**聚类**

**目的**: 用于寻找数据内在的分布结构，也可作为分类等其它学习任务的先驱过程。

**概念**: 将数据中的样本划分为若干个通常是不相交的子集, 每个子集称为一个“簇”。每个簇可能对应于一些潜在的概念(类别), 这类概念对聚类算法而言事先是未知的，聚类过程仅能自动形成簇结构，簇所对应的概念语义需由使用者来把握和命名。

**目标**: “簇内相似度”高且“簇间相似度”低.

**性能度量**: 亦称聚类“ 有效性指标”,用来评估结果好坏,同时可以直接作为聚类过程的优化指标.

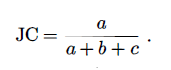
**1).外部指标**: 聚类结果与某个参考模型进行比较

们将样本两两配对考虑，定义：

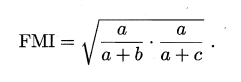
a=|SS|(聚类同簇，参考模型同簇)   
　　 b=|SD|(聚类同簇，参考模型不同簇)   
　　 c=|DS|(聚类不同簇，参考模型同簇)   
　　 d=|DD|（聚类不同簇，参考模型不同簇)

集合|SS|包含了在聚类结果中隶属于相同簇且在参考模型结果中也隶属于相同簇的样本对，集合|SD|包含了在聚类结果中隶属于相同簇但在参考模型结果中隶属于不同簇的样本  
 下面是一些常用的聚类性能度量外部指标：

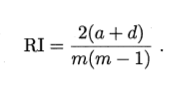
Jaccard系数(Jaccard Coefficient，简称JC) :



FMI指数(Fowlkes and Mallows Index，简称FMI)

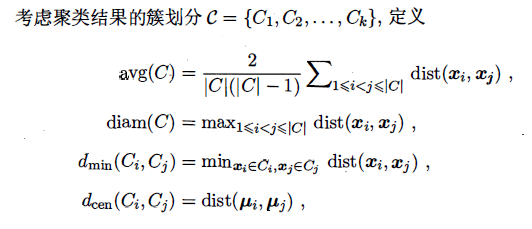


Rand指数(Rand Index，简称RI)



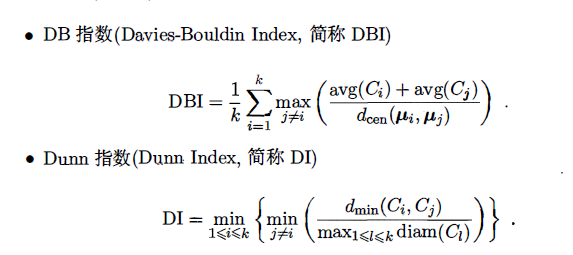
　　显然上述性能度量的结果值均在[0, 1]之间，值越大说明结果越好。

**2).内部指标**: 直接考察聚类结果而不利用任何参考模型.



dist(x,x)用于计算两个样本之间的距离；   
 u代表簇C的中心点；   
 avg(C)对应于簇C内样本间的平均距离；   
 diam(C)对应于簇C内样本的最远距离；   
 dmin(Ci,Cj)对应于簇Ci和Cj最近样本间的距离；   
 dcen(Ci,Cj)对应于簇Ci和Cj中心点之间的距离.

聚类性能度量内部指标：



显然， DBI的值越小越好，而且则相反，值越大越好．

思考：dist(x,x)用于计算两个样本之间的距离，但是，数据集并不都是可度量的啊，怎么破？

**距离计算**：

对距离度量函数dist（x,x），具备一些性质：

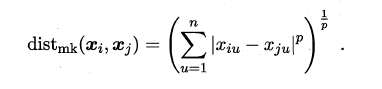
* 非负性：dist(xi,xj) >=0；
* 同一性：dist(xi,xj) =0当且仅当xi=xj；
* 对称性：dist(xi,xj) =dist(xj,xi)；
* 直递性：dist(xi,xj) <=dist(xi,xk) +dist(xk,xj)

