

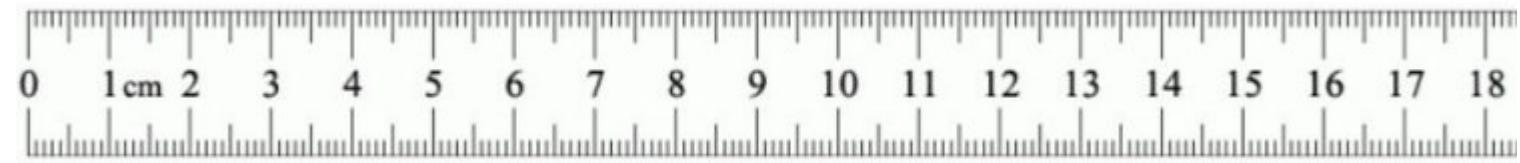
# GRAPH | NETWORK ANALYSIS (PART II)

---

Peng Wang

# MEASURES & METRICS

---



中心度

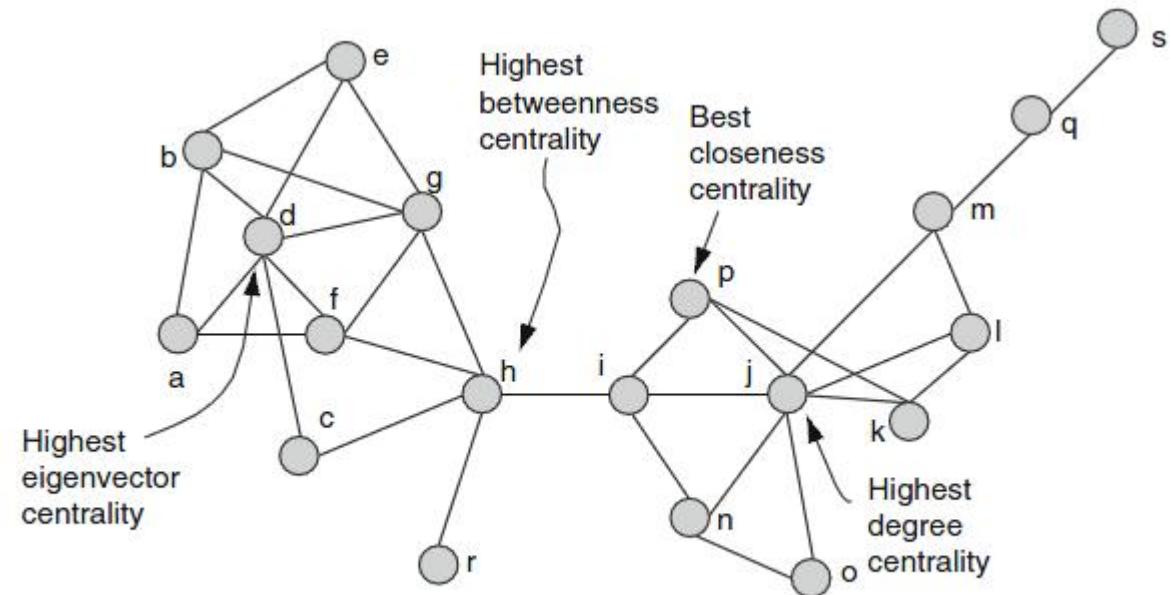
# Centrality ? finding the key players

Degree Centrality

Closeness Centrality

Betweenness Centrality

Eigenvector Centrality



Ortizarroyo D. Discovering Sets of Key Players in Social Networks[J]. Computational Social Network Analysis Computer Communications & Networks, 2010:27-47.

特征向量中心度

# Eigenvector Centrality

- an extension of degree centrality
- a player with many important friends is important

$x_i'$  表示节点*i*的中心度，是邻居节点的中心度之和

$$x'_i = \sum_j A_{ij} x_j$$

第t次迭代后的中心度向量

$$x(t) = A^t x(0)$$

$$Ax = k_1 x$$

*i*节点重要有两个条件：

*i*的邻居要多

*i*的邻居要重要

$x$ 是元素为 $x_i$ 的向量

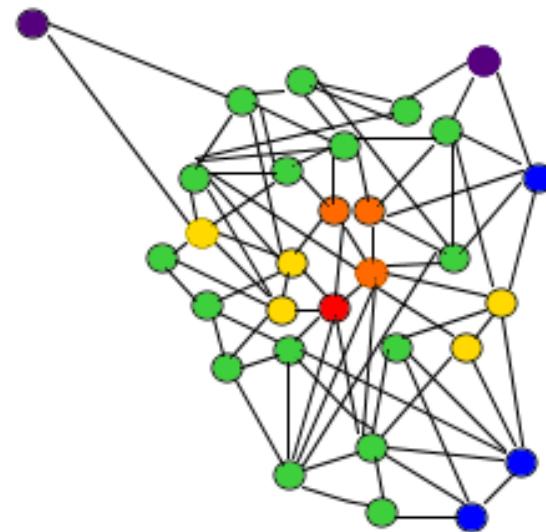
$k_1$ 是邻接矩阵 $A$ 的最大特征值(主特征值)

$x$ 是邻接矩阵 $A$ 的主特征向量

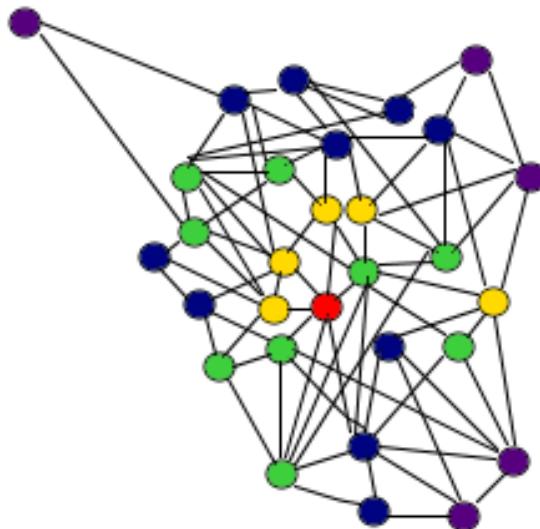
特征向量中心度

# Eigenvector Centrality

Degree Centrality



Eigenvalue Centrality



Degree And Eigenvector centrality

# Eigenvector Centrality

- For directed graph, such as WWW

$$Ax = k_1 x$$

? 

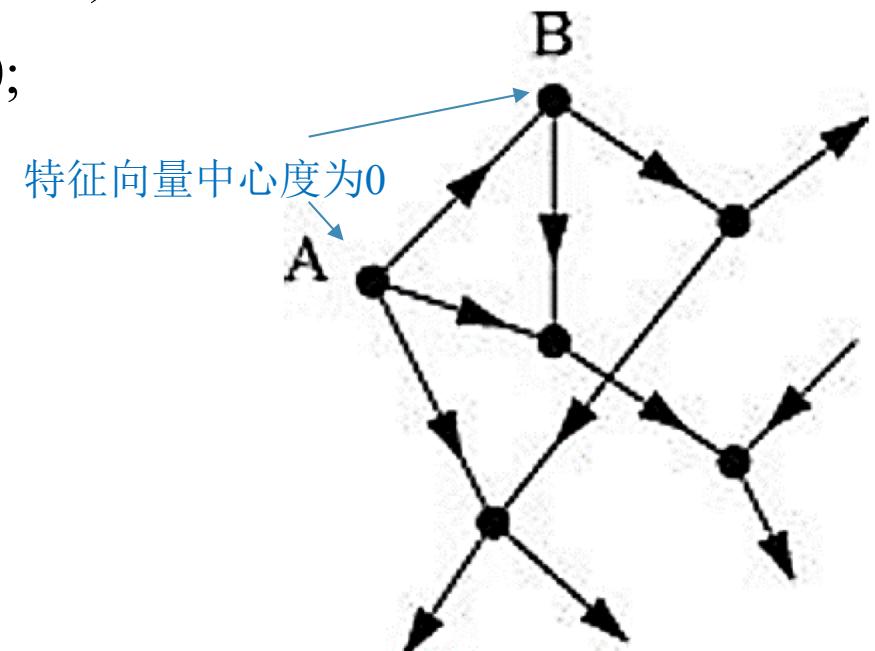
$k_1$ 是邻接矩阵 $A$ 的最大特征值(主特征值)

$x$ 是邻接矩阵 $A$ 的**右**主特征向量

为什么是右主特征向量, 而不是左主特征向量

# Eigenvector Centrality - problems

- the EC of nodes with no in-links are 0 (like node A);
- the EC of nodes with in-links from A are 0 (like node B)
- Only nodes in a SCC or its <sup>外向分支</sup>out-component are not 0;
- for DAG, each node is 0...



