1. 什么是 cluster:

- Intra-cluster　minimized　簇内距离最小
- Inter-cluster　maximized　簇群间距离最大

2. 聚类算法分类:

- Partitional clustering　分区聚类　分成不重叠子集
- Hierarchical clustering　层次聚类　树状.

3. 欧式距离:

假定两个随机点 $[x_1, x_2, \cdots, x_d]$ 和 $[y_1, y_2, \cdots, y_d]$ 之间距离为:

$$\sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$$

4. 群组间差异:

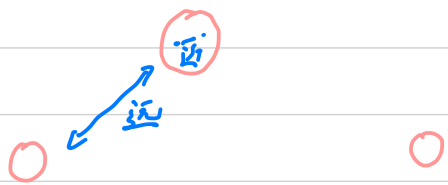
- exclusive / none-exclusive (排他-非排他): 点仅属于一个 cluster 还是可属多个.
- fuzzy cluster: (模糊聚类) 非排他的一种,属于某个 cluster 的概率在0和1之间.
- Partial / complete: 部分/完全

5. types of clusters:

① well-separated cluster 分隔良好的 clu
② prototype-based cluster 基于原型→中心点
③ contiguity-based cluster 基于连续
④ Density-based cluster 基于密度
⑤ described by an objective function 用目标函数描述

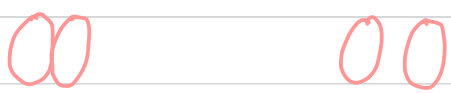
① well-separated cluster 分割良好.

聚类中任何一点 都比 不在聚类中任何一点 更接近 聚类中其他每一点.



② Prototype-based

聚类中的对象与聚类的中心点 比 与其他聚类的中心点 都近.



→ 适用于文件和图数据

③ contiguous cluster 连续

每个点 到簇中至少一个点的距离 比到不同簇中任意点距离更近

噪声 异常值

④ Density-based 密度 → 存在 noise 和 outliers 时用.

聚类是密度高区域，被低密度围绕.

⑤ defined by an objective function:

用函数衡量点分到聚类中的效果好坏.

6. K-means：提前知道要分成k个cluster. 每个cluster都有一个中心点. 每个点都指派到距离中心点最近的簇.

点数↑ 聚类数↑ →迭代数 →属性数(不变)

过程：$O(n*k*I*d)$

初始化：找k个点做k个簇的中心点.

repeat：→中心点一般取平均值.

把每个点指派到对应的簇.

重新计算每个cluster的中心点.

until 中心点不再变化. →通常改为较少点发生变动.

SSE (sum of squared error)

$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} dist^2(m_i, x)$ 每个点到中心点距离平方之和.

从k个cluster选初始中心，每个cluster选一个点的概率

$P = \frac{k!}{k^k}$

7. 关于K-means选中心点问题：

① 多跑几次.

② K-means++ →选距离最大的分割

③ 用hierachical选 →分层聚类

④ Bisecting K-means

k-means++ 选取初始中心点过程：→ 与k-means只有选中心点过程不一致（初始）

1. 从数据集中随机选一个点为中心点
2. 计算剩余每个点到目前已选定的中心点集合的最小距离.
3. 根据正比于最小距离的值作为选该点为中心点的概率
4. until 选出k个中心点.

Bisecting k-means：通过二分法，选2簇总SSE最小的方法.

1. 初始时，将代聚类数据集D做为一个簇C_0，即$C=\{C_0\}$
2. 取C中具有最大SSE的簇C_p，进行二分实验m次，调用k-means算法，k为2实验m次，共得到m个二分结果$B=\{B_1, B_2, \cdots, B_m\}$，其中$B_i=\{C_{i1}, C_{i2}\}$

每个B_i都是一个划分结果　　每一次二分实验得到的2个簇.

3. 取上一步中具有最小总SSE的划分$B_j=\{C_{j1}, C_{j2}\}$，将C_{j1}、C_{j2}加入C中并删掉C_p.
4. 重复直到k个簇.

8. Hierarchical cluster　　无需指定k.

两种类型：
- 聚类式：从点作为单个簇开始，依次合并.
- 分裂式：将所有点视为一个簇，依次拆分.

采用 Proximity matrix（相似度矩阵计算）

↳ 可用 max、min、average group、ward's method 等.

