Web Search Engines

- SI650 / EECS 549
  Information Retrieval

- September 25, 2024

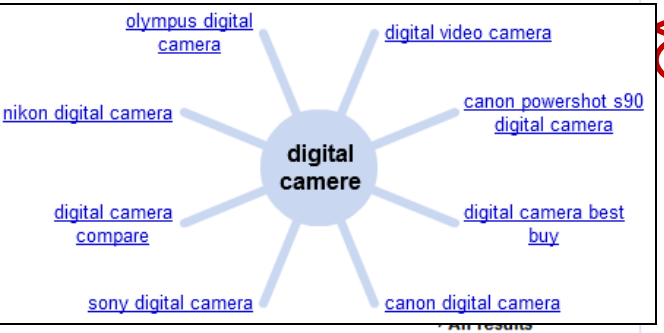
2023 © *University of Michigan*

Today's lecture plan

- Gain a better understanding of how web search works
- Learn how web search isn't just IR ranking—or is it?
- Rank using non-textual information

2023 © *University of Michigan*

Web Search is Hard



Google digital camere Search Advanced Search

Web Hide options Results 1 - 10 of about 76,700,000 for digital camere. (0.27 seconds)

› All results
Images
Videos
Nearby
› Standard view
Related searches
Wonder wheel
Timeline
› Standard results
Images from the page
Fewer shopping sites
More shopping sites
Page previews
Translated search

Did you mean: digital *camera*

Digital Camera Reviews and News: Digital Photography Review ...
Mar 12, 2010 ... Digital Photography Review: All the latest **digital camera** reviews and digital imaging news. Lively discussion forums.
Canon - Reviews - Nikon - Most popular cameras
www.dpreview.com/ - Cached - Similar

Camera Labs: **Digital Camera**, DSLR, and Lens Reviews. Recommended ...
Camera Labs: **Digital camera**, digital SLR and lens reviews, workshops, news.
www.**camera**labs.com/ - Cached - Similar

Unbiased **Digital Camera** Reviews and News | **Digital Camera** Resource ...
The **Digital Camera** Resource Page has been providing unbiased **digital camera** reviews, news, discussion forums, buyers guides, and frequently asked questions ...
www.dcresource.com/ Cached Similar

News results for **digital camere**
Panasonic's Lumix DMC-ZS5 **Digital Camera** Features Manual Controls ... - 14 hours ago
by PCWorld Videos | 0 Comments Recommends Panasonic's Lumix DMC-ZS5 is a 12.1 megapixel **digital camera** with an ultrawide lens that has excellent picture ...
PC World - 6 related articles »
Canon U.S.A. Joins Creative Forces with Fashion Brand LeSportsac - MarketWatch (press release) - 29 related articles »

Image results for **digital camere** - Report images

Video results for **digital camere**
Nikon Coolpix P90 **Digital Camera**
7 min - Apr 7, 2009
www.youtube.com
FUJIFILM FinePix S2000HD **Digital Camera**
1 min 46 sec - Oct 2, 2008
www.youtube.com

Sponsored Links
Digital Camere
Compare Deals on Electronics & More Low Prices On **Digital Camere**
shopping.yahoo.com
See your ad here »

2023 © University of Michigan

The Data-User-Service (DUS) Triangle

Lawyers
Scientists
Umich employees
Online shoppers
…

Users

Data

Web pages
News articles
Blog articles
Literature
Email
…

Search
Browsing
Mining
Task support, …

Services

- Slide from ChengXiang Zhai

2023 © *University of Michigan*

Challenges in Web Search

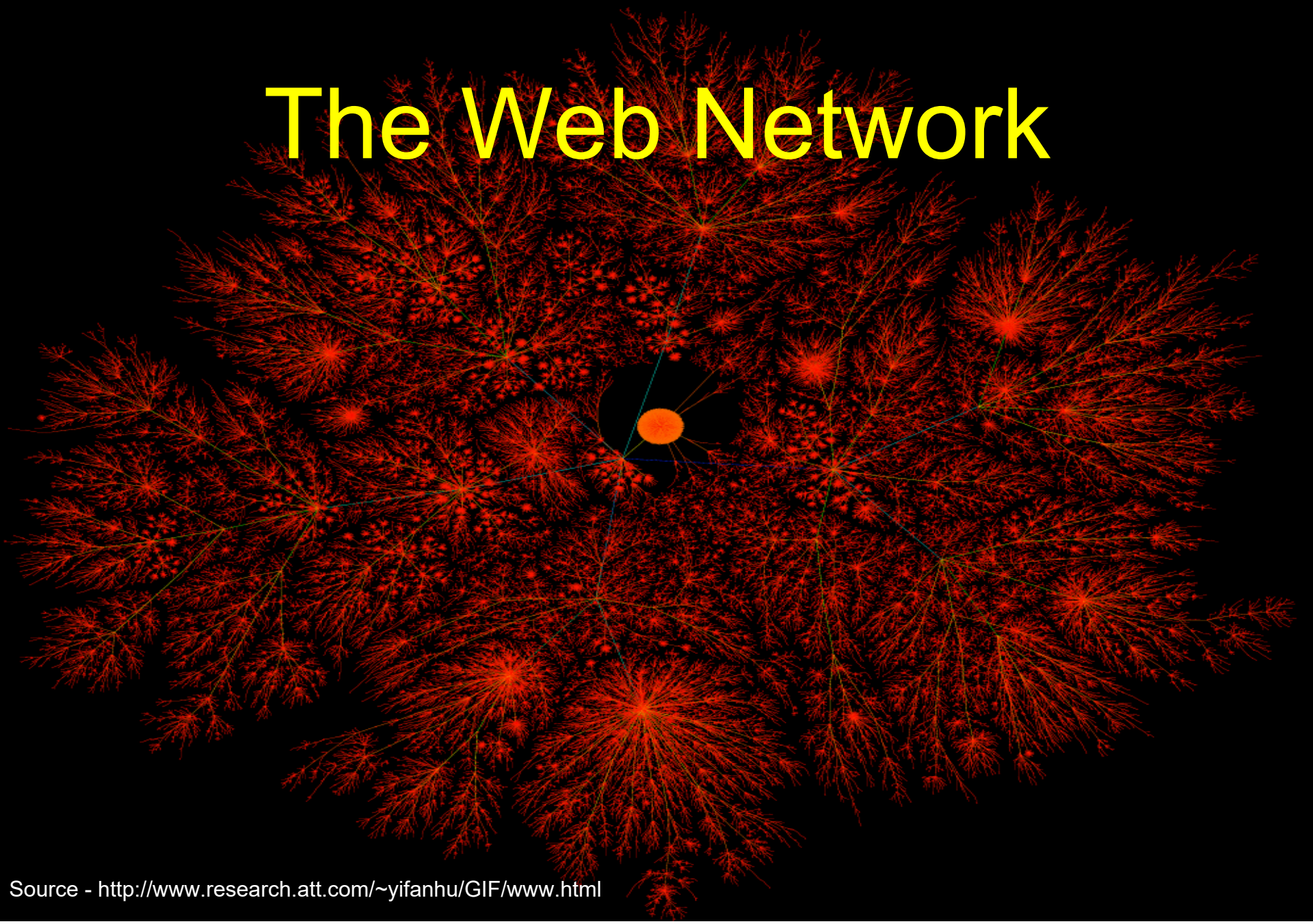
- Content relevance isn't the only objective
  - Quality, diversity, novelty, …
- Queries are vague, poorly formulated
- Relevance is a highly personalized notion
- The web is messy
  - 30% content on the web are near-duplicate
- Spam! Spam!! Spam!!!
- So, really quality! How do we get at quality?
  - …without looking at text!

2023 © *University of Michigan*

Search as a Network Problem

2023 © *University of Michigan*

The Web Network

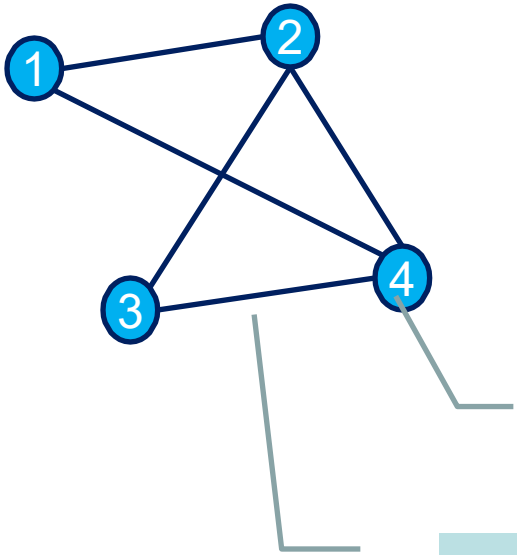


Source - http://www.research.att.com/~yifanhu/GIF/www.html

2023 © *University of Michigan*

Representation of Networks

- Networks are collections of points joined by lines.



Mathematically (Topologically):
"Network" ≡ "Graph"

$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

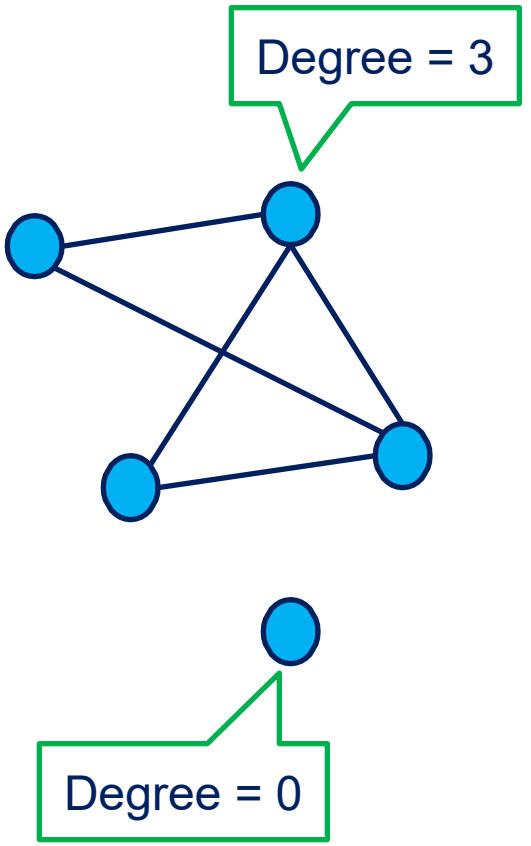
Vertex, node, site, actor, …

Edge, arc, link, bond, tie, relation …

But a real network is much more than a topological structure

2023 © University of Michigan

Network Measures: Characterize Networks



- Degree: how many friends do I have?
- Weights: how strong are the ties?
- Path: how far am I from another vertex?
- Connectivity: can I reach all other vertices?
- Diameter: how dense are they?
- Centrality (e.g., betweenness, closeness): Am I in the center of everyone?
- Density: are the vertices well connected?
- Clustering Coefficient: Are my neighbors well connected?

2023 © University of Michigan

The Web as a Network

- The web graph
  - degree distribution
  - clustering
  - motif profile
  - communities