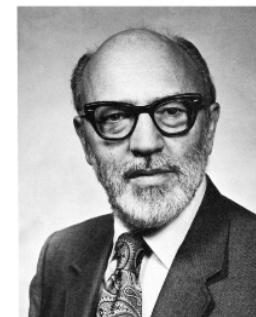
# Katz Centrality

$$x'_i = \alpha \sum_j A_{ij} x_j + \beta$$

为每个节点加上一些“免费”的中心度， $\beta$ 常设为1

↑  
自由参数，调节特征向量与常数项的平衡

- $\alpha \rightarrow 0: x_i \rightarrow \beta$
- $\alpha > \frac{1}{k_1}$ :  $x_i$  does not converge 不收敛
- $\alpha \in \left(0, \frac{1}{k_1}\right)$



**Leo Katz**

# Katz Centrality



**What is the problem of Katz Centrality?**

十大经典数据挖掘算法之一

# PageRank [1] Centrality



Larry Page

Sergey Brin

## The PageRank Citation Ranking: Bringing Order to the Web

January 29, 1998

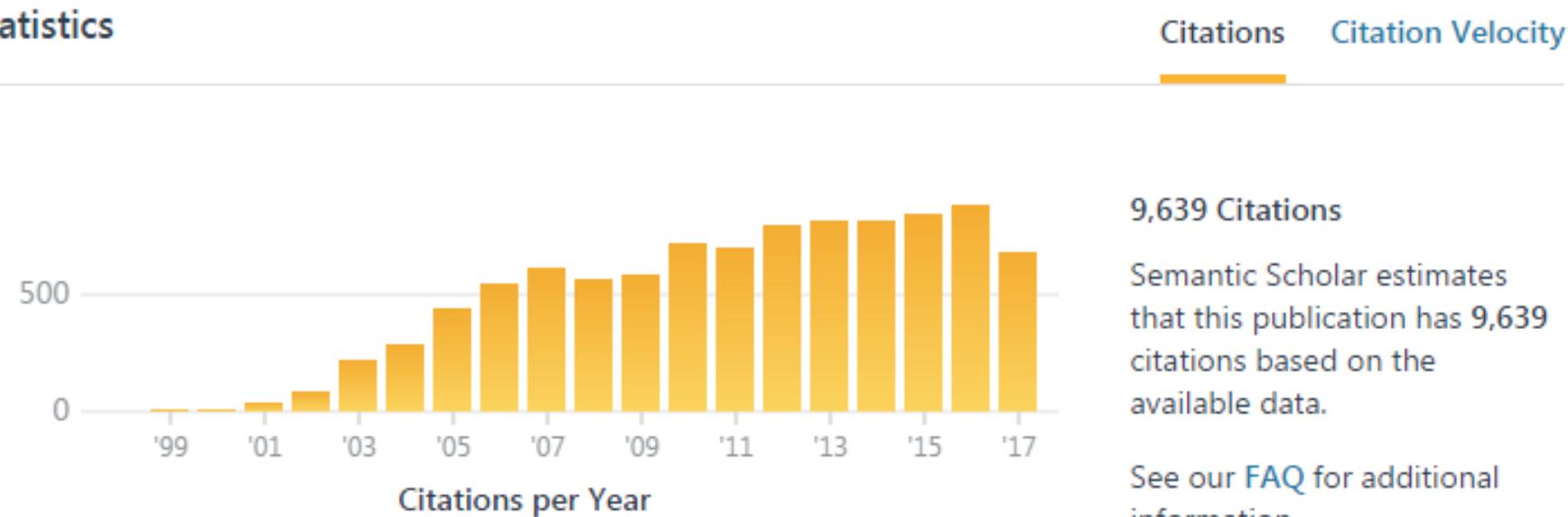
### Abstract

The importance of a Web page is an inherently subjective matter, which depends on the readers interests, knowledge and attitudes. But there is still much that can be said objectively about the relative importance of Web pages. This paper describes PageRank, a method for rating Web pages objectively and mechanically, effectively measuring the human interest and attention devoted to them.

We compare PageRank to an idealized random Web surfer. We show how to efficiently compute PageRank for large numbers of pages. And, we show how to apply PageRank to search and to user navigation.

# PageRank

## Statistics

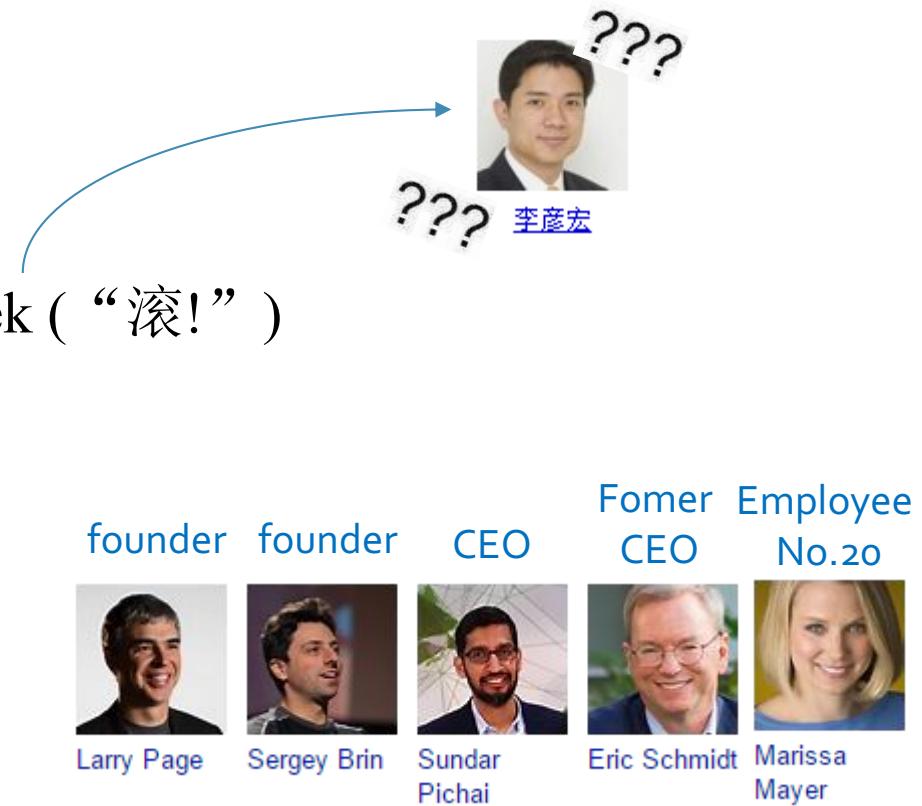


# THE RISE OF Google™

---

# Google History

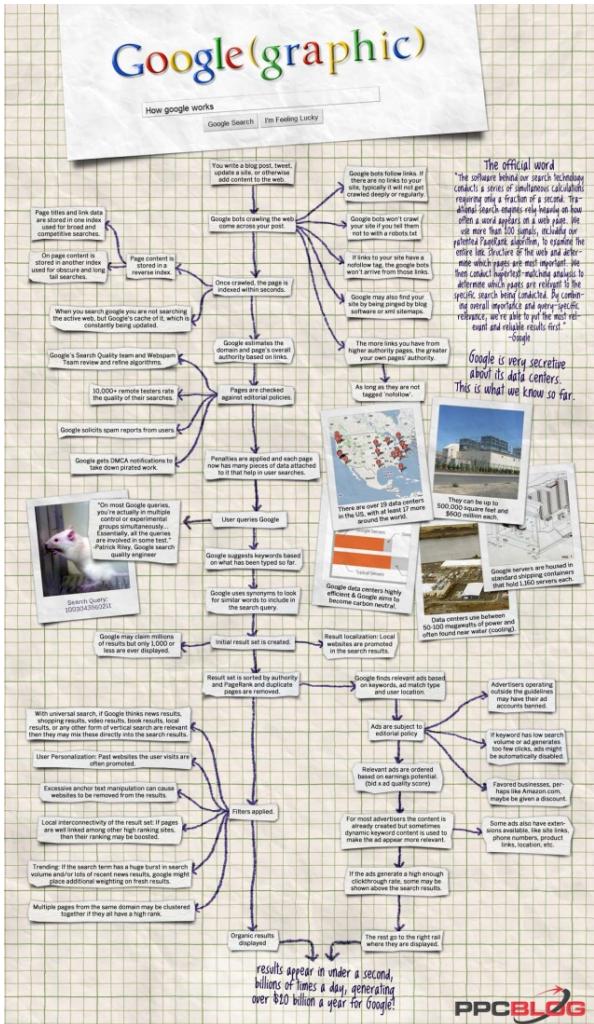
- Larry Page and Sergey Brin from Stanford
- From “What Box” to  $10^{100}$
- Refused by Altavista (say no for \$1M), InfoSeek ( “滚!” )
- 100,000 dollars from Sun
- Beat Yahoo, Excite.com, InfoSeek, Lycos





# Google Facts

- Google receives over **63,000 searches per second** on any given day.
- Google takes over **200 factors into account** before delivering you the best results to any query in a fraction of a second.
- Google has a market value of **\$739 billion**.
- Google user statistics: **2 billion users on Android** and 500M on Google Photos.
- Google search statistics: Google's search engine performed **1.2 trillion searches in 2016**.
- Google has **90.46% of the search engine market** share worldwide.
- Google office facts: the company has **88,110 full-time employees**.
- **30.9%** of the global Google employees are **female**.
- An average person conducts **3–4 searches every single day**.
- There are more than **70,000 Google queries per second**.