（其中k为任一点，又称“三角不等式”）

**属性上是否定义了“序”关系：**如定义了序的{1，2，3}中1和2比较接近，与3比较远，而没有定义序的 {飞机，火车，轮船} 则不能直接在属性值上计算距离。

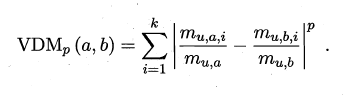
**1，有序属性的距离运算**：

常用的有：“闵可夫斯基距离”、“欧氏距离”（p=2）、“曼哈顿距离”（p=1）等，其中最常用的是“**闵可夫斯基距离**”：



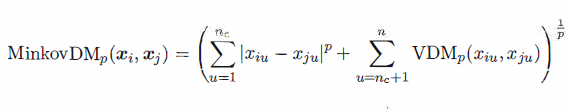
**2，无序属性**：

对于无序属性可采用VDM (Value Difference Matrix)。令mu,a表示在属性u上取值为a的样本数，mu,a,i表示在第i个样本簇中在属性u上取值为a的样本数，k为样本簇数，则属性u上两个离散值a与b之间的VDM距离为：

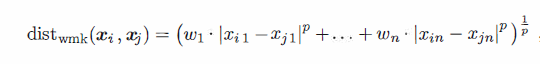


**3，混合属性**：

将闵可夫斯基距离和VDM结合即可处理混合属性．假定有nc 个有 序属性、n-nc个无序属性, 令有序属性排列在无序属性之前:



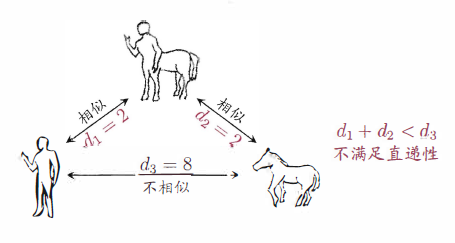
**4，属性加权**：



**5，非度量距离**：（人为距离）

有可能违背一些基本性质，尤其是直递性。

“ 人”“马”分别与“ 人马”相似，但 “人”与“ 马 ”很不相似；要达到这个目的，可以令“ 人”“马”与“人马 ” 之间的距离都比较小 但“人”与“ 马”之间的距离很大



