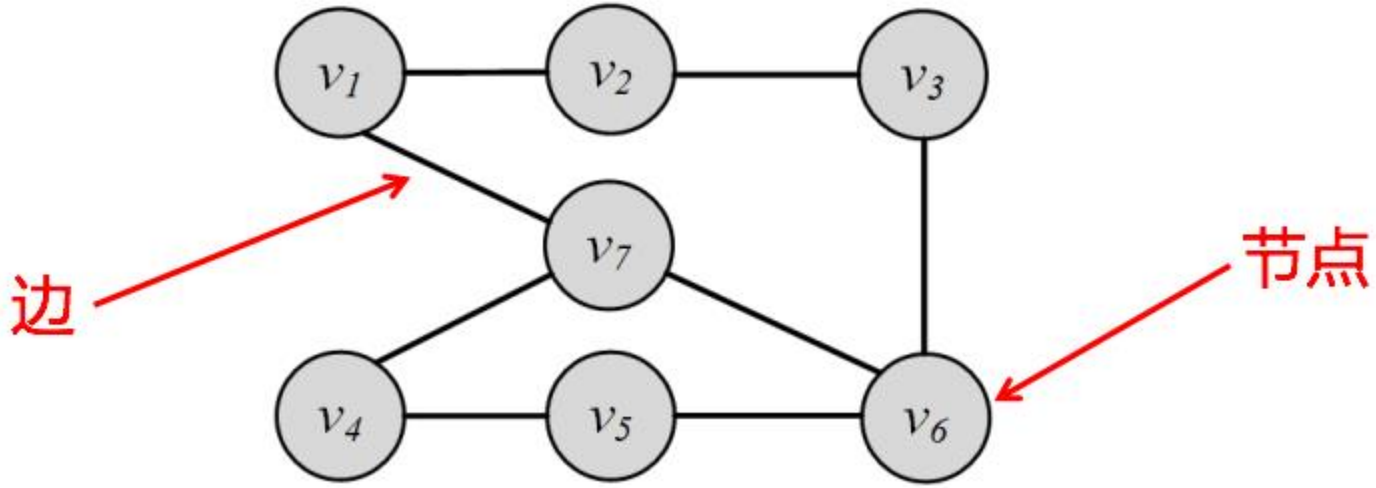
复杂网络

梅子行


终于上课了呢

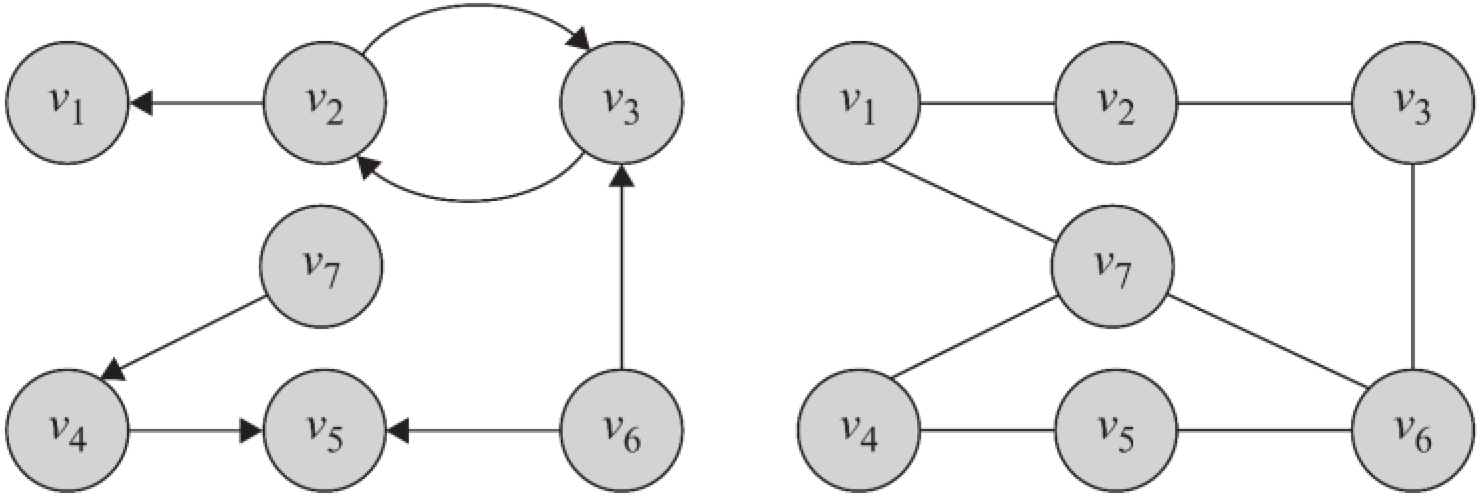
图的基本概念

- 1. 节点和边



图的基本概念

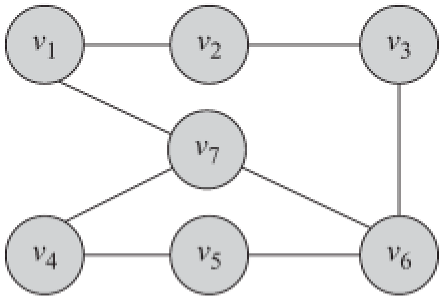
- 2. 有向图和无向图



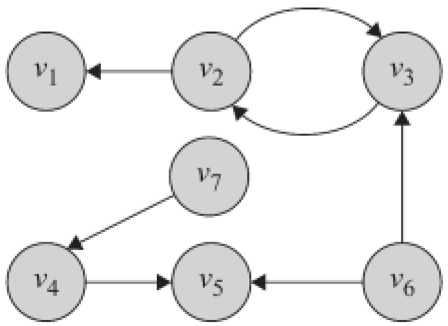
(a) Directed Graph (b) Undirected Graph

图的基本概念

- 3. 节点的度数



degree $(v_3) = 2$

