

课程链接: [CS224W: Machine Learning with Graphs](#)

课程视频: [【课程】斯坦福 CS224W: 图机器学习 \(2019 秋 | 英字\)](#)

目录

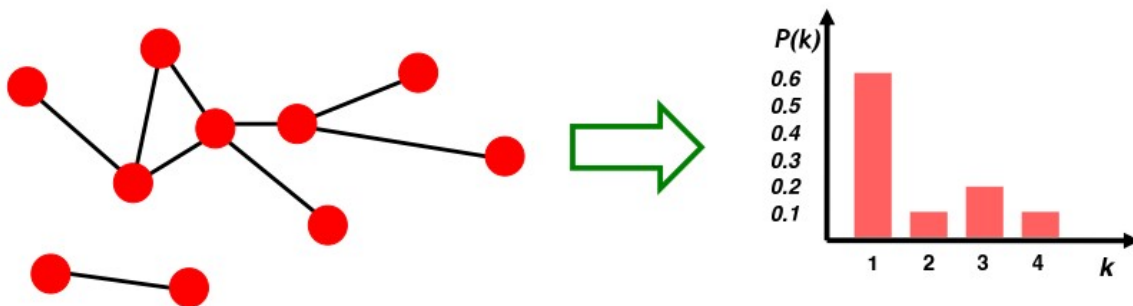
- [1. How to measure a Network?—网络的属性](#)
 - [1.1 度分布 \(Degree distribution\) \$P\(k\)\$](#)
 - [1.2. 路径长度 \$h\$](#)
 - [1.3. 聚类系数 \(Clustering coefficient\) \$C\$](#)
 - [1.4. 连通分量 \(Connected components\) \$s\$](#)
- [2. 一些真实网络案例的属性计算](#)
 - [2.1 MSN Messenger网络](#)
 - [2.2 PPI网络 \(蛋白质相互作用的网络\)](#)
- [3. 最简单的图模型——ER Random Graph Model \(ER随机图模型\)](#)
 - [3.1 \$G_{np}\$ 图](#)
 - [3.2 \$G_{np}\$ 图与实际网络的比较](#)
- [4. 小世界网络模型 The Small-world model](#)
- [5. Kronecker Graph Model——Generating large realistic graphs](#)

1. How to measure a Network?—网络的属性

1.1 度分布 (Degree distribution) $P(k)$

度分布 $P(k)$ 是指, 网络中, 度为 k 的节点的出现概率; 对于有向图来说又分为入度分布和出度分布。如果网络中总共有 N 个节点, 其中有 N_k 个节点, 他的度为 k , 那么

$$P(k) = \frac{N_k}{N} \quad \text{where} \quad N_k = \text{nodes with degree } k$$



1.2. 路径长度 h

路径 是一个节点序列，其中每个节点都链接到下一个节点。（A path is a sequence of nodes in which each node is linked to the next one）。路径可以重复经过节点和边；在有向图里面，路径的方向必须要沿着箭头的方向。

有了路径的概念，就可以定义图中两个点之间的 距离（distance）。距离即最短路径，如果两个节点不相连，则通常会将这两个节点之间的距离定义为 $h = \infty$ 或者 $h = 0$ 。

在此基础上，可以定义图的 直径（diameter）——The maximum (shortest path) distance between any pair of nodes in a graph，即图中节点距离的最大值。

对于连通图或者强连通有向图来说，图的 平均路径长度（Average path length）可以由下面这个式子计算：

$$\bar{h} = \frac{1}{2E_{max}} \sum_{i,j \neq 1} h_{ij}$$

其中 h_{ij} 是节点 i 和节点 j 之间的距离， $E_{max} = n(n-1)/2$ 是图中的最大边数。通常情况下，在计算时我们会忽略不相连的节点之间的距离。

1.3. 聚类系数（Clustering coefficient） C

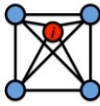
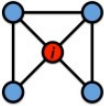
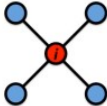
参考资料：[聚类系数的含义和计算](#)