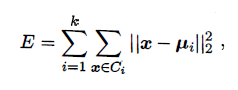
引：已经说了目标，看看有那些方法来实现聚类。

**原型聚类**：基于原型的聚类， 算法假设聚类结构能通过一组原型刻画，在现实聚类任务中极为常用。先对原型进行初始化，然后对原型进行迭代更新求解．采用不同的原型表示、不同的求解方式，将产生不同的算法.

**1，K均值算法（K-means）：**

样本集D = {x1，x2，...,xm｝，k均值 （k-means）算法针对聚类所得簇划分C={C1,C2, . ,Ck｝

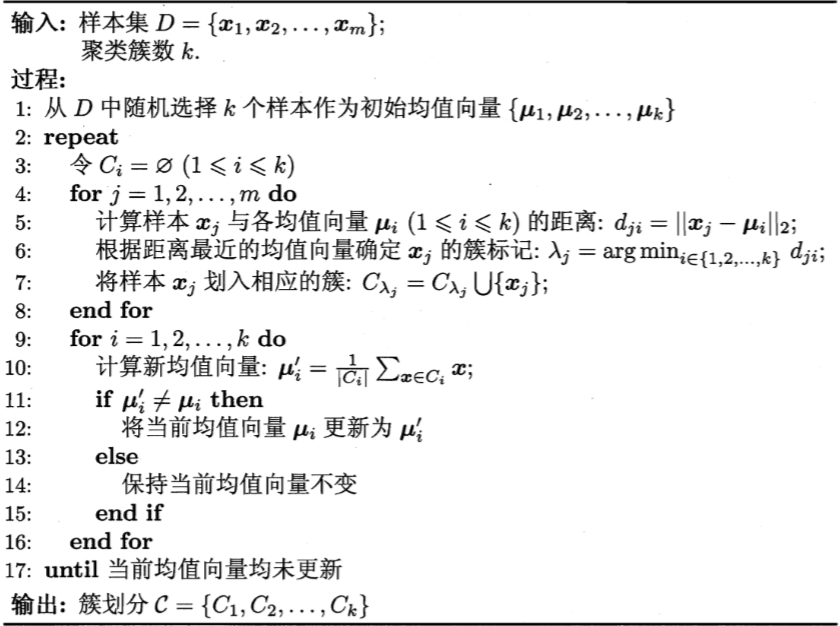
最小化平方误差：

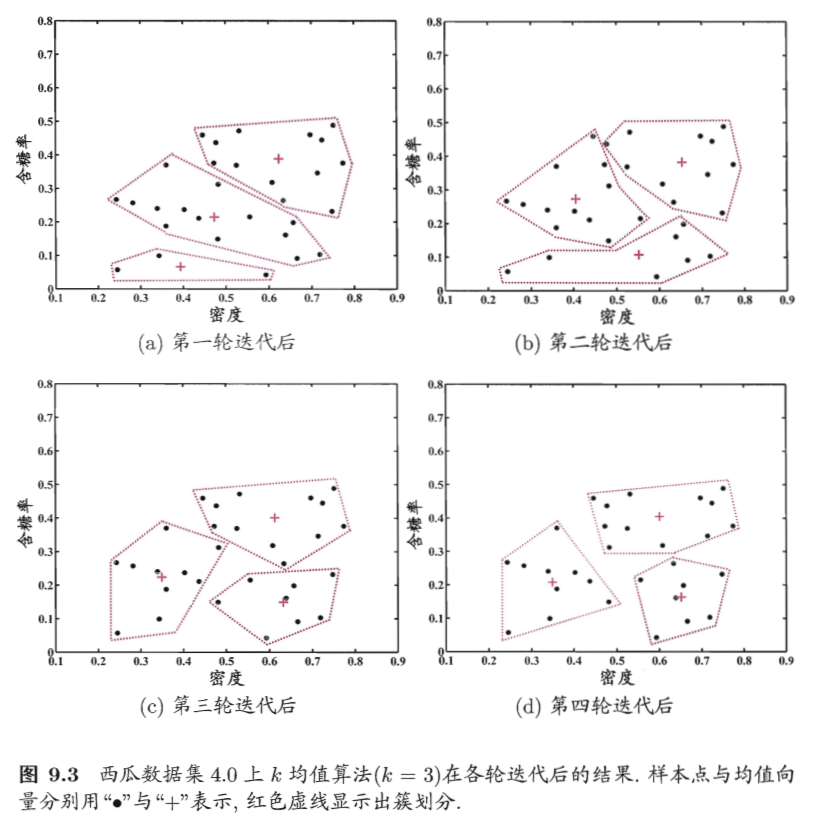
 

下标2表示L2范数，根号下的平方和。右上角2表示去掉根号，则是平方和。刻画了簇内样本围绕簇均值向量的紧密程度，E值越小则簇内样本相似度越高，则效果越好。

具体实现过程：

最小化E是个NP-hard问题（NP是指非确定性多项式。非确定性是指，可用一定数量的运算解决的问题。e.g.: 推销员旅行问题: 假设一个推销员需要从香港出发，经过广州，北京，上海，…，等 n 个城市， 最后返回香港。 任意两个[城市 之间](https://baike.baidu.com/item/%E5%9F%8E%E5%B8%82%E4%B9%8B%E9%97%B4)都有飞机直达，但票价不等。假设公司只给报销 C 元钱，问是否存在一个行程安排，使得他能遍历所有城市，而且总的路费小于 C？因为如果你任意给出一个行程安排，可以很容易算出旅行总开销。但是，要想知道一条总路费小于 C 的行程是否存在，在最坏情况下，必须检查所有可能的旅行安排！）。K均值算法采用贪心策略（一种分级处理办法：做出当前看来最好的决定），迭代优化来近似求解。

