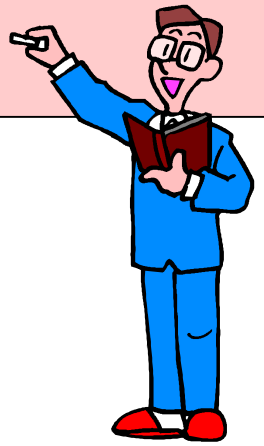
计算机模式识别与机器学习

—— 聚类分析



主讲：图像处理与模式识别研究所
赵群飞

邮　　箱：zhaoqf@sjtu.edu.cn
办公室：电院 2-441
电　　话：13918191860

第7章　聚类

- **本章学习目标**
  - ✓ 了解无监督学习
  - ✓ 理解聚类的基本思想
  - ✓ 熟练掌握常用的k-均值聚类算法
  - ✓ 了解密度聚类和层次聚类
  - ✓ 了解聚类的评价指标

目录

监督学习和无监督学习的区别

- 监督学习

在一个典型的监督学习中，训练集**有标签*y***，我们的目标是找到能够区分正样本和负样本的决策边界，需要据此拟合一个假设函数。

- 无监督学习

与此不同的是，在无监督学习中，我们的数据**没有附带任何标签*y***，无监督学习主要分为聚类、降维、关联规则、推荐系统等方面。

主要的无监督学习方法

- 聚类（Clustering）

  “物以类聚”如何将教室里的学生按爱好、身高划分为5类？

- 降维（Dimensionality Reduction）

  如何将将原高维空间中的数据点映射到低维度的空间中？

- 关联规则（Association Rules）

  很多买尿布的男顾客，同时买了啤酒，可以从中找出什么规律来提高超市销售额？

- 推荐系统（Recommender systems）

  很多客户经常上网购物，根据他们的浏览商品的习惯，给他们推荐什么商品呢？

聚类

- 聚类分析方法是决定描述一个经验数据集的结构类型的一种非参数方法。
- 相似的数据被集中在一起，从数据集中分离出来，包含在特征空间中的一个模式集，其模式的密度比起周围区域中的密度大，就为一个聚类。

- 相似性：相当于对于特征空间中的点，以特征空间中，各点之间的距离函数作为模式相似性的测量，以“距离”作为模式分类的依据，距离越小，越“相似”。

给定以某种相似测度定义的距离，希望聚类结果：

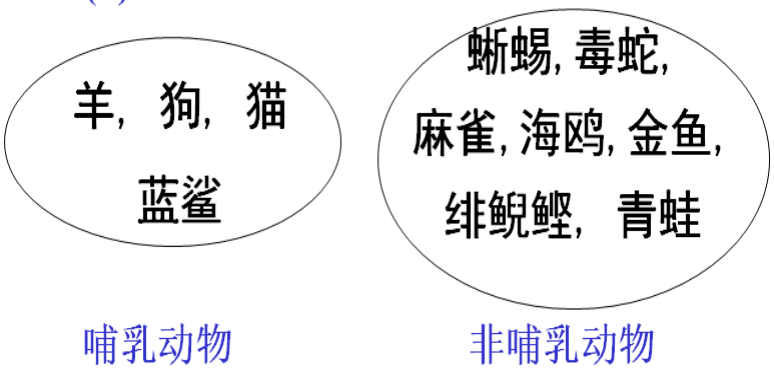
- 类间距离越大越好（异类中尽可能不同）
- 类内距离越小越好（同类中尽可能相似）

选择什么特征？
选择多少个特征？
选择什么样的量纲？
选择什么样的距离测度？

对分类结果都会产生极大影响

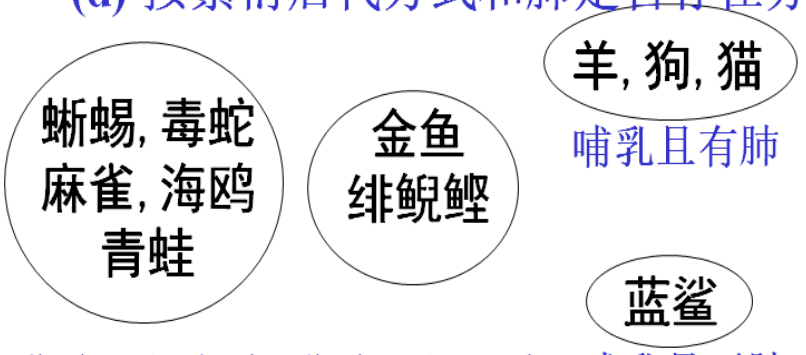
特征选取不同对聚类结果的影响

(a) 按繁衍后代的方式分



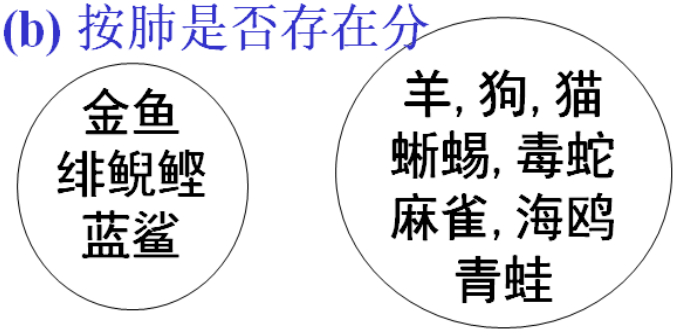
哺乳动物　　非哺乳动物

(d) 按繁衍后代方式和肺是否存在分



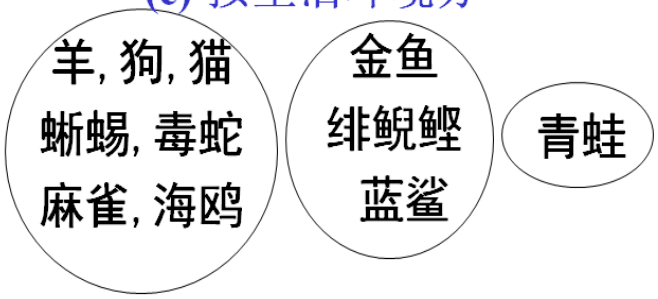
非哺乳且有肺　非哺乳且无肺　哺乳且有肺　哺乳且无肺

(b) 按肺是否存在分



无肺　　有肺

(c) 按生活环境分



陆地　　水里　　两栖

聚类

主要算法：K-means、密度聚类、层次聚类

主要应用

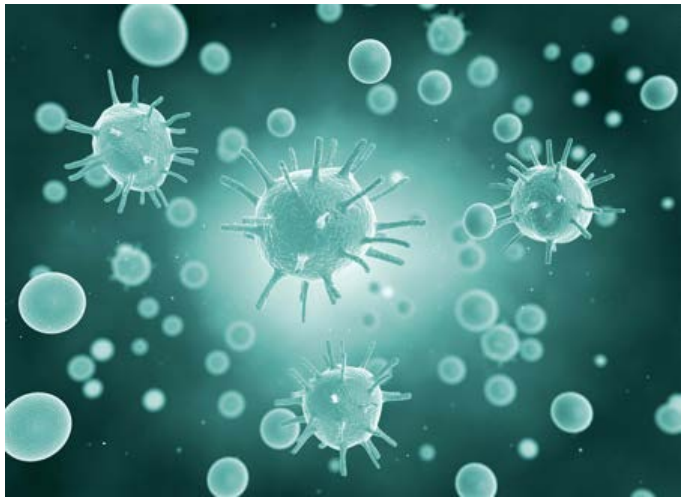
- 市场细分、文档聚类、图像分割、图像压缩、聚类分析、特征学习或者词典学习、确定犯罪易发地区、保险欺诈检测、公共交通数据分析、IT资产集群、客户画像、客户细分、识别癌症数据、搜索引擎应用、医疗应用、药物活性预测……