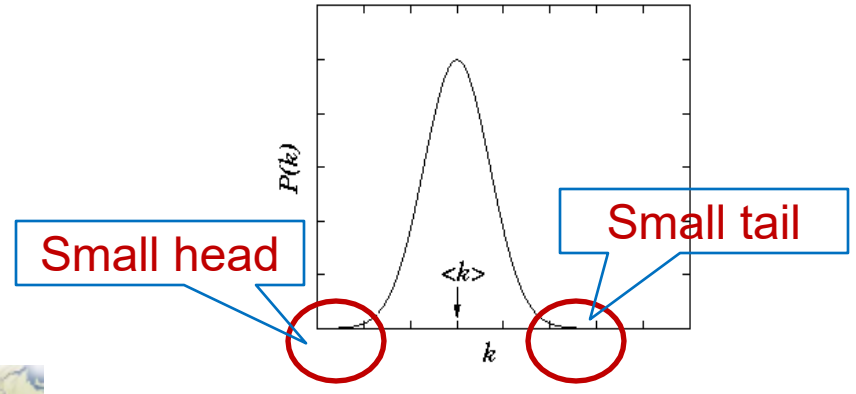
2023 © *University of Michigan*

Properties: Degree Distribution

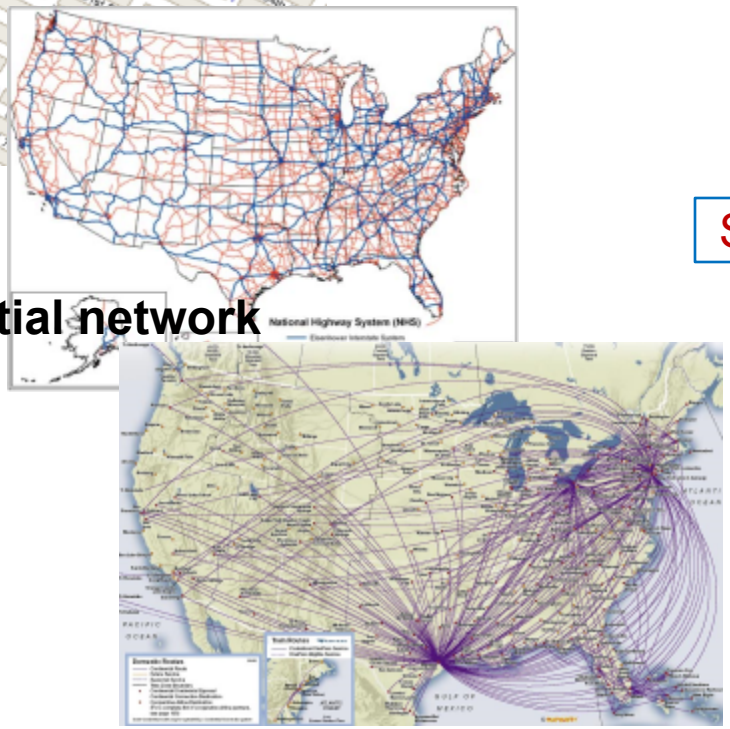
Lattice

All vertices have the same degree

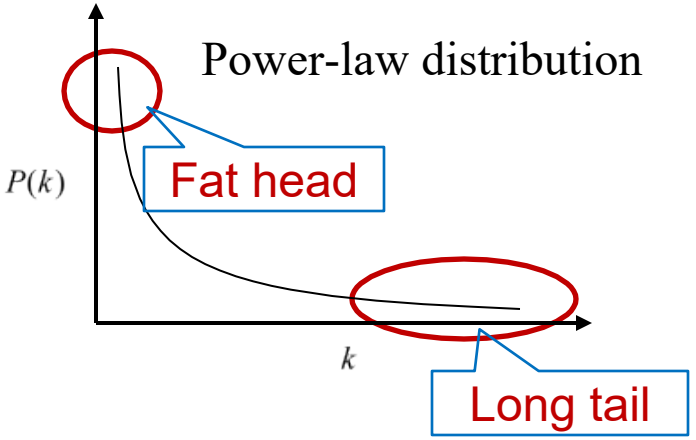
Poisson distribution



Exponential network



Power-law distribution



scale-free network

2023 © University of Michigan

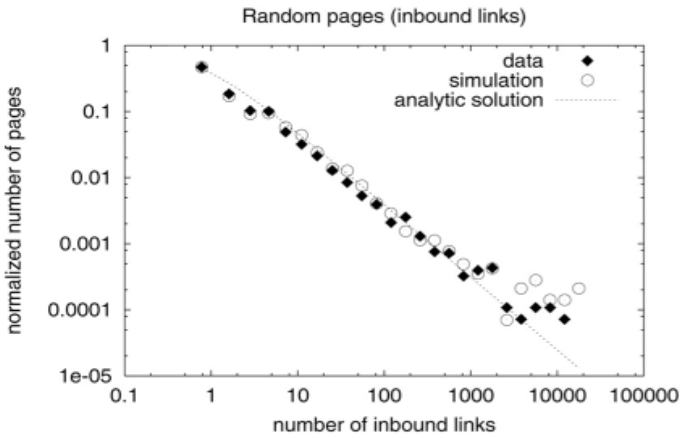
The Web is Scale Free

- The web is a scale free network
- Has power law degree distribution. The probability of observing an item of size (degree) 'x' is given by

$$p(x) = Cx^{-\alpha}$$

normalization constant (probabilities over all *x* must sum to 1)

α : scaling exponent, or power law exponent



indegree, a ~ 2.1
outdegree, a ~ 2.4
source: Pennock et al.: Winners don't take all: Characterizing the competition for links on the web
PNAS April 16, 2002 vol. 99 no. 8 5207-5211

- Straight line on a log-log plot

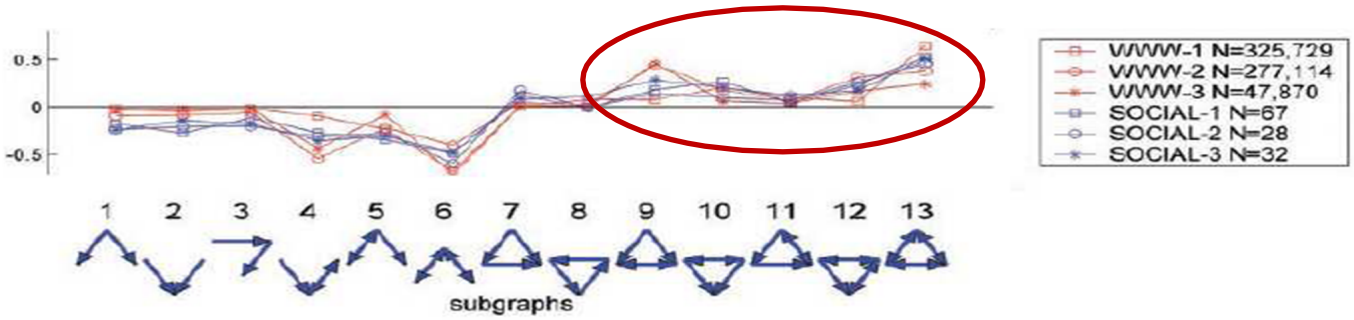
$$\ln(p(x)) = c - \alpha \ln(x)$$

度为 x的节点出现的概率随着 x的增大而快速下降，但仍然存在一些极少数度数特别大的节点。

2023 © University of Michigan

Clustering & Motifs

- Web network has high clustering compared to random networks
- Pages linked to from a given domain tend to link to each other as well
- Clustering coefficient ~ 0.11 (at the site level)



Source: Milo et al., "Superfamilies of evolved and designed networks", Science 303 (5663), p. 1538-1542, 2004.

2023 © *University of Michigan*

1. Clustering（聚类/聚集系数）

- **概念**：在网络中，某个节点的邻居之间是否也彼此相连。
  - 举例：如果网页 A 链接到网页 B 和 C，那么 **B 和 C 之间是否也互相链接**?
- **Web 的特点**：
  - 相比随机网络，Web 的聚类系数更高。
  - 例如：同一个域名下的网页，往往会互相链接（如某大学网站的不同院系页面）。
- **数据**：在站点级别，Web 的聚类系数大约为 **0.11**（Milo et al., Science 2004）
  SI650-Week-05-WebSearch 。

这说明：Web 网络里存在大量"小圈子"，局部紧密连接。

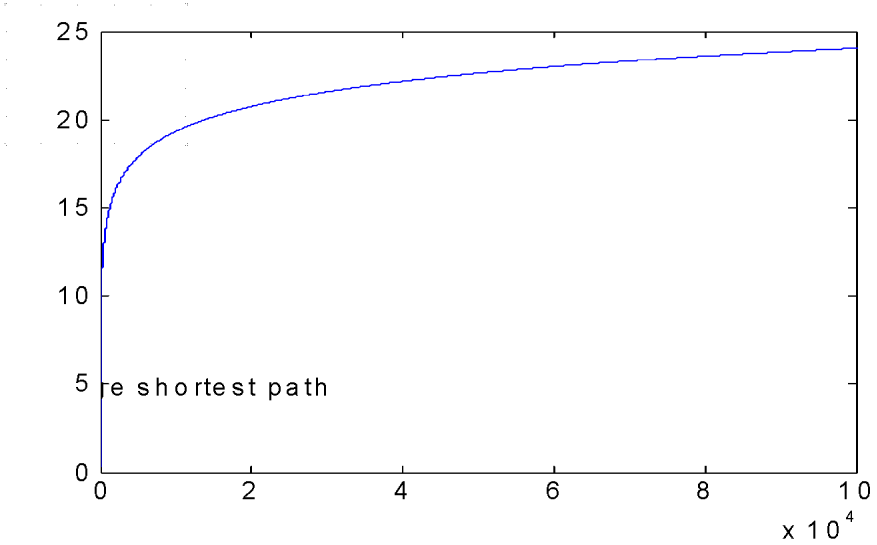
2. Motifs（结构基元/模式）

- **定义**：在网络中反复出现的小规模连接模式。
  - 例如：三角形结构（A→B, B→C, C→A），双向结构（A↔B），星型结构（一个节点指向多个节点）。
- **意义**：Motifs 被认为是网络的"功能模块"，不同类型的网络（如社交、Web、基因调控网络）都有其典型 motif 分布。
- **在 Web 上**：
  - 高聚类意味着许多 **三角形结构**（三个页面相互链接）。
  - 同一类网站（如新闻站、学术站点）可能有特定的 motif 特征。

2023 © *University of Michigan*

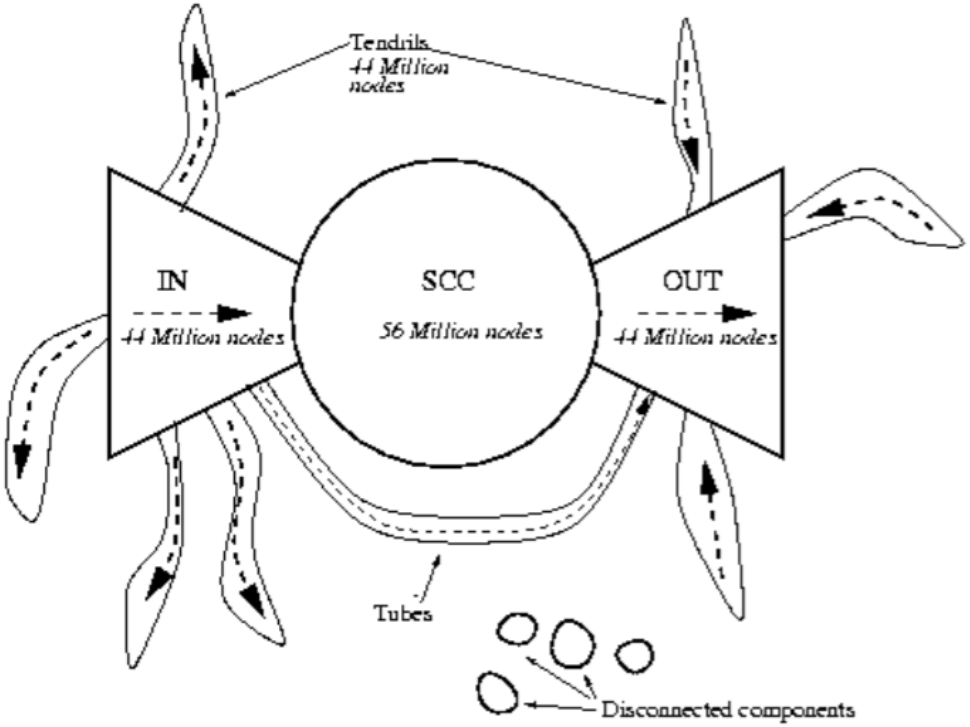
Shortest Paths

- On average web pages are within a small number of hops from each other
- <d> = 0.35 + 2.06 log(N)
- prediction: <d> = 17.5 for 200 million nodes
- actual: <d> = 16 for reachable pairs



2023 © University of Michigan

Components: the Bowtie Model



Outdated: from 1999 AltaVista data

2023 © *University of Michigan*

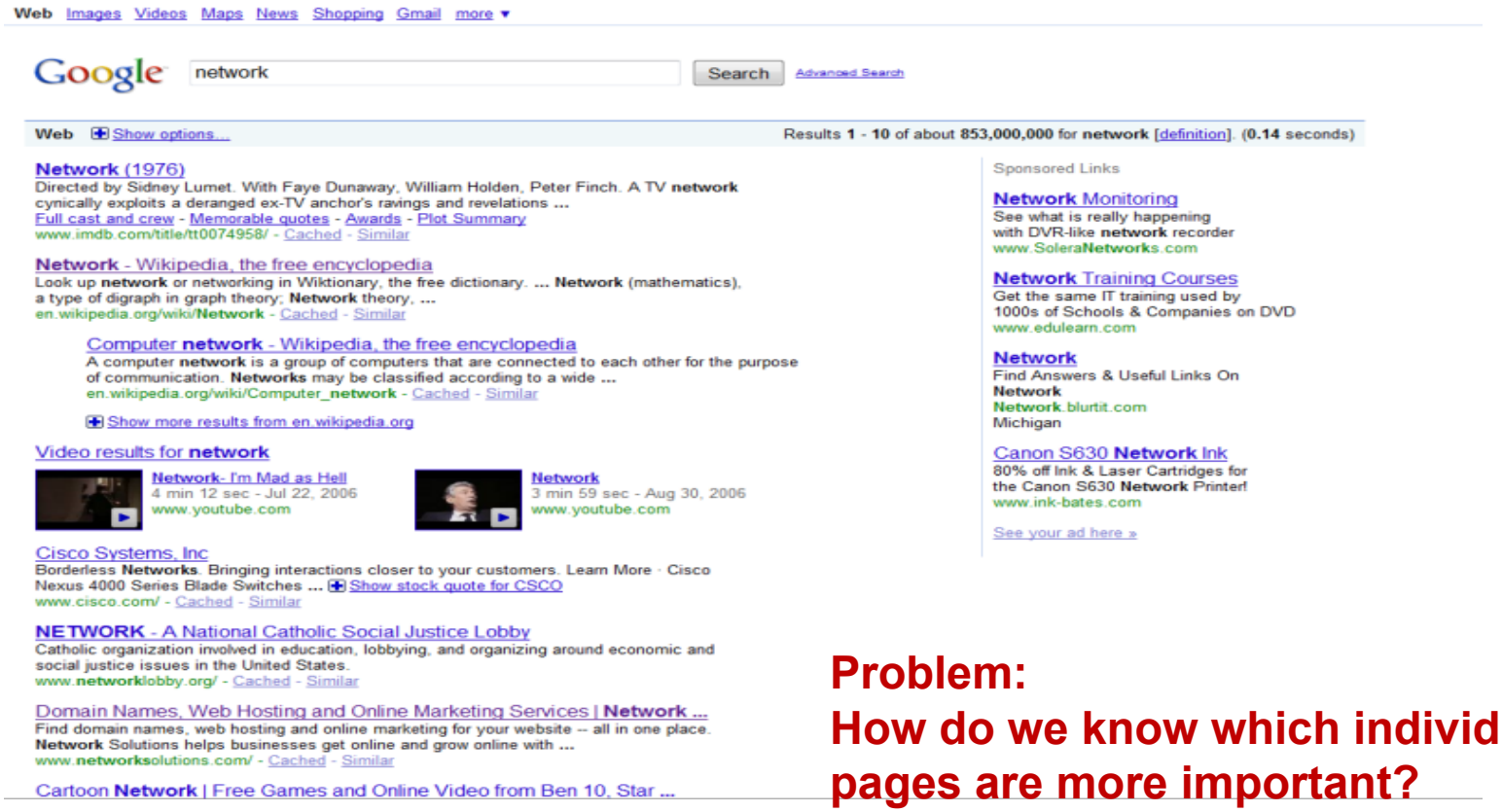
2. Bowtie 模型的主要组成部分

“蝴蝶结”之所以得名，是因为整个网络结构看起来像一个蝴蝶结，中间有核心，两边有入口和出口：

1. **SCC（Strongly Connected Component, 强连通核心）**
   - 位于中心。
   - 从其中任何一个网页出发，都可以沿超链接路径返回到另一个网页。
   - 这是 **Web 的“核心”区域**，大型网站和高度互链的页面通常在这里。
2. **IN 组件**
   - 能进入 SCC，但 **无法从 SCC 返回。**
   - 就像“入口”，比如一些只指向核心的网站，但核心不会再链接回它们。
3. **OUT 组件**
   - 可以从 SCC 到达，但 **无法返回到 SCC。**
   - 就像“出口”，例如一些终端网站（如个人主页、文件下载页）。
4. **Tendrils（触须）**
   - 连接在 IN 和 OUT 的边缘，但既不在 SCC，也不属于纯 IN 或 OUT。
   - 它们可能能连到 IN/OUT，但和核心没有直接联系。
5. **Disconnected Components（孤立部分）**
   - 完全与主结构不连通的网页或小型网络。

