Clustering

> 常用的无监督学习算法，按照某个特定标准把一个数据集分割成不同的类或簇，使得某一个簇内数据对象的相似性尽可能大，不同簇内数据对象的差异性尽可能大

聚类问题

> 对无标签的数据，基于数据分布进行分组，使得相似的数据尽量落在同一个簇

聚类与分类的区别：

1. 聚类是无监督学习，分类是从训练集学习分类的方法
2. 聚类只需人工指定相似度的标准和类别数即可，分类需从训练集学习分类的方法

主流聚类方法

1. 划分聚类(Partitioning Clustering)：给出一系列扁平结构的簇(分开的几个类)，它们之间无任何显示结构来表明彼此的关联性

   > 常见算法有：K-Means/K-Medoids、Gaussian Mixture Model、Spectral Clustering、Centroid-based Clustering

2. 层次聚类(Hierarchical Clustering)：输出一个具有层次结构的簇集合

   > 常见算法有：Single-linkage、Complete-linkage、Connectivity-based Clustering

划分聚类

K-Means

将n个数据点按照分布分成K类，通过聚类算法得到K个中心点，以及每个数据点属于哪个中心点的划分。中心点可以通过**迭代算法**得到，满足条件：所有数据点到聚类中心的距离之和是最小的

> Q1：数据点到中心点距离计算：选择几何距离(L_2距离的平方)
>
> Q2：中心点是否唯一：理论存在，但是找局部最优解
>
> Q3：聚类结果如何表示：采用空间分割的方式，将空间分割成多个多边形，每个多边形对应一个cluster中心

算法步骤

采用EM算法(Expectation Maximization Algorithm)迭代确定中心点：

1. `更新中心点`：初始化时随机取点作为起始点；迭代过程中，取同一类所有数据点的重心(算数平均值)作为新中心点
2. `分配数据点`：把所有数据点分配到最近的中心点

重复直至中心点不再改变

K-Medoids算法

针对K-Means算法给出的改进：

1. 限制聚类中心点来自数据点

   > 求中心点方法：计算出同一类所有数据点的重心之后，在重心附近找一个数据点作为新的中心点

2. 距离计算由平方变为绝对值(L_2距离变为L_1距离，避免对离群点的敏感)
3. 起始点是任选数据集中的点，而不是随机点

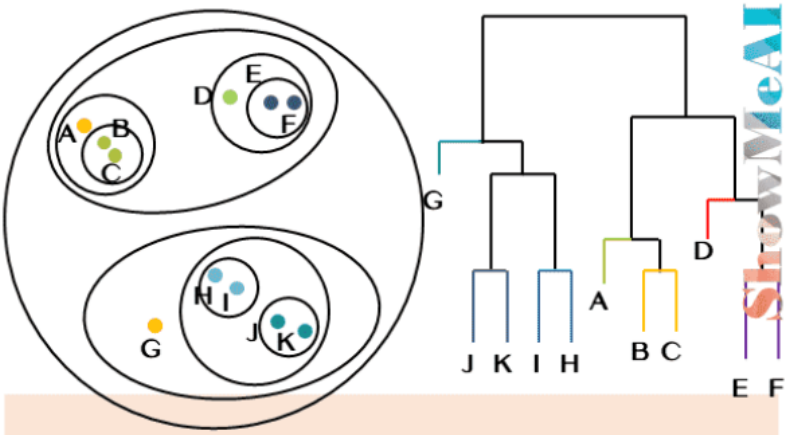
层次聚类

Single-Linkage算法

构造一棵二叉树，用叶结点代表数据，每个二叉树每一个内部结点代表一个聚类。是一个从下而上的聚类，先有叶子再有树根。将两个类之间的距离定义为**两个类中距离最小的两个点**$d(S_i, S_j) = \min_{x_i \in S_i, x_j \in S_j} ||x_i - x_j||$

算法过程

1. 选择距离最近的两个类进行合并
2. 将被合并的两个类从现有类中删除
3. 将合并后得到的新类加入现有类中
4. 迭代直至只有一个类



从下而上的聚类：这棵树先有叶子，顺着叶子逐渐长树枝，一直到树根。图中有11个数据点，一共发生了10次合并。

Complete-Linkage算法

迭代思路与Single-Linkage算法相同，但是将两个类之间的距离定义为**两个类中距离最大的两个点**$d(S_i, S_j) = \max_{x_i \in S_i, x_j \in S_j} ||x_i - x_j||$

DB-SCAN算法

> 基于密度的聚类

1. 核心对象(Core Object)：如果x_j的$\epsilon-$邻域至少有$MinPts$个样本，即$|N_\epsilon(x_j)| \geq MinPts$，则$x_j$是一个核心对象
2. 密度直达(directly density-reachable)/密度可达(density-reachable)：
   - 如果x_i位于x_j的$\epsilon-$邻域中，且x_j是核心对象，则称x_j由x_i密度直达
   - 对于x_i与x_j，如果存在样本序列$p_1, p_2, \cdots, p_n$，其中$p_1 = x_i$，$p_n = x_j$且$p_i + 1$由p_i密度直达，则称x_j由x_i密度可达
3. 密度相连(density-connected)：所有密度可达的核心点构成密度相连，即对于x_i与x_j如果存在x_k使得x_i与x_j均由x_k密度可达，则称x_i与x_j密度相连

算法过程

1. 规定$MinPts$和半径范围
2. 找出核心对象：如果在半径范围内密度大于$MinPts$，则该点是核心对象。将所有的核心对象放入一个集合
3. 从核心对象集合中，随机找一个核心对象，判断其它数据点与它是否密度直达。如果密度直达，则归入聚类簇中
4. 继续判断其它点与聚类簇中的点是否密度直达，直至检查完所有点。