- Vector Space Model (1960s and 1970s)
- Latent Semantic Indexing (1990s)

Part 2: Web IR

- PageRank (1998)
- HITS (1998)

Vector Space Model (1960s and 1970s)

Gerard Salton's Information Retrieval System

SMART: System for the Mechanical Analysis and Retrieval of Text

(Salton's Magical Automatic Retriever of Text)

- turn n textual documents into n document vectors $\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n$
- create term-by-document matrix $\mathbf{A}_{m \times n} = [\, \mathbf{d}_1 | \mathbf{d}_2 | \cdots | \mathbf{d}_n \,]$
- to retrieve info., create query vector $\mathbf{q}$, which is a pseudo-doc

Vector Space Model (1960s and 1970s)

Gerard Salton's Information Retrieval System

SMART: System for the Mechanical Analysis and Retrieval of Text

(Salton's Magical Automatic Retriever of Text)

- turn n textual documents into n document vectors $\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n$
- create term-by-document matrix $\mathbf{A}_{m \times n} = [\, \mathbf{d}_1 | \mathbf{d}_2 | \cdots | \mathbf{d}_n \,]$
- to retrieve info., create query vector $\mathbf{q}$, which is a pseudo-doc

GOAL: find doc. $\mathbf{d}_i$ closest to $\mathbf{q}$

— angular cosine measure used: $\delta_i = cos\,\theta_i = \mathbf{q}^T\mathbf{d}_i / (\|\mathbf{q}\|_2 \|\mathbf{d}_i\|_2)$

Example from Berry's book

Terms	Documents
T1: Bab(y,ies,y's)	D1: Infant & Toddler First Aid
T2: Child(ren's)	D2: Babies & Children's Room (For Your Home)
T3: Guide	D3: Child Safety at Home
T4: Health	D4: Your Baby's Health & Safety : From Infant to Toddler
T5: Home	D5: Baby Proofing Basics
T6: Infant	D6: Your Guide to Easy Rust Proofing
T7: Proofing	D7: Beanie Babies Collector's Guide
T8: Safety	
T9: Toddler	

Example from Berry's book

Terms	Documents
T1: Bab(y,ies,y's)	D1: Infant & Toddler First Aid
T2: Child(ren's)	D2: Babies & Children's Room (For Your Home)
T3: Guide	D3: Child Safety at Home
T4: Health	D4: Your Baby's Health & Safety : From Infant to Toddler
T5: Home	D5: Baby Proofing Basics
T6: Infant	D6: Your Guide to Easy Rust Proofing
T7: Proofing	D7: Beanie Babies Collector's Guide
T8: Safety	
T9: Toddler	

$$\mathbf{A} = \begin{array}{c} \\ t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \\ t_7 \\ t_8 \\ t_9 \end{array} \begin{array}{c} \begin{array}{ccccccc} d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & d_7 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \end{array} \quad \mathbf{q} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \boldsymbol{\delta} = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ \delta_5 \\ \delta_6 \\ \delta_7 \end{bmatrix} = \begin{bmatrix} 0 \\ .5774 \\ 0 \\ .8944 \\ .7071 \\ 0 \\ .7071 \end{bmatrix}$$

Latent Semantic Indexing (1990s)

Susan Dumais's improvement to VSM = LSI

Idea: use low-rank approximation to **A** to filter out noise

- Use truncated SVD as low-rank approximation to **A**

SVD

$\mathbf{A}_{m \times n}$: rank r term-by-document matrix

- SVD: $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$
- LSI: use $\mathbf{A}_k = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$ in place of $\mathbf{A}$
- Why?
  - reduce storage when $k << r$
  - filter out uncertainty, so that performance on text mining tasks (e.g., query processing and clustering) improves

What's Really Happening?

Change of Basis

using truncated SVD $\mathbf{A}_k = \mathbf{U}_k \boldsymbol{\Sigma}_k \mathbf{V}_k^T$

- Original Basis: docs represented in Term Space using Standard Basis $S = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m\}$
- New Basis: docs represented in smaller Latent Semantic Space using Basis $B = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k\}$ $(k << \min(m,n))$

$$\begin{array}{c} \\ nonneg. \\ entries \end{array} \overset{doc_1}{\begin{pmatrix} \vdots \\ \mathbf{A}_{*1} \\ \vdots \end{pmatrix}}_{m \times 1} \approx \begin{bmatrix} \vdots \\ \mathbf{u}_1 \\ \vdots \end{bmatrix} \sigma_1 v_{11} + \begin{bmatrix} \vdots \\ \mathbf{u}_2 \\ \vdots \end{bmatrix} \sigma_2 v_{12} + \cdots + \begin{bmatrix} \vdots \\ \mathbf{u}_k \\ \vdots \end{bmatrix} \sigma_k v_{1k}$$

What's Really Happening?

