日期: /

Louvian Algorithm

Louvian 算法是一种对图中社群结构的贪婪搜索，时间复杂度为 $O(n\log n)$。支持有权图和层次化聚类。

Louvian 算法每轮包含2个阶段：

one node per community for start

Phase 1：将图中的每个节点置入独立的社区，对每个节点 i，计算将其移入其他社区产生的模块度增益 ΔQ，并将其移入 ΔQ 最大的社区，直至达到局部极大值

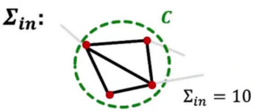
处理节点的顺序无关紧要

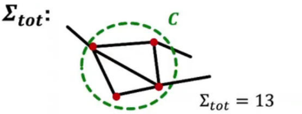
$$\Delta Q(D \to i \to C) = \underbrace{\Delta Q(D \to i)}_{\text{将}i\text{移出}D\text{的影响}} + \underbrace{\Delta Q(i \to C)}_{\text{将}i\text{移入}C\text{的影响}}$$

$\Delta Q(i \to C)$ 的推导：

定义 $\Sigma_{in} \equiv \sum_{i,j\in C} A_{ij}$ 为 C 中所有节点之间的连接权重之和

$\Sigma_{tot} \equiv \sum_{i\in C} k_i$ 为 C 中节点的所有连接权重之和





日期:　　/

那么，$Q(C) \equiv \frac{1}{2m}\sum_{i,j\in C}\left[A_{ij}-\frac{k_i k_j}{2m}\right] = \frac{\sum_{i,j\in C}A_{ij}}{2m} - \frac{(\sum_{i\in C}k_i)(\sum_{j\in C}k_j)}{(2m)^2}$

$= \frac{\Sigma_{in}}{2m} - \left(\frac{\Sigma_{tot}}{2m}\right)^2$

进一步定义 $k_{i,in} \equiv \sum_{j\in C}A_{ij} + \sum_{i\in C}A_{ij}$，为节点 i 与 C 中节点的连接权重之和

k_i 为节点 i 的度数

则 $Q_{Before} = Q(C) + Q(\{i\})$

$= \left[\frac{\Sigma_{in}}{2m} - \left(\frac{\Sigma_{tot}}{2m}\right)^2\right] + \left[0 - \left(\frac{k_i}{2m}\right)^2\right]$

$Q_{After} = Q(C+\{i\})$

$= \frac{\Sigma_{in}+k_{i,in}}{2m} - \left(\frac{\Sigma_{tot}+k_i}{2m}\right)^2$

故 $\Delta Q(C\to 1) = Q_{After} - Q_{Before}$

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

$\left[\frac{\Sigma_{in}}{2m} - \left(\frac{\Sigma_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right]$

类似地可表示出 $\Delta Q(i\to D)$，从而表示出 $\Delta Q(C\to 1\to D)$

对于每个节点，可以迭代地计算 $\Delta Q(C\to 1\to C')$，当 $\Delta Q>0$ 时将 C 移入 C'，直至达最优

日期: /

Phase 2：将每个集群压缩为单个节点，对于任意2个集群 C_1, C_2，其间的连接权重定义为 $\sum_{i\in C_1, j\in C_2} A_{ij}$，对于集群 C，其自环的连接权重为 $\sum_{i,j\in C} A_{ij}$

然后对社群聚合后的图执行 Phase 1 操作，从而实现层次化聚类。