- 聚类还可以发现数据中的离群点，用于异常检测任务。

聚类案例

1.医疗

医生可以使用聚类算法来发现疾病。以甲状腺疾病为例：当我们对包含甲状腺疾病和非甲状腺疾病的数据集应用无监督学习时，可以使用聚类算法来识别甲状腺疾病数据集。



2.市场细分

为了吸引更多的客户，每家公司都在开发易于使用的功能和技术。为了解客户，公司可以使用聚类。聚类将帮助公司了解用户群，然后对每个客户进行归类。这样，公司就可以了解客户，发现客户之间的相似之处，并对他们进行分组。

聚类案例

3.金融业

银行可以观察到可能的金融欺诈行为，就此向客户发出警告。在聚类算法的帮助下，保险公司可以发现某些客户的欺诈行为，并调查类似客户的保单是否有欺诈行为。

4.搜索引擎

百度是人们使用的搜索引擎之一。举个例子，当我们搜索一些信息，如在某地的超市，百度将为我们提供不同的超市的选择。这是聚类的结果，提供给你的结果就是聚类的相似结果。

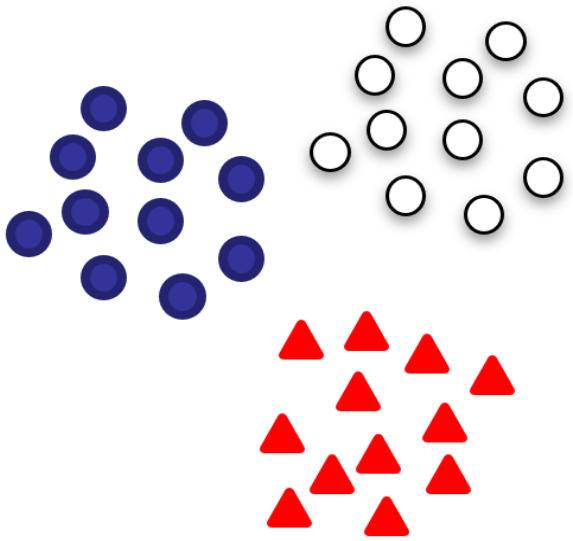
5.社交网络

比如在社交网络的分析上。已知你朋友的信息，比如经常发email的联系人，或是你的微博好友、微信的朋友圈，我们可运用聚类方法自动地给朋友进行分组，做到让每组里的人们彼此都熟识。

K-means聚类--基本思想

- K-means算法是一种无监督学习方法，是最普及的聚类算法，算法使用一个没有标签的数据集，然后将数据聚类成不同的组。如图中的数据可以分成三个分开的点集(称为簇)。

- K-means算法具有一个迭代过程，在这个过程中，数据集被分组成若干个预先定义的不重叠的聚类或子组，使簇的内部点尽可能相似，同时试图保持簇在不同的空间，它将数据点分配给簇，以便簇的质心和数据点之间的平方距离之和最小，在这个位置，簇的质心是簇中数据点的算术平均值。



聚类算法示例

设数据可分为K类，将观测值下标$\{1,2,\ldots n\}$划分为K个不相交的集合$\{\boldsymbol{C}_1,\boldsymbol{C}_2,\ldots \boldsymbol{C}_K\}$。其中$\boldsymbol{C}_k \cap \boldsymbol{C}_{k'} = \varnothing(\forall k \neq k')$，且$\boldsymbol{C}_1 \cup \boldsymbol{C}_2 \cup \cdots \cup \boldsymbol{C}_K = \{1,\ldots,n\}$,故每个观测值都有唯一类别归属。$i \in \boldsymbol{C}_K$即为第$i$个观测值$\boldsymbol{x}_i$属于第$k$个聚类。聚类的“组内变动”因该越小越好，记聚类$k$的均值或中心位置为

$$\boldsymbol{c}_k = \frac{1}{|\boldsymbol{C}_k|}\sum_{i\in \boldsymbol{C}_k}\boldsymbol{x}_i$$

其中$|\boldsymbol{C}_k|$表示聚类k的观测值个数。显然，样本中共有K个中心位置(均值)，故将此算法称为K均值聚类。

对于聚类k中的某观测值$\boldsymbol{x}_i(i \in \boldsymbol{C}_k)$，称其到聚类中心位置的离差$(\boldsymbol{x}_i - \boldsymbol{c}_k)$为“误差”。将聚类$k$中所有的误差平方和加总，即为聚类$k$的**误差平方和**，记SSE

$$SSE_k = \sum_{i\in \boldsymbol{C}_k}||\boldsymbol{x}_i - \boldsymbol{c}_k||^2$$

其中$||\boldsymbol{x}_i - \boldsymbol{c}_k||$称为欧氏距离，即各个分量平方和的算术平方根。

- 将所有聚类的误差平方和加总，即为全样本的误差平方和：

$$SSE = \sum_{k=1}^{K} \sum_{i \in \boldsymbol{C}_k} ||\boldsymbol{x}_i - \boldsymbol{c}_k||^2$$

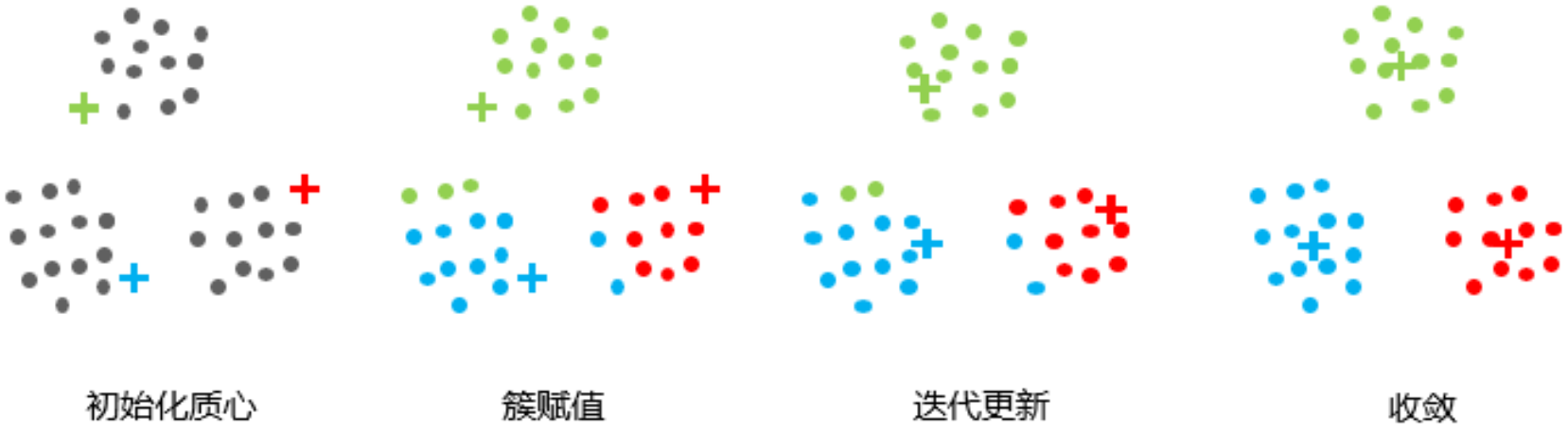
其中SSE称为组内平方总和，我们的目标是，对于样本下标集$\{1, \ldots, n\}$的一个划分$\{\boldsymbol{C}_1, \boldsymbol{C}_2, \ldots \boldsymbol{C}_K\}$，使得全样本得误差平方和最小化：

$$\min_{\boldsymbol{C}_1, \boldsymbol{C}_2, \ldots \boldsymbol{C}_K} SSE = \sum_{k=1}^{K} \sum_{i \in \boldsymbol{C}_k} ||\boldsymbol{x}_i - \boldsymbol{c}_k||^2$$

