



北京郵電大學

BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS

计算机应用编程 实验三

图分析器

熊永平@网络技术研究院

ypxiong@bupt.edu.cn

周一10:10-12:00@3-134

2015.11.23

告诉我，你来了



课程表

学 期

二〇一五年 秋 季 学

寒 假

年份	二〇一五年												二〇一六年														
月份	九月			十月			十一月			十二月			一月			二月											
周次	〇	一	二	三	四	五	六	七	八	九	十	十一	十二	十三	十四	十五	十六	十七	十八	十九	二十	廿一	廿二	廿三	廿四	廿五	
星期一	31	7	研究生新生上课	21				12	19	26	2	9	16	23	30	7	14	21	28	4	11	18	25	1	春节	15	22
星期二	1	本科生新生报到		5	20	27	国庆节假期	13	20	27	3	10	17	24	1	8	15	22	29	5	12	19	26	2	9	16	23
星期三	2			6	23	30		14	21	28	4	11	18	25	2	9	16	23	30	6	13	20	27	3	10	17	24
星期四	3	10*		7	24		8		22	29	5	12	19	26	3	10	17	24	31	7	14	21	28	4	11	18	25
星期五	4	研究生开学典礼		8	25		9		23	30	6	13	20	27	4	11	18	25	元旦	8	15	22	29	5	12	19	26
星期六	5	12		9	26		10		24	31	7	14	21	28	5	12	19	26	9	16	23	30	6	13	20	27	
星期日	6	1		10			11		25	1	8	15	22		6	13	20	27	9	17	24	31	7	14	21	28	

注：以党政办通知为准

课程介绍

实验一讨论课

实验二讨论课

实验三讨论课

实验四讨论结束

实验三：图分析器

实验目标

➤ 目标

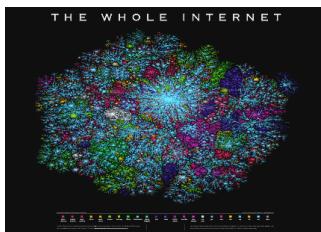
- 设计一个链接分析程序，实现
- 构建16W个节点的网页链接关系图
- 寻找图中的存在的自环
- 利用PageRank算法计算重要度最高的前10个页面

➤ 编程技能

- C++语言练习
- 图分析
- 随机过程实现
- 稀疏矩阵
- PageRank算法

世界是相互联系的

Communication



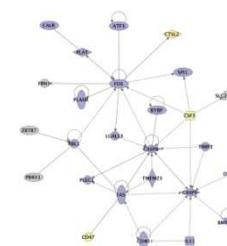
Social



Transportation



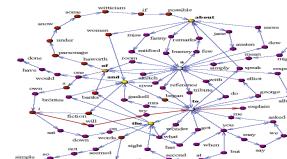
Biological



Power Grid

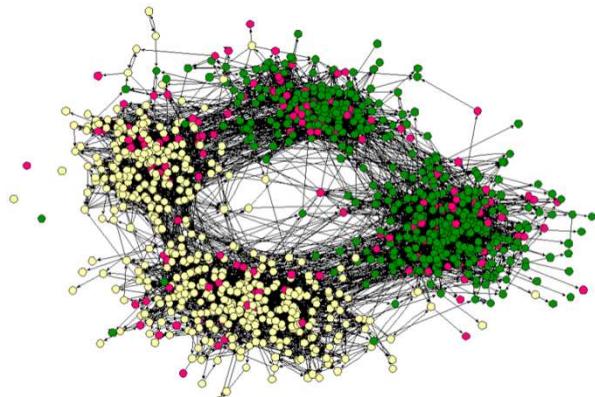


Language

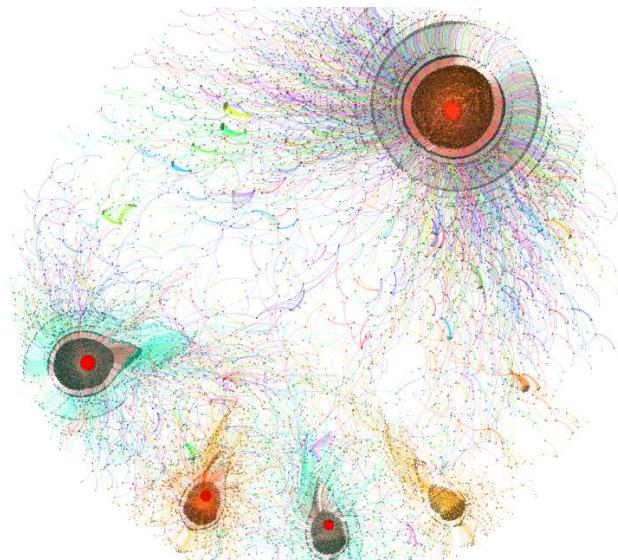
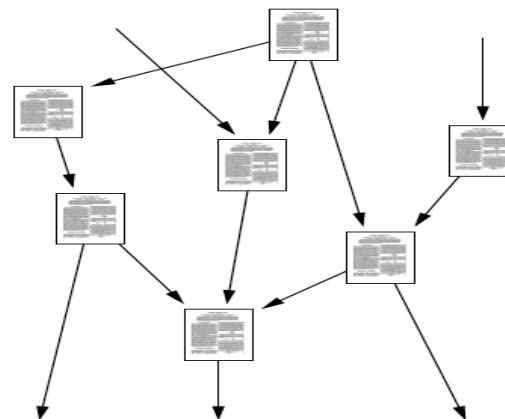


真实网络（1）：社会网络

朋友关系网



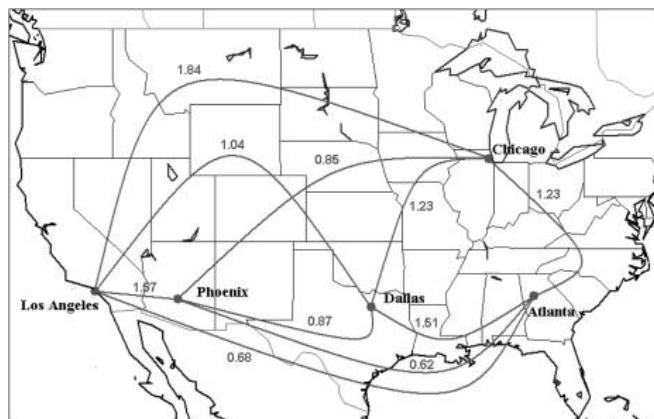
科学引文网



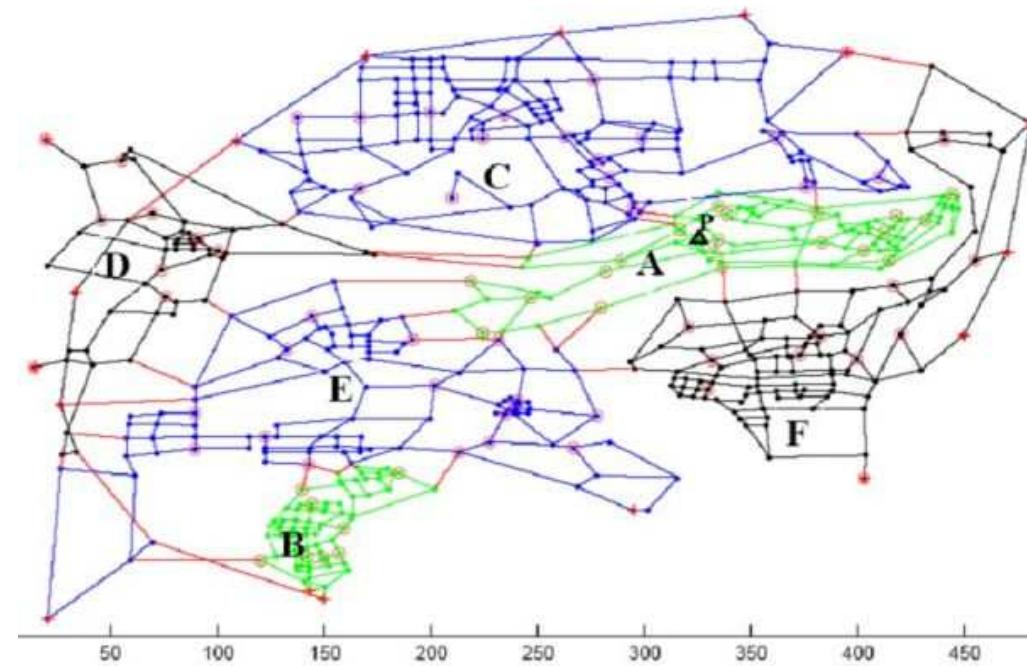
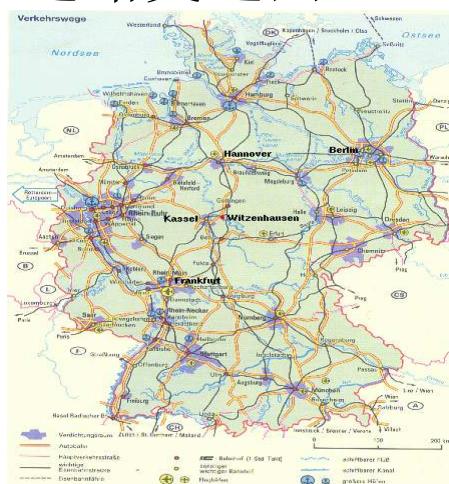
✓ 社会网：演员合作网，
友谊网，姻亲关系网，科
研合作网，Email网