2023 © *University of Michigan*

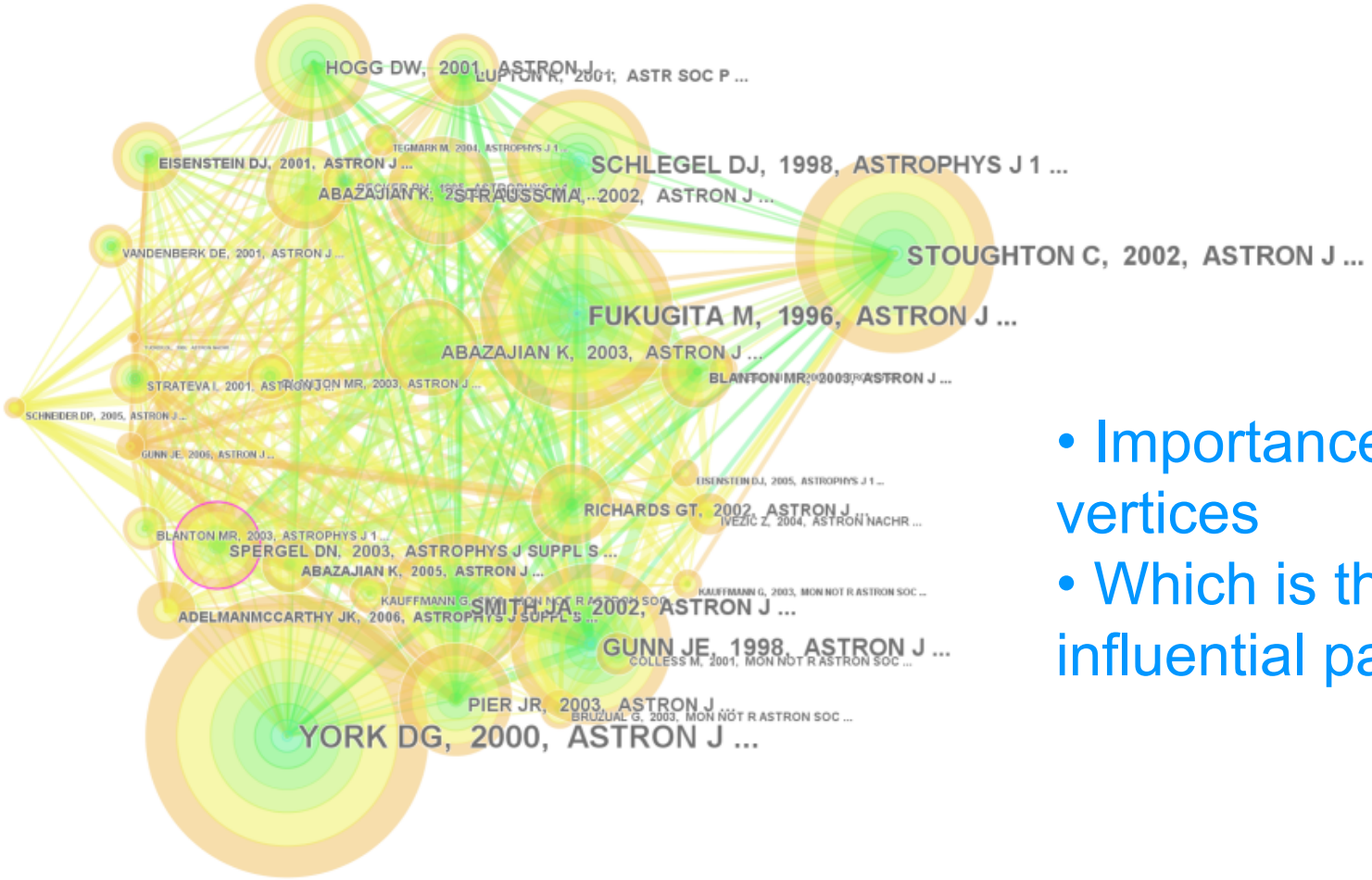
That was the Web Graph Overall



Problem:
How do we know which individual pages are more important?

2023 © *University of Michigan*

Ranking in Networks



- Importance of vertices
- Which is the most influential paper?

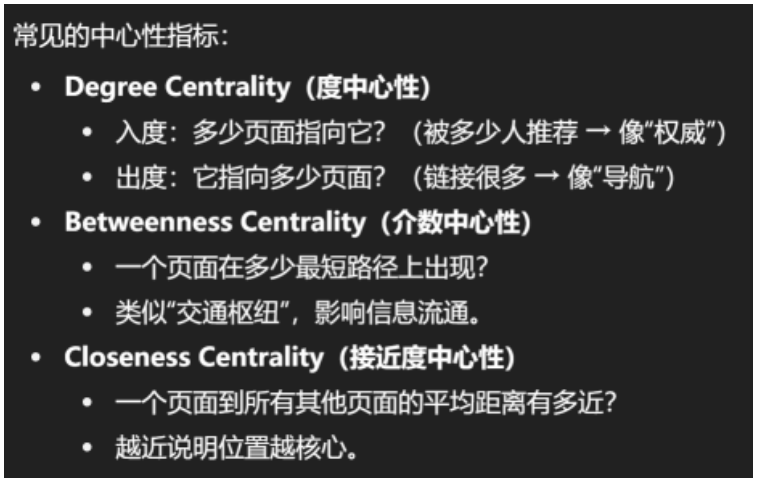
Co-citation network of Sloan Digital Sky Survey

- http://nevac.ischool.drexel.edu/~james/infovis09/FP-tree-visual.html

2023 © University of Michigan

First Try: Network Centrality

- A good ranking measure:
  - Making sense;
  - Robust to spamming;
  - Easy to compute;



- Can we simply go with degree centrality (number of connections)?
- Who is more important?
  - A page that links to many other pages
  - Or a page that is linked by many other pages
- Not stable: one can easily spam the ranking

2023 © University of Michigan

Betweenness/Closeness Centrality

- Betweenness: how many pairs of vertices have to go through you on their shortest path?

$$C_B(i) = \sum_{j,k \neq i,\, j<k} g_{jk}(i) / g_{jk}$$

Where:

- g_{jk} = the number of shortest paths connecting *j and k*
- $g_{jk}(i)$= the number of SPs that vertex *i* is on.

Sum over all pairs of nodes excluding i itself

- Closeness: average distance from everyone else to you

$$C_C(i) = \left[\sum_{j=1,\, j \neq i}^{N} d(i,j) \right]^{-1}$$

The smaller the length, the larger the closeness

All vertices but i

2023 © *University of Michigan*

Betweenness/Closeness Centrality

- Can we use them to rank webpages?
- Pros:
  - Makes sense (can be explained by how information propagates)
  - More robust than in-degree (adding many spam in-links won't change much on the average distance/shortest path of all nodes)
- But no one really tried this on ranking web pages
  - Hard to compute: need to compute all pair shortest paths, which takes $O(n^3)$.

2023 © *University of Michigan*

Ranking by Network Centrality

- Degree
- Betweenness
- Closeness
- …

- Vertices that are linked to by many important vertices are important – recursive definition!
- Question: how to model this in a network?

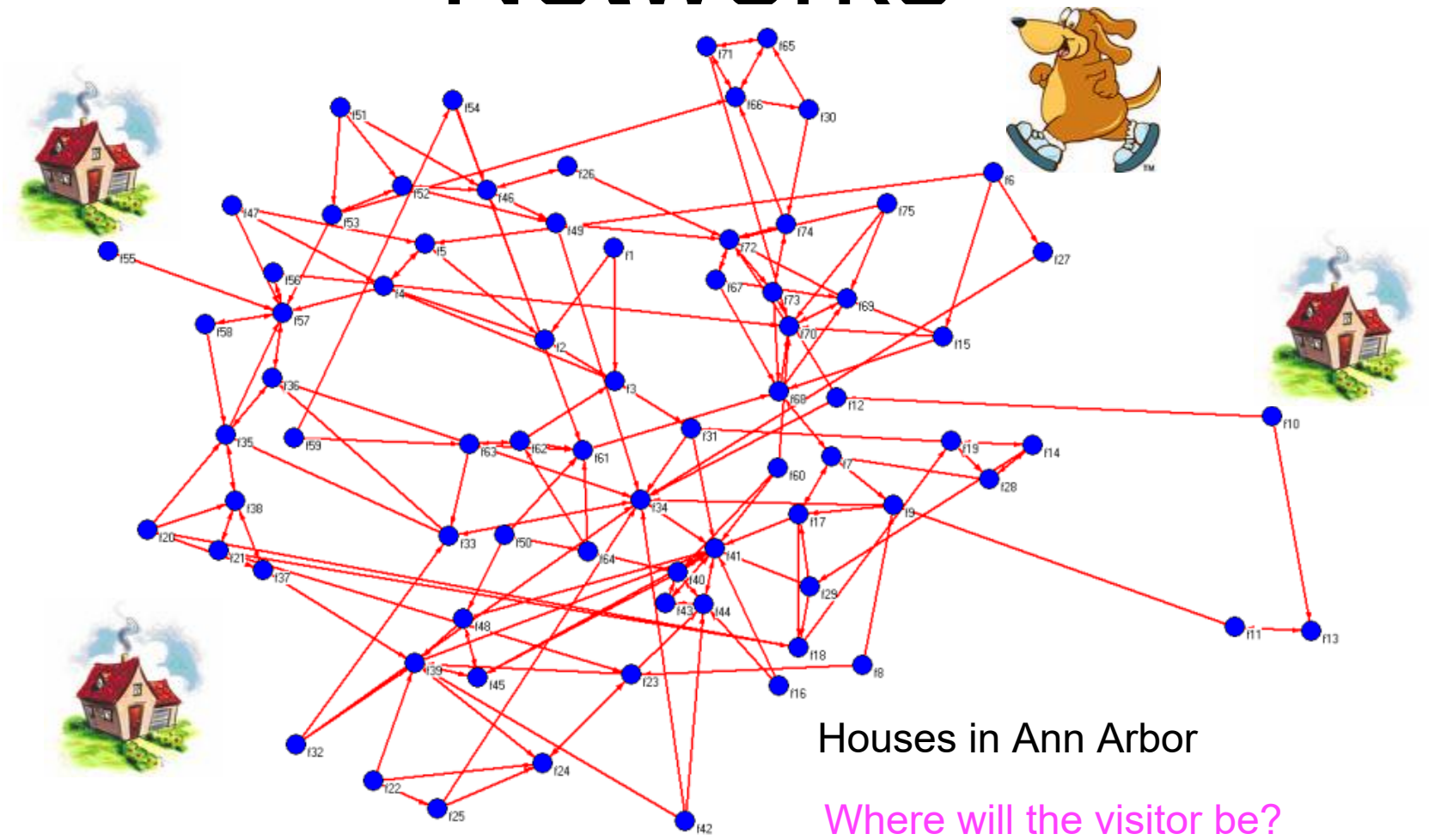
2023 © *University of Michigan*

Two methods we will cover today

- PageRank
  - To understand PageRank we will first cover **Random Walk**
- **HITS**