将n个观测值分到K个聚类，每个观测值有K个聚类可供选择，故共有K^n中分法。对于$K = 5, n = 100$的小样本，就多达$5^{100} \approx 7.89 \times 10^{69}$种分法。因此上述最优化问题一般很难找到全局最优解。

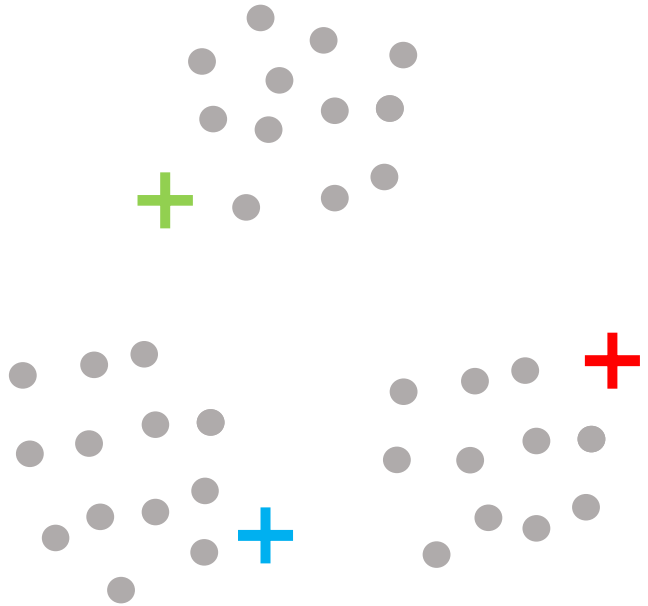
K-means 聚类算法流程

1. 选择K个点作为初始质心。
2. 将每个点指派到最近的质心，形成K个簇。
3. 对于上一步聚类的结果，进行平均计算，得出该簇的新的聚类中心。
4. 重复上述两步/直到迭代结束：质心不发生变化。



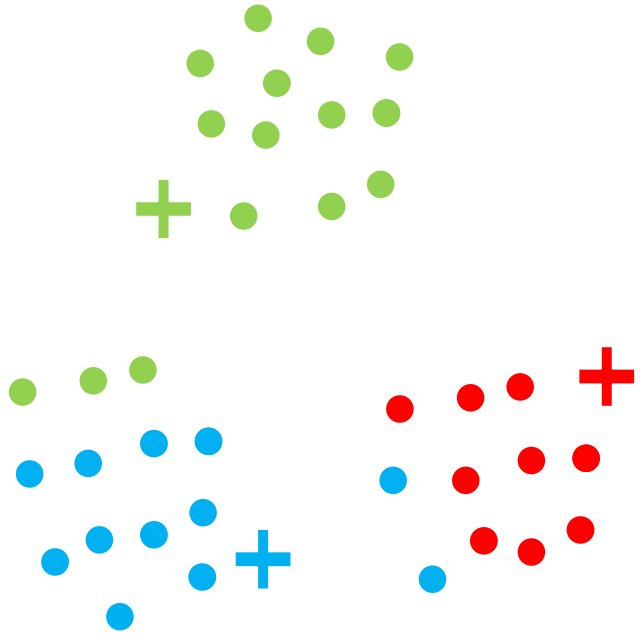
K-means 算法流程

- 首先，初始化称为簇质心的任意点。初始化时，必须注意簇的质心必须小于训练数据点的数目。因为该算法是一种迭代算法，接下来的两个步骤是迭代执行的。



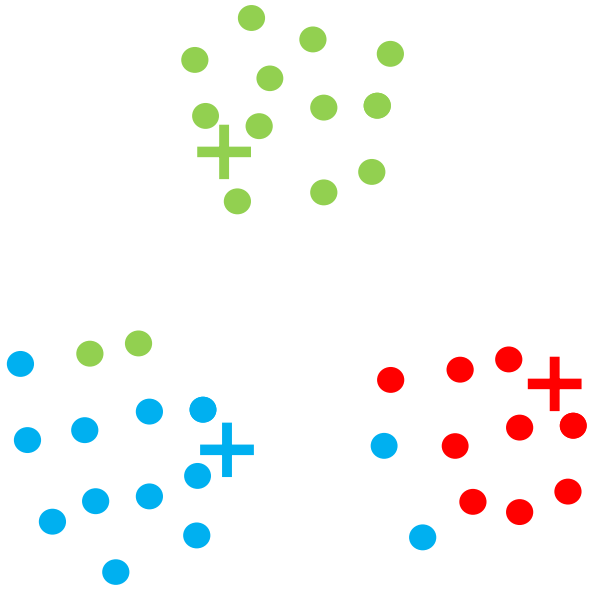
初始化质心

- 初始化后，遍历所有数据点，计算所有质心与数据点之间的距离。现在，这些簇将根据与质心的最小距离而形成。在本例中，数据分为3个簇($K = 3$)。



簇赋值

- 第三步：移动质心，因为上面步骤中形成的簇没有优化，所以需要形成优化的簇。为此，我们需要迭代地将质心移动到一个新位置。取一个簇的数据点，计算它们的平均值，然后将该簇的质心移动到这个新位置。对所有其他簇重复相同的步骤。



迭代更新

K-means优化过程

记k个簇中心为$\mu_1, \mu_2, \ldots, \mu_k$，每个簇的样本数目为$N_1$，$N_2, \ldots,$ N_k

使用平方误差作为目标函数：

$$J(\mu_1, \mu_2, \cdots \mu_k) = \frac{1}{2}\sum_{j=1}^{K}\sum_{i=1}^{N_j}(x_i - \mu_j)^2$$

对关于从$\mu_1, \mu_2, \cdots \mu_k$的函数求偏导，这里的求偏导是对第$j$个簇心$\mu_j$求的偏导。故而其驻点为：

$$\frac{\partial J}{\partial \mu_j} = -\sum_{i=1}^{N_j}(x_i - \mu_j) \xrightarrow{令} 0 \Rightarrow \mu_j = \frac{1}{N_j}\sum_{i=1}^{N_j} x_i$$