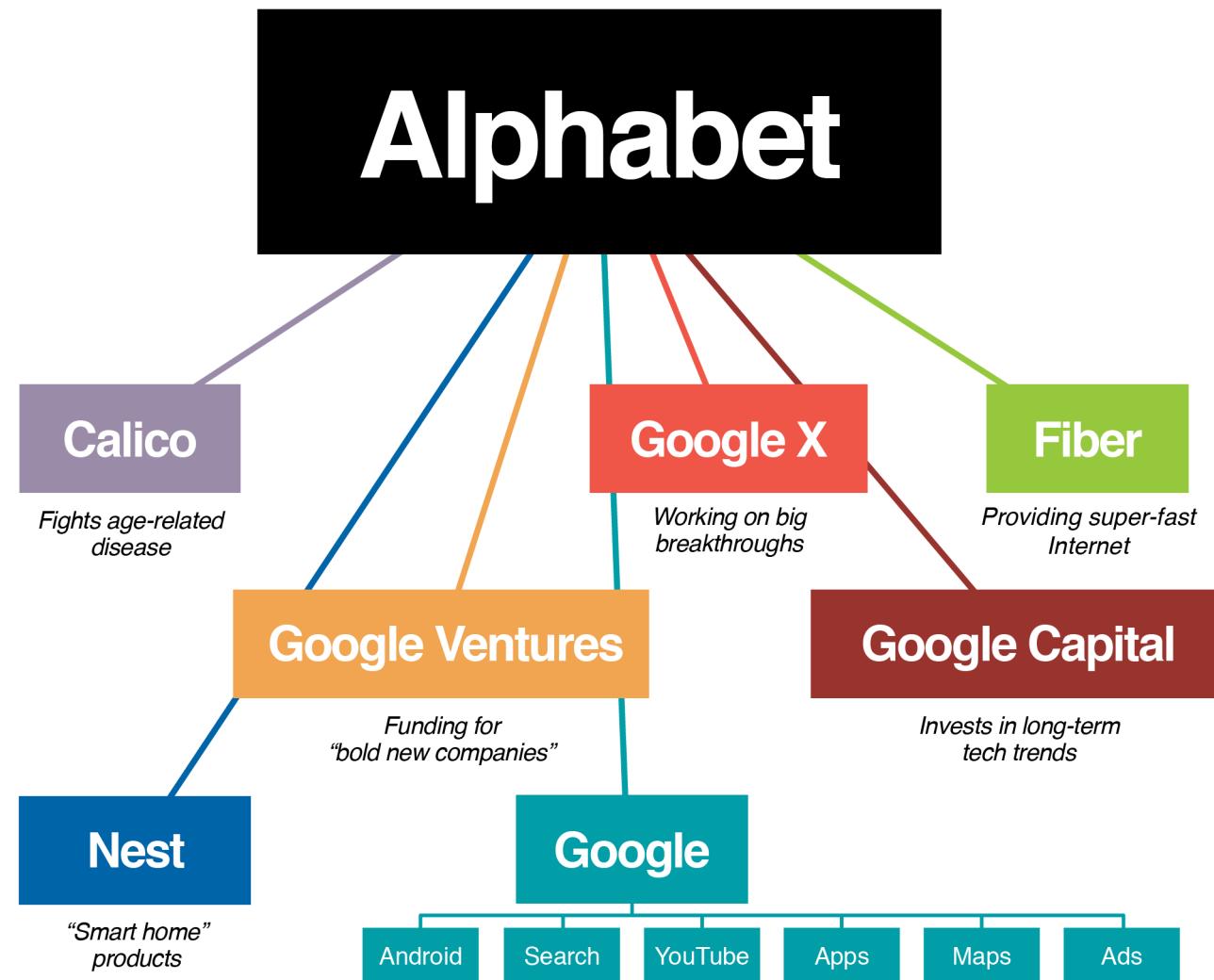


# Understanding How Google Works

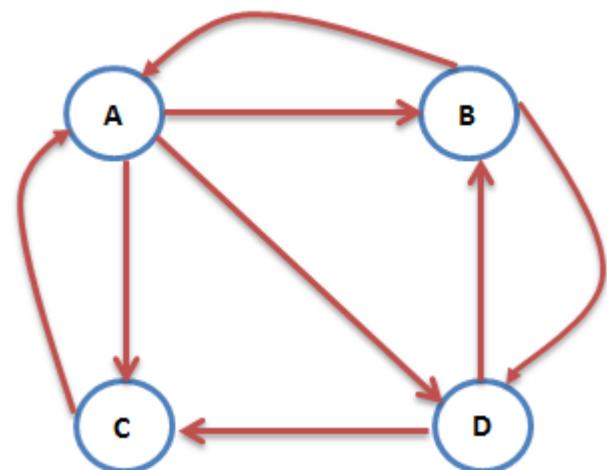
English: <http://ppcblog.com/how-google-works/>

Chinese:

<http://www.kuqin.com/searchengine/20130320/334054.html>



# PageRank Basic Idea 1:



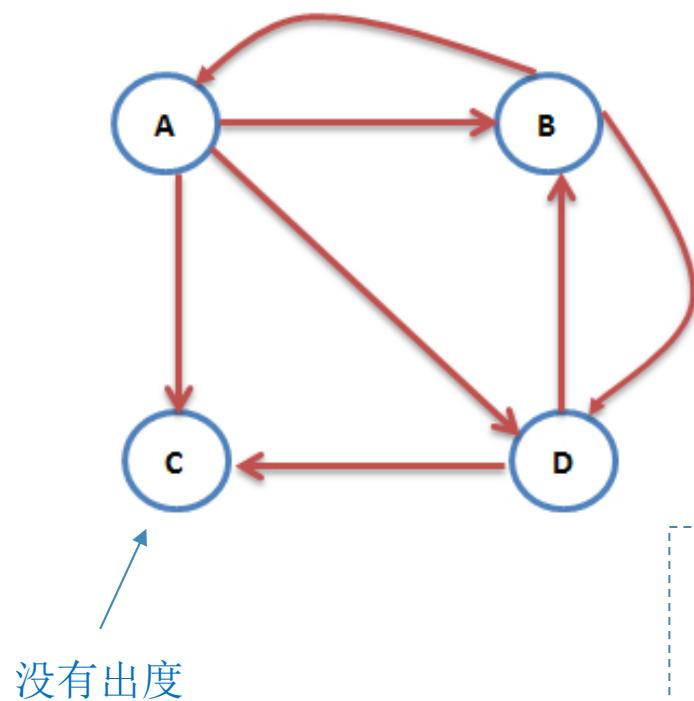
PageRank算法从原理到实现  
<https://www.cnblogs.com/rubinorth/p/5799848.html>

每一位指向A的邻居为A投票

$$\begin{aligned} PR(A) &= \frac{PR(B)}{k_{out}(B)} + \frac{PR(C)}{k_{out}(C)} \\ &= \frac{PR(B)}{2} + \frac{PR(C)}{1} \end{aligned}$$

票面价值按出度等分

# PageRank Basic Idea 2:



?

为何需要为自私节点进行自动投票?  
为了保证马尔科夫过程的收敛性：“马  
尔科夫收敛的条件之一是能从任意一个  
状态能够变到任意其他一个状态。”

$$\begin{aligned} PR(A) &= \frac{PR(B)}{k_{out}(B)} + \frac{PR(C)}{N} \\ &= \frac{PR(B)}{2} + \frac{PR(C)}{4} \end{aligned}$$

对“自私”的节点进行自动投票

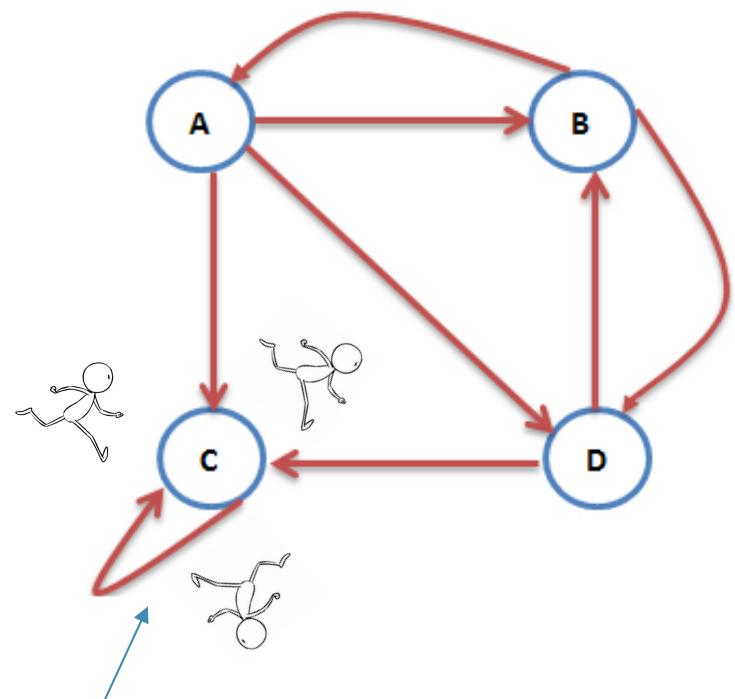
马尔科夫收敛定理

# Markov Convergence Theorem

如果满足下述四个条件，一个马尔科夫过程将收敛到一个均衡状态，且此均衡唯一：

1. 可能的状态数量是有限的。
2. 转移概率固定不变。
3. 从任意一个状态能够变到任意其他一个状态。有可能不是从状态A直接变到状态C，而是先变到状态B再变到C，但只要有路径从状态A变成状态C就行。
4. 过程不是简单循环。比如不能是从全A变到全B，然后又自动从全B变到全A。