在图论中，集聚系数（也称群聚系数、集群系数）是用来描述一个图中的顶点之间结集成团的程度的系数。

对于图中的节点*i*来说，其聚类系数 $C_i \in [0, 1]$ 。在计算聚类四叔时，找出其直接邻居节点集合 N_i ，计算 N_i 构成的网络中的边数 k ，除以 N_i 集合中可能存在的边数 $|N_i| * |N_i - 1|/2$ 。即：

$$C_i = \frac{n}{C_k^2} = \frac{2e_i}{k_i(k_i - 1)}$$

其中 e_i 表示节点*i*的邻居节点构成的边， k_i 表示节点*i*的度， $k_i(k_i - 1)$ 是节点*i*与邻居节点所能相连的最大的边的数量。

图	聚类系数计算
 <p>$C_i = 1$</p>	<p>对于节点<i>i</i>来说，邻居节点一共有4个，这4个邻居节点构成了6条边，他们所有可能构成的边为$C_4^2 = \frac{4 \times 3}{2} = 6$因此其聚类系数为$C_i = \frac{6}{6} = 1$</p>
 <p>$C_i = 1/2$</p>	<p>对于节点<i>i</i>来说，邻居节点一共有4个，这4个邻居节点构成了3条边，他们所有可能构成的边为$C_4^2 = \frac{4 \times 3}{2} = 6$因此其聚类系数为$C_i = \frac{3}{6} = \frac{1}{2}$</p>
 <p>$C_i = 0$</p>	<p>对于节点<i>i</i>来说，邻居节点一共有4个，这4个邻居节点构成了0条边，他们所有可能构成的边为$C_4^2 = \frac{4 \times 3}{2} = 6$因此其聚类系数为$C_i = \frac{0}{6} = 0$</p>

1.4. 连通分量（Connected components） *s*

无向图G的极大连通子图称为G的连通分量(Connected Component)。任何连通图的连通分量只有一个，即是其自身，非连通的无向图有多个连通分量。

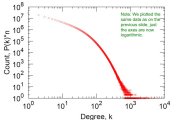
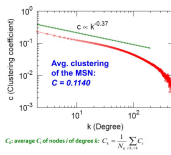
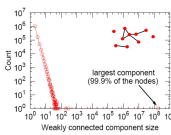
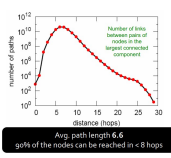
如何找到一张图的联通分量？

- 随机选择一个节点作为起点，进行广度优先搜索；
- 将广度优先搜索经过的节点进行标记；
- 如果所有的节点都进行了标记，则该图是一个连通图；
- 如果存在未标记的节点，从未标记的节点中随机选择一个节点作为起点进行广度优先搜索；重复第2步和第4步，直至所有节点都标记完毕；最后得到的多个连通子图中对的极大连通子图就是该图的连通分量。

2. 一些真实网络案例的属性计算

2.1 MSN Messenger网络

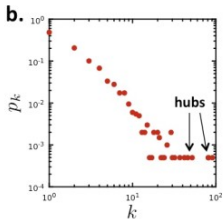
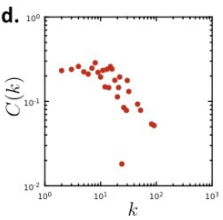
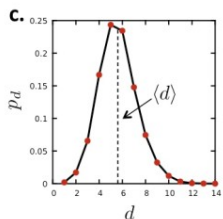
通过MSN一个月的对话活动构建。网络中有180M的用户（节点），1.3B的边（即连个用户之间至少发了一条信息）。我们可以看一下MSN网络的一些属性：

属性	结果	讨论
度分布 (Degree distribution)		严重倾斜。 $\bar{k} = 14.4$ 。
聚类系数 (Clustering coefficient)		$\bar{C} = 0.11$
连通分量 (Connected components)		强连通，MSN网络的连通分量中包含了99%的节点。
路径长度 (Path length)		平均路径长度为6.6。从图中可以看出，超过 10^{10} 对节点之间的距离是6，可能这就是所谓的认识6个人，就认识了全世界吧hh。两个人之间最多通过30人就可以互相认识。

2.2 PPI网络（蛋白质相互作用的网络）

PPI网络中包含2018个结点（2018种蛋白质），2930条边。

属性	结果	讨论
度分布 (Degree distribution)		倾斜。 $\langle k \rangle = 2.9$ 。

属性	结果	讨论
		
聚类系数 (Clustering coefficient)		$\overline{C} = 0.12$
连通分量 (Connected components)		有185个连通子图，其连通分量中包含1647个结点（81%的结点）。
路径长度 (Path length)		平均路径长度为5.8。

