



# Information Retrieval Web Search

Carl Meyer

Department of Mathematics  
North Carolina State University  
Raleigh, NC

SAC Cap Man 7/14/2005






















A man with a beard and mustache, wearing a dark sweater, is sitting in front of a bookshelf. He is looking towards the camera. The bookshelf behind him is filled with books. A white banner at the bottom of the image contains the text "Christian Pilet".

Christian Pilet

# Early Search Engines

**S**ystem for the **M**echanical **A**nalysis and **R**etrieval of **T**ext

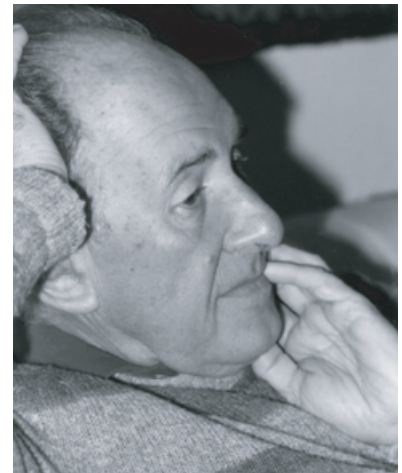
Harvard 1962 – 1965

IBM 7094 & IBM 360

Gerard Salton

Implemented at Cornell (1965 – 1970)

Based on matrix methods





# Term–Document Matrices

**Start with dictionary of terms**

Words or phrases ( e.g., *landing gear*)



# Term–Document Matrices

## Start with dictionary of terms

Words or phrases ( e.g., *landing gear*)

## Index Each Document

Humans scour pages and mark key terms



# Term–Document Matrices

## Start with dictionary of terms

Words or phrases ( e.g., *landing gear*)

## Index Each Document

Humans scour pages and mark key terms

Count  $f_{ij}$  = # times term  $i$  appears in document  $j$



# Term–Document Matrices

## Start with dictionary of terms

Words or phrases ( e.g., *landing gear*)

## Index Each Document

Humans scour pages and mark key terms

Count  $f_{ij}$  = # times term  $i$  appears in document  $j$

## Term–Document Matrix

$$\begin{array}{c} \text{TERM 1} \\ \text{TERM 2} \\ \vdots \\ \text{TERM } m \end{array} \begin{pmatrix} \text{Doc 1} & \text{Doc 2} & \cdots & \text{Doc } n \\ f_{11} & f_{12} & \cdots & f_{1n} \\ f_{21} & f_{22} & \cdots & f_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ f_{m1} & f_{m2} & \cdots & f_{mn} \end{pmatrix} = \mathbf{A}_{m \times n}$$





# Query Matching

## Query Vector

$$\mathbf{q}^T = (q_1, q_2, \dots, q_m)$$

$$q_i = \begin{cases} 1 & \text{if Term } i \text{ is requested} \\ 0 & \text{if not} \end{cases}$$



# Query Matching

## Query Vector

$$\mathbf{q}^T = (q_1, q_2, \dots, q_m) \quad q_i = \begin{cases} 1 & \text{if Term } i \text{ is requested} \\ 0 & \text{if not} \end{cases}$$

## How Close is Query to Each Document?

i.e., how close is  $\mathbf{q}$  to each column  $\mathbf{A}_i$ ?

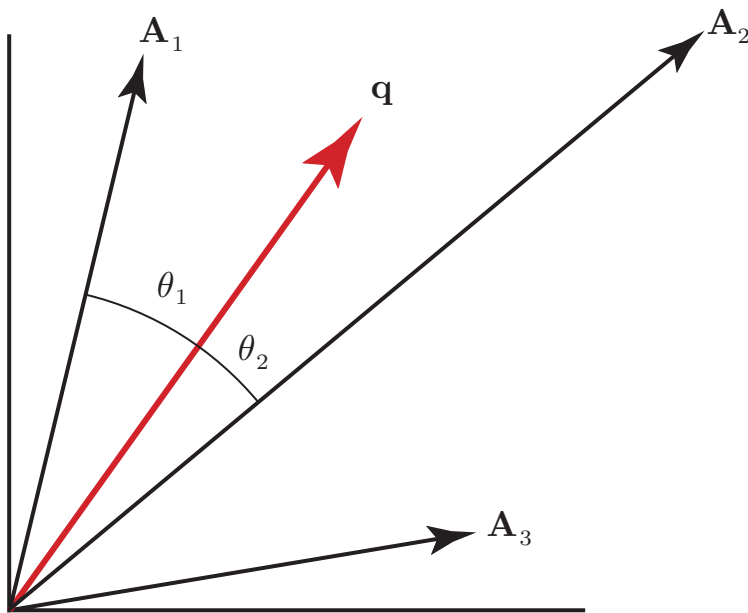
# Query Matching

## Query Vector

$$\mathbf{q}^T = (q_1, q_2, \dots, q_m) \quad q_i = \begin{cases} 1 & \text{if Term } i \text{ is requested} \\ 0 & \text{if not} \end{cases}$$

## How Close is Query to Each Document?

i.e., how close is  $\mathbf{q}$  to each column  $\mathbf{A}_i$ ?



$$\|\mathbf{q} - \mathbf{A}_1\| < \|\mathbf{q} - \mathbf{A}_2\| \text{ but } \theta_2 < \theta_1$$



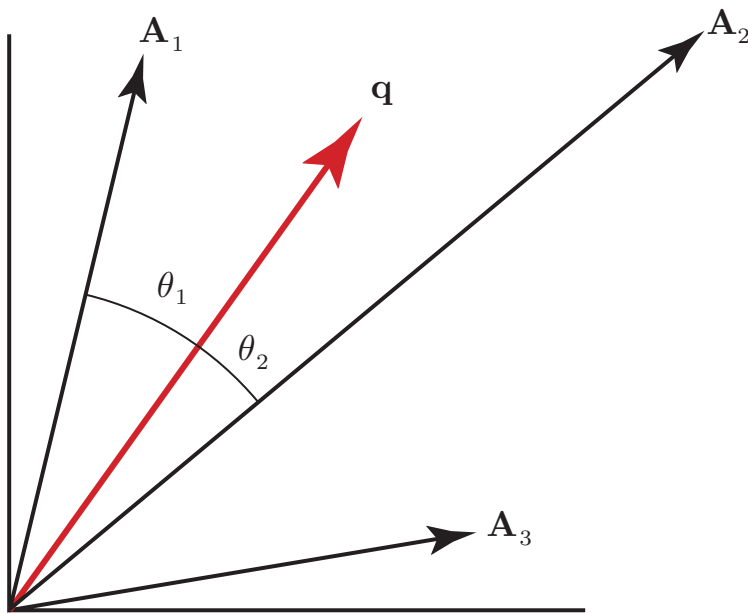
# Query Matching

## Query Vector

$$\mathbf{q}^T = (q_1, q_2, \dots, q_m) \quad q_i = \begin{cases} 1 & \text{if Term } i \text{ is requested} \\ 0 & \text{if not} \end{cases}$$

## How Close is Query to Each Document?

i.e., how close is  $\mathbf{q}$  to each column  $\mathbf{A}_i$ ?



$$\|\mathbf{q} - \mathbf{A}_1\| < \|\mathbf{q} - \mathbf{A}_2\| \text{ but } \theta_2 < \theta_1$$

$$\text{Use } \delta_i = \cos \theta_i = \frac{\mathbf{q}^T \mathbf{A}_i}{\|\mathbf{q}\| \|\mathbf{A}_i\|}$$

Rank documents by size of  $\delta_i$

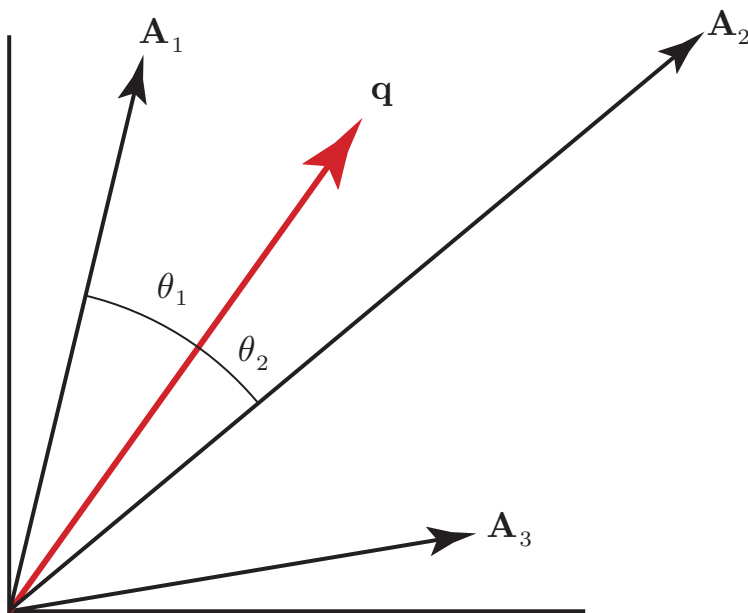
# Query Matching

## Query Vector

$$\mathbf{q}^T = (q_1, q_2, \dots, q_m) \quad q_i = \begin{cases} 1 & \text{if Term } i \text{ is requested} \\ 0 & \text{if not} \end{cases}$$

## How Close is Query to Each Document?

i.e., how close is  $\mathbf{q}$  to each column  $\mathbf{A}_i$ ?



$$\|\mathbf{q} - \mathbf{A}_1\| < \|\mathbf{q} - \mathbf{A}_2\| \text{ but } \theta_2 < \theta_1$$

$$\text{Use } \delta_i = \cos \theta_i = \frac{\mathbf{q}^T \mathbf{A}_i}{\|\mathbf{q}\| \|\mathbf{A}_i\|}$$

Rank documents by size of  $\delta_i$

Return Document  $i$  to user when  $\delta_i \geq tol$



# Term Weighting

## A Problem

Suppose query = *HEDGE FUND*

If *HEDGE FUND* occurs once in  $D_1$  and twice in  $D_2$

▷ Then  $\delta_2 \approx 2\delta_1$  ( if  $\|\mathbf{A}_1\| \approx \|\mathbf{A}_2\|$  )



# Term Weighting

## A Problem

Suppose query = *HEDGE FUND*

If *HEDGE FUND* occurs once in  $D_1$  and twice in  $D_2$

▷ Then  $\delta_2 \approx 2\delta_1$  ( if  $\|\mathbf{A}_1\| \approx \|\mathbf{A}_2\|$  )

## To Compensate

Set  $a_{ij} = \log(1 + f_{ij})$  (Other weights also used)



# Term Weighting

## A Problem

Suppose query = *HEDGE FUND*

If *HEDGE FUND* occurs once in  $D_1$  and twice in  $D_2$

▷ Then  $\delta_2 \approx 2\delta_1$  ( if  $\|\mathbf{A}_1\| \approx \|\mathbf{A}_2\|$  )

## To Compensate

Set  $a_{ij} = \log(1 + f_{ij})$  (Other weights also used)

## Query Weighting Also Performed





# Uncertainties

## Ambiguity in Vocabulary

A *plane* could be ...



# Uncertainties

## Ambiguity in Vocabulary

A *plane* could be ...

- A flat geometrical object



# Uncertainties

## Ambiguity in Vocabulary

A *plane* could be ...

- A flat geometrical object
- A woodworking tool



# Uncertainties

## Ambiguity in Vocabulary

A *plane* could be ...

- A flat geometrical object
- A woodworking tool
- A Boeing product



# Uncertainties

## Ambiguity in Vocabulary

A *plane* could be ...

- A flat geometrical object
- A woodworking tool
- A Boeing product

## Variation in Writing Style

No two authors write the same way



# Uncertainties

## Ambiguity in Vocabulary

A *plane* could be ...

- A flat geometrical object
- A woodworking tool
- A Boeing product

## Variation in Writing Style

No two authors write the same way

- One author may write *car* and *laptop*



# Uncertainties

## Ambiguity in Vocabulary

A *plane* could be ...

- A flat geometrical object
- A woodworking tool
- A Boeing product

## Variation in Writing Style

No two authors write the same way

- One author may write *car* and *laptop*
- Another author may write *automobile* and *portable*



# Uncertainties

## Ambiguity in Vocabulary

A *plane* could be ...

- A flat geometrical object
- A woodworking tool
- A Boeing product

## Variation in Writing Style

No two authors write the same way

- One author may write *car* and *laptop*
- Another author may write *automobile* and *portable*

## Variation in Indexing Conventions

- No two people index documents the same way
- Computer indexing is inexact and can be unpredictable





# Theory vs Practice

**In Theory — it's simple and elegant**



# Theory vs Practice

In Theory — it's simple and elegant

— Index Docs — Weight frequencies in  $\mathbf{A}$  — Normalize  $\|\mathbf{A}_i\| = 1$



# Theory vs Practice

## In Theory — it's simple and elegant

- Index Docs — Weight frequencies in  $\mathbf{A}$  — Normalize  $\|\mathbf{A}_i\| = 1$
- For each query, Weight terms — Normalize  $\|\mathbf{q}\| = 1$



# Theory vs Practice

## In Theory — it's simple and elegant

- Index Docs — Weight frequencies in  $\mathbf{A}$  — Normalize  $\|\mathbf{A}_i\| = 1$
- For each query, Weight terms — Normalize  $\|\mathbf{q}\| = 1$
- Compute  $\delta_i = \cos \theta_i = (\mathbf{q}^T \mathbf{A})_i$  to return the most relevant docs



# Theory vs Practice

## In Theory — it's simple and elegant

- Index Docs — Weight frequencies in  $\mathbf{A}$  — Normalize  $\|\mathbf{A}_i\| = 1$
- For each query, Weight terms — Normalize  $\|\mathbf{q}\| = 1$
- Compute  $\delta_i = \cos \theta_i = (\mathbf{q}^T \mathbf{A})_i$  to return the most relevant docs

## In Practice — it breaks down



# Theory vs Practice

## In Theory — it's simple and elegant

- Index Docs — Weight frequencies in  $\mathbf{A}$  — Normalize  $\|\mathbf{A}_i\| = 1$
- For each query, Weight terms — Normalize  $\|\mathbf{q}\| = 1$
- Compute  $\delta_i = \cos \theta_i = (\mathbf{q}^T \mathbf{A})_i$  to return the most relevant docs

## In Practice — it breaks down

- Suppose query = *car*



# Theory vs Practice

## In Theory — it's simple and elegant

- Index Docs — Weight frequencies in  $\mathbf{A}$  — Normalize  $\|\mathbf{A}_i\| = 1$
- For each query, Weight terms — Normalize  $\|\mathbf{q}\| = 1$
- Compute  $\delta_i = \cos \theta_i = (\mathbf{q}^T \mathbf{A})_i$  to return the most relevant docs

## In Practice — it breaks down

- Suppose query = *car*
- $D_1$  indexed by *gas, car, tire* (found)



# Theory vs Practice

## In Theory — it's simple and elegant

- Index Docs — Weight frequencies in  $\mathbf{A}$  — Normalize  $\|\mathbf{A}_i\| = 1$
- For each query, Weight terms — Normalize  $\|\mathbf{q}\| = 1$
- Compute  $\delta_i = \cos \theta_i = (\mathbf{q}^T \mathbf{A})_i$  to return the most relevant docs

## In Practice — it breaks down

- Suppose query = *car*
- $D_1$  indexed by *gas, car, tire* (found)
- $D_2$  indexed by *automobile, fuel, and tire* (missed)





# Theory vs Practice

## In Theory — it's simple and elegant

- Index Docs — Weight frequencies in  $\mathbf{A}$  — Normalize  $\|\mathbf{A}_i\| = 1$
- For each query, Weight terms — Normalize  $\|\mathbf{q}\| = 1$
- Compute  $\delta_i = \cos \theta_i = (\mathbf{q}^T \mathbf{A})_i$  to return the most relevant docs

## In Practice — it breaks down

- Suppose query = *car*
- $D_1$  indexed by *gas, car, tire* (found)
- $D_2$  indexed by *automobile, fuel, and tire* (missed)

## The Challenge

- Find  $D_2$  by revealing the latent connection through *tire*

# Susan Dumais's Improvement



Approximate **A** with a lower rank matrix

- Great Idea! —> 2 patents for Bell/Telcordia
  - Computer information retrieval using latent semantic structure. U.S. Patent No. 4,839,853, June 13, 1989.
  - Computerized cross-language document retrieval using latent semantic indexing. U.S. Patent No. 5,301,109, April 5, 1994.

