LEC3 Social Network Analysis

1. **更深入的网络科学基础**
2. **羊群效应（Herding）与信息级联（Information Cascading）**
3. **社会网络中的信息扩散（Information Diffusion）**
4. **聚类与社区结构（Clusters and Communities）**
5. **使用 Python 进行网络分析（Python NetworkX）**

网络与网络结构（Networks and Network Structure）

> 社交网络（无论是在线还是物理世界）可以促进或限制信息、思想、疾病等的传播。

✅ 网络的作用：

- **媒介作用**：信息通过连接路径在节点间传递。
- **结构影响行为**：网络拓扑决定了动态过程是否容易发生。

🧩 示例：

- **电网故障级联**：一个电站停电 → 引发连锁反应 → 大面积断电
- **全球金融网络**：一家银行倒闭 → 触发系统性风险 → 全球金融危机

📌 **关键启示：**

> 网络不仅是静态图，更是动态系统的骨架。其结构决定了系统的稳定性、鲁棒性和脆弱性。

基本概念（Some Basic Concepts）

❓ 问题提出：

> 网络结构与复杂网络的动态行为之间有什么关系？

✅ 三大核心概念：

1. **度分布（Degree Distribution）**
2. **级联（Cascading）**
3. **聚类（Clusters）**

度分布（Degree Distribution）

📊 定义：

- 节点 i 的**度** k_i 是指它拥有的边数（即连接数量）。
- 度越大，说明该节点在网络中越"重要"。

📈 度分布函数 $P(k)$：

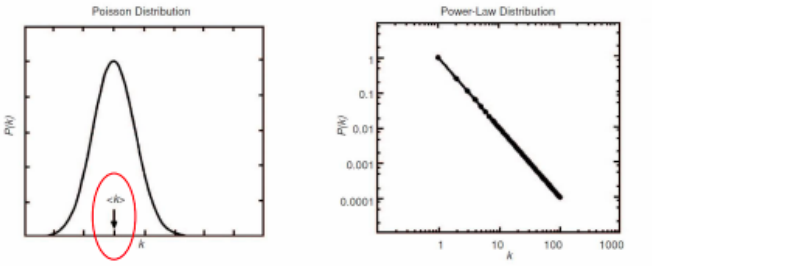
> 表示随机选择一个节点时，其恰好有 k 条边的概率。

$P(k) =$ 概率（随机节点的度为 k）

📌 平均度 $\langle k \rangle$：

> 所有节点度的平均值，反映网络的整体连接强度。

幂律分布（Power Law Distribution）



Homogeneous network with a characteristic scale <k>

Heterogeneous network (scale-free network) without characteristic scale

🔄 对比两种网络类型：

类型	特征	图形表现
同质网络（Homogeneous）	大多数节点具有相似度数	正态分布（钟形曲线）有典型尺度 $ \langle k \rangle $
异质网络（Heterogeneous / Scale-Free）	极少数节点拥有极高连接数（hubs）	幂律分布（直线）无特征尺度

📉 数学表达：

$P(k) \sim k^{-\gamma}$
其中 γ 通常在 2~3 之间。

📌 **现实意义**：

- 在互联网中，少数网站（如 Google、Facebook）被大量链接；
- 在社交网络中，少数人拥有百万粉丝；
- 在科研合作中，少数科学家发表大量论文。

✅ 这种“长尾”结构被称为 **无标度网络（Scale-Free Network）**。

无标度模型（Scale-Free Models）

⚠️ 为什么复杂网络不是均匀的？

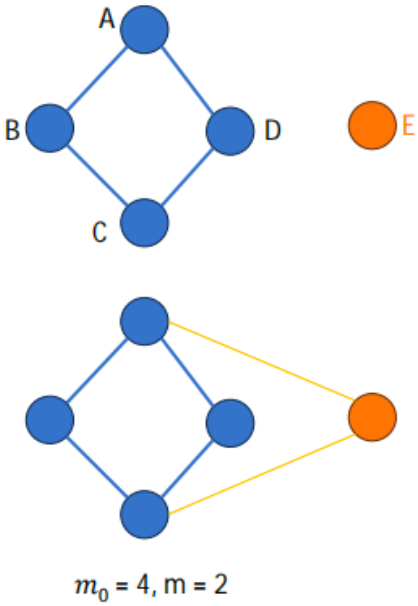
- 复杂网络是**开放且动态演化**的（不断新增节点）
- 存在“**富者愈富**”（rich-get-richer）的现象

✅ Barabási & Albert (BA) 模型的两大机制：

1. **增长（Growth）**：网络随时间持续扩展
2. **优先连接（Preferential Attachment）**：新节点更倾向于连接已有高连接度的节点

📌 这两个机制共同解释了为何真实网络呈现幂律分布。

BA 模型算法（BA Scale-Free Model Algorithm）



🧩 参数设定：

- m_0：初始网络的节点数（例如 4 个）
- m：每个新加入的节点会创建 m 条边（例如 $m = 2$）

🔄 算法步骤：

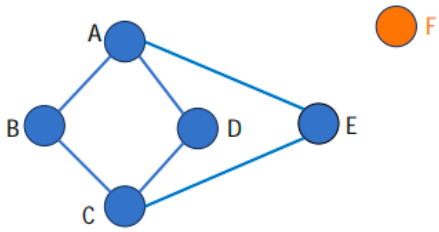
1. 从一个小网络开始（如 4 个节点构成完全图）
2. 每次添加一个新节点（如 E、F）
3. 新节点连接到现有节点，连接概率与该节点当前度成正比：

$$p_i = \frac{k_i}{\sum_j k_j}$$

其中 k_i 是节点 i 的当前度。

📌 **例子**：当 F 加入时，A 和 C 的度最高 → 更可能被选中连接。

BA 模型实例演示



F 加入网络

初始网络：A, B, C, D, E（共 5 个节点）

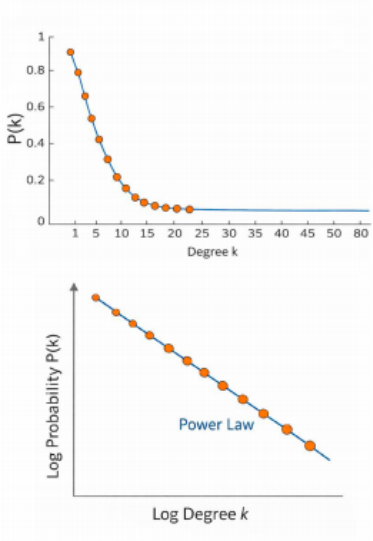
- 总度数：$\sum k_j = 12$
- 各节点连接概率：
  - $p(A) = 3/12 = 0.25$
  - $p(B) = 2/12 \approx 0.167$
  - $p(C) = 3/12 = 0.25$
  - $p(D) = 2/12 \approx 0.167$
  - $p(E) = 2/12 \approx 0.167$