Change of Basis

using truncated SVD $\mathbf{A}_k = \mathbf{U}_k \boldsymbol{\Sigma}_k \mathbf{V}_k^T$

- Original Basis: docs represented in Term Space using Standard Basis $S = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m\}$
- New Basis: docs represented in smaller Latent Semantic Space using Basis $B = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k\}$ $(k << \min(m,n))$

$$\begin{array}{cc} & doc_1 \\ \begin{array}{c} nonneg. \\ entries \end{array} & \begin{pmatrix} \vdots \\ \mathbf{A}_{*1} \\ \vdots \end{pmatrix}_{m\times 1} \end{array} \approx \begin{bmatrix} \vdots \\ \mathbf{u}_1 \\ \vdots \end{bmatrix} \sigma_1 v_{11} + \begin{bmatrix} \vdots \\ \mathbf{u}_2 \\ \vdots \end{bmatrix} \sigma_2 v_{12} + \cdots + \begin{bmatrix} \vdots \\ \mathbf{u}_k \\ \vdots \end{bmatrix} \sigma_k v_{1k}$$

- still use angular cosine measure

$$\delta_i = cos\,\theta_i = \mathbf{q}^T \mathbf{d}_i / (\|\mathbf{q}\|_2 \|\mathbf{d}_i\|_2) = \mathbf{q}^T \mathbf{A}_k \mathbf{e}_i / (\|\mathbf{q}\|_2 \|\mathbf{A}_k \mathbf{e}_i\|_2)$$

$$= \mathbf{q}^T \mathbf{U}_k \boldsymbol{\Sigma}_k \mathbf{V}_k^T \mathbf{e}_i / (\|\mathbf{q}\|_2 \|\boldsymbol{\Sigma}_k \mathbf{V}_k^T \mathbf{e}_i\|_2)$$

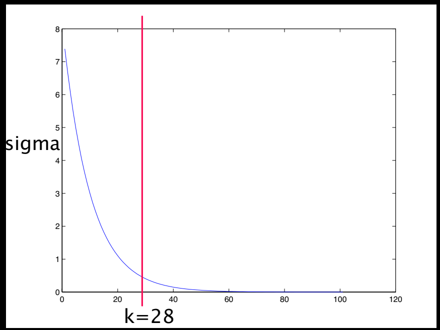
Strengths and Weaknesses of LSI

Strengths

- using $\mathbf{A}_k$ in place of $\mathbf{A}$ gives improved performance
- dimension reduction considers only essential components of term-by-document matrix, filters out noise
- best rank-k approximation: $\|\mathbf{A} - \mathbf{A}_k\|_F = \min_{rank(\mathbf{B}) \leq k} \|\mathbf{A} - \mathbf{B}\|_F$

Weaknesses

- storage—$\mathbf{U}_k$ and $\mathbf{V}_k$ are usually completely dense
- interpretation of basis vectors $\mathbf{u}_i$ is impossible due to mixed signs
- good truncation point k is hard to determine
- orthogonality restriction



Web Information Retrieval

IR before the Web = traditional IR

IR on the Web = web IR

Web Information Retrieval

IR before the Web = traditional IR
IR on the Web = web IR

How is the Web different from other document collections?

Web Information Retrieval

IR before the Web = traditional IR
IR on the Web = web IR

How is the Web different from other document collections?

- It's huge.
  - over 10 billion pages, average page size of 500KB
  - 20 times size of Library of Congress print collection
  - Deep Web - 550 billion pages

Web Information Retrieval

IR before the Web = traditional IR
IR on the Web = web IR

How is the Web different from other document collections?

- It's huge.
  - over 10 billion pages, average page size of 500KB
  - 20 times size of Library of Congress print collection
  - Deep Web - 550 billion pages
- It's dynamic.
  - content changes: 40% of pages change in a week, 23% of .com change daily
  - size changes: billions of pages added each year

Web Information Retrieval

IR before the Web = traditional IR
IR on the Web = web IR

How is the Web different from other document collections?

- It’s huge.
  - over 10 billion pages, average page size of 500KB
  - 20 times size of Library of Congress print collection
  - Deep Web - 550 billion pages
- It’s dynamic.
  - content changes: 40% of pages change in a week, 23% of .com change daily
  - size changes: billions of pages added each year
- It’s self-organized.
  - no standards, review process, formats
  - errors, falsehoods, link rot, and spammers!

Web Information Retrieval

IR before the Web = traditional IR
IR on the Web = web IR

How is the Web different from other document collections?

- It's huge.
  - over 10 billion pages, average page size of 500KB
  - 20 times size of Library of Congress print collection
  - Deep Web - 550 billion pages
- It's dynamic.
  - content changes: 40% of pages change in a week, 23% of .com change daily
  - size changes: billions of pages added each year
- It's self-organized.
  - no standards, review process, formats
  - errors, falsehoods, link rot, and spammers!

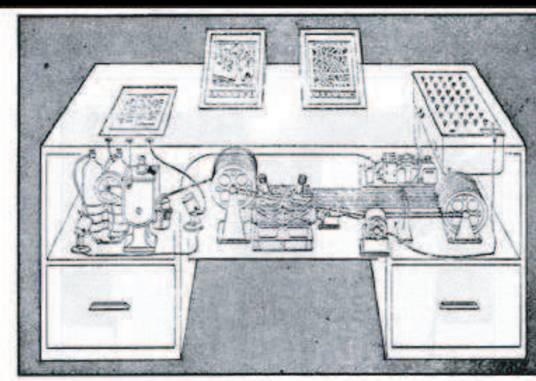
A Herculean Task!