在计算了这些网络的参数之后，这些参数有哪些实质的意义呢？这两个网络的参数也——聚类参数和路径长度也很接近。为了获得这些参数的意义，需要一些图模型作为benchmark。

3. 最简单的图模型——ER Random Graph Model（ER随机图模型）

在数学中，随机图是指由 随机过程 产生的图。

在图论的数学理论部分中，ER模型（Erdős-Rényi model）可指代两个密切相关的随机图生成模型中的任意一个。这两个模型的名称来自于数学家Paul Erdős（保尔•厄多斯）和Alfréd Rényi（阿尔弗烈德•瑞利），他们在1959年首次提出了其中一个模型，而另一个模型则是Edgar Gilbert（埃德加•吉尔伯特）同时并且独立于Erdős和Rényi提出的。在Erdős和Rényi的模型中，节点集一定、

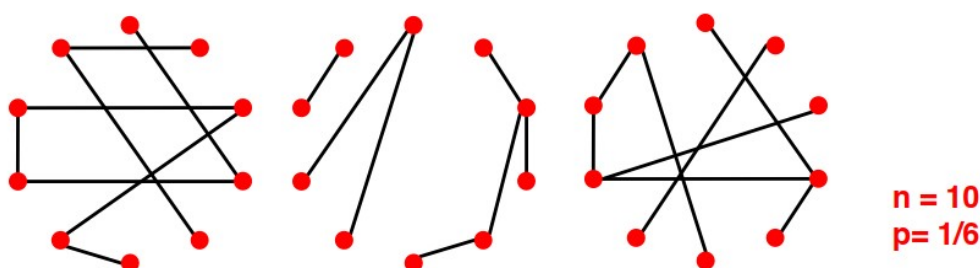
连边数也一定的所有图是等可能的；在Gilbert的模型中，每个连边存在与否有着固定的概率，与其他连边无关。在概率方法中，这两种模型可用来证明满足各种性质的图的存在，也可为几乎所有图的性质提供严格的定义。

ER随机图模型有两种，一个是 G_{np} 图，给定 n 个节点和节点之间生成边的概率 p ；一个是 G_{nm} 图，给定 n 个节点和 m 条边，这 m 条边随机在这 n 个节点间生成。

3.1 G_{np} 图

在模型 G_{np} 中，随机连接节点构成一个图。图中每个连边彼此独立，连接的概率为 p 。等价地，拥有 n 个节点、 M 个连边的所有图具有相同的概率。

需要注意的是，给定节点 n 和概率 p ，并不意味着生成唯一的图。下面是 $n = 10$ 和 $p = 1/6$ 时生成图的一些例子：



下面介绍 G_{np} 图的一些性质。

G_{np} 图的度分布 (Degree distribution)

G_{np} 图的度分布是二项分布 (binomial) 。

$$P(k) = C_{n-1}^k p^k (1-p)^{n-1-k}$$

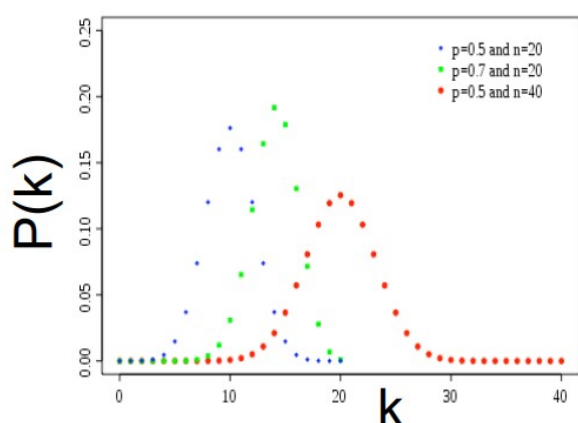
对于某个节点来说，度为 k 的概率为从剩下的 $n - 1$ 个点中选取 k 个点，并与之相连的概率。

由二项分布的性质，可以得到：

$$\bar{k} = p(n-1)$$

$$\sigma^2 = p(1-p)(n-1)$$

$$\frac{\sigma}{\bar{k}} = \left[\frac{1-p}{p} \frac{1}{n-1} \right]^{\frac{1}{2}} \approx \frac{1}{(n-1)^{1/2}}$$