推导：

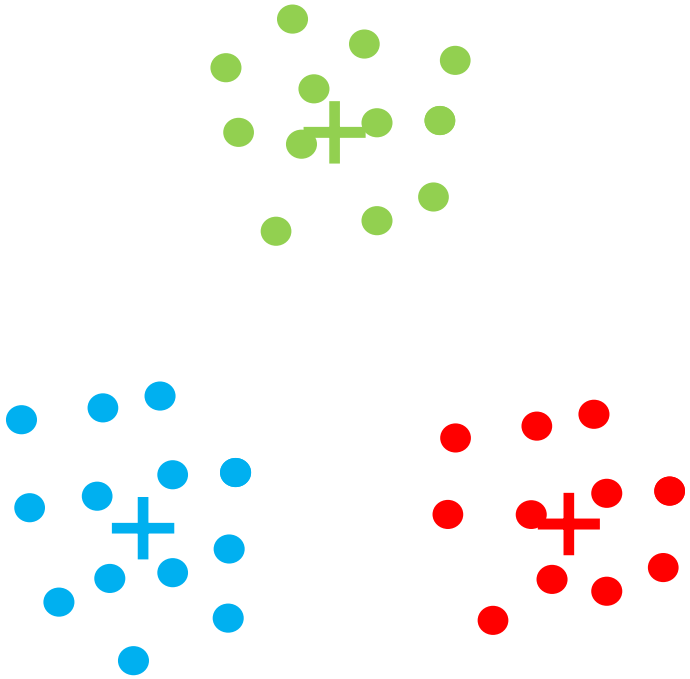
$$\frac{\partial J}{\partial \mu_j} = \frac{\partial \frac{1}{2}\sum_{j=1}^{k}\sum_{i=1}^{N_j}(x_i - \mu_j)^2}{\partial \mu_j}$$

$$= \frac{\partial \frac{1}{2}\sum_{i=1}^{N_j}(x_i - \mu_j)^2}{\partial \mu_j}$$

$$= \sum_{i=1}^{N_j}(x_i - \mu_j)\cdot(-1)$$

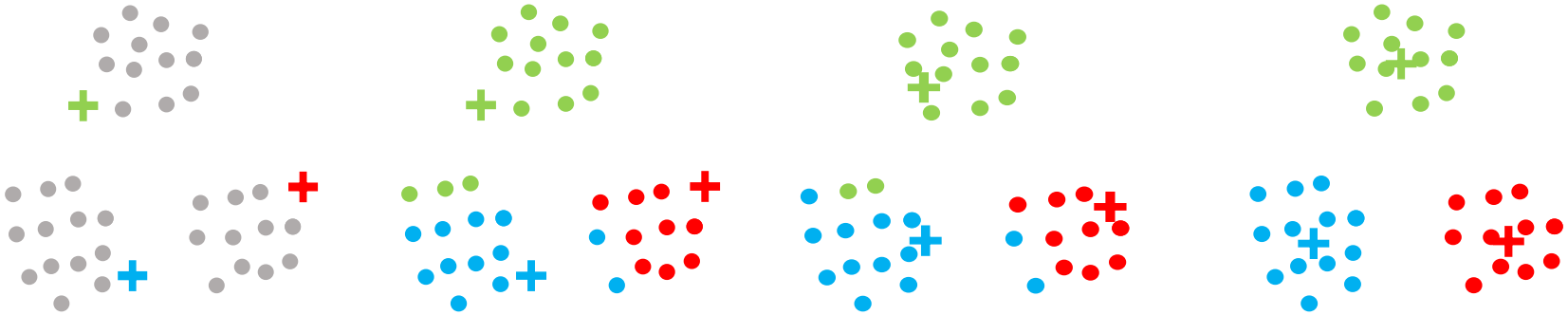
$$= -\sum_{i=1}^{N_j}(x_i - \mu_j)$$

- 上述步骤是迭代进行的，直到质心停止移动，即它们不再改变自己的位置，并且成为静态的。一旦这样做，K-均值算法被称为收敛。
- 算法收敛后，形成了清晰可见的不同簇。
- 该算法可以根据簇在第一步中的初始化方式给出不同的结果。



收敛结果

K-means算法流程总结



初始化质心　　簇赋值　　迭代更新　　收敛

- 优化迭代算法收敛后，形成了清晰可见的不同簇。显然，在每次迭代循环种，全样本SSE肯定下降(因为每次迭代后观测值分配到更近的聚类)。若SSE不再下降，则达到一个局部最小解。
- 但是K均值算法仅能找到局部最小值（可能不是全局最小值），在实际操作中，一般尝试索格不同初始聚类中心位置（设置不同随机数种子），然后选择最佳结果，最小化全样本SSE。

K值的选择

如何找到合适的簇的数量K？

- 一种方法是根据专业领域知识来选择，但比较主观。
- 另一种方法是尝试不同的聚类数目，考察全样本SSE的下降幅度，然后用“手肘法”。手肘法依然具有主观性，如何判断手肘的**拐弯**处争议较大。
- 较为客观的可以用信息准则：

  AIC信息准则（赤池信息量 akaike information criterion）

  BIC信息准则（贝叶斯信息量 bayesian information criterion）

- AIC信息准则

$$\min_k AIC(K) = SSE(K) + 2pK$$

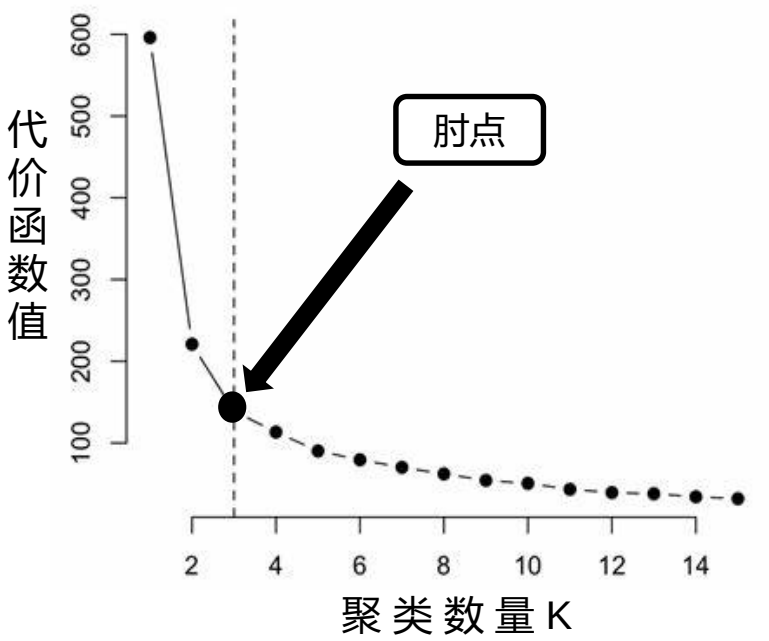
其中SSE(K)表示全样本误差平方和，是对模型模拟的奖励。但随着K 增大，尽管SSE下降，但容易出现过度拟合。因此需要加入惩罚因子$2pK$，其中p为C_k 的维度。

- BIC信息准则

$$\min_k BIC(K) = SSE(K) + \ln(n)pK$$

K值的选择--肘部法则

选择不同的K值，计算代价函数，可能会得到一条类似于人的肘部的曲线。如右图中，代价函数的值会迅速下降，在 $K = 3$ 时达到一个肘点，然后，代价函数就开始缓慢下降，所以，我们选择 $K = 3$。这个方法叫“肘部法则”。



- K-均值的一个问题在于，它有可能会停留在一个局部最小值处，而这取决于初始化的情况。
- 为了解决这个问题，通常需要多次运行K-均值算法，每一次都重新进行随机初始化，最后再比较多次运行K-均值的结果，选择代价函数最小的结果。

K-means的优点

- 原理比较简单，实现也是很容易，收敛速度快。
- 聚类效果较优。
- 算法的可解释度比较强。
- 主要需要调参的参数仅仅是簇数K。

K-means的缺点

- 需要预先指定簇的数量；
- 如果有两个高度重叠的数据，那么它就不能被区分，也不能判断有两个簇；
- 欧几里德距离可以不平等的权重因素，限制了能处理的数据变量的类型；
- 有时随机选择质心并不能带来理想的结果；
- 无法处理异常值和噪声数据；
- 不适用于非线性数据集；
- 对特征尺度敏感；
- 如果遇到非常大的数据集，那么计算机可能会崩溃。