(Resource: USPTO <http://patft.uspto.gov/netahtml/srchnum.htm>)



# Latent Semantic Indexing

Use a Fourier expansion of  $\mathbf{A}$

$$\mathbf{A} = \sum_{i=1}^r \sigma_i \mathbf{Z}_i, \quad \langle \mathbf{Z}_i | \mathbf{Z}_j \rangle = \begin{cases} 1 & i=j, \\ 0 & i \neq j, \end{cases} \quad |\sigma_1| \geq |\sigma_2| \geq \cdots \geq |\sigma_r|$$



# Latent Semantic Indexing

Use a Fourier expansion of  $\mathbf{A}$

$$\mathbf{A} = \sum_{i=1}^r \sigma_i \mathbf{Z}_i, \quad \langle \mathbf{Z}_i | \mathbf{Z}_j \rangle = \begin{cases} 1 & i=j, \\ 0 & i \neq j, \end{cases} \quad |\sigma_1| \geq |\sigma_2| \geq \cdots \geq |\sigma_r|$$
$$|\sigma_i| = | \langle \mathbf{Z}_i | \mathbf{A} \rangle | = \text{amount of } \mathbf{A} \text{ in direction of } \mathbf{Z}_i$$



# Latent Semantic Indexing

Use a Fourier expansion of  $\mathbf{A}$

$$\mathbf{A} = \sum_{i=1}^r \sigma_i \mathbf{Z}_i, \quad \langle \mathbf{Z}_i | \mathbf{Z}_j \rangle = \begin{cases} 1 & i=j, \\ 0 & i \neq j, \end{cases} \quad |\sigma_1| \geq |\sigma_2| \geq \cdots \geq |\sigma_r|$$
$$|\sigma_i| = | \langle \mathbf{Z}_i | \mathbf{A} \rangle | = \text{amount of } \mathbf{A} \text{ in direction of } \mathbf{Z}_i$$

Realign data along dominant directions  $\{\mathbf{Z}_1, \dots, \mathbf{Z}_k, \mathbf{Z}_{k+1}, \dots, \mathbf{Z}_r\}$

— Project  $\mathbf{A}$  onto  $\text{span} \{\mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_k\}$



# Latent Semantic Indexing

Use a Fourier expansion of  $\mathbf{A}$

$$\mathbf{A} = \sum_{i=1}^r \sigma_i \mathbf{Z}_i, \quad \langle \mathbf{Z}_i | \mathbf{Z}_j \rangle = \begin{cases} 1 & i=j, \\ 0 & i \neq j, \end{cases} \quad |\sigma_1| \geq |\sigma_2| \geq \cdots \geq |\sigma_r|$$
$$|\sigma_i| = |\langle \mathbf{Z}_i | \mathbf{A} \rangle| = \text{amount of } \mathbf{A} \text{ in direction of } \mathbf{Z}_i$$

Realign data along dominant directions  $\{\mathbf{Z}_1, \dots, \mathbf{Z}_k, \mathbf{Z}_{k+1}, \dots, \mathbf{Z}_r\}$

— Project  $\mathbf{A}$  onto  $\text{span}\{\mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_k\}$

**Truncate:**  $\mathbf{A}_k = P(\mathbf{A}) = \sigma_1 \mathbf{Z}_1 + \sigma_2 \mathbf{Z}_2 + \cdots + \sigma_k \mathbf{Z}_k$



# Latent Semantic Indexing

Use a Fourier expansion of  $\mathbf{A}$

$$\mathbf{A} = \sum_{i=1}^r \sigma_i \mathbf{Z}_i, \quad \langle \mathbf{Z}_i | \mathbf{Z}_j \rangle = \begin{cases} 1 & i=j, \\ 0 & i \neq j, \end{cases} \quad |\sigma_1| \geq |\sigma_2| \geq \cdots \geq |\sigma_r|$$
$$|\sigma_i| = | \langle \mathbf{Z}_i | \mathbf{A} \rangle | = \text{amount of } \mathbf{A} \text{ in direction of } \mathbf{Z}_i$$

Realign data along dominant directions  $\{\mathbf{Z}_1, \dots, \mathbf{Z}_k, \mathbf{Z}_{k+1}, \dots, \mathbf{Z}_r\}$

— Project  $\mathbf{A}$  onto  $\text{span}\{\mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_k\}$

**Truncate:**  $\mathbf{A}_k = P(\mathbf{A}) = \sigma_1 \mathbf{Z}_1 + \sigma_2 \mathbf{Z}_2 + \cdots + \sigma_k \mathbf{Z}_k$

**LSI: Query matching with  $\mathbf{A}_k$  in place of  $\mathbf{A}$**

—  $D_2$  forced closer to  $D_1 \implies$  better chance of finding  $D_2$



# Latent Semantic Indexing

## Use a Fourier expansion of $\mathbf{A}$

$$\mathbf{A} = \sum_{i=1}^r \sigma_i \mathbf{Z}_i, \quad \langle \mathbf{Z}_i | \mathbf{Z}_j \rangle = \begin{cases} 1 & i=j, \\ 0 & i \neq j, \end{cases} \quad |\sigma_1| \geq |\sigma_2| \geq \cdots \geq |\sigma_r|$$
$$|\sigma_i| = | \langle \mathbf{Z}_i | \mathbf{A} \rangle | = \text{amount of } \mathbf{A} \text{ in direction of } \mathbf{Z}_i$$

## Realign data along dominant directions $\{\mathbf{Z}_1, \dots, \mathbf{Z}_k, \mathbf{Z}_{k+1}, \dots, \mathbf{Z}_r\}$

- Project  $\mathbf{A}$  onto  $\text{span}\{\mathbf{Z}_1, \mathbf{Z}_2, \dots, \mathbf{Z}_k\}$

**Truncate:**  $\mathbf{A}_k = P(\mathbf{A}) = \sigma_1 \mathbf{Z}_1 + \sigma_2 \mathbf{Z}_2 + \cdots + \sigma_k \mathbf{Z}_k$

## LSI: Query matching with $\mathbf{A}_k$ in place of $\mathbf{A}$

- $D_2$  forced closer to  $D_1 \implies$  better chance of finding  $D_2$

## “Best” mathematical solution

- SVD:  $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T = \sum \sigma_i \mathbf{u}_i \mathbf{v}_i^T$   $\mathbf{Z}_i = \mathbf{u}_i \mathbf{v}_i^T$





# Pros & Cons

## Cons

- Rankings are query dependent  
Rank of each doc is recomputed for each query
- Only semantic content used (Any link structure ignored)
- Difficult to add & delete documents
- Finding optimal  $k$  not easy (Empirical tuning required)
- Doesn't scale up well (Impractical for WWW)
- $\mathbf{u}_i, \mathbf{v}_i$  mixed sign  $\implies$  no good interpretation

## Pro

- Good at clustering  $\implies$  reveals patterns for text mining



# Pros & Cons

## Cons

- Rankings are query dependent  
Rank of each doc is recomputed for each query
- Only semantic content used (Any link structure ignored)
- Difficult to add & delete documents
- Finding optimal  $k$  not easy (Empirical tuning required)
- Doesn't scale up well (Impractical for WWW)
- $\mathbf{u}_i, \mathbf{v}_i$  mixed sign  $\implies$  no good interpretation

## Pro

- Good at clustering  $\implies$  reveals patterns for text mining



# Pros & Cons

## Cons

- Rankings are query dependent  
Rank of each doc is recomputed for each query
- Only semantic content used (Any link structure ignored)
- Difficult to add & delete documents
- Finding optimal  $k$  not easy (Empirical tuning required)
- Doesn't scale up well (Impractical for WWW)
- $\mathbf{u}_i, \mathbf{v}_i$  mixed sign  $\implies$  no good interpretation

## Pro

- Good at clustering  $\implies$  reveals patterns for text mining



# Pros & Cons

## Cons

- Rankings are query dependent  
Rank of each doc is recomputed for each query
- Only semantic content used (Any link structure ignored)
- Difficult to add & delete documents
- Finding optimal  $k$  not easy (Empirical tuning required)
- Doesn't scale up well (Impractical for WWW)
- $\mathbf{u}_i, \mathbf{v}_i$  mixed sign  $\implies$  no good interpretation

## Pro

- Good at clustering  $\implies$  reveals patterns for text mining



# Pros & Cons

## Cons

- Rankings are query dependent  
Rank of each doc is recomputed for each query
- Only semantic content used (Any link structure ignored)
- Difficult to add & delete documents
- Finding optimal  $k$  not easy (Empirical tuning required)
- Doesn't scale up well (Impractical for WWW)
- $\mathbf{u}_i, \mathbf{v}_i$  mixed sign  $\implies$  no good interpretation

## Pro

- Good at clustering  $\implies$  reveals patterns for text mining



# Pros & Cons

## Cons

- Rankings are query dependent  
Rank of each doc is recomputed for each query
- Only semantic content used (Any link structure ignored)
- Difficult to add & delete documents
- Finding optimal  $k$  not easy (Empirical tuning required)
- Doesn't scale up well (Impractical for WWW)
- $\mathbf{u}_i, \mathbf{v}_i$  mixed sign  $\implies$  no good interpretation

## Pro

- Good at clustering  $\implies$  reveals patterns for text mining



# Pros & Cons

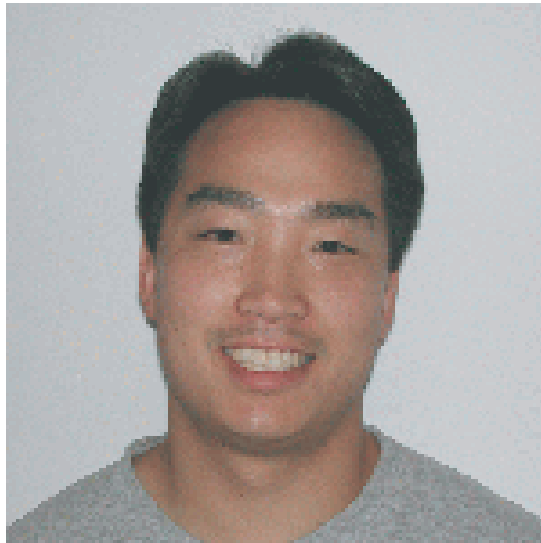
## Cons

- Rankings are query dependent  
Rank of each doc is recomputed for each query
- Only semantic content used (Any link structure ignored)
- Difficult to add & delete documents
- Finding optimal  $k$  not easy (Empirical tuning required)
- Doesn't scale up well (Impractical for WWW)
- $\mathbf{u}_i, \mathbf{v}_i$  mixed sign  $\implies$  no good interpretation

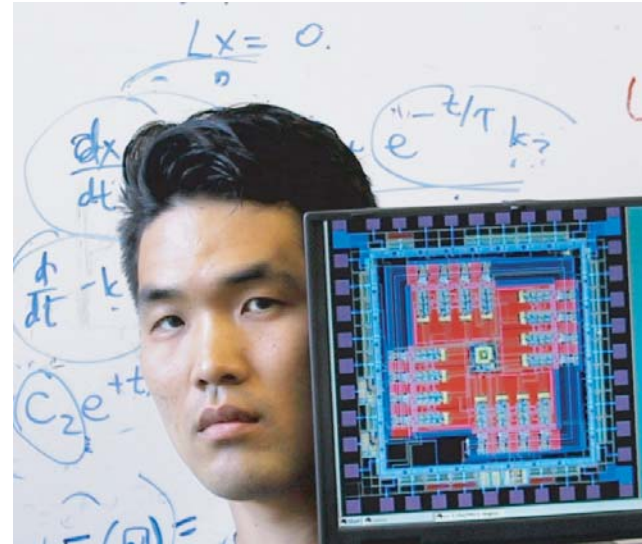
## Pro

- Good at clustering  $\implies$  reveals patterns for text mining

# Another Improvement (2000)



Daniel Lee



Sebastian Seung

Use low-rank approximation with **sparse nonnegative** factors

$$\begin{matrix} \mathbf{A}_{m \times n} \\ \text{nonneg} \end{matrix} \approx \begin{matrix} \mathbf{U}_{m \times k} \\ \text{mixed} \end{matrix} \begin{matrix} \mathbf{\Sigma}_{k \times k} \\ \text{nonneg} \end{matrix} \begin{matrix} \mathbf{V}_{k \times m}^T \\ \text{mixed} \end{matrix}$$

$$\begin{matrix} \mathbf{A}_{m \times n} \\ \text{nonneg} \end{matrix} \approx \begin{matrix} \mathbf{W}_{m \times k} \\ \text{nonneg} \end{matrix} \begin{matrix} \mathbf{H}_{k \times m} \\ \text{nonneg} \end{matrix}$$





# Nonnegative Matrix Factorization

## Constrained Nonlinear Least Squares Problem

$$\mathbf{A}_{m \times n} \approx \mathbf{W}_{m \times k} \mathbf{H}_{k \times n} \implies \begin{cases} \min \|\mathbf{A} - \mathbf{WH}\|_F^2 \\ \mathbf{W} \geq 0, \quad \mathbf{H} \geq 0, \quad \text{both sparse} \end{cases}$$



# Nonnegative Matrix Factorization

## Constrained Nonlinear Least Squares Problem

$$\mathbf{A}_{m \times n} \approx \mathbf{W}_{m \times k} \mathbf{H}_{k \times n} \implies \begin{cases} \min \|\mathbf{A} - \mathbf{WH}\|_F^2 \\ \mathbf{W} \geq 0, \quad \mathbf{H} \geq 0, \quad \text{both sparse} \end{cases}$$

$\mathbf{W}_k = [\mathbf{w}_1 | \mathbf{w}_2 | \dots | \mathbf{w}_k]$  yields sparse nonnegative basis

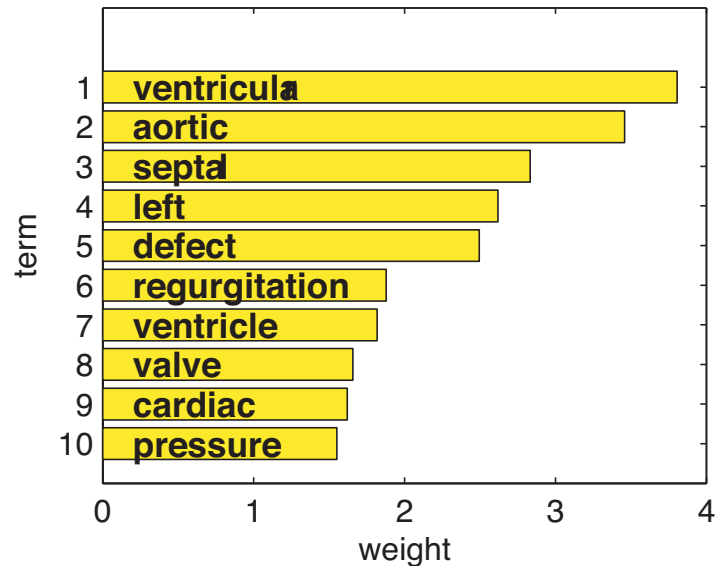
$$\begin{array}{c} \text{term 1} \\ \text{term 2} \\ \vdots \\ \text{term m} \end{array} \begin{array}{c} \text{doc}_j \\ \left[ \begin{array}{c} \vdots \\ \mathbf{A}_j \\ \vdots \end{array} \right] \end{array} \approx \left[ \begin{array}{c} \vdots \\ \mathbf{w}_1 \\ \vdots \end{array} \right] h_{1j} + \left[ \begin{array}{c} \vdots \\ \mathbf{w}_2 \\ \vdots \end{array} \right] h_{2j} + \dots + \left[ \begin{array}{c} \vdots \\ \mathbf{w}_k \\ \vdots \end{array} \right] h_{kj}$$

- Each  $\mathbf{w}_i$  can be interpreted as a topic vector
  - Large  $\{w_{ij}, w_{ik}, \dots, w_{il}\} \implies \mathbf{w}_i$  mostly about terms  $j, k, \dots, l$
  - $h_{ij}$  indicates how much of  $\text{doc}_j$  is related to topic vector  $\mathbf{w}_i$

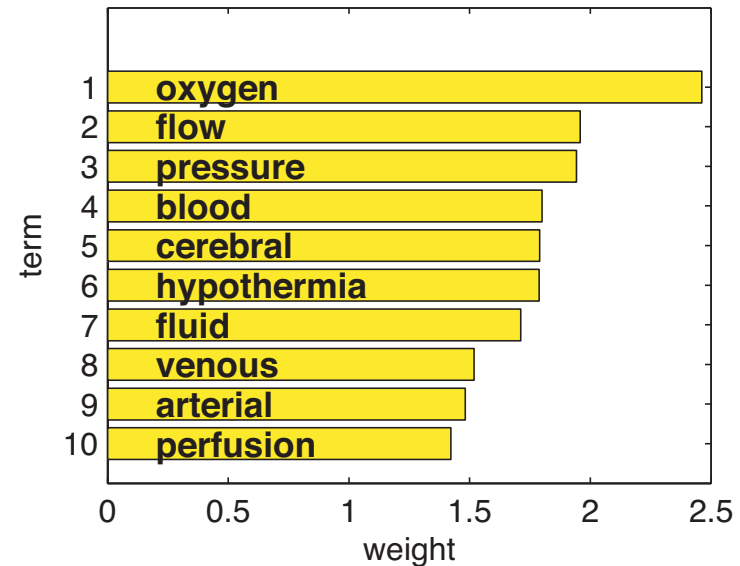
# Example

(MEDLINE Amy Langville  $k = 10$ )