Web Information Retrieval

IR before the Web = traditional IR
IR on the Web = web IR

How is the Web different from other document collections?

- It's huge.
  - over 10 billion pages, each about 500KB
  - 20 times size of Library of Congress print collection
  - Deep Web - 550 billion pages
- It's dynamic.
  - content changes: 40% of pages change in a week, 23% of .com change daily
  - size changes: billions of pages added each year
- It's self-organized.
  - no standards, review process, formats
  - errors, falsehoods, link rot, and spammers!
- Ah, but it's *hyperlinked* !
  - Vannevar Bush's 1945 memex



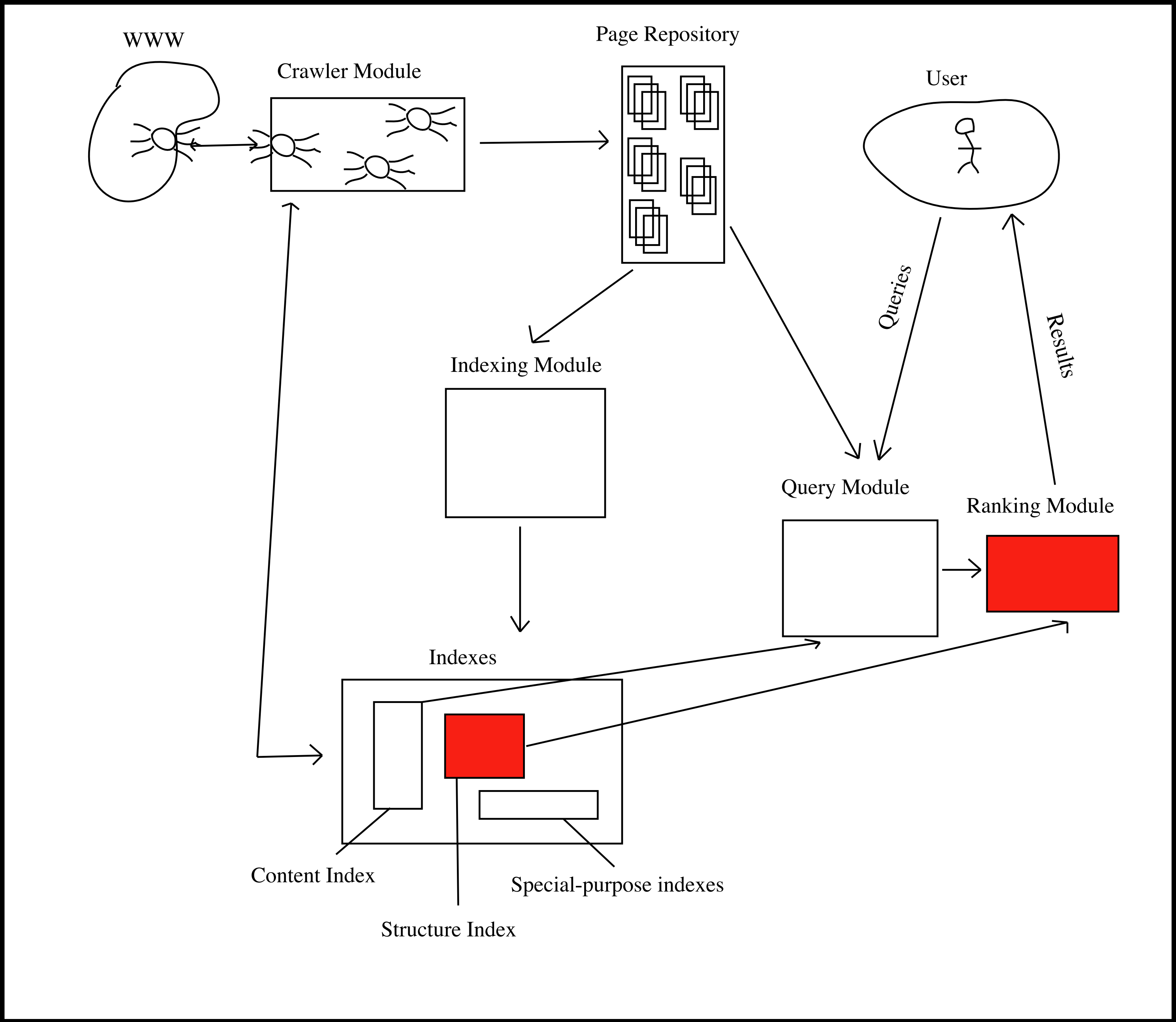
Memex in the form of a desk would instantly bring files and material on any subject to the operator's fingertips. Slanting translucent viewing screens magnify supermicrofilm filed by code numbers. At left is a mechanism which automatically photographs longhand notes, pictures and letters, then files them in the desk for future reference (*LIFE* 19(11), p. 123).