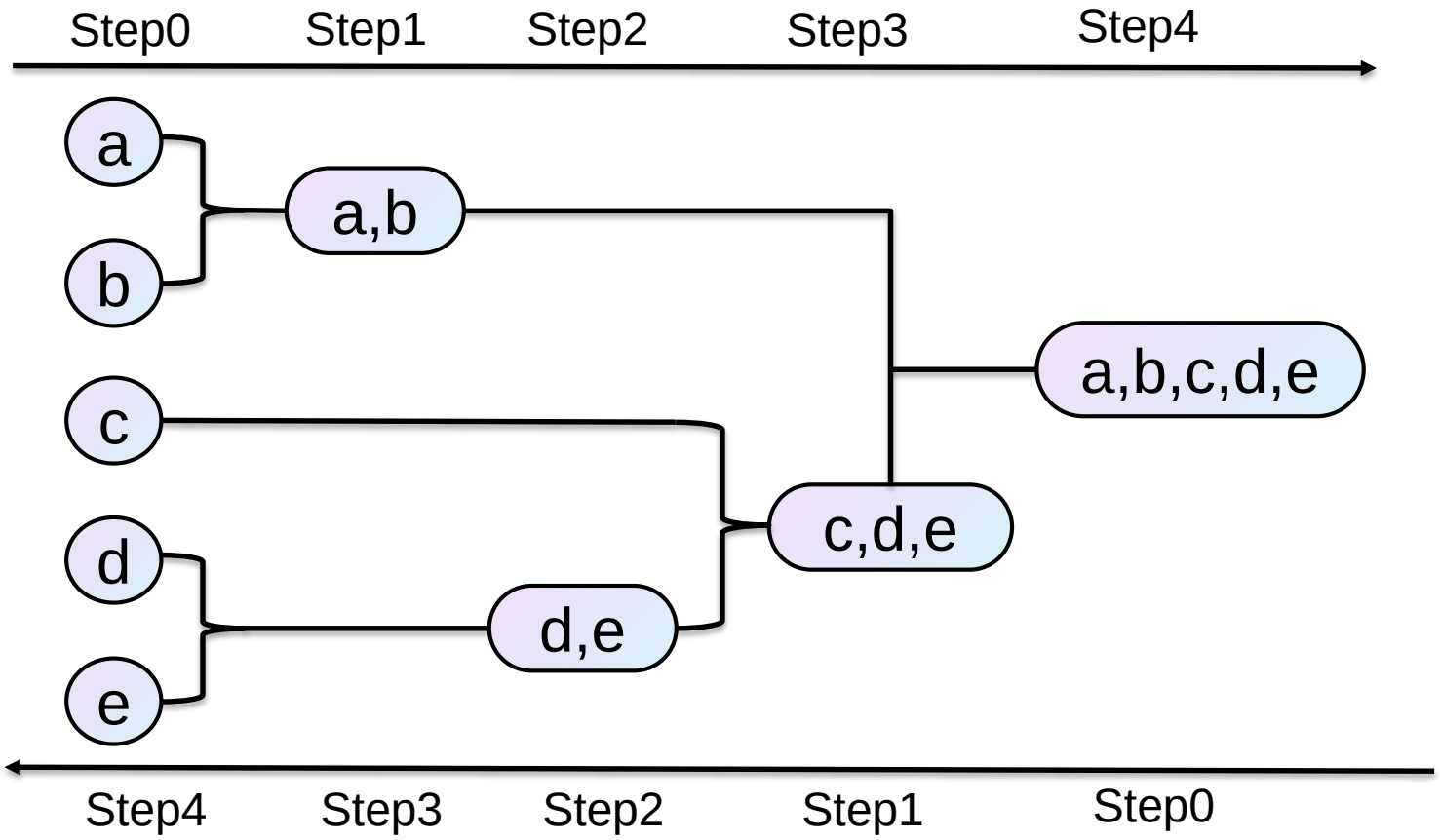
1. 层次聚类

- 层次聚类假设簇之间存在层次结构，将样本聚到层次化的簇中。
- 层次聚类又有聚合聚类（自下而上）、分裂聚类（自上而下）两种方法。
- 因为每个样本只属于一个簇，所以层次聚类属于硬聚类。

背景知识:

- 如果一个聚类方法假定一个样本只能属于一个簇，或簇的交集为空集，那么该方法称为硬聚类方法。
- 如果一个样本可以属于多个簇，或簇的交集不为空集，那么该方法称为软聚类方法。