2023 © *University of Michigan*

Random Walk on Networks

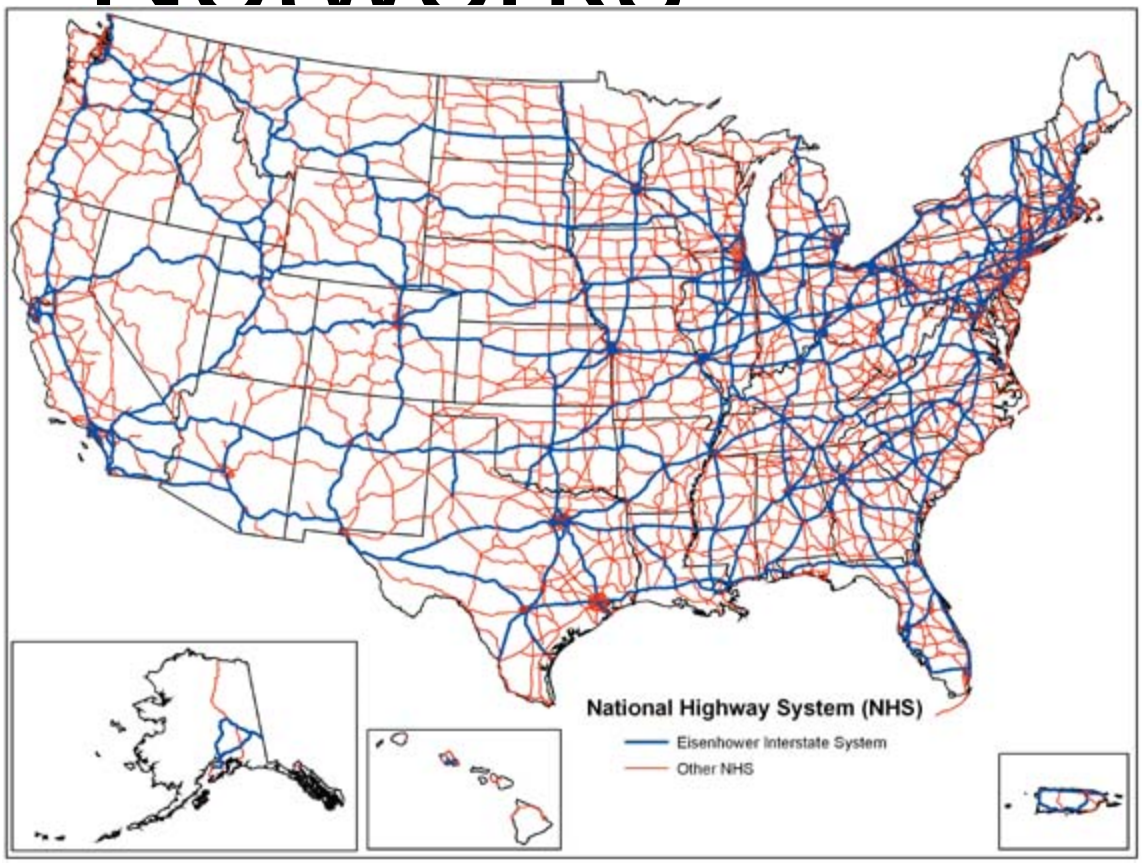


Houses in Ann Arbor

Where will the visitor be?
Which family gets the most visitors?

2023 © *University of Michigan*

Random Walk on Networks



- Track the trace of Forrest Gump:
  - Where to find Forrest in a month from now?
  - Where will Forrest spend most of his time?

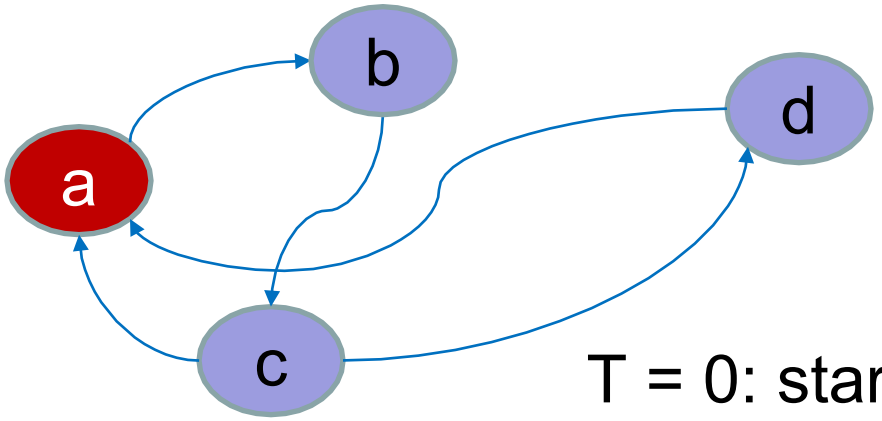
2023 © *University of Michigan*

What is a Random Walk?

- A “stochastic process” of walking on a network
- At every step, “walk” from the current position to one of its (directed) neighbors
- If more than one neighbor, “randomly” choose one neighbor to walk to.
- $p(X_t = i)$: the probability that at time t, the walk is at vertex i. (sometimes abbreviated as $p_t(i)$, or $p(i)$)
- For all vertices i, at any given time t, the sum of $p(X_t = i)$ is 1.

2023 © *University of Michigan*

Random Walk: Illustration

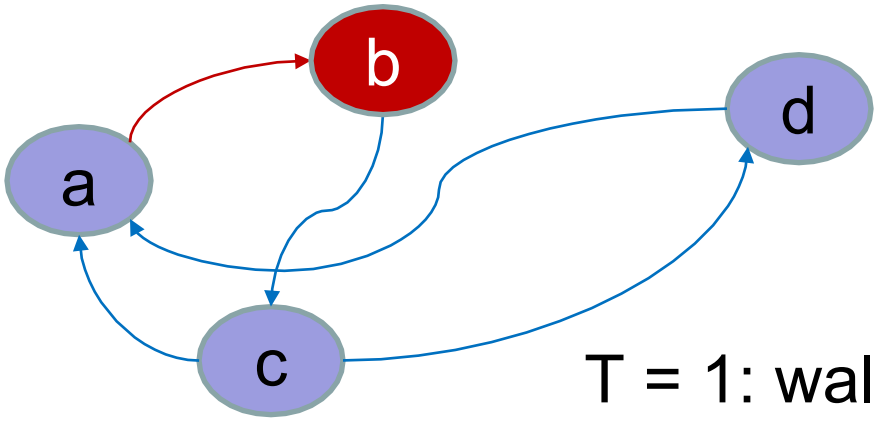


p(a) = 1
p(b) = 0
p(c) = 0
p(d) = 0

T = 0: start at the node a

2023 © *University of Michigan*

Random Walk: Illustration

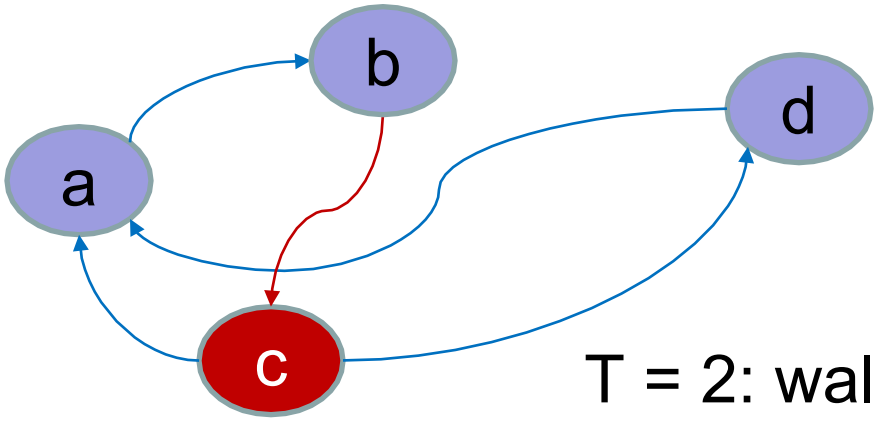


p(a) = 0
p(b) = 1
p(c) = 0
p(d) = 0

T = 1: walk to node b

2023 © *University of Michigan*

Random Walk: Illustration

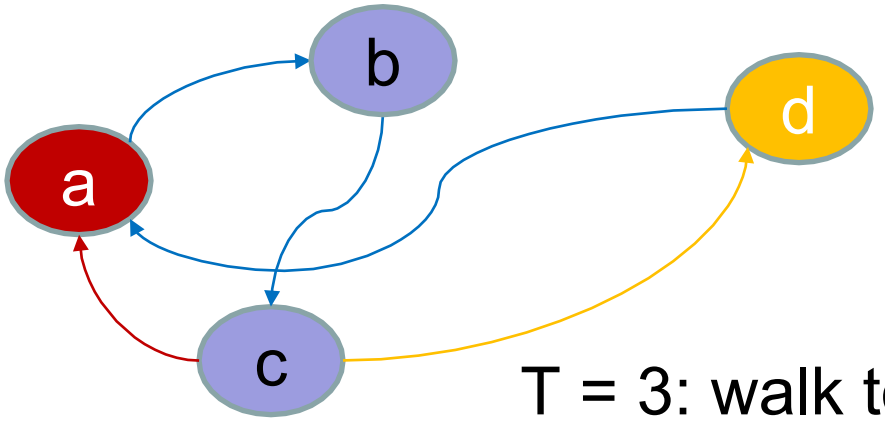


p(a) = 0
p(b) = 0
p(c) = 1
p(d) = 0

T = 2: walk to node c

2023 © *University of Michigan*

Random Walk: Illustration



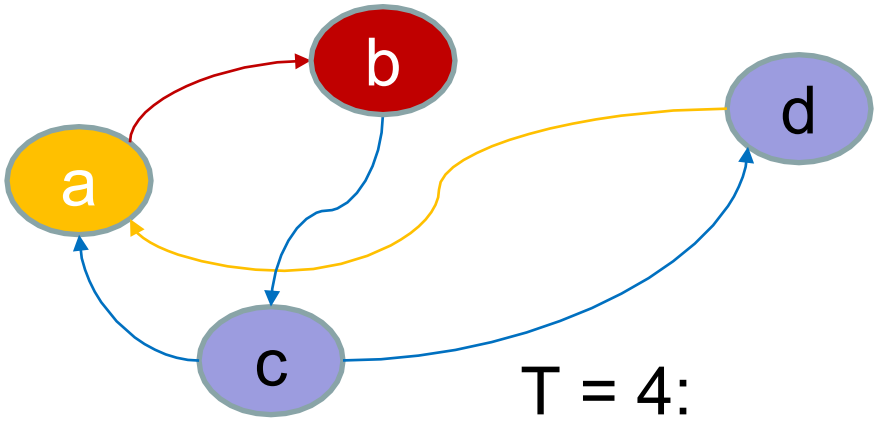
p(a) = 0.5
p(b) = 0
p(c) = 0
p(d) = 0.5

T = 3: walk to either a or d

Question: where will you be next?

2023 © University of Michigan

Random Walk: Illustration



p(a) = 0.5
p(b) = 0.5
p(c) = 0
p(d) = 0

T = 4:

50% prob. to node a (if it was at node d)
50% prob. to node b (if it was at node a)

Where will you be next?

2023 © *University of Michigan*

Network as a Markov Chain

- With such a random walk process, the network is known as a Markov chain
  - Named after Andrey Markov
- Basic property:
  - Directed graph (an undirected graph can be viewed as a directed graph with mutual links)
  - Every vertex is a *state*
  - The probability of visiting each vertex at time t is known as the *state* distribution at time t.
  - The state to visit next only depends on the current state

2023 © *University of Michigan*

Transition Probability

- Transition Probability: $p(X_{t+1} = j | X_t = i)$
- **转移概率** 表示：如果现在在节点 i，下一步走到节点 j 的概率是多少。
  - Usually abbreviated as p(j|i), or p(i→j), or p_{ij}
  - If the current state is i, how likely the walk will visit state j next?
  - P(j|i) = 1/out-degree(i)
  - Or w(i, j)/out-degree(i) if the edges are weighted.
- Adjacency matrix → transition matrix:

$$p(i \rightarrow j) = \frac{1}{\text{outdegree}(i)}$$

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1/2 & 1/2 & 0 & 0 & 0 \end{pmatrix}$$

2023 © University of Michigan

Random Walk in a Markov Chain

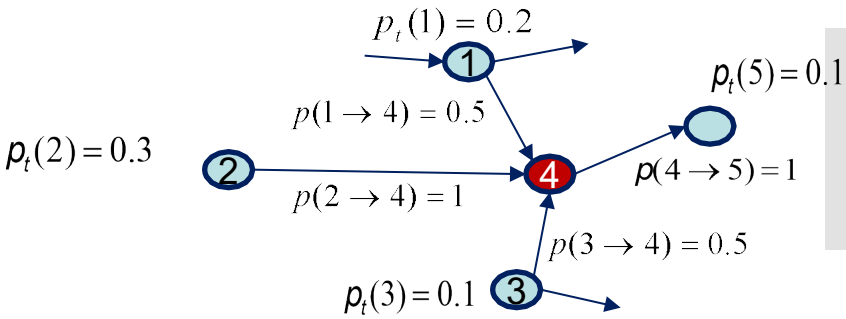
- Given
  - Probabilities: $p_t(i)$
  - Transition Probabilities: p(i→j)
- The distribution at time t+1 can be computed as:

$$p_{t+1}(j) = \sum_{i \in V} p_t(i) \cdot p(i \rightarrow j)$$

How likely the walk will be at j next

How likely the walk is currently at i

How likely the walk is going from i to j



Poll: What is $p_{t+1}(4)$?