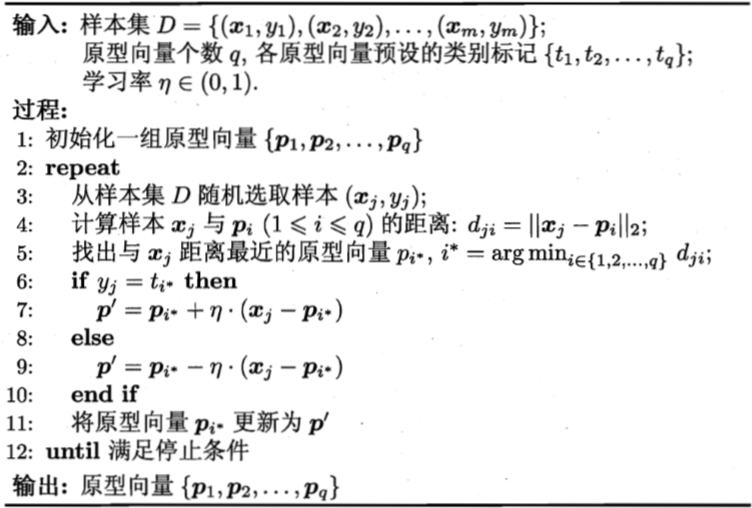




k是靠人工选择的结果，大家可以通过对不同的 k 值进行试验，或通过对数据集的理解来找到最合适的 k 的取值。

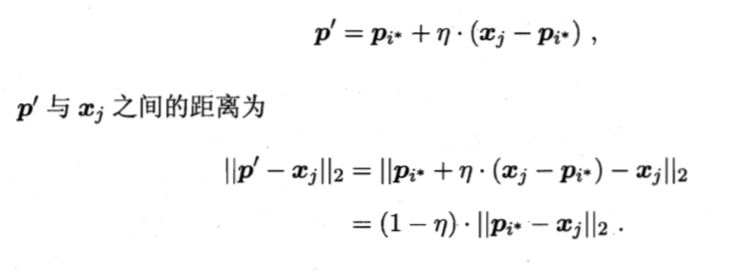
**2，学习向量量化（LVQ）：**

LVQ也是试图找到一组原型向量（k均值算法中的簇均值向量集合就是一组原型向量）来刻画聚类结构，但是LVQ假设数据样本带有类别标记。学习时利用样本的这些监督信息来辅助聚类。