Term-by-Document Matrix for Web

- Too big for factorizations

- $\Rightarrow$ $fast$ inverted file + $\underline{link}$ analysis

Elements of a Web Search Engine



Query Processing

Step 1: User enters query, i.e., aztec baby

Step 2: Inverted file consulted

- term 1 (aardvark) - 3, 117, 3961

.

.

.

- term 10 (aztec) - 3, 15, 19, 101, 673, 1199
- term 11 (baby) - 3, 31, 56, 94, 673, 909, 11114, 253791

.

.

.

- term m (zymurgy) - 1159223

Step 3: Relevant set identified, i.e. $(\mathbf{3}, \mathbf{673})$

Simple traditional engines stop here.

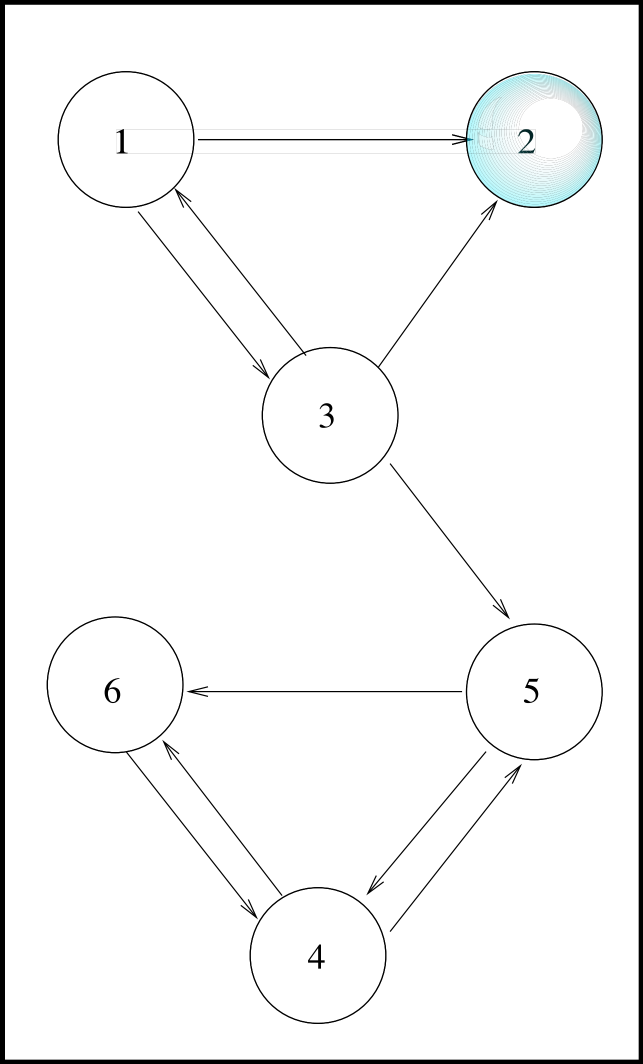
Link Analysis

- uses *hyperlink* structure to focus the relevant set
- combine IR score with popularity or importance score

PageRank - Brin and Page $\Rightarrow$ Google

HITS - Kleinberg $\Rightarrow$ TEOMA

The Web as a Graph



Nodes = webpages

Arcs = hyperlinks

How to Use Web Graph for Search

$\underline{Hyperlink}$ = **Recommendation**

- page with 20 recommmendations (inlinks) must be more important than page with 2 inlinks.
- but status of recommender matters.
  EX: letters of recommendation: 1 letter from Trump vs. 20 from unknown people
- but what if recommender is generous with recommendations?
  EX: suppose Trump has written over 40,000 letters.
- each inlink should be weighted to account for status of recommender and # of outlinks from that recommender

How to Use Web Graph for Search

$\underline{Hyperlink}$ = **Recommendation**

- page with 20 recommmendations (inlinks) must be more important than page with 2 inlinks.
- but status of recommender matters.
  EX: letters of recommendation: 1 letter from Trump vs. 20 from unknown people
- but what if recommender is generous with recommendations?
  EX: suppose Trump has written over 40,000 letters.
- each inlink should be weighted to account for status of recommender and # of outlinks from that recommender

PAGERANK - importance/popularity score given to each page

Ranking by PageRank

The PageRank Idea

(Sergey Brin & Lawrence Page 1998)

- Ranking is preassigned (An off-line calculation)
- Your page P has some rank $r(P)$
- Adjust $r(P)$ higher or lower depending on ranks of pages that point to P
- Importance is not just number, but $quality$ of in-links
  - — role of outlinks relegated
  - — much less sensitive to spamming

PageRank

The Definition

- $r(P) = \sum_{P \in \mathcal{B}_P} \frac{r(P)}{|P|}$
  - $\mathcal{B}_P$ = {all pages pointing to P}
  - $|P|$ = number of out links from P