根据大数定律，随着网络规模的增大，分布变得越来越窄（分布会越来越集中）——我们越来越确信一个节点的度在 \bar{k} 附近。

G_{np} 图的聚类系数 (Clustering coefficient)

G_{np} 图的聚类系数 (Clustering coefficient) 由公式 $C_i = \frac{2e_i}{k_i(k_i-1)}$ 得到。

我们首先得到 e_i 的期望值

$$E[e_i] = p \frac{k_i(k_i - 1)}{2}$$

这里 p 为一对节点相连的概率； $\frac{k_i(k_i-1)}{2}$ 表示节点 i 的邻居节点集能产生的边的数量。则

$$E(C_i) = \frac{2E[e_i]}{k_i(k_i - 1)} = p = \frac{\bar{k}}{n - 1} \approx \frac{\bar{k}}{n}$$

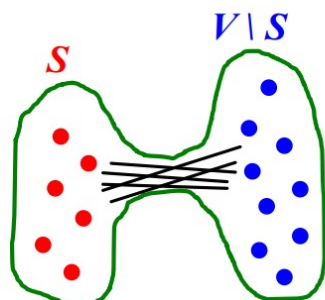
随机图的聚类系数通常都比较小。If we generate bigger and bigger graphs with fixed avg. degree \bar{k} (that is we set $p = \bar{k} \cdot 1/n$), then C decreases with the graph size n .

G_{np} 图的路径长度 (Path length)

G_{np} 图的路径长度 (Path length) 通过 扩展数 (expansion) 来衡量。

- Graph $G(V, E)$ has **expansion α** : if $\forall S \subseteq V$:
 $\# \text{ of edges leaving } S \geq \alpha \cdot \min(|S|, |V \setminus S|)$
- **Or equivalently:**

$$\alpha = \min_{S \subseteq V} \frac{\# \text{ edges leaving } S}{\min(|S|, |V \setminus S|)}$$



扩展数 α 的计算如图所示，它描述了图的任意节点集与剩余节点之间边的数量。对于图中节点 V 的任意子集 S ，从该子集中的节点指向其补集 $V \setminus S$ 中的节点的边
 $\# \text{ of edges leaving } S \geq \alpha \cdot \min(|S|, |V \setminus S|)$ 。扩展数 α 通常也用来衡量图的鲁棒性（可扩展性）。

简单理解为：任取图中节点的一个子集，相对应的从子集中离开的（也就是和这些节点相关的）最小节点数目。

或者还可以理解为：为了让图中一些节点不具备连接性，需要cut掉图中至少多少条边？

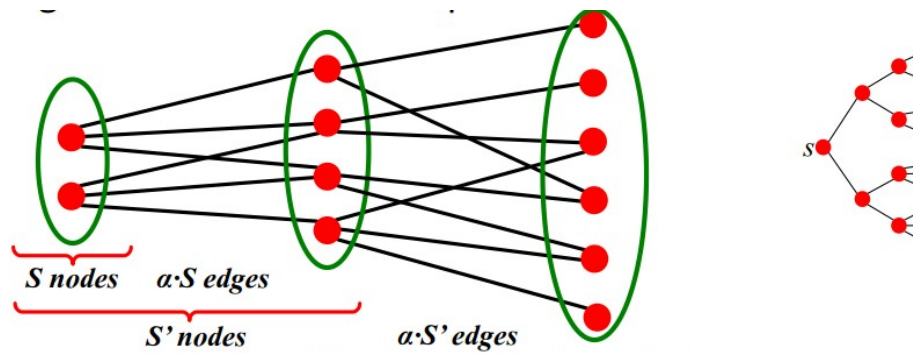
(answer：需要至少cut掉 $\alpha \cdot N$ 条边)

——摘自：[图机器学习 2.2-2.4 Properties of Networks, Random Graph](#)

如果一个图有 n 个节点，其扩展系数为 α ，对于所有的节点对来说，路径长度为 $O((\log n)/\alpha)$ 。

对于 G_{np} 图来说，存在一个常数 c ，使得 $\log n > np > c$ ， $\text{diam}(G_{np}) = O((\log n)/\log(np))$ 。 p 衡量的是两个节点之间随机生成边的概率， p 越大，遍历 G_{np} 图所需要的步数越小。