arg的理解：表示使目标函数取最小值时的变量值

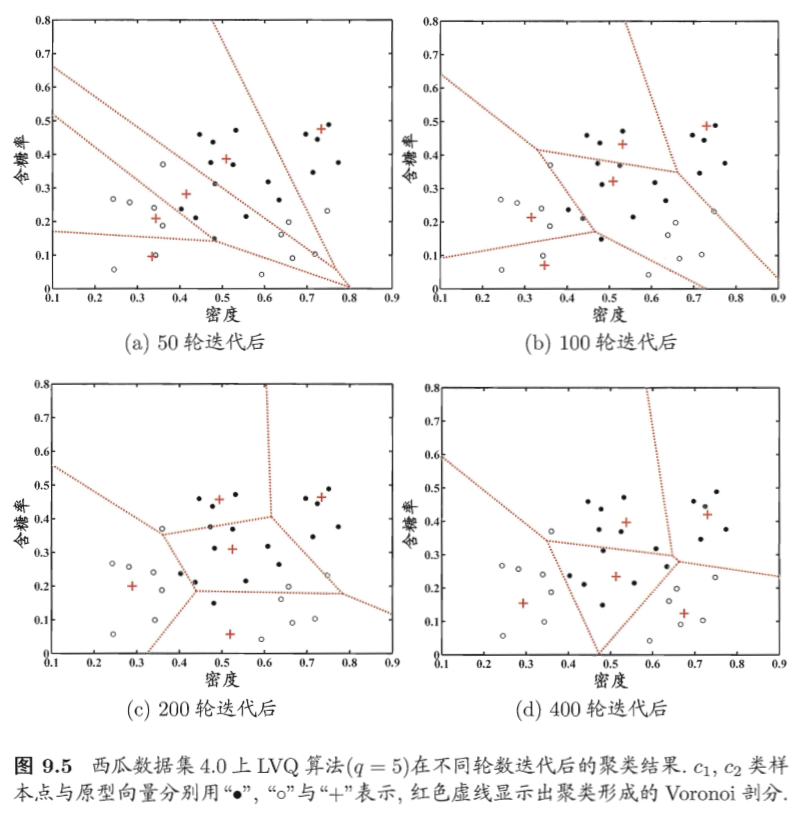
理解第六行到第九行：（距离的增大与减小）



得到了一组原型向量，依此实现对样本空间X的簇划分：（称为Voronoi剖分）

对任意样本x，它将被划入与其距离最近的原型向量所代表的簇中。

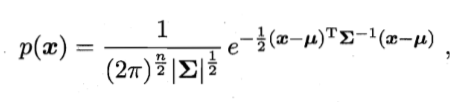
停止条件：已达到最大迭代轮数或原型向量更新很小或不再更新。



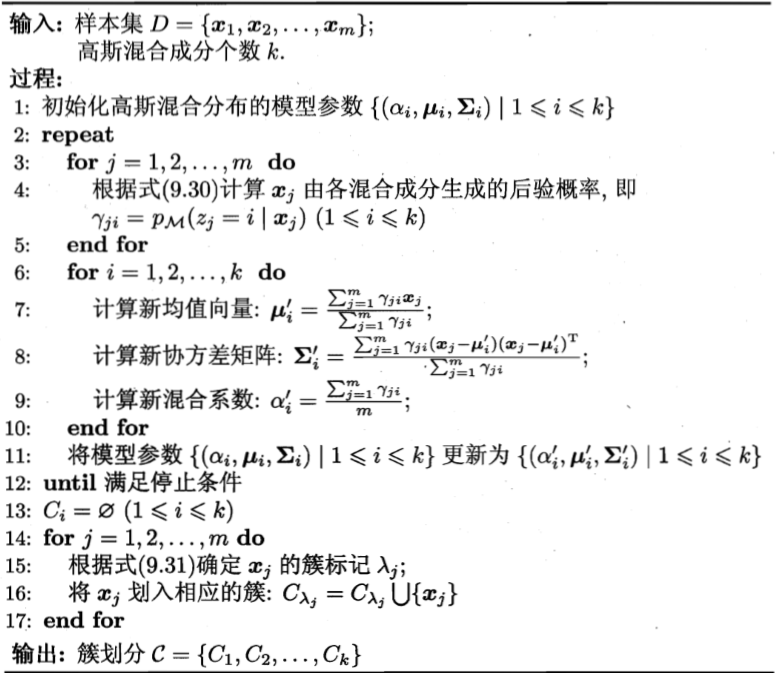
**3，高斯混合聚类：**

与k均值、LVQ用原型向量来刻画聚类结构不同，高斯混合聚类采用概率模型来表达聚类原型。相对于k均值聚类算法使用 k 个原型向量来表达聚类结构，高斯混合聚类使用 k 个高斯概率密度函数混合来表达聚类结构：每个高斯概率密度函数代表一个簇。

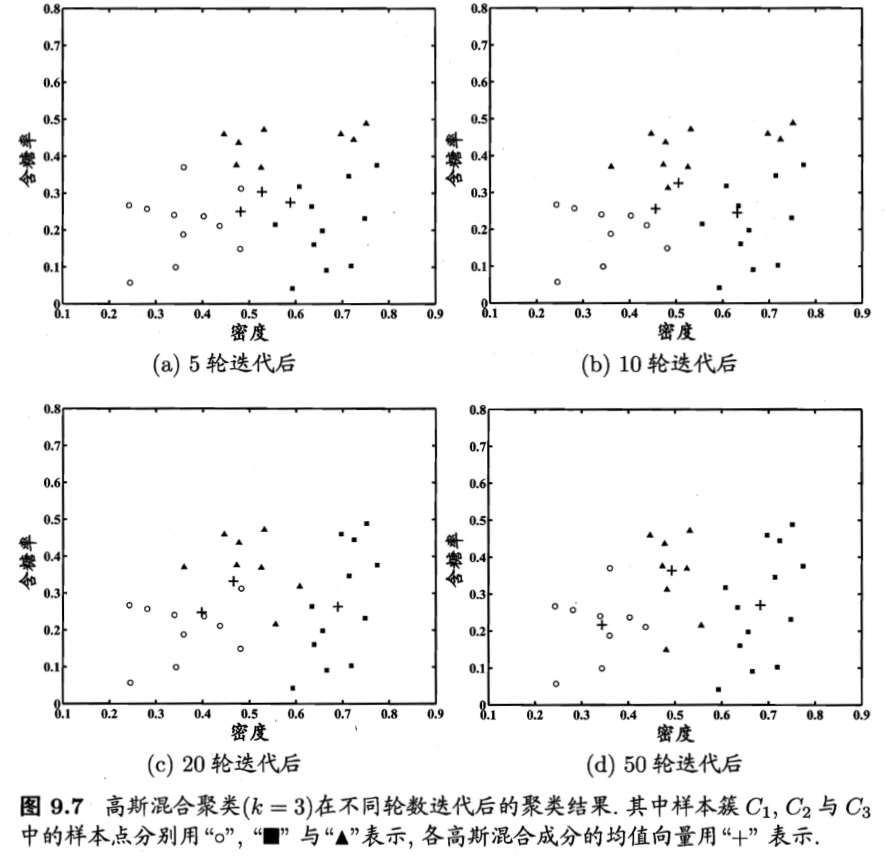
多维高斯分布：



表示n\*n的协方差矩阵



搞不懂



**3.密度聚类**：

基于密度的聚类，通过样本分布的紧密程度确定。密度聚类算法从样本密度的角度来考察样本之间的可连接性，并基于可连接样本不断扩展聚类簇以获得最终的聚类结果。

密度聚类算法：DBSCAN

它基于一组“邻域”参数来刻画样本分布的紧密程度。中心思想是将满足一定可达关系能导出的最大密度相连样本定义为一个簇，其通过两个基本参数 e 和 MinPts 来定义这个可达关系。