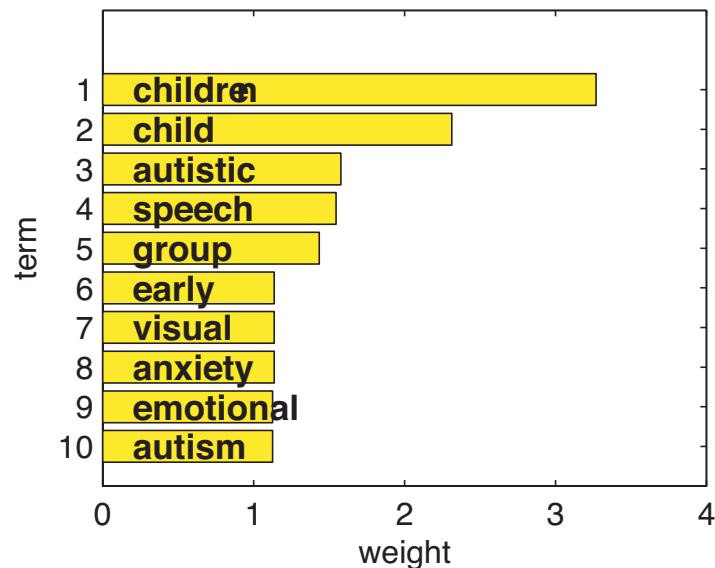
Highest Weighted Terms in Basis Vector  $W_1$



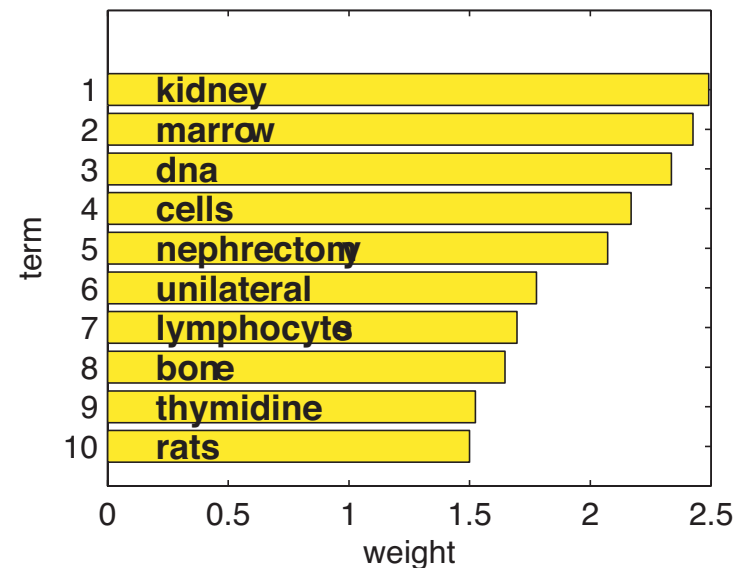
Highest Weighted Terms in Basis Vector  $W_2$



Highest Weighted Terms in Basis Vector  $W_5$



Highest Weighted Terms in Basis Vector  $W_6$



# Example (cont)

$$\mathbf{doc}_5 \approx \begin{pmatrix} \mathbf{w}_9 \\ \text{fatty} \\ \text{glucose} \\ \text{acids} \\ \text{ffa} \\ \text{insulin} \\ \vdots \end{pmatrix} .1646 + \begin{pmatrix} \mathbf{w}_6 \\ \text{kidney} \\ \text{marrow} \\ \text{dna} \\ \text{cells} \\ \text{neph.} \\ \vdots \end{pmatrix} .0103 + \begin{pmatrix} \mathbf{w}_7 \\ \text{hormone} \\ \text{growth} \\ \text{hgh} \\ \text{pituitary} \\ \text{mg} \\ \vdots \end{pmatrix} .0045 + \dots$$



# Enron E-mail Data (1999–2001)

**Fed investigation studied 15 million e-mail messages**

- Over 500,000 messages made public



# Enron E-mail Data (1999–2001)

## Fed investigation studied 15 million e-mail messages

- Over 500,000 messages made public

## Enron's Troubles 1999-2001

- Problems with Dabhol Power Company (DPC) in India



# Enron E-mail Data (1999–2001)

## Fed investigation studied 15 million e-mail messages

- Over 500,000 messages made public

## Enron's Troubles 1999-2001

- Problems with Dabhol Power Company (DPC) in India
- Deregulation of Calif. energy industry
  - ▷ Rolling blackouts in the summer of 2000
  - ▷ Subsequent investigations



# Enron E-mail Data (1999–2001)

## Fed investigation studied 15 million e-mail messages

- Over 500,000 messages made public

## Enron's Troubles 1999-2001

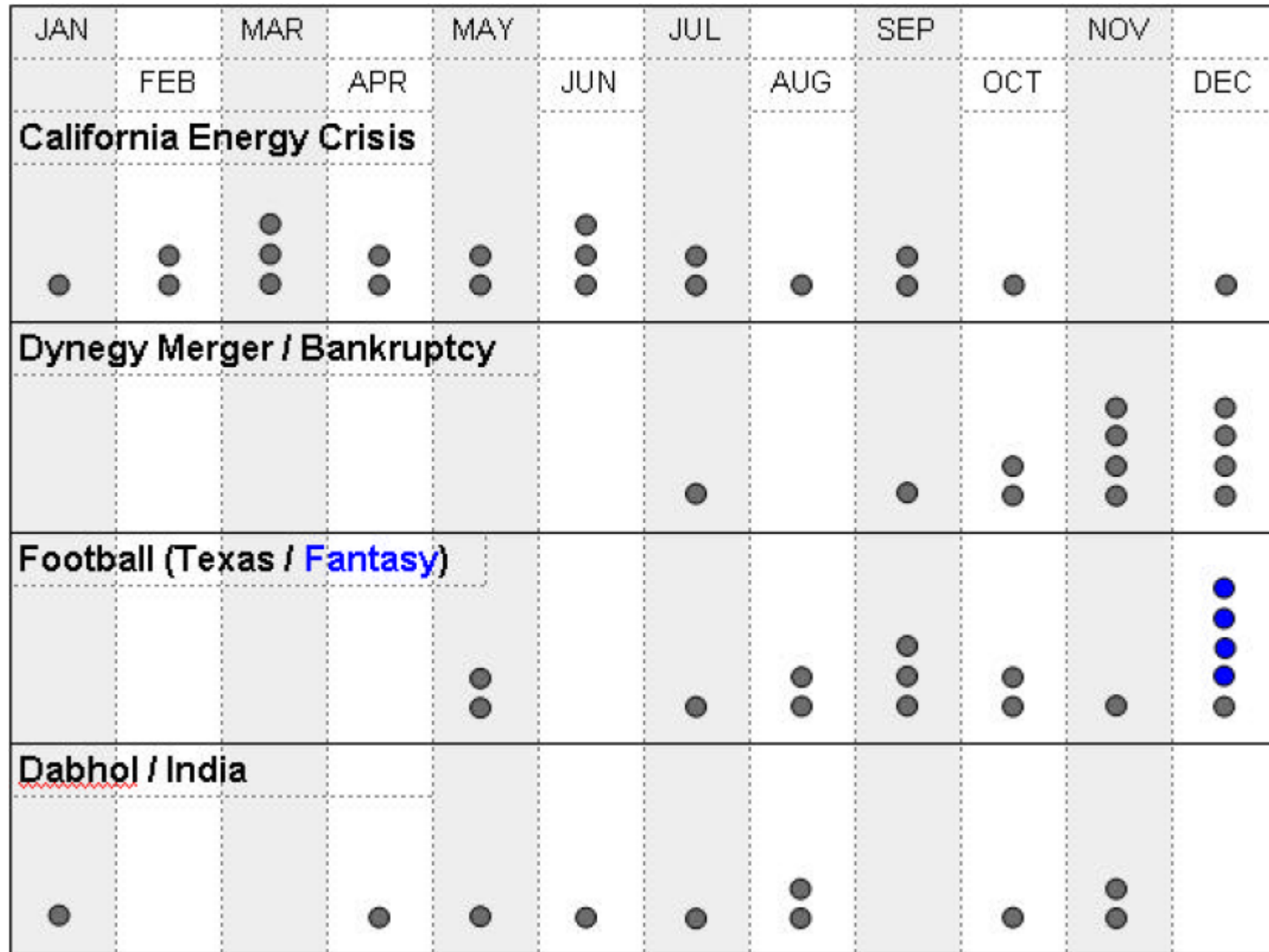
- Problems with Dabhol Power Company (DPC) in India
- Deregulation of Calif. energy industry
  - ▷ Rolling blackouts in the summer of 2000
  - ▷ Subsequent investigations
- Ill-fated Dynergy merger, Oct-Nov 2001
  - ▷ Revelation of Enron's deceptive practices
  - ▷ Enron filed for bankruptcy in December 2001





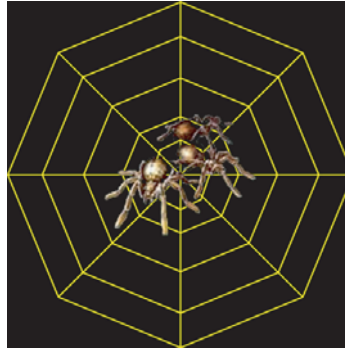
# Mining 2001 E-mail

(M. Berry, Univ. Tenn)



# Web Search Components

## Web Crawlers

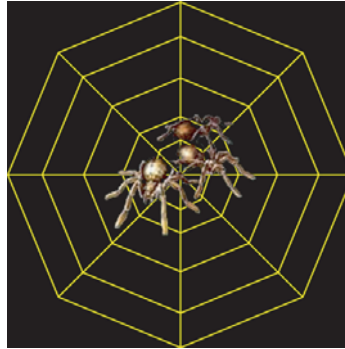


Software robots  
gather web pages



# Web Search Components

## Web Crawlers



Software robots  
gather web pages

## Doc Server

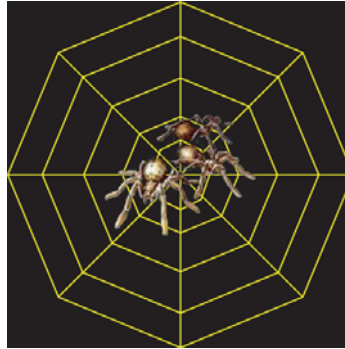


Stores docs  
and snippets



# Web Search Components

## Web Crawlers



Software robots  
gather web pages

## Doc Server



Stores docs  
and snippets

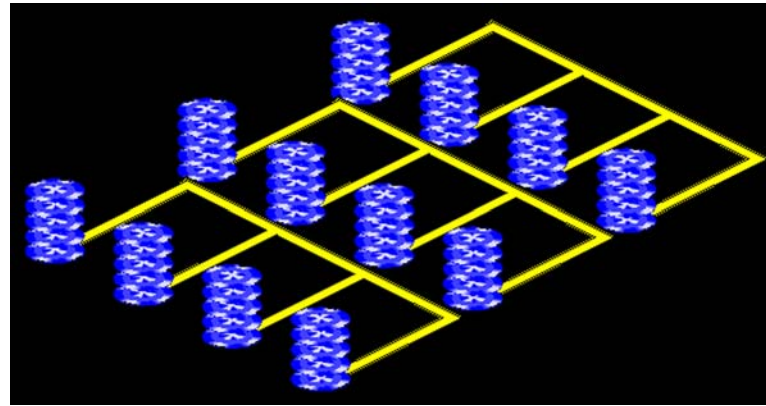
## Index Server



Scans pages and does term indexing  
Terms → Pages (similar to book index)