👉 所以 A 和 C 最有可能被 F 连接。

🔁 “富者愈富”效应：

- 一旦 F 连接到 A，A 的度增加 → 下一次被选中的概率更高
- 形成正反馈循环 → 产生“超级节点”（hubs）

BA 模型结果



📈 度分布特性：

- 随着网络增长，**度分布保持不变**（形状不随规模变化）
- 即使网络变大，仍符合幂律分布 $P(k) \propto k^{-\gamma}$
- 因此称为“**无标度状态**”（scale-invariant state）

✅ 关键结论：

> BA 模型成功复现了真实复杂网络的幂律特性，揭示了其背后的自组织机制。

羊群效应与信息级联（Herding and Information Cascading）

🐑 羊群效应（Herding）：

> 当个体做决策时，不仅依赖自己的私人信息，还会参考他人的行为。

举个例子：

- 你在餐厅排队，看到很多人在等 → 你会认为这家店好吃 → 决定加入队伍
- 即使你没尝过，也会模仿别人的行为

📊 两种信息来源：

1. **私人信息**：你自己的知识、经验
2. **社会信息**：他人做出的选择（公开行为）

📌 **关键点**：

> 有时，群体选择的信息比个人私有信息更有说服力。

信息级联（Information Cascading）

🔄 定义：

> 信息级联是指个体基于前人决策而做出相同选择的过程，导致集体行为迅速蔓延。

✅ 特征：

- 决策是**顺序发生的**（sequential）
- 早期行动者影响后期者
- 可能形成“盲目跟随”现象

💡 实际案例：

- 股票市场：某只股票突然上涨 → 投资者跟风买入 → 推动价格进一步上升
- 社交媒体：一条帖子被转发 → 更多人看到 → 越来越多的人参与讨论
- 流行病：一人感染 → 周围人注意 → 出现恐慌性抢购口罩

🧠 深层含义：

> 人们常常放弃独立判断，转而信任群体行为 → 导致非理性繁荣或崩溃。

✅ 总结

1. 网络结构决定动态行为

- 网络不只是连接图，更是信息流动的通道
- 结构会影响传播速度、稳定性、抗灾能力

2. 复杂网络的普遍特征：幂律分布

- 少数"枢纽"控制大部分流量
- 由"增长 + 优先连接"机制产生（BA 模型）

3. 人类行为受网络影响：羊群效应与信息级联

- 个体并非孤立决策，而是受到他人行为的影响
- 信息可以在网络中快速扩散，甚至失控

💡 核心思想总结

> 我们生活在由连接构建的世界里：
> 你的朋友是谁？他们做了什么？
> 决定了你能否获得信息、是否会被影响、是否会成为下一个"爆款"。

> "网络塑造行为，行为反过来塑造网络——这是一个自我强化的循环。"

信息级联（Information Cascading）是如何发生的？

✅ 三个必要条件：

1. **个体按顺序做决策（Sequential decisions）**
   - 比如顾客一个接一个进书店选书
2. **无法获取他人的私人信息（No access to private information）**
   - 你不知道别人是否真的喜欢这本书
3. **行动空间有限（Limited action space）**
   - 只能选择"采纳"或"拒绝"，比如买/不买、用/不用

📌 常见例子：

- 去哪家餐厅吃饭？
- 读哪本书？

- 用哪个手机App？

📌 这些场景中，我们常常不是基于自己的判断，而是看“别人怎么选”。

信息级联示例1 —— 超市果汁选择

🍹 场景：

在超市里面对多种相似口味的果汁，你会怎么选？

> 如果所有果汁看起来差不多，你可能不会仔细研究成分，而是：
>
> - 看谁正在拿哪种
> - 看货架上哪种卖得快
> - 看广告推荐

👉 这就是典型的**信息级联**：你根据别人的购买行为来决定自己选什么。

📌 **关键点**：即使你没有尝过，也会相信“多数人选择的就是好的”。

信息级联示例2 —— 书店购书

📚 场景：

书店有两本类似的书 B_1 和 B_2，但没人知道哪本更好。

- C_1 是第一个顾客 → 随机买了 B_1
- C_2 来了 → 看到 C_1 买了 B_1 → 认为 B_1 更好 → 也买 B_1
- C_3 来了 → 看到前两人买 B_1 → 更倾向于买 B_1
- ……

✅ 结果：尽管 B_2 可能其实更好，但由于前面的人选择了 B_1，后续所有人都跟着选 B_1。

🧠 深层含义：

> 早期少数人的选择可以引发大规模模仿，形成“虚假共识”或“路径依赖”。

信息级联示例3 —— 即时通讯软件选择

💬 场景：

两个即时通讯工具：WhatsApp vs WeChat

- 你有两个朋友用了 WhatsApp，另一个用了 WeChat
- 但如果你的朋友都用 WeChat，你会更愿意加入 WeChat
- 因为你需要和他们沟通

👉 这是一种**网络效应**（Network Effect）驱动的信息级联。

🔄 正反馈循环：

- 有人用 → 更多人加入 → 使用者越多 → 吸引力越大 → 更多人加入

📌 **结论**：平台的成功往往不是因为产品最好，而是因为“大家都用”。

信息级联的原因

类型	描述	示例
信息性原因（Informational）	从他人行为中获得有用信息，帮助自己做决定	选餐厅、选书
直接利益原因（Direct Benefit）	与他人一致带来实际好处（兼容性、效率等）	用同一个操作系统、文件格式

