

第三次作业：谁是大V？

- 给你一个微博消息转发的数据集，请你分析哪个用户更重要
 - 要求1：根据消息转发建一个有向图
 - 要求2：计算用户的中心度
 - 要求3：计算用户的PageRank值
 - 要求4：选择自己感兴趣的话题，完成上述分析

第三次作业：谁是大V？

	origin_user_id	origin_user_name	retweet_user_id	retweet_user_name	origin_content	retweet_content	origin_user_followers_num
0	1615743184	全球奇闻趣事	1658721803	stuttgartbo	这姐妹太逗了说相声的吧哈哈粽 粽粽粽粽粽粽	NaN	4018252
1	1615743184	全球奇闻趣事	1105624024	礼乐诗书	这姐妹太逗了说相声的吧哈哈粽 粽粽粽粽粽粽	转发微博	4018252
2	1615743184	全球奇闻趣事	1774924647	inrA谜谈陈	这姐妹太逗了说相声的吧哈哈粽 粽粽粽粽粽粽	看一次笑一次哈哈哈尼玛	4018252
3	1615743184	全球奇闻趣事	1856568402	豹__是我的最_	这姐妹太逗了说相声的吧哈哈粽 粽粽粽粽粽粽	bed__bed__	4018252
4	1615743184	全球奇闻趣事	1801335805	不刷少女_代_死星人	这姐妹太逗了说相声的吧哈哈粽 粽粽粽粽粽粽	_	4018252
5	1615743184	全球奇闻趣事	1694206155	jhtzhou	这姐妹太逗了说相声的吧哈哈粽 粽粽粽粽粽粽	我擦这表情这演技分明是从电影 里截的吧哈哈	4018252
6	1615743184	全球奇闻趣事	1732359680	_雀雀	这姐妹太逗了说相声的吧哈哈粽 粽粽粽粽粽粽	琴晚睇_呢_笑到_街偷_	4018252
7	1615743184	全球奇闻趣事	1732949813	三3叁宅一生	这姐妹太逗了说相声的吧哈哈粽 粽粽粽粽粽粽	啊伊呀伊呦啊伊呀伊呦阿弟可带 一个带一个带一个他可带一个带 一个刀带一个带一个带一个他可 带一个带...	4018252
8	1615743184	全球奇闻趣事	1748832480	Vincent筱宇	这姐妹太逗了说相声的吧哈哈粽 粽粽粽粽粽粽	转发微博	4018252
9	1615743184	全球奇闻趣事	1854027245	希大爷Aiko	这姐妹太逗了说相声的吧哈哈粽 粽粽粽粽粽粽	哈哈哈哈哈哈哈	4018252
10	2282688074	狮子座频道	1838165717	微子世界	狮子座的人很敏感看似什么都不 计较不细心其实是在包容你所以 我会假装什么都不知道狮子座 的人最不...	转发微博	514893
11	2282688074	狮子座频道	1697074350	骑驴找马的村民	狮子座的人很敏感看似什么都不 计较不细心其实是在包容你所以 我会假装什么都不知道狮子座	不发言对号入座吧	514893



中國人民大學
RENMIN UNIVERSITY OF CHINA

计算传播理论与实务

2019-2020秋季学期

第四讲 网络分析

社区检测

授课教师：范举副教授、塔娜讲师
时间：2019年12月16日

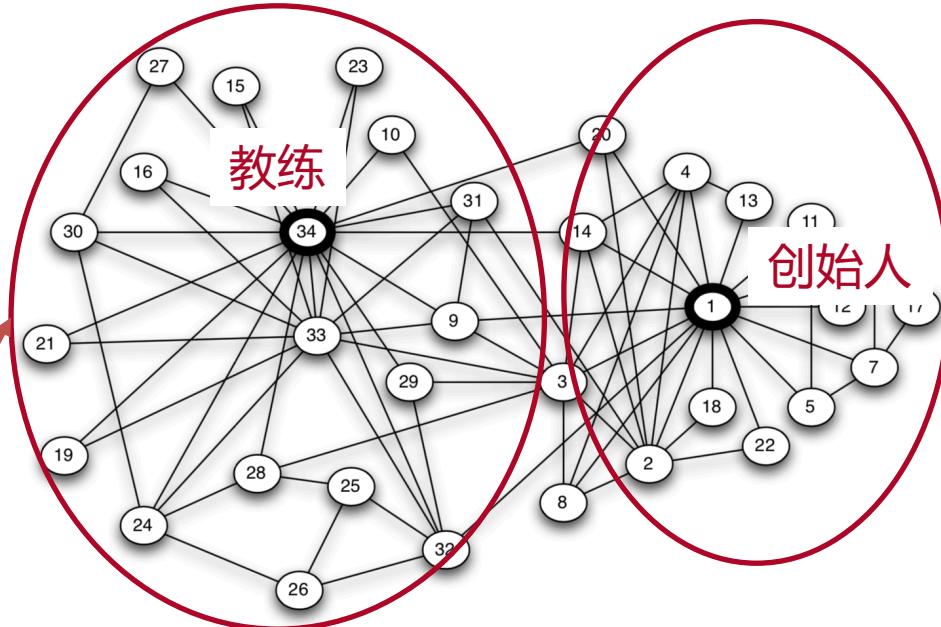
社区检测 (Community Detection)

- 为什么使用图模型对数据建模?
 - 图提供了一种观察数据结构特征的视角

最终这个俱乐部分裂成两个对立的空手道俱乐部

检测出图中的社区结构

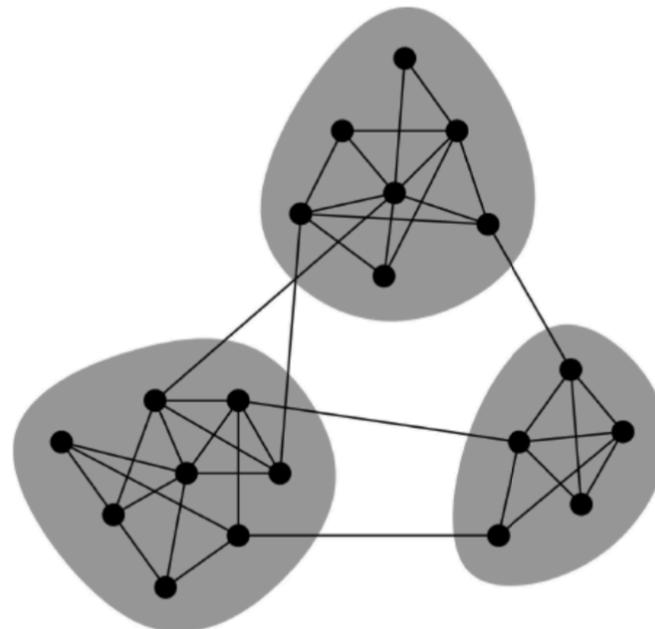
一个空手道俱乐部中34个成员之间朋友关系形成的图
你能发现什么特点?



图片来源: [Easley and Kleinberg, Cambridge University Press, 2010]

真实网络中为什么能形成社区？

- 在分析数据之前，首先要对数据的性质有足够的认识
- 为什么社交网络长成下面的样子？



Mark Granovetter的研究

- 上世纪60年代末，Mark Granovetter在做他博士论文研究
 - 研究题目：人们是如何找到新的工作的
 - 发现1：人们通过人际关系获取了新工作的信息
 - 发现2：获取新工作的人际关系通常是点头之交（casual acquaintances）而并非亲密好友（close friends）
- 这个发现很让人惊讶
 - 一般认为，亲密好友对你的帮助应该大于点头之交的熟人。

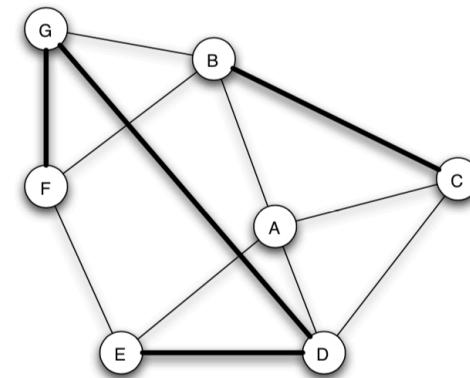
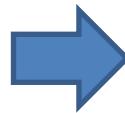
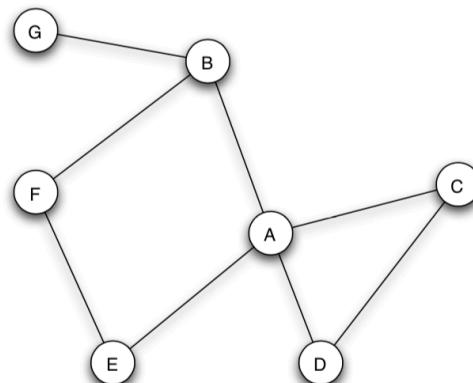