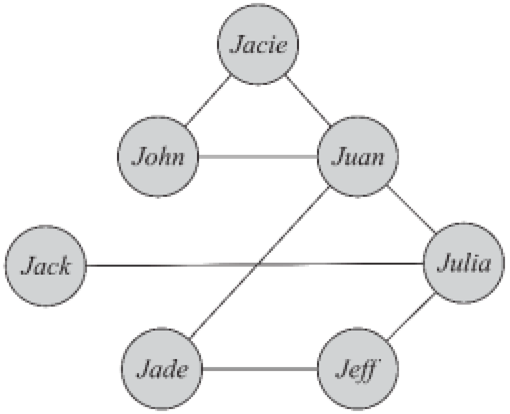


in-degree $(v_3) = 2$ 入度（粉丝数）

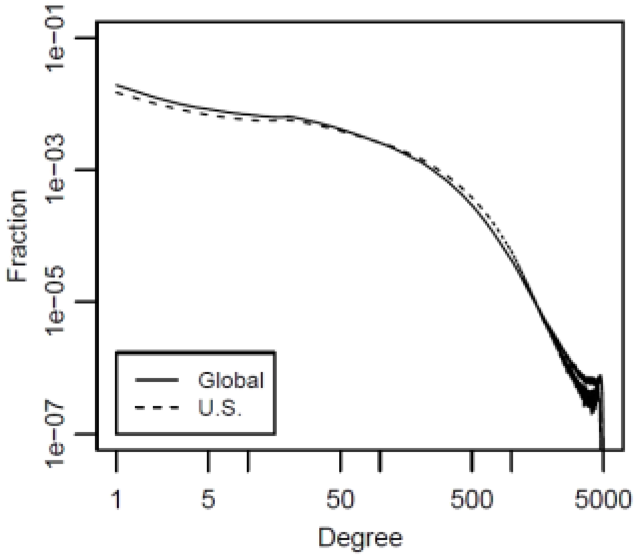
out-degree $(v_3) = 1$ 出度（关注数）

图的基本概念

- **节点度数分布**



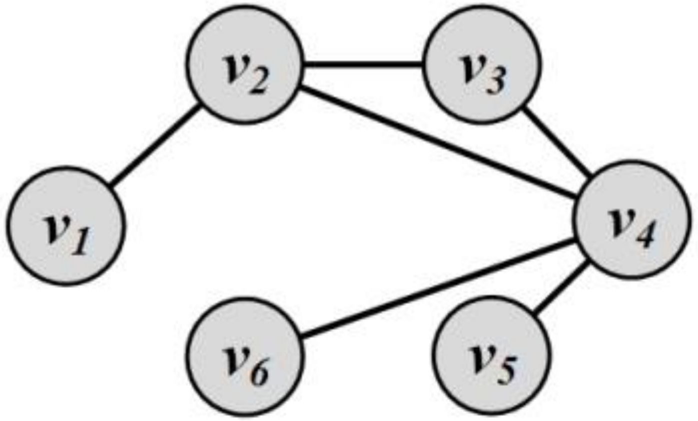
$$p_1 = \frac{1}{7}, p_2 = \frac{4}{7}, p_3 = \frac{1}{7}, p_4 = \frac{1}{7}$$