Successive Refinement

- Start with $r_0(P_i) = 1/n$ for all pages $P_1, P_2, \ldots, P_n$
- Iteratively refine rankings for each page
  - $r_1(P_i) = \sum_{P \in \mathcal{B}_{P_i}} \frac{r_0(P)}{|P|}$
  - $r_2(P_i) = \sum_{P \in \mathcal{B}_{P_i}} \frac{r_1(P)}{|P|}$

$$\ddots$$

  - $r_{j+1}(P_i) = \sum_{P \in \mathcal{B}_{P_i}} \frac{r_j(P)}{|P|}$

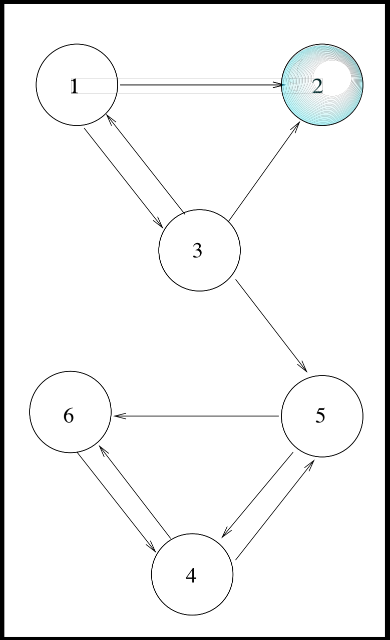
In Matrix Notation

After Step j

$$\boldsymbol{\pi}_j^T = \left[r_j(P_1),\, r_j(P_2),\, \cdots,\, r_j(P_n) \right]$$

$$\boldsymbol{\pi}_{j+1}^T = \boldsymbol{\pi}_j^T \mathbf{H} \quad \text{where} \quad h_{ij} = \begin{cases} 1/|P_i| & \text{if } i \rightarrow j \\ 0 & \text{o.w.} \end{cases}$$

In Matrix Notation



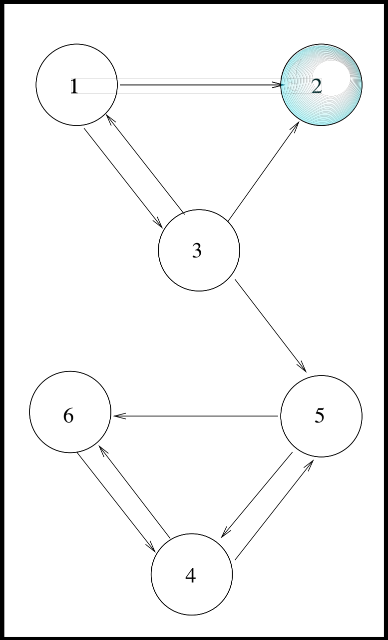
After Step j

$$\boldsymbol{\pi}_j^T = \left[r_j(P_1),\ r_j(P_2),\ \cdots,\ r_j(P_n)\right]$$

$$\boldsymbol{\pi}_{j+1}^T = \boldsymbol{\pi}_j^T \mathbf{H} \quad \text{where} \quad h_{ij} = \begin{cases} 1/|P_i| & \text{if } i \rightarrow j \\ 0 & \text{o.w.} \end{cases}$$

$$\mathbf{H} = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{array} \begin{array}{c} \begin{array}{cccccc} p_1 & p_2 & p_3 & p_4 & p_5 & p_6 \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

In Matrix Notation



After Step j

$$\boldsymbol{\pi}_j^T = \left[r_j(P_1),\, r_j(P_2),\, \cdots,\, r_j(P_n)\right]$$

$$\boldsymbol{\pi}_{j+1}^T = \boldsymbol{\pi}_j^T \mathbf{H} \quad \text{where} \quad h_{ij} = \begin{cases} 1/|P_i| & \text{if } i \rightarrow j \\ 0 & \text{o.w.} \end{cases}$$

$$\mathbf{H} = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{array} \begin{array}{c} \begin{array}{cccccc} p_1 & p_2 & p_3 & p_4 & p_5 & p_6 \end{array} \\ \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

$$\text{PageRank} = \lim_{j \to \infty} \boldsymbol{\pi}_j^T = \boldsymbol{\pi}^T$$

(provided limit exists)

It's Almost a Markov Chain

H has row sums = 1 for ND nodes, row sums = 0 for D nodes

In Matrix Notation

It's Almost a Markov Chain

- **H** has row sums = 1 for ND nodes, row sums = 0 for D nodes

In Matrix Notation

It's Almost a Markov Chain

- **H** has row sums = 1 for ND nodes, row sums = 0 for D nodes

Stochasticity Fix: $\mathbf{S} = \mathbf{H} + \mathbf{a}\mathbf{v}^T$. ($a_i$=1 for $i \in D$, 0, o.w.)