三元闭包 (Triadic Closure)

- 三元闭包 (Triadic Closure)
 - 如果两个人在网络中有共同的好友，他们成为好友的几率也会提升
- 几点原因：如果B和C都有共同好友A，那么
 - B更有可能遇到C——因为他们都与A有交集
 - B和C更可能互相信任——因为他们有共同的好友
 - A更有可能介绍B和C认识

三元闭包 (Triadic Closure)

- 量化指标：聚集因子 clustering coefficient
 - 节点A的聚集因子是A任意两个邻居是好友的概率

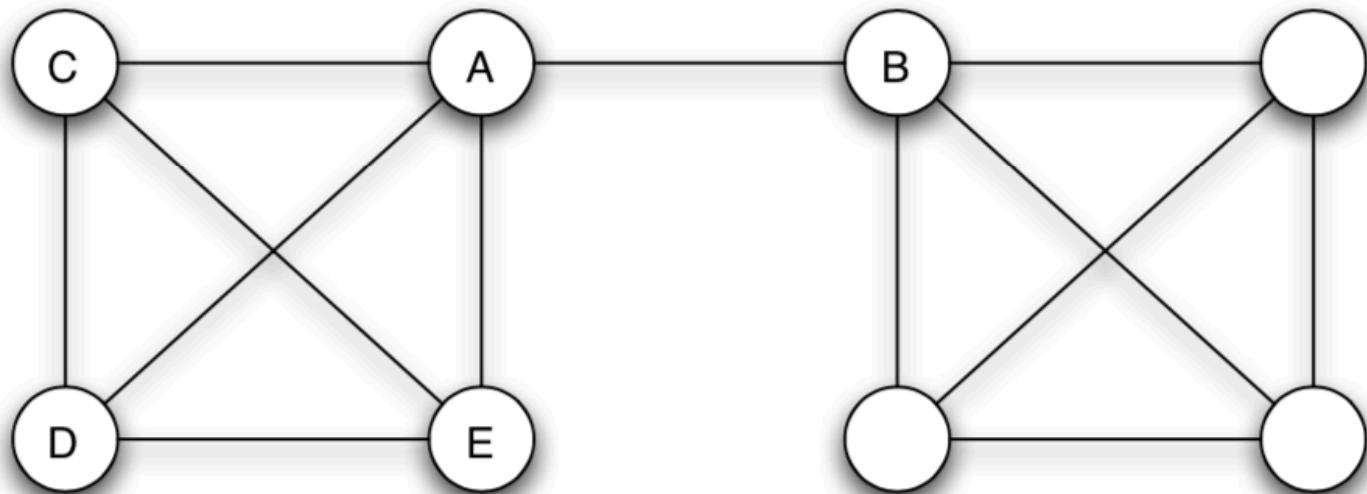


clustering coefficient of A = 1/6

clustering coefficient of A = 1/2

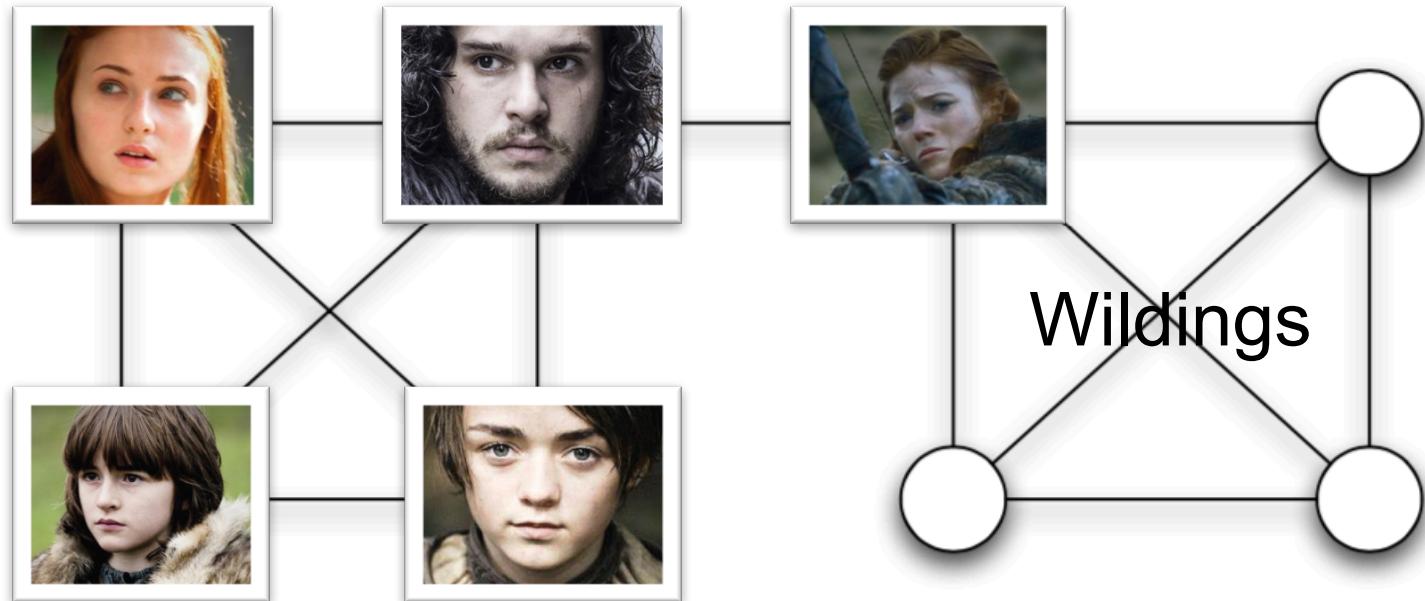
强关系 vs. 弱关系

- 怎么解释下图中A和B之间的关系？



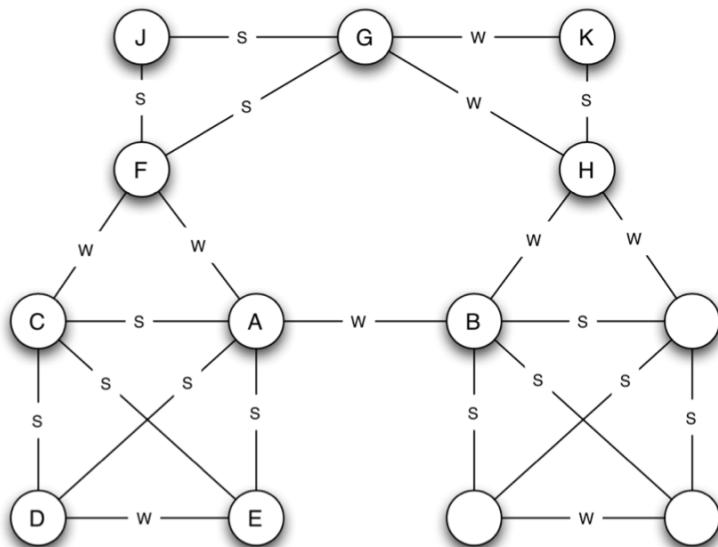
强关系 vs. 弱关系

- 怎么解释下图中A和B之间的关系？



强关系 vs. 弱关系

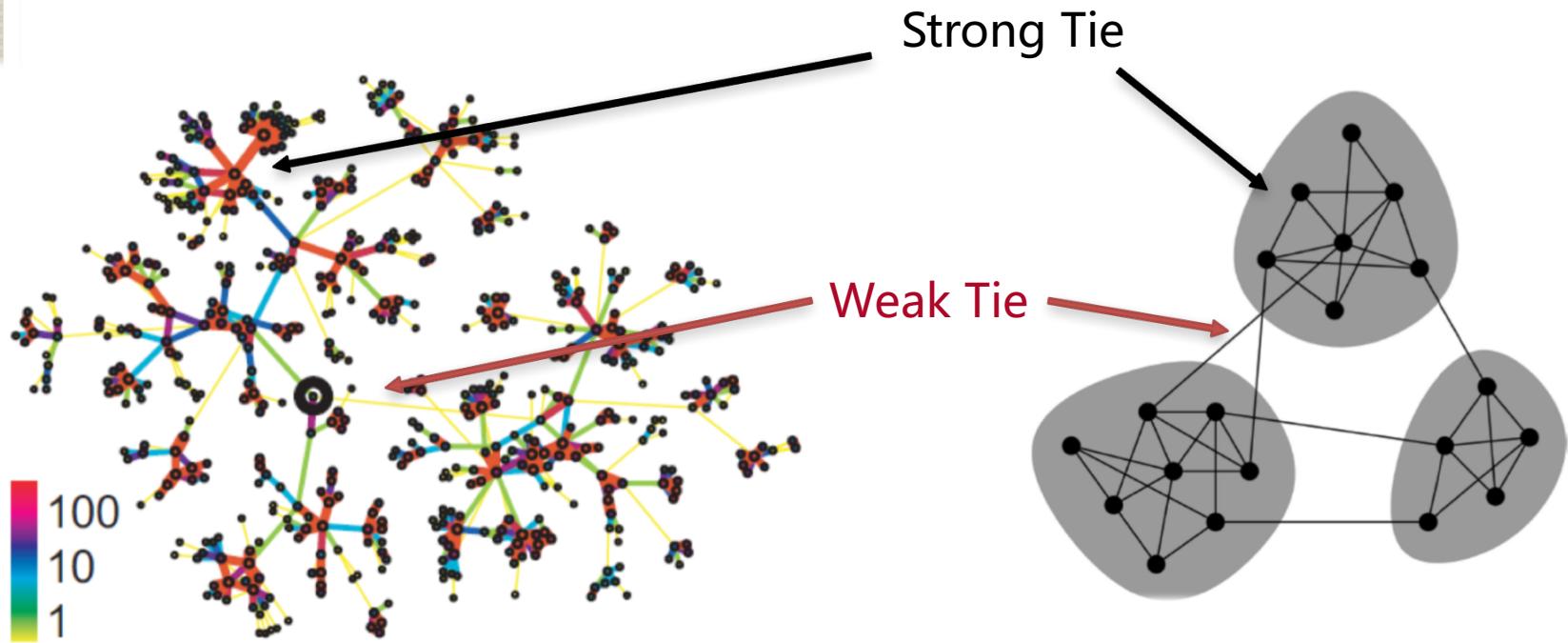
- Granovetter从结构和社交功能两个角度将边分为
 - 强关系 Strong Tie
 - 弱关系 Weak Tie



- 结构角度
 - 强关系意味着社交紧密
 - 弱关系链接网络不同部分
- 社交功能角度
 - 弱关系让你从不同角度获取信息，从而找到新工作
 - 强关系在新信息获取方面的作
用十分有限

真实数据中的强弱关系

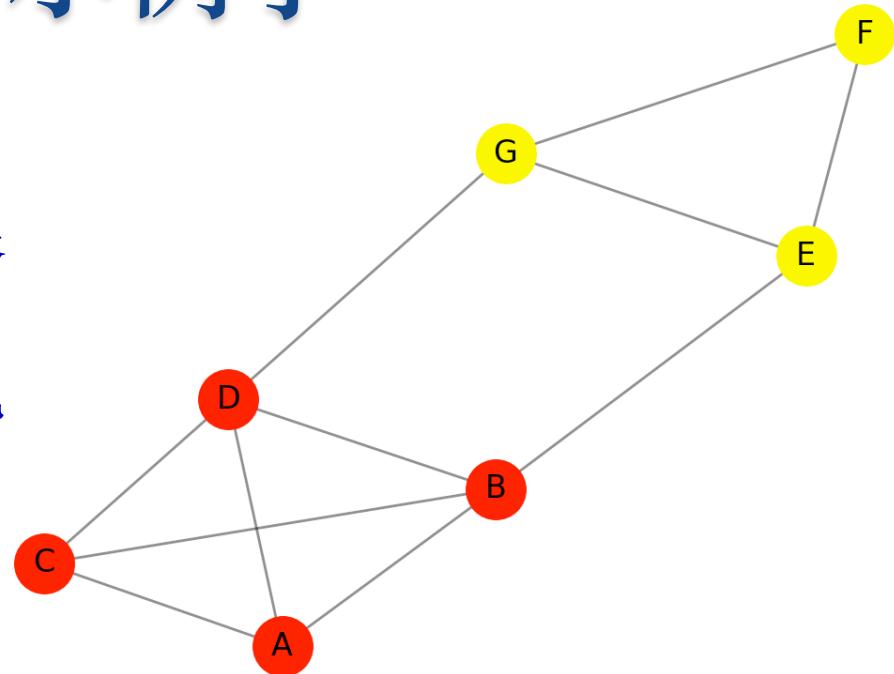
- [Onnela et al. 2007] 测量了电话通信网络
 - 边的强弱表示打电话的次数



图片来源: [Onnela et al. 2007]

我们考虑一个小例子