✅ 区别：

- **信息性**：我学到了新知识（“原来这家店不错”）
- **直接利益**：我获得了便利（“大家都能看到我的消息”）

📌 两者常同时存在，共同推动信息级联。

信息扩散模型（A Model of Information Diffusion）

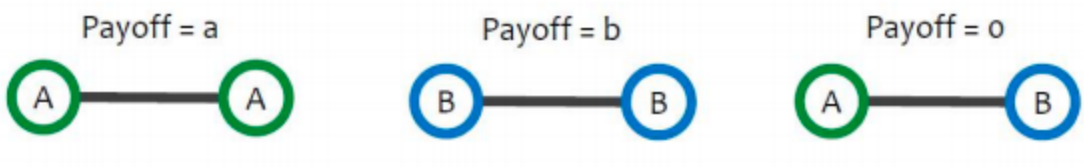
🎯 目标：

建立一个简单的数学模型来理解信息如何在网络中传播。

🧩 基本设定：

- 每个人必须在 A 或 B 之间选择
- 一个人只受邻居影响
- 若两个人是朋友，他们希望彼此行为一致（协调）

收益函数（Payoffs）



📊 收益规则（Payoff Matrix）：

	Y 选 A	Y 选 B
X 选 A	a, a	0, 0
X 选 B	0, 0	b, b

📌 解释：

- 如果双方都选 A → 各得收益 a
- 如果双方都选 B → 各得收益 b
- 如果不同 → 收益为 0（冲突）

📌 这是一个典型的 **协调博弈（Coordination Game）**

扩展到网络结构

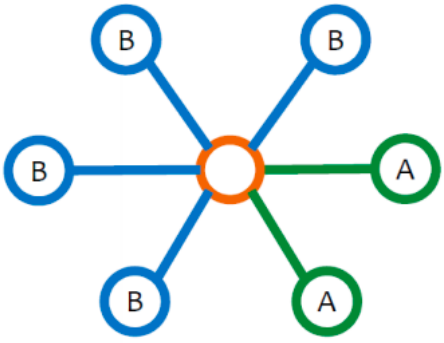
🌐 在社交网络中：

- 每个节点（人）与其他多个邻居相连
- 每条边代表一次“协调游戏”
- 总收益 = 所有邻居关系上的收益之和

📈 图形说明：

中心节点连接6个邻居，每个邻居都在进行独立的协调博弈。

网络中的收益计算



🧮 假设：

- p：邻居中选择 A 的比例
- $1-p$：邻居中选择 B 的比例

✅ 个人选择 A 的总收益：

$$\text{Payoff}_A = p \cdot a + (1-p) \cdot 0 = pa$$

✅ 个人选择 B 的总收益：

$$\text{Payoff}_B = (1-p) \cdot b + p \cdot 0 = (1-p)b$$

🔹 第十页：最优选择规则

✅ 决策原则：

> 当且仅当选择 A 的收益大于选择 B 时，才应选择 A：

$$pa > (1-p)b$$

整理得：

$p > \frac{b}{a+b}$

即：

> 如果邻居中选择 A 的比例超过阈值 $\frac{b}{a+b}$，就应该选择 A。

🔹 第十一页：临界阈值分析

📊 不同情况下的策略建议：

条件	决策逻辑
若 $a = b$	应该跟随大多数人（$ p > 0.5 $）→ “少数服从多数”
若 $a > b$	A 更有价值 → 即使只有少量人选 A，你也应该选 A
若 $a < b$	B 更有价值 → 需要更多人支持 A 才值得选

✅ 实际意义：

- 如果你用的 App 很流行（高 a），哪怕别人没用，你也可能坚持使用
- 如果某个技术标准很通用（高 b），即使现在少人用，未来也可能成为主流

🎯 现实启示：

- 企业营销可通过“制造热度”引发信息级联（如首发优惠、KOL 推荐）
- 政府应对谣言时需抢占“先发优势”，避免公众被错误信息引导
- 个人决策时要警惕“盲从”，识别是否存在真正的价值信号

> 你今天的选择，也许只是别人昨天选择的结果；而你的选择，又可能成为明天无数人效仿的理由。
>
> **这就是网络的力量。**

链式反应与平衡（Chain Reaction & Equilibrium）

✅ 核心观点：

> 当某个行为（如使用新App、购买新产品）在初始阶段被少数人采纳，并且该行为具有足够吸引力时，会引发连锁反应，最终达到一个稳定状态——即均衡（Equilibrium）。

📌 关键条件：

- **若 a > b**：选择 A 的收益高于 B → A 更具吸引力
- **存在少量初始采纳者（initial members）选择了 A**

🔄 链式反应过程：

1. 初始用户选择 A
2. 他们的邻居看到后，因收益更高或受群体影响而也选择 A
3. 这些新用户再影响更多人.....
4. 逐层扩散，形成“雪崩效应”

✅ 停止条件：

> 当所有节点都选择了 A，不再有人愿意改变 → 系统进入均衡状态（equilibrium）

📌 **注意**：这里假设所有人都理性地最大化自身收益，且决策顺序合理。

两种可能的均衡状态

🧩 两种结果：

1. 完全级联（Complete Cascade）

- 初始采纳者启动行为 A
- 行为逐渐传播到**整个网络**
- 最终所有节点都采用 A

✅ 示例：

- 新社交平台（如微信）刚推出时，只有少数人用，但很快席卷全国

2. 不完全级联（Incomplete Cascade）

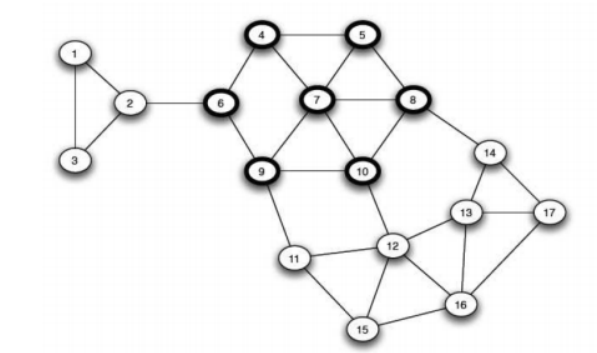
- 初始采纳者触发传播，但只扩散了一部分
- 某些节点仍然坚持原行为 B
- 传播中途停止