# PageRank Basic Idea 3:



存在自指的节点，一个浏览者(random surfer)会陷入无限循环

Surfer以 $\alpha$ 的概率顺着超链接浏览网页

$$\begin{aligned} PR(A) &= \alpha \frac{PR(B)}{k_{out}(B)} + \frac{1 - \alpha}{N} \\ &= \alpha \frac{PR(B)}{2} + \frac{1 - \alpha}{4} \end{aligned}$$

Surfer以 $1 - \alpha$ 的概率随机跳转至任意网页

# PageRank Basic Idea 3:

阻尼系数, damping factor

$$PR(i) = \alpha \sum_{j \rightarrow i} \frac{PR(j)}{k_{out}(j)} + \frac{(1 - \alpha)}{N}$$

一般 $\alpha$ 设为0.85

# Calculating PageRank with Matrix

$$PR(i) = \alpha \sum_{j \rightarrow i} \frac{PR(j)}{k_{out}(j)} + \frac{(1 - \alpha)}{N}$$

网页PR值向量  $\longrightarrow \mathbf{PR}$   $= \alpha \mathbf{AD}^{-1} \mathbf{PR} + \frac{(1 - \alpha)}{N} \mathbf{1}$  全1向量

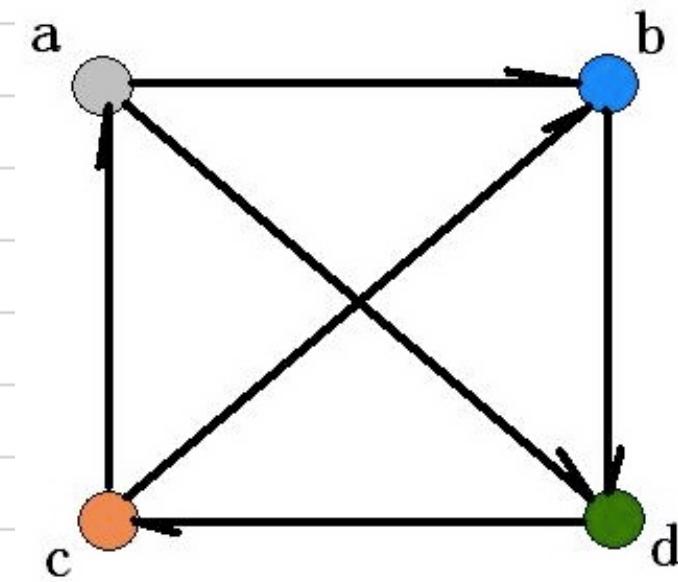
作为因子可以省略  $\longrightarrow$

$$\begin{aligned} &= \frac{(1 - \alpha)}{N} (\mathbf{I} - \alpha \mathbf{AD}^{-1})^{-1} \mathbf{1} \\ &= \mathbf{D}(\mathbf{D} - \alpha \mathbf{A})^{-1} \mathbf{1} \\ &\text{对角矩阵, 元素为 } max(k_{out}(i), 1) \end{aligned}$$

# Example

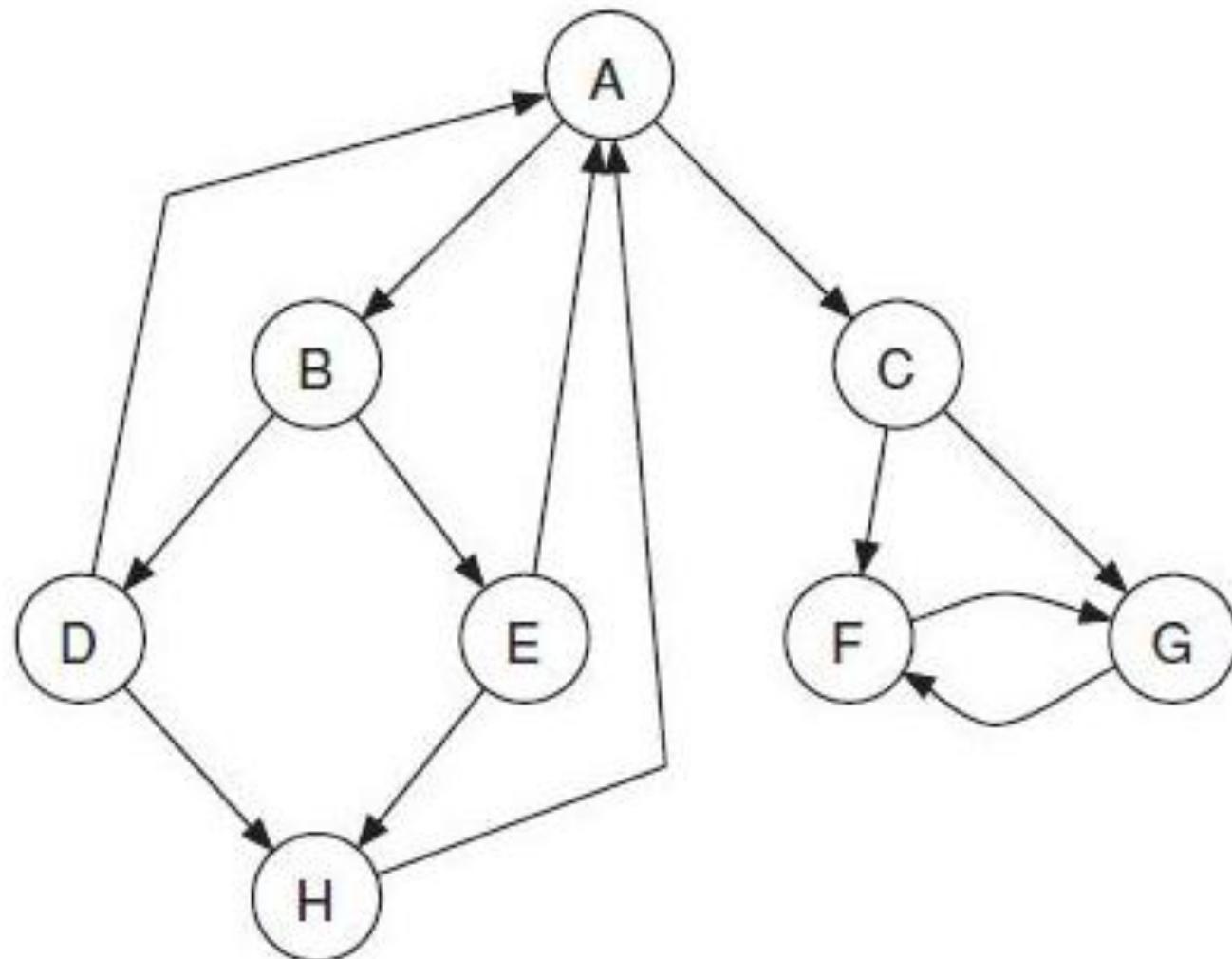
70 iterations, and finally converged :  
 $A=0.615, B=0.923, C=D=1.231$

	A	B	C	D	E
1	$a=c/2$	$b=a/2+c/2$	$c=d$	$d=a/2+b$	传递关系
2	a	b	c	d	
3	1	1	1	1	初值
4	0.5	1	1	1.5	按传递关系计算
5	0.5	0.75	1.5	1.25	
6	0.75	1	1.25	1	
7	0.625	1	1	1.375	
8	0.5	0.8125	1.375	1.3125	
9	0.6875	0.9375	1.3125	1.0625	
10	0.65625	1	1.0625	1.28125	
11	0.53125	0.859375	1.28125	1.328125	
12	0.640625	0.90625	1.328125	1.125	
13	0.6640625	0.984375	1.125	1.2265625	



# Example

Selfish nodes: F and G



## Example

F and G have all pagerank values

	A	B	C	D	E	F	G	H
62	$a=d/2+e/2+f/2$	$b=a/2$	$c=a/2$	$d=b/2$	$e=b/2$	$f=c/2+g$	$g=c/2+f$	$h=d/2+e/2$
63	a	b	c	d	e	f	g	h
64	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
65	0.25	0.0625	0.0625	0.0625	0.0625	0.0625	0.1875	0.1875
66	0.1875	0.125	0.125	0.03125	0.03125	0.21875	0.21875	0.0625
67	0.09375	0.09375	0.09375	0.0625	0.0625	0.28125	0.28125	0.03125
68	0.09375	0.046875	0.046875	0.046875	0.046875	0.328125	0.328125	0.0625
69	0.109375	0.046875	0.046875	0.0234375	0.0234375	0.3515625	0.3515625	0.046875
70	0.0703125	0.0546875	0.0546875	0.0234375	0.0234375	0.375	0.375	0.0234375
71	0.046875	0.03515625	0.03515625	0.02734375	0.02734375	0.40234375	0.40234375	0.0234375
72	0.05078125	0.0234375	0.0234375	0.01757813	0.017578125	0.41992188	0.41992188	0.02734375
73	0.04492188	0.02539063	0.02539063	0.01171875	0.01171875	0.43164063	0.43164063	0.01757813
74	0.02929688	0.02246094	0.02246094	0.01269531	0.012695313	0.44433594	0.44433594	0.01171875
75	0.02441406	0.01464844	0.01464844	0.01123047	0.011230469	0.45556641	0.45556641	0.01269531
76	0.02392578	0.01220703	0.01220703	0.00732422	0.007324219	0.46289063	0.46289063	0.01123047
77	0.01855469	0.01196289	0.01196289	0.00610352	0.006103516	0.46899414	0.46899414	0.00732422
78	0.01342773	0.00927734	0.00927734	0.00598145	0.005981445	0.47497559	0.47497559	0.00610352
79	0.01208496	0.00671387	0.00671387	0.00463867	0.00463867	0.47961426	0.47961426	0.00598145
80	0.01062012	0.00604248	0.00604248	0.00335693	0.00335693	0.48297119	0.48297119	0.00463867
81	0.00799561	0.00531006	0.00531006	0.00302124	0.00302124	0.48599243	0.48599243	0.00335693
82	0.00637817	0.0039978	0.0039978	0.00265503	0.002655029	0.48864746	0.48864746	0.00302124