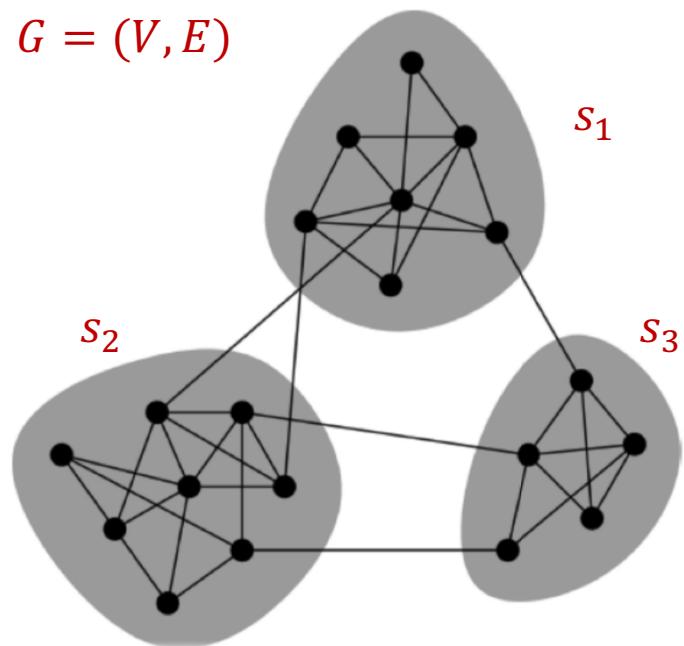
- 演示示例
 - 构造一个简单的社交网络
- 思考
 - 如何自动地将右图中红色的点与黄色的点分开?



```
def sample_community_graph () :  
    G = nx.Graph()  
    G.add_edges_from([('A', 'B'), ('A', 'C') ,  
                    ('A', 'D'), ('B', 'C'), ('B', 'D'),  
                    ('C', 'D'), ('B', 'E'), ('D', 'G'),  
                    ('E', 'G'), ('E', 'F'), ('F', 'G')])  
    return G
```

问题定义

- 输入
 - 一个无向图 $G = (V, E)$
- 输出
 - 一组点的划分 S
 - $\forall s \in S, s \subseteq V$
 - $\forall i, j, s_i \cap s_j = \emptyset$ and $\bigcup_i s_i = V$
- 设计优化目标!
 - 给定图 G , 评价 S 的质量



优化目标：Modularity Q

- 定义Modularity函数

$$Q(G, S) = \frac{1}{2m} \sum_{s \in S} \sum_{i \in s} \sum_{j \in s} \left(A_{ij} - \frac{k_i k_j}{2m} \right)$$

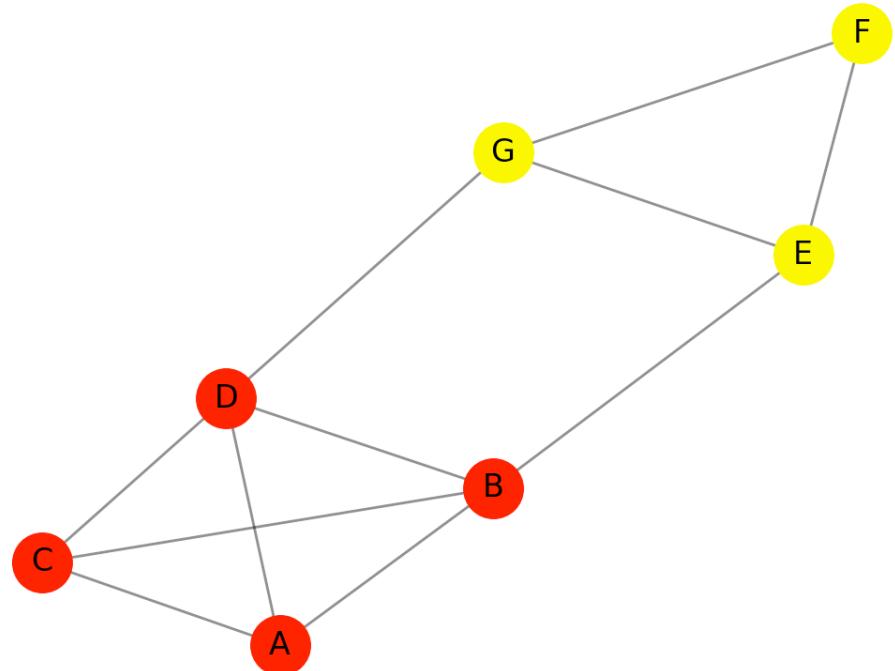
- Modularity函数的范围是[-1,1]
- Modularity函数大于0.3-0.7，成为“显著社区结构” (significant community structure)
- 社区发现的思路之一：优化modularity函数