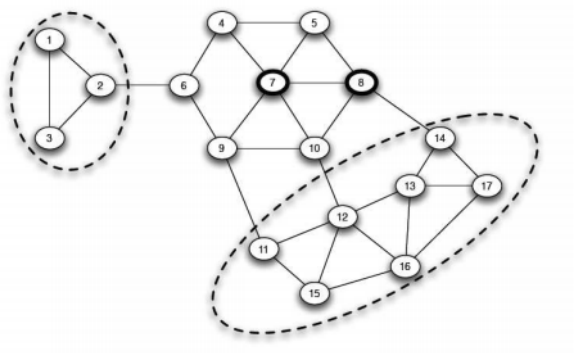
✅ 示例：

- 某款手机操作系统更新，部分用户升级，但很多人仍保留旧版本

📌 **关键问题**：为什么有些级联能覆盖全网，而有些却半途而废？

聚类与级联（Clusters and Cascades）





❓ 为什么链式反应会停止？

> 因为网络中存在紧密连接的子群（clusters），它们对外部影响有很强的“抵抗性”。

🎯 核心思想：

> 在一个高度内聚的子网络中，如果大多数成员都坚持原有行为（B），那么即使外部有很多人选择 A，也无法说服他们改变。

📌 图中展示了一个复杂网络，其中某些区域节点之间连接密集（黑色边多），这些就是潜在的“集群”（clusters）。

什么是聚类？（Definition of Clusters）

🧱 定义：

> 一个密度为 p 的聚类（cluster） 是一组节点，满足：
>
> > 每个节点至少有其邻居的 p 比例也在这个集合中。

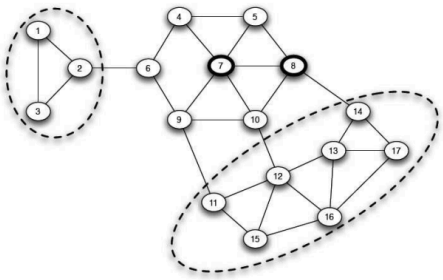
📊 数学表达：

对于任意节点 v 在聚类中：
$\frac{\text{内部邻居数}}{\text{总邻居数}} \geq p$

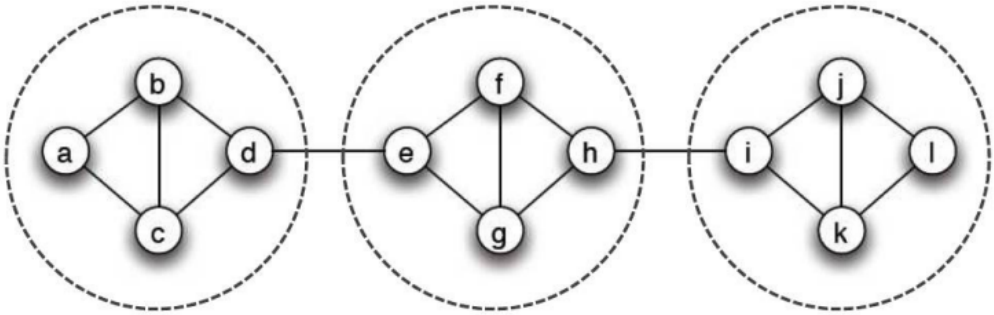
✅ 例子：

- 若 $p = 2/3$，则每个节点至少有 2/3 的朋友也在同一聚类中
- 聚类越密（p 越大），越难被外部影响打破

📌 图中虚线圈出的部分就是不同密度的聚类。



：聚类示例（Examples of Clusters）



🌀 图中三个独立子图：

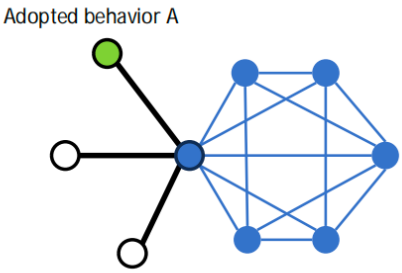
- 左：a-b-c-d 构成一个四边形，每个节点有两个内部邻居 → 密度 $= 2/3$
- 中：f-g-h-e 构成另一个四边形 → 同样密度 $= 2/3$
- 右：i-j-k-l 构成三角形加一节点 → 密度也为 $2/3$

📌 所有子图都是**高密度聚类**，彼此之间连接稀疏。

⚠️ 含义：

- 外部信息很难穿透这些“封闭社群”
- 即使外面的人都用了 A，这群人也可能继续用 B

聚类如何阻碍级联？



🧩 场景分析：

- 有一个中心节点（蓝色）连接着多个外围节点（绿色）
- 中心节点属于一个**内部高度连接的聚类（红色圆圈）**
- 外围节点已采纳行为 A（绿色）

💡 问题：

> 是否能让整个聚类都采纳 A？

🚫 结论：

> 很难！因为：

- 聚类内部成员大多彼此相连
- 即使个别成员被外部影响，也会受到“内部压力”回归原行为
- 尤其当聚类密度很高时，外部影响几乎无效

📌 **现实类比**：

- 保守社区 vs 科技爱好者群体
- 政治立场不同的朋友圈
- 不同文化背景的群体

聚类与级联的数学关系

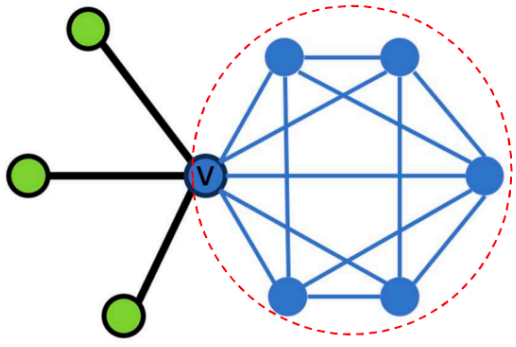
🧮 设定：

- 有一组初始采纳者选择了行为 A
- 其余节点需要满足阈值 q 才会采纳 A
  - 即：当邻居中选择 A 的比例 ≥ q 时，才会切换到 A

🔍 推理：

> 如果剩余网络中存在一个密度大于 $1 - q$ 的聚类，则：
>
> ➤ 初始采纳者**无法引发完全级联**

✅ 为什么？



- 对于聚类中的任一节点 v：

- $k(v)$：总邻居数
- $k_{\text{outside}}(v)$：外部邻居数（可能选 A）
- $k_{\text{inside}}(v)$：内部邻居数（很可能仍选 B）

$$\frac{k_{\text{outside}}(v)}{k(v)} < q$$

$$\Longrightarrow \quad \frac{k_{\text{inside}}(v)}{k(v)} > 1 - q$$

$$(\because k_{\text{inside}}(v) = k(v) - k_{\text{outside}}(v))$$

👉 这意味着：**内部连接太强，外部信号不足以突破门槛**

反向推理 —— 无完全级联 ⇒ 存在高密度聚类

🔄 重要结论（逆否命题）：

> 如果初始采纳者未能引起完全级联（complete cascade），那么：
>
> ➤ 剩余网络中一定存在一个**密度大于** $1 - q$ 的聚类。

🧠 实际应用：

- 想要推广新技术？先识别网络中的“顽固聚类”
- 想要阻止谣言传播？加强关键聚类的内部一致性
- 想要实现全面变革？必须打破高密度社群的隔离

✅ 总结：核心逻辑链条

概念	内容
链式反应	初始采纳者 → 影响邻居 → 逐层扩散 → 全面采纳
均衡状态	所有节点都做出一致选择，不再变化
完全级联	整个网络都采纳新行为
不完全级联	仅部分采纳，其余仍维持旧行为

概念	内容
聚类（Cluster）	内部连接紧密的子网络，具有强内聚力
聚类阻碍机制	高密度聚类可抵御外部影响，导致级联中断

💡 一句话总结：

> 信息不能轻易穿透“铁板一块”的群体；真正的变革，往往发生在那些没有“抱团取暖”的地方。

🎯 **现实启示**：

- 企业营销：不要只盯着“意见领袖”，更要关注“沉默多数”是否构成阻塞聚类
- 社会政策：推动改革需先瓦解保守派的内部联系
- 网络安全：防止病毒传播的关键是识别并隔离高密度恶意节点群