真实网络（2）：交通网络

航空网

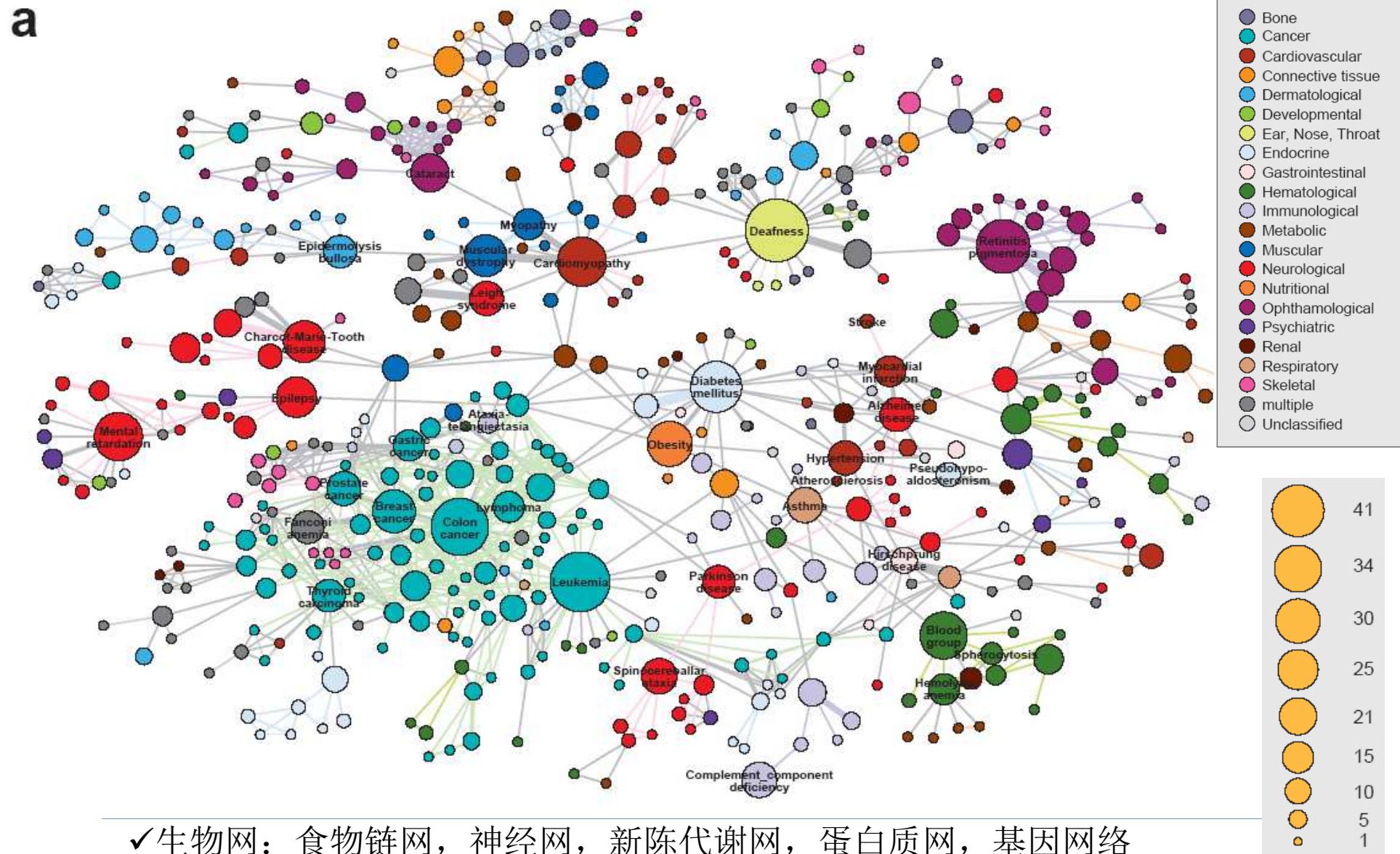


道路交通网

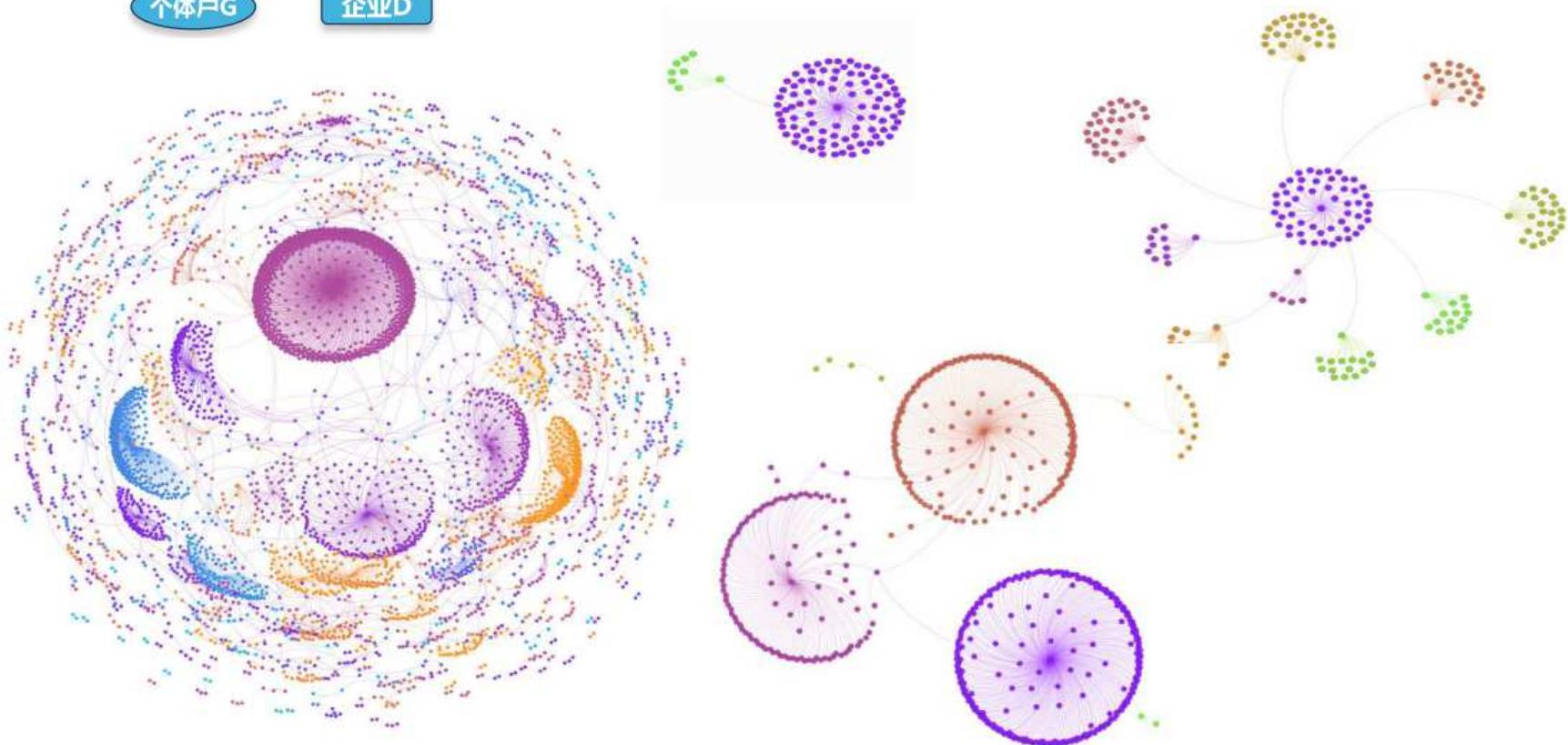
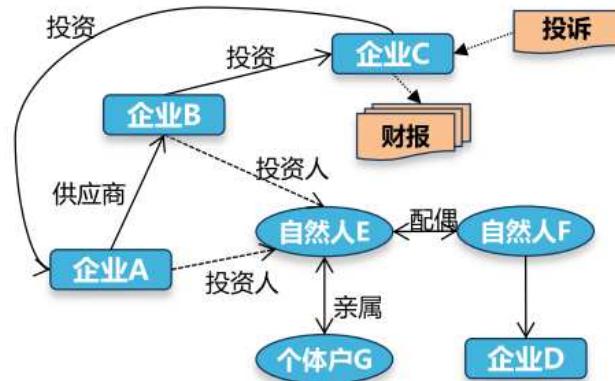


✓ 交通运输网：航线网，铁路网，公路网，
自然河流网

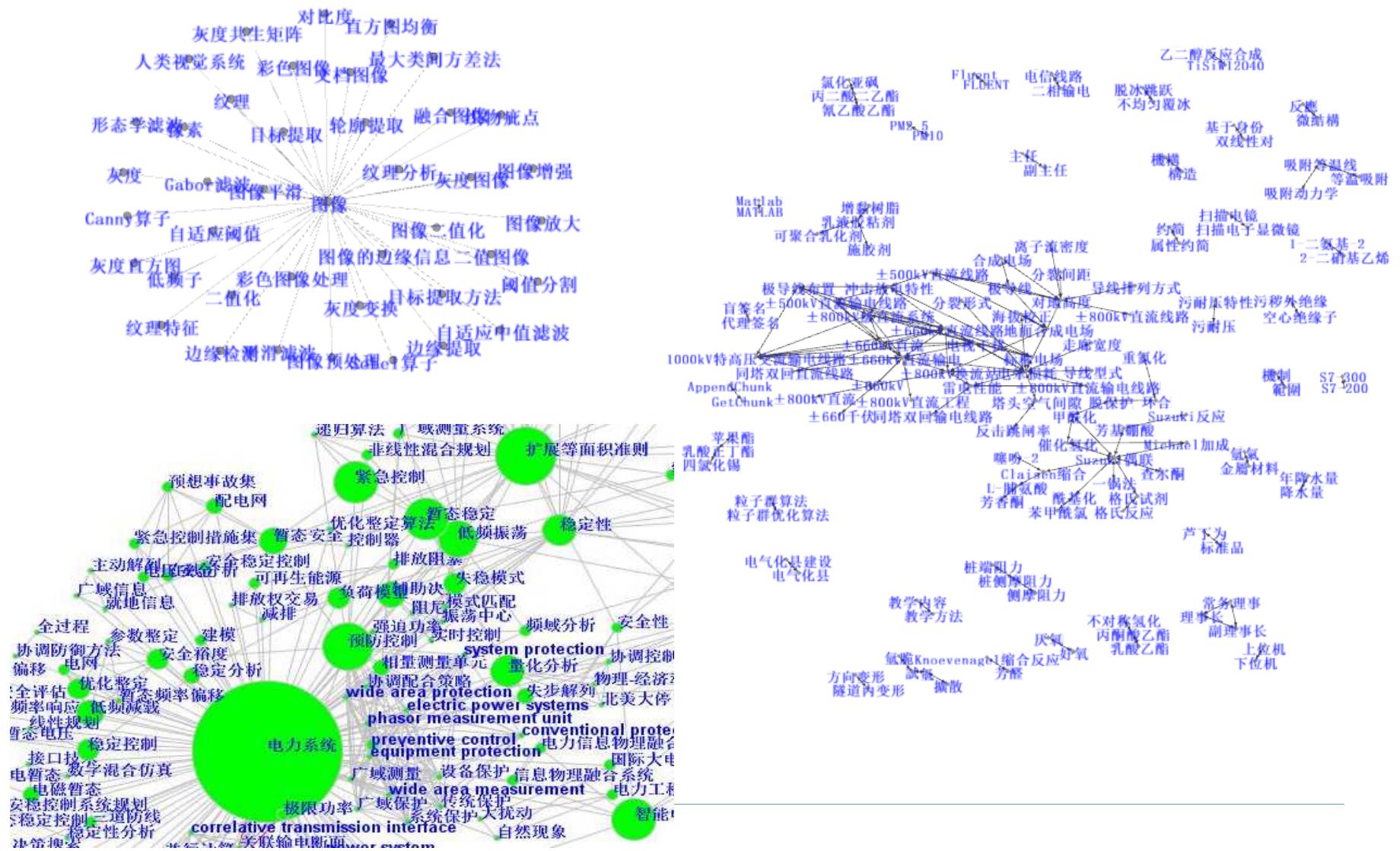
真实网络（3）：生物网络



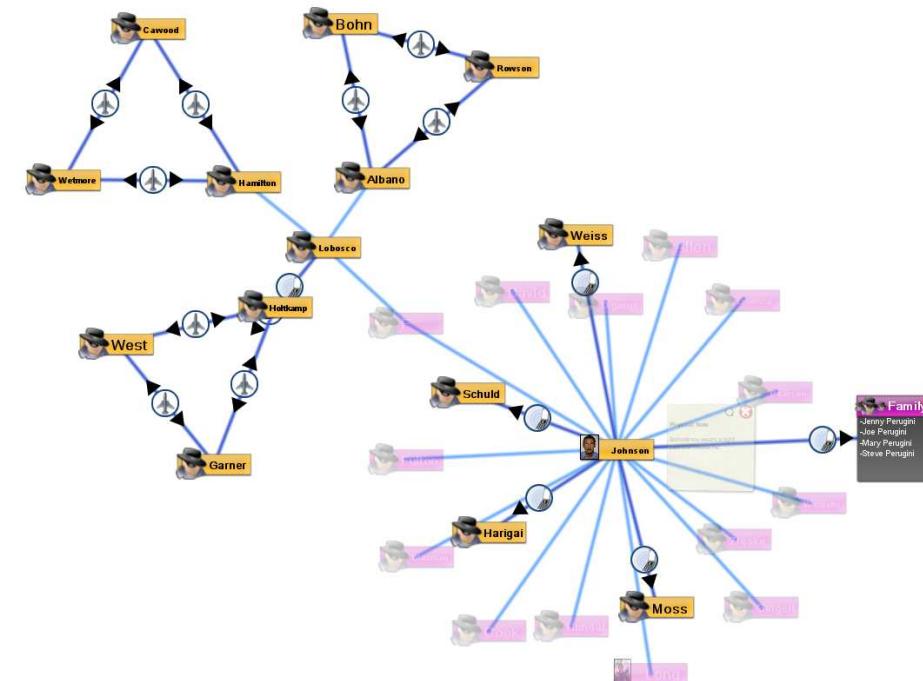
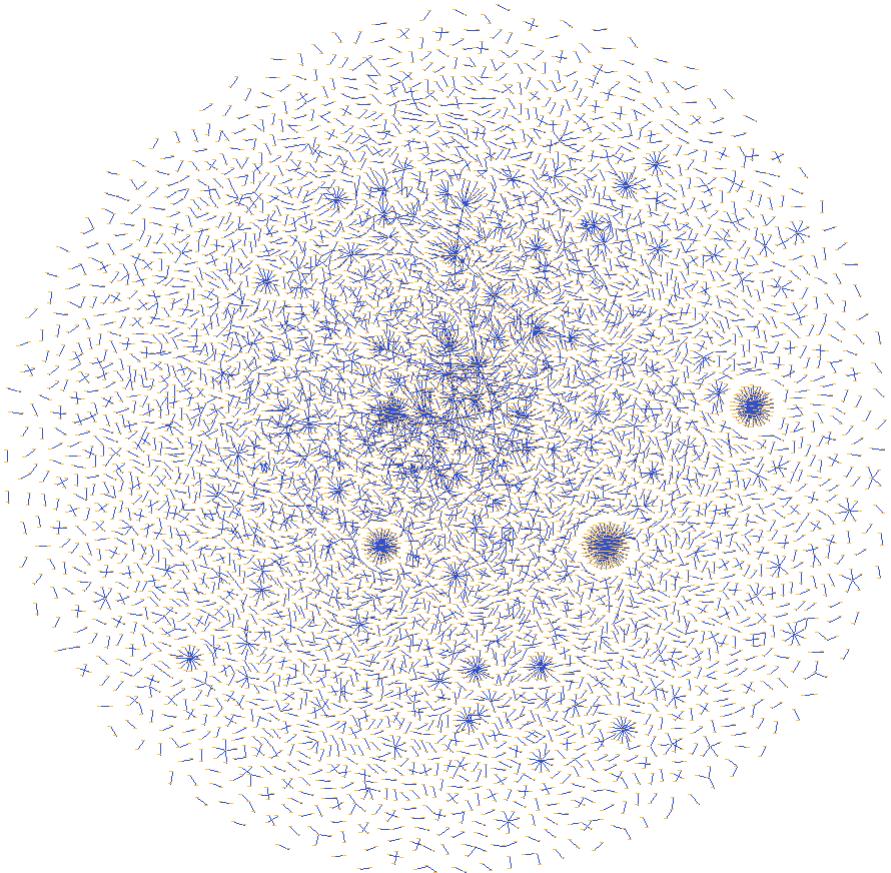
真实网络（4）：企业关联网络