课堂练习

- 计算右图中以下划分的Modularity分值
 - $\{\{A\}, \{B\}, \{C\}, \{D\}, \{E\}, \{F\}, \{G\}\}$
 - $\{\{A, C\}, \{B, D\}, \{E, G\}, \{F\}\}$
 - $\{\{A, B, C, D\}, \{E, F, G\}\}$
 - $\{\{A, B, C, D, E, F, G\}\}$



$$Q = \frac{1}{2m} \sum_{ij} [A_{ij} - \frac{k_i k_j}{2m}] \delta(c_i, c_j)$$



$$Q(G, S) = \frac{1}{2m} \sum_{s \in S} \sum_{i \in s} \sum_{j \in s} \left(A_{ij} - \frac{k_i k_j}{2m} \right)$$

Louvain算法

- 算法的基本思想

通过贪心法最大化Modularity

- 算法的优点

- 快：时间复杂性 $O(n \log n)$
- 好：在很多真实网络上能够得到较高的Modularity
- 支持边上有权重的图
- 提供层次化的划分

Louvain算法

- 算法的基本思想
 - 1. 刚开始的时候，所有的顶点都是一个小小的类簇 - **init**
 - 2. Phase 1：以局部方式，优化模块度函数，将每个顶点归到“最好”的类簇中，直到所有的顶点所属的类簇不再变化为止 - **one_level**
 - 3. Phase 2：把一个类簇中的所有顶点聚集抽象为一个顶点，重建一个网络，其中的每个顶点对应一个社区 - **induced_graph**
 - 4. 看抽象以后的网络图，是否还有优化的可能性，如果有，则迭代执行上述(2)、(3)步骤。

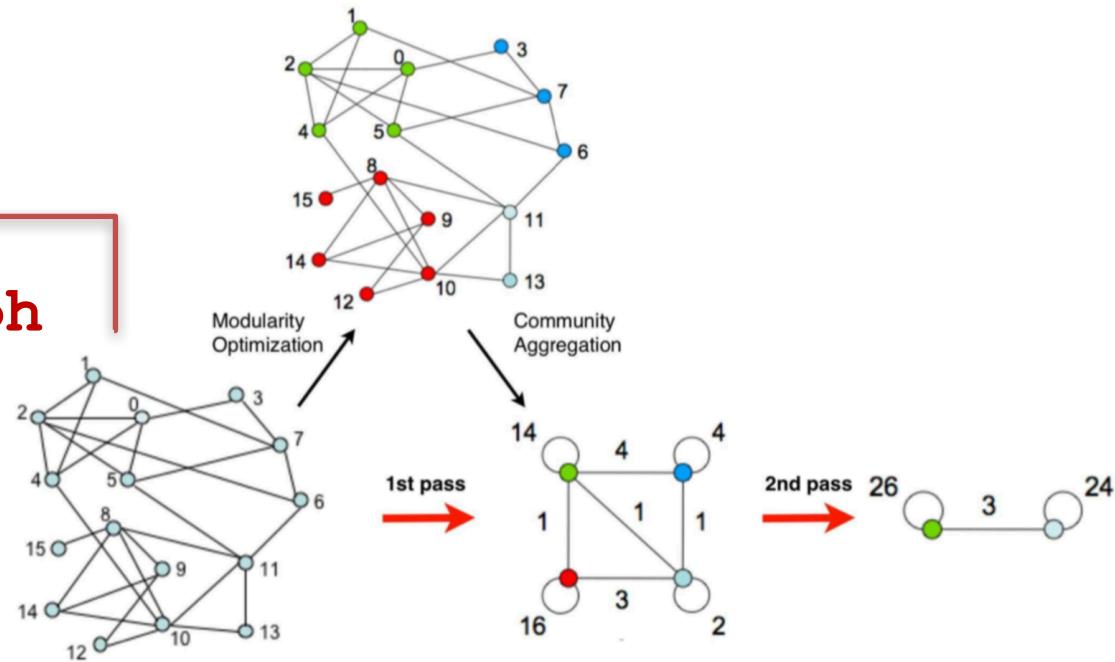
Louvain算法

- 算法的基本思想

1. `init`

2. `one_level`

3. `induced_graph`



请写出检测出的层
次化社区结构

Louvain算法

- Phase I：以局部方式，优化模块度函数，将每个顶点归到“最好”的类簇中，直到所有的顶点所属的类簇不再变化为止 – **one_level**
 - 计算将节点*i*合并到邻居*j*所在社区的modularity增益 ΔQ
 - 将节点*i*合并到能够产生最大增益 ΔQ 的节点*j*的社区中
 - 循环执行上述步骤，直到合并操作不再产生modularity的增益

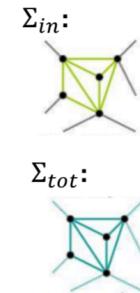
Louvain算法

- Phase I：以局部方式，优化模块度函数，将每个顶点归到“最好”的类簇中，直到所有的顶点所属的类簇不再变化为止 – **one_level**
- 如何计算将 i 合并到社区 C 中的modularity增益 ΔQ ？

$$\Delta Q(i \rightarrow C) = \left[\frac{\Sigma_{in} + k_{i,in}}{2m} - \left(\frac{\Sigma_{tot} + k_i}{2m} \right)^2 \right] - \left[\frac{\Sigma_{in}}{2m} - \left(\frac{\Sigma_{tot}}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right]$$

■ where:

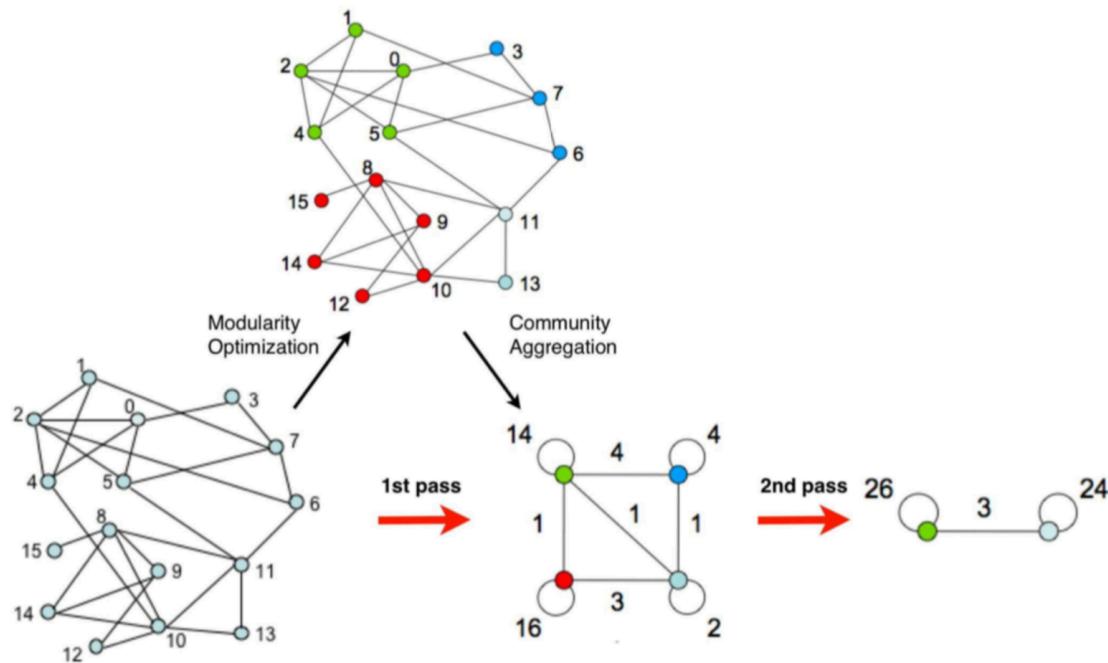
- Σ_{in} ... sum of link weights between nodes in C
- Σ_{tot} ... sum of all link weights of nodes in C
- $k_{i,in}$... sum of link weights between node i and C
- k_i ... sum of all link weights (i.e., degree) of node i



- Also need to derive $\Delta Q(D \rightarrow i)$ of taking node i out of community D .
- And then: $\Delta Q = \Delta Q(i \rightarrow C) + \Delta Q(D \rightarrow i)$