Respond at pollev.com/cerenbudak421

$$p_{t+1}(4) = p_t(1) \cdot p(1 \rightarrow 4) + p_t(2) \cdot p(2 \rightarrow 4) + p_t(3) \cdot p(3 \rightarrow 4)$$
$$= 0.3 * 1 + 0.2 * 0.5 + 0.1 * 0.5 = 0.45$$

2023 © University of Michigan

2023 © *University of Michigan*

Stationary Distribution

- Two basic questions:
  - After walking for a sufficiently long time, how much time did you spend on each vertex? (How frequently did you visit each vertex?)
  - After walking for a sufficiently long time, which vertex will you be at (or visit next)?
- They can be explained by the "stationary distribution" of a random walk
- π: stationary distribution
  - The "steady" state distribution
  - π(i): the probability that the walk is visiting vertex i after a sufficiently long time.
  - This is independent from the initial state distribution (i.e., the start point).

2023 © *University of Michigan*

- **Stationary Distribution** 指的是：

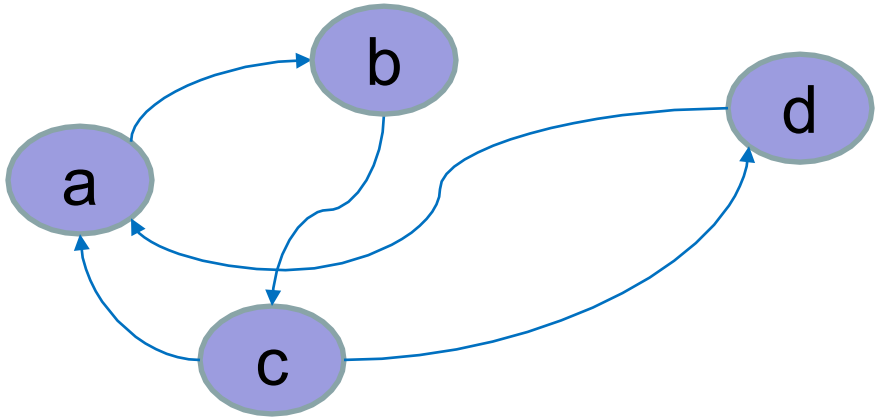
  当游走的步数趋向无穷大时，**在各个状态上的概率分布趋于稳定，不再变化。**

换句话说：

> 长期来看，随机游走者在某个节点出现的频率 = 该节点的平稳分布值。

2023 © *University of Michigan*

Example of a Stationary Distribution



Stationary distribution:

$\pi(a) = 2/7$

$\pi(b) = 2/7$

$\pi(c) = 2/7$

$\pi(d) = 1/7$

No matter where it starts, after a sufficiently large number of steps, the walk visited a, b, c twice as frequently as d; on the next step the walk is twice likely to be at a, b, or c than d.

2023 © *University of Michigan*

How Long is Sufficiently Long?

- "Stationary" distribution: the "steady" state distribution then the random walk reaches a stationary status
- When t is sufficiently large, the distribution is steady so that the distribution at time t equals the distribution at t+1

$$p_t(j) = p_{t+1}(j) = \pi(j)$$

$$\pi(j) = \sum_{i \in V} \pi(i) \cdot p(i \rightarrow j)$$

- When this condition holds for every vertex j, we have reached the steady status - stationary distribution.
- Again, this is independent from the start point.

2023 © *University of Michigan*

Will this Eventually Happen?

- For some networks, yes
  - Irreducible: possible to reach any node from any node
    - One strongly connected component
  - Positive recurrent: from any node, the walk can return to the same node in finite time.
- For some networks, no
  - Networks with unconnected components;
    - Walks starting in one component cannot reach the others
  - Network has nodes (or groups of nodes) with zero out-links;
    - Walks reaching those nodes will be trapped.
  - …

2023 © *University of Michigan*

How to Compute the Stationary Distribution?

$$\pi(j) = \sum_{i \in V} \pi(i) \cdot p(i \rightarrow j)$$

- Let P be the transition matrix (P_{ij} = p(i → j)), we have

$$\vec{\pi} = \vec{\pi} \cdot P$$

意思是：如果当前分布是 π，随机游走一步之后，分布仍然是 π。
因此 π是 **不变的、长期稳定的概率分布。**

- In other words, the stationary distribution π is a left eigenvector of the transition matrix P (with eigenvalue 1)
- π can be found by solving a linear system
  - Finds the exact solution of π , but this is usually time consuming

2023 © *University of Michigan*

How to Compute the Stationary Distribution?

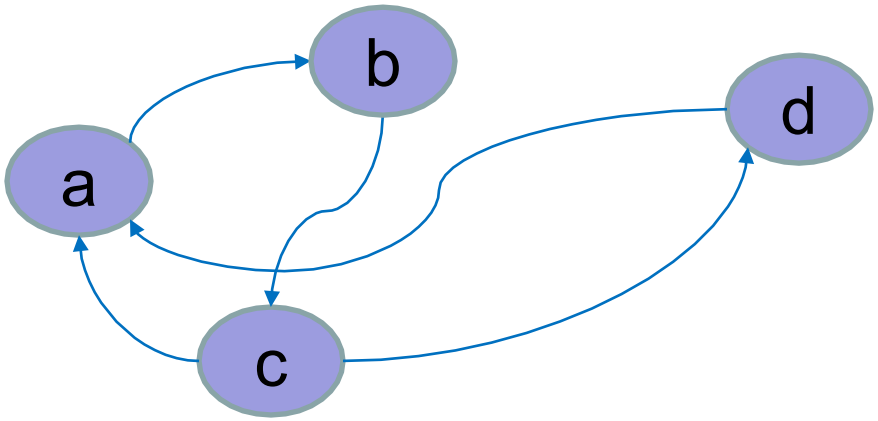
- Alternative strategy:
  - Make use of the characteristics of π:
  - π is independent from the starting state distribution
  - Choose a random start, and iteratively run the following random walk until convergence.

$$p_{t+1}(j) = \sum_{i \in V} p_t(i) \cdot p(i \rightarrow j)$$

  - π is found when for every vertex, $p_t(j) = p_{t+1}(j)$.

2023 © *University of Michigan*

Example of Finding π

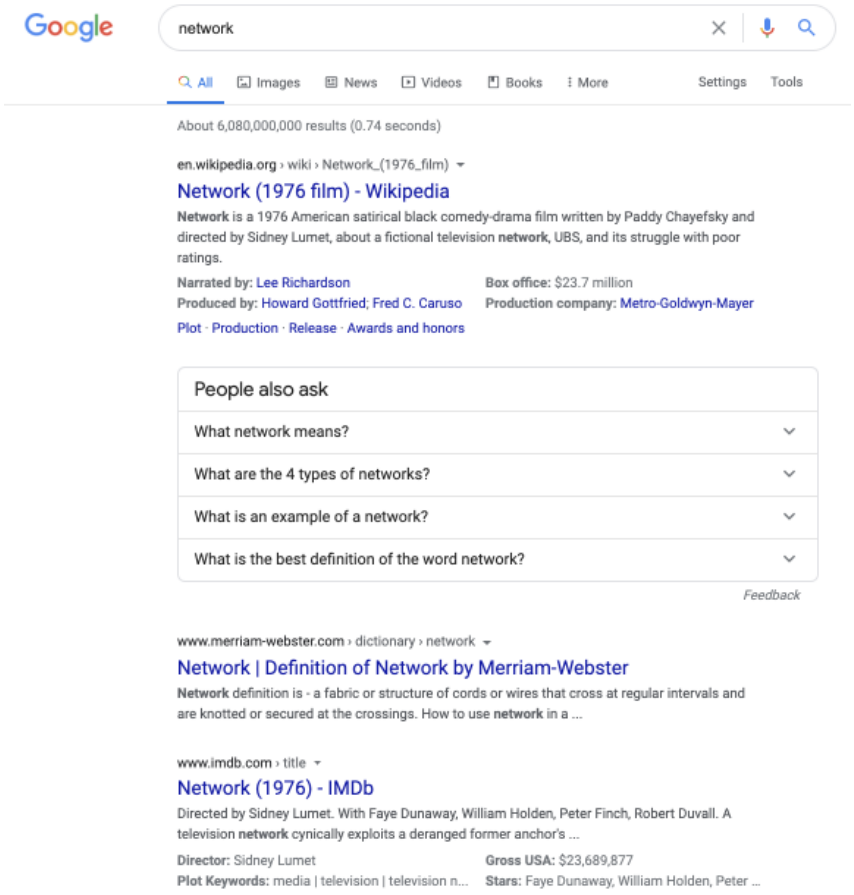


T → …
π(a) = 2/7
π(b) = 2/7
π(c) = 2/7
π(d) = 1/7

T = 0	T = 5	T = 10	…	T ~ 100
p(a) = 1	p(a) = 0	p(a) = 0.375		p(a) = 2/7
p(b) = 0	p(b) = 0.5	p(b) = 0.125		p(b) = 2/7
p(c) = 0	p(c) = 0.5	p(c) = 0.25		p(c) = 2/7
p(d) = 0	p(d) = 0	p(d) = 0.25		p(d) = 1/7

2023 © University of Michigan

Back to ranking the Web



Problem:
How do we know which individual pages are more important?

2023 © *University of Michigan*

PageRank: Bringing Order to the Web

- PageRank: Sergey Brin and Larry Page（判断哪个网页更重要）
- Intuition 1: links to a web page can be interpreted as endorsements or recommendations
  - the more links a page receives, the more likely it is to be a good/entertaining/provocative/authoritative/interesting information source
- Intuition 2: but not all link sources are equal
  - a link from a respected information source (e.g., NSF website)
  - a link from a page created by a spammer
- The autonomy of the web; bringing order to the web
- What's behind the primitive success of Google

2023 © *University of Michigan*

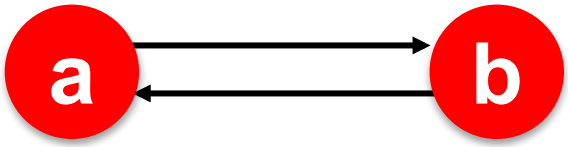
Behind PageRank

- “PageRank relies on the uniquely democratic nature of the web by using its vast link structure as an indicator of an individual page's value. In essence, Google interprets a link from page A to page B as a vote, by page A, for page B. But, Google looks at more than the sheer volume of votes, or links a page receives; it also analyzes the page that casts the vote. Votes cast by pages that are themselves "important" weigh more heavily and help to make other pages "important".” -- Google Technology

- What’s the secret behind PageRank?
- Essentially a probability distribution of how likely a person visits a page by randomly surfing on the web
- Woa, isn’t this simply the stationary distribution of a random walk on the web graph?
- Question: In what way is the web different?

2023 © *University of Michigan*

Does this converge?



$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{d_i}$$

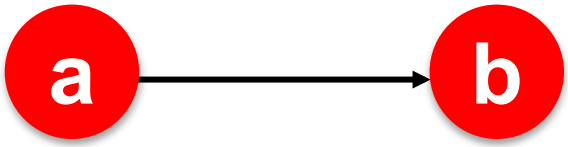
- **Example:**

$$\begin{matrix} r_a & = & 1 \\ r_b & & 0 \end{matrix}$$

Iteration 0, 1, 2, …

2023 © *University of Michigan*
J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org

Does it converge to what we want?



$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{d_i}$$

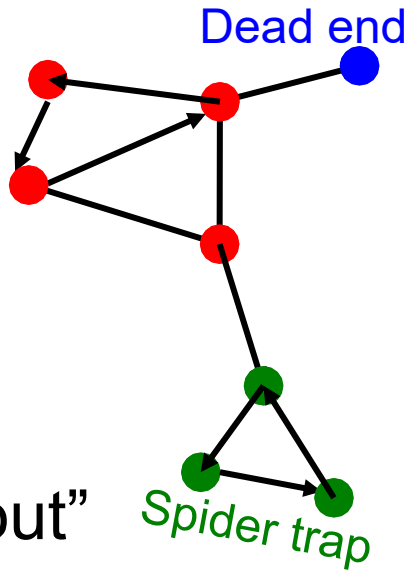
- **Example:**

r_a	=	1
r_b		0

Iteration 0, 1, 2, …

2023 © University of Michigan

PageRank: Problems



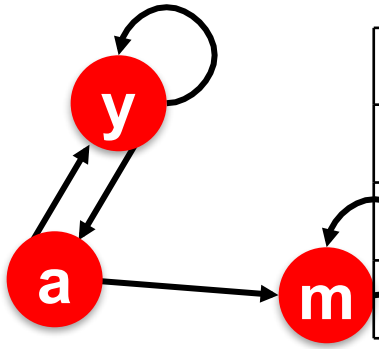
2 problems:

- **(1)** Some pages are **dead ends** (have no out-links)
  - Random walk has "nowhere" to go to
  - Such pages cause importance to "leak out"

- **(2) Spider traps:** (all out-links are within the group)
  - Random walked gets "stuck" in a trap
  - And eventually spider traps absorb all importance

Problem: Spider Traps

- **Power Iteration:**
  - Set $r_j = 1$
  - $r_j = \sum_{i \to j} \frac{r_i}{d_i}$
    - And iterate



	y	a	m
y	½	½	0
a	½	0	0
m	0	½	1

m is a spider trap

$r_y = r_y/2 + r_a/2$
$r_a = r_y/2$
$r_m = r_a/2 + r_m$

- **Example:**

$$\begin{pmatrix} r_y \\ r_a \\ r_m \end{pmatrix} = \begin{matrix} 1/3 & 2/6 & 3/12 & 5/24 & & 0 \\ 1/3 & 1/6 & 2/12 & 3/24 & \ldots & 0 \\ 1/3 & 3/6 & 7/12 & 16/24 & & 1 \end{matrix}$$