In Matrix Notation

It's Almost a Markov Chain

- **H** has row sums = 1 for ND nodes, row sums = 0 for D nodes

Stochasticity Fix: $\mathbf{S} = \mathbf{H} + \mathbf{a}\mathbf{v}^T$. ($a_i$=1 for $i \in D$, 0, o.w.)

$$\mathbf{S} = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, where\ \mathbf{a} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \mathbf{v}^T = 1/6\ \mathbf{e}^T$$

In Matrix Notation

It's Almost a Markov Chain

- **H** has row sums = 1 for ND nodes, row sums = 0 for D nodes

Stochasticity Fix: $\mathbf{S} = \mathbf{H} + \mathbf{a}\mathbf{v}^T$. ($a_i$=1 for $i \in D$, 0, o.w.)

$$\mathbf{S} = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, where\ \mathbf{a} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \mathbf{v}^T = 1/6\ \mathbf{e}^T$$

- Each π_j^T is a probability distribution vector $\left(\sum_i r_j(P_i) = 1\right)$
- $\pi_{j+1}^T = \pi_j^T \mathbf{S}$ is random walk on the graph defined by links
- $\pi^T = \lim_{j \to \infty} \pi_j^T$ = stationary probability distribution

Random Surfer

Could still encounter Convergence Problems

(dangling nodes, cycles, reducibility)

Irreducibility Fix: $\mathbf{G} = \alpha\mathbf{S} + (1-\alpha)\mathbf{E} \qquad e_{ij} = 1/n \qquad \alpha \approx .85$

$\mathbf{G} = \alpha\mathbf{H} + \alpha\,\mathbf{a}\,\mathbf{v}^T + (1-\alpha)\mathbf{E}$ (trivially irreducible)

- $\boldsymbol{\pi}^T$ is now guaranteed to exist and be unique and power method is guaranteed to converge to $\boldsymbol{\pi}^T$.

Random Surfer

Could still encounter Convergence Problems

Irreducibility Fix: $\mathbf{G} = \alpha\mathbf{S} + (1-\alpha)\mathbf{E} \qquad e_{ij} = 1/n \qquad \alpha \approx .85$

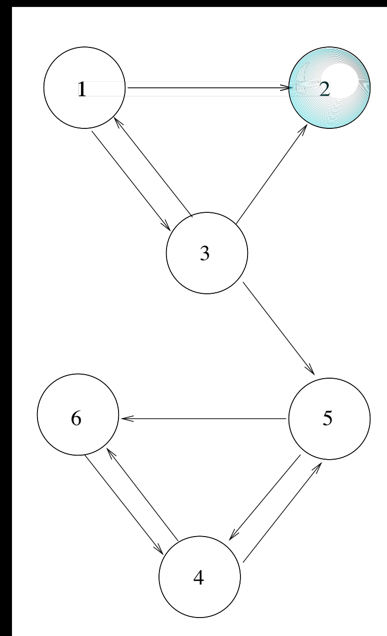
$\mathbf{G} = \alpha\mathbf{H} + \alpha\,\mathbf{a}\,\mathbf{v}^T + (1-\alpha)\mathbf{E}$ (trivially irreducible)

- $\boldsymbol{\pi}^T$ is now guaranteed to exist and be unique and power method is guaranteed to converge to $\boldsymbol{\pi}^T$.

- Different $\mathbf{E} = \mathbf{e}\mathbf{v}^T$ and α allow customization & speedup, yet rank-one update maintained; $\mathbf{G} = \alpha\mathbf{H} + (\alpha\,\mathbf{a} + (1-\alpha)\,\mathbf{e})\mathbf{v}^T$

$$\mathbf{G} = \alpha\mathbf{S} + (1-\alpha)\mathbf{E} = \begin{bmatrix} 1/60 & 7/15 & 7/15 & 1/60 & 1/60 & 1/60 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 19/60 & 19/60 & 1/60 & 1/60 & 19/60 & 1/60 \\ 1/60 & 1/60 & 1/60 & 1/60 & 7/15 & 7/15 \\ 1/60 & 1/60 & 1/60 & 7/15 & 1/60 & 7/15 \\ 1/60 & 1/60 & 1/60 & 11/12 & 1/60 & 1/60 \end{bmatrix}$$

PageRank Example



$$\pi^T = \begin{array}{cccccc} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} & \mathbf{6} \\ \end{array}$$