# Example

阻尼系数, damping factor

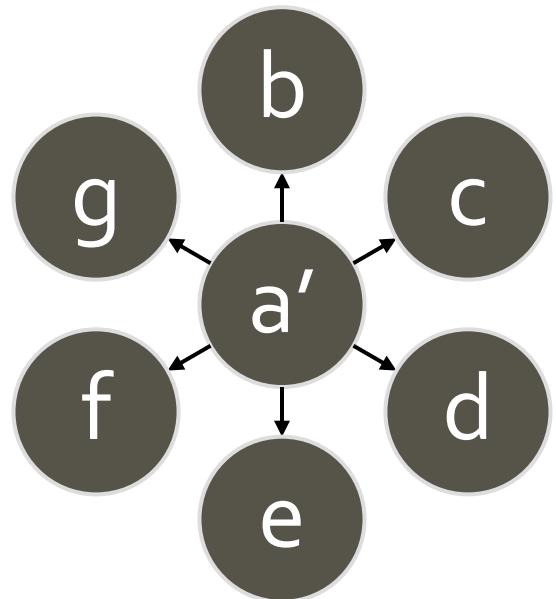
$$PR(i) = \alpha \sum_{j \rightarrow i} \frac{PR(j)}{k_{out}(j)} + \frac{(1 - \alpha)}{N}$$

一般 $\alpha$ 设为0.85

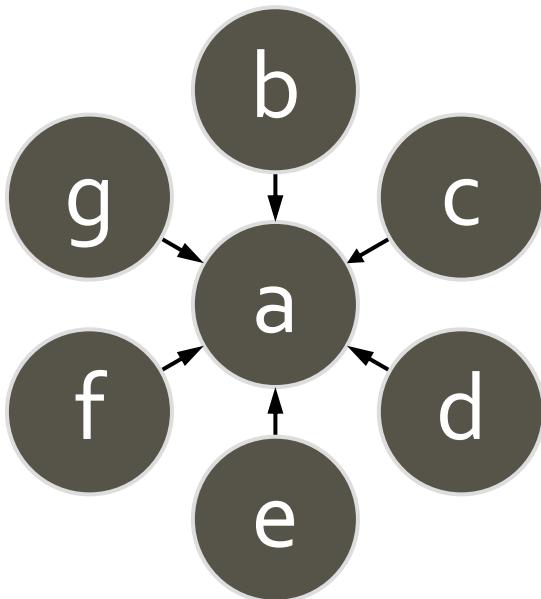
pagerank.xlsx

	A	B	C	D	E	F	G	H
62	a=(d/2+e/2)*0.8+0.2/8			取s=0.8		其他类推		
63	a	b	c	d	e	f	g	h
64	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
65	0.225	0.075	0.075	0.075	0.075	0.175	0.175	0.125
66	0.185	0.115	0.115	0.055	0.055	0.195	0.195	0.085
67	0.137	0.099	0.099	0.071	0.071	0.227	0.227	0.069
68	0.137	0.0798	0.0798	0.0646	0.0646	0.2462	0.2462	0.0818
69	0.14212	0.0798	0.0798	0.05692	0.05692	0.25388	0.25388	0.07668
70	0.13188	0.081848	0.081848	0.05692	0.05692	0.260024	0.260024	0.070536
71	0.1269648	0.077752	0.077752	0.0577392	0.0577392	0.2657584	0.2657584	0.070536
72	0.12762016	0.07578592	0.07578592	0.0561008	0.0561008	0.26870752	0.26870752	0.07119136
73	0.12683373	0.07604806	0.07604806	0.05531437	0.055314368	0.27028038	0.27028038	0.06988064
74	0.12515601	0.07573349	0.07573349	0.05541923	0.055419226	0.27164353	0.27164353	0.06925149
75	0.12473658	0.0750624	0.0750624	0.0552934	0.055293396	0.27260822	0.27260822	0.06933538
76	0.12470302	0.07489463	0.07489463	0.05502496	0.055024961	0.27311154	0.27311154	0.06923472
77	0.12440774	0.07488121	0.07488121	0.05495785	0.054957852	0.27344708	0.27344708	0.06901997
78	0.12418226	0.0747631	0.0747631	0.05495248	0.054952483	0.27371015	0.27371015	0.06896628
79	0.12413501	0.0746729	0.0746729	0.05490524	0.054905239	0.27387336	0.27387336	0.06896199
80	0.12409378	0.074654	0.074654	0.05486916	0.054869161	0.27396785	0.27396785	0.06892419
81	0.12403468	0.07463751	0.07463751	0.0548616	0.054861602	0.27403588	0.27403588	0.06889533
82	0.12400554	0.07461387	0.07461387	0.054855	0.054855005	0.27408371	0.27408371	0.06888928

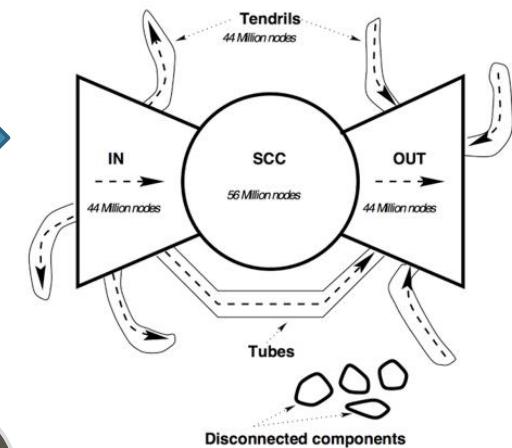
# HITS



Hub Nodes  
枢纽节点



Authoritative Nodes  
权威节点



Jon kleinberg

# HITS: Basic Idea

权威度  $\longrightarrow$   $authority(i) = \sum_{j \rightarrow i} hubness(j)$

枢纽度  $\longrightarrow$   $hubness(i) = \sum_{i \rightarrow j} authority(j)$

# Calculating HITS with Matrix

$$a(i) = \alpha \sum_j A_{ij} h(j)$$

$$h(i) = \beta \sum_{i \rightarrow j} A_{ij} a(j)$$

权威度矩阵  
枢纽度矩阵

$\Rightarrow$

$$\begin{aligned} \mathcal{A} &= \text{常数} \\ \mathcal{H} &= \text{邻接矩阵} \\ \mathcal{A} \mathcal{A}^T \mathcal{A} &= \lambda \mathcal{A} \\ \mathcal{A}^T \mathcal{A} \mathcal{H} &= \lambda \mathcal{H} \\ \mathcal{H} &= \mathcal{A}^T \mathcal{A} \end{aligned}$$

$\lambda = (\alpha \beta)^{-1}$

# Calculating HITS with Matrix

$\mathcal{A}$ 由 $\mathbf{A}\mathbf{A}^T$ 特征向量决定

$$\mathbf{A}\mathbf{A}^T \mathcal{A} = \lambda \mathcal{A}$$

$$\mathbf{A}^T \mathbf{A} \mathcal{H} = \lambda \mathcal{H}$$

$\mathcal{H}$ 由 $\mathbf{A}^T \mathbf{A}$ 特征向量决定



$\mathbf{A}\mathbf{A}^T$ 和 $\mathbf{A}^T \mathbf{A}$ 分别是什么矩阵？

# Calculating HITS with Matrix

共引矩阵

- $\mathbf{AA}^T$ : Cocitation Matrix

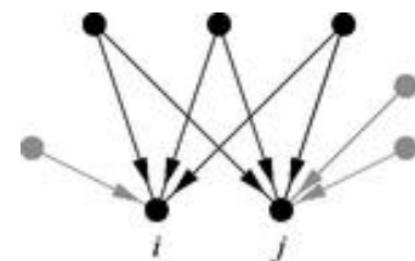
文献耦合矩阵

- $\mathbf{A}^T\mathbf{A}$ : Bibliographic Coupling Matrix

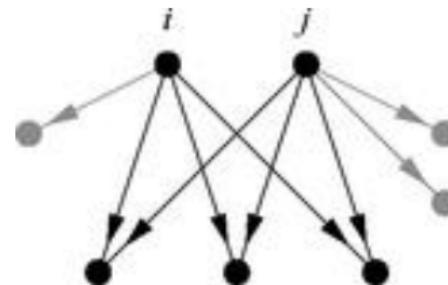
- Authority: Eigenvector Centrality of Cocitation Network