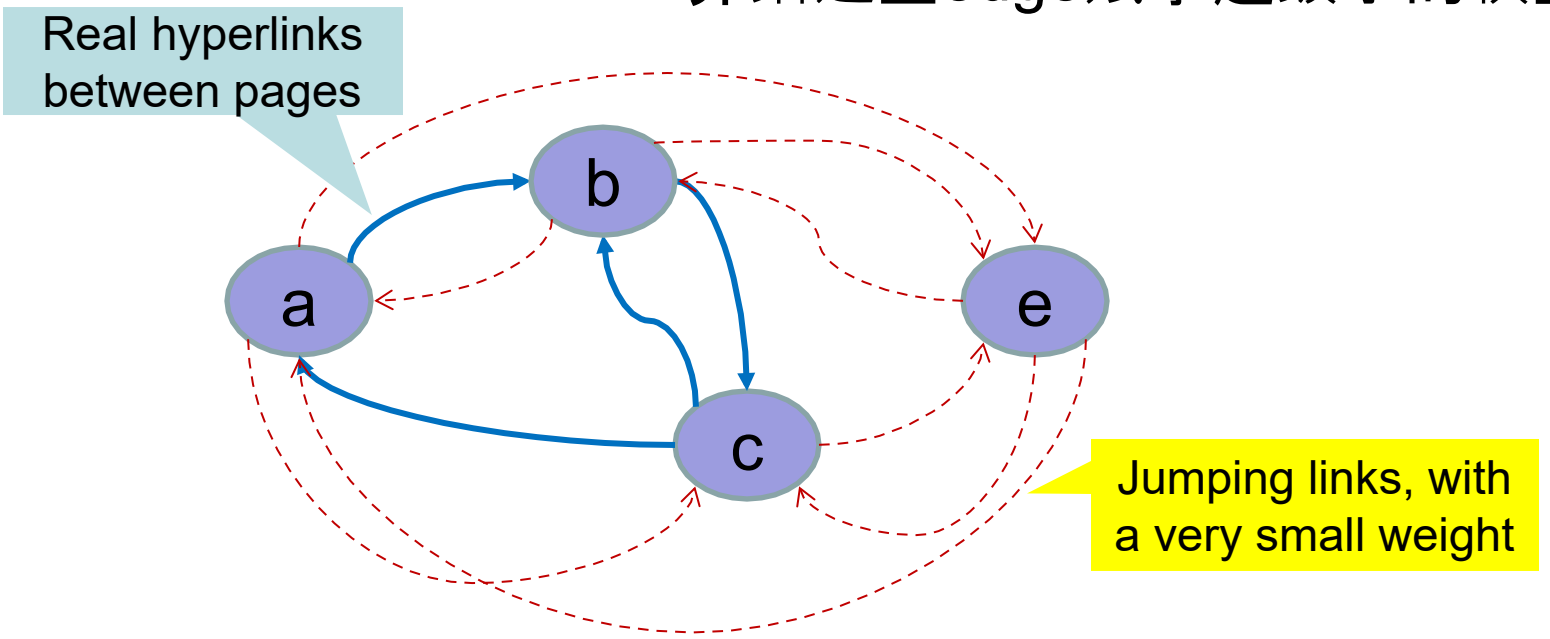
Iteration 0, 1, 2, …

All the PageRank score gets "trapped" in node m.

2023 © University of Michigan

Adding Random Jumps to the Web Network

加edge把graph变成complete graph
并给这些edge赋予超级小的权重



PageRank is essentially computing the stationary distribution of the random walk on this manipulated network

2023 © University of Michigan

Why do We Need Random Jumps?

- In PageRank, this is explained by the behavior of surfing without following the hyperlinks (surfing with jumping to a random web page).
- But essentially, they need this for a reason…
- Is the web graph connected at all? No!
- Is the web graph strongly connected? No!
  - Think about the bowtie model.
- Are there pages with no out-links? Yes!
- The truth is: there won't be a stationary distribution for a random walk on the web network without random jumps

2023 © *University of Michigan*

Computing PageRank

- d: damping factor which controls the jumping probability
- N: total number of vertices (web pages)
- D(i): the number of out-links of web page I

$$PR(j) = \frac{1-d}{N} + d \cdot \sum_{i:i\to j} \frac{PR(i)}{D(i)}$$

Visited by randomly jumping from other pages

Visited by surfing from one of its in-links

- Start with a random PageRank value for every vertex, iteratively updating the PageRank until converging

2023 © University of Michigan

PageRank: The Complete Algorithm

- **Input: Graph G and parameter β**
  - Directed graph G with **spider traps** and **dead ends**
  - Parameter β
- **Output: PageRank vector r**
  - **Set:** $r_j^{(0)} = \frac{1}{N}, \quad t = 1$
  - **do:**
    - $\forall j: r'^{(t)}_j = \sum_{i \to j} \beta \frac{r_i^{(t-1)}}{d_i}$

      $r'^{(t)}_j = 0$ if in-degree of j is 0
    - **Now re-insert the leaked PageRank:**

      $\forall j: r_j^{(t)} = r'^{(t)}_j + \frac{1-S}{N}$
    - $t = t + 1$
  - **while** $\sum_j \left| r_j^{(t)} - r_j^{(t-1)} \right| > \varepsilon$ **where:** $S = \sum_j r'^{(t)}_j$

If the graph has no dead-ends then the amount of leaked PageRank is **1-β**. But since we have dead-ends the amount of leaked PageRank may be larger. We have to explicitly account for it by computing **S**.

Alternate Solution: HITS

- HITS algorithm (developed by Jon Kleinberg):
  - start with a set of pages matching a query
  - expand the set by following forward and back links
  - Keep two scores for each web page:
    - Authority score a;
    - Hub score h;
  - compute the authority scores a, and hub scores h through an iterative approach:

$$a(j) = \sum_{i:i\to j} h(i) \qquad h(j) = \sum_{j\to i:i} a(i)$$

Sum over the hub scor
from all in-links

Sum over the authority score
from all out-links

每个网页有两种角色：

1. **Authority（权威）**
   - 网页本身内容有价值，被很多好页面指向。
   - 例子：一篇学术论文首页。
2. **Hub（枢纽）**
   - 网页本身可能没多少内容，但它指向很多好的页面。
   - 例子：一个论文索引页面，或导航目录。

👉 **好 Hub 指向好 Authority；好 Authority 被好 Hub 指向。**

这是一个 **互相强化（mutual reinforcement）** 的关系。

Alternate Solution: HITS

- A page has good content (*authority*) thus is pointed by others; a page also points to other pages (*hub*)
- Vertex pointing to many *good authorities* is a *good hub*;
- Vertex pointed to by many *good hubs* is a *good authority*
  - *Page pointing to only a lot of junk pages is junk*
  - *Page only pointed by a lot of junk pages is junk*
- Keeps two scores: A(i) and H(i)
  - A(i) = sum of H(j) for all j who points to i;
  - H(i) = sum of A(j) for all j pointed by i.
  - Iteratively updating
  - These values converge in the limit

2023 © *University of Michigan*

Example query: newspaper

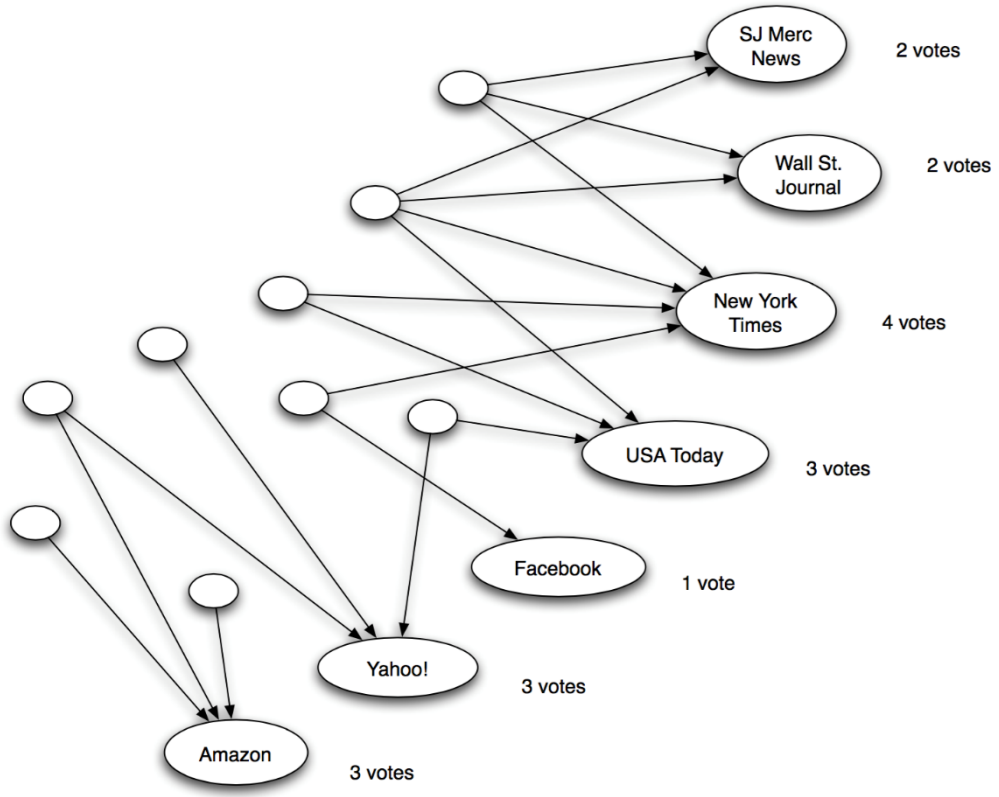


Figure 14.1: Counting in-links to pages for the query "newspapers."

2023 © *University of Michigan*

Example query: newspaper

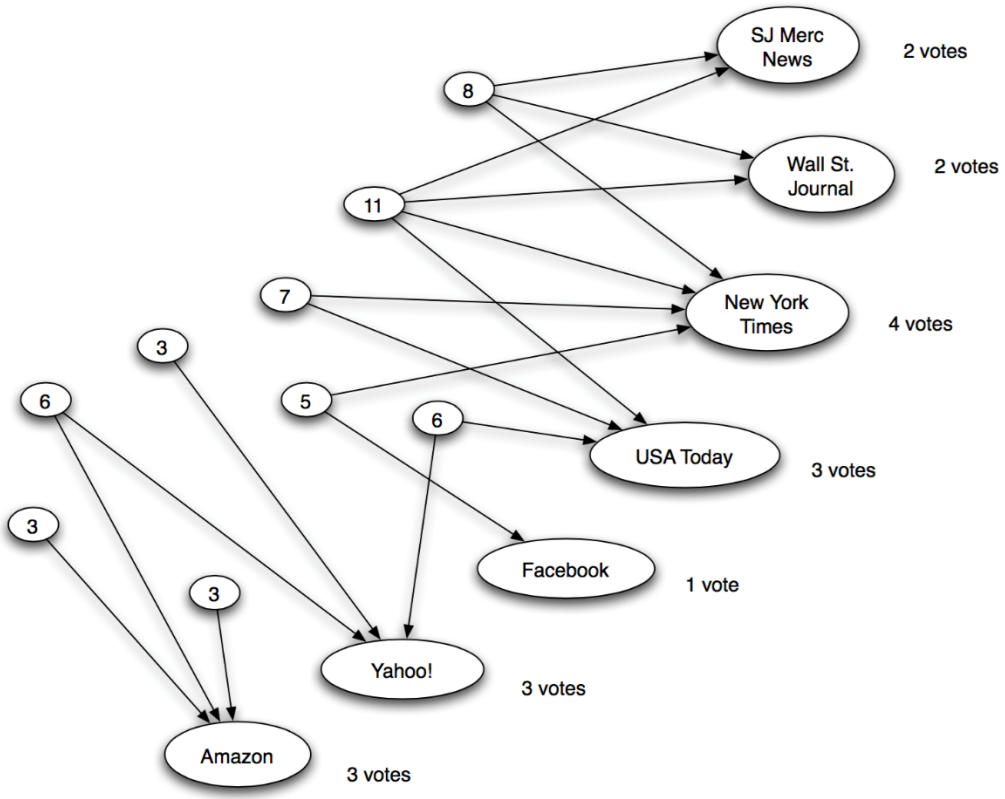


Figure 14.2: Finding good lists for the query "newspapers": each page's value as a list is written as a number inside it.