Facebook 好友数分布
（幂律分布，power-law）

计算机里图的存储方法

- **边表**



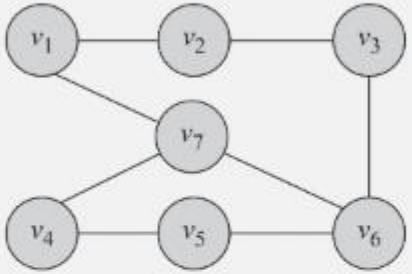
(v_1,v_2)
(v_2,v_3)
(v_2,v_4)
(v_3,v_4)
(v_4,v_5)
(v_4,v_6)

边表占用存储空间小，是目前较为常用的图存存储方法。

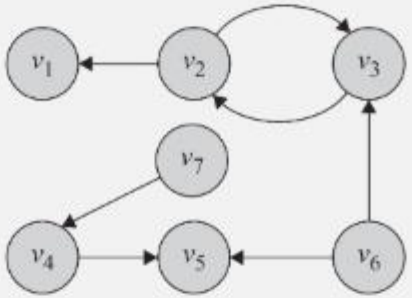
图中，一组数据代表一条边。

在有向图里，第一个数据代表起始点，第二个数据代表结束点。

Centrality： Degree



degree $(v_3) = 2$

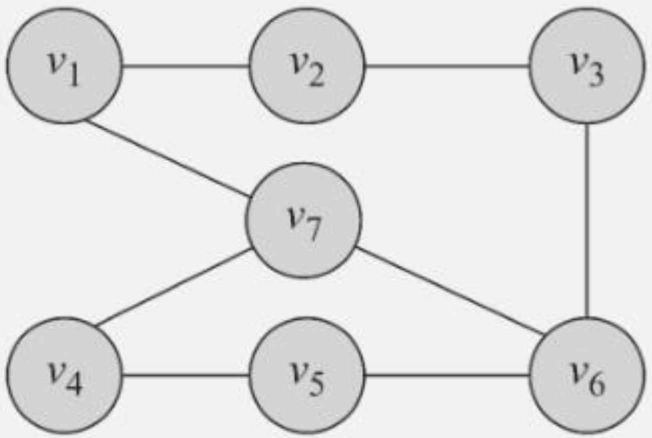


in-degree $(v_3) = 2$　入度（粉丝数）
out-degree $(v_3) = 1$　出度（关注数）

局限：邻接节点的重要性没有考虑

Centrality： Eigenvector

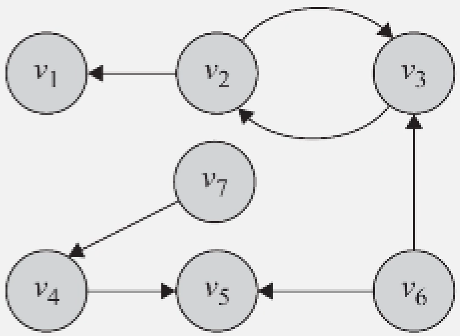
- eigenvector方法的思想：影响力大的人不仅仅是朋友多，而且他的朋友也是重要的



$$c(v_7) = \frac{1}{\lambda}[c(v_1) + c(v_4) + c(v_6)]$$

Centrality： Eigenvector

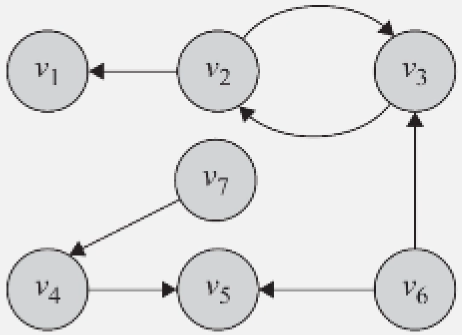
- eigenvector方法在无向图上的表现非常优异
- 但当出现在向无环图时，其中节点 eigenvector centrality变成0