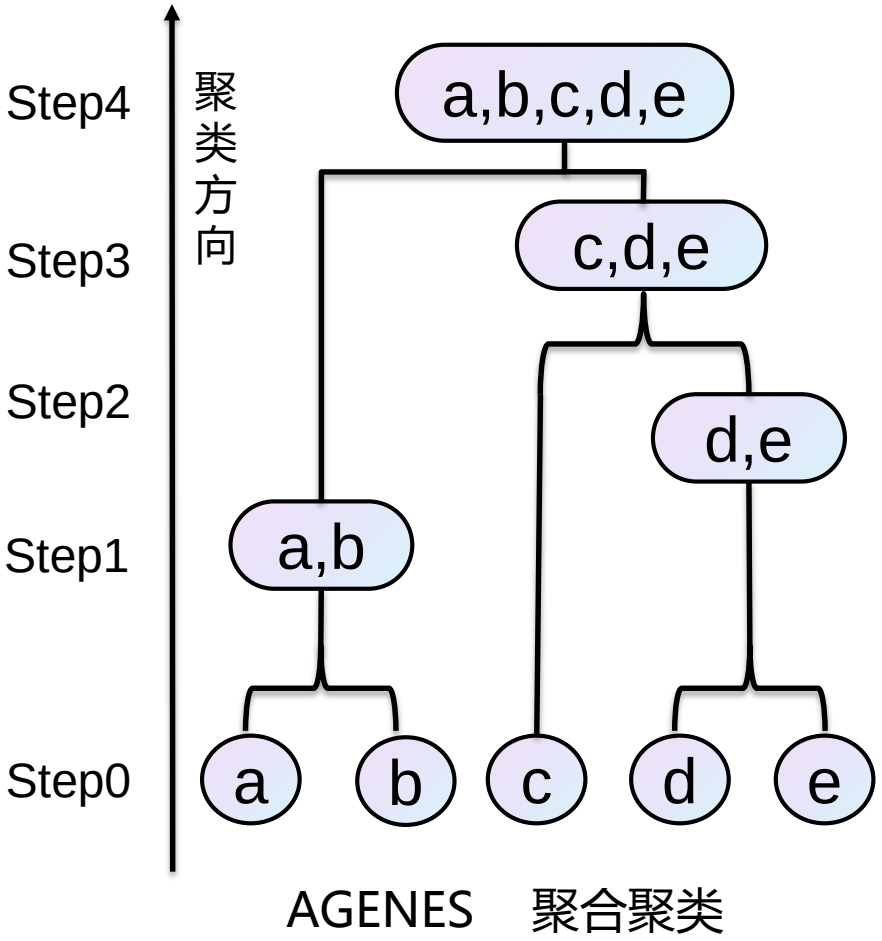
AGENES 聚合聚类



DIANA 分裂聚类

聚合聚类

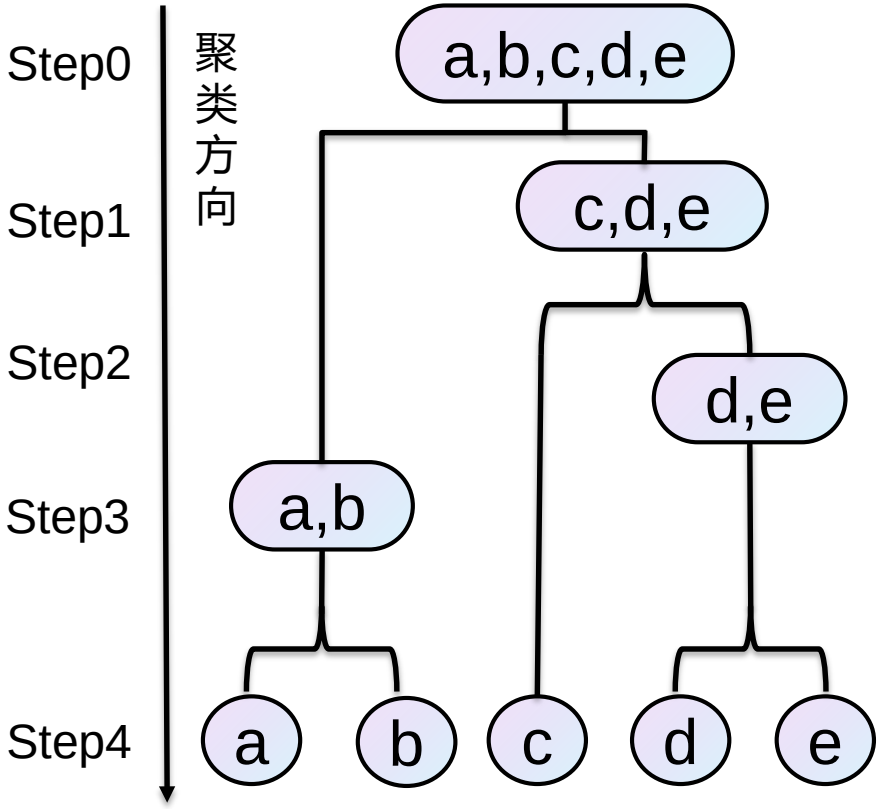
- 开始将每个样本各自分到一个簇；
- 之后将相距最近的两簇合并，建立一个新的簇；
- 重复此操作直到满足停止条件；
- 得到层次化的类别。



层次聚类-分裂聚类

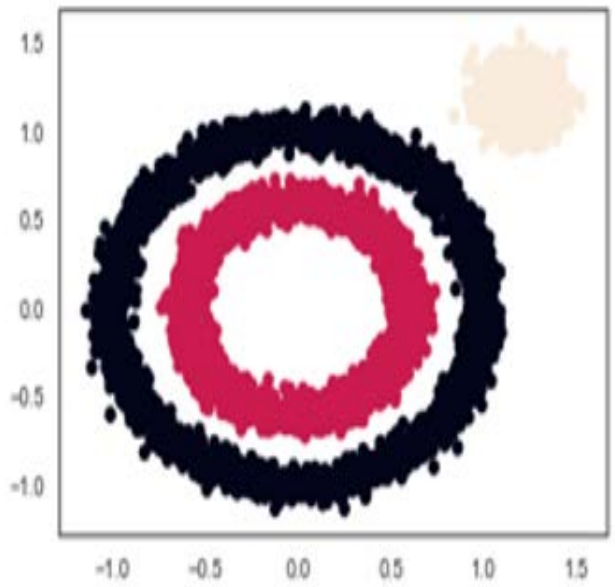
分裂聚类

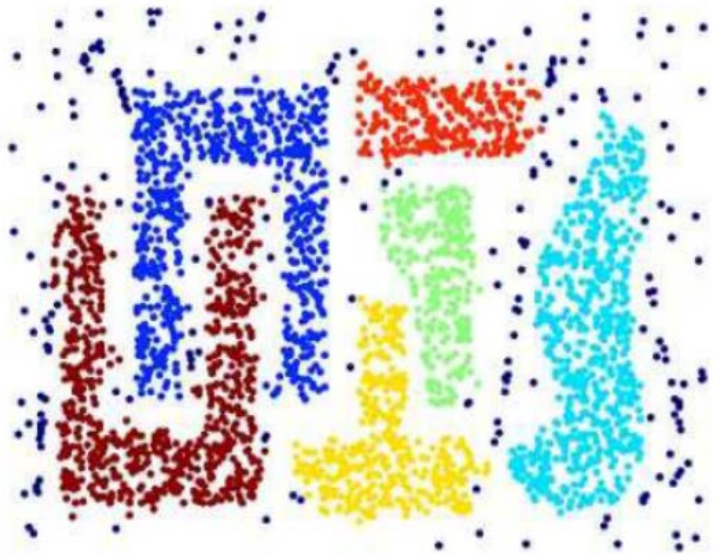
- 开始将所有样本分到一个簇；
- 之后将已有类中相距最远的样本分到两个新的簇；
- 重复此操作直到满足停止条件；
- 得到层次化的类别。



DIANA 分裂聚类

2. 密度聚类-DBSCAN





背景知识：如果 S 中任两点的连线内的点都在集合 S 内，那么集合 S 称为凸集。反之，为非凸集。

DBSCAN密度聚类

与划分和层次聚类方法不同，DBSCAN(Density-Based Spatial Clustering of Applications with Noise)是一个比较有代表性的基于密度的聚类算法。它将簇定义为密度相连的点的最大集合，能够把具有**足够高密度的区域划分为簇**，并可**在噪声的空间数据库中发现任意形状的聚类**。

密度：空间中任意一点的密度是以该点为圆心，以**扫描半径**构成的圆区域内包含的点数目。

DBSCAN使用**两个超参数：**扫描半径 (eps)和最小包含点数(minPts)来获得簇的数量，而不是猜测簇的数目。

- **扫描半径 (eps)：**

  用于定位点/检查任何点附近密度的距离度量，即扫描半径。

- **最小包含点数(minPts)：**

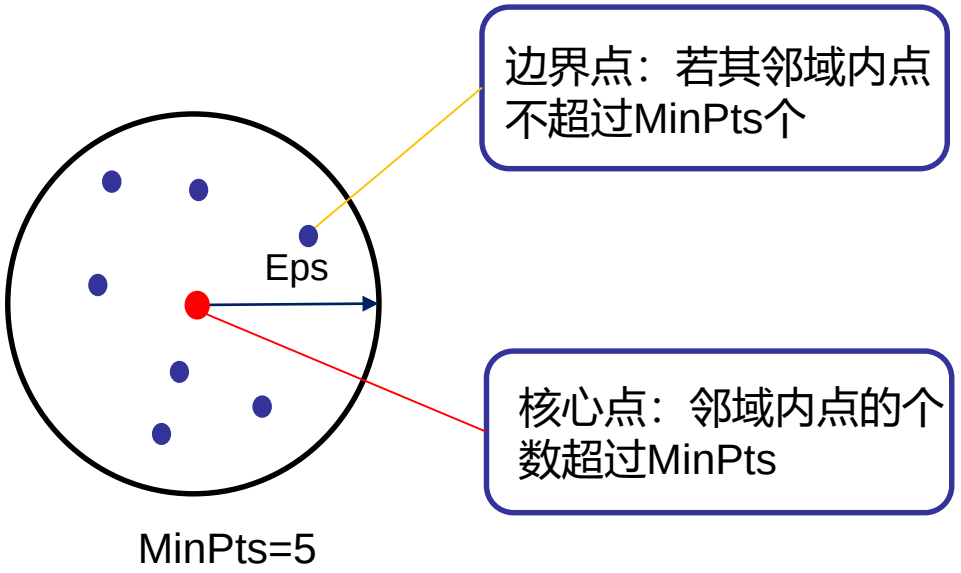
  聚集在一起的最小点数（阈值），该区域被认为是稠密的。

DBSCAN算法将数据点分为三类：

1. **核心点：**在半径Eps内含有超过MinPts数目的点。

2. **边界点：**在半径Eps内点的数量小于MinPts,但是落在核心点的邻域内的点。

3. **噪音点：**既不是核心点也不是边界点的点。



DBSCAN密度聚类的算法流程

1.将所有点标记为核心点、边界点或噪声点;