真实网络（5）：关键词关联网络

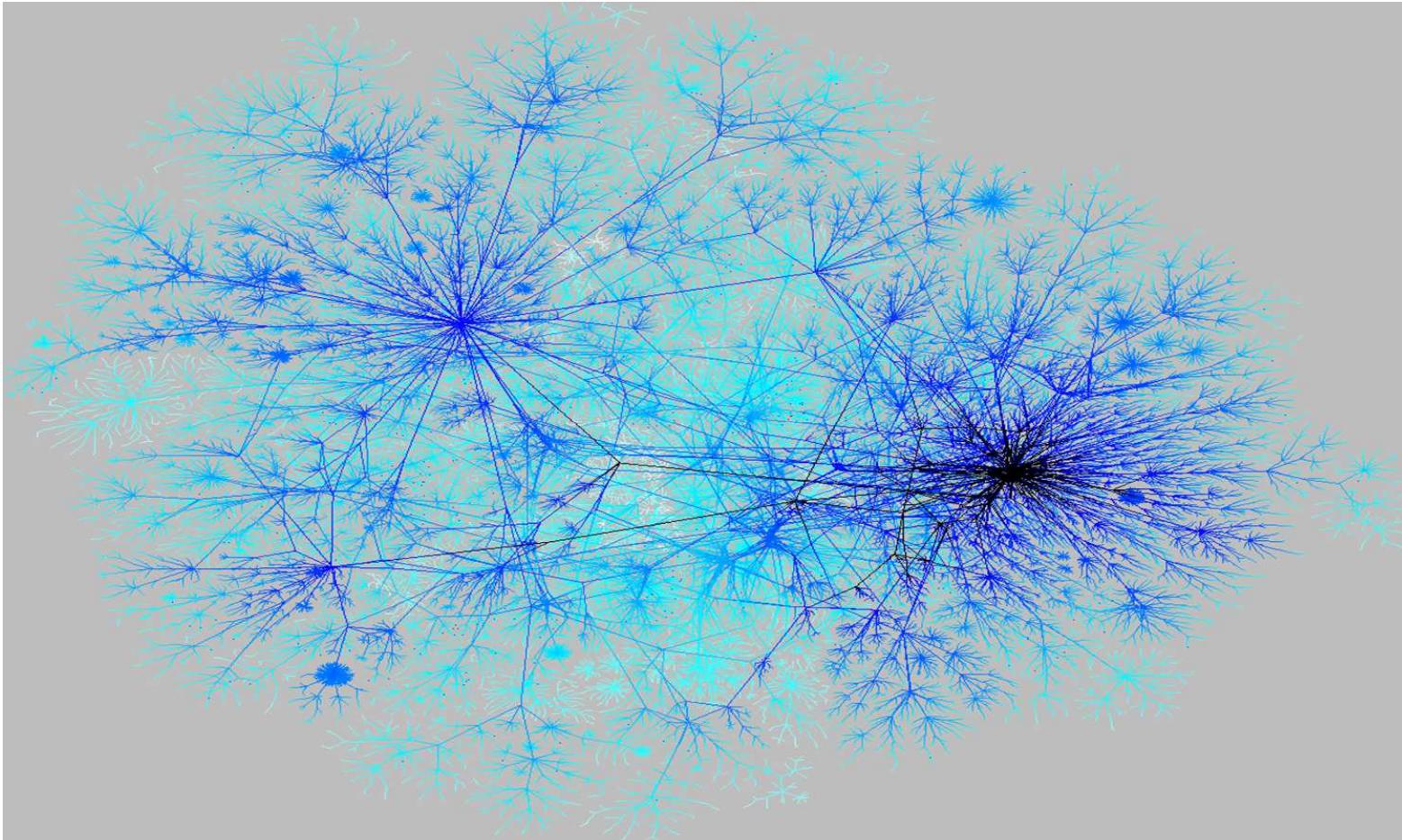


真实网络（6）：反恐网络



Palantir

真实网络（7）：Web网络



网络和图的关系

network often refers to real systems

- www,
- social network
- metabolic network.

Language: (Network, node, link)

graph: mathematical representation of a network

- web graph,
- social graph (a Facebook term)

Language: (Graph, vertex, edge)

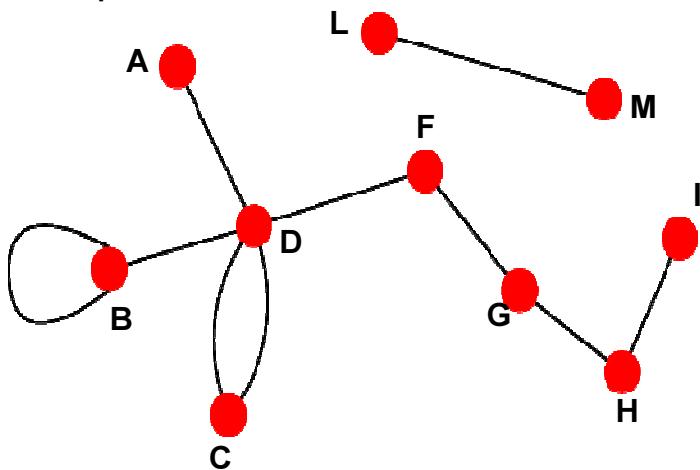
We will try to make this distinction whenever it is appropriate, but in most cases we will use the two terms **interchangeably**.

(大部分场合，我们互用网络和图这两个概念)

Undirected (无向网络)

Links: undirected (*symmetrical*, 对称关系)

Graph:



Undirected links :

合作研究网络

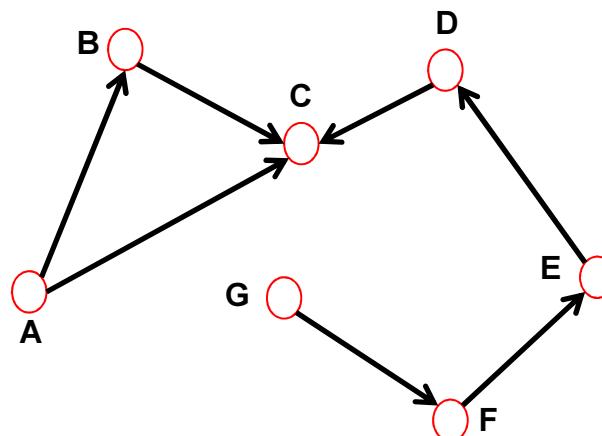
Actor network

protein interactions

Directed 有向网络

Links: directed (*arcs*).

Digraph = directed graph:



An undirected link is the superposition of two opposite directed links.

Directed links :

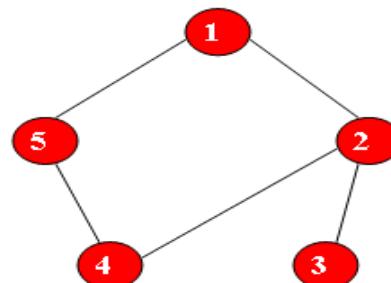
www网络URL链接

电话呼叫

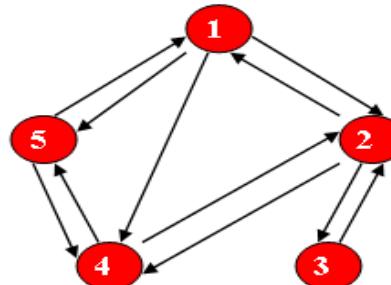
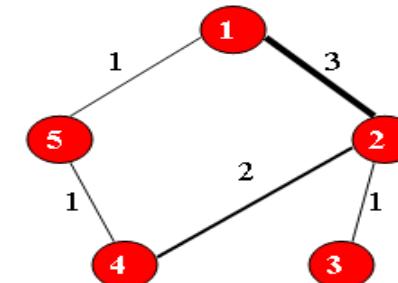
metabolic reactions(代谢反应)

网络分类

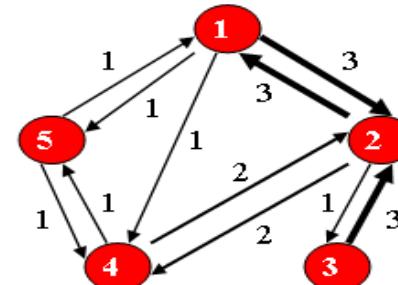
Unweighted Undirected



Weighted Undirected



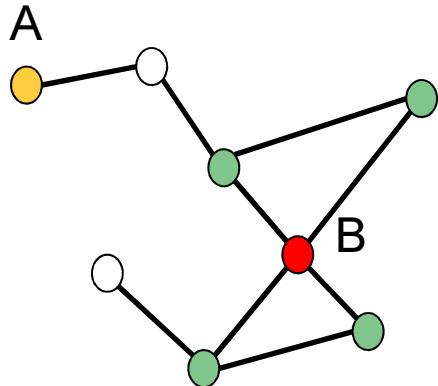
Unweighted Directed



Weighted Directed

节点的度

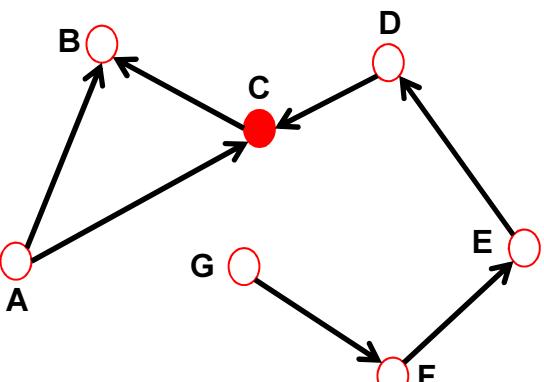
Undirected



Node degree: the number of links connected to the node.

$$k_A = 1 \quad k_B = 4$$

Directed



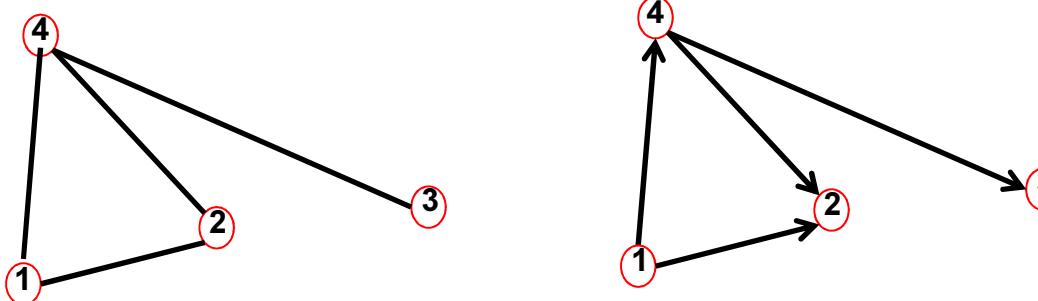
In *directed networks* we can define an **in-degree** and **out-degree**.

The (total) degree is the sum of in- and out-degree.

$$k_C^{in} = 2 \quad k_C^{out} = 1 \quad k_C = 3$$

Source: a node with $k^{in} = 0$; **Sink**: a node with $k^{out} = 0$.

网络表示形式—邻接矩阵



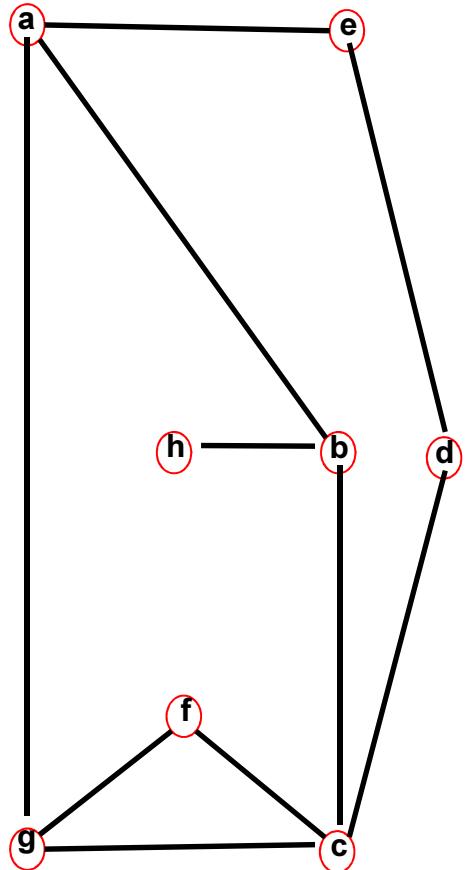
$A_{ij} = 1$ if there is a link between node i and j

$A_{ij} = 0$ if nodes i and j are not connected to each other.

$$A_{ij} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \quad A_{ij} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

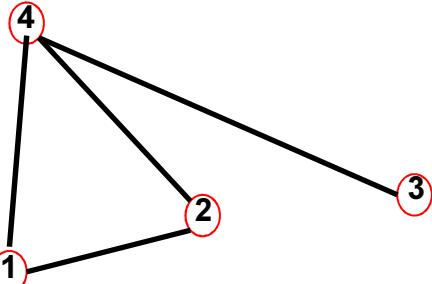
Note that for a directed graph (right) the matrix is **not symmetric**.

邻接矩阵



	a	b	c	d	e	f	g	h
a	0	1	0	0	1	0	1	0
b	1	0	1	0	0	0	0	1
c	0	1	0	1	0	1	1	0
d	0	0	1	0	1	0	0	0
e	1	0	0	1	0	0	0	0
f	0	0	1	0	0	0	1	0
g	1	0	1	0	0	0	0	0
h	0	1	0	0	0	0	0	0