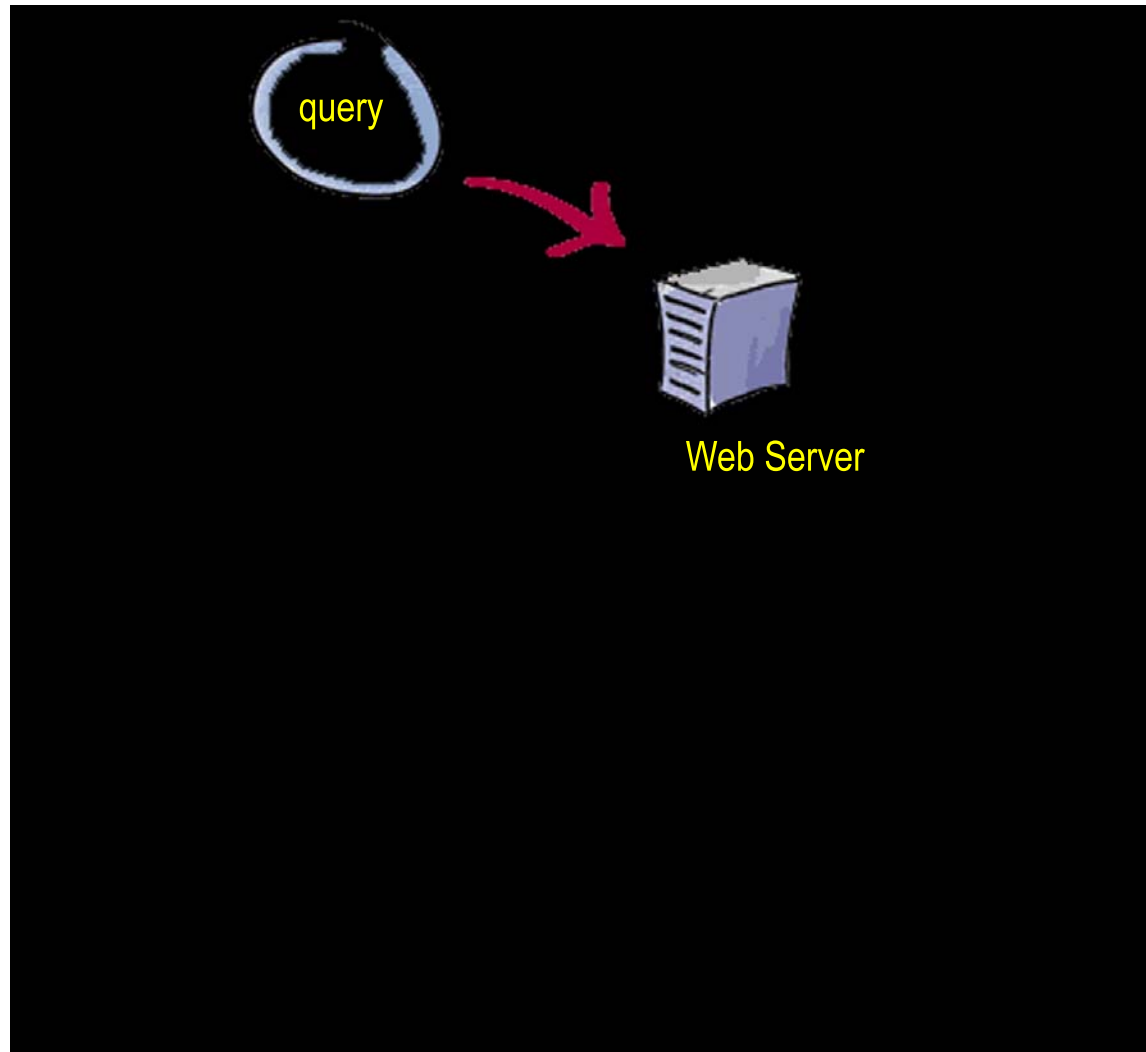
# The Heart of a Search Engine

## The Ranking Module

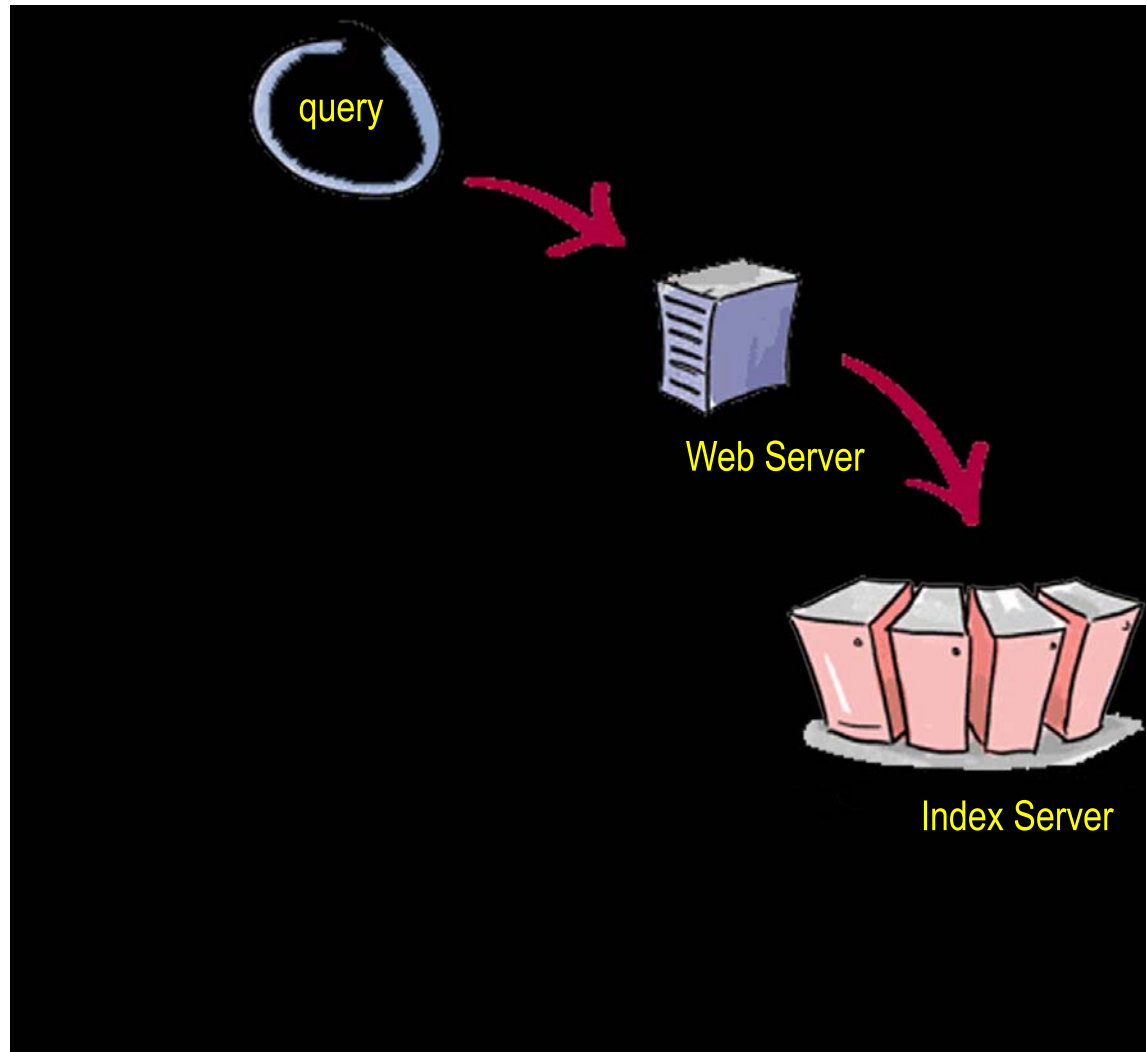


- Assign an importance value to each page
  - ▷ Independent of any query
- Google's PageRank<sup>©</sup> technology distinguishes it from all competitors

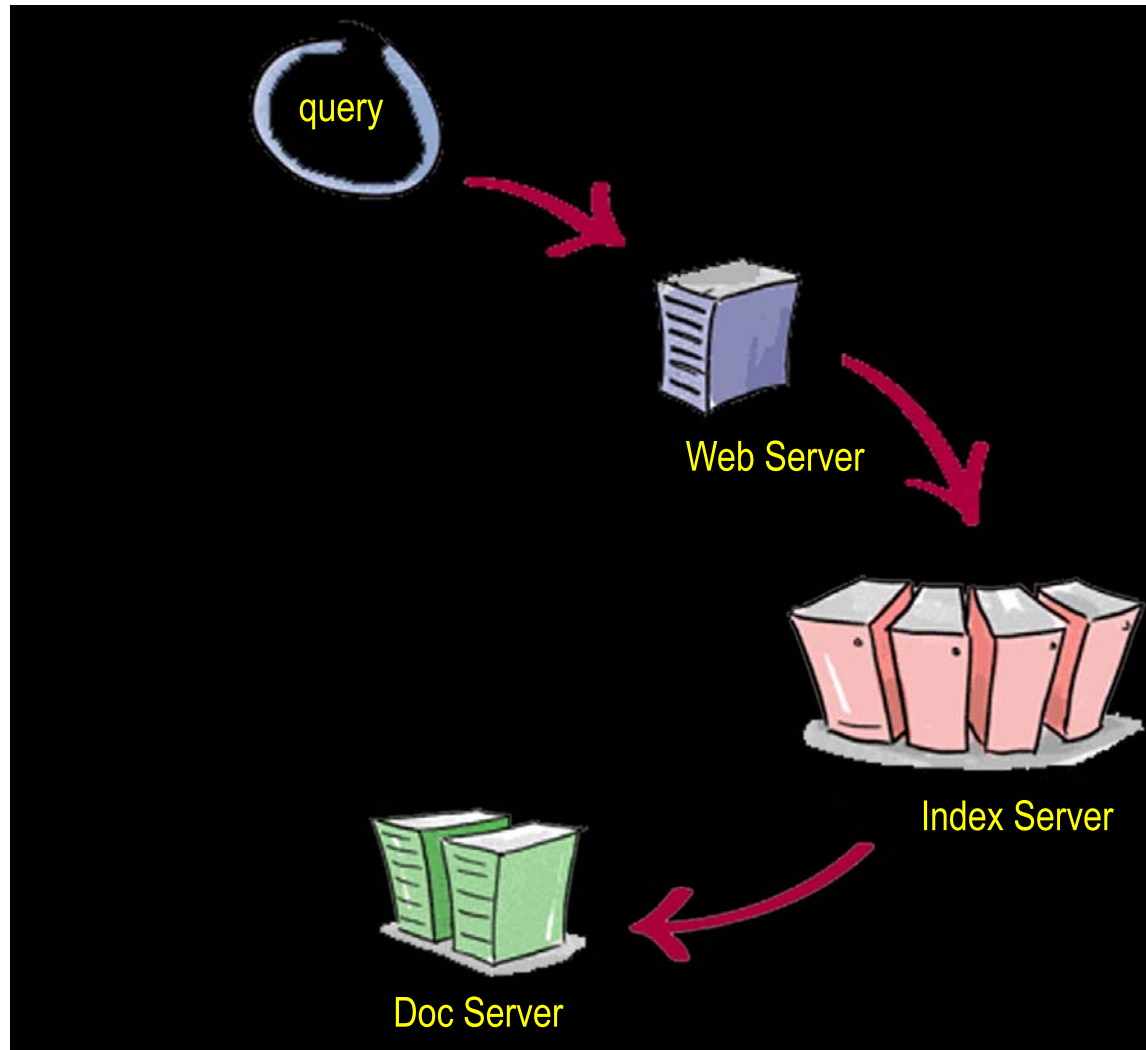
# The Process



# The Process

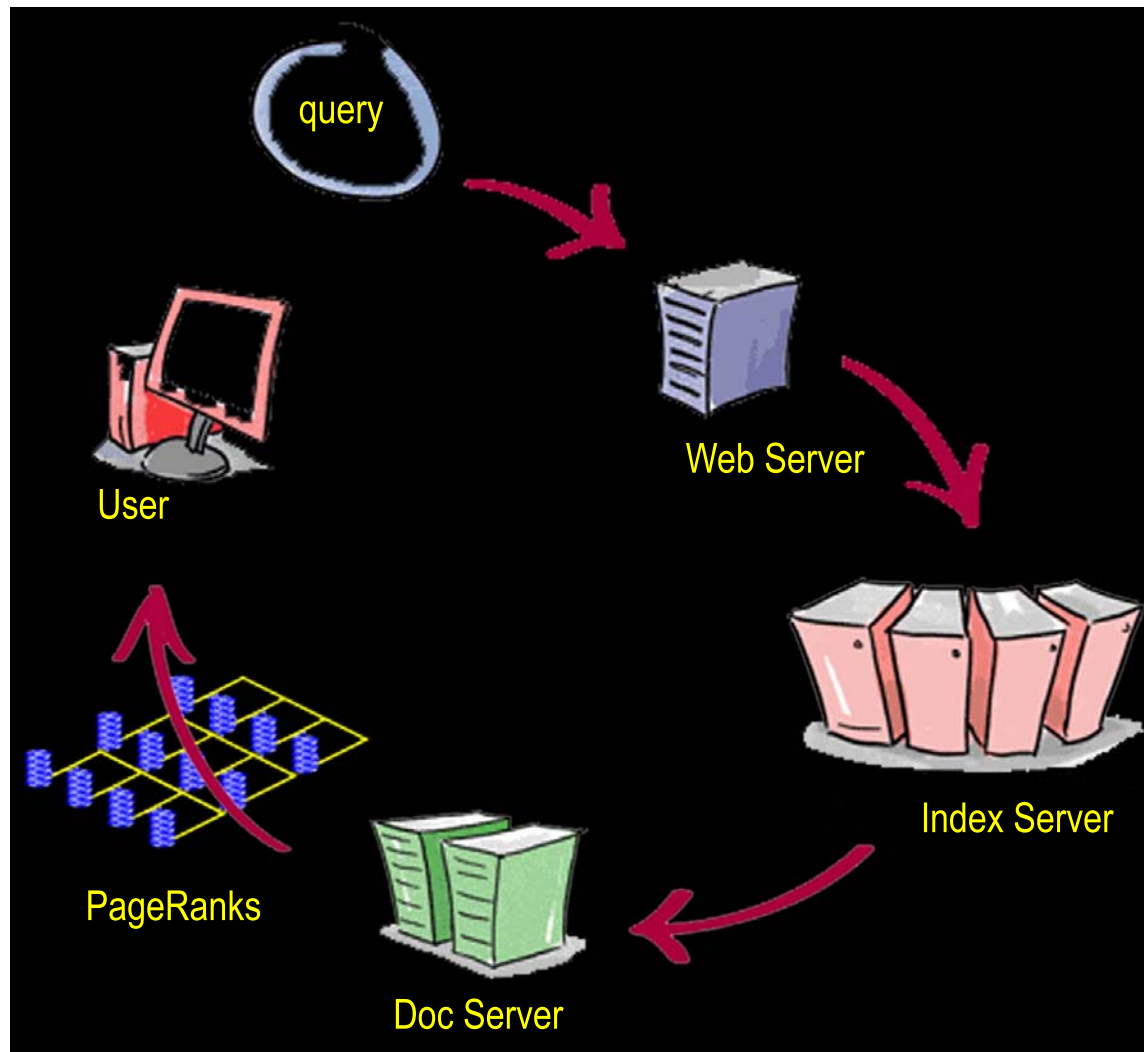


# The Process



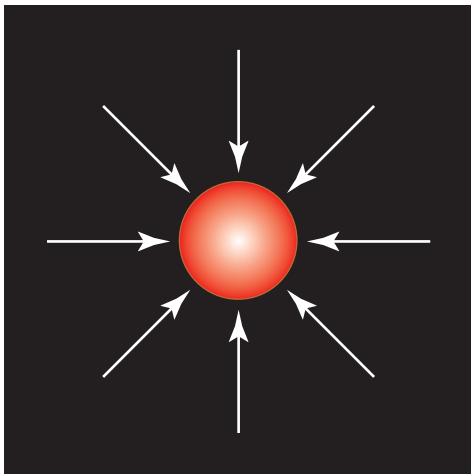


# The Process

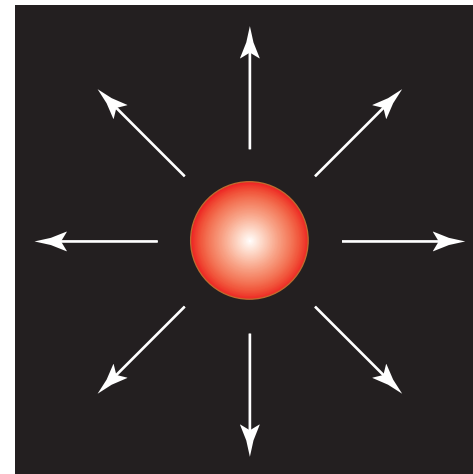


# How To Measure “Importance”

Authorities

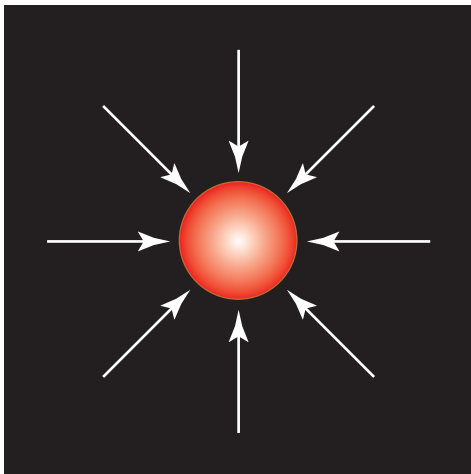


Hubs

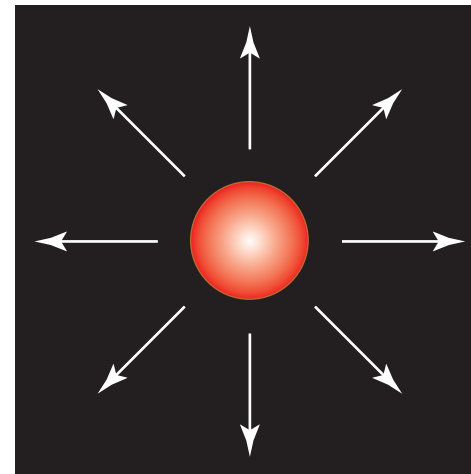


# How To Measure “Importance”

Authorities



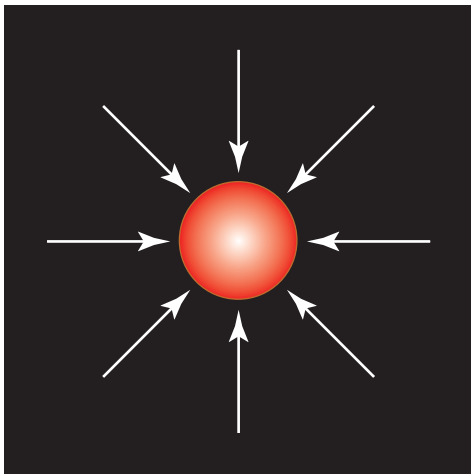
Hubs



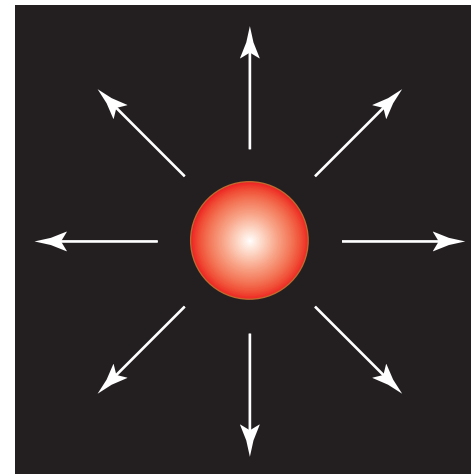
- Good hub pages point to good authority pages

# How To Measure “Importance”

Authorities



Hubs



- Good hub pages point to good authority pages
- Good authorities are pointed to by good hubs

# HITS Algorithm

Hypertext Induced Topic Search (1998)

## Determine Authority & Hub Scores

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



Jon Kleinberg

# HITS Algorithm

Hypertext Induced Topic Search (1998)



Jon Kleinberg

## Determine Authority & Hub Scores

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

## Successive Refinement

- Start with  $h_i = 1$  for all pages  $P_i \Rightarrow \mathbf{h}_0 = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$

# HITS Algorithm

Hypertext Induced Topic Search (1998)



Jon Kleinberg

## Determine Authority & Hub Scores

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

## Successive Refinement

- Start with  $h_i = 1$  for all pages  $P_i \Rightarrow \mathbf{h}_0 =$
- Define Authority Scores (first iterate)

$$\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

$$a_i = \sum_{j: P_j \rightarrow P_i} h_j$$

# HITS Algorithm

Hypertext Induced Topic Search (1998)



Jon Kleinberg

## Determine Authority & Hub Scores

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

## Successive Refinement

- Start with  $h_i = 1$  for all pages  $P_i \Rightarrow \mathbf{h}_0 = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$
- Define Authority Scores (first iterate)

$$a_i = \sum_{j: P_j \rightarrow P_i} h_j \Rightarrow \mathbf{a}_1 = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \mathbf{L}^T \mathbf{h}_0$$

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





# HITS Algorithm

## Refine Hub Scores

- $h_i = \sum_{j:P_i \rightarrow P_j} a_j \Rightarrow \mathbf{h}_1 = \mathbf{L}\mathbf{a}_1$

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



# HITS Algorithm

## Refine Hub Scores

- $h_i = \sum_{j:P_i \rightarrow P_j} a_j \Rightarrow \mathbf{h}_1 = \mathbf{L}\mathbf{a}_1$