2. 如果选择的点是核心点，则找出所有从该点出发的密度可达对象形成簇;

3. 如果该点是非核心点，将其指派到一个与之关联的核心点的簇中;

4. 重复以上步骤，直到所点都被处理过

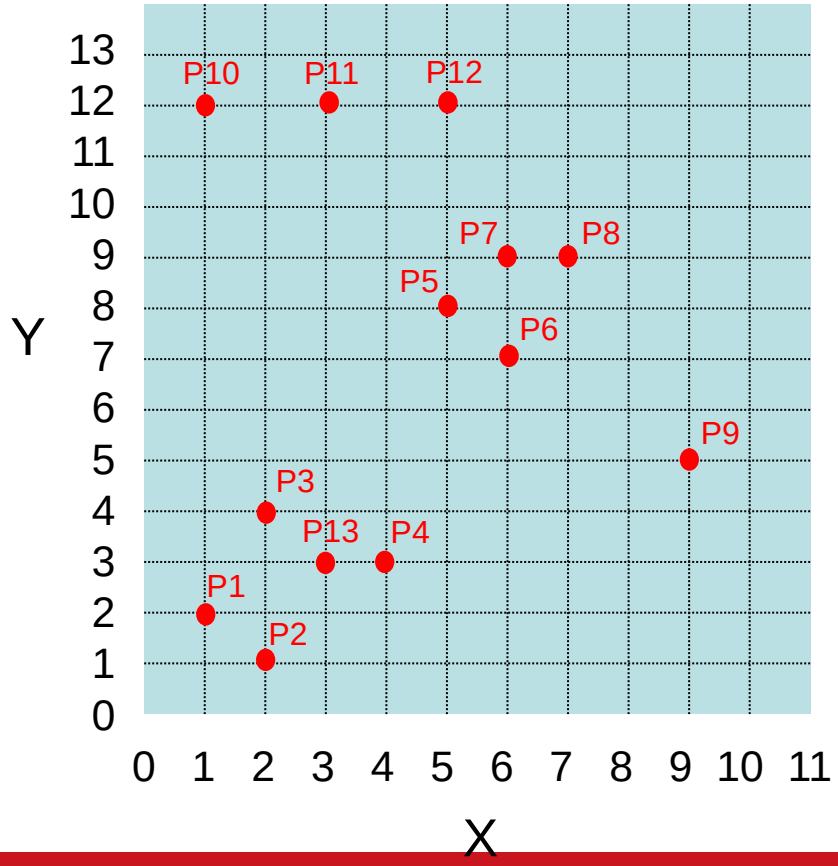
举例：有如下13个样本点，使用DBSCAN进行聚类

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13
X	1	2	2	4	5	6	6	7	9	1	3	5	3
Y	2	1	4	3	8	7	9	9	5	12	12	12	3

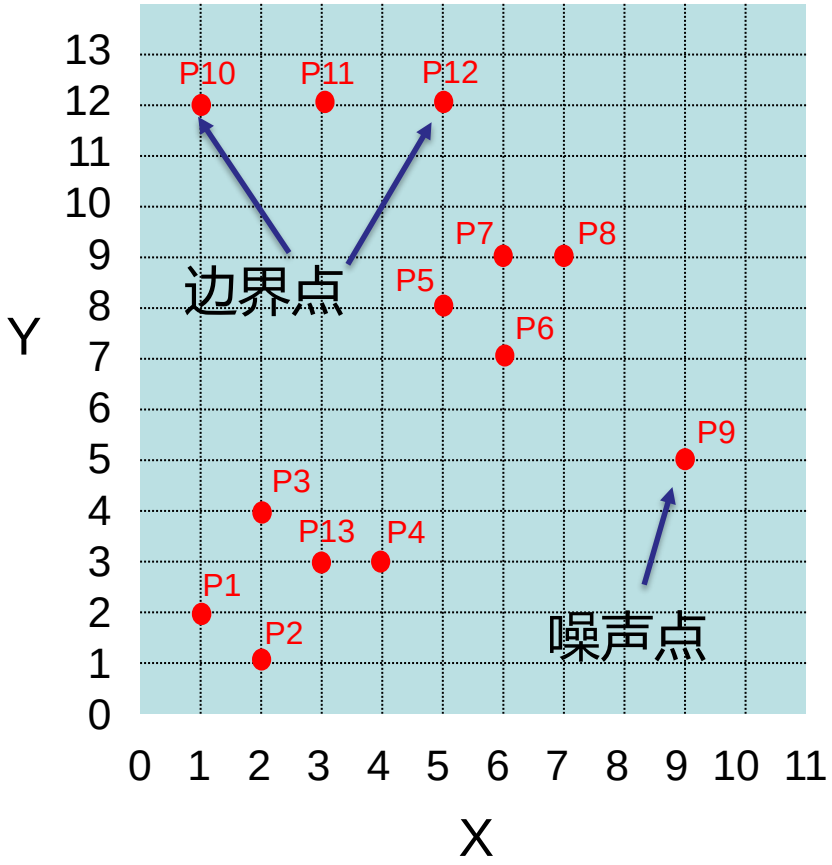
举例：有如下13个样本点，使用DBSCAN进行聚类

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13
X	1	2	2	4	5	6	6	7	9	1	3	5	3
Y	2	1	4	3	8	7	9	9	5	12	12	12	3

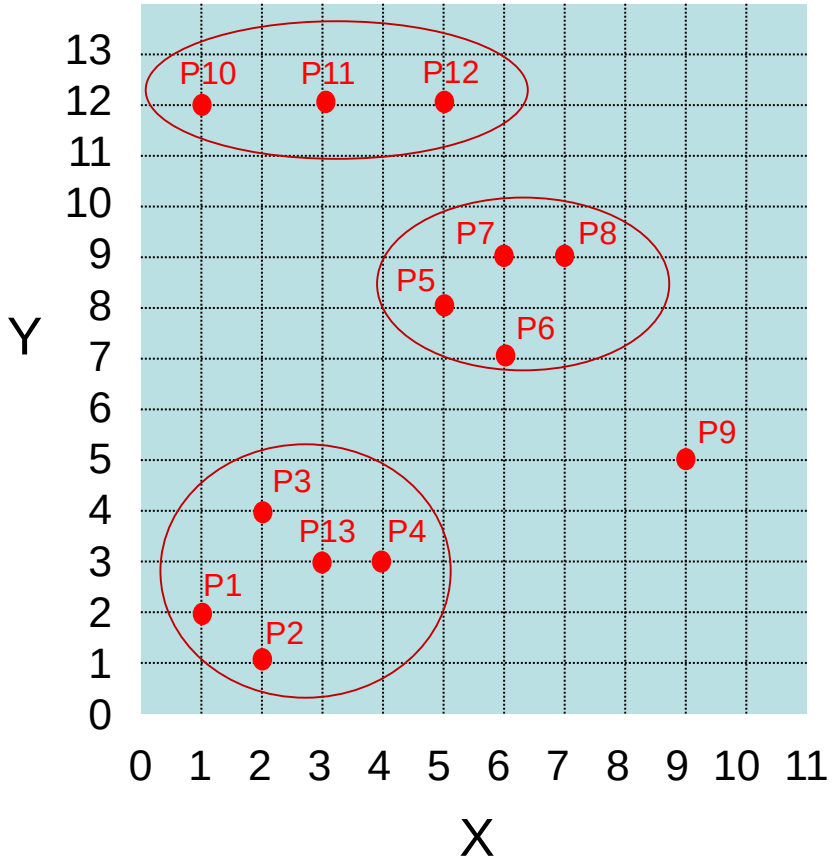
- 对每个点计算其邻域Eps=3内的点的集合。
- 集合内点的个数超过MinPts=3的点为核心点。



