网络数据挖掘

第三部分：图数据挖掘

沈华伟

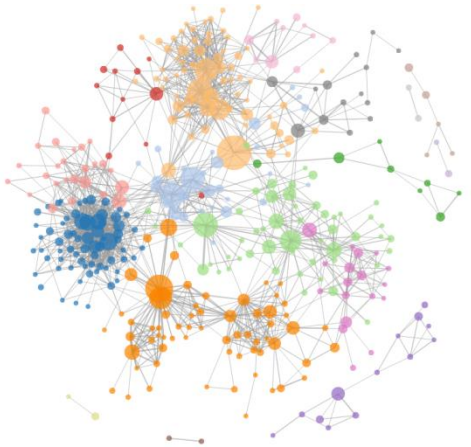
shenhuawei@ict.ac.cn

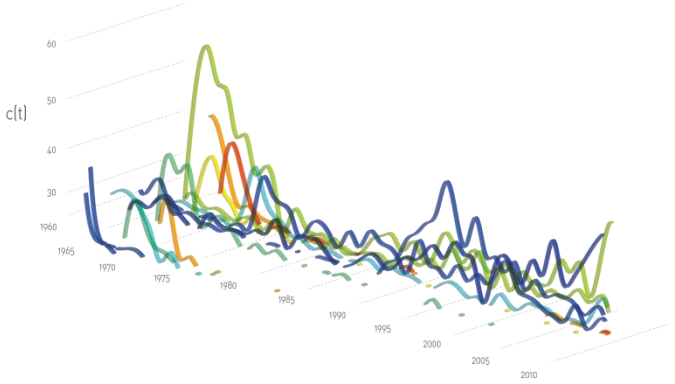
中国科学院计算技术研究所

2018.12.4

图数据挖掘

- 第一讲：图排序(11月27日)
  - 复杂网络
  - 图排序
- 第二讲：图挖掘(12月4日)
  - 图聚类
  - 社区发现
- 第三讲：图预测(12月11日)
  - 链路预测
  - 传播预测





第二讲：图挖掘

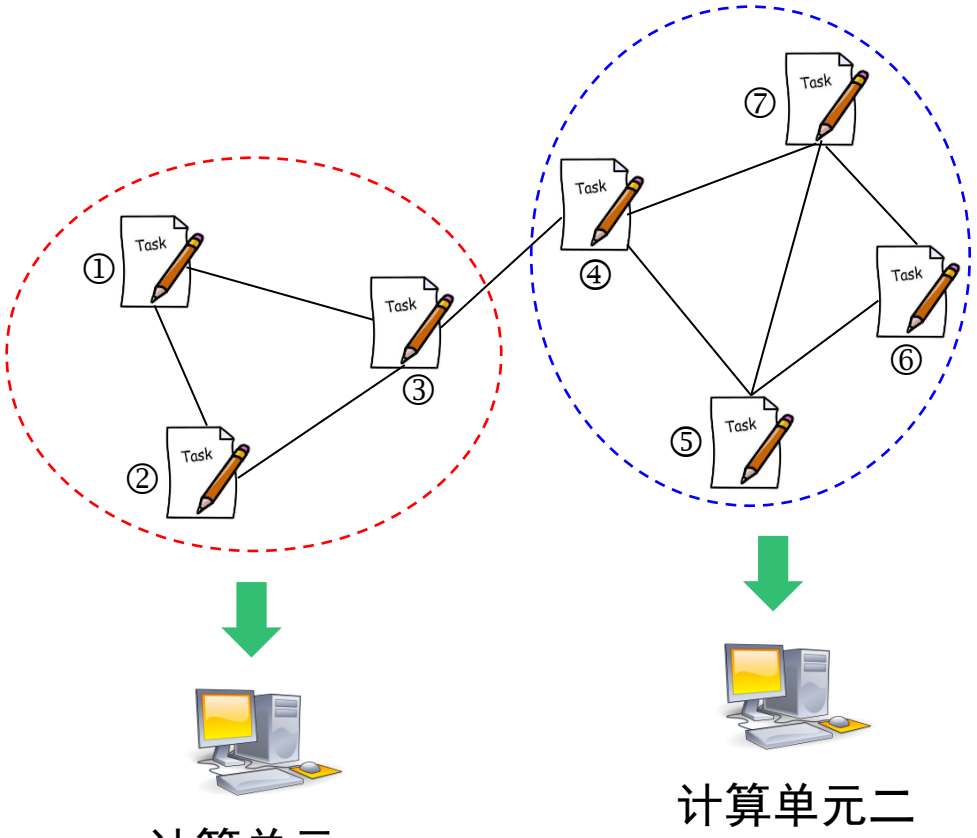
内容提纲

- 图划分
  - Min Cut, Ratio Cut, Normalized Cut
- 社区发现
  - 模块度
  - InfoMap
- 图建模
  - 非负矩阵分解
  - 随机块模型

图划分问题案例

任务调度

给定一些计算单元和计算任务，任务间需要通信（如图），计算单元内部的通信代价相对于计算单元间的通信代价可忽略不计，如何将计算任务部署到计算单元，使通信代价最小



计算任务及相互通信关系

图划分问题形式化

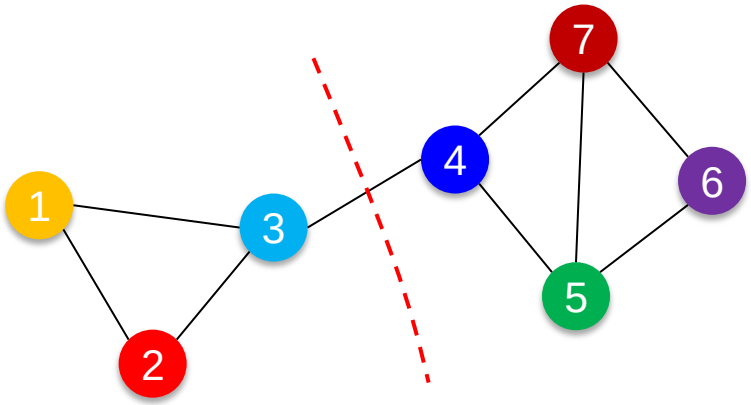
给定图$G=(V,E)$， V为节点集合，E为边集，其邻接矩阵记为A，

寻找V的一个划分$C=\{C_1,C_2,\cdots,C_K\}$，满足$V=\cup_i C_i$， $C_i\cap C_j=\emptyset\ (i\neq j)$，

使得划分的各个分量$C_i(1\leq i\leq K)$之间的连边权重之和最小

$$\mathrm{cut}(C)=\frac{1}{2}\sum_{i=1}^{K}W(C_i,\overline{C_i})$$

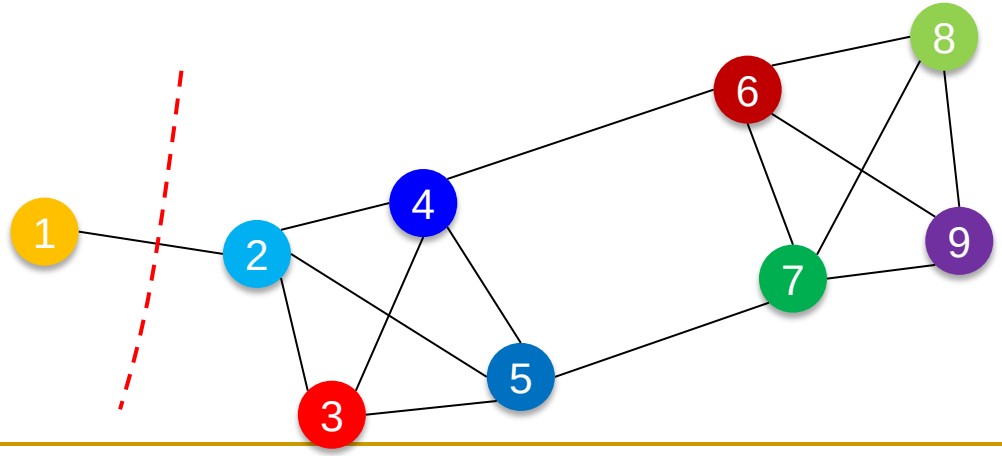
$$W(C_i,\overline{C_i})=\sum_{u\in C_i,v\in\overline{C_i}}A_{uv}$$



Min Cut $\arg\min_C \mathrm{cut}(C)$

Min Cut的问题

- 平凡解
  - 所有节点划分到同一个分量中
  - 解决办法：指定K
- 不均衡解
  - 划分的各个分量，大小差异大
  - 解决办法：限制分量大小



Min Cut的扩展

- **Ratio Cut**

$$\mathrm{RatioCut}(C) = \frac{1}{2}\sum_{i=1}^{K}\frac{W(C_i, \overline{C_i})}{|C_i|}$$

C_i中节点个数

- **Normalized Cut（NCut）**

$$\mathrm{NCut}(C) = \frac{1}{2}\sum_{i=1}^{K}\frac{W(C_i, \overline{C_i})}{\mathrm{vol}(C_i)}$$

$$\mathrm{vol}(C_i) = \sum_{u\in C_i}\sum_{v} A_{uv}$$

和C_i中节点相连的边的权重之和

图划分求解算法

- 局部方法
  - KL算法（Kernighan-Lin）
- 全局方法
  - 谱划分

KL算法

- 目标
  - 寻找图的最优两路划分

- 算法
  - 第一步：构造初始划分$C = \{C_1, C_2\}$
  - 第二步：从C_1中选择一个节点a，从C_2中选择一个节点b，交换a和b可以使$\mathrm{cut}(C)$减小，则交换
  - 重复第二步直至$\mathrm{cut}(C)$不再减小

B. W. Kernighan, S. Lin. An efficient heuristic procedure for partitioning graphs. Bell System Technical Journal 49: 291–307, 1970.