Undirected



$$A_{ij} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

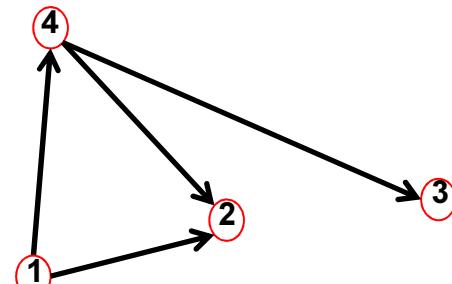
$$A_{ij} = A_{ji}$$

$$A_{ii} = 0$$

$$k_i = \sum_{j=1}^N A_{ij}$$

$$k_j = \sum_{i=1}^N A_{ij}$$

Directed



$$A_{ij} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

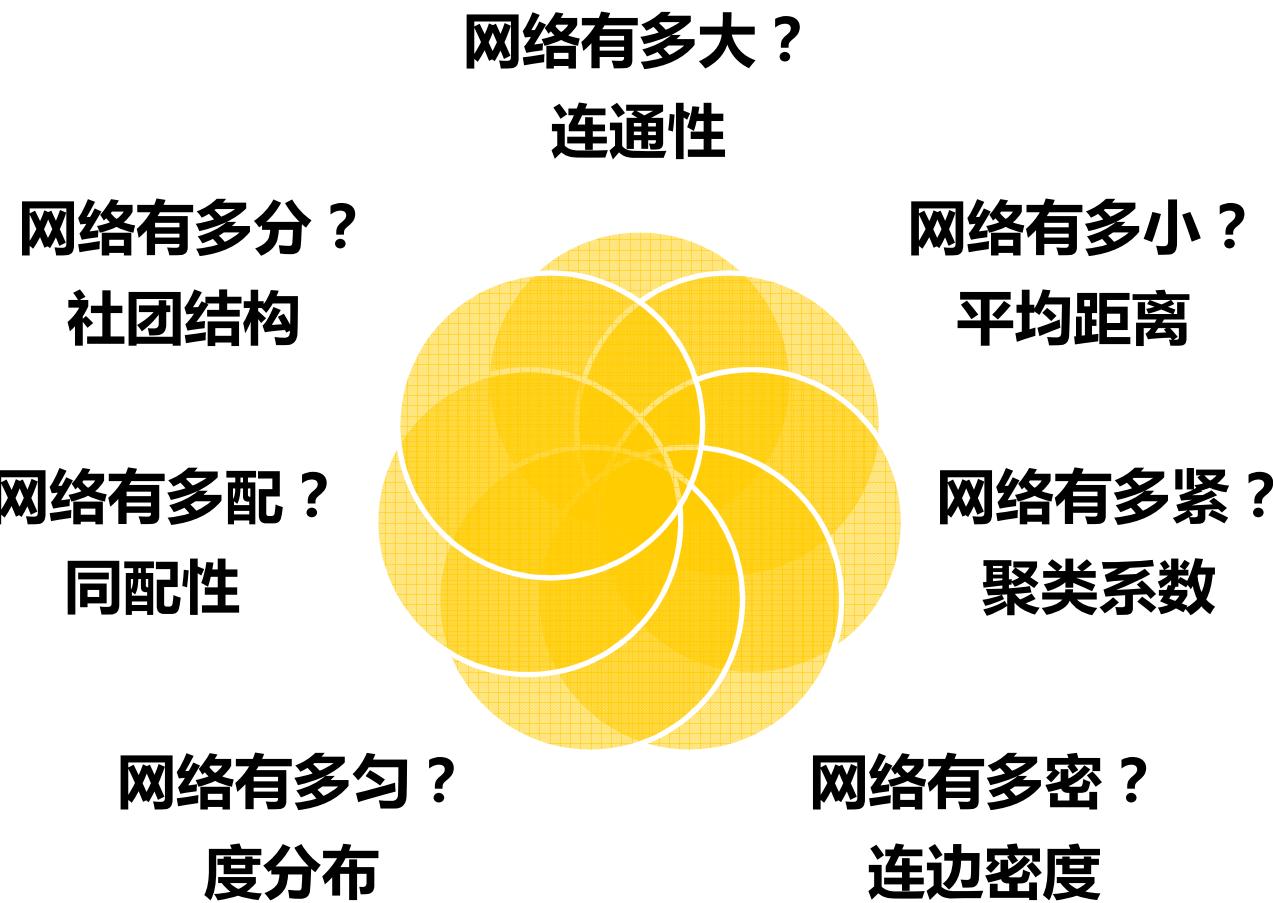
$$\nabla \sum_{i=1}^N = \nabla$$

$$k_i^{out} = \sum_{i=1}^N A_{ij}$$

$$A_{ij} \neq A_{ji}$$

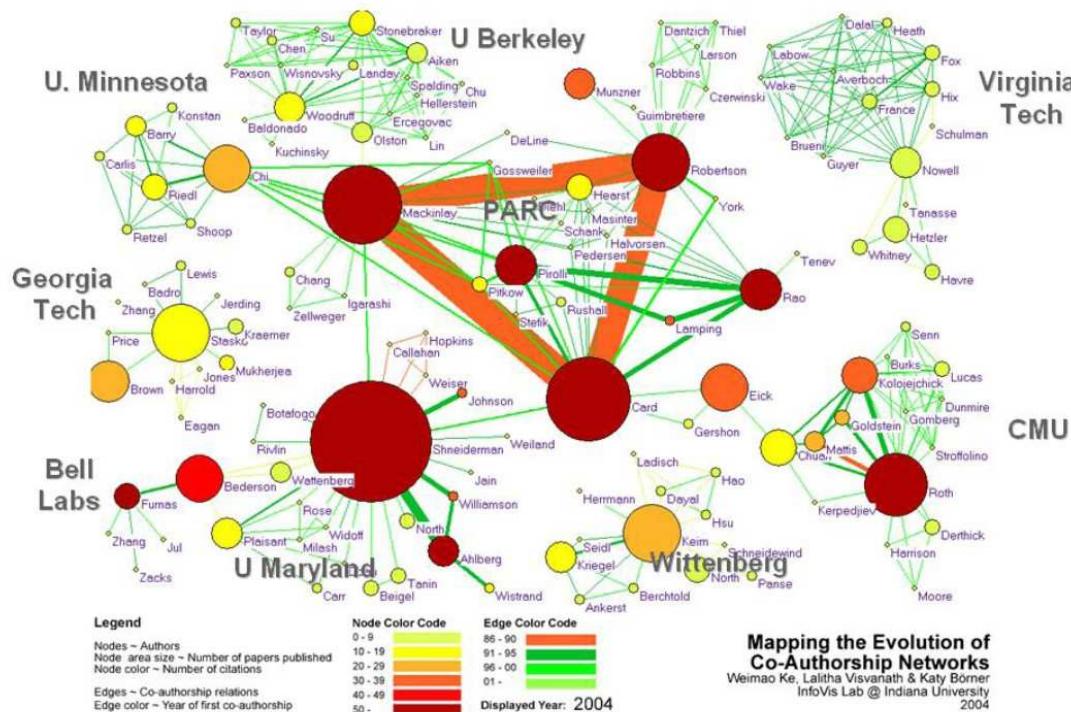
$$A_{ii} = 0$$

网络科学问题



网络节点重要性：个体影响力

Mapping the Evolution of Co-Authorship Networks
Ke, Viswanath & Bornér, (2004) Won 1st prize at the IEEE InfoVis Contest.



中心性测量

- Degree Centrality
- Eigenvector Centrality
- Katz Centrality
- Closeness Centrality
- Betweenness Centrality
- Transitivity
- PageRank

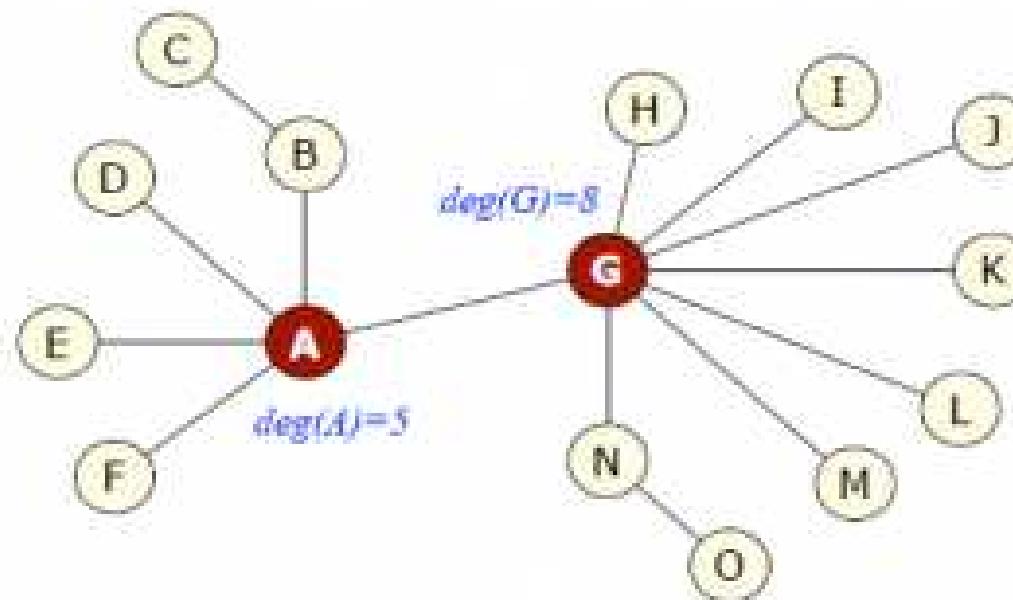
发现名人：节点度(degree centrality)

节点度是指和该节点相关联的边的条数。

特别地，对于有向图，

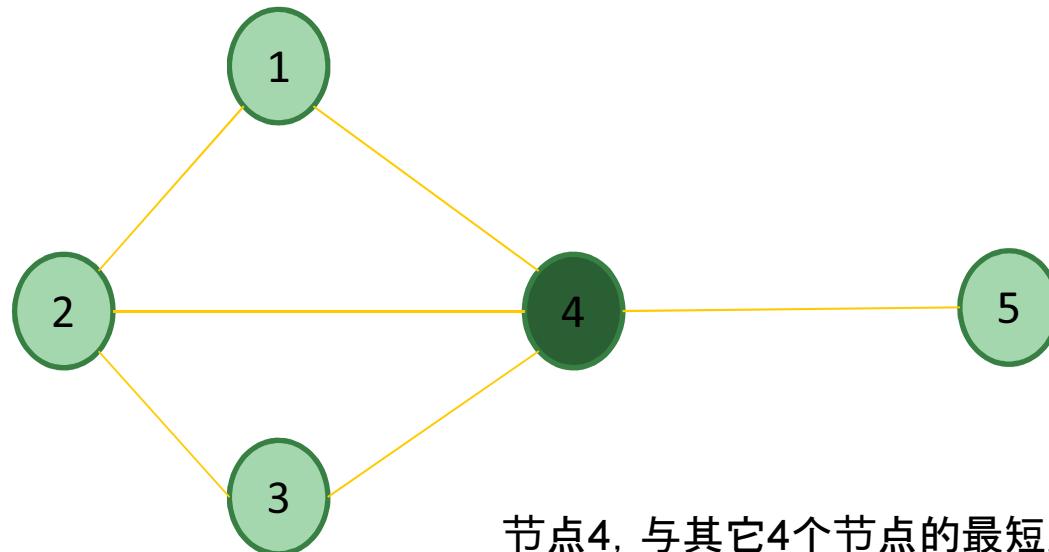
节点的入度 是指进入该节点的边的条数；

节点的出度是指从该节点出发的边的条数。



发现八卦传播者 ——接近中心性(closeness centrality)

如果一个点与网络中所有其它点的距离都很短，则该点是整体中心点。
在图中，这样的点与许多其它点都“接近”

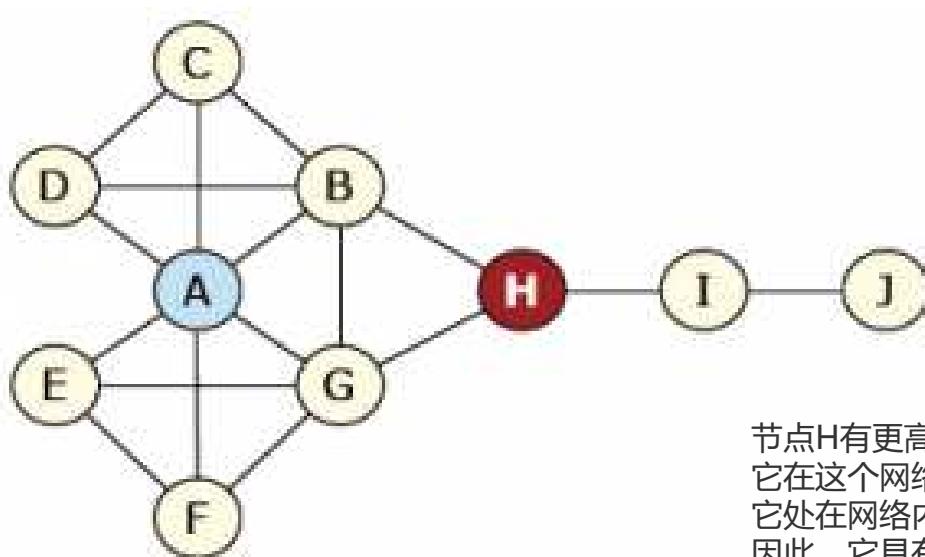


节点4，与其它4个节点的最短距离为1，和为4，节点4的接近中心度为 $1/4$

节点2，与3个节点的最短距离为1，与节点5的最短距离为2，和5，节点2的接近中心度为 $1/5$

发现社群桥梁 —— 中介中心性(betweenness centrality)

中介中心性指出现在许多其他节点间最短路径上的节点有较高的中介中心性分数。



节点H有更高的中介中心性，
它在这个网络中扮演经纪人的角色，
它处在网络内许多节点交往的路径上，
因此，它具有控制其他人交往的能力。

Betweeness Centrality

发现幕后高手 —— 特征向量中心性(eigenvector centrality)

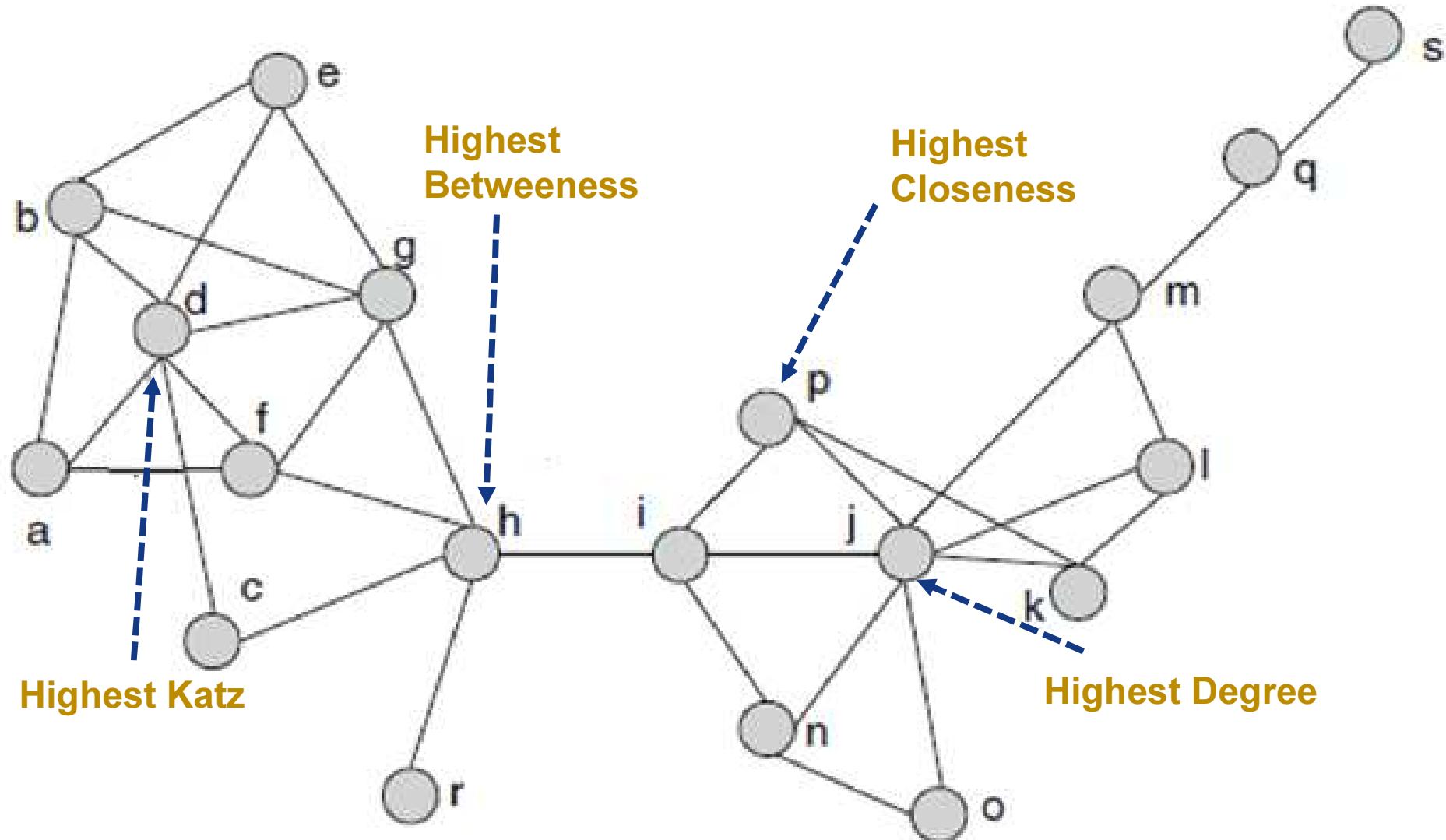
特征向量中心性，一个节点与其邻居节点的中心性得分的总和成正比。
与重要的节点连接的节点更重要。
有少量有影响的联系人的节点其中心性可能超过拥有大量平庸的联系人的
节点。