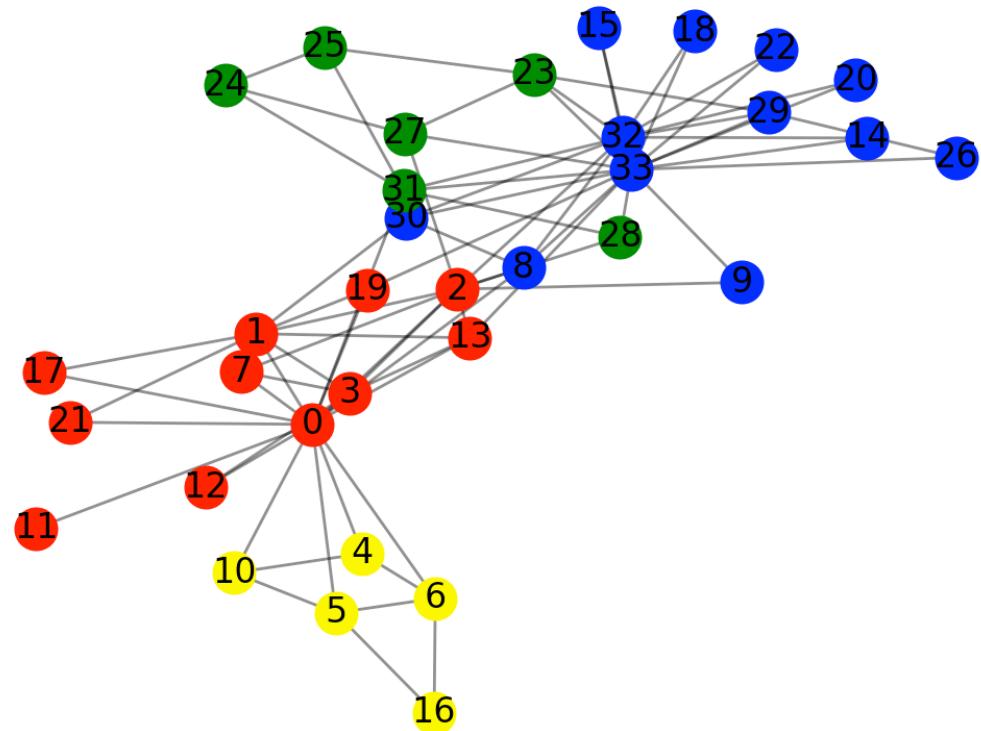
Louvain算法

- Phase 2: 把一个类簇中的所有顶点聚集抽象为一个顶点，重建一个网络，其中的每个顶点对应一个社区 - **induced graph**



空手道俱乐部社区检测示意

```
G = nx.karate_club_graph()
```



```
pip install python-louvain
```



中國人民大學
RENMIN UNIVERSITY OF CHINA

计算传播理论与实务

2019-2020秋季学期

第四讲 网络分析

影响力分析

案例分析：feeds流广告

○ 什么是feeds流广告：

- feeds广告就是与内容混排在一起的广告：最不像广告的广告，长得最像内容的广告。
- 不留意在它周围出现的“推广”、“广告”字样，可能都不会发现这是一条广告。
- feeds广告操作性简单，打扰性低，已经成为移动互联网时代主流的广告形式。

○ 微信朋友圈广告典型feeds流广告

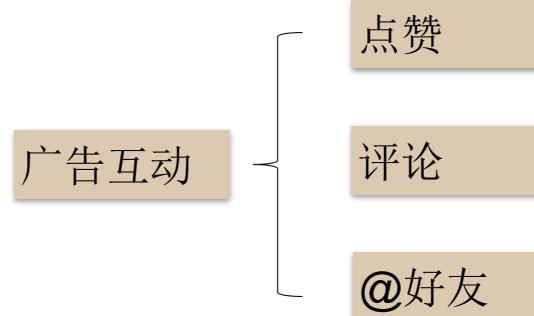
- 以原生feeds流的形式插入朋友圈好友动态列表中
- 建立在用户行为记录和大数据分析基础上，进行个性化推荐



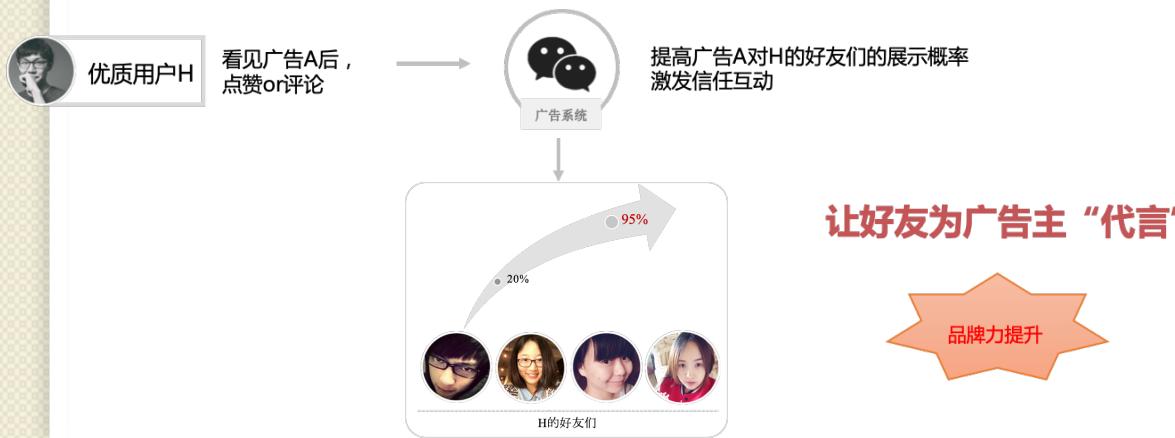
整体-大盘-曝光量	:	1,686,709,174
整体-大盘-可视曝光	:	1,015,382,271
整体-大盘-收入 (元)	:	66,735,746