$c(v_6) = ?$

Centrality： Eigenvector

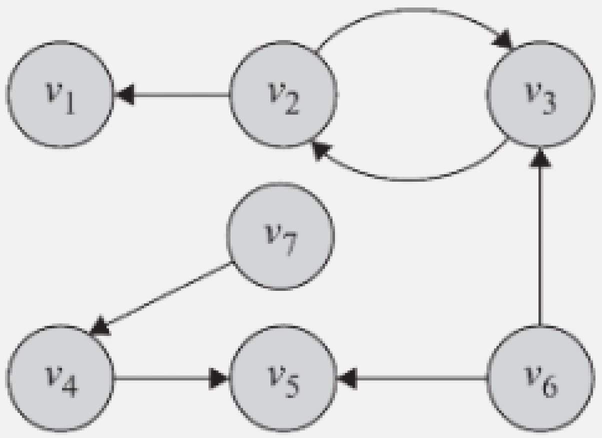
- 为此，Katz提出了一个改进方法，即每个节点初始就有一个centrality值
- 这样，上图中v6和v5节点的centrality计算方法就变成了：



$$c(v_6) = \beta$$
$$c(v_5) = \alpha * [c(v_4) + c(v_6)] + \beta$$

Centrality： PageRank

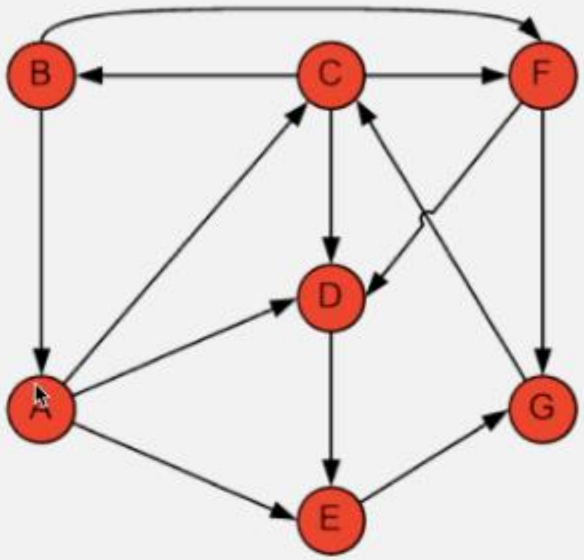
- Katz在计算时，每条出边都会带上起始节点的完整中心性值，这是不合理的
- PageRank算法在Katz的基础上，假设一个节点的出度是n，刚每条出边附上1/n的起始节点的中心度量值



$$c(v_6) = \beta$$

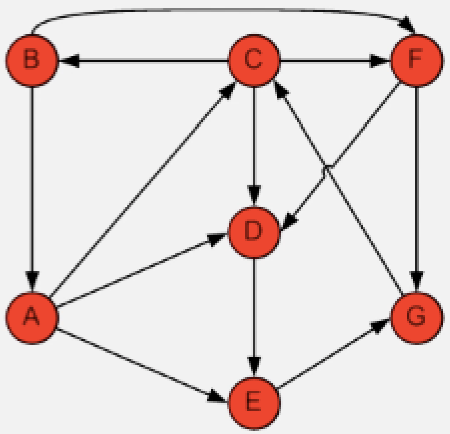
$$c(v_5) = \alpha * [1 * c(v_4) + \frac{1}{2} * c(v_6)] + \beta$$