特征向量中心性的计算：

- 1、计算图的成对邻接矩阵的特征分解
- 2、选择有最大特征值的特征向量
- 3、第 i 个节点的中心性等于特征向量中的第 i 元素



不同中心性度量对比



PageRank算法

PageRank身世



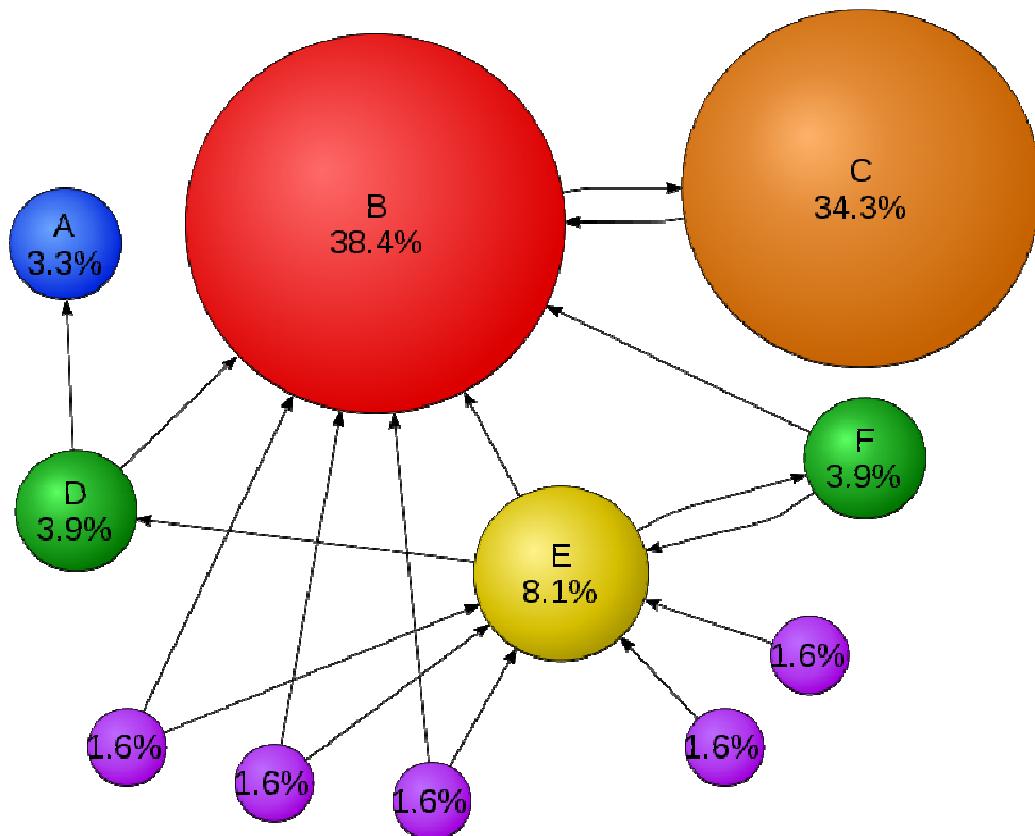
➤ 提出

- Google的创始人之一Larry Page于1998年提出了PageRank， 并应用在Google搜索引擎的检索结果排序上， 该技术也是Google早期的核心技术之一
- 有向图上的特征向量中心性

➤ 核心思想

- 一个节点的“得票数”由所有链向它的节点的重要性来决定，到一个节点的边相当于对该节点投一票。一个节点的PageRank是由所有链向它的节点的重要性经过递归算法得到的。一个有较多链入的节点会有较高的等级，相反如果一个节点没有任何链入边，那么它没有等级。
- 在网络整体结构的意义上寻找整个网络中最核心的成员—传播影响力最大

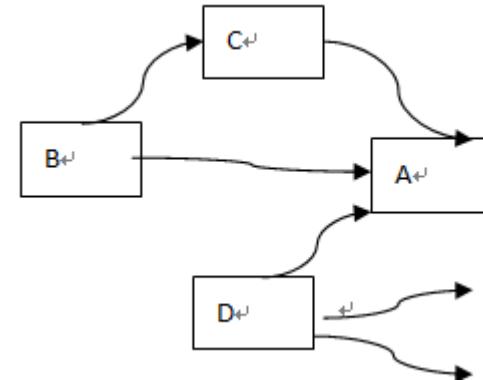
PageRank



PageRank 是基于「从许多优质的网页链接过来的网页，必定还是优质网页」的回归关系，来判定所有网页的重要性。

链向网页E的链接远远多于链向网页C的链接，但是网页C的重要性却大于网页E。这是因为因为网页C被网页B所链接，而网页B有很高的重要性。

PageRank简单计算：



- 假设一个由只有4个页面组成的集合：A，B，C和D。如果所有页面都链向A，那么A的PR（PageRank）值将是B，C及D的和。

$$PR(A) = PR(B) + PR(C) + PR(D)$$

- 继续假设B也有链接到C，并且D也有链接到包括A的3个页面。一个页面不能投票2次。所以B给每个页面半票。以同样的逻辑，D投出的票只有三分之一算到了A的PageRank上。

$$PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{1} + \frac{PR(D)}{3}$$

- 换句话说，根据链出总数平分一个页面的PR值。

$$PR(A) = \frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)}$$

PageRank的简单计算过程

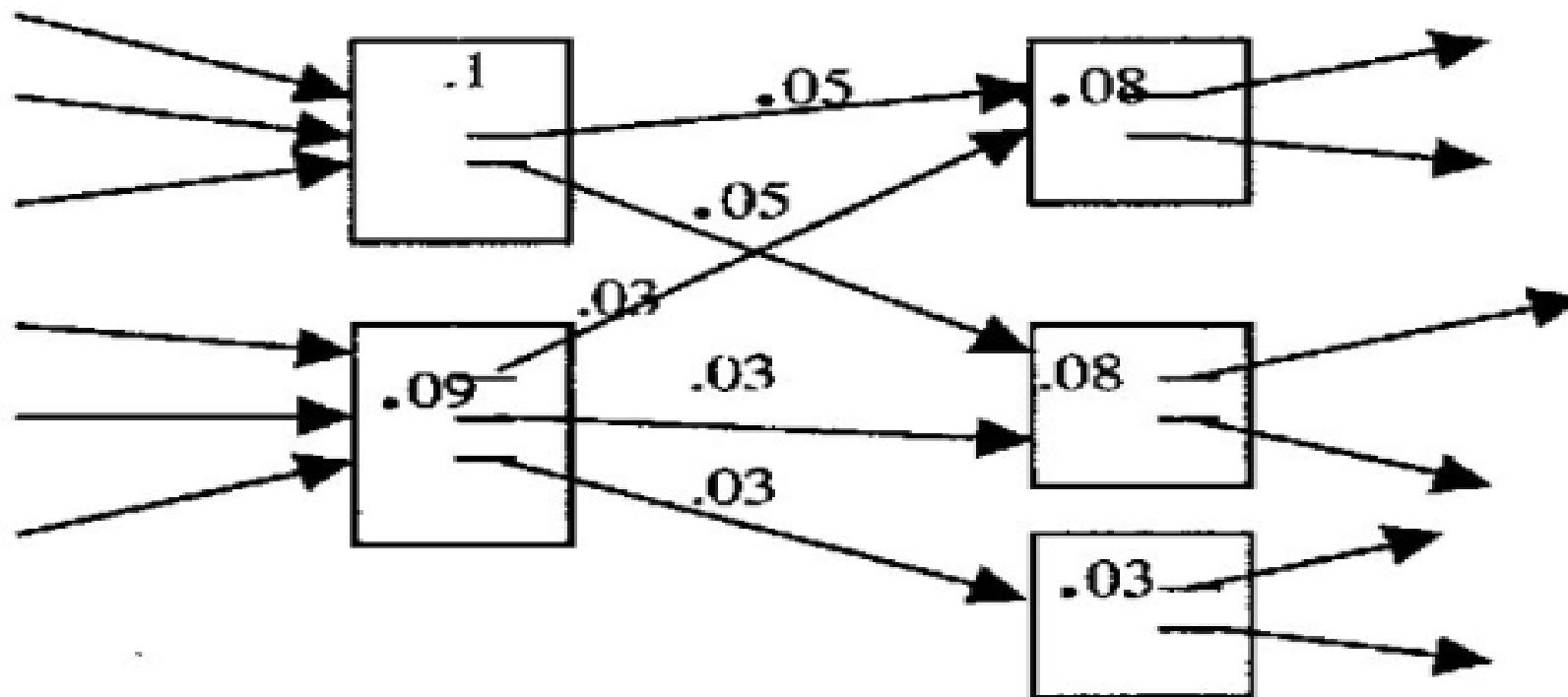


图 1 链接结构中的部分网页及其 PageRank 值

PageRank的简化模型

可以把互联网上的各网页之间的链接关系看成一个有向图。假设冲浪者浏览的下一个网页链接来自于当前网页。建立简化模型：对于任意网页 P_i ，它的PageRank值可表示为如下：其中 B_i 为所有链接到网页 i 的网页集合， L_j 为网页 j 的对外链接数（出度）。

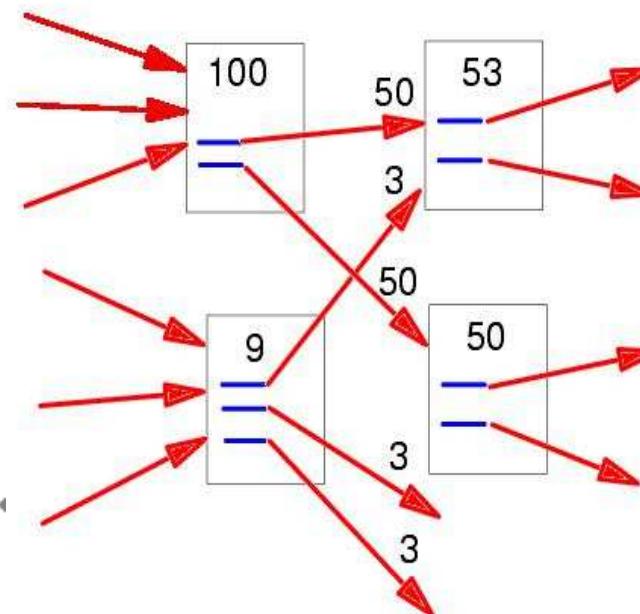
$$PR_i = \sum_{j \in B_i} \frac{PR_j}{L_j}$$

PR_i : 网页 i 的pagerank值

PR_j : 网页 j 的pagerak值

L_j : 网页 j 链出的连接数

— B_i : 链接到网页 i 的网页集合



PageRank的矩阵计算

- 定义邻接矩阵为 G , 若网页 j 到网页 i 有超链接, 则 $g_{ij} = 1$; 反之, $g_{ij} = 0$
- 设共有 m 个网页, 分别编号为 1、2、3、...、 m , 它们的级别 (重要性) 分别记为 r_1 、 r_2 、 r_3 、...、 r_m , G 表示由这些网页组成的有向图的邻接矩阵。根据有向图理论:

$$r(u) = \sum_{v \in B_u} \frac{r(v)}{n_v} \quad \Rightarrow \quad r_i = \sum_{j=1}^m \frac{g_{ij}}{n_j} r_j$$