案例分析：feeds流广告

- 广告互动：微信朋友圈广告具有互动属性

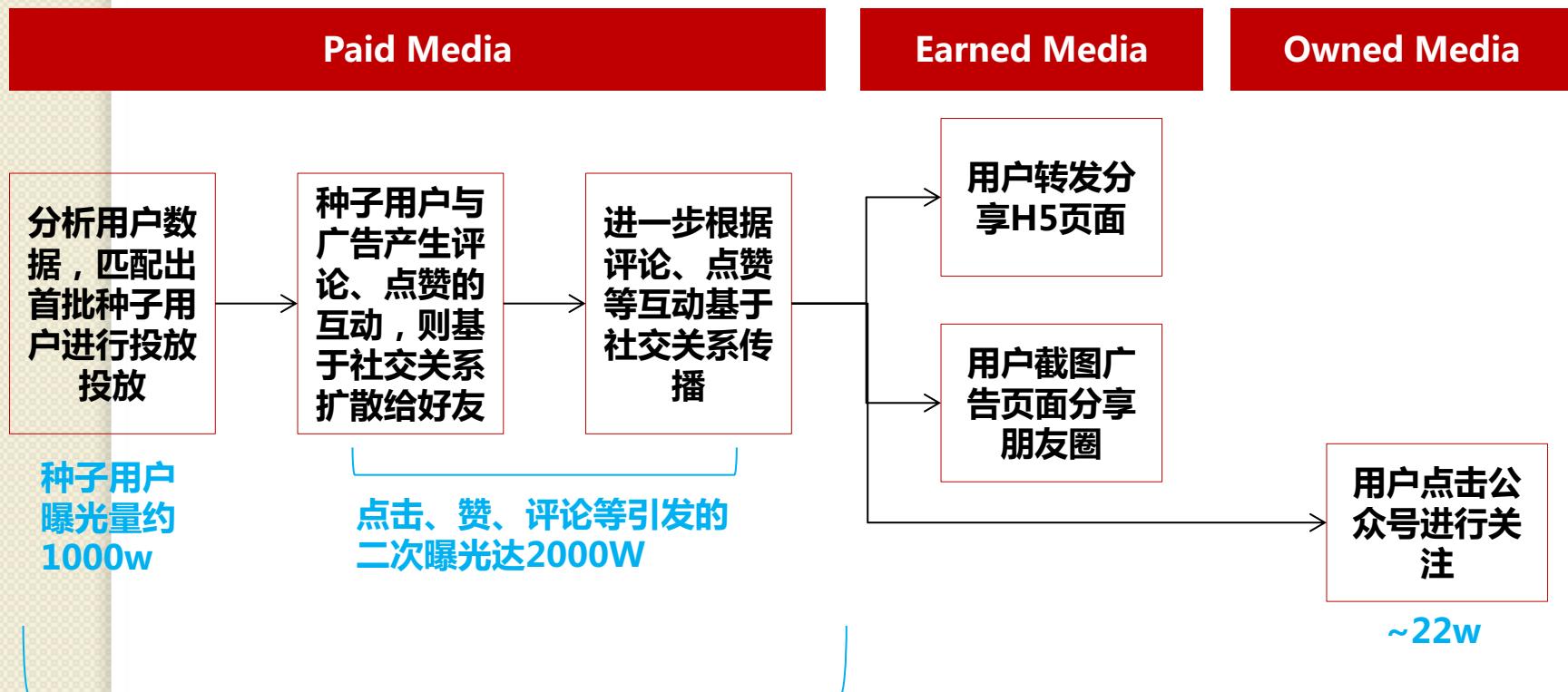


- 互动的意义：品牌随着关系链传播，好友的互动加强传播效果



案例分析：feeds流广告

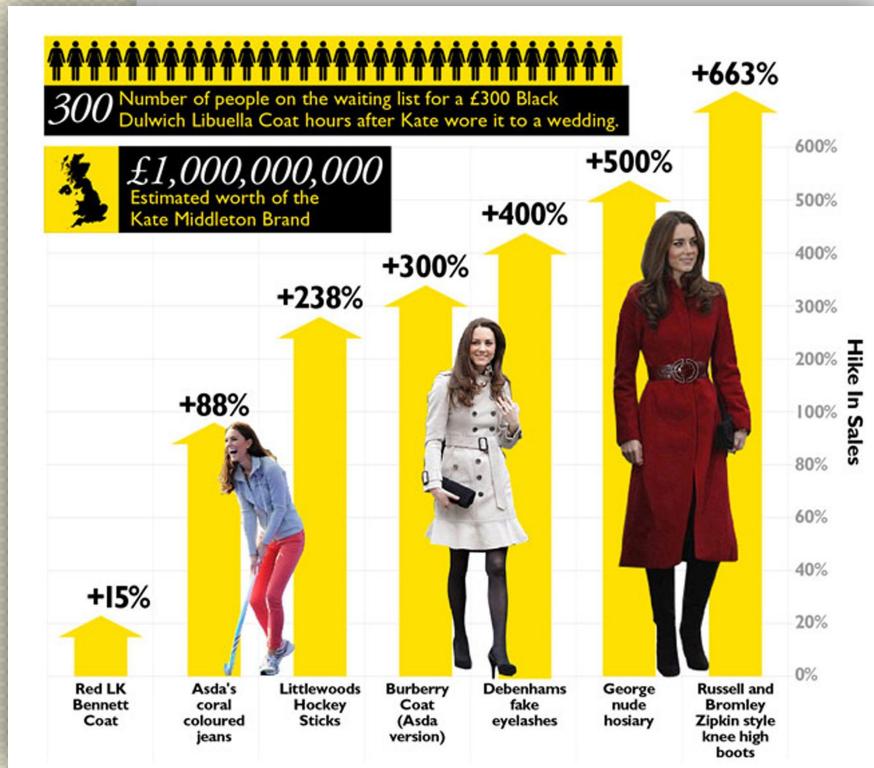
- 微信朋友圈首次投放三家广告，通过用户评论、点赞、分享等进行基于社交关系的传播，总曝光量达到亿级



BMW : 截止投放3小时内，总曝光量接近3000W

社交影响力

- 凯特王妃效应 (Kate Middleton Effect)
 - 穿衣品味影响力拉动百亿时装消费
- 最佳的营销推广形式来自熟人之间的口耳相传

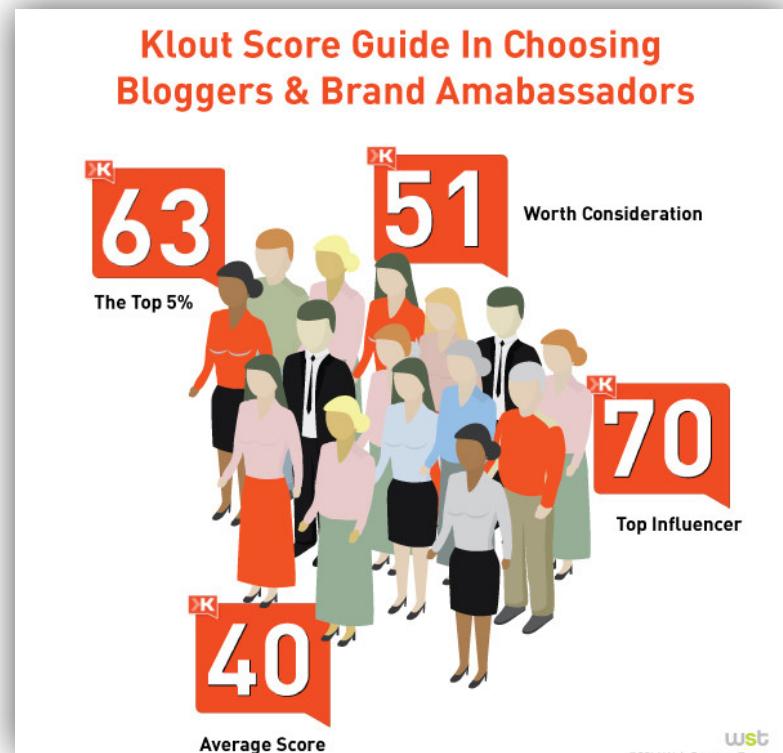
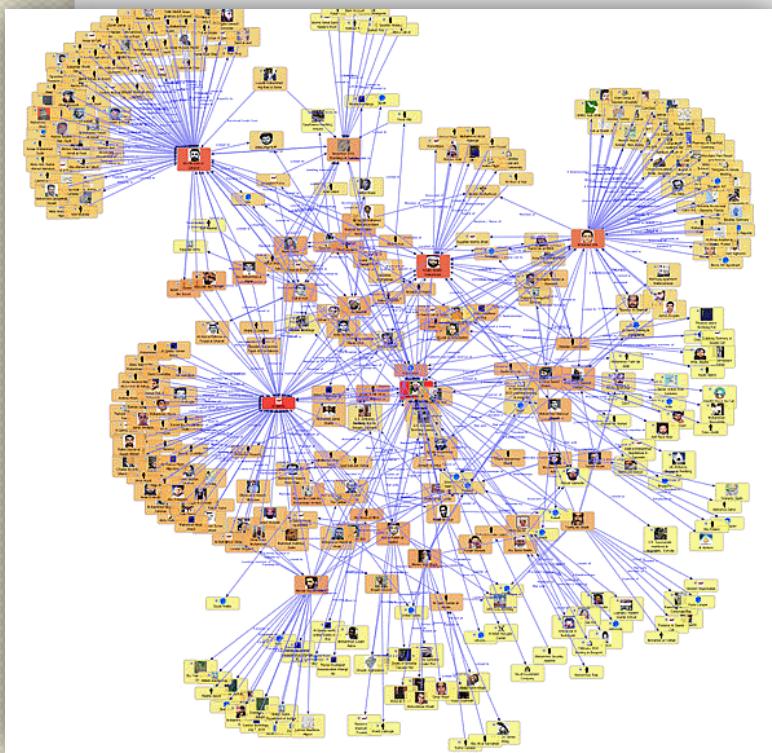


Recommendations from people known (90%)



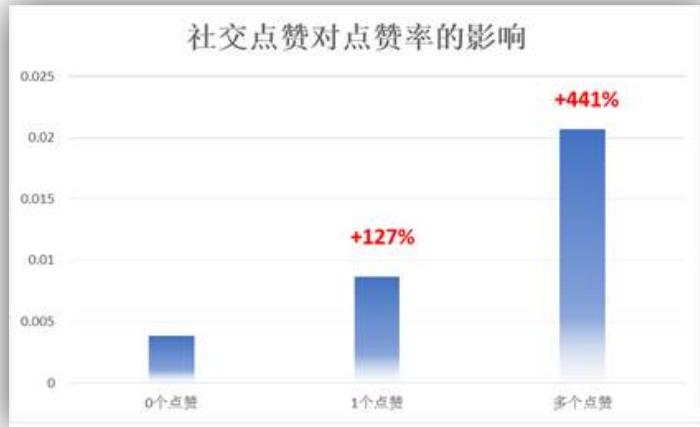
社交影响力

- 在线社交网络迅猛发展
- 在线社交影响力：度量用户在社交网络上对其听众能够影响（大V vs. 小透明）



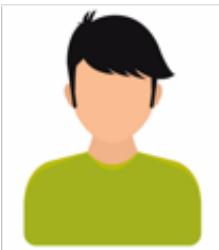
社交影响力

- 广告的好友互动显著提升用户的互动意愿



案例分析：广告互动行为预测

- 问题定义



f(,



) = 互动概率

- 为什么要预测广告互动概率?
 - 社交关键节点先投放，逐层影响其它节点参与互动
 - 微信的隐私保护决定了广告互动率的重要性

节点价值

$$quality_score = \alpha \times heart_rate_{ee} + \beta \times comment_rate_{ee} + \gamma \times influence_{ee}$$