🧠 **终极思考**：

> 网络不仅是连接的集合，更是社会结构的映射。
>
> 正是这些看不见的“聚类”，决定了一个想法能否成为主流，还是注定沉没于角落。

网络中的社区（Community in Network）

🌐 核心观点：

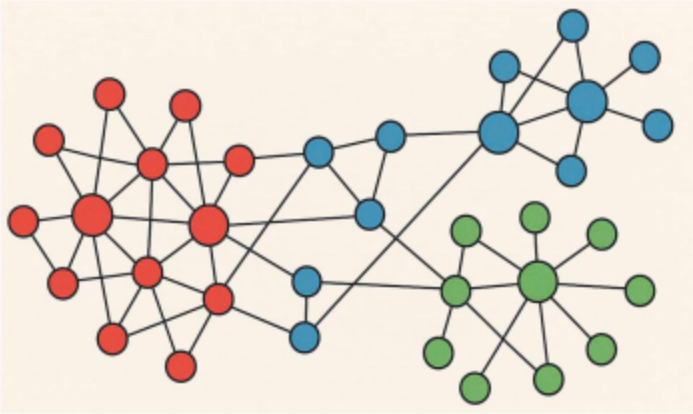
> 人们在现实中自然形成各种群体：家庭、朋友圈、工作团队、俱乐部等。

✅ 特征：

- **互动不均匀**：我们不会和所有人平等交往
- **影响集中**：信息、意见、行为往往在小圈子里传播
- **塑造行为**：社区影响我们的选择、态度和决策

🎯 目标：

> 如何从网络结构本身（即连接图）中检测并量化这些社区？



📌 图中展示了一个典型的社交网络，不同颜色代表不同的社区（如红、蓝、绿），每个社区内部连接密集，外部连接稀疏。

定义社区（H1 和 H2）

✅ H1: 基本假设（Fundamental Hypothesis）

> 网络的社区结构唯一地编码在其拓扑结构（wiring diagram）中。

👉 意思是：只要看谁跟谁连，就能发现“真实”的社区划分 —— 这个“真实”称为 **ground truth**。

> ⚠️ 注意：这是理想化的假设，现实中可能有多个合理划分。

✅ H2: 连通性假设（Connectedness Hypothesis）

> 社区对应于一个连通子图（connected subgraph）。

👉 即：同一社区内的成员必须可以通过其他成员互相到达。

📌 举个例子：

- 如果橙色节点之间彼此相连，它们构成一个社区；
- 若有两个孤立的橙色团块（无边连接），则应视为两个独立社区。

✅ H3: 密度假设（Density Hypothesis）

> 社区是网络中局部密集的邻域。

👉 更具体地说：

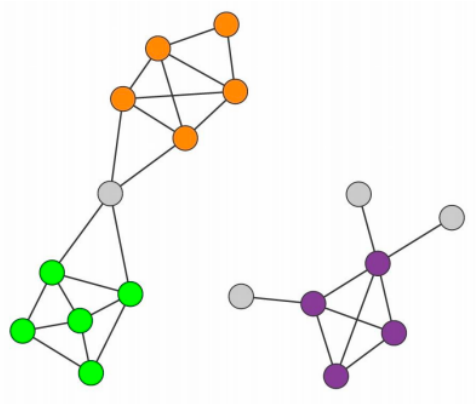
> 同一社区内的节点比跨社区的节点更有可能相互连接。

📌 这是最关键的假设之一，也是大多数社区检测算法的基础。

✅ 例如：

- 你朋友的朋友大概率也是你的朋友 → 内部连接多
- 你不认识的人通常也不会成为你的朋友 → 外部连接少

连通性假设详解（H2）



🧩 图解说明：

- 橙色节点组成一个连通子图 → 是一个社区
- 绿色节点也构成一个连通子图 → 是另一个社区
- 紫色节点同样 → 另一个社区

❌ 不能的情况：

- 如果两个子图之间没有边，就不能合并成一个社区
- 即使它们都用同一种颜色，也不能算作同一个社区

📌 所以：**社区必须是连通的**，否则就是多个独立社区。

密度假设详解（H3）

📊 核心思想：

> 社区内节点之间的连接概率 > 跨社区连接概率

✅ 示例：

- 橙色节点之间有很多边 → 高内部连接
- 橙色与绿色之间只有少量边 → 低外部连接
- 绿色与紫色之间几乎没有边 → 几乎无交叉

📌 这种“内部稠密、外部稀疏”的模式正是社区的本质特征。

团（Clique）作为社区

🔗 定义：

> 团（clique） 是一个完全子图：其中任意两个节点都有边相连。

📌 图中有三个例子：

- 三角形（3-node clique）
- 正方形（4-node clique）
- 五角星状（5-node clique）

⚠️ 但注意：

> 社区不一定是完整的团！

👉 因为现实世界中很少有人和社区里每个人都直接相连。

📌 例如：你朋友圈里的人都认识，但未必每个人都加了微信好友。

✅ 所以：**团是社区的一种极端情况，但不是必要条件**。

强社区 vs 弱社区

📌 定义两个概念：

🔹 内部度（Internal Degree）k_i^{int}

> 节点 i 与其他同社区节点之间的连接数

🔹 外部度（External Degree）k_i^{ext}

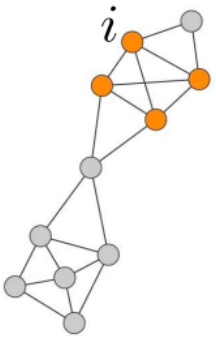
> 节点 i 与其他社区节点之间的连接数

✅ 判断标准：

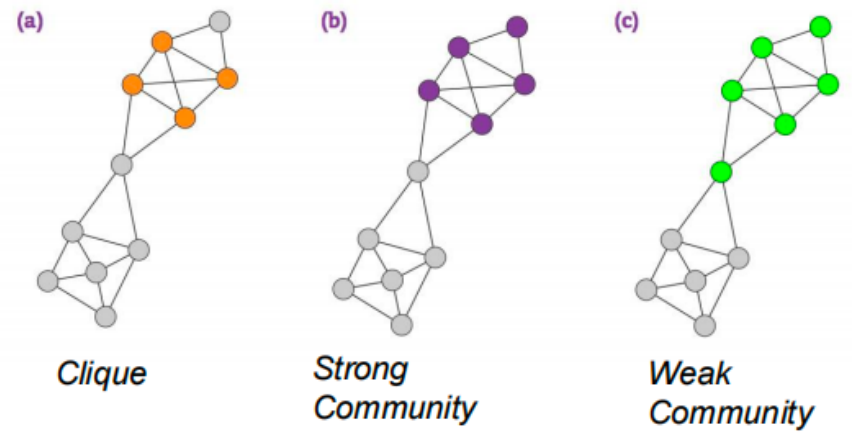
- 若 $k_i^{\text{ext}} = 0$：所有邻居都在本社区 → 是好社区成员
- 若 $k_i^{\text{int}} = 0$：所有邻居都不在本社区 → 应该分到别处

📌 图中灰色节点：有一个橙色邻居 → 属于橙色社区？需进一步判断

$$k_i^{int} = 3$$
$$k_i^{ext} = 1$$



团、强社区、弱社区对比



(a) 团（Clique）

- 所有节点两两相连 → 最强的内部连接
- 例如：正方形是最高阶的团

(b) 强社区（Strong Community）

- 每个节点的内部度 > 外部度
- 整体上：总内部度 > 总外部度
- 表现为“内聚性强”