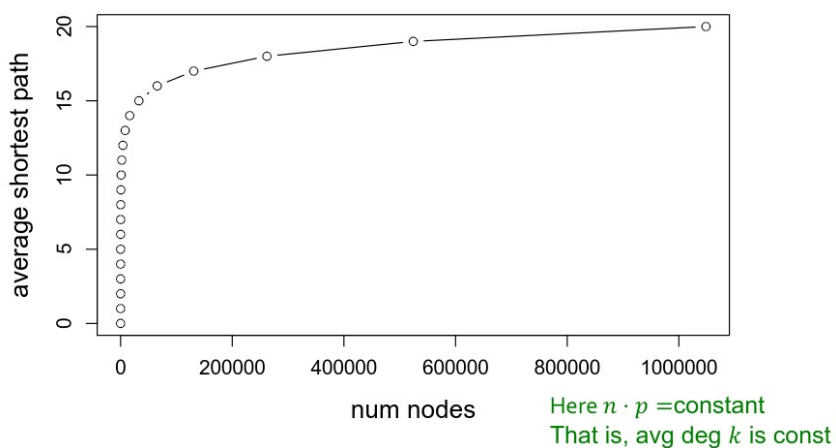
同时， G_{np} 图的可扩展性很好，所以通过广度优先搜索来遍历图中的所有节点相对比较容易：



其实我在看这一部分的时候理解起来比较费劲，而且在google上各种关键词搜索都没有搜索到讲的比较明确的资料。不过上面这张图还是画的比较明白的，可以结合图的 广度优先生成树/森林 来理解。

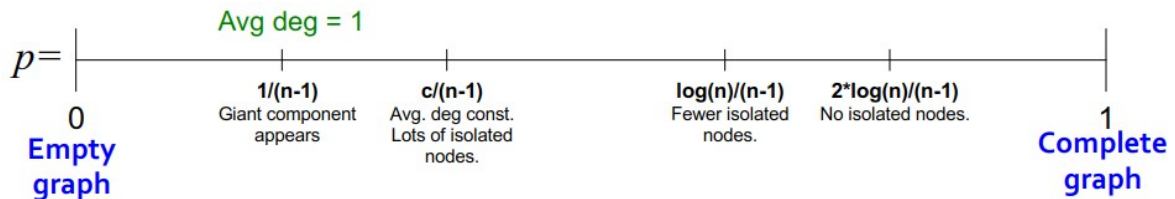
同时可以看到，在 p 一定的情况下，随着节点数的增长， G_{np} 图的平均最短路径趋于某一个值。。

Erdős-Renyi Random Graph can grow very large but nodes will be just a few hops apart

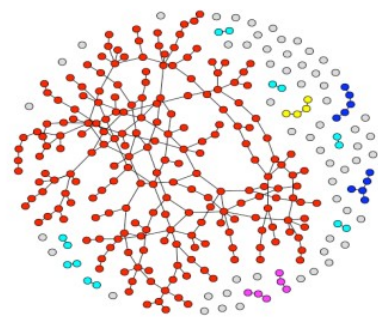
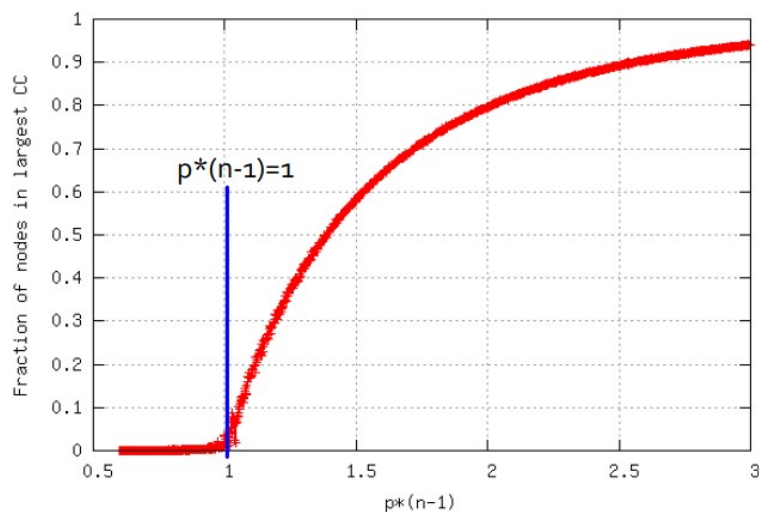