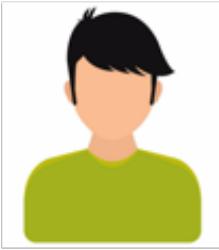
点赞概率

评论概率

个人影响力

案例分析：广告互动行为预测

- 问题定义



f(,



) = 互动概率

- 核心难点：互动概率预估

2018年7月24日品牌广告

实际

点赞数: 9.89W

评论数: 3.55W

预估

点赞数: 2.26W

评论数: 8597

Bias

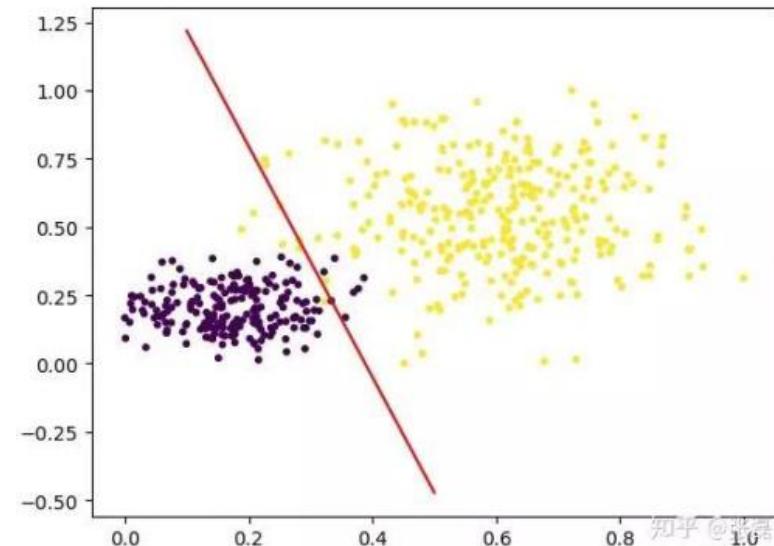
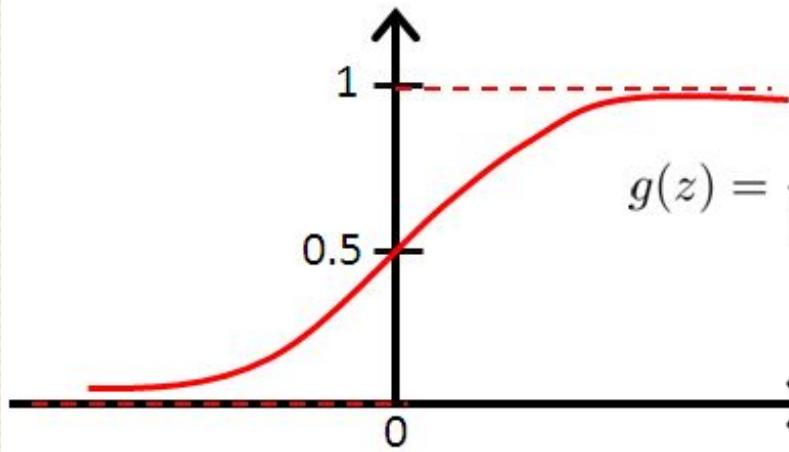
-77.15%

-75.78%

解决思路 - I

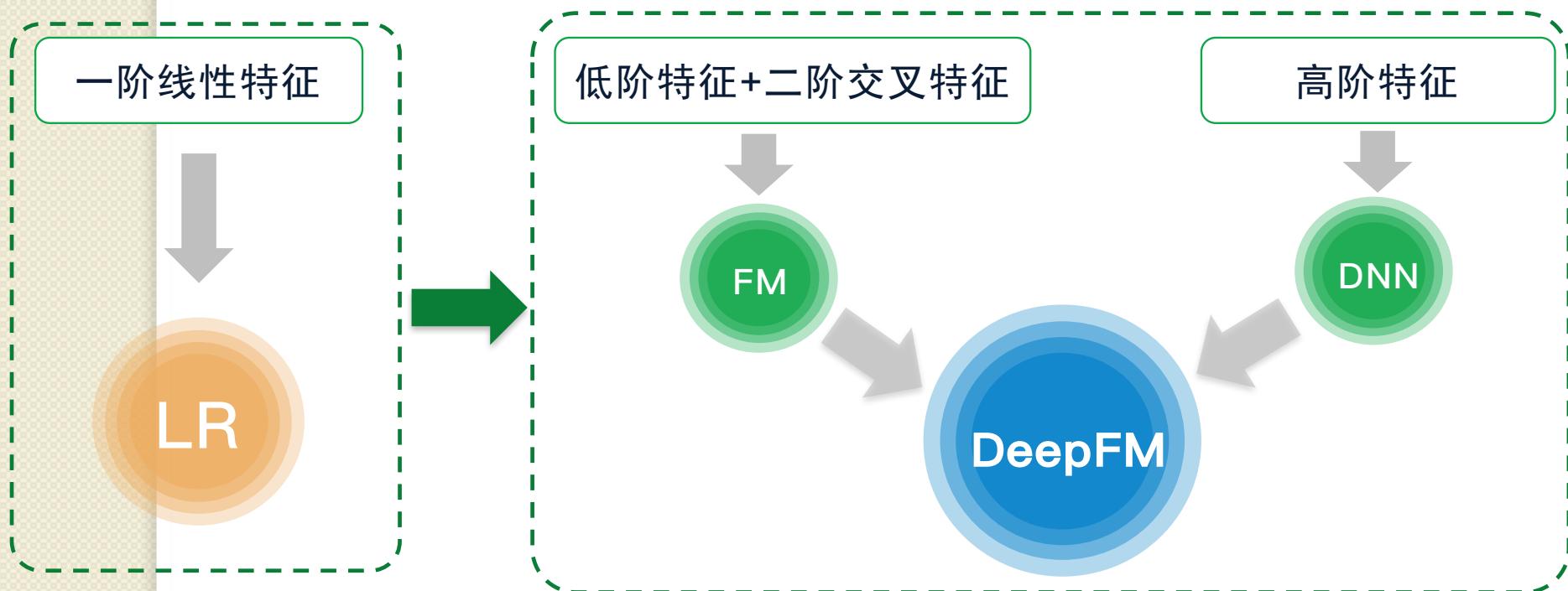
- 采用用户特征、广告特征和上下文特征，用简单的逻辑回归模型进行预测
 - 逻辑回归 (LR) 是预估模型中最基本的模型，也是工业界最喜爱使用的方案。

$$f(x) = \frac{1}{1 + e^{-\theta^T X}}, \text{ 其中 } \theta = (\theta_0, \theta_1, \theta_2 \dots \theta_n)$$