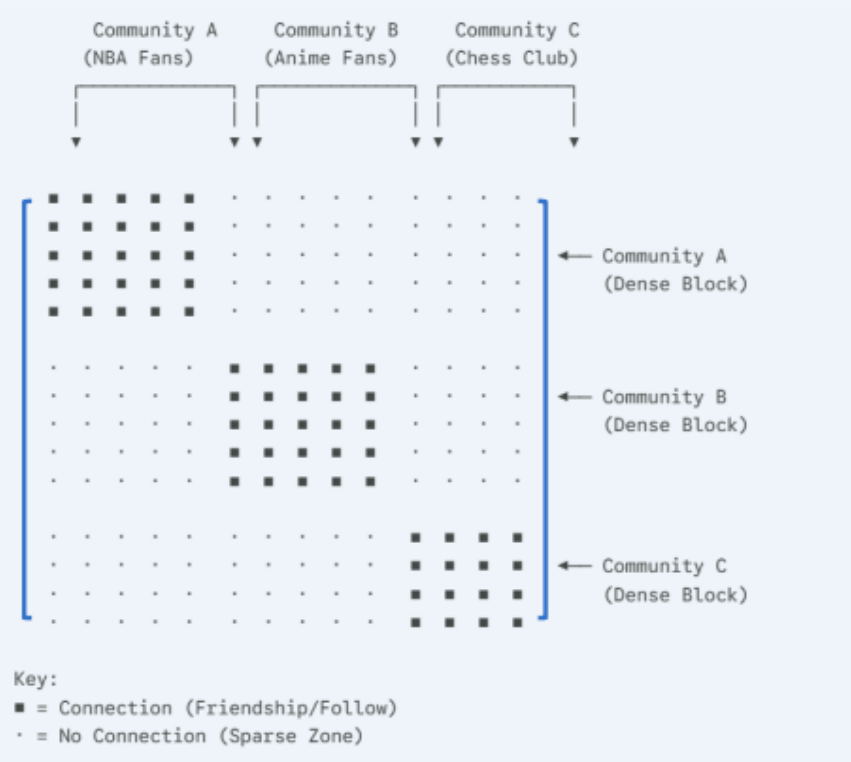
$$k_i^{\text{int}}(C) > k_i^{\text{ext}}(C)$$

(c) 弱社区（Weak Community）

- 总内部度 强社区更有意义，因为它真正形成了“封闭圈子”。

$$\sum_{i \in C} k_i^{\text{in}}(C) > \sum_{i \in C} k_i^{\text{out}}(C)$$

可视化密度——“回音室效应”（Echo Chamber）



📊 矩阵表示法（Adjacency Matrix）

- 将用户按社区排序后，画出连接矩阵
- 同一社区的用户排在一起 → 对角线上出现“密集块”

🎯 “回音室”现象：

- 社区内用户频繁互动 → 形成“信息茧房”
- 社区间几乎无交流 → 成为“过滤气泡”（Filter Bubble）

📌 推荐算法的作用：

- 保持你在“块”内活跃（engagement）
- 偶尔带你跳到新块（exploration）

强社区：现实 vs 理论

🌐 现实观察：

> 在大型社交网络中，真正的“强社区”（$k_{\text{in}} > k_{\text{out}}$ 对所有节点）极为罕见

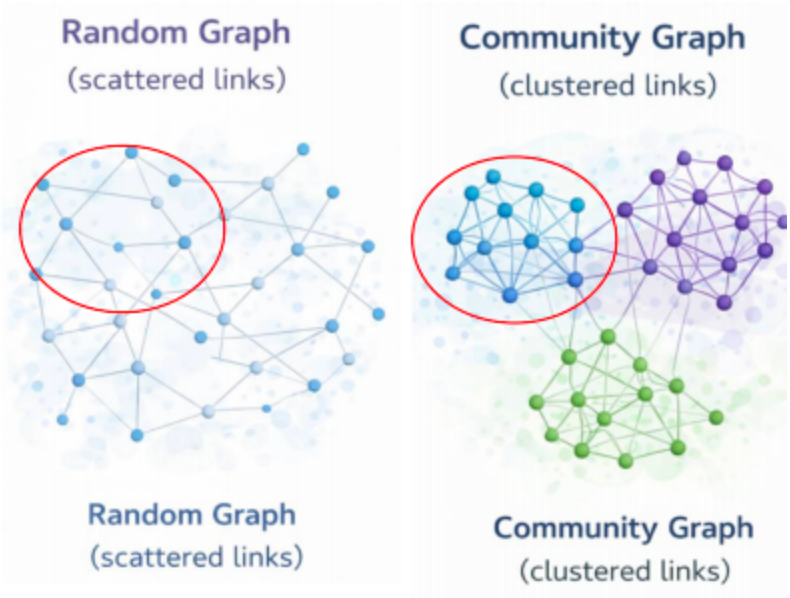
❌ 原因：

- 大多数人处于多个群体边缘（如同时是球迷、书迷、棋手）
- 不可能只和自己社区的人互动

✅ 解决方案：

> 我们不能再问：“这是一个社区吗？”（Yes/No）
> 而要问：“这个社区有多好？” → 需要一个数学评分函数

📌 这就引出了下一个重要概念：**模块度（Modularity）**



模块度（Modularity, M）

🎯 核心问题：

> 如何判断一个社区划分是否“有意义”？

✅ 模块度定义：

> M = 实际内部链接数 - 随机情况下预期的内部链接数

📌 概念上：

- 如果社区内部连接远多于随机期望 → 说明它是真实的、非偶然的
- 如果接近随机水平 → 分类无意义

模块度公式推导

🧮 公式：

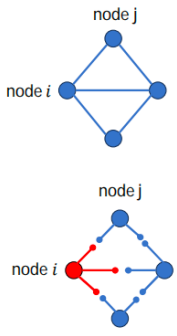
$M_c = \frac{1}{2L} \sum_{(i,j)\in C_c} (A_{ij} - p_{ij})$

其中：

- A_{ij}：邻接矩阵，1 表示有边，0 表示无边
- $p_{ij} = \frac{k_i k_j}{2L}$：在随机网络中，节点 i 和 j 之间存在边的概率
- L：总边数
- k_i, k_j：节点 i 和 j 的度

📌 p_{ij} 是"零模型"（Null Model）的关键：它假设连接仅由节点度决定，忽略结构。

零模型（The Null Model）解释 p_{ij}



🎯 目标：

> 在判断一个社区是否“有意义”时，我们需要知道：如果网络是随机连接的，两个节点之间有多大概率相连？

这个概率就是 pij*pij*，它来自**零模型**（Null Model）。

✅ 核心思想：

- 我们不关心“谁和谁连了”，而是关心：“在同样的度分布下，这些连接是不是比随机情况更密集？”
- 所以我们要构建一个**随机网络**，它的节点度与原网络相同，但边是随机分配的 → 这就是“零模型”
-