谱划分：拉普拉斯矩阵

- **拉普拉斯矩阵**

$$L = D - A$$ A是邻接矩阵/权值向量

D是一个对角阵，对角线元素$d_{ii} = \sum_j A_{ij}$ L的各行和为0

- **性质**
  - 对于任意向量x，有

$$x'Lx = \frac{1}{2}\sum_{i,j=1}^{n} A_{ij}(x_i - x_j)^2$$ 刻画x的平滑程度，越小越平滑

  - L有n个非负特征值

$$0 = \lambda_1 \leq \lambda_2 \cdots \leq \lambda_n$$

谱划分

$$\delta_{ij}=\begin{cases}1, & i=j\\ 0, & i\neq j\end{cases}$$

- **Min Cut**

2路划分：A和B

$$s_i=\begin{cases}+1, & i\in A\\ -1, & i\in B\end{cases}$$

$$\sum_{ij}A_{ij}=\sum_i k_i=\sum_{ij}s_i s_j k_i\delta_{ij}$$

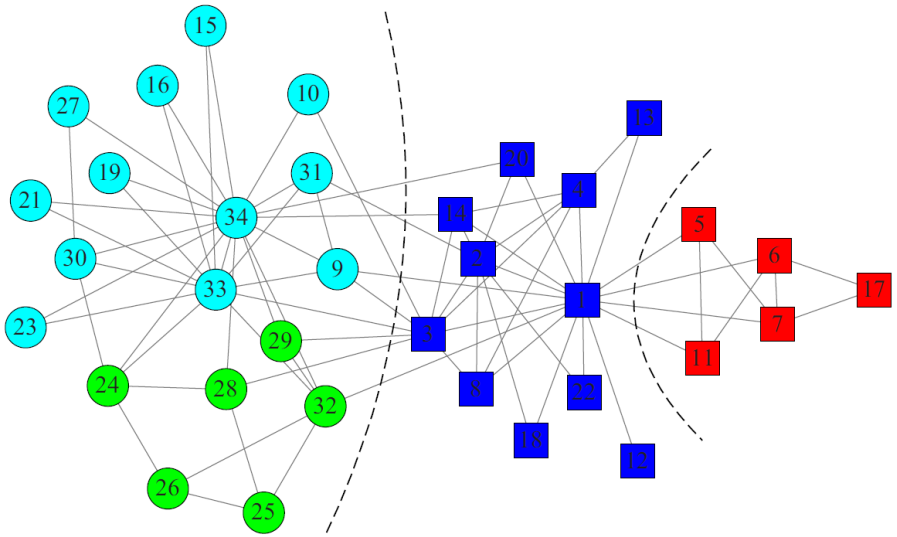
$$\text{cut}=\frac{1}{4}\sum_{ij}(1-s_i s_j)A_{ij}=\frac{1}{4}\sum_{ij}s_i s_j(k_i\delta_{ij}-A_{ij})=\frac{1}{4}s'Ls$$

令 $s=\sum_{i=1}^{n}a_i v_i$ （v是L的特征向量） $\Rightarrow$ $a_i=v_i's$ $\qquad \sum_{i=1}^{n}{a_i}^2=n$

有 $\text{cut}=\frac{1}{4}\left(\sum_{i=1}^{n}a_i v_i'\right)L\left(\sum_{j=1}^{n}a_j v_j\right)=\frac{1}{4}\sum_{i,j=1}^{n}a_i a_j\lambda_j\delta_{ij}=\frac{1}{4}\sum_i a_i^2\lambda_i$ （λ是L的特征值）

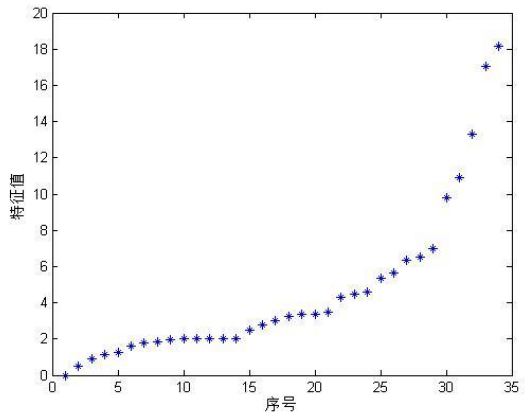
启示：让s尽可能和最小特征值所对应的特征向量(Fiedler vector)方向一致

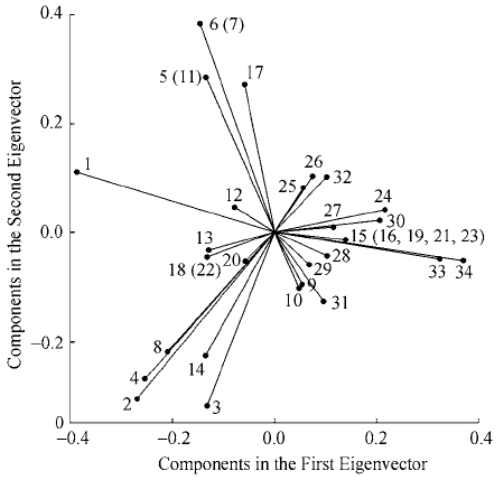
谱划分：案例



Fiedler vector $v_2 =$

$$\begin{pmatrix} 0.1121 \\ 0.0413 \\ -0.0232 \\ 0.0555 \\ 0.2846 \\ 0.3237 \\ 0.3237 \\ 0.0526 \\ -0.0516 \\ -0.0928 \\ 0.2846 \\ 0.2110 \\ 0.1095 \\ 0.0147 \\ -0.1628 \\ -0.1628 \\ 0.4228 \\ 0.1002 \\ -0.1628 \\ 0.0136 \\ -0.1628 \\ 0.1002 \\ -0.1628 \\ -0.1557 \\ -0.1530 \\ -0.1610 \\ -0.1871 \\ -0.1277 \\ -0.0952 \\ -0.1677 \\ -0.0735 \\ -0.0988 \\ -0.1303 \\ -0.1189 \end{pmatrix}$$



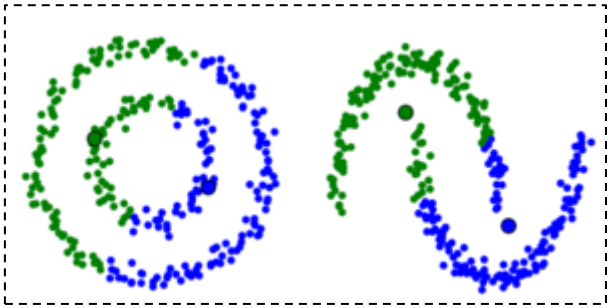


谱聚类

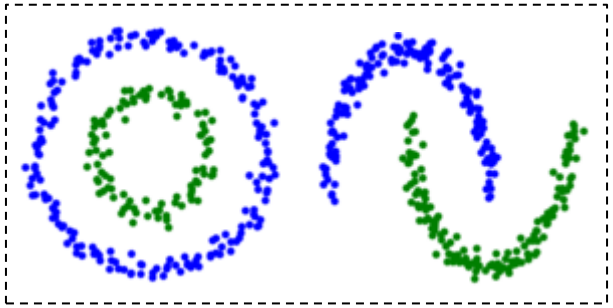
- 使用拉普拉斯矩阵最小的k个非负特征值的特征向量，将图中节点表示为k 维空间的一个点

- 采用 k -means进行聚类

找最近若干邻居构建网络，然后通过Laplace做Network Embedding（投影）到欧氏空间，然后用K-Means聚类



K-means



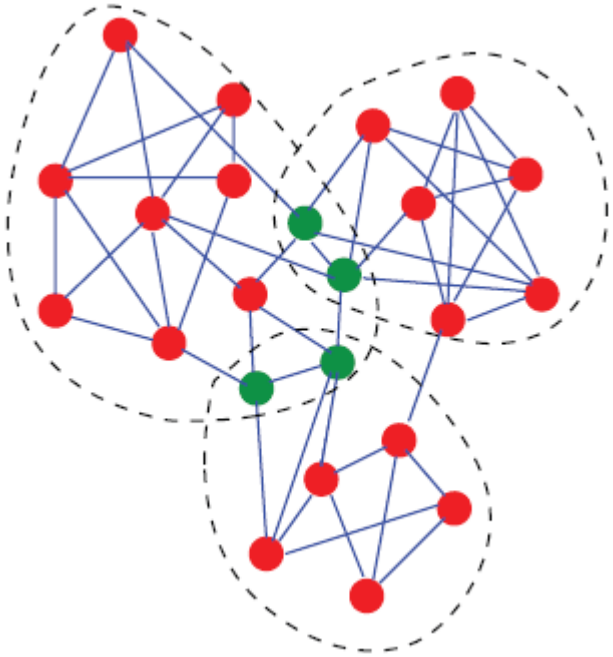
谱聚类
(构建图，谱分析，k-means）

内容提纲

- 图划分
  - Min-cut, Ratio-cut, Normalized-cut
- 社区发现
  - 模块度
  - InfoMap
- 图建模
  - 非负矩阵分解
  - 随机块模型