G_{np} 图的连通分量 (Connected components)

p 衡量的是两个节点之间随机生成边的概率， G_{np} 图的连通性也会随着 p 而改变：



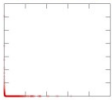
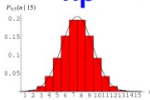
在 $p = 1/(n - 1)$ 时，度平均为1，出现最大连通子图，每个节点至少有一条期望边。



Fraction of nodes in the largest component

3.2 G_{np} 图与实际网络的比较

我们将MSN网络与 G_{np} 图模型进行比较：

	MSN	G_{np} <small>n=180M</small>	
Degree distribution:			✗
Avg. path length:	6.6	$O(\log n)$	✓
Avg. clustering coef.:	0.11	\bar{k} / n	✗
Largest Conn. Comp.:	99%	GCC exists when $\bar{k} > 1$.	✓
		$h \approx 8.2$	
		$C \approx 8 \cdot 10^{-8}$	
		$\bar{k} \approx 14$	

随机图模型与真实网络的比较：

- Giant connected component: 😊
- Average path length: 😊
- Clustering Coefficient: 😞
- Degree Distribution: 😞

随机图模型存在的一些问题：

- Degree distribution differs from that of real networks
- Giant component in most real networks does NOT emerge through a phase transition
- No local structure – clustering coefficient is too low

- 真实网络的度分布与随机图模型不同；
- 真实网络的最大连通子图并不是通过相变（phase transition）产生的。

- 由于没有局部结构——导致随机图模型的聚类系数过低。

更重要的一点，真实网络并不是随机图模型。那么，既然 G_{np} 图模型是错误的，我们了解和研究 G_{np} 图模型的性质有什么意义呢？

■ **If G_{np} is wrong, why did we spend time on it?**

- It is the reference model for the rest of the class
- It will help us calculate many quantities, that can then be compared to the real data
- It will help us understand to what degree a particular property is the result of some random process

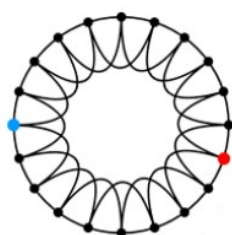
So, while G_{np} is WRONG, it will turn out to be extremely USEFUL!

4. 小世界网络模型 The Small-world model

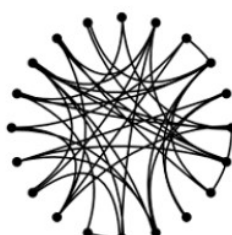
前面讲到，随机图模型其实不符合真实网络的分布。随机图模型正确模拟真实世界网络中的平均路径长度，但是低估了聚类系数。为了解决这个问题，我们引入第二个模型——小世界网络模型（The Small-world model）

小世界网络模型是一类具有**较短的平均路径长度**又具有**较高的聚类系数**的网络的总称。

我们怎样去构建一个小世界网络模型呢？可以先看下面这两张图：



High clustering
High diameter



Low clustering
Low diameter

Clustering implies edge “locality”