🧩 如何计算 p_{ij}？

- 节点 i 有 k_i 条“连接端口”（stubs）
- 节点 j 有 k_j 条“连接端口”
- 总共 $2L$ 个端口

→ 任一端口来自 i 的概率 ≈ $\frac{k_i}{2L}$

→ 于是 i 和 j 之间有边的概率：

$p_{ij} = \frac{k_i}{2L} \cdot \frac{k_j}{2L} \cdot (2L - 1) \approx \frac{k_i k_j}{2L}$

📌 这个 p_{ij} 就是我们用来比较“实际 vs 随机”的基准。

总模块度（Total Modularity）

📈 总模块度公式：

$M = \sum_{c=1}^{n_c} \left[\frac{L_c}{L} - \left(\frac{k_c}{2L} \right)^2 \right]$

其中：

- L_c：社区 c 内部的实际边数
- k_c：社区 c 中所有节点的度之和
- L：总边数

📌 两项含义：

- $\frac{L_c}{L}$：**覆盖（Coverage）** —— 实际有多少边在社区内
- $\left(\frac{k_c}{2L}\right)^2$：**惩罚项（Penalty）** —— 随机网络中预期的边数

📌 **M 越大，说明社区划分越好**

▼ **详细解释**

这两页幻灯片是社交网络分析中**社区检测（Community Detection）** 的核心内容，聚焦于 **“零模型”（Null Model）** 和 **“模块度”（Modularity, M）** 的数学原理。我们来逐页深入解释，并用**具体例子**帮助你理解。

🔹 第一页：零模型（Null Model）—— 解释 $ p_{ij} $

🎯 目标：

> 在判断一个社区是否“有意义”时，我们需要知道：如果网络是随机连接的，两个节点之间有多大概率相连？

这个概率就是 $ p_{ij} $，它来自**零模型**（Null Model）。

✅ 核心思想：

- 我们不关心“谁和谁连了”，而是关心：“在同样的度分布下，这些连接是不是比随机情况更密集？”
- 所以我们要构建一个**随机网络**，它的节点度与原网络相同，但边是随机分配的 → 这就是“零模型”

🧮 如何计算 $ p_{ij} $？

假设：

- 网络总共有 $ L $ 条边 → 总共 $ 2L $ 个“连接端口”（stubs）
- 节点 $ i $ 有 $ k_i $ 个 stubs
- 节点 $ j $ 有 $ k_j $ 个 stubs

推导过程：

1. **每个 stub 从节点 $ i $ 出发，连接到节点 $ j $ 的某个 stub 的概率是多少？**
   - 总共有 $ 2L - 1 $ 个其他 stub 可选（减去自己正在考虑的那个）
   - 其中属于节点 $ j $ 的有 $ k_j $ 个
   - 所以概率 ≈ $ \frac{k_j}{2L - 1} $
2. **但由于 $ 2L - 1 \approx 2L $，我们可以简化为：**
   $$
   p_{ij} = \mathbb{E}[A_{ij}] \approx k_i \cdot \frac{k_j}{2L} = \frac{k_i k_j}{2L}
   $$

📌 $ A_{ij} = 1 $ 表示有边，0 表示无边

👉 所以 $ p_{ij} $ 就是“在随机网络中，$ i $ 和 $ j $ 之间存在边的概率”。

📊 图解说明：

上图（蓝色菱形）：

- 四个节点构成完全图（clique），每条边都存在
- 实际上 $ A_{ij} = 1 $ 对所有对

下图（红色+蓝色）：

- 同样的四个节点，但只画出了红色的边
- 红色边表示“我们观察到的连接”
- 蓝色虚线表示“可能存在的随机连接”

✅ **重点**：我们不是看有没有边，而是比较：

> “实际存在的边” vs “随机情况下应该有的边”

✅ 举个例子：

网络数据：

节点	度 $ k_i $
A	3
B	2
C	4
D	1

- 总边数 $ L = (3+2+4+1)/2 = 5 $
- 总 stub 数 $ 2L = 10 $

计算 $ p_{AB} $：

$$
p_{AB} = \frac{k_A \cdot k_B}{2L} = \frac{3 \cdot 2}{10} = 0.6
$$

👉 意思是：在一个随机网络中，A 和 B 之间有边的概率是 **60%**

- 如果现实中 A 和 B 真的连了 → 不奇怪
- 如果没连 → 有点反常（低于预期）

🔷 第二页：总模块度（Total Modularity）

🎯 目标：

> 综合评估整个网络划分成多个社区的好坏程度

✅ 公式：

$$
M = \sum_{c=1}^{n_c} \left[\frac{L_c}{L} - \left(\frac{k_c}{2L} \right)^2 \right]
$$

其中：

- $ n_c $：社区总数
- $ L_c $：第 $ c $ 个社区内部的边数
- $ k_c $：第 $ c $ 个社区中所有节点的度之和

- $ L $：网络总边数

🧩 拆解公式含义：

项	名称	含义
$ \frac{L_c}{L} $	**Coverage（覆盖）**	实际有多少比例的边落在该社区内
$ \left(\frac{k_c}{2L} \right)^2 $	**Penalty（惩罚）**	随机网络中期望的内部边比例