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

## Successively Re-refine Authority & Hub Scores

- $\mathbf{a}_2 = \mathbf{L}^T \mathbf{h}_1$



# HITS Algorithm

## Refine Hub Scores

- $h_i = \sum_{j:P_i \rightarrow P_j} a_j \Rightarrow \mathbf{h}_1 = \mathbf{L}\mathbf{a}_1$

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

## Successively Re-refine Authority & Hub Scores

- $\mathbf{a}_2 = \mathbf{L}^T \mathbf{h}_1$ 
  - $\mathbf{h}_2 = \mathbf{L}\mathbf{a}_2$



# HITS Algorithm

## Refine Hub Scores

- $h_i = \sum_{j:P_i \rightarrow P_j} a_j \Rightarrow \mathbf{h}_1 = \mathbf{L}\mathbf{a}_1$

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

## Successively Re-refine Authority & Hub Scores

- $\mathbf{a}_2 = \mathbf{L}^T \mathbf{h}_1$

- $\mathbf{h}_2 = \mathbf{L}\mathbf{a}_2$

- $\mathbf{a}_3 = \mathbf{L}^T \mathbf{h}_2$



# HITS Algorithm

## Refine Hub Scores

- $h_i = \sum_{j:P_i \rightarrow P_j} a_j \Rightarrow \mathbf{h}_1 = \mathbf{L}\mathbf{a}_1$

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

## Successively Re-refine Authority & Hub Scores

- $\mathbf{a}_2 = \mathbf{L}^T \mathbf{h}_1$

- $\mathbf{h}_2 = \mathbf{L}\mathbf{a}_2$

- $\mathbf{a}_3 = \mathbf{L}^T \mathbf{h}_2$

- $\mathbf{h}_3 = \mathbf{L}\mathbf{a}_3$

•  
•  
•

# HITS Algorithm

## Refine Hub Scores

$$\bullet \quad h_i = \sum_{j: P_i \rightarrow P_j} a_j \quad \Rightarrow \quad \mathbf{h}_1 = \mathbf{L}\mathbf{a}_1 \quad L_{ij} = \begin{cases} 1 & P_i \rightarrow P_j \\ 0 & P_i \not\rightarrow P_j \end{cases}$$

## Successively Re-refine Authority & Hub Scores

$$\begin{aligned} \bullet \quad \mathbf{a}_2 &= \mathbf{L}^T \mathbf{h}_1 \\ \bullet \quad \mathbf{h}_2 &= \mathbf{L}\mathbf{a}_2 \\ \bullet \quad \mathbf{a}_3 &= \mathbf{L}^T \mathbf{h}_2 \\ \bullet \quad \mathbf{h}_3 &= \mathbf{L}\mathbf{a}_3 \\ &\vdots \end{aligned}$$

## Combined Iterations

$$\begin{aligned} \bullet \quad \mathbf{A} &= \mathbf{L}^T \mathbf{L} \text{ (authority matrix)} & \mathbf{a}_k &= \mathbf{A}\mathbf{a}_{k-1} \\ \bullet \quad \mathbf{H} &= \mathbf{L}\mathbf{L}^T \text{ (hub matrix)} & \mathbf{h}_k &= \mathbf{H}\mathbf{h}_{k-1} \end{aligned}$$



# HITS Algorithm

## Refine Hub Scores

$$\bullet \quad h_i = \sum_{j: P_i \rightarrow P_j} a_j \quad \Rightarrow \quad \mathbf{h}_1 = \mathbf{L}\mathbf{a}_1 \quad L_{ij} = \begin{cases} 1 & P_i \rightarrow P_j \\ 0 & P_i \not\rightarrow P_j \end{cases}$$

## Successively Re-refine Authority & Hub Scores

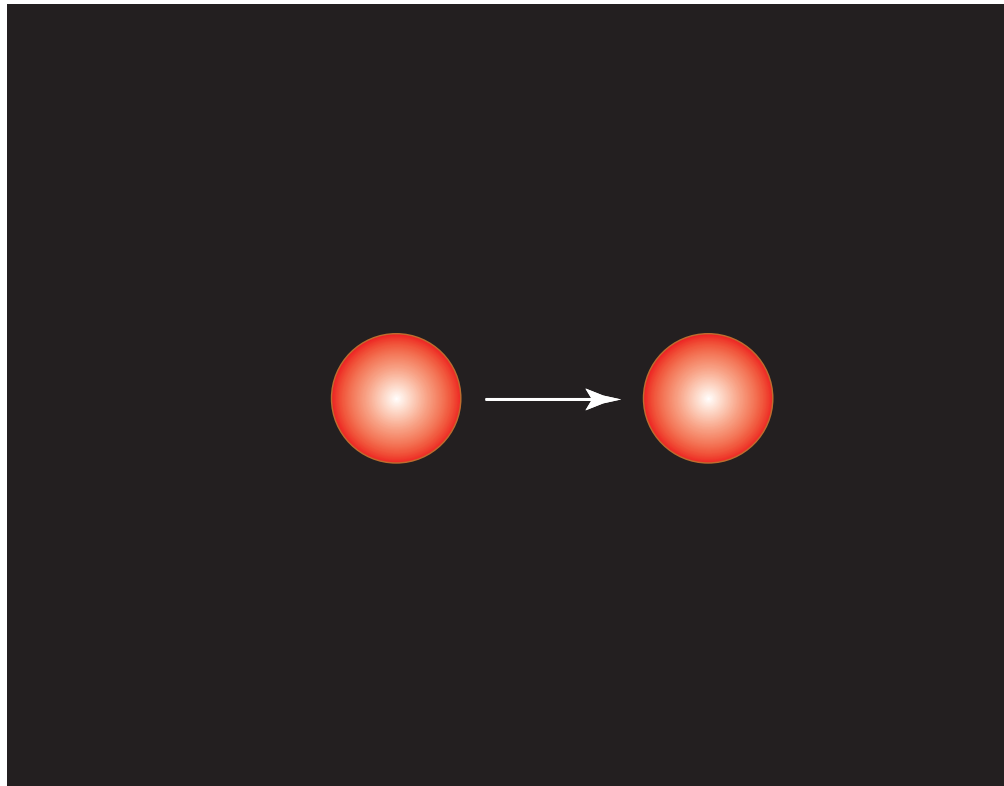
$$\begin{aligned} \bullet \quad \mathbf{a}_2 &= \mathbf{L}^T \mathbf{h}_1 \\ \bullet \quad \mathbf{h}_2 &= \mathbf{L} \mathbf{a}_2 \\ \bullet \quad \mathbf{a}_3 &= \mathbf{L}^T \mathbf{h}_2 \\ \bullet \quad \mathbf{h}_3 &= \mathbf{L} \mathbf{a}_3 \\ &\vdots \end{aligned}$$

## Combined Iterations

$$\begin{aligned} \bullet \quad \mathbf{A} &= \mathbf{L}^T \mathbf{L} \text{ (authority matrix)} & \mathbf{a}_k &= \mathbf{A} \mathbf{a}_{k-1} \rightarrow \text{e-vector} & \text{(direction)} \\ \bullet \quad \mathbf{H} &= \mathbf{L} \mathbf{L}^T \text{ (hub matrix)} & \mathbf{h}_k &= \mathbf{H} \mathbf{h}_{k-1} \rightarrow \text{e-vector} & \text{(direction)} \end{aligned}$$

# Compromise

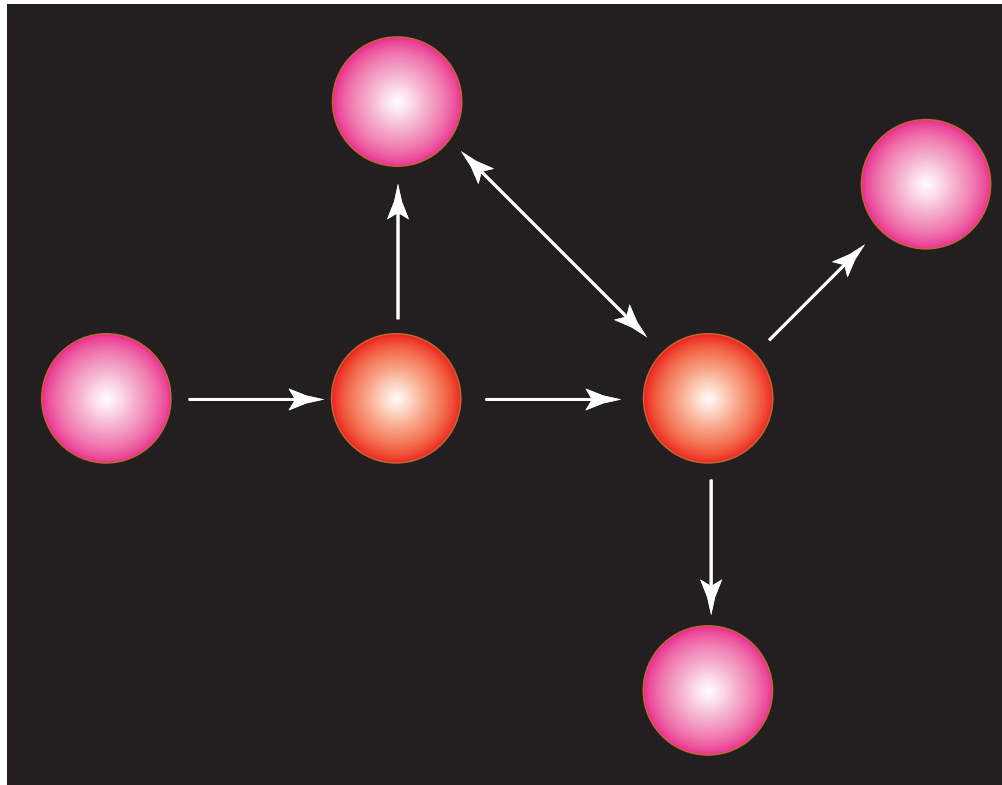
1. Do direct query matching





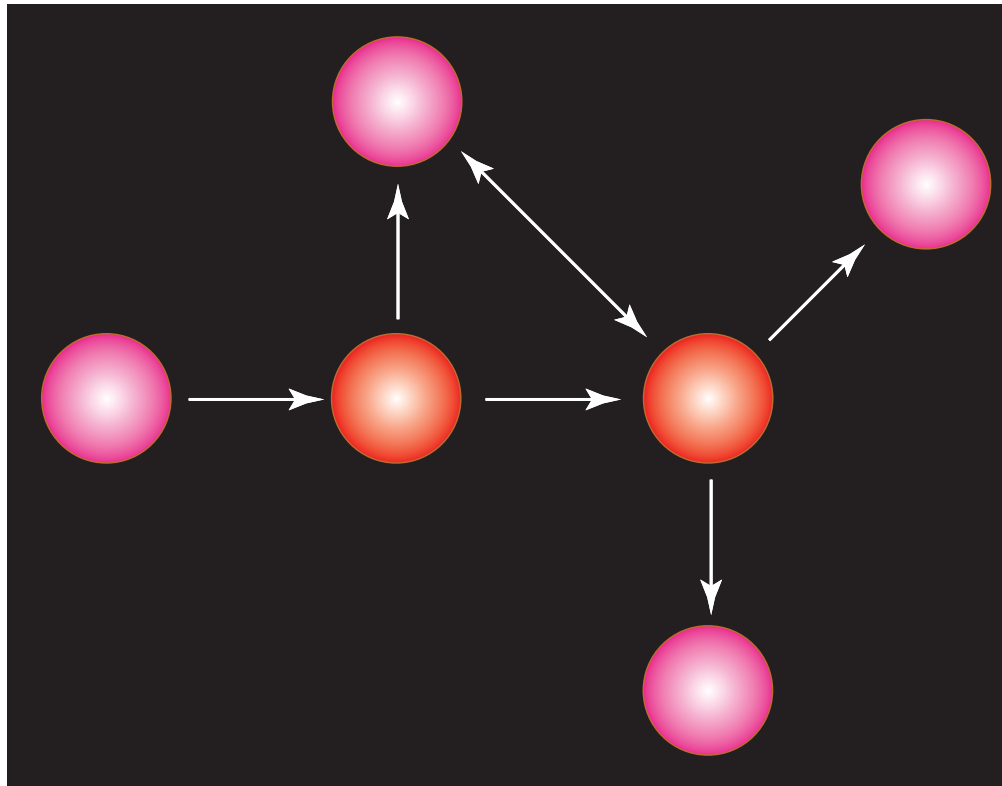
# Compromise

1. Do direct query matching
2. Build neighborhood graph



# Compromise

1. Do direct query matching
2. Build neighborhood graph



3. Compute authority & hub scores for just the neighborhood



# Pros & Cons

## Advantages

- Returns satisfactory results



# Pros & Cons

## Advantages

- Returns satisfactory results
  - Client gets both authority & hub scores



# Pros & Cons

## Advantages

- Returns satisfactory results
  - Client gets both authority & hub scores
- Some flexibility for making refinements



# Pros & Cons

## Advantages

- Returns satisfactory results
  - Client gets both authority & hub scores
- Some flexibility for making refinements

## Disadvantages

- Too much has to happen while client is waiting



# Pros & Cons

## Advantages

- Returns satisfactory results
  - Client gets both authority & hub scores
- Some flexibility for making refinements

## Disadvantages

- Too much has to happen while client is waiting
  - Custom built neighborhood graph needed for each query



# Pros & Cons

## Advantages

- Returns satisfactory results
  - Client gets both authority & hub scores
- Some flexibility for making refinements

## Disadvantages

- Too much has to happen while client is waiting
  - Custom built neighborhood graph needed for each query
  - Two eigenvector computations needed for each query





# Pros & Cons

## Advantages

- Returns satisfactory results
  - Client gets both authority & hub scores
- Some flexibility for making refinements

## Disadvantages

- Too much has to happen while client is waiting
  - Custom built neighborhood graph needed for each query
  - Two eigenvector computations needed for each query
- Scores can be manipulated by creating artificial hubs

# Newsweek

March 29, 2004 : \$3.95

newsweek.msnbc.com

**The Next Frontiers**

## **The New Age of** **Google**

**The Search Giant Has Changed  
Our Lives. Can Anybody  
Catch These Guys? By Steven Levy**

Google founders Larry Page and Sergey Brin



# Google's PageRank

(Lawrence Page & Sergey Brin 1998)

## The Google Goals

- Create a PageRank  $r(P)$  that is not query dependent
  - ▷ Off-line calculations — No query time computation
- Let the Web determine importance
  - ▷ But not by simple link counts
    - One link to  $P$  from Yahoo! is important
    - Many links to  $P$  from me is not
- Share The Vote
  - ▷ Yahoo! casts many “votes”
    - value of vote from *Yahoo!* is diluted
  - ▷ If Yahoo! “votes” for  $n$  pages
    - Then  $P$  receives only  $r(Y)/n$  credit from  $Y$



# Google's PageRank

(Lawrence Page & Sergey Brin 1998)

## The Google Goals

- Create a PageRank  $r(P)$  that is not query dependent
  - ▷ Off-line calculations — No query time computation
- Let the Web determine importance
  - ▷ But not by simple link counts
    - One link to  $P$  from Yahoo! is important
    - Many links to  $P$  from me is not
- Share The Vote
  - ▷ Yahoo! casts many “votes”
    - value of vote from *Yahoo!* is diluted
  - ▷ If Yahoo! “votes” for  $n$  pages
    - Then  $P$  receives only  $r(Y)/n$  credit from  $Y$



# Google's PageRank

(Lawrence Page & Sergey Brin 1998)

## The Google Goals

- Create a PageRank  $r(P)$  that is not query dependent
  - ▷ Off-line calculations — No query time computation
- Let the Web determine importance
  - ▷ But not by simple link counts
    - One link to  $P$  from Yahoo! is important
    - Many links to  $P$  from me is not
- Share The Vote
  - ▷ Yahoo! casts many “votes”
    - value of vote from *Yahoo!* is diluted
  - ▷ If Yahoo! “votes” for  $n$  pages
    - Then  $P$  receives only  $r(Y)/n$  credit from  $Y$