矩阵形式

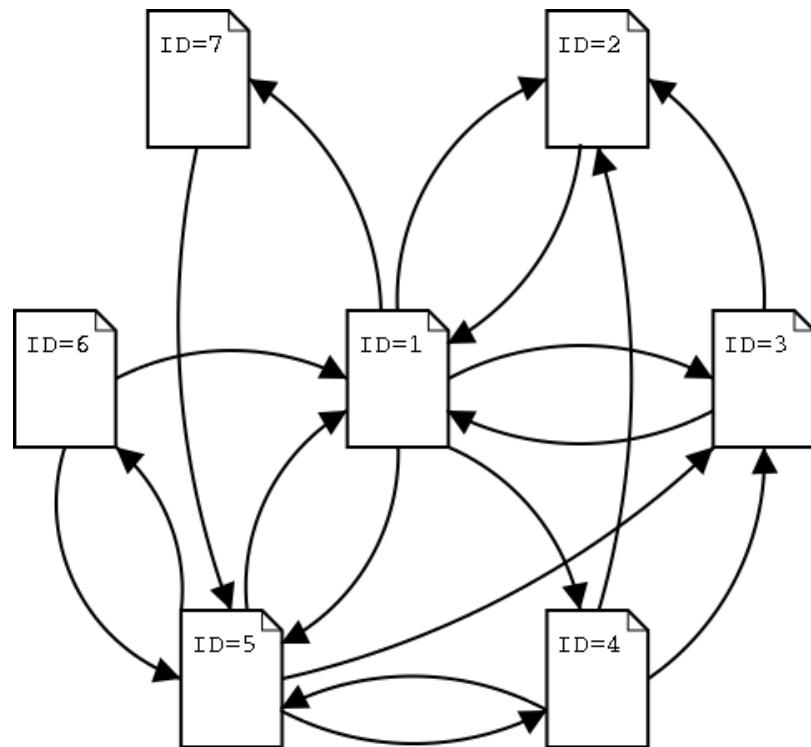
$$\boxed{r = G_m \cdot r}$$

其中 $\begin{cases} \mathbf{r} = (r_1, r_2, \dots, r_m)^T \\ G_m = \{g_{ij} / n_j\} \end{cases}$

G 中第 j 列
的列和

➤ 可知 r 是 G_m 的对应于特征值为 1 的特征向量

某7个网页的链接关系图与邻接矩阵



$$G = \begin{matrix} 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

PageRank的计算： Gm矩阵

$$Gm = \begin{matrix} & \begin{matrix} 0 & 1 & 1/2 & 0 & 1/4 & 1/2 & 0 \end{matrix} \\ \begin{matrix} 0 \\ 1/5 \\ 1/5 \\ 1/5 \\ 1/5 \\ 0 \\ 1/5 \end{matrix} & \begin{matrix} 1/2 \\ 0 \\ 1/2 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} \end{matrix} \begin{matrix} & \begin{matrix} 1/3 & 0 & 0 & 1/3 & 1/4 & 0 & 0 \end{matrix} \\ \begin{matrix} 1/3 \\ 0 \\ 1/3 \\ 0 \\ 0 \\ 1/4 \\ 0 \end{matrix} & \begin{matrix} 0 & 1/4 & 0 & 1/3 & 0 & 1/2 & 1 \end{matrix} \end{matrix} \begin{matrix} & \begin{matrix} 0 & 1/4 & 0 & 0 & 1/4 & 0 & 0 \end{matrix} \\ \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} & \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \end{matrix}$$

PageRank的计算

0.699456533837389
0.382860418521518
0.323958815672054
0.242969111754040
0.412311219946251
0.103077804986563
0.139891306767478

求矩阵 G_m 特
征值1对应的
特征向量

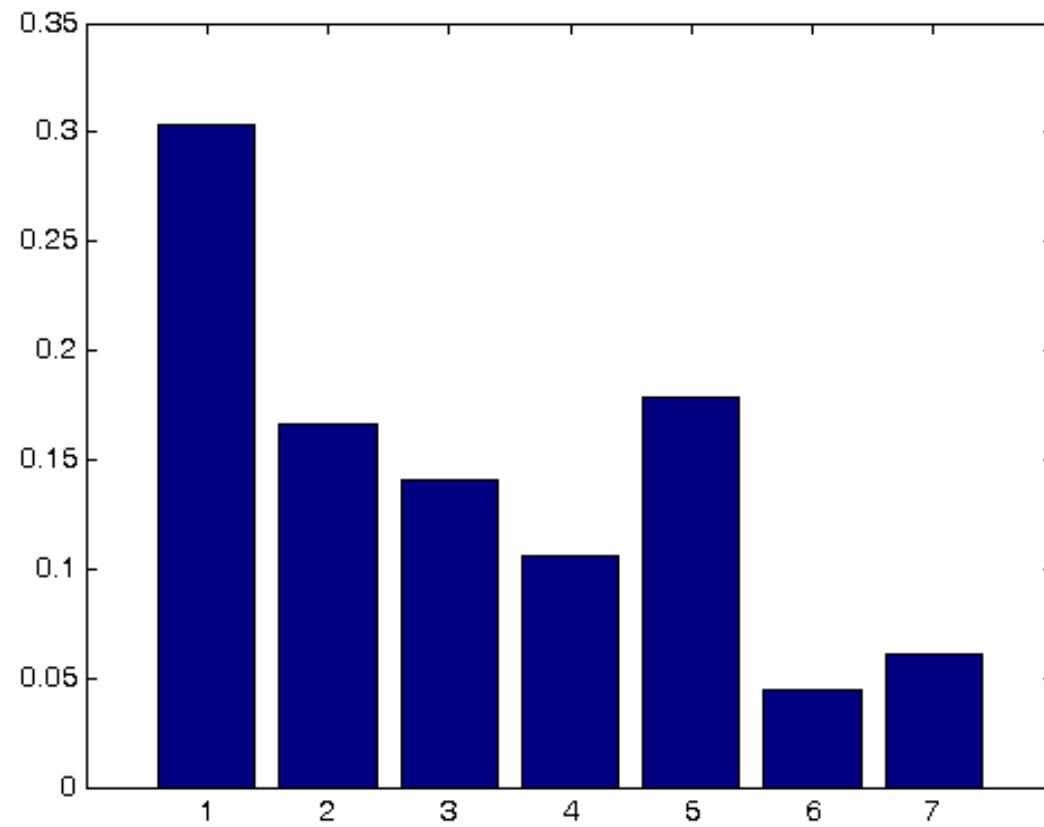
$\alpha =$

归一化



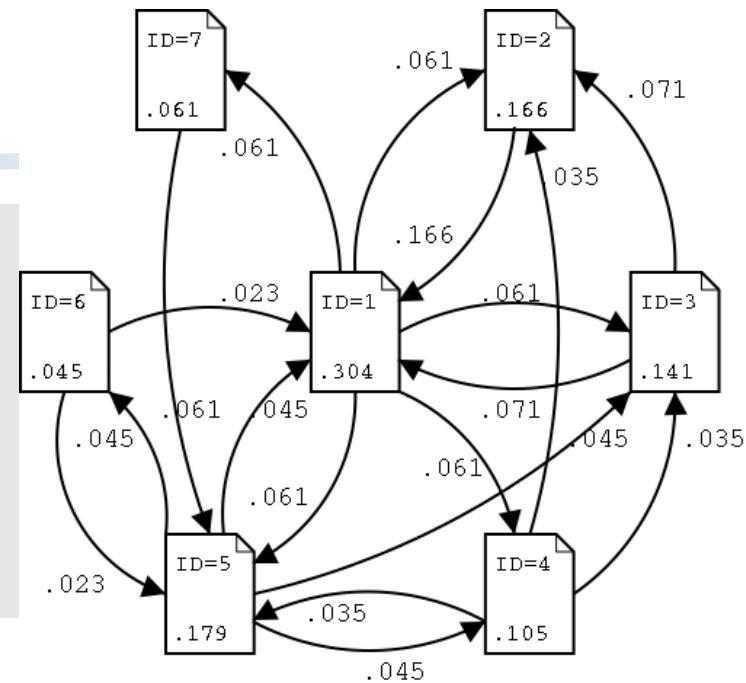
0.303514376996805
0.166134185303514 α
0.140575079872204
0.105431309904153
0.178913738019169
0.0447284345047923
0.0607028753993610

7个网页的PageRank值



PageRank结果的评价

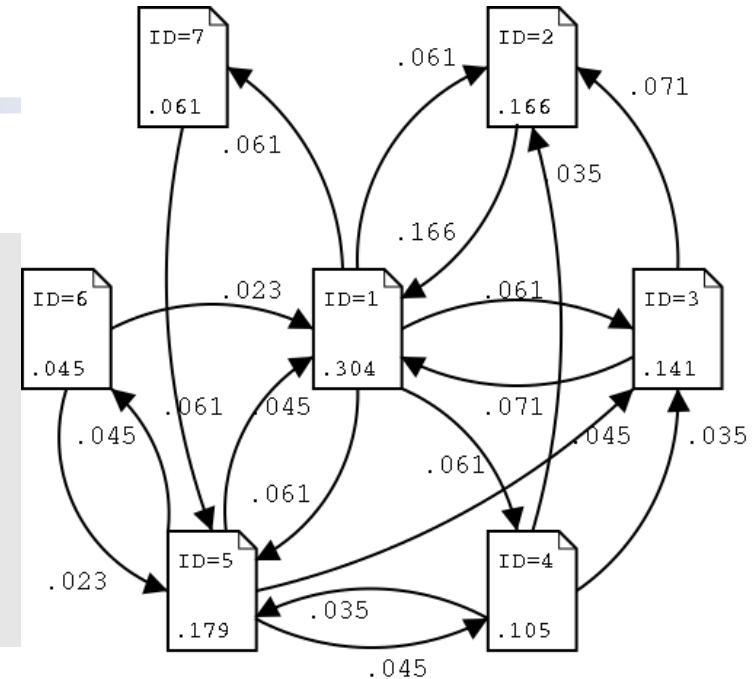
名次	PageRank	文件 ID	发出链接 ID	被链接 ID
1	0.304	1	2, 3, 4, 5, 7	2, 3, 5, 6
2	0.179	5	1, 3, 4, 6	1, 4, 6, 7
3	0.166	2	1	1, 3, 4
4	0.141	3	1, 2	1, 4, 5
5	0.105	4	2, 3, 5	1, 5
6	0.061	7	5	1
7	0.045	6	1, 5	5



- 我们详细地看一下。ID=1 的页面的PageRank 是0. 304，占据全体的三分之一，成为了第1位。
- 特别需要说明的是，起到相当大效果的是从排在第3位的 ID=2 页面中得到了所有的PageRank (0. 166) 数。ID=2页面有从3个地方过来的链入链接，而只有面向 ID=1页面的一个链接，因此(面向ID=1页面的)链接就得到ID=2的所有PageRank数。
- 不过，就因为ID=1页面是链出链接和链入链接最多的页面，也可以理解它是最受欢迎的页面。

PageRank结果的评价

名次	PageRank	文件 ID	发出链接 ID	被链接 ID
1	0.304	1	2, 3, 4, 5, 7	2, 3, 5, 6
2	0.179	5	1, 3, 4, 6	1, 4, 6, 7
3	0.166	2	1	1, 3, 4
4	0.141	3	1, 2	1, 4, 5
5	0.105	4	2, 3, 5	1, 5
6	0.061	7	5	1
7	0.045	6	1, 5	5



- 反过来，最后一名的 ID=6 页面只有 ID=1 的 15% 的微弱评价。
- 总之，即使有同样的链入链接的数目，链接源页面评价的高低也影响 PageRank 的高低。

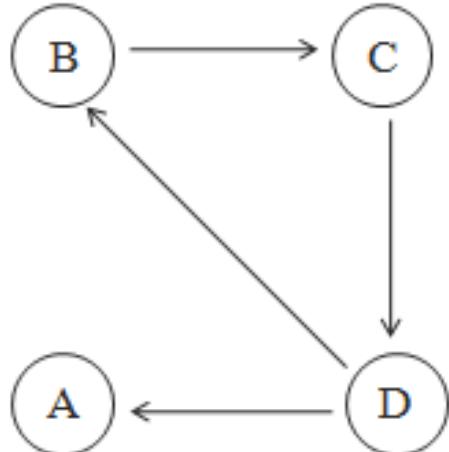
So Easy·····

PageRank不就是求解 G_m
的特征值为1的特征向量

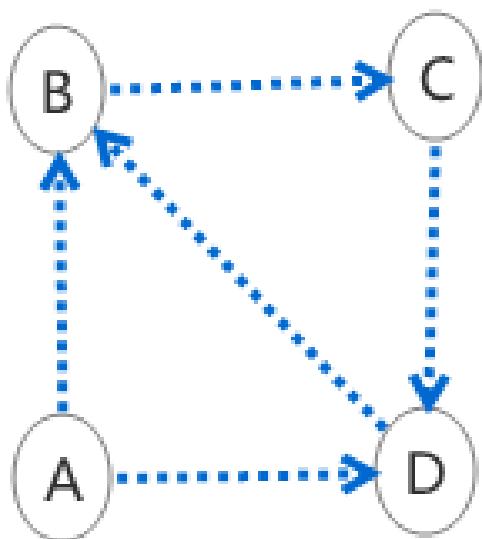
矩阵 G_m 一定有特征值 1 吗？即上面的方程是否有解？

如果 $G = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, 则 $r_1 = r_2$, 此时就无法进行求解

回到现实



	PR(A)	PR(B)	PR(C)	PR(D)
初始	0.25	0.25	0.25	0.25
一次迭代	0.125	0.125	0.25	0.25
二次迭代	0.125	0.125	0.125	0.25
三次迭代	0.125	0.125	0.125	0.125
...
n次迭代	0	0	0	0



	PR(A)	PR(B)	PR(C)	PR(D)
初始	0.25	0.25	0.25	0.25
一次迭代	0	0.375	0.25	0.375
二次迭代	0	0.375	0.375	0.25
三次迭代	0	0.25	0.375	0.375
四次迭代	0	0.375	0.25	0.375
五次迭代	0