Centrality： PageRank

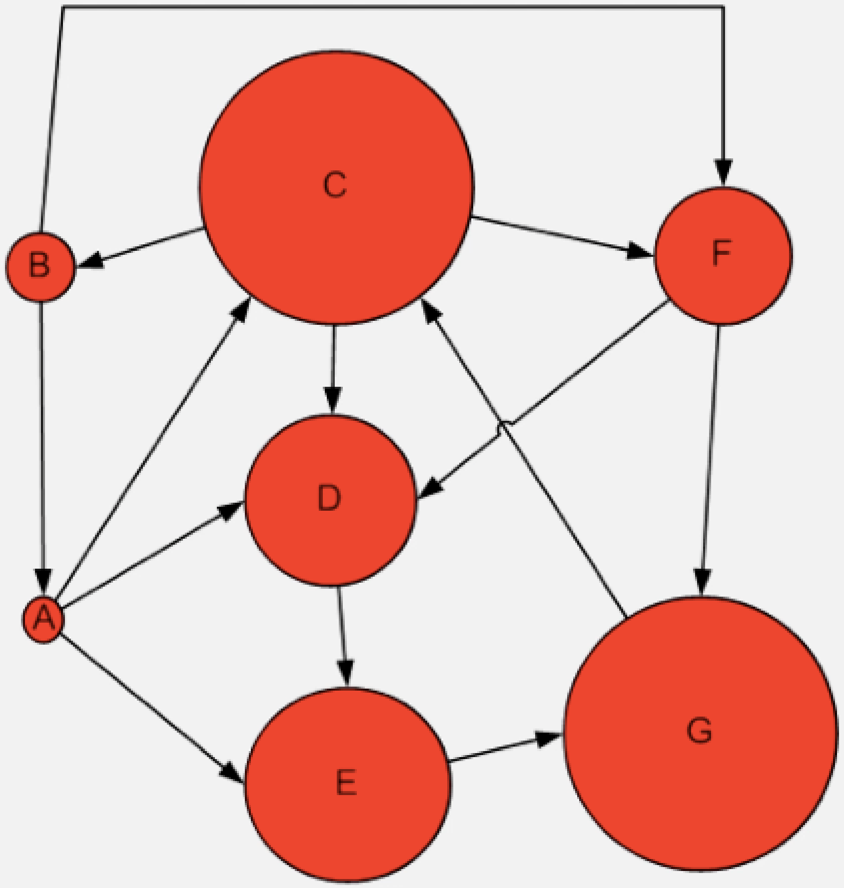


Step	A	B	C	D	E	F	G
0	1/7	1/7	1/7	1/7	1/7	1/7	1/7
1	B/2	C/3	A/3 + G	A/3 + C/3 + F/2	A/3 + D	C/3 + B/2	F/2 + E
	0.071	0.048	0.190	0.167	0.190	0.119	0.214