# PageRank

## The Definition

$$r(P) = \sum_{P \in \mathcal{B}_P} \frac{r(P)}{|P|}$$

$\mathcal{B}_P = \{\text{all pages pointing to } P\}$

$|P| = \text{number of out links from } P$



# PageRank

## The Definition

$$r(P) = \sum_{P \in \mathcal{B}_P} \frac{r(P)}{|P|}$$

$\mathcal{B}_P = \{\text{all pages pointing to } P\}$

$|P| = \text{number of out links from } P$

## Successive Refinement

Start with  $r_0(P_i) = 1/n$  for all pages  $P_1, P_2, \dots, P_n$



# PageRank

## The Definition

$$r(P) = \sum_{P \in \mathcal{B}_P} \frac{r(P)}{|P|}$$

$\mathcal{B}_P = \{\text{all pages pointing to } P\}$

$|P| = \text{number of out links from } P$

## Successive Refinement

Start with  $r_0(P_i) = 1/n$  for all pages  $P_1, P_2, \dots, P_n$

Iteratively refine rankings for each page

$$r_1(P_i) = \sum_{P \in \mathcal{B}_{P_i}} \frac{r_0(P)}{|P|}$$





# PageRank

## The Definition

$$r(P) = \sum_{P \in \mathcal{B}_P} \frac{r(P)}{|P|}$$

$\mathcal{B}_P = \{\text{all pages pointing to } P\}$

$|P| = \text{number of out links from } P$

## Successive Refinement

Start with  $r_0(P_i) = 1/n$  for all pages  $P_1, P_2, \dots, P_n$

Iteratively refine rankings for each page

$$r_1(P_i) = \sum_{P \in \mathcal{B}_{P_i}} \frac{r_0(P)}{|P|}$$

$$r_2(P_i) = \sum_{P \in \mathcal{B}_{P_i}} \frac{r_1(P)}{|P|}$$



# PageRank

## The Definition

$$r(P) = \sum_{P \in \mathcal{B}_P} \frac{r(P)}{|P|}$$

$\mathcal{B}_P = \{\text{all pages pointing to } P\}$

$|P| = \text{number of out links from } P$

## Successive Refinement

Start with  $r_0(P_i) = 1/n$  for all pages  $P_1, P_2, \dots, P_n$

Iteratively refine rankings for each page

$$r_1(P_i) = \sum_{P \in \mathcal{B}_{P_i}} \frac{r_0(P)}{|P|}$$

$$r_2(P_i) = \sum_{P \in \mathcal{B}_{P_i}} \frac{r_1(P)}{|P|}$$

$\vdots$

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



# In Matrix Notation

After Step  $j$

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



# In Matrix Notation

After Step  $j$

$$\pi_j^T = [r_j(P_1), r_j(P_2), \dots, r_j(P_n)]$$

$$\pi_{j+1}^T = \pi_j^T \mathbf{P} \quad \text{where} \quad p_{ij} = \begin{cases} 1/|P_i| & \text{if } i \rightarrow j \\ 0 & \text{otherwise} \end{cases}$$



# In Matrix Notation

After Step  $j$

$$\pi_j^T = [r_j(P_1), r_j(P_2), \dots, r_j(P_n)]$$

$$\pi_{j+1}^T = \pi_j^T \mathbf{P} \quad \text{where} \quad p_{ij} = \begin{cases} 1/|P_i| & \text{if } i \rightarrow j \\ 0 & \text{otherwise} \end{cases}$$

$$\text{PageRank} = \lim_{j \rightarrow \infty} \pi_j^T = \pi^T \quad \text{(provided limit exists)}$$



# In Matrix Notation

After Step  $j$

$$\pi_j^T = [r_j(P_1), r_j(P_2), \dots, r_j(P_n)]$$

$$\pi_{j+1}^T = \pi_j^T \mathbf{P} \quad \text{where} \quad p_{ij} = \begin{cases} 1/|P_i| & \text{if } i \rightarrow j \\ 0 & \text{otherwise} \end{cases}$$

$$\text{PageRank} = \lim_{j \rightarrow \infty} \pi_j^T = \pi^T \quad (\text{provided limit exists})$$

A Markov Chain?

If  $\mathbf{P} = [p_{ij}]$  is a stochastic matrix (  $p_{ij} \geq 0$  and  $\sum_j p_{ij} = 1$  )



# In Matrix Notation

## After Step $j$

$$\pi_j^T = [r_j(P_1), r_j(P_2), \dots, r_j(P_n)]$$

$$\pi_{j+1}^T = \pi_j^T \mathbf{P} \quad \text{where} \quad p_{ij} = \begin{cases} 1/|P_i| & \text{if } i \rightarrow j \\ 0 & \text{otherwise} \end{cases}$$

$$\text{PageRank} = \lim_{j \rightarrow \infty} \pi_j^T = \pi^T \quad (\text{provided limit exists})$$

## A Markov Chain?

If  $\mathbf{P} = [p_{ij}]$  is a stochastic matrix (  $p_{ij} \geq 0$  and  $\sum_j p_{ij} = 1$  )

Each  $\pi_j^T$  is a probability vector (  $\pi_i \geq 0$  and  $\sum_i \pi_i = 1$  )



# In Matrix Notation

## After Step $j$

$$\pi_j^T = [r_j(P_1), r_j(P_2), \dots, r_j(P_n)]$$

$$\pi_{j+1}^T = \pi_j^T \mathbf{P} \quad \text{where} \quad p_{ij} = \begin{cases} 1/|P_i| & \text{if } i \rightarrow j \\ 0 & \text{otherwise} \end{cases}$$

$$\text{PageRank} = \lim_{j \rightarrow \infty} \pi_j^T = \pi^T \quad (\text{provided limit exists})$$

## A Markov Chain?

If  $\mathbf{P} = [p_{ij}]$  is a stochastic matrix (  $p_{ij} \geq 0$  and  $\sum_j p_{ij} = 1$  )

Each  $\pi_j^T$  is a probability vector (  $\pi_i \geq 0$  and  $\sum_i \pi_i = 1$  )

$\pi_{j+1}^T = \pi_j^T \mathbf{P}$  is random walk on the graph defined by links





# In Matrix Notation

## After Step $j$

$$\pi_j^T = [r_j(P_1), r_j(P_2), \dots, r_j(P_n)]$$

$$\pi_{j+1}^T = \pi_j^T \mathbf{P} \quad \text{where} \quad p_{ij} = \begin{cases} 1/|P_i| & \text{if } i \rightarrow j \\ 0 & \text{otherwise} \end{cases}$$

$$\text{PageRank} = \lim_{j \rightarrow \infty} \pi_j^T = \pi^T \quad (\text{provided limit exists})$$

## A Markov Chain?

If  $\mathbf{P} = [p_{ij}]$  is a stochastic matrix (  $p_{ij} \geq 0$  and  $\sum_j p_{ij} = 1$  )

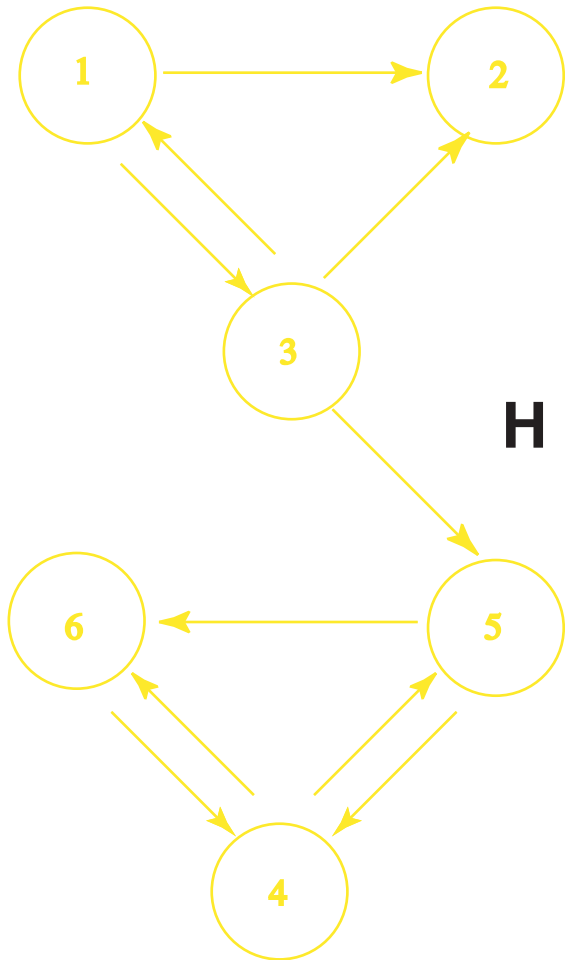
Each  $\pi_j^T$  is a probability vector (  $\pi_i \geq 0$  and  $\sum_i \pi_i = 1$  )

$\pi_{j+1}^T = \pi_j^T \mathbf{P}$  is random walk on the graph defined by links

$\pi^T = \lim_{j \rightarrow \infty} \pi_j^T$  = steady-state probability distribution



# Tiny Web

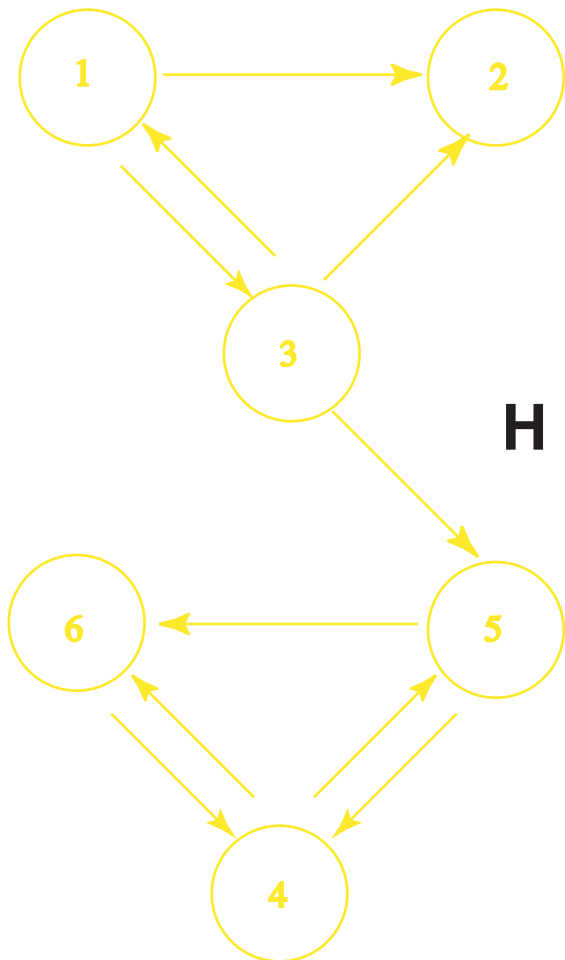


**H** =

$$\begin{matrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{matrix} \begin{pmatrix} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \end{pmatrix}$$



# Tiny Web

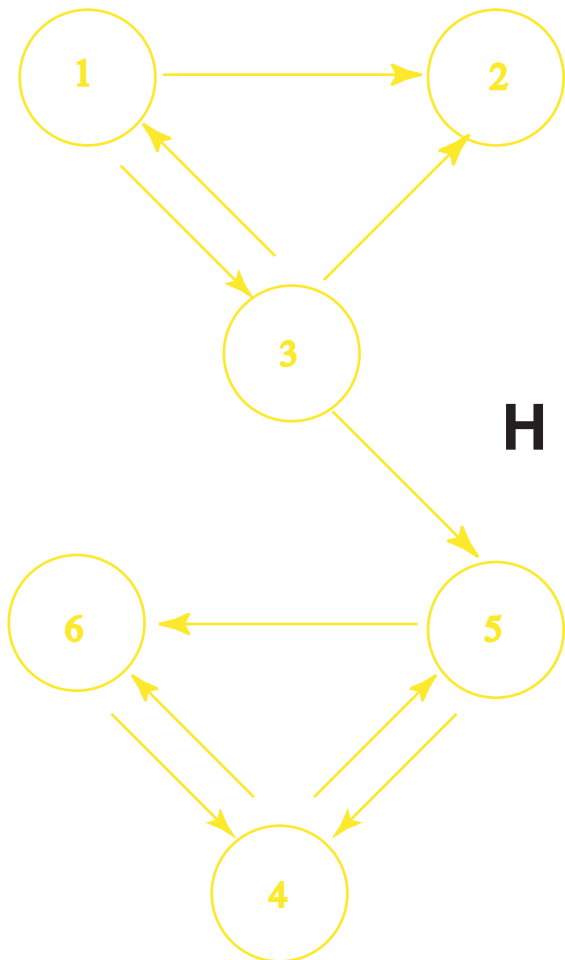


**H** =

$$\begin{matrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{matrix} \begin{pmatrix} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \\ 0 & 1/2 & 1/2 & 0 & 0 & 0 \end{pmatrix}$$



# Tiny Web

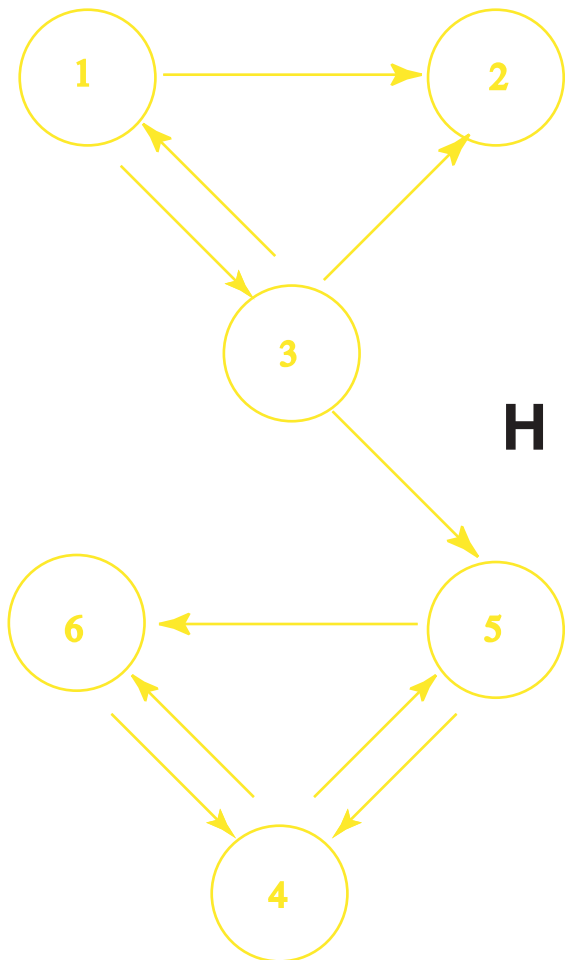


**H** =

$$\begin{matrix} & P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \\ \begin{matrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{matrix} & \left( \begin{array}{cccccc} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \end{array} \right) \end{matrix}$$



# Tiny Web

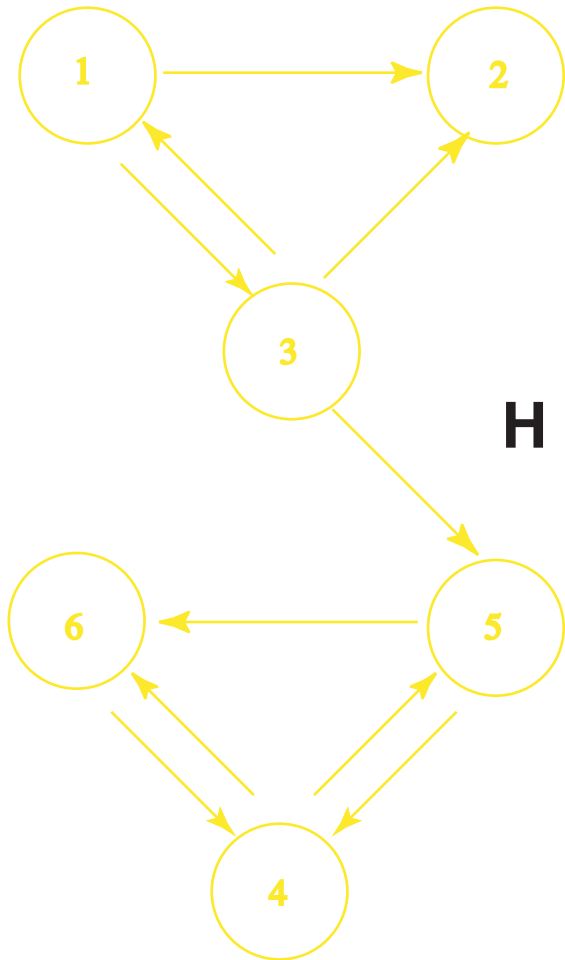


**H** =

$$\begin{matrix} & P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \\ \begin{matrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{matrix} & \left( \begin{array}{cccccc} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ & & & & & \\ & & & & & \\ & & & & & \end{array} \right) \end{matrix}$$