社区发现

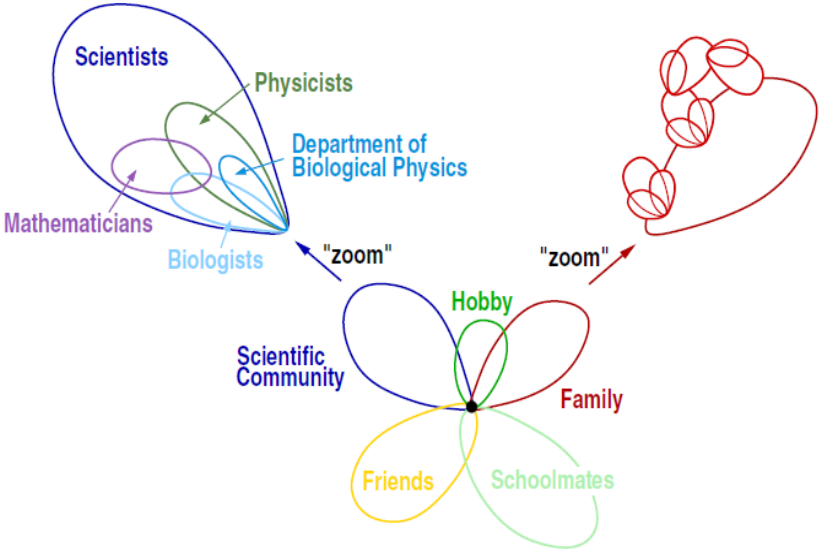
- 识别出网络中“内部连接紧密、与外部连接稀疏”的节点组



社区发现和图划分的区别

- 图划分
  - 按照任务需求对网络进行划分
  - 划分的分量数通常已知
  - 各个分量彼此不重叠

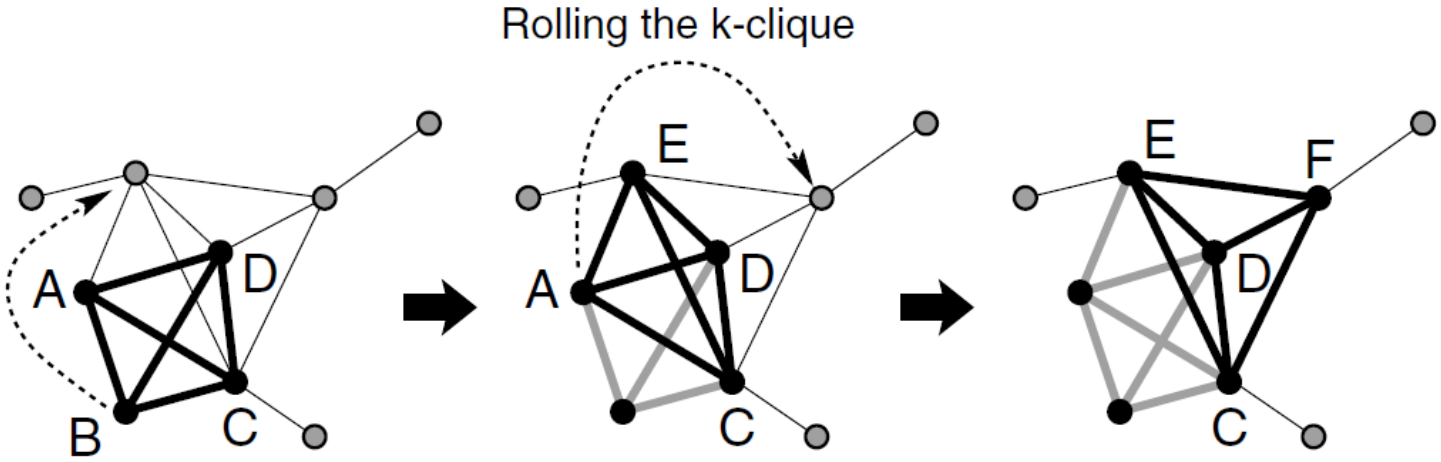
- 社区发现
  - 寻找网络固有的结构规则
  - 社区个数通常未知
  - 社区可以重叠、嵌套



社区发现之初

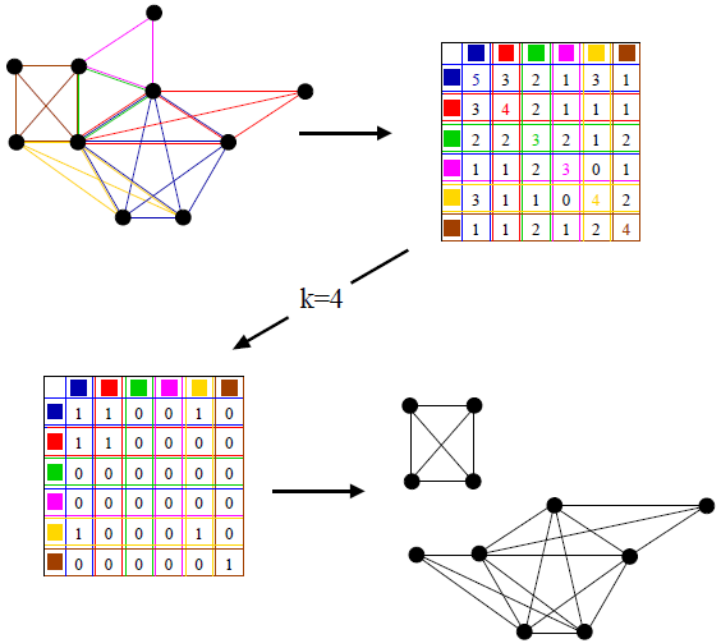
- 每个连通分量视为一个社区
  - 巨连通分量（Giant Connected Component）效应
  - 连通分量相当于边渗流（edge percolation）