2023 © *University of Michigan*

Example query: newspaper

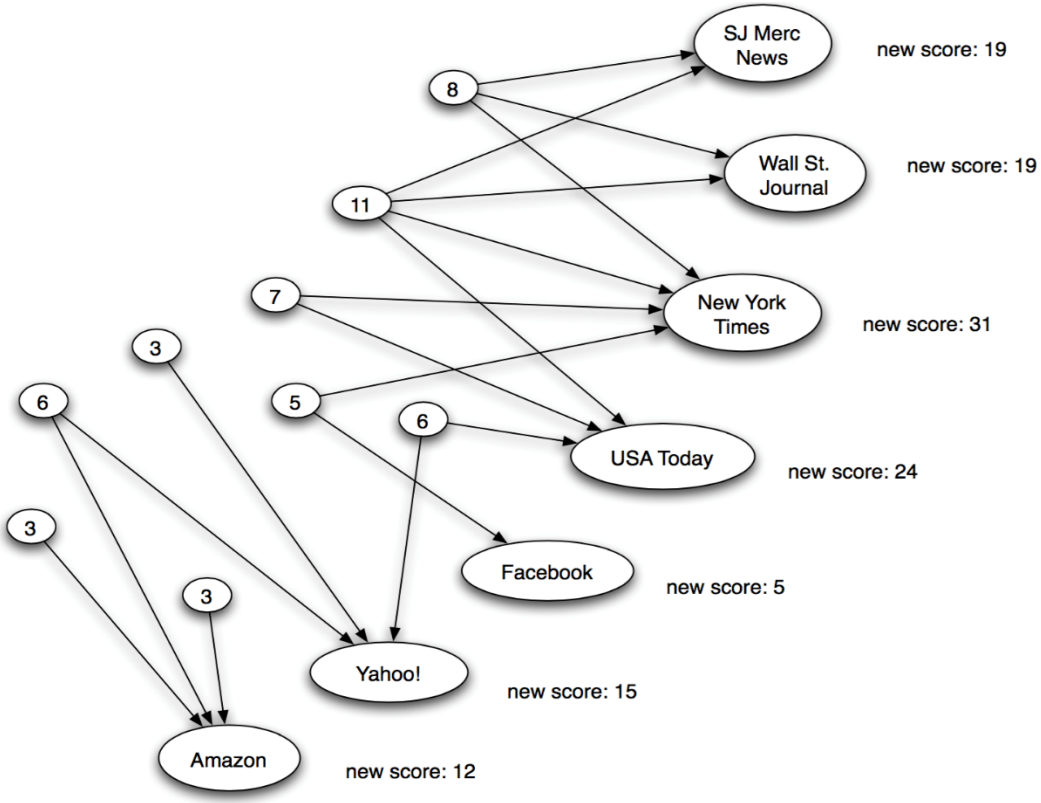


Figure 14.3: Re-weighting votes for the query "newspapers": each of the labeled page's new score is equal to the sum of the values of all lists that point to it.

2023 © *University of Michigan*

Task: Finding the best webpage for the query: newspaper

- In the limit:

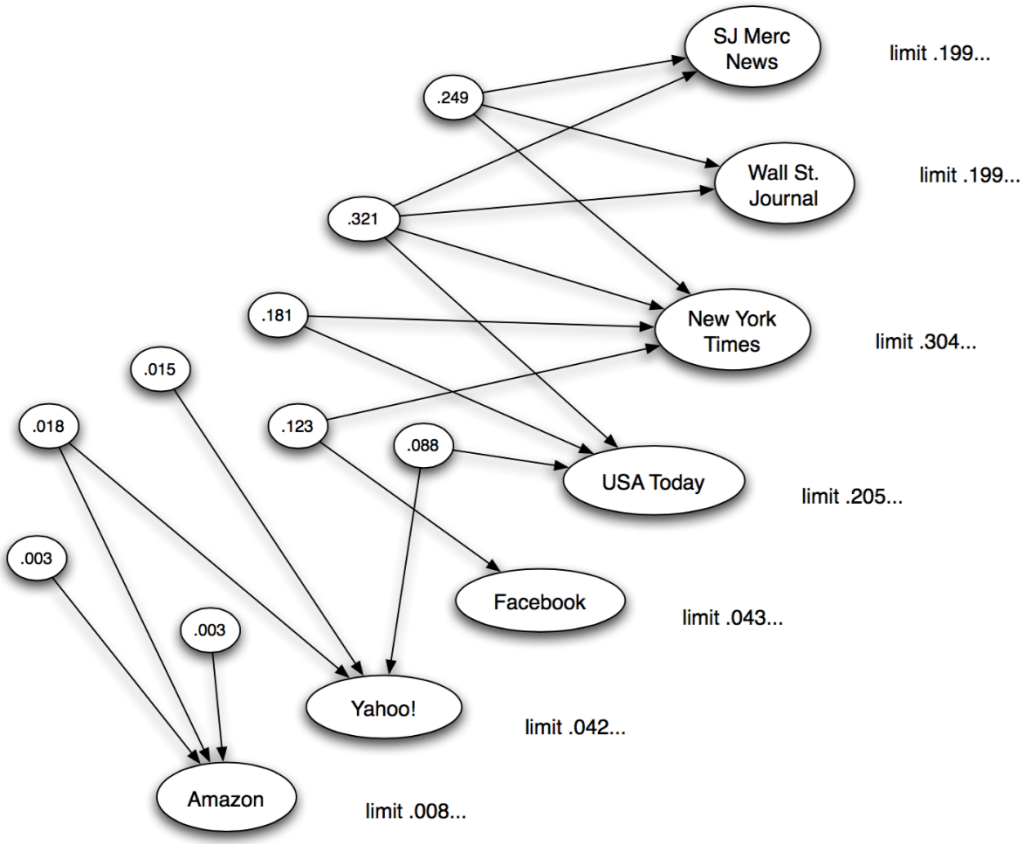


Figure 14.5: Limiting hub and authority values for the query "newspapers."

2023 © *University of Michigan*

3. 算法流程

1. **构建子图**
   - 给定查询词（例如 “newspaper”）。
   - 先找到一批相关页面（候选集），再扩展它们的入链和出链页面。
2. **初始化分数**
   - 每个页面的 **authority 分数 a(i)** 和 **hub 分数 h(i)** 初始设为 1。
3. **迭代更新**
   - Authority 更新：

$$a(i) = \sum_{j:j\to i} h(j)$$

     即：权威分数 = 所有指向它的 hub 的分数之和。
   - Hub 更新：

$$h(i) = \sum_{j:i\to j} a(j)$$

     即：枢纽分数 = 它所指向的所有 authority 的分数之和。
4. **归一化**
   - 为防止数值无限增长，每次迭代后对向量归一化。
5. **收敛**
   - 不断迭代，直到分数稳定。

2023 © *University of Michigan*

Implementation

- Given a query, first construct a focused subgraph, and then:

Iterate(G,k)

G: a collection of n linked pages

k: a natural number

Let z denote the vector $(1, 1, 1, \ldots, 1) \in \mathbf{R}^n$.
Set $x_0 := z$.
Set $y_0 := z$.

Initialization

For $i = 1, 2, \ldots, k$

Apply the $\mathcal{I}$ operation to (x_{i-1}, y_{i-1}), obtaining new x-weights x'_i.

Update authority scores

Apply the $\mathcal{O}$ operation to (x'_i, y_{i-1}), obtaining new y-weights y'_i.

Update hub scores

Normalize x'_i, obtaining x_i.
Normalize y'_i, obtaining y_i.

Normalize (important for convergence)

End

Return (x_k, y_k).

2023 © *University of Michigan*

Topic-Specific PageRank

Topic-Specific PageRank

随机跳转不再是“均匀分布到所有页面”，而是 限制在一组与主题相关的页面集合 S。

- Instead of generic popularity, can we measure popularity within a topic?
- Goal: Evaluate Web pages not just according to their popularity, but by how close they are to a particular topic, e.g. “sports” or “history”
- Allows search queries to be answered based on interests of the user
  - Example: Query “Trojan” wants different pages depending on whether you are interested in sports, history and computer security

Topic-Specific PageRank

- Random walker has a small probability of teleporting at any step
- Teleport can go to:
  - Standard PageRank: Any page with equal probability
    - To avoid dead-end and spider-trap problems
  - Topic Specific PageRank: A topic-specific set of "relevant" pages (teleport set)
- Idea: Bias the random walk
  - When walker teleports, she pick a page from a set S
  - S contains only pages that are relevant to the topic
    - E.g., Open Directory (DMOZ) pages for a given topic/query
  - For each teleport set S, we get a different vector r_S

2023 © *University of Michigan*
J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org

Matrix Formulation

- **To make this work all we need is to update the teleportation part of the PageRank formulation:**

$$A_{ij} = \begin{cases} \beta M_{ij} + (1-\beta)/|S| & \text{if } i \in S \\ \beta M_{ij} + 0 & \text{otherwise} \end{cases}$$

  - A is stochastic!
- We weighted all pages in the teleport set S equally
  - **Could also assign different weights to pages!**
- **Compute as for regular PageRank:**
  - Multiply by M, then add a vector
  - Maintains sparseness

2023 © *University of Michigan*
J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org

Discovering the Topic Vector S

- **Create different PageRanks for different topics**
  - The 16 DMOZ top-level categories:
    - arts, business, sports,…
- **Which topic ranking to use?**
  - User can pick from a menu
  - Classify query into a topic
  - Can use the **context** of the query
    - E.g., query is launched from a web page talking about a known topic
    - History of queries e.g., “basketball” followed by “Jordan”
  - User context, e.g., user’s bookmarks, …

2023 © *University of Michigan*
J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org

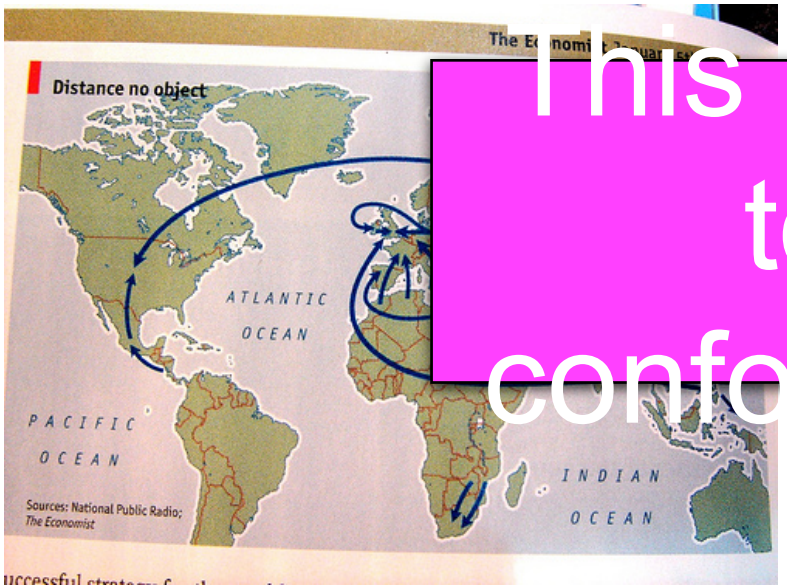
Diversity & Page Rank

Why is there an issue and what to do

2023 © *University of Michigan*

Reinforcements in Random Walks

- Random walks are not random - rich gets richer;
  - e.g., civilization/immigration – big cities attract larger population
  - Tourism – busy restaurants attract more visitors



This leads to conformity!

Source - http://www.resettlementagency.co.uk/modern-world-migration/

2023 © University of Michigan

1. 背景：随机游走的“富者愈富”效应

- 在普通的随机游走里，节点的访问概率会逐渐稳定在某个分布（比如 PageRank 的平稳分布）。
- 但现实中常见的现象是：
  - **文明/移民**：大城市越大，吸引的人越多。
  - **旅游/餐馆**：热门景点或餐馆因为人多而更受欢迎。

  👉 这是典型的 **“rich gets richer”（马太效应）**。

---

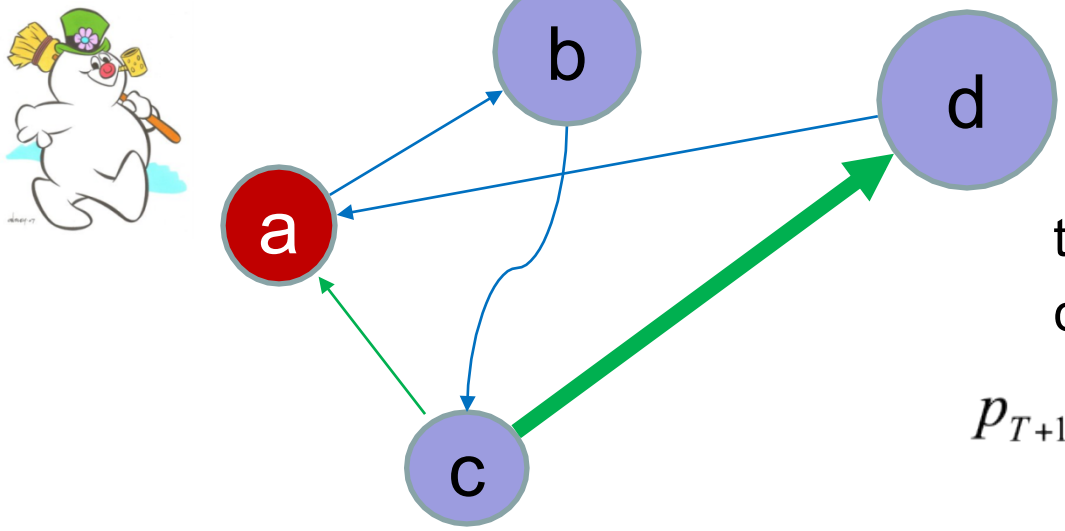
2. 强化随机游走 (Reinforced Random Walk)

- 在这种模型里，**转移概率会随着访问次数发生变化。**
- 即：你走过某条边/访问过某个节点的次数越多，未来再次走这条边/访问这个节点的概率就更大。
- 这种机制会导致 **收敛到少数“热门”节点**，产生更强的集中效应。

2023 © *University of Michigan*

Vertex-Reinforced (顶点强化) Random Walk

(Pemantle 92)



transition probabilities change over time

$$p_{T+1}(v) = \sum_{(u,v)\in E} p_T(u,v)\,p_T(u)$$

Reinforced random walk: transition probability is reinforced by the weight (number of visits) of the target state

$$p_T(u,v) \propto N_T(v)$$

2023 © University of Michigan

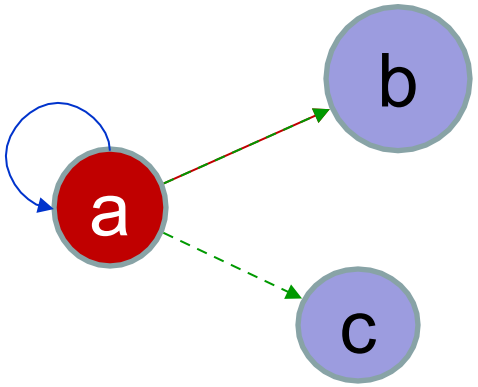
3. 区别总结

特性	强化随机游走 (RRW)	顶点强化随机游走 (VRRW)
强化对象	节点或边（更一般）	只针对节点
转移概率依赖	边的权重 or 节点的权重	邻居节点的访问次数
表现	可以模拟路径依赖（ERRW）或节点吸引力（VRRW）	更专注“节点富者愈富”现象
范围关系	更广泛的概念	RRW 的一个特例

2023 © *University of Michigan*

DivRank (Mei et al. 2010)

- A smoothed version of Vertex-reinforced Random Walk



$$p_T(u,v) = (1-\lambda)p^*(v) + \lambda \cdot \frac{p_0(u,v)N_T(v)}{D_T(u)}$$

Random jump, could be personalized

"organic" transition probability

Number of times *v* has been visited up to time *T*

- Adding self-links;
- Efficient approximations: use $E[N_T(v)]$ to approximate $N_T(v)$

Cumulative DivRank:

$$E[N_T(v)] \propto \sum_{t=0}^{T} p_t(v)$$

Pointwise DivRank:

$$E[N_T(v)] \propto p_T(v)$$

2023 © University of Michigan

PageRank vs. DivRank

2023 © *University of Michigan*

Another application of random walks:

Query Recommendation!