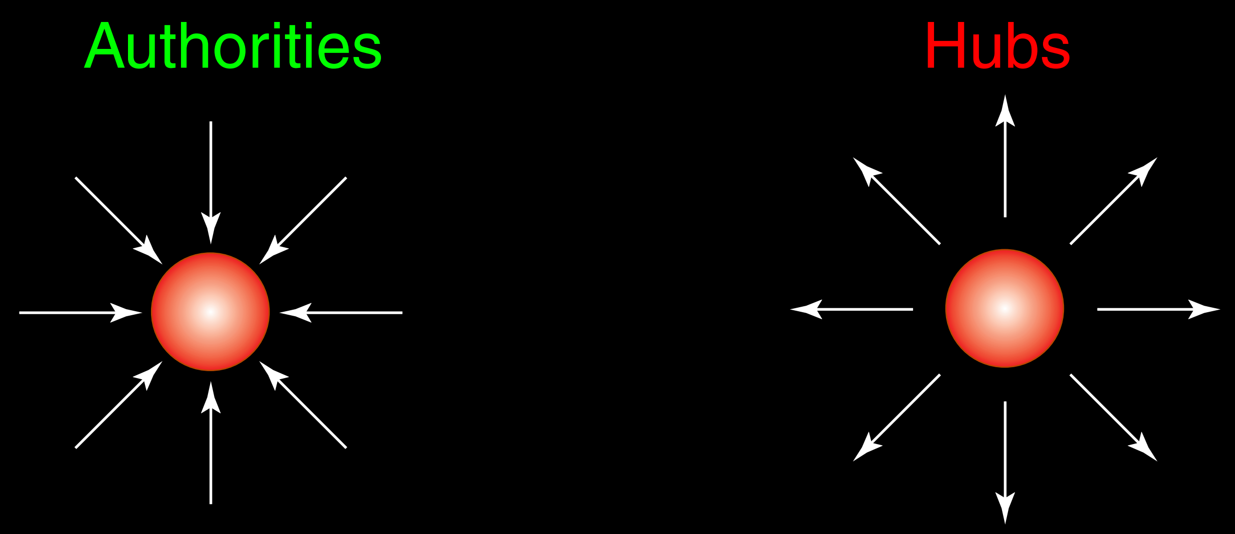
$$\pi^T = \begin{pmatrix} .03721 & .05396 & .04151 & .3751 & .206 & .2862 \end{pmatrix}$$

Global ranking of pages = [4 6 5 2 3 1]

Query-independent way of ranking relevant set

Ranking by HITS

- give each page 2 scores (hub and authority scores) instead of just 1.
- DEFN:



- pages can be both hubs and authorities (EX: ATL airport)
- Good hub pages point to good authority pages, and good authorities are pointed to by good hubs.

HITS - hub and authority score given to each page

HITS - (Hypertext Induced Topic Search) ⇒ Teoma

HITS Algorithm

Hypertext Induced Topic Search (J. Kleinberg 1998)

Determine Authority & Hub Scores

- a_i = authority score for P_i
- h_i = hub score for P_i

Successive Refinement

- Start with $h_i(0) = 1$ for all pages P_i

$$\mathbf{L}_{ij} = \begin{cases} 1 & P_i \rightarrow P_j \\ 0 & P_i \not\rightarrow P_j \end{cases}$$

- Successively refine rankings
  - For $k = 1, 2, \ldots$

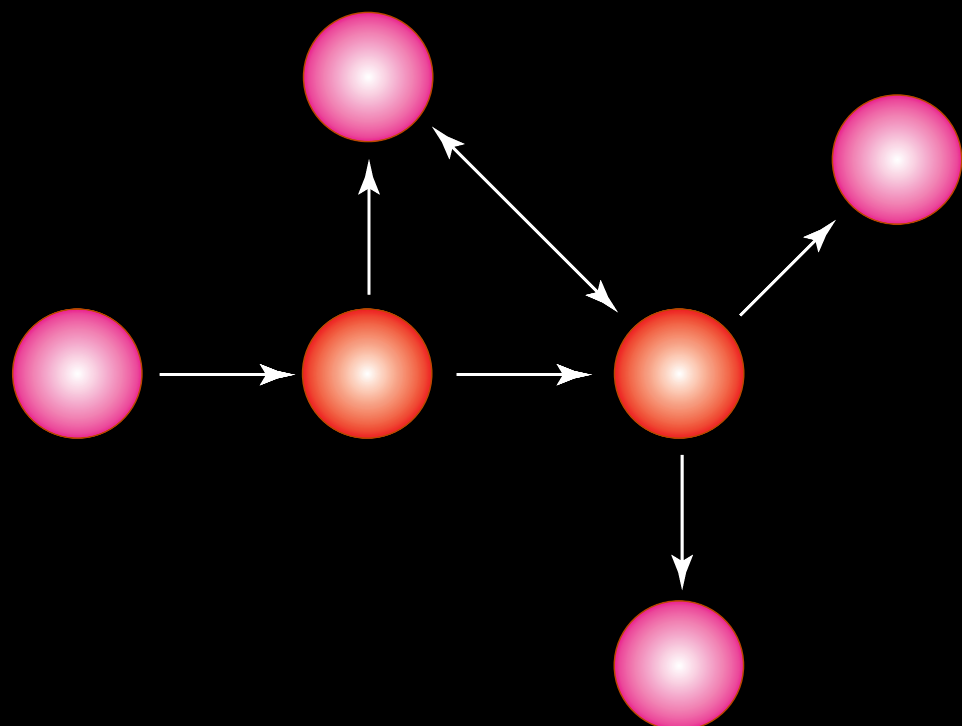
$$a_i(k) = \sum_{j:P_j \rightarrow P_i} h_j(k-1) \quad \Rightarrow \quad \mathbf{a}_k = \mathbf{L}^T \mathbf{h}_{k-1}$$

$$h_i(k) = \sum_{j:P_i \rightarrow P_j} a_j(k) \quad \Rightarrow \quad \mathbf{h}_k = \mathbf{L}\mathbf{a}_k$$