- 团渗流（Clique percolation）

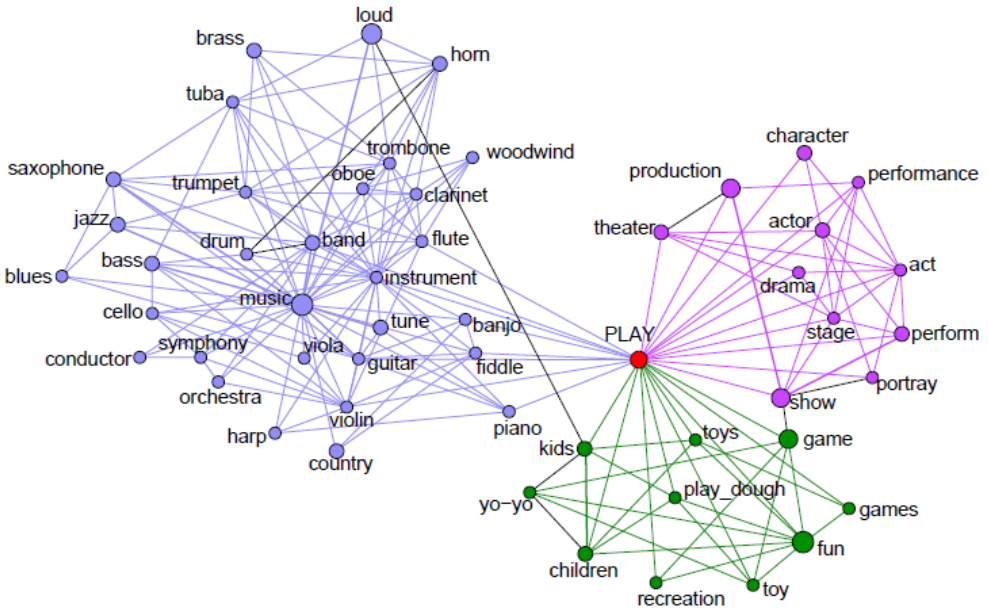


社区发现之初

- 基于团渗流的社区发现方法

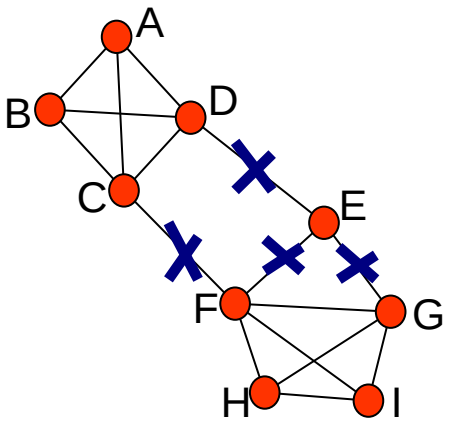


Clique percolation method



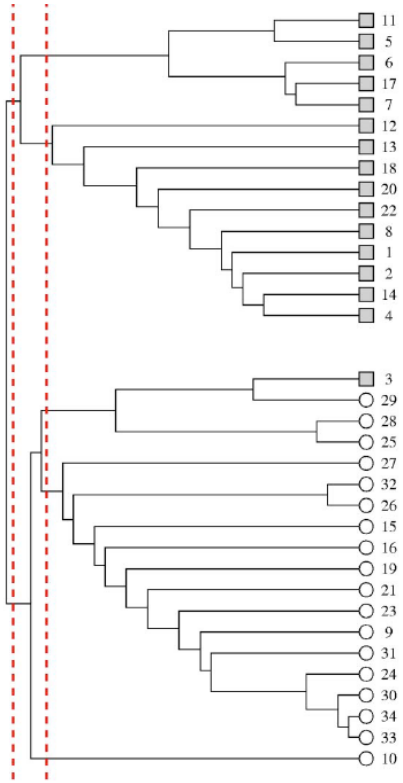
词网上的示例

社区发现问题的提出



- 2002年提出
  - Girvan & Newman, PNAS

- GN算法
  - 基于边介数的算法
    - 步骤1：计算每条边的介数
    - 步骤2：删除介数最大的边
  - 得到一颗谱系树

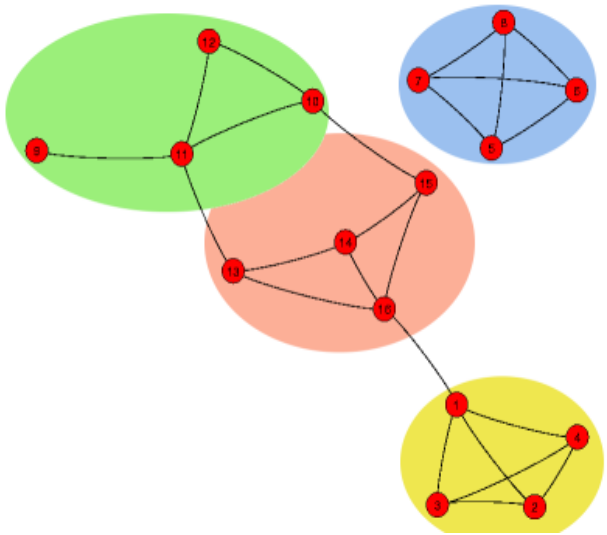


问题：在哪里切割这棵树呢？依据是什么？

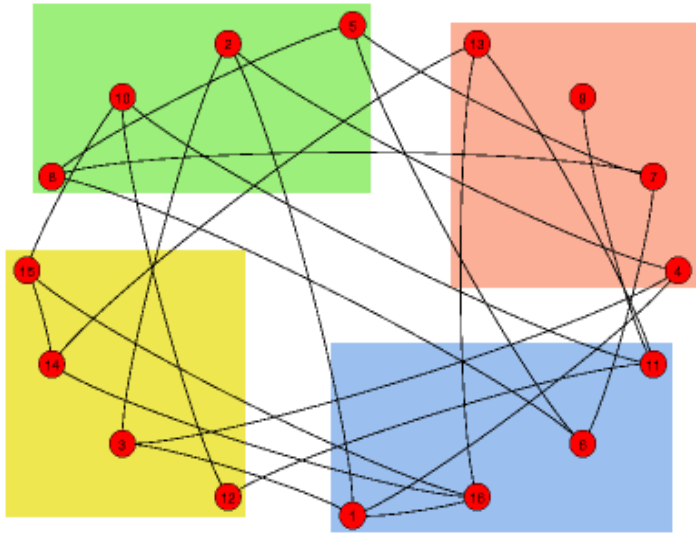
模块度

Mark Newman

- 2004年
  - Newman & Girvan, PRE
- 回答的问题：什么样的网络划分是一个好划分？



好的划分



不好的划分

直观认识：内部连边多、外部连边少

模块度

- 给定一个划分$C = \{C_1, C_2, \cdots, C_K\}$，网络邻接矩阵可以映射为$K \times K$的矩阵$e$
  - e_{st}表示C_s和C_t之间的边占所有边的比例
  - $a_s = \sum_t e_{st}$，　$b_t = \sum_s e_{st}$

注：当网络为无向网络时，$a_s = b_s$

- **模块度**

$$Q = \sum_s (e_{ss} - a_s b_s)$$

		women				
		black	hispanic	white	other	a_i
men	black	0.258	0.016	0.035	0.013	0.323
	hispanic	0.012	0.157	0.058	0.019	0.247
	white	0.013	0.023	0.306	0.035	0.377
	other	0.005	0.007	0.024	0.016	0.053
	b_i	0.289	0.204	0.423	0.084	

Q=(0.258-0.323*0.289)+(0.157-0.247*0.204)+(0.306-0.377*0.423)+(0.016-0.053*0.084)=0.43

模块度的性质

- 取值范围
  - -1和1之间
  - 值越大，划分质量越好
- 可加性
  - 社区上的定义和节点上的定义一致

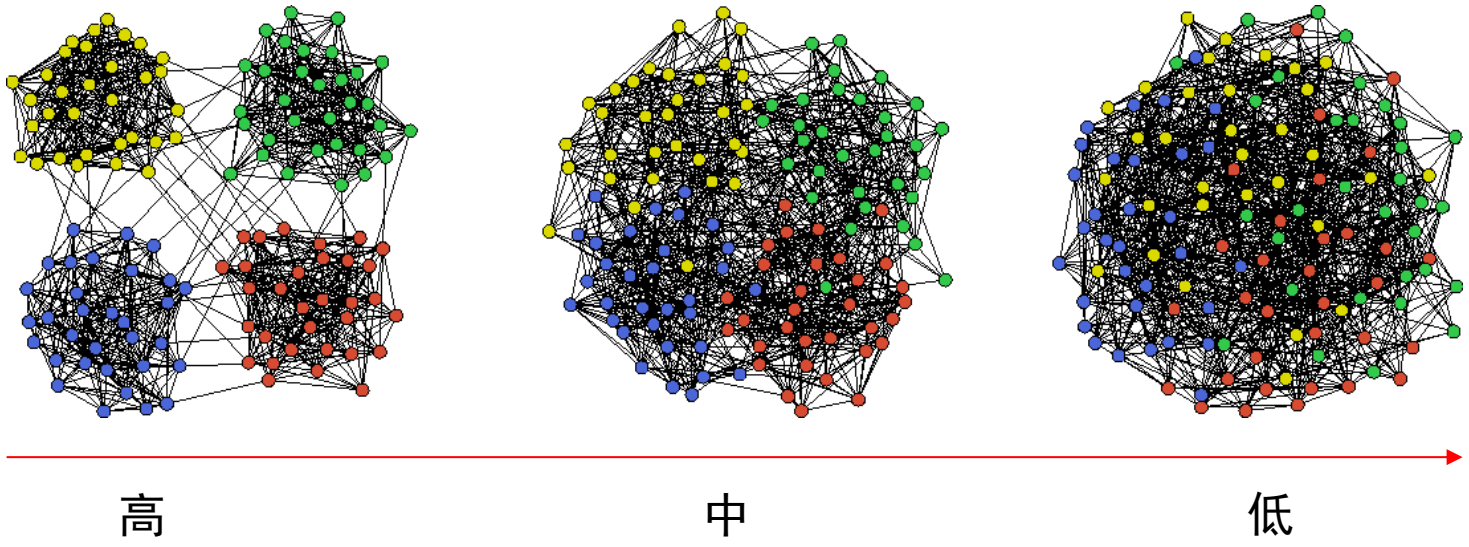
$$Q = \sum_s (e_{ss} - a_s b_s)$$

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

注：$m = \frac{1}{2} \sum_{i,j} A_{ij}$

模块度的价值

- 给定一个网络，评价其各个划分的好坏
- 评估网络的模块化程度

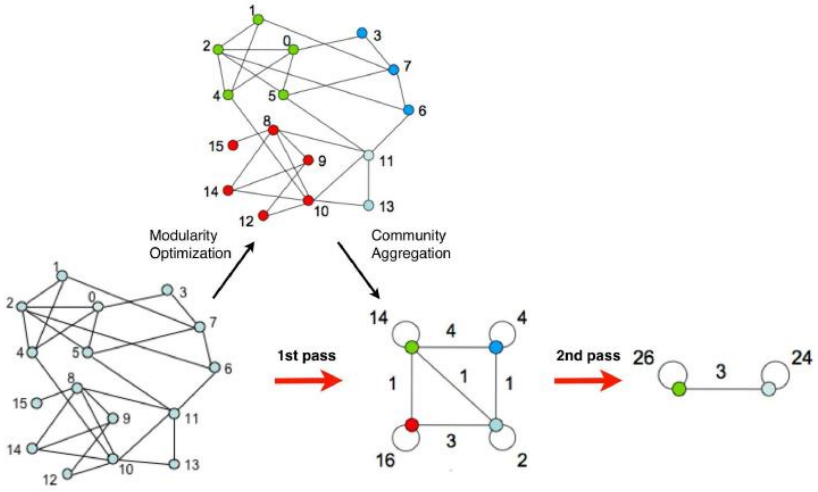


模块度优化

- 社区发现问题变成了模块度优化问题
  - 给定一个网络，寻找该模块度最大的划分
  - NP-hard问题
- 求解方法
  - 模拟退火[Nature, 2004]
    - 遍历所有可能的划分，寻找模块度最大的划分
  - 贪心算法[Newman, Phys. Rev. E, 2004]
    - 初始时每个节点视为一个社区
    - 每次选择使模块度增加最大的社区进行合并
  - 局部优化[Blondel et al., J. Stat. Mech., 2008]
  - 谱优化[Newman, PNAS, 2006]
  - ……

模块度优化算法示例-局部优化

- 局部优化
  - 初始化：每个节点属于一个社区
  - 步骤1
    - 对于每个节点，判定加入其邻居节点所属的社区是否可以增加模块度，如果能够增加，加入使模块度增加最大的社区
    - 直到所有节点所属的社区都不再变动为止
  - 步骤2
    - 将每个社区视为一个节点，构造新网络
  - 重复上述步骤至模块度不再增加



模块度优化算法示例-谱优化

- 模块度矩阵

$$B = A - P$$

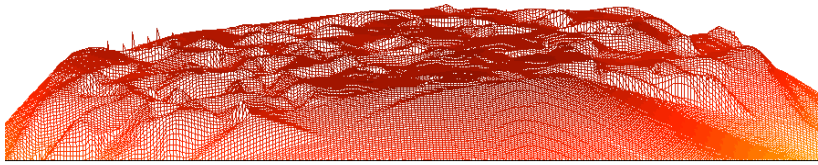
$$B_{ij} = A_{ij} - P_{ij} = A_{ij} - \frac{k_i k_j}{2m}$$