Centrality： PageRank



Node	Rank
A	7
B	6
C	1
D	4
E	3
F	5
G	2

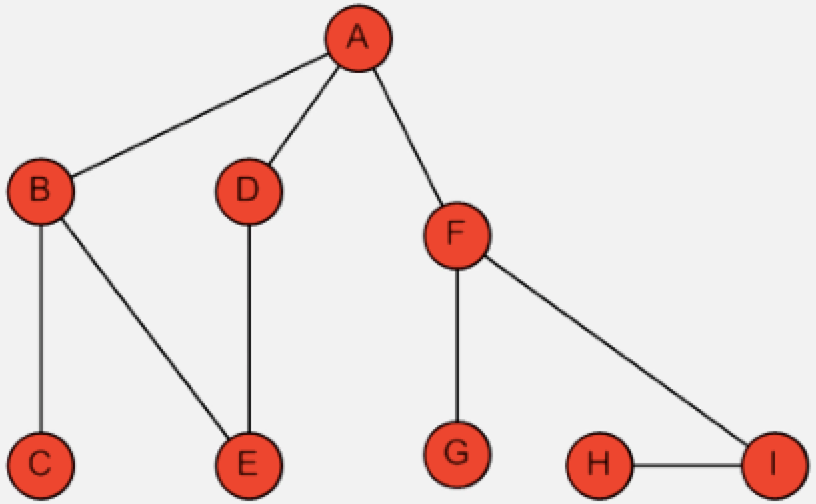


Centrality： Betweenness

$$C_b(v_i) = \sum_{s \neq t \neq v_i} \frac{\sigma_{st}(v_i)}{\sigma_{st}}$$

σ_{st} 从节点 s 到 t 的最短路径数目

$\sigma_{st}(v_i)$ 从节点 s 到 t 的，且经过v_i的最短路径数目

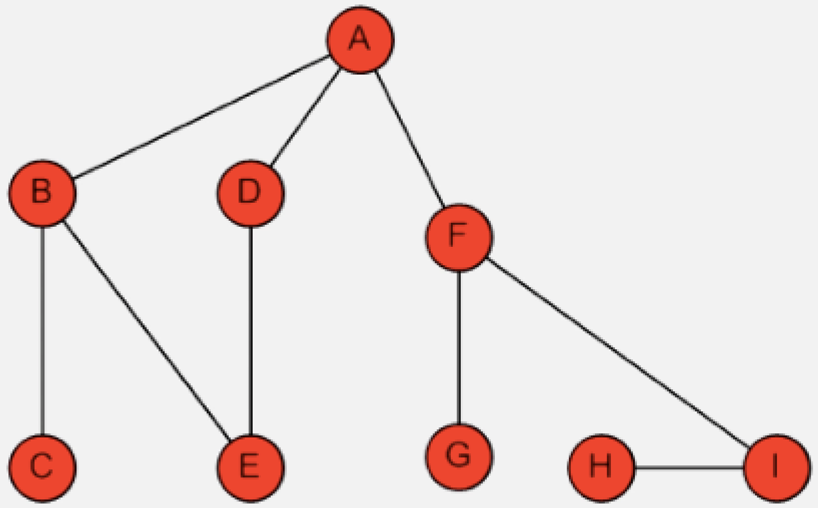


Centrality： Closeness

$$C_c(v_i) = \frac{1}{\bar{l}_{v_i}}$$

$$\bar{l}_{v_i} = \frac{1}{n-1} \sum_{v_j \neq v_i} l_{i,j}$$

$l_{i,j}$: 从节点 v_i 到节点 v_j 的最短路径长度

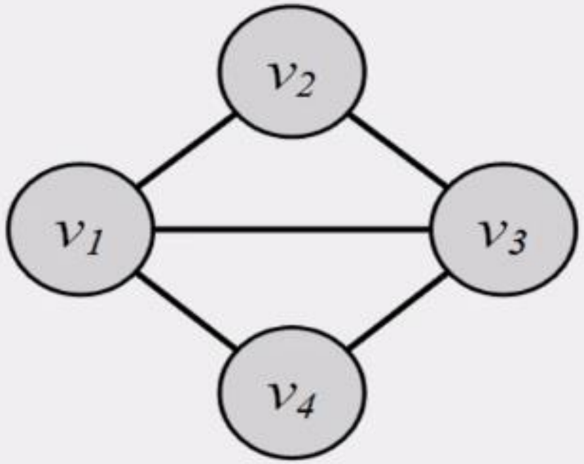


传递性 Transitivity

传递性→三角形

$$C = \frac{|\text{Closed Paths of Length 2}|}{|\text{Paths of Length 2}|}$$

$$C = \frac{(\text{Number of Triangles})\times 3}{\text{Number of Connected Triples of Nodes}}$$