- Hubness: Eigenvector Centrality of Bibliographic Coupling Network

共引矩阵



文献耦合矩阵



# PageRank vs. HITS

PageRank从原理到实现: <http://blog.csdn.net/rubinorth/article/details/52215036>  
HITS 从原理到实现: <http://blog.csdn.net/rubinorth/article/details/52231620>

## ■ PageRank:

- query-independent 与查询无关，用于全局网页的静态排序，可离线完成计算
- topic-irrelevant 与主题无关，没有区分页面内的导航链接、广告链接和功能链接等，容易对广告页面有过高评价
- favoring old pages 旧页面等级往往比新页面高，因为新页面的链接较少

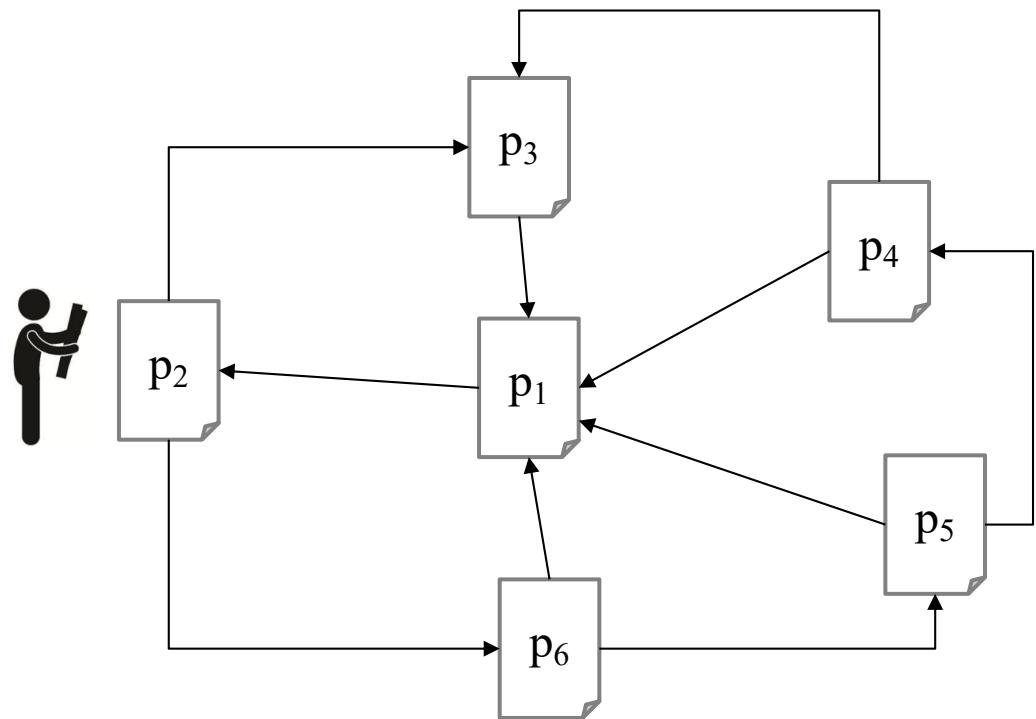
## ■ HITS:

- □ query-dependent 通常用于用户查询结果的排序，但无法用于大型搜索引擎的实时排序，并不实用
- topic drift and topic Distillation 主题漂移：扩展集合里可能包含主题无关网页，因此需要主题精化
- sensitive to spam links 思考：如何通过造假来显著提高某个网页的authority？

1. 用户查询生成根集合
2. 利用链接关系生成扩展集合

# A Probabilistic Framework of WPSS [3]

## A Single-Surfer Model



- 1. Jump ( $P(J|i)$ )
- 2. Follow ( $P(L|i)$ )
- 3. Back ( $P(B|i)$ )
- 4. Stay ( $P(S|i)$ )

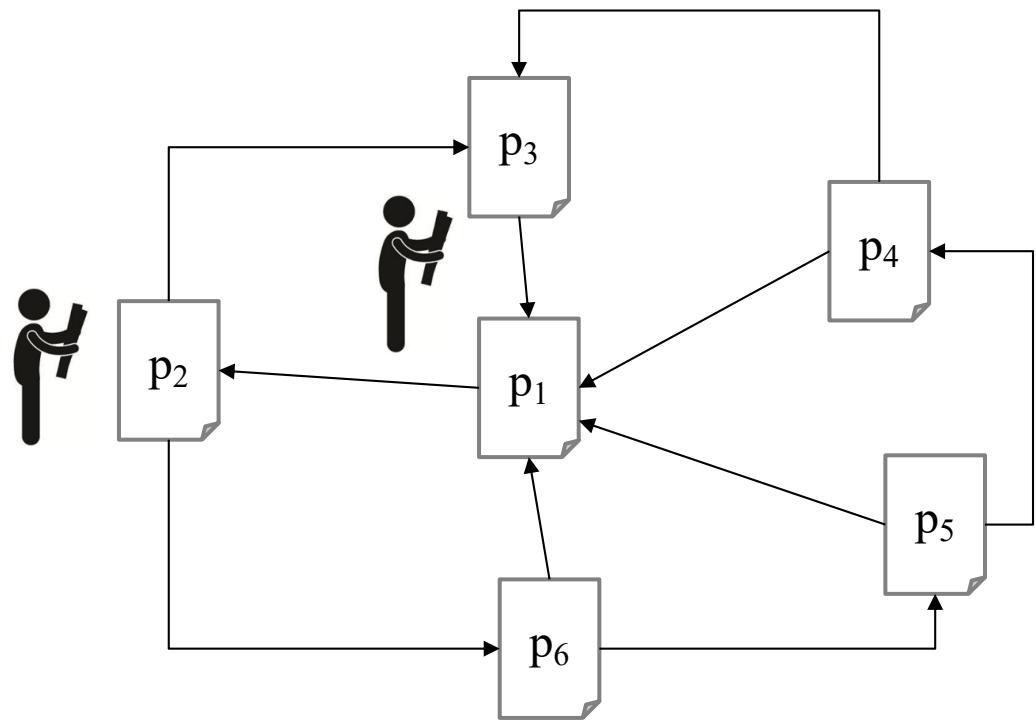
$$\sum P(J, L, B, S|i) = 1$$

$$\sum_{j \in G} P(j|J, i) = 1 \quad \sum_{i \rightarrow j} P(j|F, i) = 1 \quad \sum_{j \rightarrow i} P(j|B, i) = 1$$

The centrality of  $i$  is a **stationary distribution** of the Markov chain

$$c(i) = P^{(t)}(i)$$

# A Multi-Surfer Model



假设有 $k$ 个surfer同时浏览WWW，则 $t$ 时刻停留在 $i$ 上的概率分别为：

$$P_{s1}^{(t)}(i), P_{s2}^{(t)}(i), \dots, P_{sk}^{(t)}(i)$$

假设每个surfer会以一定的概率听取其他surfer的建议跳转至其他surfer正在浏览的网页

$$P(s_j|s_i)$$

对任意一个 $s_i$ ，该surfer对所有其他surfer的信任度之和为1

$$\sum_{j=1}^k P(s_j|s_i) = 1$$

# Reference

- [1] Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web. *World Wide Web Internet Web Inf. Syst.* 54, 1–17 (1998).
- [2] Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. *J. ACM.* 46, 604–632 (1999).
- [3] Diligenti M, Gori M, Maggini M: A unified probabilistic framework for Web page scoring systems. *IEEE Transactions on Knowledge & Data Engineering.* 16(1):4-16 (2004)

接近度中心度

# Closeness Centrality

该点到所有其他点的平均距离  $\longrightarrow l_i = \frac{1}{n} \sum_j d_{ij}$

图中节点个数

两点间最短路径长度

有时会将  $n$  改为  $(n-1)$ ,  $j$  要求不等于  $i$

$$l_i = \frac{1}{n} \sum_j d_{ij}$$

$$C_i = \frac{1}{l_i}$$

# Problems of Closeness Centrality

最大值与最小值之间跨度太小

- span a small dynamic range from largest to smallest
- in a typical network,  $c_{max}/c_{min} \leq 5$
- prone to change with a subtle change of the network structure
- has some problems dealing with disconnected components

In IMDB:

Christopher Lee in  
the Lord of Rings



$$c_{max}=0.4143$$



500,000 movie stars

Leia Zanganeh



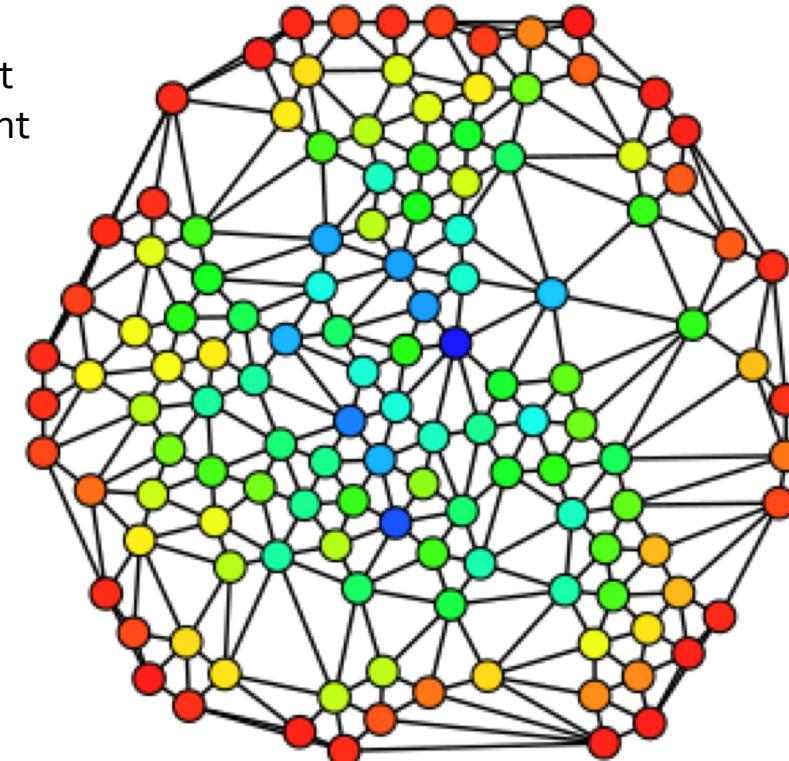
$$c_{min}=0.1154$$



介数中心度

# Betweenness Centrality

Red: least important  
Blue: most important



Considering any pair of nodes are sending messages to each other, after a long time, how many messages have passed through each node? Given that:

- 1) Nodes are communicating equally;
- 2) Messages are send through shortest path;
- 3) A random path is selected for each message if there are multiple choices.

# Betweenness: Finding the Broker

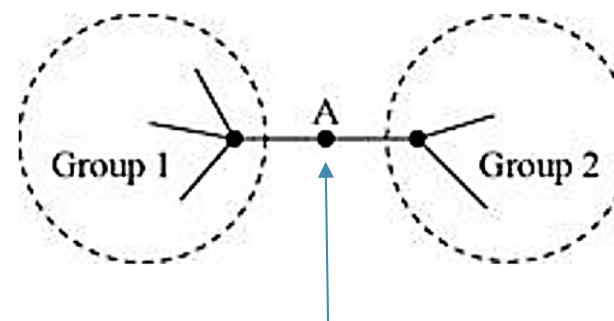
[qián]  
中间人、掮客

$$B_i = \sum_{st} \frac{n_{st}^i}{g_{st}}$$

s, t 为图中任意两个节点

$n_{st}^i$  s, t 间经过 i 的最短路径数

$g_{st}$  s, t 间的总最短路径数



割集中的节点，通常具有很低的degree，但很高的betweenness

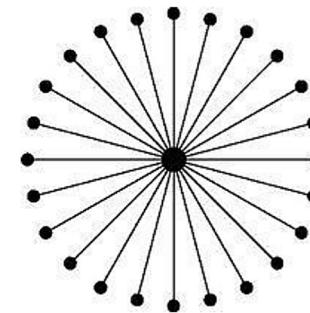
# Properties of Betweenness

最大值与最小值之间跨度很大

- span a small dynamic range from largest to smallest
- considering a star graph
- $B_{max}/B_{min} = \frac{n^2-n+1}{2n-1} \approx \frac{n}{2}$
- stable with the change of network structure

星形图中心点的介数中心度

星形图边缘点的介数中心度



# Example of Betweenness

1<sup>st</sup>:  $7.47 \times 10^8$



In IMDB:

Fernando Rey

2<sup>nd</sup>:  $6.49 \times 10^8$



Christopher Lee

least:  $8.91 \times 10^5$



...

# 0-1 Normalization of Betweenness

$$B_i = \frac{1}{n^2} \sum_{st} \frac{n_{st}^i}{g_{st}}$$

$$B_i \in [0,1]$$

流介数

# Flow Betweenness

*What if the flow of information is **not always** go through the **shortest path**?*

*For Example,*

*A confirmation of a break-up with you is **not directly** from your girl,  
**but indirectly** from her roommate...*



# Calculating Flow Betweenness

$$B_i = \frac{1}{n^2} \sum_{st} \frac{n_{st}^i}{g_{st}}$$

s,t之间的独立路径数

# Random-walk Betweenness

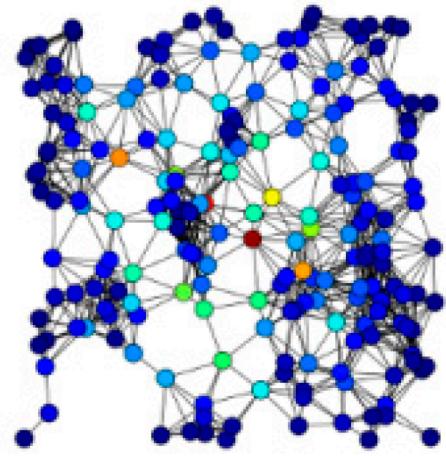
- A variant of betweenness
- the flow is a random walk from  $s$  to  $t$

$$B_i = \sum_{s,t} n_{st}^i \quad \text{从}s\text{到}t\text{多次游走经过}i\text{的平均值}$$

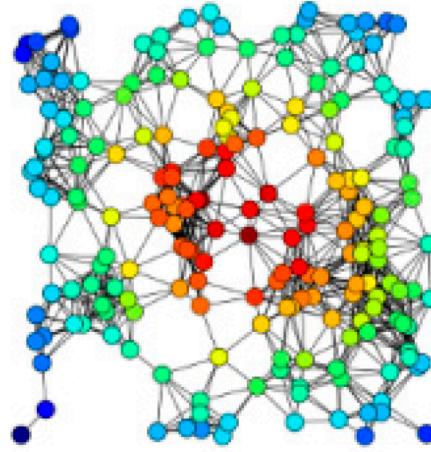
# A Faster Algorithm for Betweenness Centrality

- $O(mn)$  for unweighted graph
- $O(mn + n\log n)$  for weighted graph
- close to  $O(n^2)$  for sparse graph

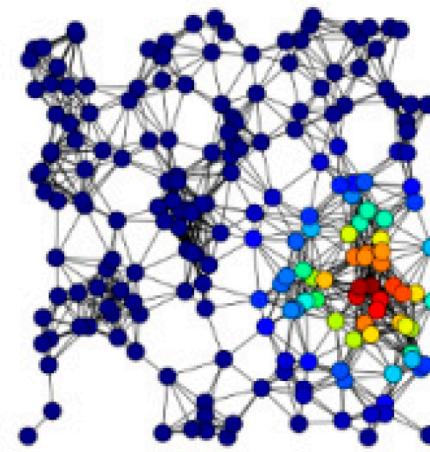
Brandes, U.: A faster algorithm for betweenness centrality. J. Math. Sociol. 25, 163–177 (2001).