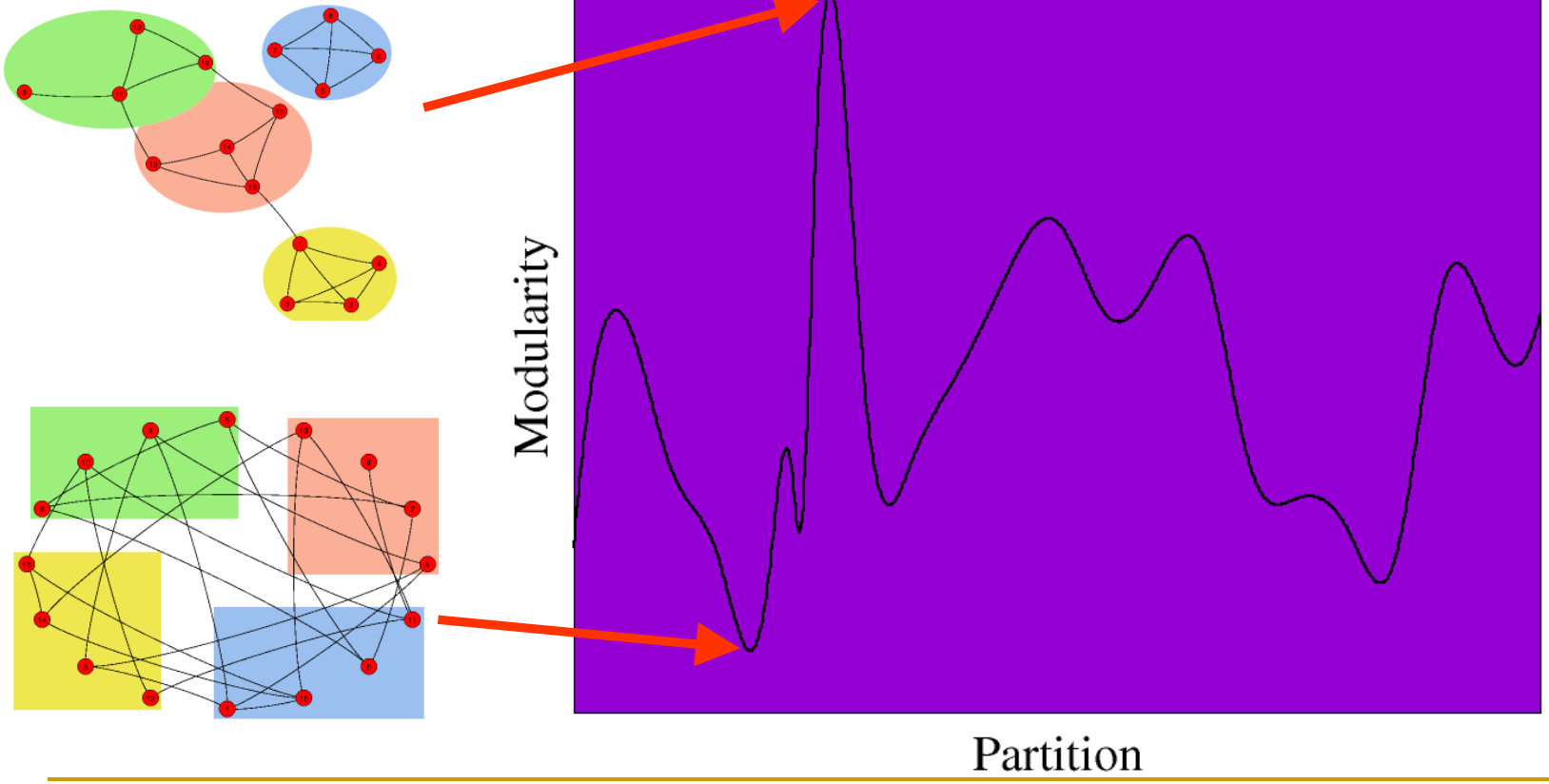
注：$m = \frac{1}{2}\sum_{i,j} A_{ij}$

- 模块度优化等价于寻找模块度矩阵最大特征值所对应的特征向量

模块度的缺陷1

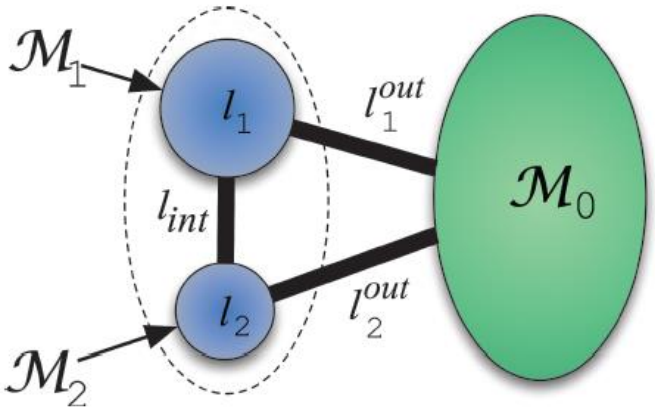
- 爬山算法 vs. 崎岖的山路

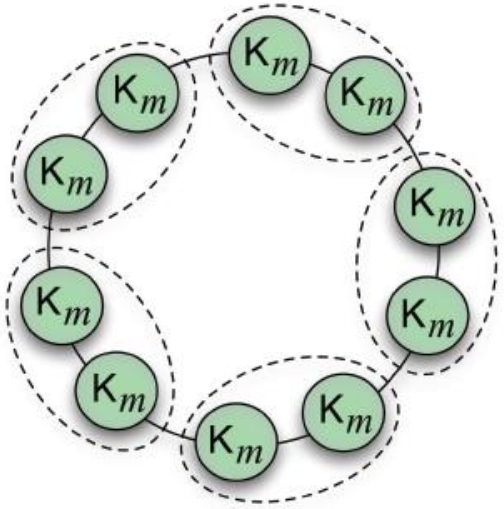




模块度的缺陷2

- 分辨率问题
  - 网络存在一个固有的尺度，小于该尺度的社区无法通过模块度优化识别出来[Fortunato and Barthélemy, PNAS, 2007]

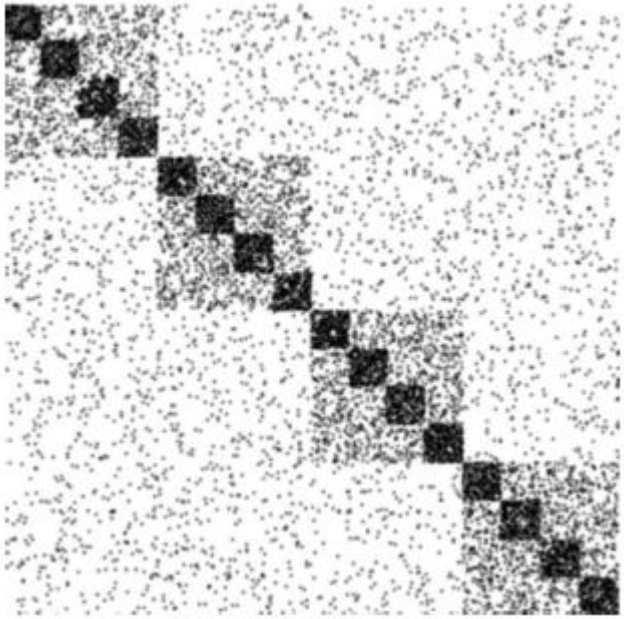




模块度分辨率问题解决方案

- 识别多尺度社区
  - 添加可调节的参数

$$Q = \sum_{ij} \left(A_{ij} - \gamma \frac{k_i k_j}{2m} \right) \delta(c_i, c_j)$$



模块度的缺陷3

- 不支持社区重叠
- 如何克服和拓展？

$$Q = \frac{1}{2m}\sum_{i,j}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta_{c(i),c(j)}$$

节点i和节点j是否属于同一个社区

$$Q_f = \frac{1}{2m}\sum_{i,j}\left(A_{ij} - \frac{k_i k_j}{2m}\right)s_{ij}$$

节点i和节点j属于同一个社区的可能性

T. Nepusz, et al., Fuzzy communities and the concept of bridgeness in complex networks. Phys. Rev. E, 77:016107, 2008.

课间休息

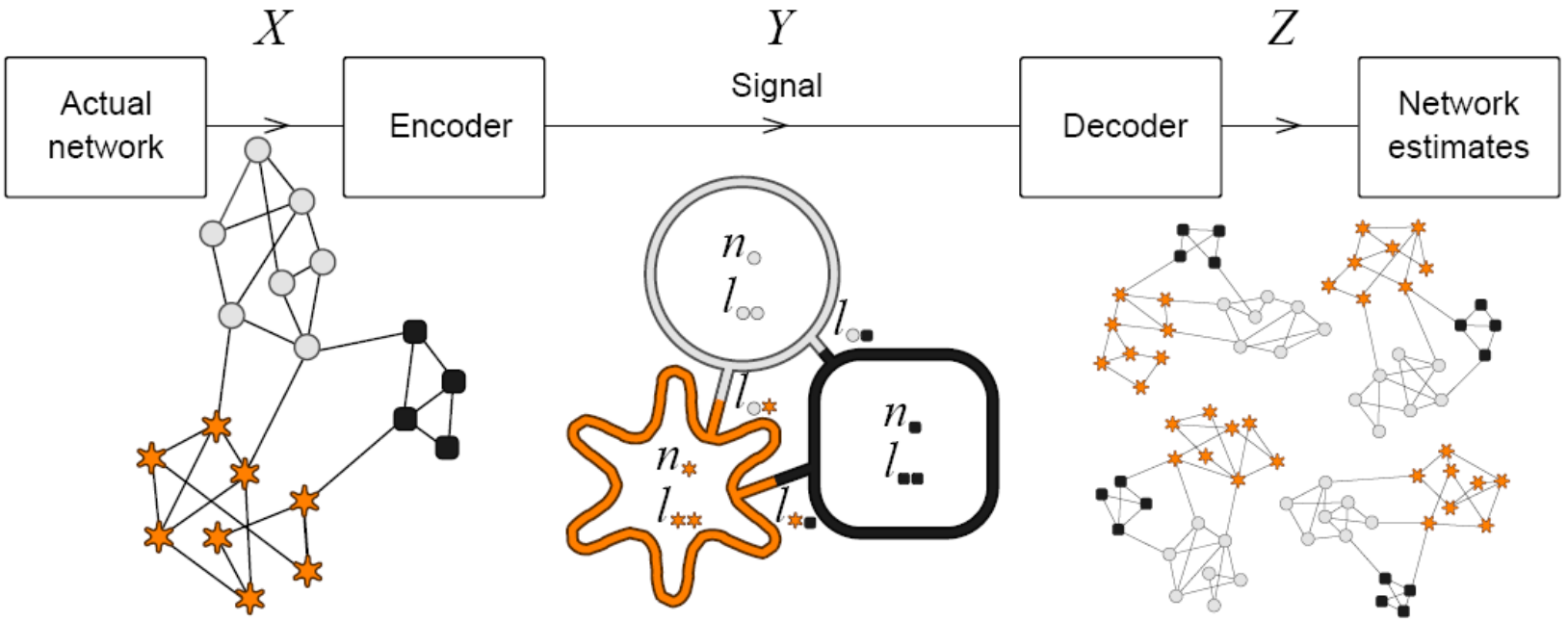
内容提纲

- 图划分
  - Min-cut, Ratio-cut, Normalized-cut
- 社区发现
  - 模块度
  - InfoMap
- 图建模
  - 非负矩阵分解
  - 随机块模型

网络通信游戏(1/2)