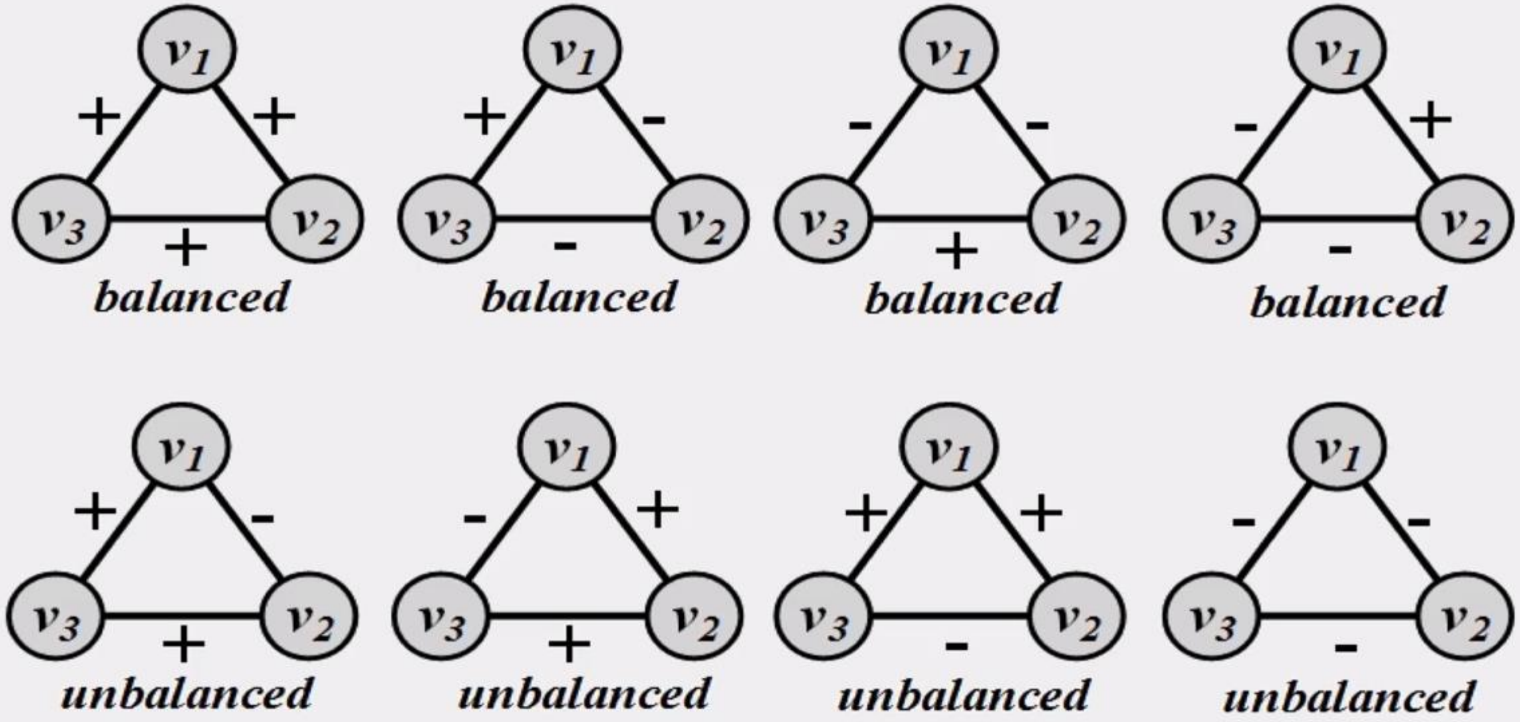


聚类系数越高，越接近于完全图

$$\begin{aligned} C &= \frac{(\text{Number of Triangles}) \times 3}{\text{Number of Connected Triples of Nodes}} \\ &= \frac{2 \times 3}{2 \times 3 + \underbrace{2}_{v_2 v_1 v_4, v_2 v_3 v_4}} = 0.75. \end{aligned}$$

结构平衡理论

The friend of my friend is my friend,
The friend of my enemy is my enemy,
The enemy of my enemy is my friend,
The enemy of my friend is my enemy.



朋友边权重+1，敌人边权重-1
那么，三个节点形成的三角形是平衡的，当且仅当

$$w_{ij}w_{jk}w_{ki} \geq 0.$$

网络相似性

根据节点自身的属性值

年龄
性别
地域
兴趣
收入
……

根据节点在网络中的位置

邻居是否相同
邻居的邻居是否相同
……

结构等价性

两个节点的共有邻居节点数目定义了两个节点之间的相似程度

兄弟： 共同的姐妹、父母、祖父母……
随机两人： 并没有太多共同的邻居节点

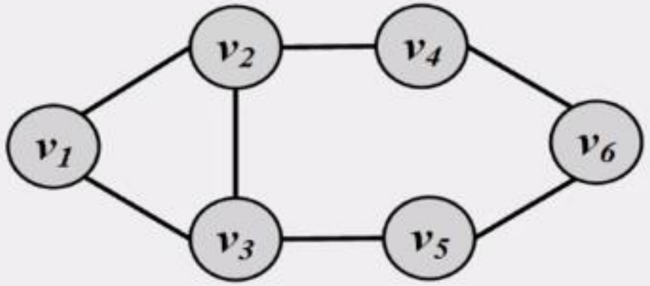
$$\sigma(v_i, v_j) = |N(v_i) \cap N(v_j)|$$

如果A和B共享5个好友，C和D也共享5个好友
A、B各自有10个好友，C、D各自有100个好友
哪组相似性更高？

归一化

Jaccard Similarity: $$\sigma_{Jaccard}(v_i, v_j) = \frac{|N(v_i) \cap N(v_j)|}{|N(v_i) \cup N(v_j)|}$$

Cosine Similarity: $$\sigma_{Cosine}(v_i, v_j) = \frac{|N(v_i) \cap N(v_j)|}{\sqrt{|N(v_i)||N(v_j)|}}$$



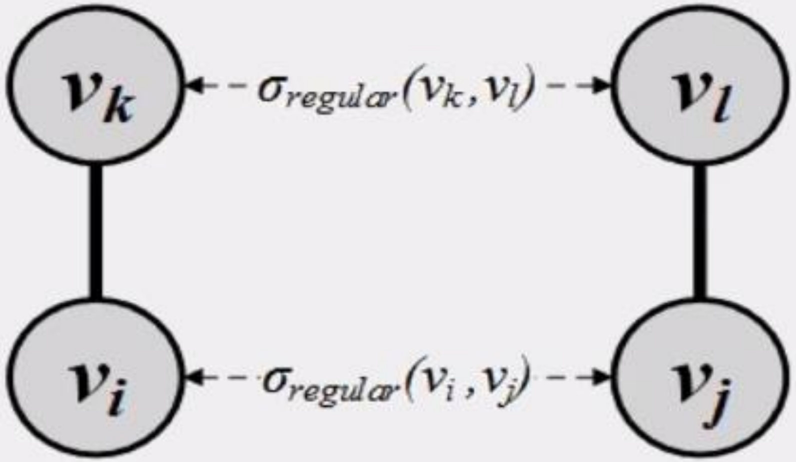
$$\sigma_{\text{Jaccard}}(v_2, v_5) = \frac{|\{v_1, v_3, v_4\} \cap \{v_3, v_6\}|}{|\{v_1, v_3, v_4, v_6\}|} = 0.25$$