# Tiny Web

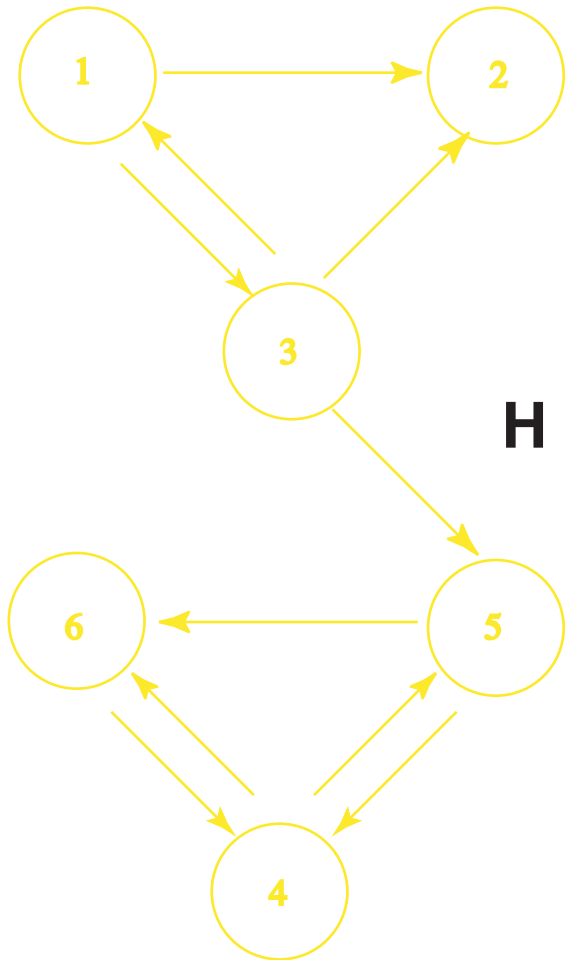


**H** =

$$\begin{matrix} & P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \\ \begin{matrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{matrix} & \left( \begin{array}{cccccc} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right) \end{matrix}$$



# Tiny Web

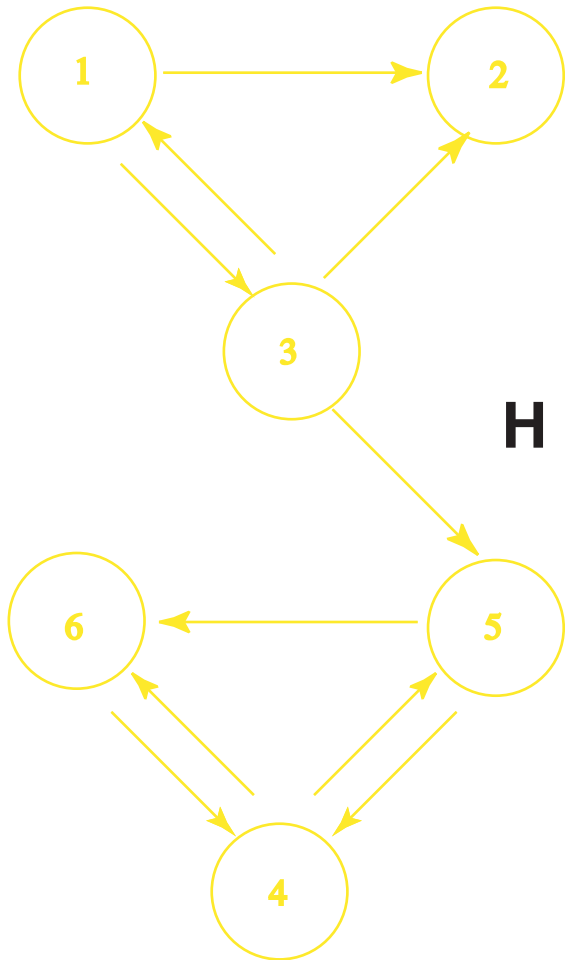


**H** =

$$\begin{matrix} & P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \\ \begin{matrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{matrix} & \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \end{pmatrix} \end{matrix}$$



# Tiny Web



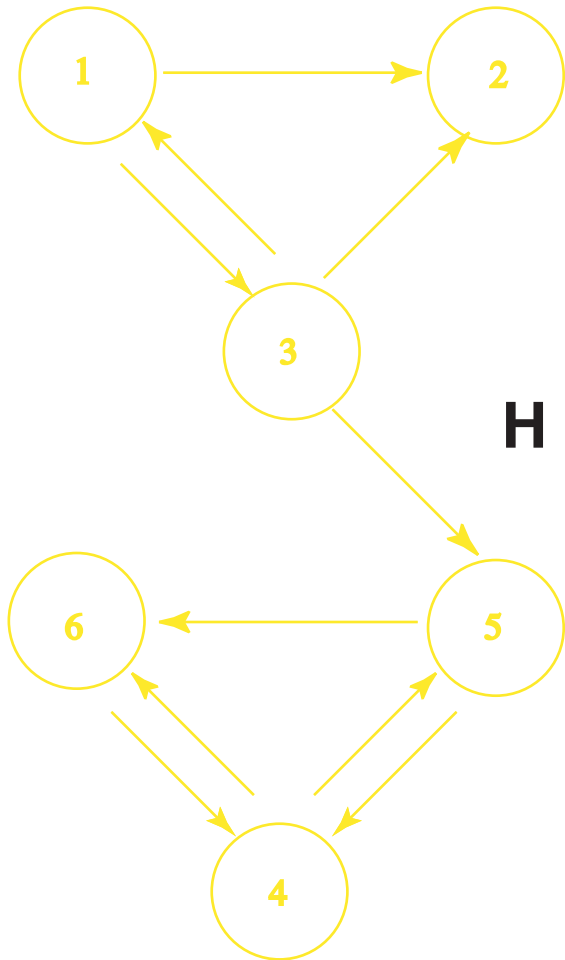
**H** =

$$\begin{matrix} & P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \\ \begin{matrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{matrix} & \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{matrix}$$





# Tiny Web

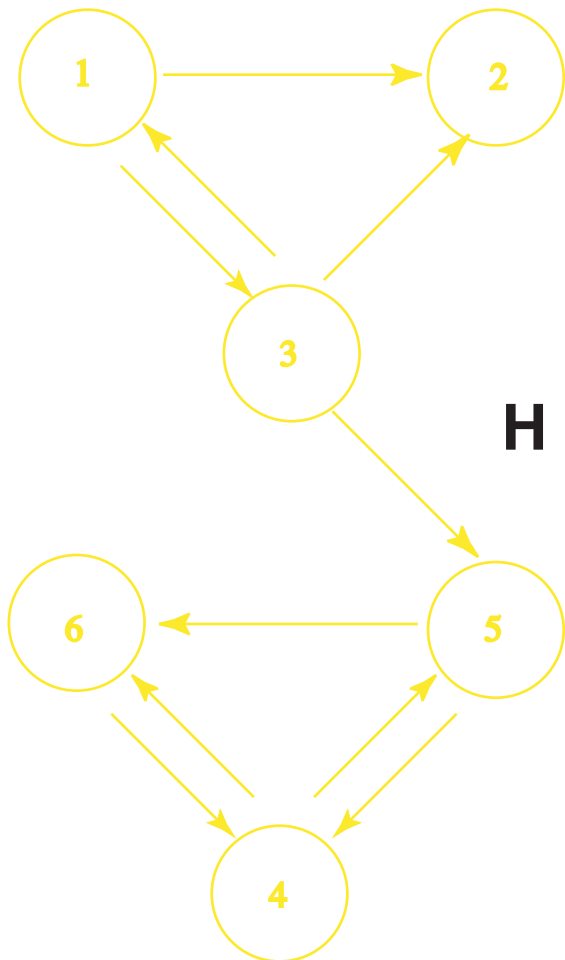


**H** =

$$\begin{matrix} & P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \\ \begin{matrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{matrix} & \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ \color{red}{0} & \color{red}{0} & \color{red}{0} & \color{red}{0} & \color{red}{0} & \color{red}{0} \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{matrix}$$

- Dead end page (nothing to click on) — a “dangling node”

# Tiny Web



**H** =

$$\begin{matrix} & P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \\ \begin{matrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{matrix} & \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ \color{red}{0} & \color{red}{0} & \color{red}{0} & \color{red}{0} & \color{red}{0} & \color{red}{0} \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{matrix}$$

- Dead end page (nothing to click on) — a “dangling node”

✓  $\pi^T$  not well defined



# The Fix

Replace zero rows with  $(1/n)\mathbf{e}^T = (1/n, 1/n, \dots, 1/n)$

$$\mathbf{S} = \begin{matrix} & \begin{matrix} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \end{matrix} \\ \begin{matrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{matrix} & \left( \begin{array}{cccccc} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ \mathbf{1/6} & \mathbf{1/6} & \mathbf{1/6} & \mathbf{1/6} & \mathbf{1/6} & \mathbf{1/6} \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{array} \right) \end{matrix}$$



# The Fix

Replace zero rows with  $(1/n)\mathbf{e}^T = (1/n, 1/n, \dots, 1/n)$

$$\mathbf{S} = \begin{matrix} & \begin{matrix} P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \end{matrix} \\ \begin{matrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{matrix} & \left( \begin{array}{cccccc} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ \mathbf{1/6} & \mathbf{1/6} & \mathbf{1/6} & \mathbf{1/6} & \mathbf{1/6} & \mathbf{1/6} \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{array} \right) \end{matrix}$$

$$\mathbf{S} = \mathbf{H} + \frac{\mathbf{a}\mathbf{e}^T}{6} = \mathbf{H} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \frac{1}{6} (1 \ 1 \ 1 \ 1 \ 1 \ 1)$$



# Another Problem

S is reducible

$$\mathbf{S} = \begin{array}{c|ccc} & P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \\ \hline P_1 & 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ P_2 & \mathbf{1/6} & \mathbf{1/6} & \mathbf{1/6} & \mathbf{1/6} & \mathbf{1/6} & \mathbf{1/6} \\ P_3 & 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ \hline P_4 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ P_5 & 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ P_6 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}$$

✓  $\pi^T$  may not be well defined



# Yet More Problems

Could get trapped into a cycle  $(P_i \rightarrow P_j \rightarrow P_i)$



# Yet More Problems

Could get trapped into a cycle  $(P_i \rightarrow P_j \rightarrow P_i)$

✓  $\pi_{j+1}^T = \pi_j^T \mathbf{P}$  won't convergence



# Yet More Problems

Could get trapped into a cycle  $(P_i \rightarrow P_j \rightarrow P_i)$

✓  $\pi_{j+1}^T = \pi_j^T \mathbf{P}$  won't convergence

## Convergence Requirement

Markov chain must be irreducible and aperiodic





# Yet More Problems

Could get trapped into a cycle  $(P_i \rightarrow P_j \rightarrow P_i)$

✓  $\pi_{j+1}^T = \pi_j^T \mathbf{P}$  won't convergence

## Convergence Requirement

Markov chain must be irreducible and aperiodic

- This means  $\mathbf{P}$  must be a primitive matrix

✓ No eigenvalues other than  $\lambda = 1$  on unit circle

# Yet More Problems

Could get trapped into a cycle  $(P_i \rightarrow P_j \rightarrow P_i)$

✓  $\pi_{j+1}^T = \pi_j^T \mathbf{P}$  won't convergence

## Convergence Requirement

Markov chain must be irreducible and aperiodic

- This means  $\mathbf{P}$  must be a primitive matrix

✓ No eigenvalues other than  $\lambda = 1$  on unit circle

✓  $\mathbf{P}^k > 0$  for some  $k$



# Yet More Problems

Could get trapped into a cycle  $(P_i \rightarrow P_j \rightarrow P_i)$

✓  $\pi_{j+1}^T = \pi_j^T \mathbf{P}$  won't convergence

## Convergence Requirement

Markov chain must be irreducible and aperiodic

- This means  $\mathbf{P}$  must be a primitive matrix

✓ No eigenvalues other than  $\lambda = 1$  on unit circle

✓  $\mathbf{P}^k > 0$  for some  $k$

## The Google Fixes

- $\mathbf{P} = \alpha \mathbf{S} + (1 - \alpha) \mathbf{e} \mathbf{e}^T / n$        $\alpha \approx .85$



# Yet More Problems

Could get trapped into a cycle  $(P_i \rightarrow P_j \rightarrow P_i)$

✓  $\pi_{j+1}^T = \pi_j^T \mathbf{P}$  won't convergence

## Convergence Requirement

Markov chain must be irreducible and aperiodic

- This means  $\mathbf{P}$  must be a primitive matrix

✓ No eigenvalues other than  $\lambda = 1$  on unit circle

✓  $\mathbf{P}^k > 0$  for some  $k$

## The Google Fixes

- $\mathbf{P} = \alpha \mathbf{S} + (1 - \alpha) \mathbf{e} \mathbf{e}^T / n$        $\alpha \approx .85$

- $\mathbf{P} = \alpha \mathbf{S} + (1 - \alpha) \mathbf{e} \mathbf{v}^T$        $\mathbf{v}^T = \text{positive probability vector}$



# Yet More Problems

Could get trapped into a cycle  $(P_i \rightarrow P_j \rightarrow P_i)$

✓  $\pi_{j+1}^T = \pi_j^T \mathbf{P}$  won't convergence

## Convergence Requirement

Markov chain must be irreducible and aperiodic

- This means  $\mathbf{P}$  must be a primitive matrix

✓ No eigenvalues other than  $\lambda = 1$  on unit circle

✓  $\mathbf{P}^k > 0$  for some  $k$

## The Google Fixes

- $\mathbf{P} = \alpha \mathbf{S} + (1 - \alpha) \mathbf{e} \mathbf{e}^T / n$        $\alpha \approx .85$

- $\mathbf{P} = \alpha \mathbf{S} + (1 - \alpha) \mathbf{e} \mathbf{v}^T$        $\mathbf{v}^T =$  positive probability vector

- $\mathbf{P} = \alpha \mathbf{H} + (\alpha \mathbf{a} + (1 - \alpha) \mathbf{e}) \mathbf{v}^T$



# THE WALL STREET JOURNAL

© 2003 Dow Jones & Company. All Rights Reserved

WEDNESDAY, FEBRUARY 26, 2003 - VOL. CCXLI NO. 39 - ★★★ \$1.00

WSJ.com

## What's News—

### Business and Finance

### World-Wide

**NEWS CORP.** and Liberty are no longer working together on a joint offer to take control of Hughes, with News Corp. proceeding on its own and Liberty considering an independent bid. The move threatens to cloud the process of finding a new owner for the GM unit.

(Article on Page A3)

■ **The SEC signaled it may file civil charges against Morgan Stanley,** alleging it doled out IPO shares based partly on investors' commitments to buy more stock.

(Article on Page C1)

■ **Ahold's problems deepened** as U.S. authorities opened inquiries into accounting at the Dutch company's U.S. Foodservice unit.

■ **Fleming said the SEC up-graded** to a formal investigation an inquiry into the food wholesaler's trade practices with suppliers.

(Articles on Page A2)

■ **Consumer confidence fell** to its lowest level since 1993, hurt by energy costs, the terrorism threat and a stagnant job market.

(Article on Page A3)

■ **The industrials rebounded** on

■ **BUSH IS PREPARING** to present Congress a huge bill for Iraq costs.

The total could run to \$95 billion depending on the length of the possible war and occupation. As horse-trading began at the U.N. to win support for a war resolution, the president again made clear he intends to act with or without the world body's imprimatur. Arms inspectors said Baghdad provided new data, including a report of a possible biological bomb. Gen. Franks assumed command of the war-operations center in Qatar. Allied warplanes are aggressively taking out missile sites that could threaten the allied troop buildup. (Column 4 and Pages A4 and A6)

Turkey's parliament debated legislation to let the U.S. deploy 62,000 to open a northern front. Kurdish soldiers lined roads in a show of force as U.S. officials traveled into Iraq's north for an opposition conference.

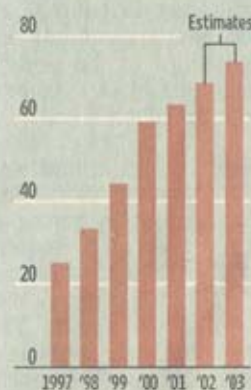
■ **Powell said North Korea** hasn't restarted a reactor and plutonium-processing facility at Yongbyon, hinting such forbearance might constitute an overture. But saber rattling continued a day after a missile test timed for the inauguration in Seoul. Pyongyang accused U.S. spy planes of violating its airspace and told its army to prepare for U.S. attack. (Page A14)

■ **The FBI came under** withering bipartisan criticism in a Senate Judiciary report in which Sen. Specter

## Web Master

### As the Web spreads...

Total Internet users, by household, in millions



Sources: Forrester Research; Nielsen NetRatings

### Google's U.S. presence expands

Top search engines, in millions of unique visitors<sup>1</sup>



<sup>1</sup>Including visitors from home and work, in January 2003

Top shopping-referral sites, in millions of referrals<sup>2</sup>



<sup>2</sup>Number of people the sites send to major online stores, including only visitors from home, for Q4 2002

## Bush to Seek up to \$95 Billion To Cover Costs of War on Iraq

By GREG JAFFE  
And JOHN D. MCKINNON

WASHINGTON—The Bush administration is preparing supplemental spending requests totaling as much as \$95 billion for a war with Iraq, its aftermath and new expenses to fight terrorism, officials said.