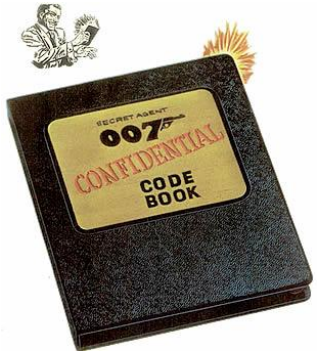
$$A_{ij}=\begin{cases}1 & \text{if there is a link between nodes } i \text{ and } j\\ 0 & \text{otherwise}\end{cases}$$

$$H(Z)=\log\left[\prod_{i=1}^{m}\binom{n_i(n_i-1)/2}{l_{ii}}\prod_{i>j}\binom{n_i n_j}{l_{ij}}\right]$$



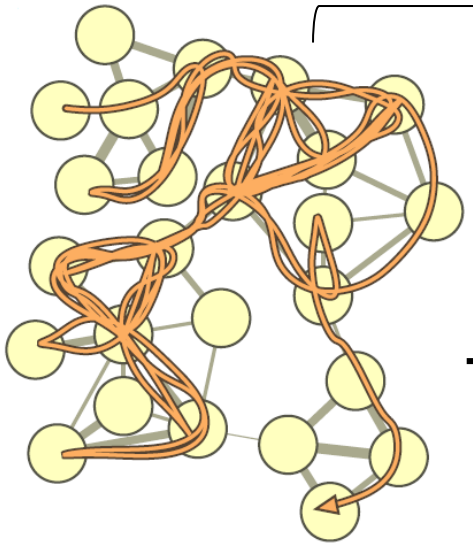
$$Y=\left\{\mathbf{a}=\begin{pmatrix}a_1\\ \vdots\\ a_n\end{pmatrix},\ \mathbf{M}=\begin{pmatrix}l_{11} & \cdots & l_{1m}\\ \vdots & \ddots & \vdots\\ l_{m1} & \cdots & l_{mm}\end{pmatrix}\right\}$$

网络通信游戏(2/2)



问题：

如何设置密码本，
使得通信代价最小？



A

111 0000 11 01 101 100 101 01 0001 0 110 011 00 110 00 111 1011 10
111 000 10 111 000 111 10 011 10 000 111 10 111 10 0010 10 011 010
011 10 000 111 0001 0 111 010 100 011 00 111 00 011 00 111 00 111
110 111 110 1011 111 01 101 01 0001 0 110 111 00 011 110 111 1011
10 111 000 10 000 111 0001 0 111 010 1010 010 1011 110 00 10 011

A在哪？

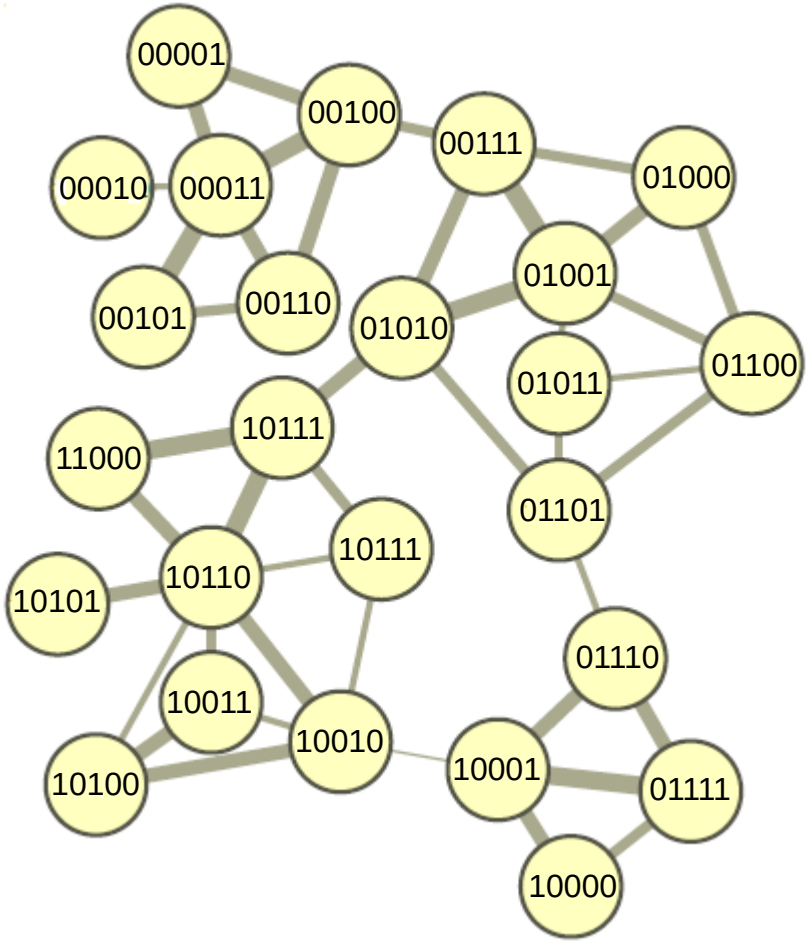
B

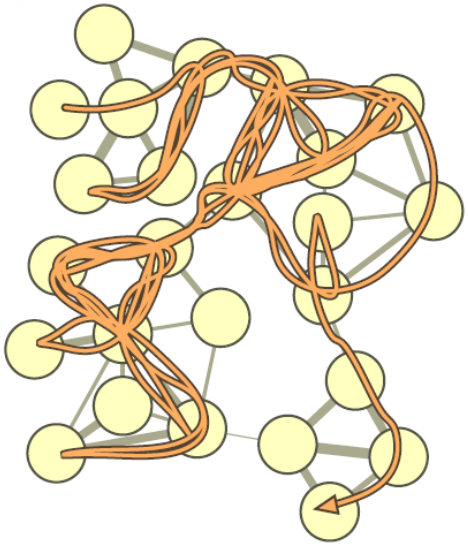
A在网络上随机行走

B在根据A发来的信号进行解码，
得到A在网络上的位置

编码方案1

- 等长编码

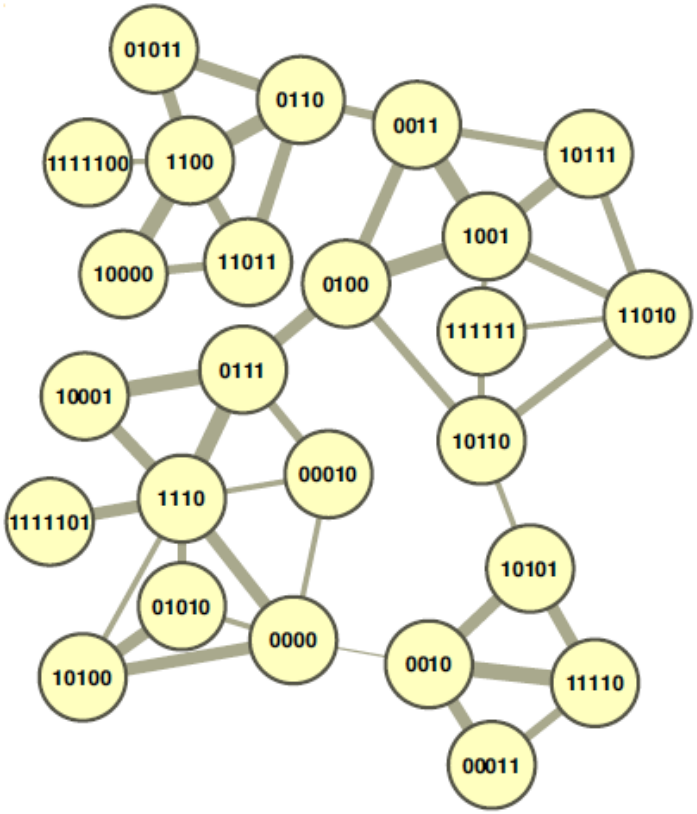




编码长度：71*5=355 bits

编码方案2

- 哈夫曼编码：单符号频率编码



1111100 1100 0110 11011 10000 11011 0110 0011 10111 1001 0011 1001 0100 0111 10001 1110 0111 10001 0111 1110 0000 1110 10001 0111 1110 0111 1110 1111101 1110 0000 10100 0000 1110 10001 0111 0100 10110 11010 10111 1001 0100 1001 10111 1001 0100 1001 0100 0011 0100 0011 0110 11011 0110 0011 0100 1001 10111 0011 0100 0111 10001 1110 10001 0111 0100 10110 111111 10110 10101 11110 00011

平均编码长度：4.5 bits

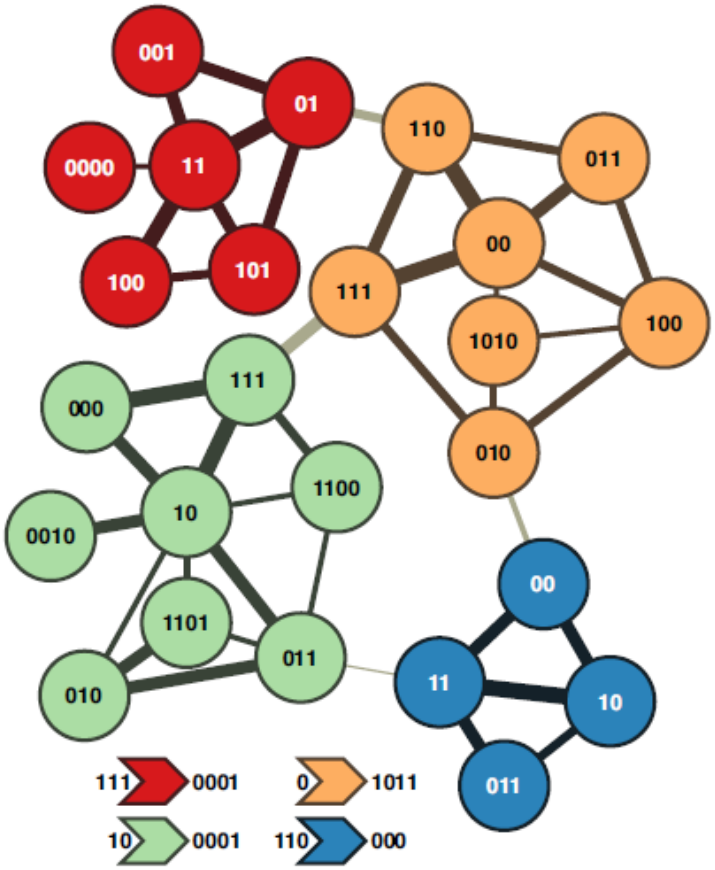
编码方案3

- 还有更好的编码吗？
  - 两级编码
    - 华盛顿大街、解放路
  - 利用网络社区结构，进行编码

- 洞察
  - 将编码问题变成社区发现的对偶问题
  - 寻找网络最优二级编码对应网络社区发现

InfoMap

- 两级编码



111 0000 11 01 101 100 101 01 0001 0 110 011 00 110 00 111 1011 10
111 000 10 111 000 111 10 011 10 000 111 10 111 10 0010 10 011 010
011 10 000 111 0001 0 111 010 100 011 00 111 00 011 00 111 00 111
110 111 110 1011 111 01 101 01 0001 0 110 111 00 011 110 111 1011
10 111 000 10 000 111 0001 0 111 010 1010 010 1011 110 00 10 011