其中 average group: $proximity(cluster_i, cluster_j) = \frac{\sum proximity(P_i, P_j)}{|cluster_i| \times |cluster_j|}$

其中 ward's method: 相似性基于两个聚类合并时平方误差的增加值

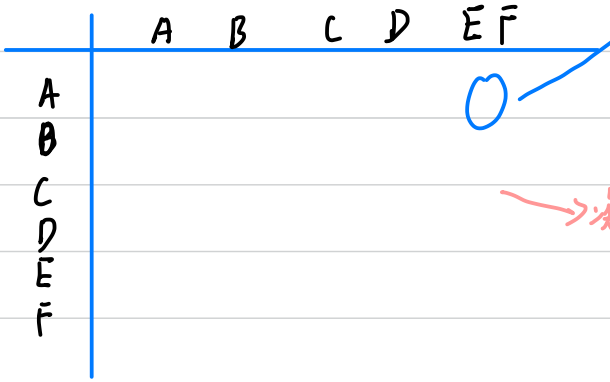
空间复杂度: $O(n^2)$　　时间复杂度: $O(n^3)$

过程: 假定有 A B C D E F 几个点.

1. 将每个点视为一个簇

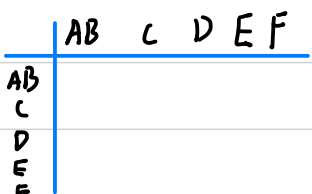
2. 计算 proximity matrix.

内容:用欧式距、max、min average group 等



→凝聚式(agglomerative)

找最小的合并 →



更新矩阵并循环直到合并成一个簇.

分裂式(Divisive)

所有点视作同一个簇,也构建 proximity matrix

每次找距离最远的,将其拆分成两个簇,直到全拆分完.

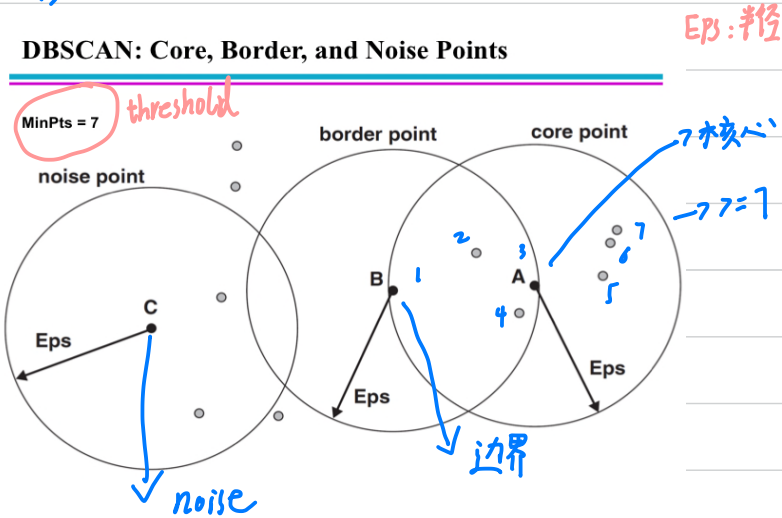
9. DBSCAN：一种density-based

① 密度density = 指定半径(EPS)内的点数.

② eps范围内至少有指定数量的点，则该点为核心点. → 位于聚类内部且参与统计. 计算点本身.

③ 边界点不是核心点，但在核心点内部.

④ 噪声指不是核心点也不是边界点的位行点.



EPS：半径

→ 核心

→ 7=7

1. 将所有点标记为核心点、边界点或噪声点

2. 消除噪声点.

3. 在距离eps范围内的所有核心点之间加一条边

4. 将相组相连的核心点组成一个独立的群组

5. 将边界点分配到其相关核心点的一个群组中.

10. 关于簇：

→ 聚类内聚性

cluster cohesion (聚类中对象间密切程度) — SSE → 无监督

cluster Separation (聚类分离度) — SSB

SSE：簇内数据点与其簇中心之间的误差平方和.

$$SSE = \sum_i \sum_{x \in C_i} (x - m_i)^2$$

SSB：每个簇的数据点与所有簇的中心之间的平方和（误差），然后对所有聚类簇求和.

$$SSB = \sum_i |C_i| (m - m_i)^2$$

$(m - m_i)^2$ → 算i簇的中心与整体所有点的中心之间距离

$|C_i|$ 是簇i的大小.

$$SSE + SSB = \text{constant}$$

Silhouette coefficient（轮廓系数）：

对于单个点 i：

a = i到该簇内各点的平均距离

b = min（i与别的簇中各点的平均距离）.

$$s = (b - a) / \max(a, b)$$

在 -1 和 1 之间，越近1越好.