用理论照进现实

➤ Perron-Frobenius定理

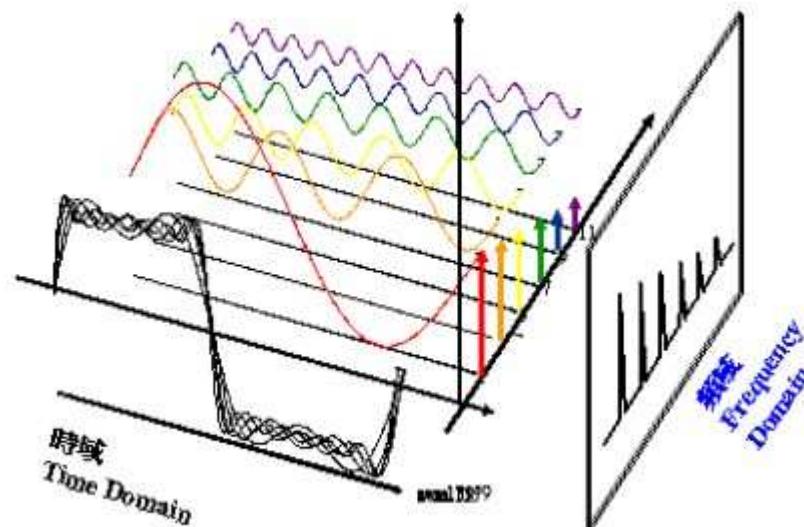
- 矩阵A不可约，即满足
 - 强连通
 - 非周期
- 结论

$$\mathbf{x} = \mathbf{A} \mathbf{x}, \quad \mathbf{x} \text{ 满足: } \sum_{i=1}^n x_i = 1$$

- 该方程组解存在且唯一
- \mathbf{x} 是A的最大特征值1所对应的特征向量

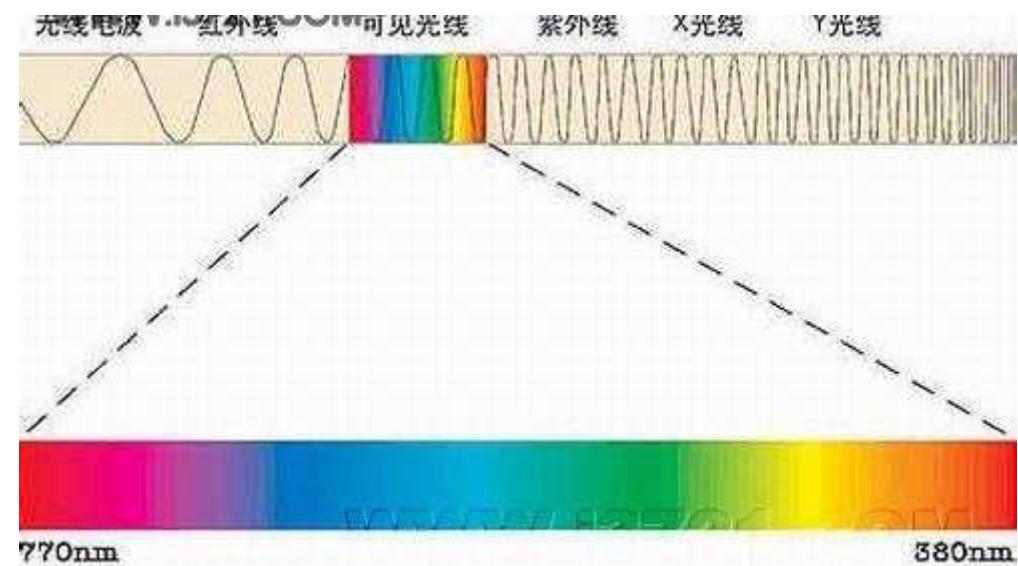
如何构造A使之不可约？

谱



圖二 時域與頻域的差異

- 简单地说，谱这个概念来自“分而治之”的策略。一个复杂的东西不好直接研究，就把它分解成简单的分量。
- 如果我们把一个东西看成是一些分量叠加而成，那么这些分量以及它们各自所占的比例，就叫这个东西的谱。
- 所谓频谱，就是把一个信号分解成多个频率单一的分量。



靠谱？

矩阵的谱结构

➤ 矩阵谱

- 就是它的特征值和特征向量，普通的线性代数课本会告诉你定义：如果 $A v = c v$ ，那么 c 就是 A 的特征值， v 就叫特征向量。
- 这仅仅是数学家发明的一种数学游戏么？

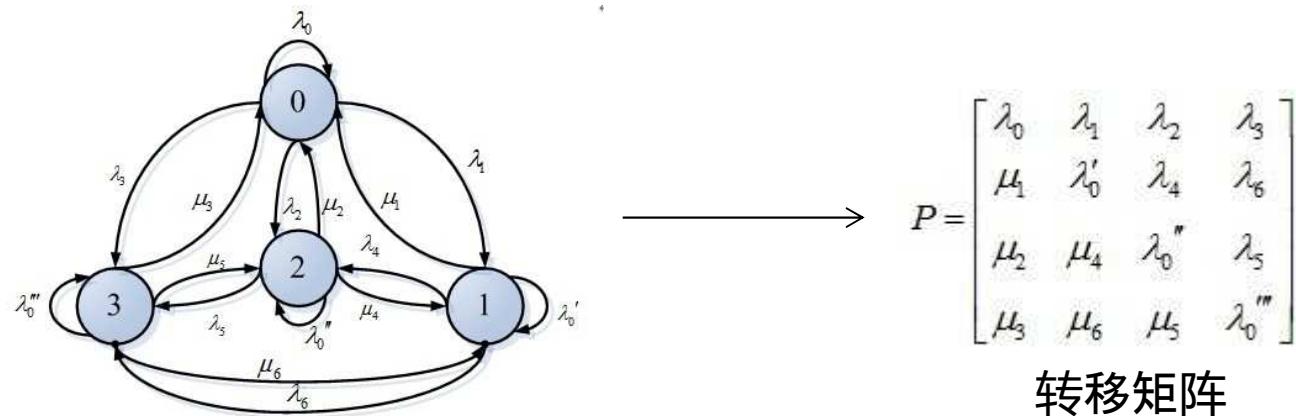
➤ 矩阵的空间角度

- 这里的谱代表了一种分量结构，它使用“分而治之”策略来研究矩阵，这里可以把矩阵理解为一个变换，它的作用就是把一个向量变成另外一个向量： $y = A x$ 。对于某些向量，矩阵对它的作用很简单， $A v = cv$ ，相当于把这个向量 v 拉长了 c 倍。把这种和矩阵 A 能如此密切配合的向量 v_1, v_2, \dots 叫做特征向量，这个倍数 c_1, c_2, \dots 叫特征值。
- 当出现一个新的向量 x 的时候，可以把 x 分解为这些向量的组合， $x = a_1 v_1 + a_2 v_2 + \dots$ ，那么 A 对 x 的作用就可以分解： $A x = A (a_1 v_1 + a_2 v_2 + \dots) = a_1 c_1 v_1 + a_2 c_2 v_2 \dots$ 所以，矩阵的谱就是用于分解一个矩阵的。

矩阵的时间角度

➤ 马尔可夫过程

- “将来只由现在决定，和过去无关”。
- 考虑一个图，图上每个点有一个值，会被不断更新。每个点通过一些边连接到其它一些点上，对于每个点，这些边的值都是正的，和为1。在图上每次更新一个点的值，就是对和它相连接的点的值加权平均。



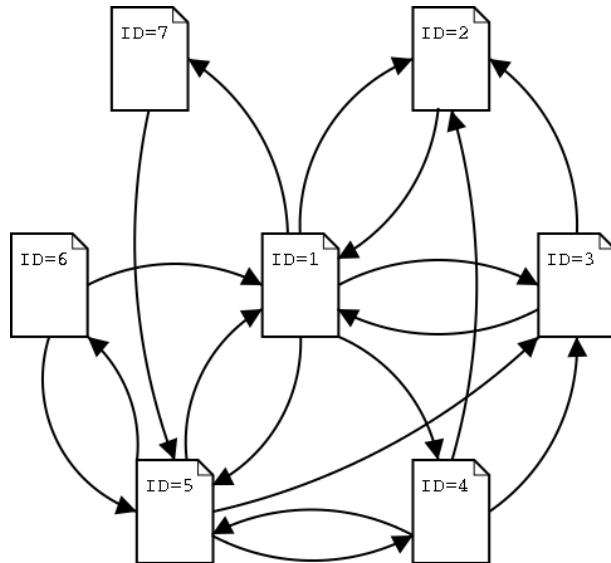
- 如果图是联通并且非周期（数学上叫各态历经性，ergodicity），那么这个过程最后会收敛到一个唯一稳定的状态（平衡状态）

图-谱

➤ 矩阵谱结构

- 矩阵 G_m 其实就是这个马尔可夫过程的转移概率矩阵。
- 把各个节点的值放在一起可以得到一个向量 v ，可以获得对这个过程的代数表示， $v(t+1) = A v(t)$ 。
- 稳态情况下， $v = Av$ ，稳定状态就是 A 的一个特征向量，特征值就是1。
- 谱的概念
 - 把 A 的特征向量都列出来 v_1, v_2, \dots ，它们有 $A v_i = c_i v_i$ 。 v_i 其实就是一种很特殊，但是很简单状态，对它每进行一轮更新，所有节点的值就变成原来的 c_i 倍。
 - 如果 $0 < c_i < 1$ ，那么，相当于所有节点的值呈现指数衰减，直到大家都趋近于0。

网页浏览过程：Gm矩阵的时间角度



$$G_m = \begin{matrix} 0 & 1 & 1/2 & 0 & 1/4 & 1/2 & 0 \\ 1/5 & 0 & 1/2 & 1/3 & 0 & 0 & 0 \\ 1/5 & 0 & 0 & 1/3 & 1/4 & 0 & 0 \\ 1/5 & 0 & 0 & 0 & 1/4 & 0 & 0 \\ 1/5 & 0 & 0 & 1/3 & 0 & 1/2 & 1 \\ 0 & 0 & 0 & 0 & 1/4 & 0 & 0 \\ 1/5 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

- PR的马尔可夫浏览模型
 - 设想有一个永不休止浏览网页的人，每次随机选择一个指向链接继续访问，这个过程与过去浏览的页面无关，而仅依赖于当前页面。
 - 稳态情况下，每个网页 v 会有一个被访问的概率 $p(v)$ ，等价于网页的重要程度rank，依赖于上一个时刻到达“链向” v 的网页的概率，以及那些网页中超链的个数。

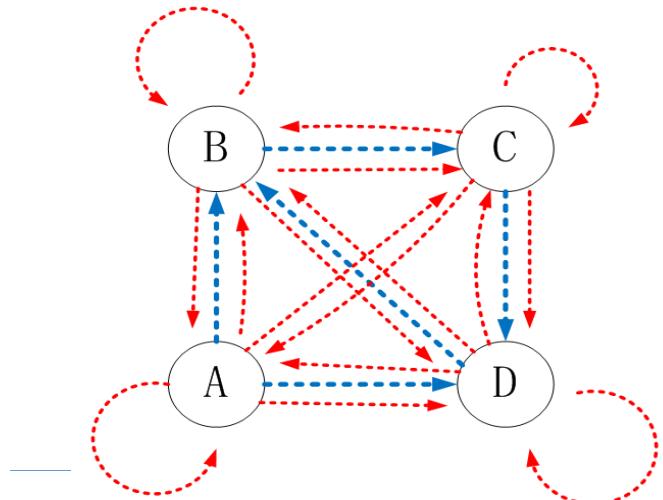
随机浏览修正

➤ 不可约

- 当浏览器所浏览的网页矩阵存在不可达或周期连通分量时，该浏览器将无法继续浏览其它网页。

➤ 修正

- 每次访问网页时，可以随机选择一个其它的网页重新开始浏览
 - ① 这种随机模型更加接近于用户的浏览行为
 - ② 一定程度上解决了rank leak和rank sink的问题
 - ③ 保证pagerank具有唯一值



设定任意两个顶点之间都有直接通路，在每个顶点处以概率d按原来蓝色方向转移，以概率1-d按红色方向转移。

PageRank改进

➤ PR修正模型

- 让浏览者每次以一定的概率 $(1-\alpha)$ 沿着超链走，以概率 (α) 重新随机选择一个新的起始节点
- α 选在0.1和0.2之间，被称为阻尼系数
- 矩阵 $M=(1-\alpha)G_m + \alpha/N(1_N)$ 满足不可约特性，存在平稳分布 r ，

$$r = \left((1 - \alpha)G_m + \frac{\alpha}{n} (1_N) \right) r$$

一般取0.15

各元素均为1的
N阶矩阵

PageRank计算挑战

- 挑战1：存储限制
 - 假设 N 是 10000 的 order。通常，数值计算程序内部行列和矢量是用双精度记录的，N 次正方行列 A 的存储空间为 `sizeof(double)* N * N = 8 * 104 * 104 = 800MB。`
 - 本实验 N 是 160000 个，方阵存储空间超过 200GB
- 挑战2：计算问题
 - 特征向量计算规模是 $O(n^3)$
 - 特征向量的求解，就是求解方程 $A\alpha = \alpha$ 是 N 元一次方程组，一般不能得到分析解，所以只能解其数值。
 - 然而，常用的迭代求解方法会导致收敛速度很慢。

Power Iteration计算

➤ 解决方法-Power Iteration幂迭代

当矩阵 A 的阶很大，无法直接计算其特征值和特征向量时，需要使用该方法

- 1) 输入矩阵 A 和迭代初始向量 v ，以及精度 $\epsilon > 0$ 例如0.0001，向量各元素对应差值绝对值)，令 $k = 0$ ；
- 2) 计算： $v_{k+1} = Av_k$ ；
- 3) 如果 $|v_{k+1} - v_k| < \epsilon$ ，则计算 PageRank 值并停止。否则转第二步。

$$x = A^k v / \text{sum}(A^k v)$$

$A^k v$ 即 v_k

PageRank算法只有两步：构造矩阵A和迭代求解

幂迭代收敛速度

➤ 收敛速度

- 对于任意一个初始状态 U
- 用谱的方法来分析，把 U 分解成 $U = v_1 + c_2 v_2 + c_3 v_3 + \dots$ (在数学上可以严格证明，对于上述的转移概率矩阵，最大的特征值就是 1，这里对应于平衡状态 v_1 ，其它的特征状态 v_2, v_3, \dots 对应于特征值 $1 > c_2 > c_3 > \dots > -1$)。
- 可以看到，当更新进行了 t 步之后，状态变成 $U(t) = v_1 + c_2^t v_2 + c_3^t v_3 + \dots$ ，除了代表平衡状态的分量保持不变外，其它分量随着 t 增长而指数衰减，最后，其它整个趋近于平衡状态。
- 从上面的分析看到，这个过程的收敛速度，和衰减得最慢的那个非平衡分量是密切相关的，它的衰减速度取决于第二大特征值 c_2 ， c_2 的大小越接近于 1，收敛越慢，越接近于 0，收敛越快。
 -