平均编码长度：3.05 bits

InfoMap

- 解码

111 0000 11 01 101 100 101 01 0001 0 110 011 00 110 00 111 1011 10
111 000 10 111 000 111 10 011 10 000 111 10 111 10 0010 10 011 010
011 10 000 111 0001 0 111 010 100 011 00 111 00 011 00 111 00 111
110 111 110 1011 111 01 101 01 0001 0 110 111 00 011 110 111 1011
10 111 000 10 000 111 0001 0 111 010 1010 010 1011 110 00 10 011

InfoMap：编码过程

- 平均编码程度

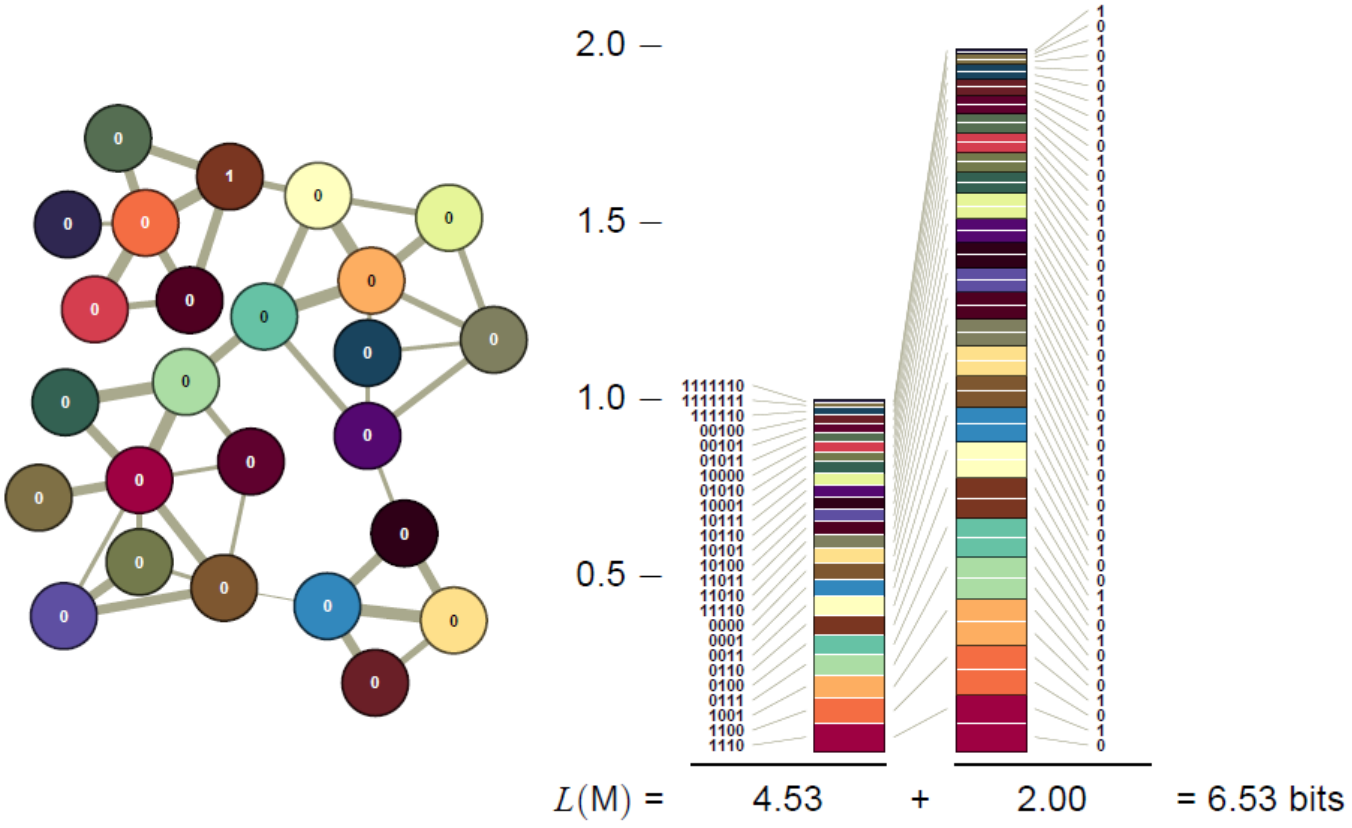
$$L(\mathsf{M}) = q_{\curvearrowright} H(\mathcal{Q}) + \sum_{i=1}^{m} p^{i}_{\circlearrowright} H(\mathcal{P}^{i})$$

这里，M是编码方案，包括一级密码本Q，和m个二级密码本P_i

- 两级哈夫曼编码
  - 社区之间
  - 社区内部
    - 需要退出码

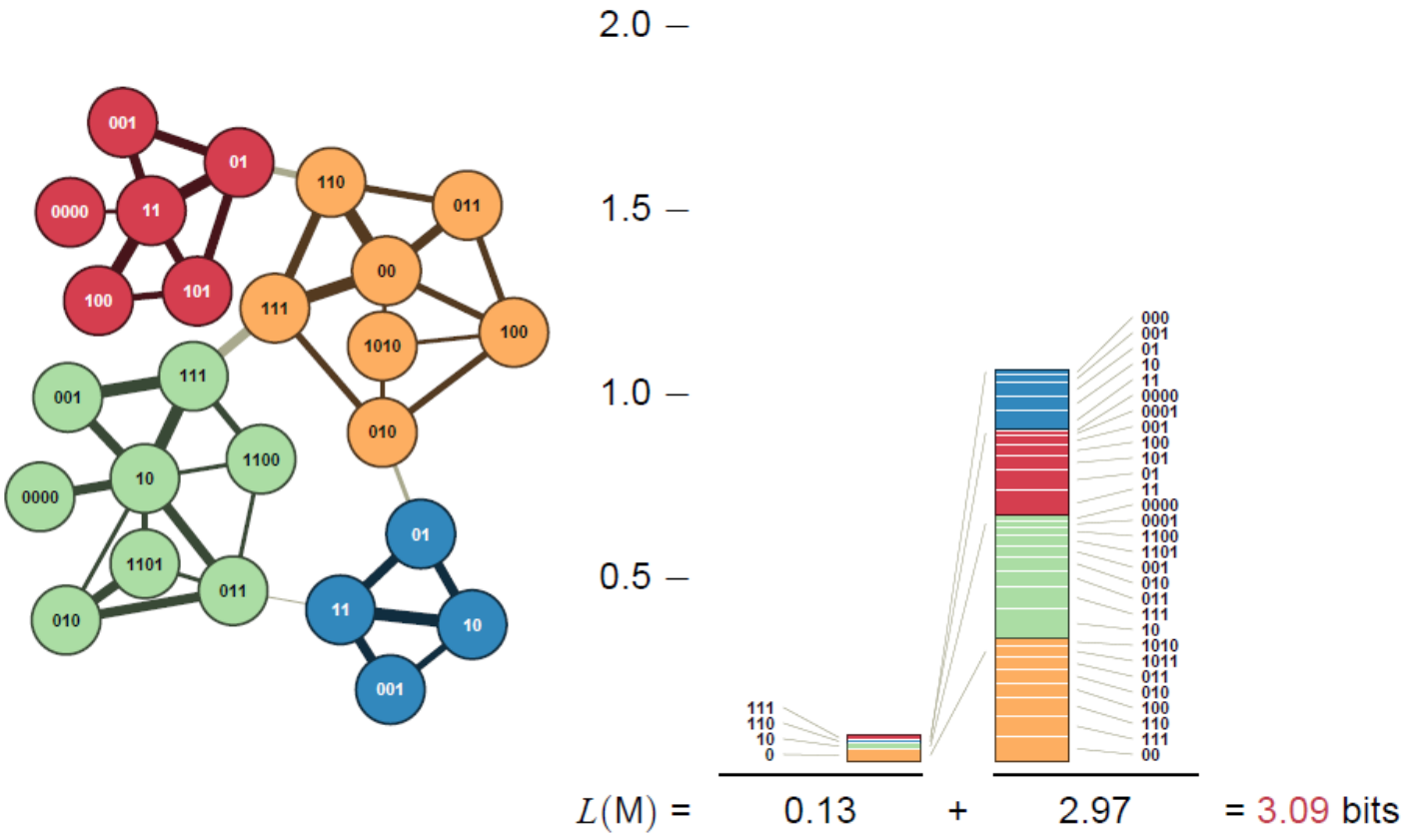
编码示例

- 社区比较小时


2.0
1.5
1.0
0.5
1111110
1111111
111110
00100
00101
01011
10000
01010
10001
10111
10110
10101
10100
11011
11010
11110
0000
0001
0011
0110
0100
0111
1001
1100
1110
L(M) =
4.53
+
2.00
= 6.53 bits