👉 **模块度 $ M $ = 实际内部边占比 - 随机预期内部边占比**

> 若 $ M > 0 $：说明该社区比随机情况更稠密 → 是真实社区
>
> 若 $ M 随机预期 → 是真实社区**

💡 一句话总结：

> 真正的社区不是靠“看起来像一群”，而是靠“内部连接远多于随机预期”。

就像你在朋友圈看到一群人天天聊天，而别人只是偶尔互动 ——

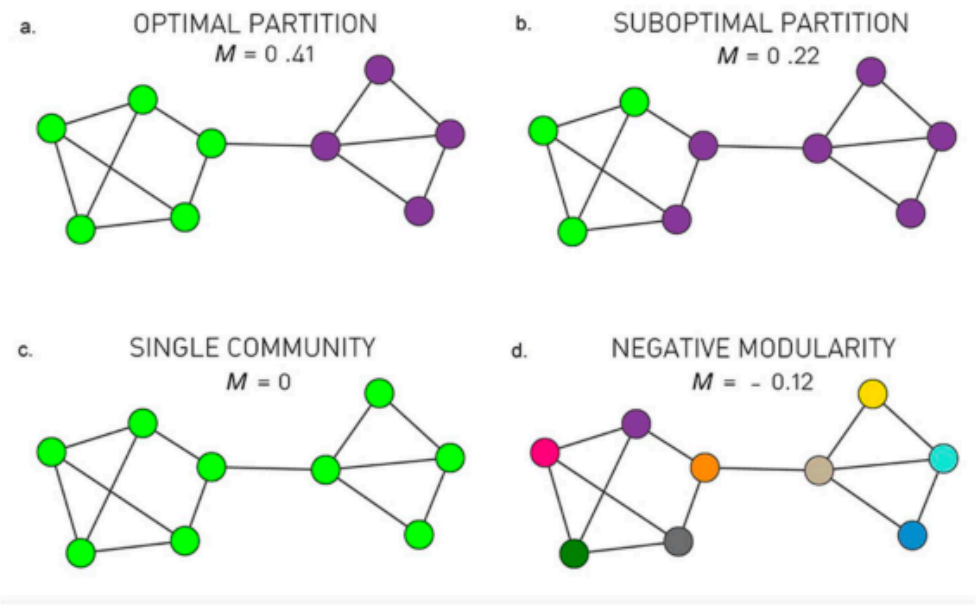
这才是社区的本质：**不是偶然聚集，而是结构性凝聚**。

🎯 **现实应用**：

- 推荐系统：识别兴趣小组 → 推送相关内容
- 疫情防控：发现高风险人群聚集 → 实施精准干预
- 金融监管：检测隐藏的关联交易 → 防范系统性风险

如果你想要我写一段 Python 代码来计算模块度，也可以告诉我！

网络划分示例



🎯 使用模块度评估不同划分：

划分	模块度 M	说明
a. 最优划分	M = 0.41	清晰区分两个社区，高度凝聚
b. 次优划分	M = 0.22	能看出趋势，但不够精确
c. 单一社区	M = 0	所有节点在一个组，无结构
d. 负模块度	M = -0.12	划分方式比随机还差，完全错误

📌 结论：

> 模块度是一个有效的评价指标，可用于自动寻找最佳社区结构。

Python NetworkX

🎯 介绍

> NetworkX 是一个用于创建、操作和研究复杂网络结构的 Python 库。

✅ 特点：

- 功能强大且易学

- 广泛应用于数据科学、人工智能、Web 开发和教育

🔗 相关库推荐：

库名	用途
`NetworkX`	图结构建模与分析
`Pandas`	数据处理（如读取 CSV 文件）
`Matplotlib`	可视化图表
`igraph`	高性能图分析工具
`Graph-tool`	适用于大规模网络

📌 官网：https://networkx.org

🔹 第二页：用 NetworkX 创建图

```
import networkx as nx

G = nx.DiGraph()  # 创建有向图
G.add_edges_from([
    (1, 3), (1, 4), (1, 5), (1, 6),
    (2, 4), (2, 5), (3, 1)
])

print("Nodes:", G.nodes())
print("Edges:", G.edges())
```

✅ 输出结果：

```
Nodes: [1, 2, 3, 4, 5, 6]
Edges: [(1, 3), (1, 4), (1, 5), (1, 6), (2, 4), (2, 5), (3,
1)]
```

🧠 解释：

- `nx.DiGraph()`：表示这是一个**有向图**
- `add_edges_from()`：批量添加边
- 每条边是元组 `(source, target)`，例如 `(1, 3)` 表示从节点 1 指向节点 3 的连接

分析图的属性

💡 示例代码：

```
your_node = 1  # 选择要分析的节点

print("In-degree:", G.in_degree(your_node))
print("Out-degree:", G.out_degree(your_node))

closeness = nx.closeness_centrality(G)
betweenness = nx.betweenness_centrality(G)

print("Closeness Centrality:", closeness.get(your_node))
print("Betweenness Centrality:", betweenness.get(your_node))
```

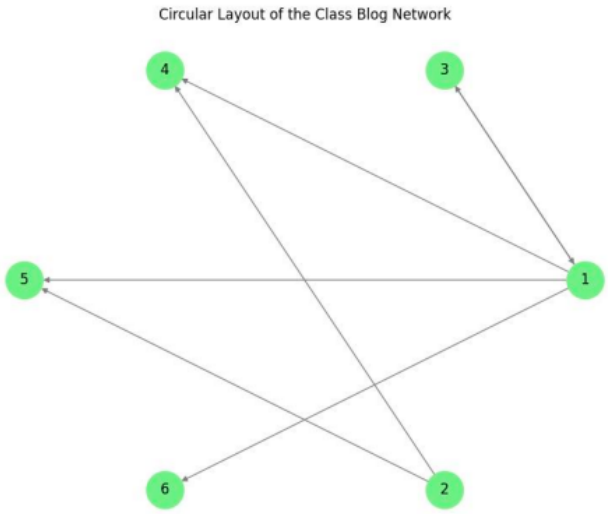
✅ 含义：

指标	说明
In-degree	被多少个其他节点指向 → “被关注数”
Out-degree	指向了多少个其他节点 → “主动关注数”
Closeness Centrality	到所有其他节点的平均最短路径越短，值越高 → “信息传播效率高”
Betweenness Centrality	经过该节点的最短路径数量越多，值越高 → “中介作用强”

📌 举个例子：

- 在 Twitter 上，一个账号如果有很多人关注它（高入度），并且它是信息传递的关键节点（高介数），那它就是“影响力大”的用户。

可视化图



💡 示例代码：

```
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 6))
nx.draw_circular(G,
                 with_labels=True,
                 node_color='lightgreen',
                 node_size=1000,
                 font_size=12,
                 edge_color='gray')
plt.title("Circular Layout of the Class Blog Network")
plt.show()
```

🎨 效果：

- 使用圆形布局将节点均匀分布在圆周上
- 节点颜色为浅绿色，大小为 1000
- 边为灰色线
- 显示标签（节点编号）

🔹 第五页：多种可视化布局函数

🖼️ NetworkX 提供多种内置布局方式：

函数	描述
`nx.draw()`	默认布局（内部使用弹簧模型）
`nx.draw_circular()`	将节点排成一圈
`nx.draw_shell()`	分层排列（同心圆壳）
`nx.draw_spectral()`	使用谱方法（特征向量）排列
`nx.draw_spring()`	弹簧力优化布局（Fruchterman-Reingold）
`nx.draw_kamada_kawai()`	距离优化布局
`nx.draw_random()`	随机放置节点
`nx.draw_planar()`	仅适用于平面图

✅ 自定义选项：

- `node_color`，`node_size`，`edge_color`，`font_size`，`with_labels` 等都可以调整

读取 CSV 文件（网络数据）

📄 典型网络数据格式：

Source node	Target node
1	2
1	3
…	…

每行代表一条边（从 source 到 target）

💡 代码示例：

```
import pandas as pd

# 读取无标题的 CSV 文件，并手动命名列
```

```
df = pd.read_csv('blog_edges.csv', header=None, names=['fro
m', 'to'])
```

✅ 注意事项：

- `header=None`：因为文件没有表头
- `_names=['from', 'to']`：指定两列名称为 "from" 和 "to"

用 NetworkX 创建有向图（从 DataFrame）

💡 代码示例：

```
import networkx as nx

G = nx.DiGraph()  # 创建有向图
G.add_edges_from(zip(df['from'], df['to']))  # 用 zip 将两列合
并为边列表
```

🧠 解释：

- `zip(df['from'], df['to'])` 生成形如 `(1,2), (1,3), ...` 的边对
- `add_edges_from()` 批量添加这些边到图中

✅ 实现了从真实数据文件 → 构建图结构的完整流程！