解决思路 - 2

- 采用用户特征、广告特征和上下文特征，用更复杂的深度学习模型进行预测



- 深度学习模型的代表：DeepFM模型

解决思路 - 3

- 已有方案没有考虑微信广告的社交传播价值，没有挖掘社交影响力相关特征
 - 对互动行为高频/低频用户的影响力结构进行统计



图 20 社交互动率最高的结构

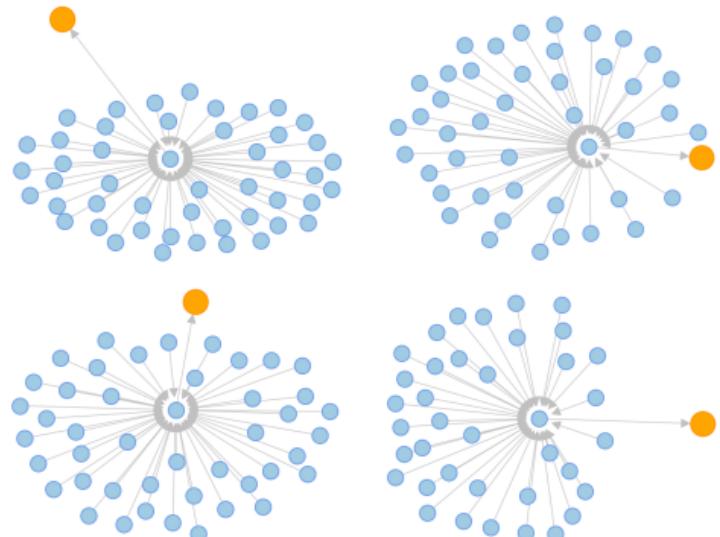
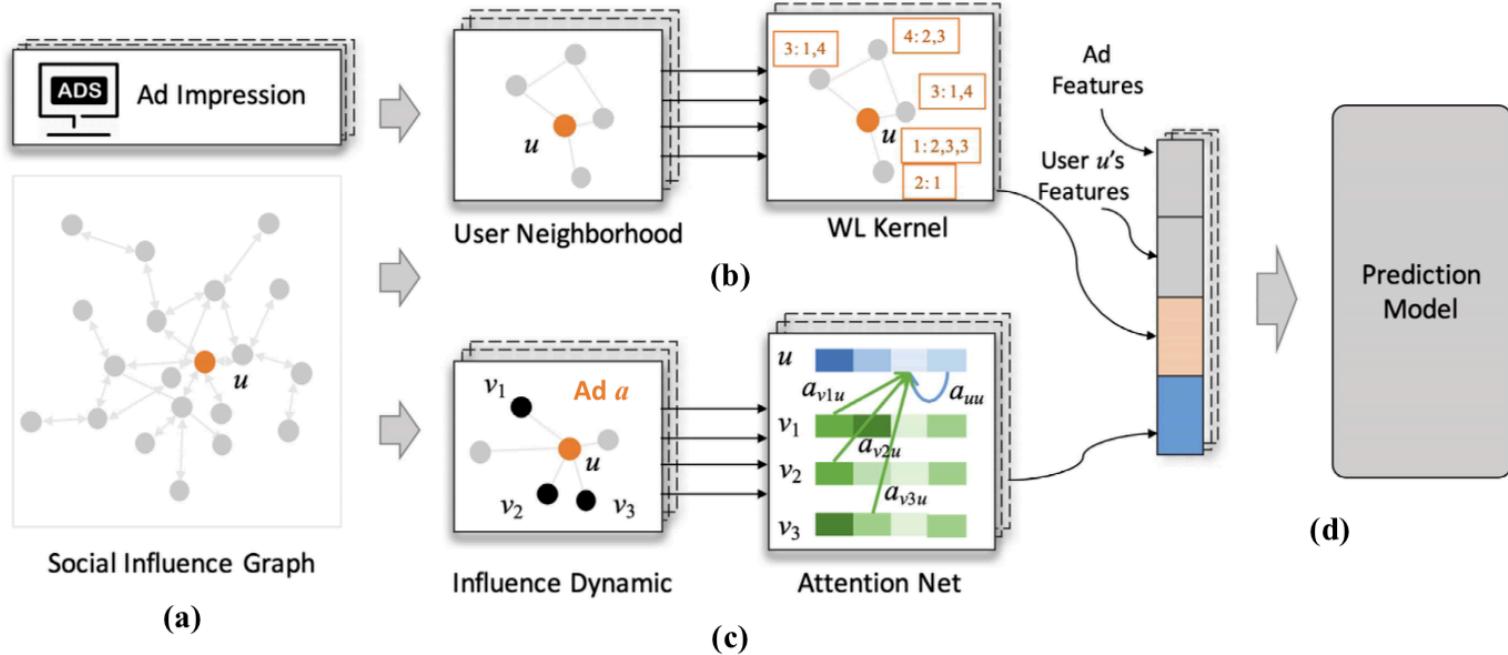


图 21 社交互动率最低的结构

解决思路 - 3

- 两个用户在相近时间先后执行相同动作，那么认为这是先执行动作的用户对后执行动作用户的一次成功影响
- 传统社交网络的构建是基于好友关系，数据相对丰富
- 广告上的社交网络构建是基于好友在相同广告上的历史交互行为，数据比较稀疏
- 根据历史数据，同一广告上平均只有3.14个人发生过社交交互

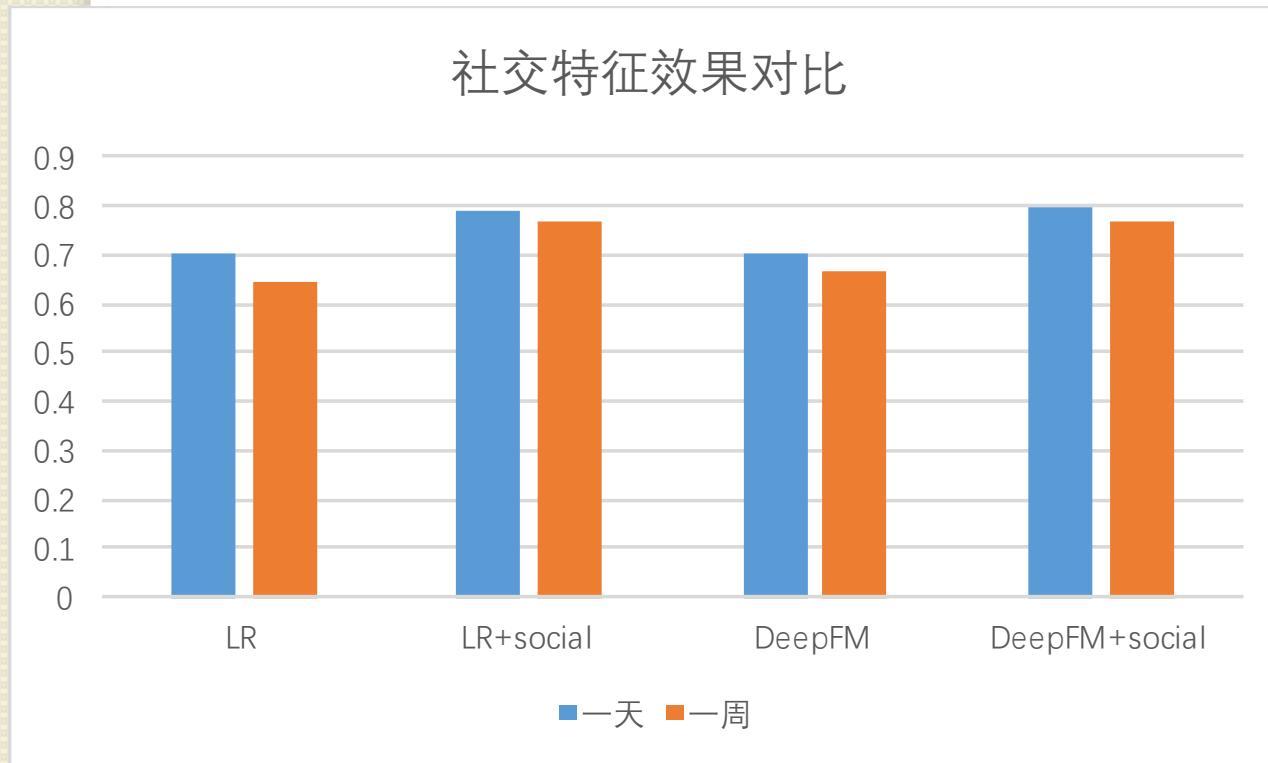
解决思路 - 3



- Phase I: 社交影响图的构建
- Phase II: 社交特征的学习
- Phase III: 融合社交特征进行互动率的预测

解决思路 - 3

- 离线效果评估 (AUC)
 - 真实数据：微信朋友圈广告投放
 - 日期：20180715—20180721 数据量：1958,539



社交特征的加入使预测效果提升超过10%

解决思路 - 3

- 在线效果评估
 - 线上实际效果提升

实验数据明细					
时间	实验名称	总曝光量	点赞率	评论率	互动率
2018-08-30	BaseLine2	4,630,337 -70.08%	0.0362% -4.37% [-9.42%,0.96%]	0.0103% 6.56% [-3.90%,18.15%]	0.0467% -2.69%
2018-08-30	算法9998【异步预估】	15,492,358 0.10%	0.0385% 1.59% [-2.01%,5.32%]	0.0100% 3.45% [-3.65%,11.09%]	0.0489% 1.78%
2018-08-30	算法10000【同步预估】	12,549,955 -18.91%	0.0386% 1.81% [-1.99%,5.76%]	0.0101% 4.45% [-3.09%,12.59%]	0.0492% 2.40%
2018-08-30	算法9999【同步预估】	16,276,884 5.17%	0.0364% -3.91% [-7.31%,-0.38%]	0.0096% -0.70% [-7.51%,6.61%]	0.0464% -3.37%
2018-08-30	Base_Line	15,477,087	0.0379%	0.0096%	0.0480%
2018-08-29	算法9998【异步预估】	11,551,515 0.03%	0.0494% -0.66% [-4.23%, 3.04%]	0.0139% 1.23% [-5.56%, 8.51%]	0.0642% -0.12%
2018-08-29	算法10000【同步预估】	9,463,743 -18.05%	0.0538% 8.12% [4.11%, 12.28%]	0.0150% 9.15% [1.60%, 17.26%]	0.0696% 8.37%
2018-08-29	算法9999【同步预估】	11,871,274 2.80%	0.0523% 5.29% [1.58%, 9.13%]	0.0140% 2.00% [-4.78%, 9.27%]	0.0673% 4.82%
2018-08-29	Base_Line	11,547,501	0.0497%	0.0137%	0.0642%

模型小流量上线后：

- 8.29日点赞率提升8.12%，评论率提升9.15%，互动率提升8.37%
- 虽然8.30日提升有所下降，但跟原先比都有提升。