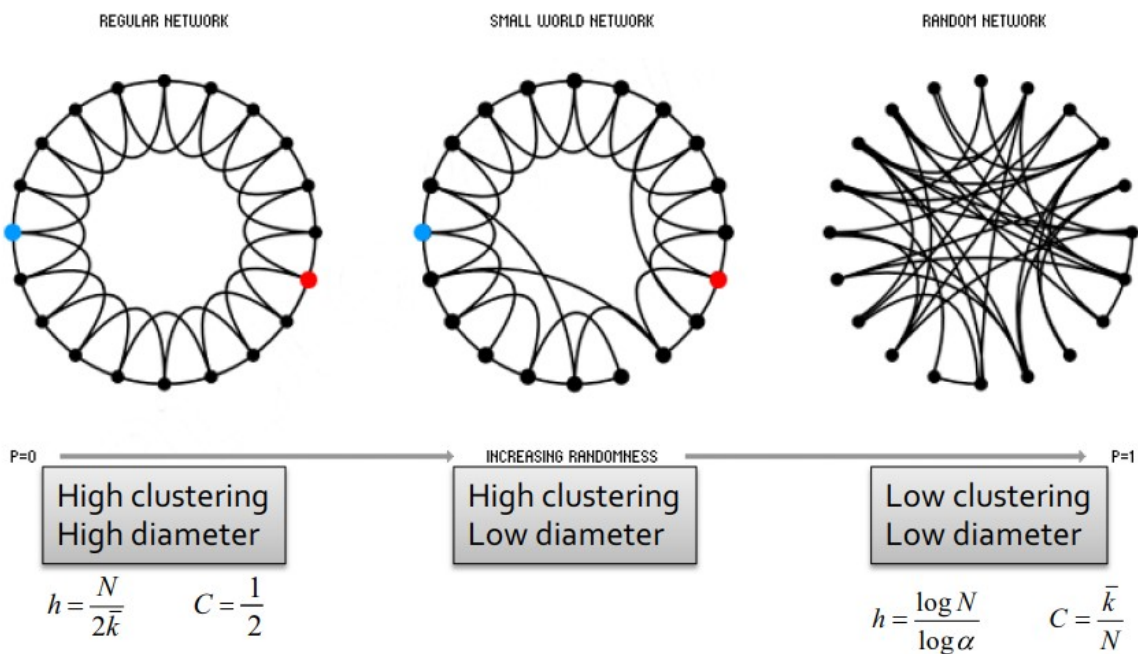
Randomness enables “shortcuts”

左边这张图保证了聚集性（比如在现实社会中，我的朋友的朋友就是我的朋友这种情况），但是这种时候图的直径就会比较大，就好比大家都认识，想要传递消息的

时候不能直接打电话，得有无数人中间传话一样；右边这张图是随机图模型，图的直径比较小，两两之间能够直接沟通的渠道变多了，但是缺少聚集性的特点。

这里我们介绍WS小世界模型构造算法 [Watts-Strogatz '98]：

- 1、**从一个环状的规则网络开始：网络含有 N 个结点，每个节点向与它最临近的 K 个节点连出 K 条边，并满足 $N \gg K \gg \ln(N) \gg 1$ 。
- 2、**随机化重连**：以概率 p 随机地重新连接网络中的每个边，即将边的一个端点保持不变，而另一个端点取为网络中随机选择的一个节点。其中规定，任意两个不同的节点之间至多只能有一条边，并且每一个节点都不能有边与自身相连。这样就会产生 $pNK/2$ 条长程的边把一个节点和远处的结点联系起来。改变 p 值可以实现从规则网络($p = 0$)向随机网络($p = 1$)转变。



Rewiring allows us to “interpolate” between a regular lattice and a random graph

小世界网络模型给我们描述真实的网络提供了一个比较贴切的模型：

■ The Watts Strogatz Model:

- Provides insight on the interplay between clustering and the small-world
- Captures the structure of many realistic networks
- Accounts for the high clustering of real networks
- Does not lead to the correct degree distribution

5. Kronecker Graph Model—Generating large realistic graphs

我们怎样有效地构建超大的图呢？这里引入一个概念：自相似性（Self-similarity）——物体总是和自身的某些局部是相似的。Kronecker Graph Model就是通过递归来生成网络。其递归的方法是通过Kronecker乘积来实现的，Kronecker乘积就是生成自相似矩阵的一种方式。

$A \otimes B$ 如果 A 是一个 $m \times n$ 的矩阵，而 B 是一个 $p \times q$ 的矩阵，克罗内克积则是一个 $mp \times nq$ 的分块矩阵

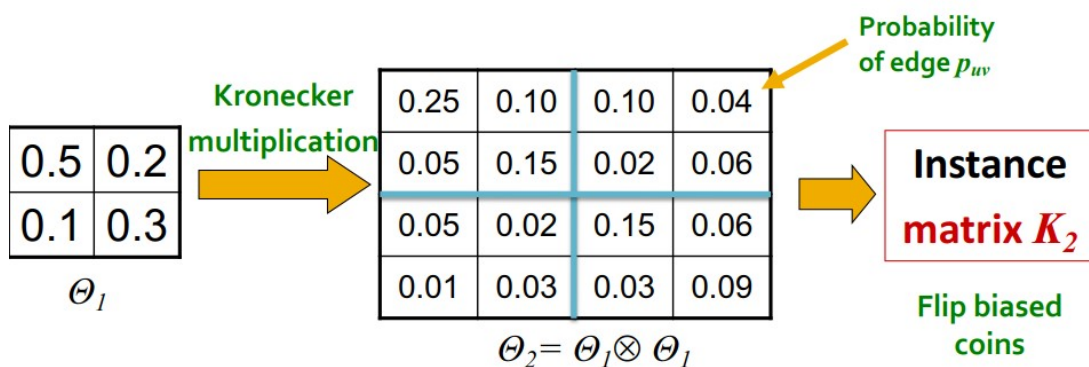
$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}.$$