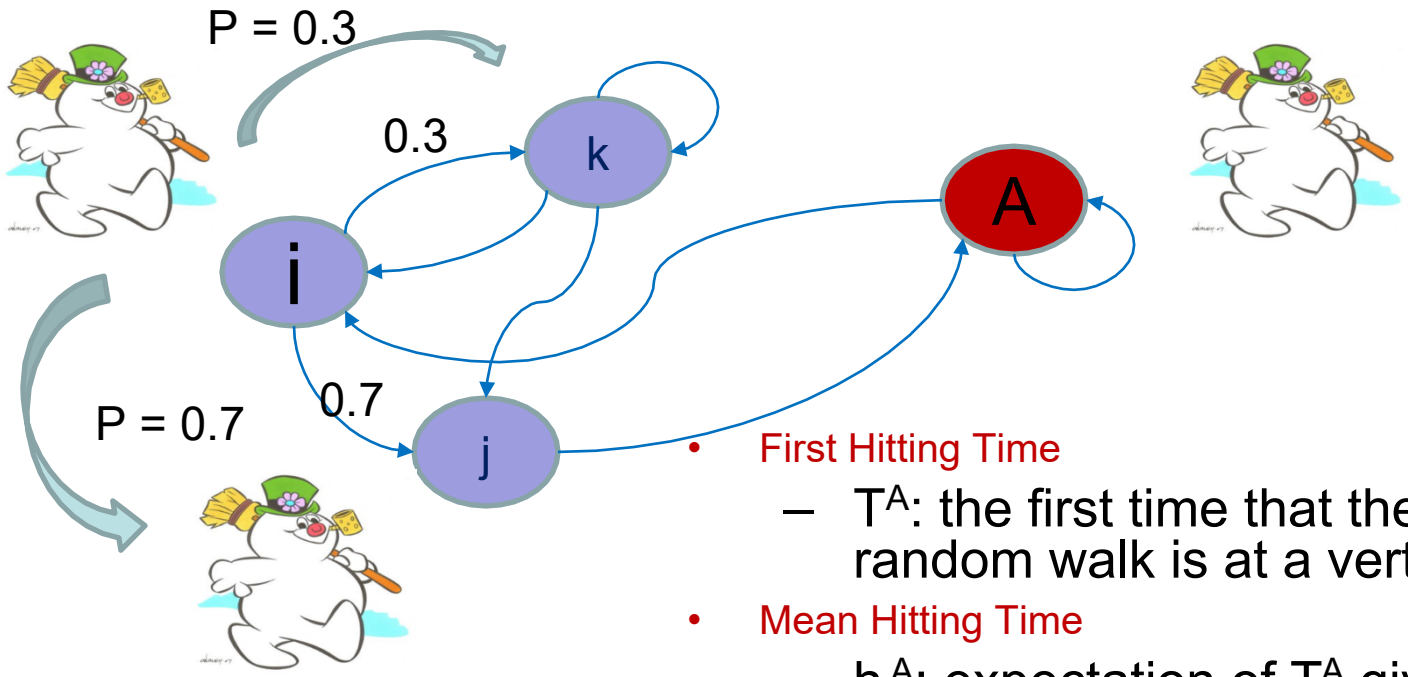
2023 © *University of Michigan*

Other Interesting Concepts of Random Walks

- **Hitting time** – h is the expected time in a random walk to reach vertex v starting from vertex u
- **Absorbing random walks –** If the random walk starts from vertex i, how likely is it to be eventually absorbed by vertex s?
- **Commute time** – The expected time to start from u, reach v, and then return to u
- **Cover time** – The expected time to start from u and visit every other vertex at least once
- **Meeting time** – The expected time of two random walks to start from u and v and meet at some vertex in between
- …

2023 © *University of Michigan*

Random Walk and Hitting Time

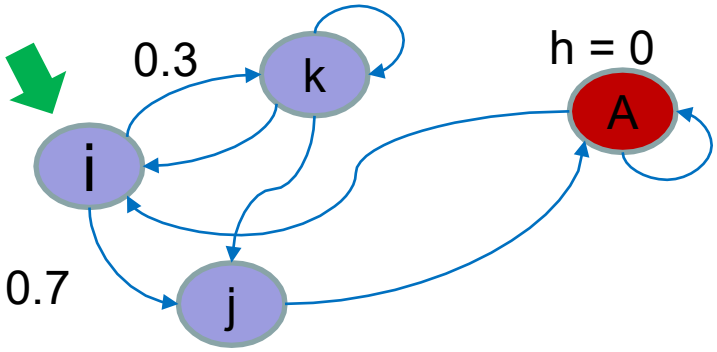


- First Hitting Time
  - T^A: the first time that the random walk is at a vertex in A
- Mean Hitting Time
  - h_i^A: expectation of T^A given that the walk starts from vertex i

2023 © *University of Michigan*

Computing Hitting Time

$h_i^A = 0.7\, h_j^A + 0.3\, h_k^A + 1$



T^A: the first time that the random walk is at a vertex in A

$$T^A = \min\{\, t : X_t \in A, t \geq 0\}$$

h_i^A: expectation of T^A given that the walk starting from vertex i

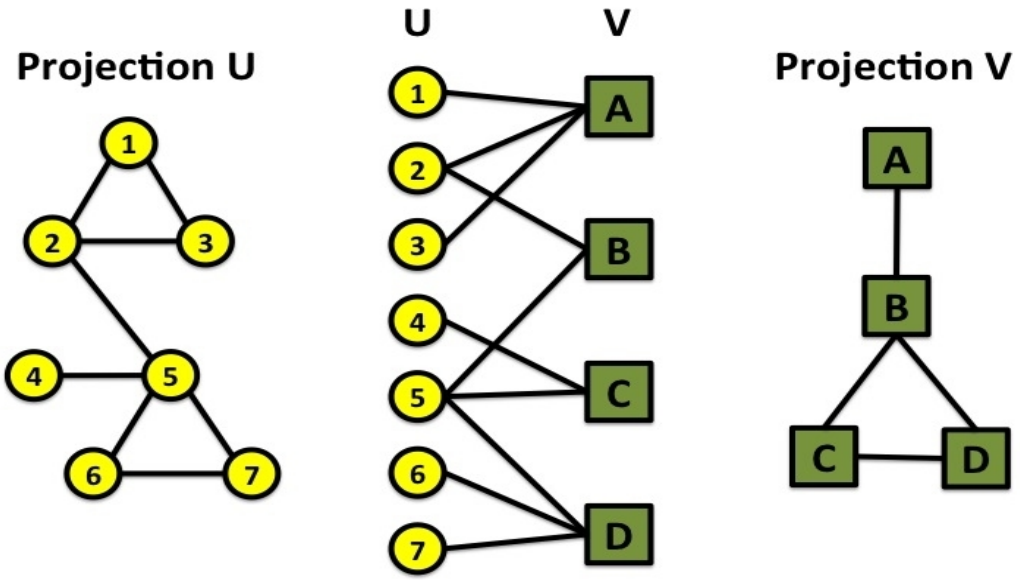
$$h_i^A = \begin{cases} \sum_{j \in V} p(i \to j) h_j^A + 1, & \text{for } i \notin A \\ 0, & \text{for } i \in A \end{cases}$$

Iterative Computation

2023 © University of Michigan

Quick digression: Bipartite Graphs

- Bipartite graph (or bigraph) is a graph whose nodes can be divided into two disjoint sets U and V such that every link connects a node in U to one in V; that is, U and V are independent sets.

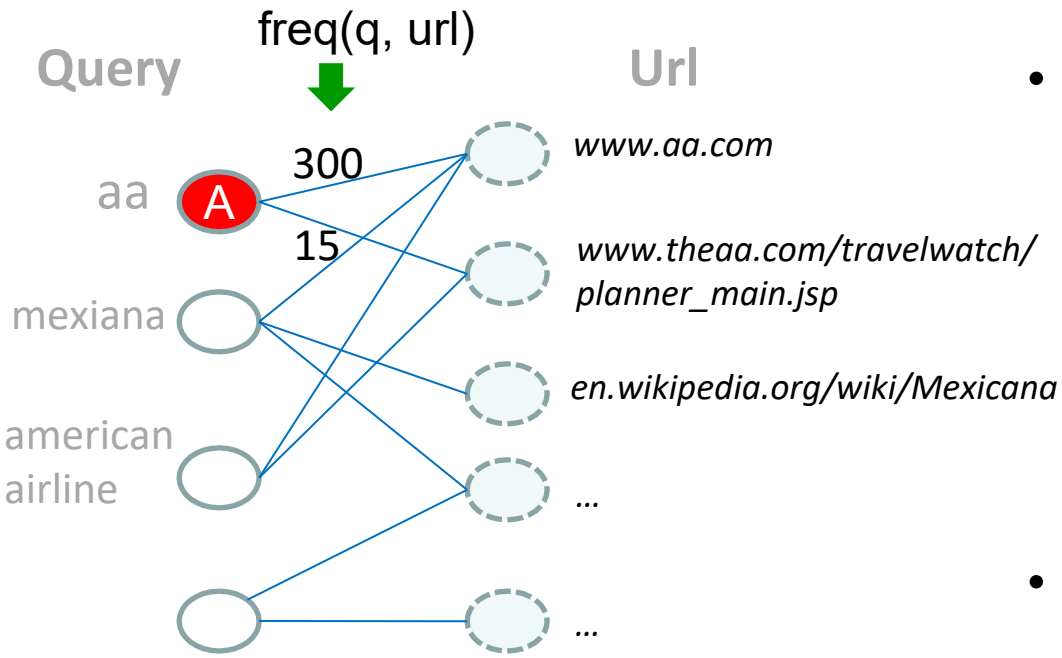


Examples:

Hollywood actor network
Collaboration networks
Disease network (diseasome)
Query networks (queries leading to pages)

2023 © *University of Michigan*

Using Hitting Time for Recommendation



Why not consider all <Query, Url> pairs?

- Mei et al. 2008. Query suggestion using hitting time

- Construct a (kNN) subgraph from the query log data (of a predefined number of queries/urls)
  - Given a query, construct a subgraph using DFS from that node until certain number of queries is reached.
- Compute transition probabilities p(i → j)
- Compute hitting time h_i^A
- Rank candidate queries using h_i^A

2023 © University of Michigan

Next Generation Search Engine (2010)?

- Better support for query formulation
  - Allow querying from any task context
  - Query by examples
  - Automatic query generation (recommendation)
- Better search accuracy
  - More accurate information need understanding (more personalization and context modeling)
  - More accurate document content understanding (more powerful content analysis)
- More complex retrieval criteria
  - Consider multiple utility aspects of information items (e.g., readability, quality, communication cost)
  - Consider collective value of information items (context-sensitive ranking)
- Better result presentation
  - Better organization of search results to facilitate navigation
  - Better summarization
- More effective and robust retrieval models
  - Automatic parameter tuning
- More scalable retrieval architecture
  - P2P

2023 © *University of Michigan*

Next Generation Search Engines?

- More specialized/customized
  - Special group of users (community engines, e.g., Citeseer)
  - Personalized (better understanding of users)
  - Special genre/domain (better understanding of documents)
- Learning over time (evolving)
- Integration of search, navigation, and recommendation/filtering (full-fledged information management)
- Beyond search to support tasks (e.g., shopping)

2023 © *University of Michigan*

What do You Think?

- Perspective 1: Google Rules
  - No matter what you call as “the next generation search engine”, it will be from Google
- Perspective 2: Bing (or Baidu, Yandex…) has a chance
  - Vertical search engines have their advantage that will beat Google one day
- Perspective 3: None of them
  - The next generation search engine will grow in a totally different soil (Meta? Tiktok? Twitter? Quora? Somewhere else?).

2023 © *University of Michigan*

You Should Know

- There are much more challenge in a Web search engine than in a standard IR system
  - More data, more complexity, more opportunities
- How PageRank and HITS work
  - What are issues with PageRank and associated extensions
- Research issues in Web search engines

2023 © *University of Michigan*