08 - Clustering

聚类

- 定义：给定一个包含 n 个元素的集合 U，k-clustering 是将 U 划分为 k 个非空子集 $C_1, \ldots, C_k$
- 定义：k-clustering 的间隔是不同聚类中任意两点之间的最小距离
- 目标：在所有可能的 k-clustering 中，找到拥有 maximum possible spacing 的聚类，即找到一个使得不同聚类之间的最小距离最大化的聚类

贪心算法实现聚类

1. 选择两个具有最小距离 $d(p_i, p_j)$ 的对象 p_i 和 p_j
2. 用边 $e = (p_i, p_j)$ 连接它们
3. 继续这样做，直到我们得到 k 个聚类
4. 如果正在考虑的边 e 连接了已经在同一组件中的两个对象 p_i 和 p_j，则跳过它

这其实和 Kruskal 算法计算最小生成树几乎一样：

1. 选择具有最小代价 $d(p_i, p_j)$ 的边 (p_i, p_j)
2. 将该边包含在输出中
3. 继续这样做，直到我们连接所有节点
4. 如果正在考虑的边 e 引入了一个循环，则跳过它

（上面的看看就行了）
因此，我们便得到了算法

Kruskal算法用于聚类

1. 选择具有最小代价 $d(p_i, p_j)$ 的边 (p_i, p_j)
   - 最小代价就是最短距离，使用什么距离度量则根据实际要求来决定
2. 将该边包含在输出中
3. 在还剩余 $k-1$ 条边时停止
   - 这和我们生成了 MST 之后移除 $k-1$ 条代价最高的边是一样的
4. 如果正在考虑的边 e 引入了一个循环，则跳过它

算法正确性

Lemma：设 $C_1, \ldots, C_k$ 是通过在最小生成树 T 中删除 $k-1$ 条代价最高的边所形成的 k 个连通分量。这些连通分量构成了最大间隔的 k-聚类

设 $C = \{C_1, \ldots, C_k\}$，C 显然是一个聚类
将最小生成树中 $k-1$ 条最昂贵的边按照非递增顺序排列：$e_{k-1}, e_{k-2}, \ldots, e_1$
那么 C 的 spacing 就是 e_1 的代价

设 $C' = \{C'_1, \ldots, C'_k\}$ 是任何其他 k-聚类（并非通过在最小生成树 T 中删除 $k-1$ 条代价最高的边所形成的 k 个连通分量）
由于其是其他聚类，因此存在一个 C 中的聚类 C_r，它不包含在 C' 的任何聚类中
这意味着存在属于 C_r 的点 p_i 和 p_j，它们分别属于 C' 中的不同聚类。所以我们不妨设 $p_i \in C'_i$，$p_j \in C'_j$

在 p_i 到 p_j 的路径上，设 p 是 C_i' 的最后一个节点，p' 是 C_j' 的第一个节点
那么 (p, p') 的代价至多是 e_1
这是因为被删除的边的代价最小是 e_1，而 (p, p') 的代价不可能大于 e_1

而边 (p, p') 相对于聚类 C' 是一条跨越边，其距离至少是 $d(p, p')$，从而因此，C' 的间隔不大于 C 的间隔