$$\sigma_{\text{Cosine}}(v_2, v_5) = \frac{|\{v_1, v_3, v_4\} \cap \{v_3, v_6\}|}{\sqrt{|\{v_1, v_3, v_4\}||\{v_3, v_6\}|}} = 0.40.$$

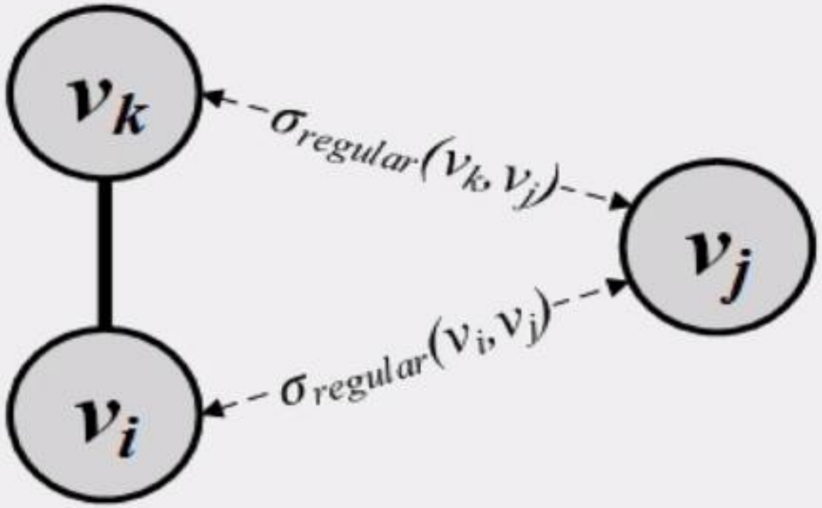
Regular Equivalence

不仅仅看相同的邻居，对于不相同的邻居，也看他们是否相似

$$\sigma_{\text{regular}}(v_i, v_j) = \alpha \sum_{k,l} A_{i,k} A_{j,l} \sigma_{\text{regular}}(v_k, v_l)$$



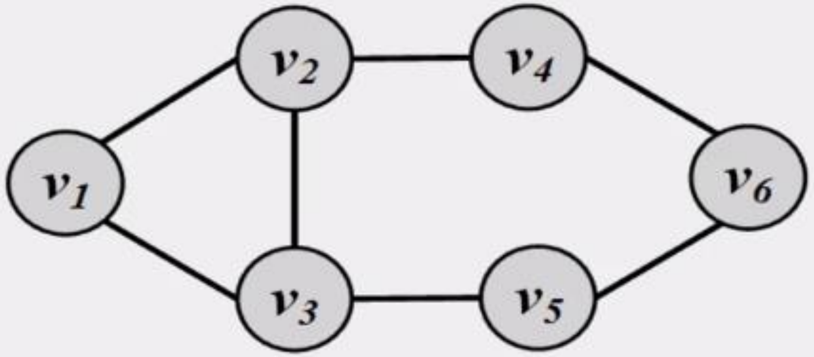
Regular Equivalence



$$\sigma_{regular}(v_i, v_j) = \alpha \sum_k A_{i,k} \sigma_{Regular}(v_k, v_j)$$

设定合适的α，我们就可以得到收敛之后的Regular Equivalence

Regular Equivalence



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

$$\alpha = 0.3$$

$$\begin{bmatrix} 1.43 & 0.73 & 0.73 & 0.26 & 0.26 & 0.16 \\ 0.73 & 1.63 & 0.80 & 0.56 & 0.32 & 0.26 \\ 0.73 & 0.80 & 1.63 & 0.32 & 0.56 & 0.26 \\ 0.26 & 0.56 & 0.32 & 1.31 & 0.23 & 0.46 \\ 0.26 & 0.32 & 0.56 & 0.23 & 1.31 & 0.46 \\ 0.16 & 0.26 & 0.26 & 0.46 & 0.46 & 1.27 \end{bmatrix}$$

v_2和v_3的相似性最大

谢谢大家