  - $\mathbf{A} = \mathbf{L}^T\mathbf{L}$ $\quad \mathbf{a}_k = \mathbf{A}\mathbf{a}_{k-1} \rightarrow$ e-vector
  - $\mathbf{H} = \mathbf{L}\mathbf{L}^T$ $\quad \mathbf{h}_k = \mathbf{H}\mathbf{h}_{k-1} \rightarrow$ e-vector

HITS Neighborhood Graph

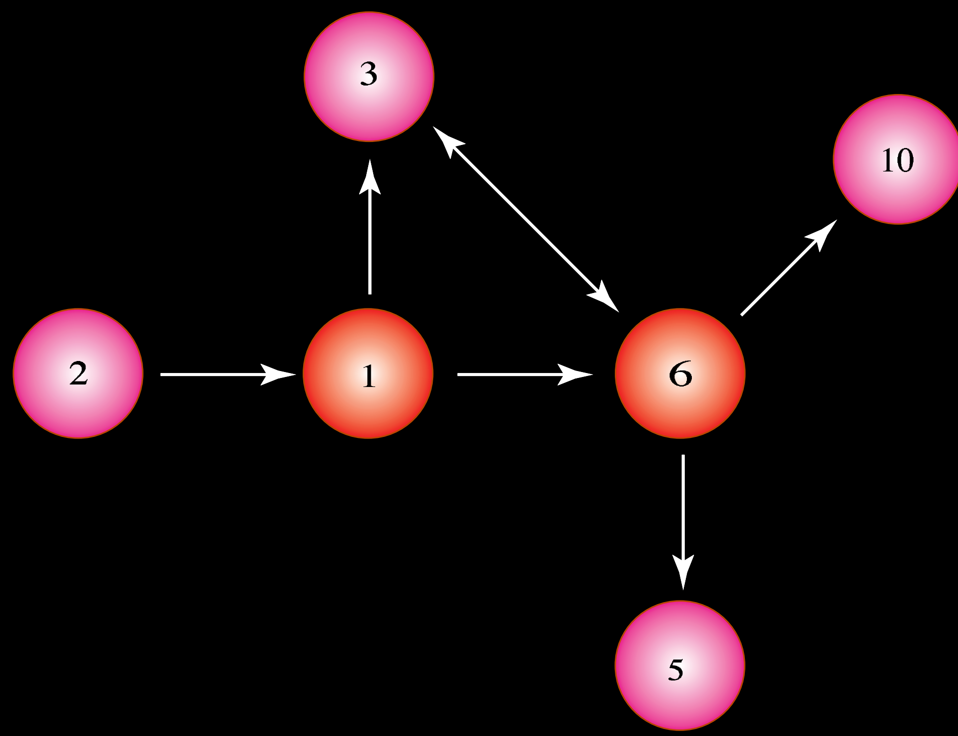
1. Find relevant set by consulting inverted file
2. Build neighborhood graph



3. Compute authority & hub scores for just the neighborhood

HITS Example

1. Relevant set = $[1, 6]$
2. Neighborhood graph N



3. Compute authority & hub scores.

Adjacency matrix for N = **L** =

$$
\begin{array}{c|cccccc}
 & \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{5} & \mathbf{6} & \mathbf{10} \\
\hline
\mathbf{1} & 0 & 0 & 1 & 0 & 1 & 0 \\
\mathbf{2} & 1 & 0 & 0 & 0 & 0 & 0 \\
\mathbf{3} & 0 & 0 & 0 & 0 & 1 & 0 \\
\mathbf{5} & 0 & 0 & 0 & 0 & 0 & 0 \\
\mathbf{6} & 0 & 0 & 1 & 1 & 0 & 0 \\
\mathbf{10} & 0 & 0 & 0 & 0 & 1 & 0
\end{array}
$$

HITS Example (cont.)

Authority matrix $\mathbf{A} = \mathbf{L}^T\mathbf{L}$ Hub matrix $\mathbf{H} = \mathbf{L}\mathbf{L}^T$

$$\mathbf{L}^T\mathbf{L} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 5 \\ 6 \\ 10 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 5 & 6 & 10 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}, \mathbf{L}\mathbf{L}^T = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 5 \\ 6 \\ 10 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 5 & 6 & 10 \end{array} \\ \begin{pmatrix} 2 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 2 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Authority score vector $\mathbf{a}$

$$\mathbf{a}^T = \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 5 & 6 & 10 \end{array} \\ \begin{pmatrix} 0 & 0 & .3660 & .1340 & .5 & 0 \end{pmatrix} \end{array}$$

Hub score vector $\mathbf{h}$

$$\mathbf{h}^T = \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 5 & 6 & 10 \end{array} \\ \begin{pmatrix} .3660 & 0 & .2113 & 0 & .2113 & .2113 \end{pmatrix} \end{array}$$

Conclusions

- These three information retrieval methods rely on eigenvector calculations.
- Large-scale matrices involved.
- Robust, efficient algorithms are essential.