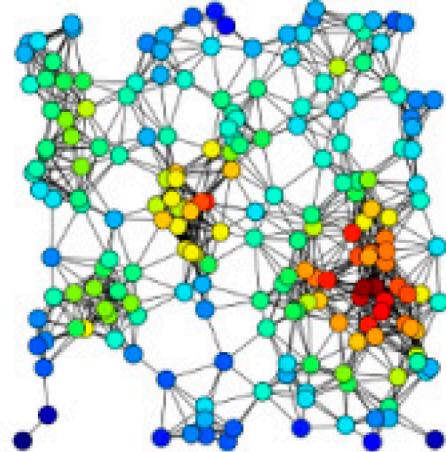
betweenness centrality  
**(a)**



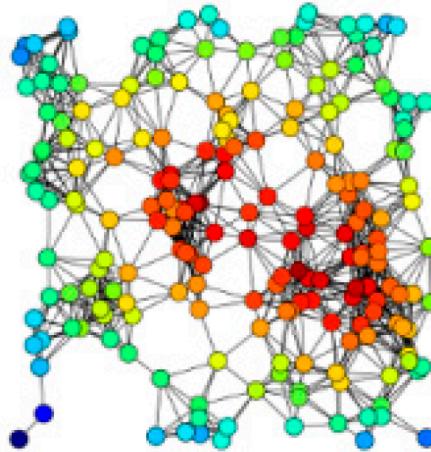
closeness centrality  
**(b)**



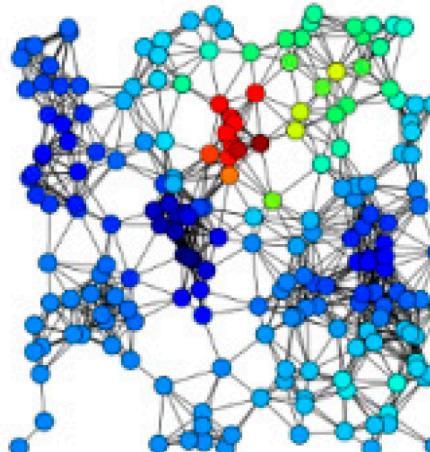
eigenvector centrality  
**(c)**



degree centrality



harmonic centrality



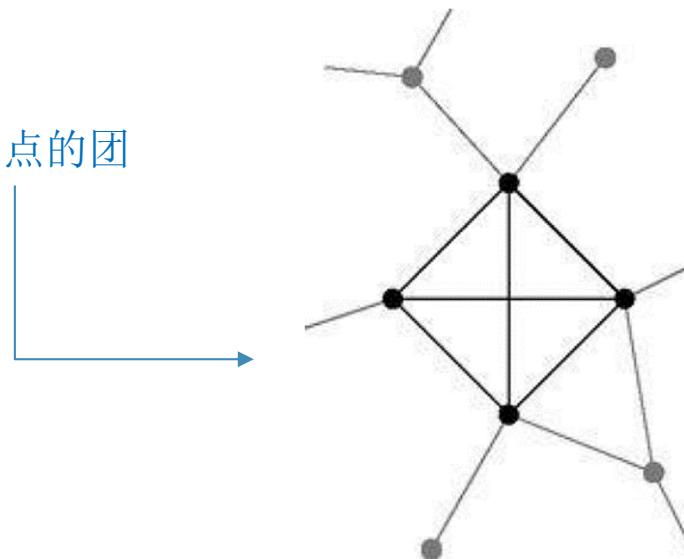
Katz centrality

团  
Cliques

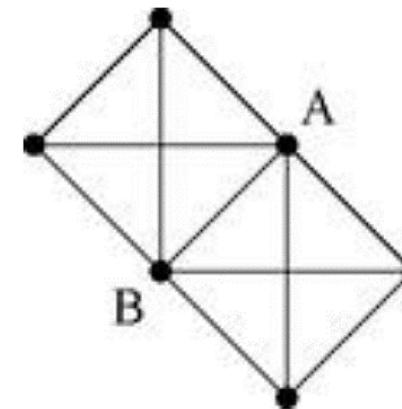
1. undirected graph
2. maximal subset of the vertices
3. every member of the set is connected by an edge to every other

无向图中的最大完全子图

4个顶点的团

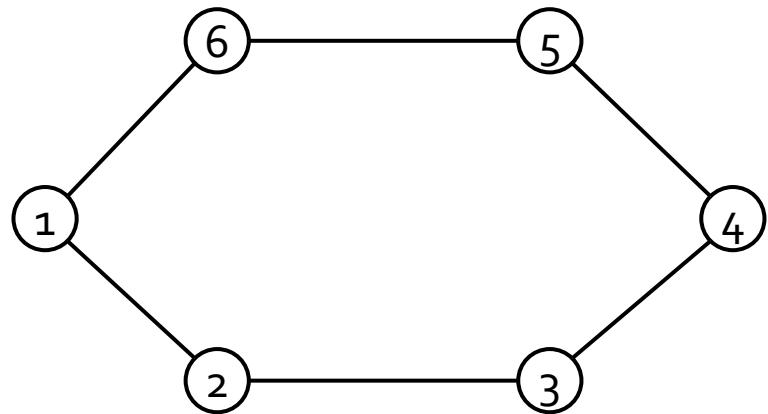


两个相互重叠的团



# k-Plex

- Clique is too rigid
- $k$ -Plex: near cliques, not perfect cliques
  - maximal subset of the vertices
  - A  $k$ -Plex of size  $n$  is
    - a maximal subset of  $n$  vertices
    - each vertex is connected to at least  $n - k$  of the others
    - 1-plex = clique



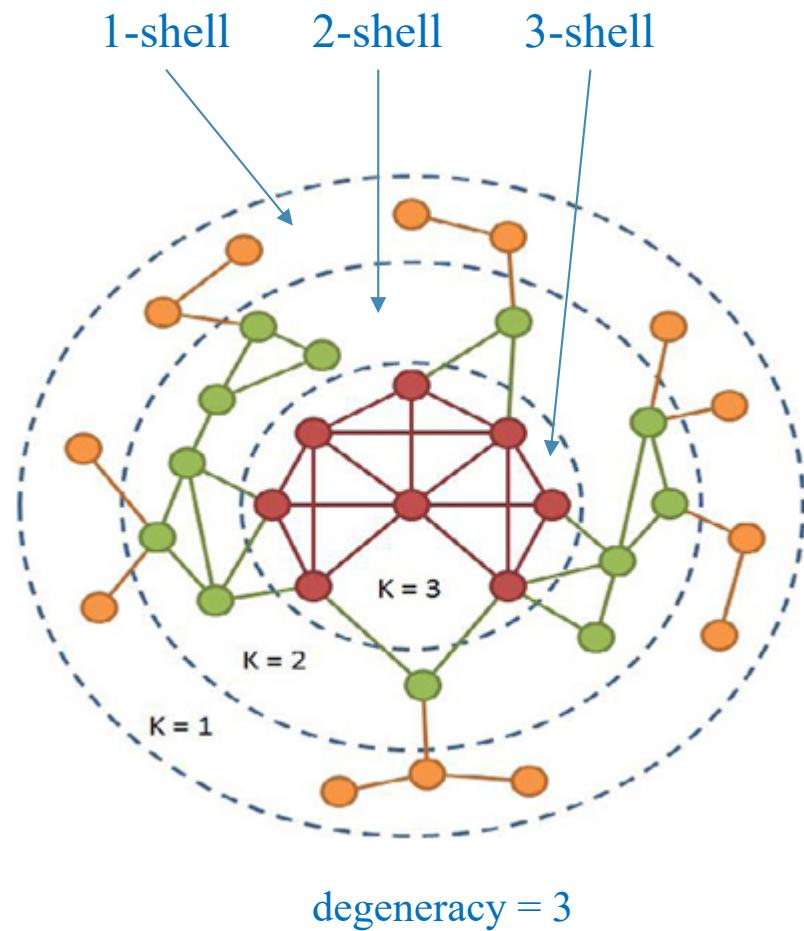
?

1-plexes  
How many 2-plexes in this graph  
3-plexes  
4-plexes

$k$ 核

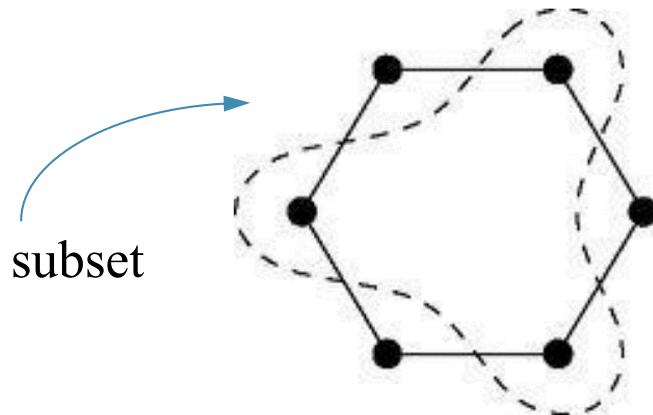
# $k$ -Core and $k$ -Shell

1. maximal subset of vertices
2. each is connected to at least  $k$  others in the subset  
简并度, 用于衡量图的稀疏程度
3. **Degeneracy**: highest  $k$  such that there exists a  $k$ -core in the graph



k团  
 $k$ -Clique

1. maximal subset of vertices
2. each is connected to at least  $k$  others in the subset  
每点与其他点最大距离不超过 $k$ 步
3. each is no more than a distance  $k$  away from any of the others
4. 1-clique = clique  $\leftarrow$   $k$ -clique是对1-clique放宽松了条件
5. There is an issue in the definition:  
一个 $k$ -clique中的节点之间的边可能会涉及该团之外的节点  
a  $k$ -clique need not be connected via paths that run within the subset



虚线画出了一个2-clique,  
但这个团中并没有使其连通的路径

# ASSIGNMENT

---



# Web Analysis and Ranking Algorithms

## ■ Goal

- (mandatory) Network measurements: Centralities
- (mandatory) Baseline: PageRank and HIST, (implement, experiments, and analysis)
- (optional) Topic-sensitive PageRank [1]
- (optional) SALSA and TKC effect [2]
- (optional) TrustRank [3]
- (optional) Dynamic Personalized PageRank [4]

## ■ Datasets

- Google Web graph: <http://snap.stanford.edu/data/web-Google.html>
- Social Networks: <http://snap.stanford.edu/data/ego-Facebook.html>
- Others: SNAP <http://snap.stanford.edu/data/index.html>

# Web Ranking Algorithms

- Paper writing: a team paper can be collaborated written including:

- abstract / problem definition / approach overview
  - details of algorithms / experiments / related works / reference
  - LNCS template with latex
  - PPT

- References

- [1] Haveliwala, T.H.: Topic-sensitive PageRank. In: Proceedings of the eleventh international conference on World Wide Web - WWW2002. p. 517 (2002).
- [2] Lempel, R., Moran, S.: Stochastic approach for link-structure analysis (SALSA) and the TKC effect. Comput. Networks. 33, 387–401 (2000).
- [3] Gyöngyi, Z., Garcia-Molina, H., Pedersen, J.: Combating web spam with TrustRank. Proc. Thirtieth Int. Conf. Very large data bases. 30, 576–587 (2004).
- [4] Chakrabarti, S.: Dynamic personalized pagerank in entity-relation graphs. WWW2007. (2007).

# (Optional) Reading

- Network embedding
- Knowledge graph embedding