- 查看剩余点是否在核点的邻域内，若在，则为边界点，否则为噪声点。



- 将距离不超过Eps=3的点相互连接，构成一个簇，核心点邻域内的点也会被加入到这个簇中。



聚类的评价指标

1) **均一性p**：类似于精确率，一个簇中只包含一个类别的样本，则满足均一性。其实也可以认为就是正确率(每个聚簇中正确分类的样本数占该聚簇总样本数的比例和)

$$p = \frac{1}{k}\sum_{i=1}^{k}\frac{N(C_i == K_i))}{N(K_i)}$$

2) **完整性r**：类似于召回率，同类别样本被归类到相同簇中，则满足完整性；(每个聚簇中正确分类的样本数占该类型的总样本数比例的和)

$$r = \frac{1}{k}\sum_{i=1}^{k}\frac{N(C_i == K_i))}{N(C_i)}$$

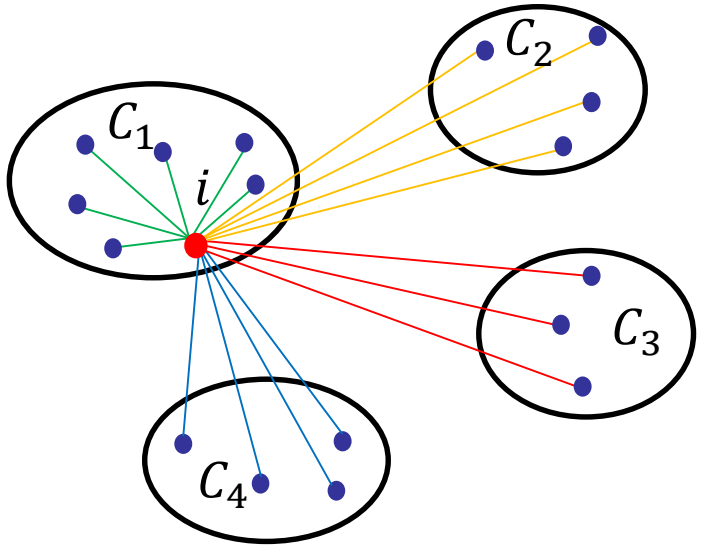
3) **V-measure**: 均一性和完整性的加权平均

$$V = \frac{(1+\beta^2) * pr}{\beta^2 * p + r}$$

4) 轮廓系数：样本i的轮廓系数

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}$$

- 簇内不相似度：计算样本i到同簇其它样本的平均距离为$a(i)$，应尽可能小。
- 簇间不相似度：计算样本i到其它簇C_j的所有样本的平均距离b_{ij}，应尽可能大。
- 轮廓系数$s(i)$值越接近1表示样本i聚类越合理，越接近-1，表示样本i应该分类到另外的簇中，近似为0，表示样本i应该在边界上；所有样本的$s(i)$的均值被成为聚类结果的轮廓系数。



假设数据集被拆分为4个簇，样本i对应的$a(i)$值就是所有C_1 中其他样本点与样本i的距离平均值；

样本对应的$b(i)$值分两步计算，首先计算该点分别到C_2、 C_3和C_4中样本点的平均距离，然后将三个平均值中的最小值作为$b(i)$的度量.

5)调整兰德系数 (ARI：Adjusted Rand Index)

设数据集S共有N个元素，两个聚类结果分别是：

$$X = \{X_1, X_2, \ldots, X_r\}, Y = \{Y_1, Y_2, \ldots, Y_s\}$$

且X和Y的元素个数为：

$$a = \{a_1, a_2, \ldots, a_r\}, b = \{b_1, b_2, \ldots, b_s\}$$

C	Y_1	Y_2	$\cdots$	Y_s	sum
X_1	n_{11}	n_{12}	$\cdots$	n_{1s}	a_1
X_2	n_{21}	n_{22}	$\cdots$	n_{2s}	a_2
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
X_r	n_{r1}	n_{r2}	$\cdots$	n_{rs}	a_r
sum	b_1	b_2	$\cdots$	b_s	N

ARI取值范围为$[-1,1]$，值越大意味着聚类结果与真实情况越吻合。从广义的角度来讲，ARI衡量的是两个数据分布的吻合程度

令 $n_{ij} = |X_i \cap Y_i|$

$$ARI = \frac{\sum_{i,j} C_{n_{ij}}^2 - \left[\left(\sum_i C_{a_i}^2\right) \cdot \left(\sum_i C_{b_i}^2\right)\right] / C_n^2}{\frac{1}{2}\left[\left(\sum_i C_{a_i}^2\right) + \left(\sum_i C_{b_i}^2\right)\right] - \left[\left(\sum_i C_{a_i}^2\right) \cdot \left(\sum_i C_{b_i}^2\right)\right] / C_n^2}$$

$$ARI = \frac{\sum_{i,j} C_{n_{ij}}^2 - \left[\left(\sum_i C_{a_i}^2\right) \cdot \left(\sum_i C_{b_i}^2\right)\right] / C_n^2}{\frac{1}{2}\left[\left(\sum_i C_{a_i}^2\right) + \left(\sum_i C_{b_i}^2\right)\right] - \left[\left(\sum_i C_{a_i}^2\right) \cdot \left(\sum_i C_{b_i}^2\right)\right] / C_n^2}$$

- 优点：

  1）对任意数量的聚类中心和样本数，随机聚类的ARI都非常接近于0；

  2）取值在［－1，1］之间，负数代表结果不好，越接近于1越好；

  3）可用于聚类算法之间的比较

- 缺点：

  ARI需要真实标签

KNN和K-Means的对比

KNN	K-Means
1.KNN是分类算法 2.监督学习 3.喂给它的数据集是带label的数据，已经是完全正确的数据	1.K-Means是聚类算法 2.非监督学习 3.喂给它的数据集是无label的数据，是杂乱无章的，经过聚类后才变得有点顺序，先无序，后有序
没有明显的前期训练过程，属于memory-based learning	有明显的前期训练过程
K的含义：输入一个样本x，要给它分类，即求出它的标签y，就从数据集中，在x附近找离它最近的K个数据点，这K个数据点，类别c占的个数最多，就把x的label设为c	K的含义：K是人工固定好的数字，假设数据集合可以分为K个簇，由于是依靠人工定好，需要一点先验知识
相似点：都包含这样的过程，给定一个点，在数据集中找离它最近的点。即二者都用到了NN(Nears Neighbor)算法，一般用KD树来实现NN。 文中参考：http://blog.csdn.net/chlele0105/article/details/12997391	

参考文献：10种聚类算法的完整python操作示例
https://mp.weixin.qq.com/s/jtcDiAV_MC1_OGeR-8YVOg

谢 谢!

本课件制作过程中，多处引用了国内外同行的网页、教材、以及课件PPT的内容或图片，没有随处标注，特此说明，并在此向各位作者表示感谢！