Kronecker图就是通过一系列初始矩阵 K_1 与自身的Kronecker乘积来得到：

$$K_1^{[m]} = K_m = \underbrace{K_1 \otimes K_1 \otimes \cdots \otimes K_1}_{m \text{ times}} = K_{m-1} \otimes K_1$$

Stochastic Kronecker Graph Model 随机Kronecker图模型

- 生成 $N_1 \times N_1$ 的概率矩阵 Θ_1
- 递归计算第 k 个Kronecker乘积 Θ_k
- Θ_k 中的值 p_{uv} 表示节点 u 和节点 v 之间生成边的概率
- 按照这个概率产生实际的邻接矩阵，得到最后的大图模型



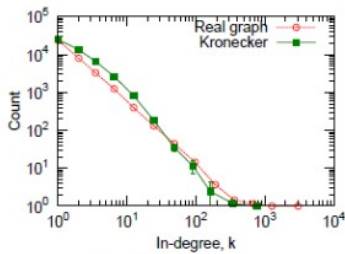
但是，这时候生成邻接矩阵的计算量回达到 N^2 次，速度会很慢。为了提高计算效率，我们引入drop操作——就是在 $n = 2^m$ 个结点中递归生成边。

快速克罗内克图生成算法（Fast Kronecker Generator Algorithm）（用于生成有向图）

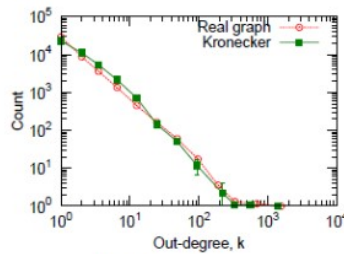
- 构建归一化概率矩阵 $L_{u,v} = \Theta_{u,v} / (\sum \Theta \text{矩阵内的素元素之和})$
- For $i = 1 \dots m$:
 - 令 $x = 0, y = 0$,
 - 通过概率矩阵中的概率值随机选取一个元素 (u, v)
 - 在 u, v 所在的象限内计算一个求和值 $X+ = u \cdot 2^{m-i}, Y+ = v \cdot 2^{m-i}$
- 在图 G 中添加边 (X, Y)

这是真实网络和克罗内克图模型的比较：

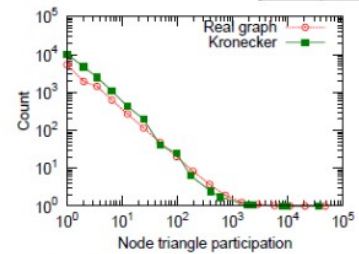
■ **Real and Kronecker are very close:**

$$\Theta_1 = \begin{bmatrix} 0.99 & 0.54 \\ 0.49 & 0.13 \end{bmatrix}$$


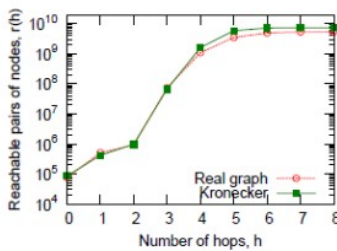
(a) In-Degree



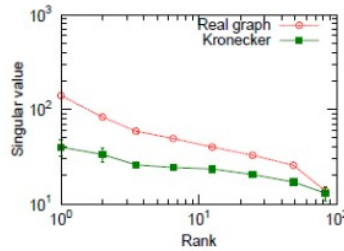
(b) Out-degree



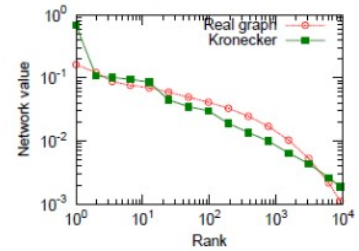
(c) Triangle participation



(d) Hop plot



(e) Scree plot



(f) "Network" value