编码示例

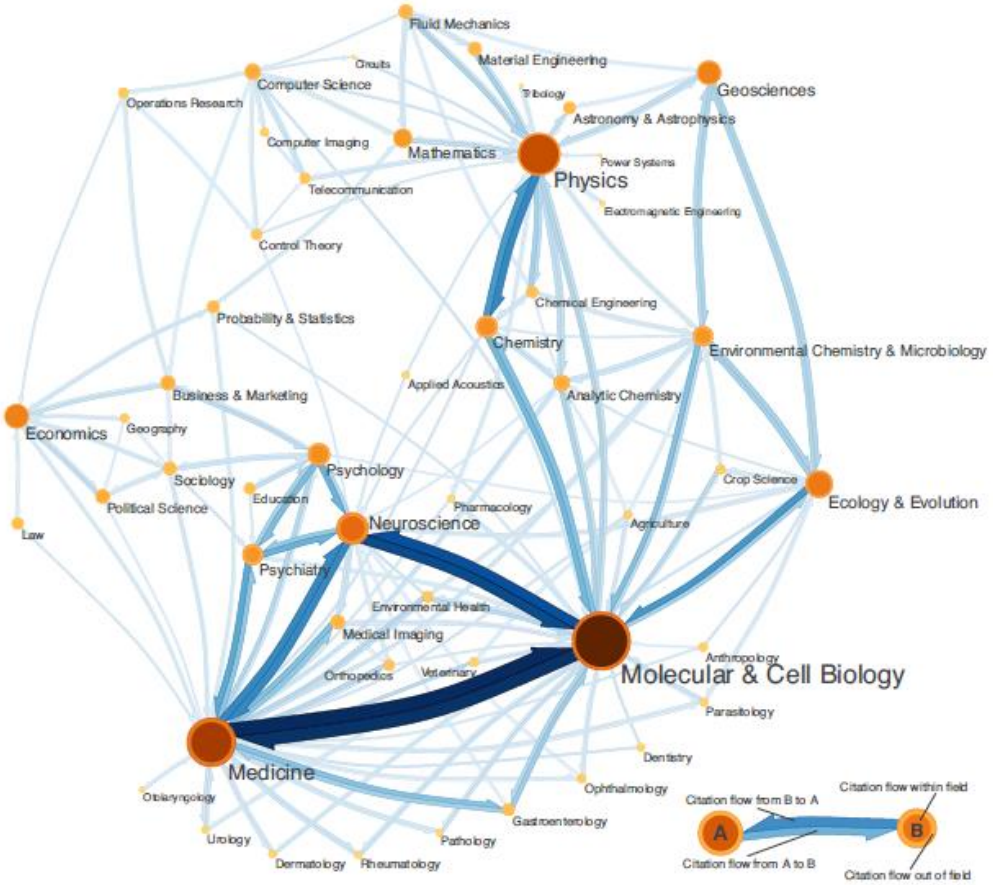
- 体现社区结构的编码



InfoMap应用案例

- Map of Science

期刊引用网络



内容提纲

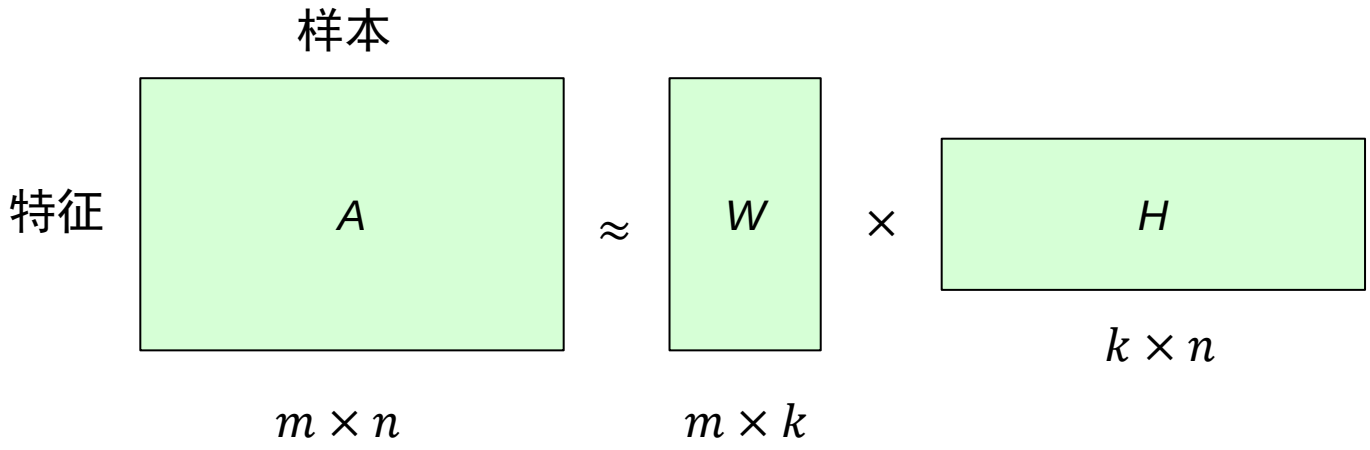
- 图划分
  - Min-cut, Ratio-cut, Normalized-cut
- 社区发现
  - 模块度
  - InfoMap
- 图嵌入 非欧空间->欧式空间，保持某性质不变
  - 非负矩阵分解
  - 其他图嵌入方法

非负矩阵分解

- 非负矩阵分解——正态分布的损失函数

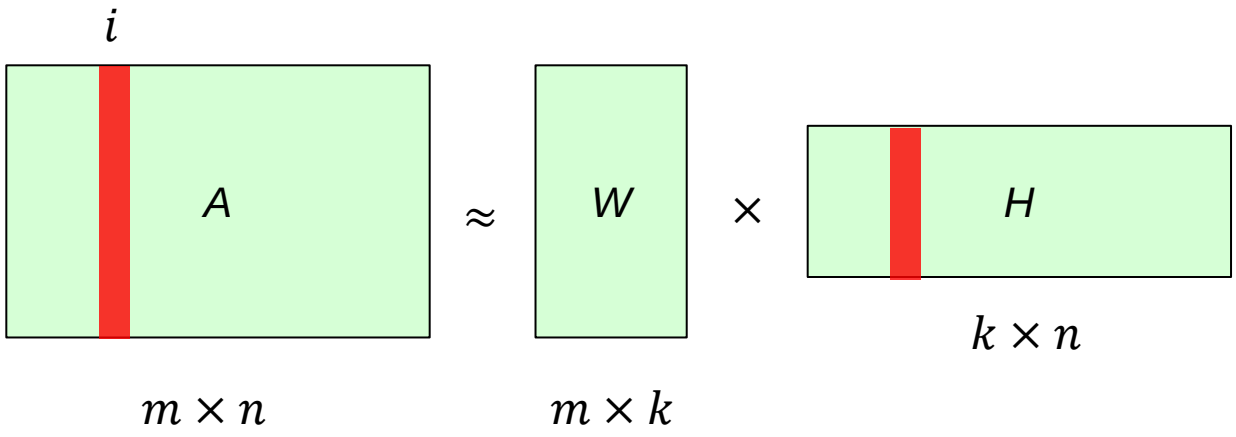
$$\arg\min_{W,H} \|A - WH\|$$

$$s.t. W \geq 0; H \geq 0$$



非负矩阵分解

- 非负矩阵分解的解读



原始表示$A_{\cdot i}$　　基向量$W_{\cdot 1}, \cdots, W_{\cdot k}$　　新表示$H_{\cdot i}$

$$A_{\cdot i} = H_{1i}W_{\cdot 1} + H_{2i}W_{\cdot 2} + \cdots + H_{ki}W_{\cdot k}$$

非负矩阵分解

- 泊松分布的损失函数

$$\arg\max_{W,H} \{A\log(WH) - WH\}$$

$$s.t. W \geq 0; H \geq 0$$

非负矩阵分解

- 扩展
  - 对称性：适用于图聚类

$$\arg\min_{W,H} \|A - WW^T\|$$

$$s.t. W \geq 0$$

  - 正交性：适用于稀疏聚类

$$\arg\min_{W,H} \|A - WH\|$$

$$s.t. W \geq 0; H \geq 0;\ W^T W = I; H^T H = I;$$

内容提纲

- 图划分
  - Min-cut, Ratio-cut, Normalized-cut
- 社区发现
  - 模块度
  - InfoMap
- 图嵌入
  - 非负矩阵分解
  - 其他图嵌入方法

图嵌入

- 图嵌入是目前图数据挖掘的一个研究热点，目标是将图数据从高维稀疏的非欧空间嵌入到一个低维的欧式空间

- 近几年提出的一些方法
  - DeepWalk [Perozzi et al., KDD 2014]
  - LINE [Tang et al., WWW 2015]
  - SDNE[Wang et al., KDD 2016]

B. Perozzi, R. Al-Rfou, S. Skiena. DeepWalk: Online Learning of Social Representations, KDD 2014.
J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, Q. Mei. LINE: Large-scale Information Network Embedding, WWW 2015.
D. Wang, P. Cui, W. Zhu. Structural Deep Network Embedding. KDD 2016.

参考文献

- M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E, 74: 036104, 2006.
- U. von Luxburg. A tutorial on spectral clustering. Stat. Comput. 17: 395-416, 2007.
- M. Girvan, M. E. J. Newman. Community structure in social and biological networks. PNAS, 99: 7821-7826, 2002.
- M. E. J. Newman, M. Girvan. Finding and evaluating community structure in networks. Phys. Rev. E, 69: 026113, 2004.
- Blondel et al., Fast unfolding of communities in large networks. J. Stat. Mech., 2008
- M. Rosvall, C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. PNAS, 105: 1118-1123, 2008.
- E. M. Airoldi et al. Mixed Membership Stochastic Blockmodels. JMLR, 9: 1981-2014, 2008.
- J. Leskovec. Kronecker Graphs: An Approach to Modeling Networks. JMLR, 11: 985-1042, 2010.