The total could be as low as \$60 billion because Pentagon budget planners don't know how long a military conflict will last, whether U.S. allies will contribute more than token sums to the effort and what damage Saddam Hussein might do

to his own country to retaliate against conquering forces.

Budget planners also are awaiting the outcome of an intense internal debate over whether to include \$13 billion in the requests to Congress that the Pentagon says it needs to fund the broader war on terrorism, as well as for stepped up homeland security. The White House Office of Management and Budget argues that the money might not be necessary. President Bush, Defense Secretary Donald Rumsfeld and budget director Mitchell Daniels Jr. met yesterday to discuss the matter but didn't reach a final agreement. Mr. Rumsfeld plans to continue pressing his

## Cat and Mouse

### As Google Becomes Web's Gatekeeper, Sites Fight to Get In

Search Engine Punishes Firms That Try to Game System; Outlawing the 'Link Farms'

Exoticleatherwear Gets Cut Off

By MICHAEL TOTTY  
And MYLENE MANGALINDAN

Joy Holman sells provocative leather clothing on the Web. She wants what nearly everyone doing business online wants: more exposure on Google.

So from the time she launched exoticleatherwear.com last May, she tried all sorts of tricks to get her site to show up among the first listings when a user of Google Inc.'s popular search engine typed in "women's leatherwear" or "leather apparel." She buried hidden words in her Web pages intended to fool Google's computers. She signed up with a service that promised to have hundreds of sites link to her online store—thereby boosting a crucial measure in Google's system of ranking sites.

The techniques worked—for a





# Web Sites Fight for Prime Real Estate on Google

*Continued From First Page*

advertising that tried to capitalize on Google's formula for ranking sites. In effect, SearchKing was offering its clients a chance to boost their own Google rankings by buying ads on more-popular sites. SearchKing filed suit against the search company in federal court in Oklahoma, claiming that Google "purposefully devalued" SearchKing and its customers, damaging its reputation and hurting its advertising sales.

Google won't comment on the case. In court filings, the company said SearchKing "engaged in behavior that would lower the quality of Google search results" and alter the company's ranking system.

Google, a closely held company founded by Stanford University graduate students Sergey Brin and Larry Page, says Web companies that want to rank high should concentrate on improving their Web pages rather than gaming its system. "When people try to take scoring into their own hands, that turns into a worse experience for users," says Matt Cutts, a Google software engineer.

## Coding Trickery

Efforts to outfox the search engines have been around since search engines first became popular in the early 1990s. Early tricks included stuffing thousands of widely used search terms in hidden coding, called "metatags." The coding fools a search engine into identifying a site with popular words and phrases that may not actually appear on the site.

Another gimmick was hiding words or terms against a same-color background. The hidden coding deceived search engines that relied heavily on the number of times a word or phrase appeared in ranking a site. But Google's system, based on links, wasn't fooled.

Mr. Brin, 29, one of Google's two founders and now its president of technology, boasted to a San Francisco search-engine conference in 2000 that Google wasn't worried about having its results clogged with irrelevant results because its search methods couldn't be manipulated.

That didn't stop search optimizers from finding other ways to outfox the system. Attempts to manipulate Google's results even became a sport, called Google-baiting. Baitsters would try to

creating Web sites that were nothing more than collections of links to the clients' site, called "link farms." Since Google ranks a site largely by how many links or "votes" it gets, the link farms could boost a site's popularity.

In a similar technique, called a link exchange, a group of unrelated sites would agree to all link to each other, thereby fooling Google into thinking the sites have a multitude of votes. Many sites also found they could buy links to themselves to boost their rankings.

Ms. Holman, the leatherwear retailer, discovered the consequences of trying to fool Google. The 42-year-old hospital laboratory technician, who learned computer skills by troubleshooting her hospital's

**'The big search engines determine the laws of how commerce runs,' says Mr. Massa.**

equipment, operates her online apparel store as a side business that she hopes can someday replace her day job.

When she launched her Exotic Leather Wear store from her home in Mesa, Ariz., she quickly learned the importance of appearing near the top of search-engine results, especially on Google. She boned up on search techniques, visiting online discussion groups dedicated to search engines and reading what material she could find on the Web.

At first, Ms. Holman limited herself to modest changes, such as loading her page with hidden metatag coding that would help steer a search toward her site when a user entered words such as "haltertops" or "leather miniskirts." Since Google doesn't give much weight to metatags in determining its rankings, the efforts had little effect on her search results.

She then received an e-mail advertisement from AutomatedLinks.com, a Wirral, England, company that promised to send traffic "through the roof" by linking more than 2,000 Web sites to hers. Aside from attracting customers, the links were designed to improve her site's search engine rankings by taking

In theory, when Google encounters the AutomatedLinks code, it treats it as a legitimate referral to the other sites and counts them in totting up the sites' popularity.

Shortly after Ms. Holman signed up with AutomatedLinks in July, she read on an online discussion group that Google objected to such link arrangements. She says she immediately stripped the code from her Web pages. For a while her site gradually worked its way up in Google search results, and business steadily improved because links to her site still remained on the sites of other AutomatedLinks customers. Then, sometime in November, her site was suddenly no longer appearing among the top results. Her orders plunged as much as 80%.

Ms. Holman, who e-mailed Google and AutomatedLinks, says she has been unable to get answers. But in the last few months, other AutomatedLinks customers say they have seen their sites apparently penalized by Google. Graham McLeay, who runs a small chauffeur service north of London, saw revenue cut in half during the two months he believes his site was penalized by Google.

The high-stakes fight between Google and the optimizers can leave some Web-site owners confused. "I don't know how people are supposed to judge what is right and wrong," says Mr. McLeay.

AutomatedLinks didn't respond to requests for comment. Google declined to comment on the case. But Mr. Cutts, the Google engineer, warns that the rules are clear and that it's better to follow them rather than try to get a problem fixed after a site has been penalized. "We want to return the most relevant pages we can," Mr. Cutts says. "The best way for a site owner to do that is follow our guidelines."

## Crackdown

Google has been stepping up its enforcement since 2001. It warned Webmasters that using trickery could get their sites kicked out of the Google index and it provided a list of forbidden activities, including hiding text and "link schemes," such as the link farms. Google also warned against "cloaking"—showing a search engine a page that's designed to score well while giving visitors a different, more attractive page—or creating multiple Web addresses that take visitors to a single site.

To stay one step ahead of the Web

homa City-based SearchKing, an online directory for hundreds of small, specialty Web sites. SearchKing also sells advertising links designed both to deliver traffic to an advertiser and boost its rankings in Google and other search results.

Bob Massa, SearchKing's chief executive, last August launched the PR Ad Network as a way to capitalize on Google's page-ranking system, known as PageRank. PageRank rates Web sites on a scale of one to 10 based on their popularity, and the rankings can be viewed by Web users if they install special Google software. PR Ad Network sells ads that are priced according to a site's PageRank, with higher-ranked sites commanding higher prices. When a site buys an advertising link on a highly ranked site, the ad buyer could see its ratings improve because of the greater weight Google gives to that link.

Shortly after publicizing the ad network, Mr. Massa discovered that his site suddenly dropped in Google's rankings. What's more, sites that participated in the separate SearchKing directory also had their Google rankings lowered. He filed a lawsuit in Oklahoma City federal court, claiming Google was punishing him for trying to profit from the company's page-ranking system.

A Google spokesman won't comment on the case. In its court filings, Google said it demoted pages on the SearchKing site because of SearchKing's attempts to manipulate search results. The company has asked for the suit to be dismissed, arguing that the PageRank represents its opinion of the value of a Web site and as such is protected by the First Amendment.

"The big search engines determine the laws of how commerce runs," says Mr. Massa, who is persisting with the lawsuit even though the sites have had their page rankings partly restored. "Someone needs to demand accountability."

Google is taking steps that many say could satisfy businesses trying to boost their rankings. Google has long sold sponsored links that show up on the top of many search-results pages, separate from the main listings. Last year, the company expanded its paid-listings program, so that there are now more slots where sites can pay for a prominent place in the results. Many sites now are turning to advertising instead of tactics to optimize their rankings.

# Home Depot Amid First

By CHAD TERHUN

ATLANTA—Home Depot Inc. fiscal fourth-quarter earnings fell 3.4% on disappointing sales.

Speaking to investors and analysts, the company's chief executive, Bob Nardelli, said Home Depot is prepared to meet dissatisfied customers and competitive challenge from Home Depot's remodelers, inventory and improved customer service.

The nation's largest home improvement retailer said net income for the quarter ended Feb. 2 decreased to 30 cents a share, from \$71.30 cents a share, a year earlier. Sales rose 2% to \$13.21 billion from \$13.4 billion a year earlier. Home Depot's first quarterly sales decline in its 24-year history. Home Depot's latest quarter was a week earlier. Using comparable periods, the company said quarterly sales increased 5% and net income rose 5%.

Same-store sales, or sales from stores open at least a year, declined 1% in the quarter. Home Depot said the decline last month offset a disastrous quarter and helped the retailer avoid a stock price decline. Home Depot estimated that same-store sales rose as much as 10%. In 4 p.m. trading on the New York Stock Exchange composite trading, Home Depot shares rose 66 cents to \$54.38.

# Fiat Patria Is Set to Be

By ALESSANDRA GAIARDI

ROME—Umberto Agnelli, the 68-year-old Fiat patriarch Gianni Agnelli's son, is set to be named Fiat SpA chairman on Feb. 1, replacing his father. Agnelli is expected to resign last month, was widely expected to be replaced by his son, Umberto Agnelli, later this year. But his son has served as chairman of Fiat's





# Back To Tiny Web

## The Google Matrix

$$\mathbf{P} = \alpha \mathbf{H} + (\alpha \mathbf{a} + (1 - \alpha) \mathbf{e}) \mathbf{v}^T \quad (\text{with } \alpha = .9 \text{ and } \mathbf{v} = \mathbf{e})$$

$$= \begin{bmatrix} 1/60 & 7/15 & 7/15 & 1/60 & 1/60 & 1/60 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 19/60 & 19/60 & 1/60 & 1/60 & 19/60 & 1/60 \\ 1/60 & 1/60 & 1/60 & 1/60 & 7/15 & 7/15 \\ 1/60 & 1/60 & 1/60 & 7/15 & 1/60 & 7/15 \\ 1/60 & 1/60 & 1/60 & 11/12 & 1/60 & 1/60 \end{bmatrix}$$

## The PageRank Vector

$$\pi_{j+1}^T = \pi_j^T \mathbf{P} \rightarrow \pi^T$$

$$\pi^T = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ .03721 & .05396 & .04151 & .3751 & .206 & .2862 \end{pmatrix}$$





# Computing $\pi^T$

## A Big Problem

$$\text{Solve } \pi^T = \pi^T \mathbf{P}$$

(eigenvector problem)



# Computing $\pi^T$

## A Big Problem

$$\text{Solve } \pi^T = \pi^T \mathbf{P}$$

(eigenvector problem)

$$\pi^T (\mathbf{I} - \mathbf{P}) = 0$$

(too big for direct solves)

Google's PageRank is an eigenvector of a matrix of order 2.7 billion.

One of the reasons why Google is such an effective search engine is the PageRank™ algorithm, developed by Google's founders, Larry Page and Sergey Brin, when they were graduate students at Stanford University. PageRank is determined entirely by the link structure of the Web. It is recomputed about once a month and does not involve any of the actual content of Web pages or of any individual query. Then, for any particular query, Google finds the pages on the Web that match that query and lists those pages in the order of their PageRank.

Imagine surfing the Web, going from page to page by randomly choosing an outgoing link from one page to get to the next. This can lead to dead ends at pages with no outgoing links, or cycles around cliques of interconnected pages. So, a certain fraction of the time, simply choose a random page from anywhere on the Web. This theoretical random walk of the Web is a *Markov chain* or *Markov process*. The limiting probability that a dedicated random surfer visits any particular page is its PageRank. A page has high rank if it has links to and from other pages with high rank.

Let  $W$  be the set of Web pages that can be reached by following a chain of hyperlinks starting from a page at Google and let  $n$  be the number of pages in  $W$ . The set  $W$  actually varies with time, but in May 2002,  $n$  was about 2.7 billion. Let  $G$  be the  $n$ -by- $n$  connectivity matrix of  $W$  that is,  $G_{ij}$  is 1 if there is a hyperlink from page  $i$  to page  $j$  and 0

BY CLEVE MOLER

It tells us that the largest eigenvalue of  $A$  is equal to one and that the corresponding eigenvector, which satisfies the equation

$$x = Ax,$$

exists and is unique to within a scaling factor. When this scaling factor is chosen so that

$$\sum_i x_i = 1$$

then  $x$  is the state vector of the Markov chain. The elements of  $x$  are Google's PageRank.

If the matrix were small enough to fit in MATLAB, one way to compute the eigenvector  $x$  would be to start with a good approximate solution, such as the PageRanks from the previous month, and simply repeat the assignment statement

$$x = Ax$$

until successive vectors agree to within specified tolerance. This is known as the power method and is about the only possible approach for very large  $n$ . I'm not sure how Google actually computes PageRank, but one step of the power method would require one pass over a database of Web pages, updating weighted reference counts generated by the hyperlinks between pages.



# Computing $\pi^T$

## A Big Problem

Solve  $\pi^T = \pi^T \mathbf{P}$  (eigenvector problem)

$\pi^T (\mathbf{I} - \mathbf{P}) = 0$  (too big for direct solves)

Start with  $\pi_0^T = \mathbf{e}/n$  and iterate  $\pi_{j+1}^T = \pi_j^T \mathbf{P}$  (power method)



# Computing $\pi^T$

## A Big Problem

Solve  $\pi^T = \pi^T \mathbf{P}$  (eigenvector problem)

$\pi^T (\mathbf{I} - \mathbf{P}) = 0$  (too big for direct solves)

Start with  $\pi_0^T = \mathbf{e}/n$  and iterate  $\pi_{j+1}^T = \pi_j^T \mathbf{P}$  (power method)

## Convergence Time

Measured in days



# Computing $\pi^T$

## A Big Problem

Solve  $\pi^T = \pi^T \mathbf{P}$  (eigenvector problem)

$\pi^T (\mathbf{I} - \mathbf{P}) = 0$  (too big for direct solves)

Start with  $\pi_0^T = \mathbf{e}/n$  and iterate  $\pi_{j+1}^T = \pi_j^T \mathbf{P}$  (power method)

## Convergence Time

Measured in days

## A Bigger Problem — Updating

Pages & links are added, deleted, changed continuously



# Computing $\pi^T$

## A Big Problem

Solve  $\pi^T = \pi^T \mathbf{P}$  (eigenvector problem)

$\pi^T (\mathbf{I} - \mathbf{P}) = 0$  (too big for direct solves)

Start with  $\pi_0^T = \mathbf{e}/n$  and iterate  $\pi_{j+1}^T = \pi_j^T \mathbf{P}$  (power method)

## Convergence Time

Measured in days

## A Bigger Problem — Updating

Pages & links are added, deleted, changed continuously

Google says just start from scratch every 3 to 4 weeks



# Computing $\pi^T$

## A Big Problem

Solve  $\pi^T = \pi^T \mathbf{P}$  (eigenvector problem)

$\pi^T (\mathbf{I} - \mathbf{P}) = 0$  (too big for direct solves)

Start with  $\pi_0^T = \mathbf{e}/n$  and iterate  $\pi_{j+1}^T = \pi_j^T \mathbf{P}$  (power method)

## Convergence Time

Measured in days

## A Bigger Problem — Updating

Pages & links are added, deleted, changed continuously

Google says just start from scratch every 3 to 4 weeks

Prior results don't help to restart





# Conclusion



Google Now Uses Many Other “Metrics” to augment PR





# Conclusion

- ✦ Google Now Uses Many Other “Metrics” to augment PR ✦
- ✦ Elegant Blend of NA, LA, Graph Theory, Prob, & CS ✦



# Conclusion

- ✦ Google Now Uses Many Other “Metrics” to augment PR ✦
- ✦ Elegant Blend of NA, LA, Graph Theory, Prob, & CS ✦
- ✦ Search Is Opening New Areas Ripe For Innovative Ideas ✦

# Conclusion

- ✦ Google Now Uses Many Other “Metrics” to augment PR ✦
- ✦ Elegant Blend of NA, LA, Graph Theory, Prob, & CS ✦
- ✦ Search Is Opening New Areas Ripe For Innovative Ideas ✦
- ✦ Exciting Work That Is Changing The World ✦



# Conclusion

- ✦ Google Now Uses Many Other “Metrics” to augment PR ✦
- ✦ Elegant Blend of NA, LA, Graph Theory, Prob, & CS ✦
- ✦ Search Is Opening New Areas Ripe For Innovative Ideas ✦
- ✦ Exciting Work That Is Changing The World ✦
- ✦ **Thanks For Your Attention** ✦