幂迭代计算

➤ 幂迭代算法

- 确定合适的初始向量 v ，例如 $v = \text{indegree}(\text{node}) / |E|$
- 每次迭代是一次矩阵向量乘法复杂度 $> O(n^2)$ ，但 A 是稀疏矩阵，所以整个迭代速度非常快
- 收敛速度取决于 c_2
- 3亿个页面的 Web Graph
-> 50 iterations to convergence (Brin and Page, 1998)

➤ 幂迭代算法的马尔可夫链模型

- page importance \Leftrightarrow steady-state Markov probabilities \Leftrightarrow eigenvector
- Larry Page 和 Sergey Brin 的贡献，一方面加入随机游走解决了矩阵收敛问题，另一方面由于互联网网页数量巨大，生成的二维矩阵巨大，两人利用稀疏矩阵计算简化了计算量。

引申：图的空间角度

➤ 聚类结构

- c_2 的大小取决于图上的聚类结构。图上的聚类结构越明显， c_2 越大， $c_2 = 1$ 时，整个图就断裂成非连通的两块或者多块。
- c_2 越大，越容易对这个图上的点进行聚类。机器学习中谱聚类就是利用了第二大特征值对应的谱结构。
- 如果图上的点分成几组，各自聚成一团，缺乏组与组之间的联系，那么这种结构是很不利于扩散的。在某些情况下，甚至需要 $O(\exp(N))$ 的时间才能收敛。这也符合我们的直观想象，好比两个大水缸，它们中间的只有一根很细的水管相连，那么就需要好长时间才能达到平衡。

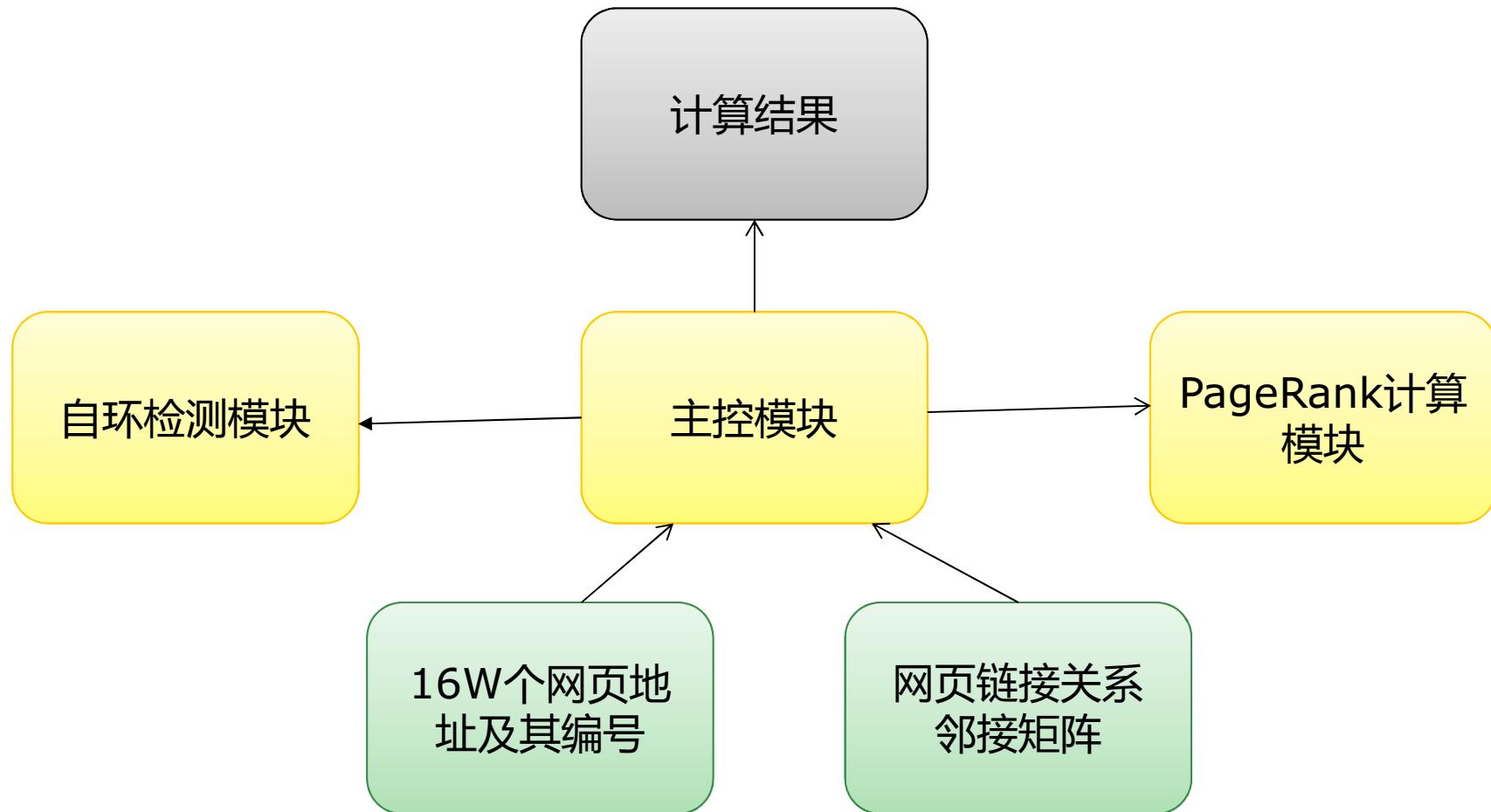
引申：图谱的时空关系

➤ 总结

- 图是表达事物关系和传递扩散过程的重要数学抽象
- 图的矩阵表达提供了使用代数方法研究图的途径
- 谱，作为一种重要的代数方法，其意义在于对复杂对象和过程进行分解
- 图上的马尔可夫更新过程是很多实际过程的一个重要抽象
- 图的谱结构的重要意义在于通过它对马尔可夫更新过程进行分解分析
- 图的第一特征值对应于马尔可夫过程的平衡状态，第二特征值刻画了这个过程的收敛速度（采样的效率，扩散和传播速度，网络的稳定程度）
 -
- 图的第二特征分量与节点的聚类结构密切相关。可以通过谱结构来分析图的聚类结构。
- 马尔可夫过程代表了一种时间结构，聚类结构代表了一种空间结构，“谱”把它们联系在一起了，在数学刻画了这种时与空的深刻关系。

图分析器的工程 实现

分析器总体设计



构造链接图

➤ 要点

- url存储
- hash

稀疏矩阵

➤ 概念

- 简单说，设矩阵A中有s个非零元素，若s远远小于矩阵元素的总数（即 $s \ll m \times n$ ），则称A为稀疏矩阵。
- 精确点，设在矩阵A中，有s个非零元素。令 $e=s/(m \times n)$ ，称e为矩阵的稀疏因子。通常认为 $e \leq 0.05$ 时称之为稀疏矩阵。

➤ 存储方式

- 在存储稀疏矩阵时，为了节省存储单元使用压缩存储方法。
- 非零元素的分布一般没有规律，存储非零元素的同时，还必须同时记下它所在的行和列的位置 (i, j) 。
- 两种存储方式：
 - 三元组 (i, j, a_{ij}) 唯一确定了矩阵A的一个非零元。因此，稀疏矩阵可由表示非零元的三元组及其行列数唯一确定。
 - 十字链表方法，矩阵的每一个非零元素用一个结点表示，该结点除了 $(row, col, value)$ 以外，还要有以下两个链域：right：用于链接同一行中的下一个非零元素；down：用于链接同一列中的下一个非零元素。

稀疏矩阵的三元组存储

```
#define MAXSIZE 1000 /*非零元素的个数最多为1000*/
#define MAXROW 1000 /*矩阵最大行数为1000*/
typedef struct
{
    int row, col; /*该非零元素的行下标和列下标*/
    ElementType e; /*该非零元素的值*/
}Triple;
typedef struct
{
    Triple data [MAXSIZE+1]; /* 非零元素的三元组表，data [0] 未用*/
    int first [MAXROW+1]; /* 三元组表中各行第一个非零元素所在的位置 */
    int m, n, len; /*矩阵的行数、列数和非零元素的个数*/
}TriSparMatrix;
```

三元组存储示例

所示的稀疏矩阵的三元组的表示如下：

$$\left(\begin{array}{ccccccc} 0 & 12 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & 0 & 0 & 0 & 0 & 14 & 0 \\ 0 & 0 & 24 & 0 & 0 & 0 & 0 \\ 0 & 18 & 0 & 0 & 0 & 0 & 0 \\ 15 & 0 & 0 & -7 & 0 & 0 & 0 \end{array} \right)$$

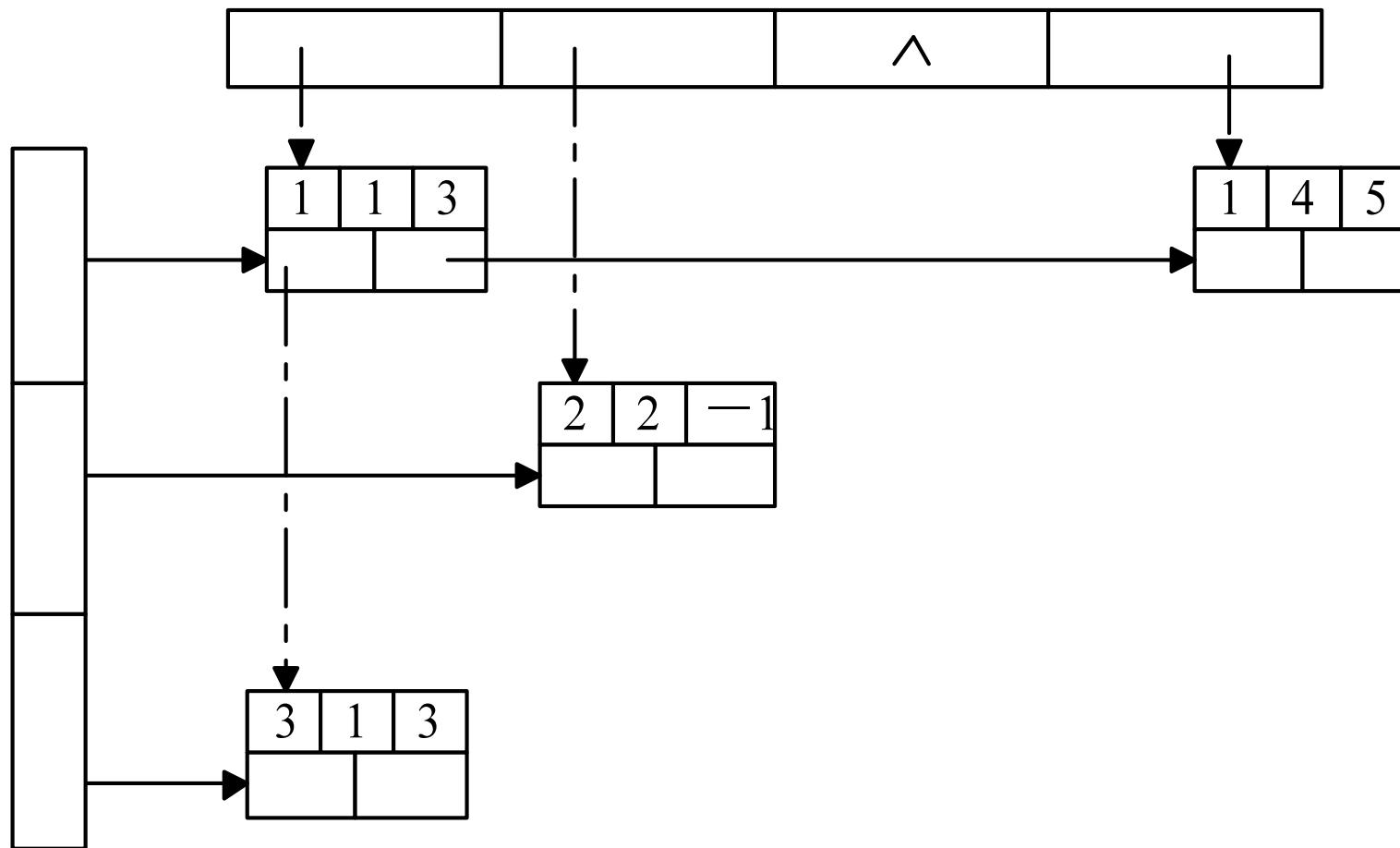
i	j	v
1	2	12
1	3	9
3	1	-3
3	6	14
4	3	24
5	2	18
6	1	15
6	4	-7

十字链表存储结构

十字链表的结构类型说明如下：

```
typedef struct OLNode
{
    int         row, col;      /* 非零元素的行和列下标 */
    ElementType value;
    struct OLNode * right, *down; /* 非零元素所在行表、列表的后继链域 */
}OLNode; *OLink;

typedef struct
{
    OLink * row_head, *col_head; /* 行、列链表的头指针向量 */
    int   m, n, len; /* 稀疏矩阵的行数、列数、非零元素的个数 */
}CrossList;
```



十字链表的结构

实验说明

➤ 程序数据来源

- 大约16万个网页
- 其链接地址已保存到本地文件

➤ 运行要求

- `./graphanalyzer graph.txt rank.txt loop.txt`
- `rank.txt`
 - 显示PageRank最大的前10个URL地址
 - 格式：第一列pagerank值，第二列url
- `loop.txt`
 - 第一行：环的个数
 - 从第二行开始每行一个环，用逗号分隔每个url

➤ 参数设计建议

- PageRank的 α 取值0.15，迭代精度 取值0.0001



THE END