基本概念：

［1］“e - 邻域”：e为邻域半径，任意一个样本 xj 的 “e - 邻域” 包含样本集 D 中与样本 xj 的距离不大于 e 的样本集合；

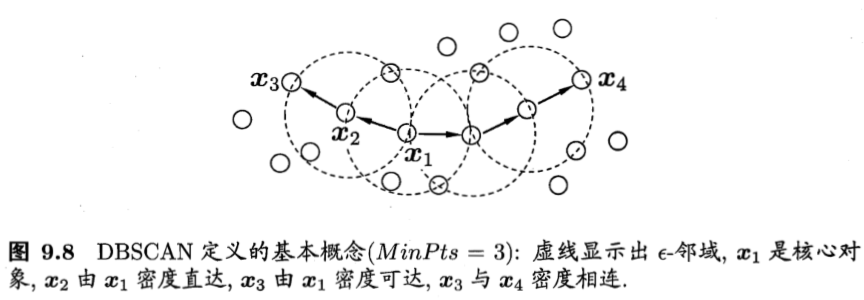
［2］核心对象：当一个样本 x 的 “e - 邻域”内含有至少 MinPts 个样本时，该样本 x 是一个核心对象；

［3］密度直达：若 xj 位于 xi 的 “e - 邻域”中，且 xi 是核心对象，则称 xj 与 xi 密度直达

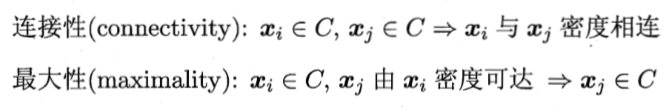
［4］密度可达：若 xi 与 xj 能通过一系列密度直达的点关联起来，则 xi 与 xj 密度可达

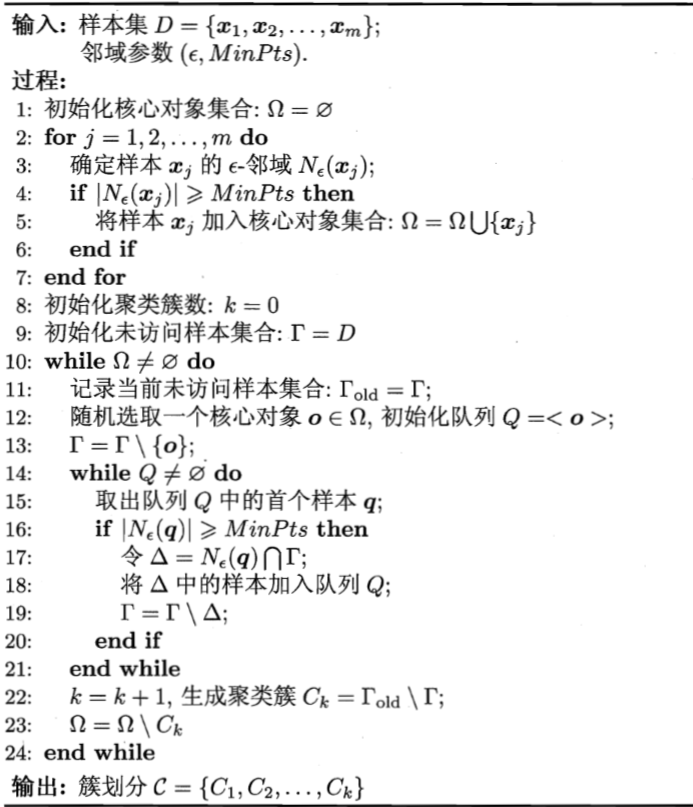
［5］密度相连：若 xi 与 xj 都能通过 xk 密度可达，则称 xi 与 xj 密度相连

图示：



簇的定义：由密度可达关系导出的最大的密度相连样本集合。满足以下2个性质的集合即为一个簇。（由X（核心对象）元素密度可达的样本组成）



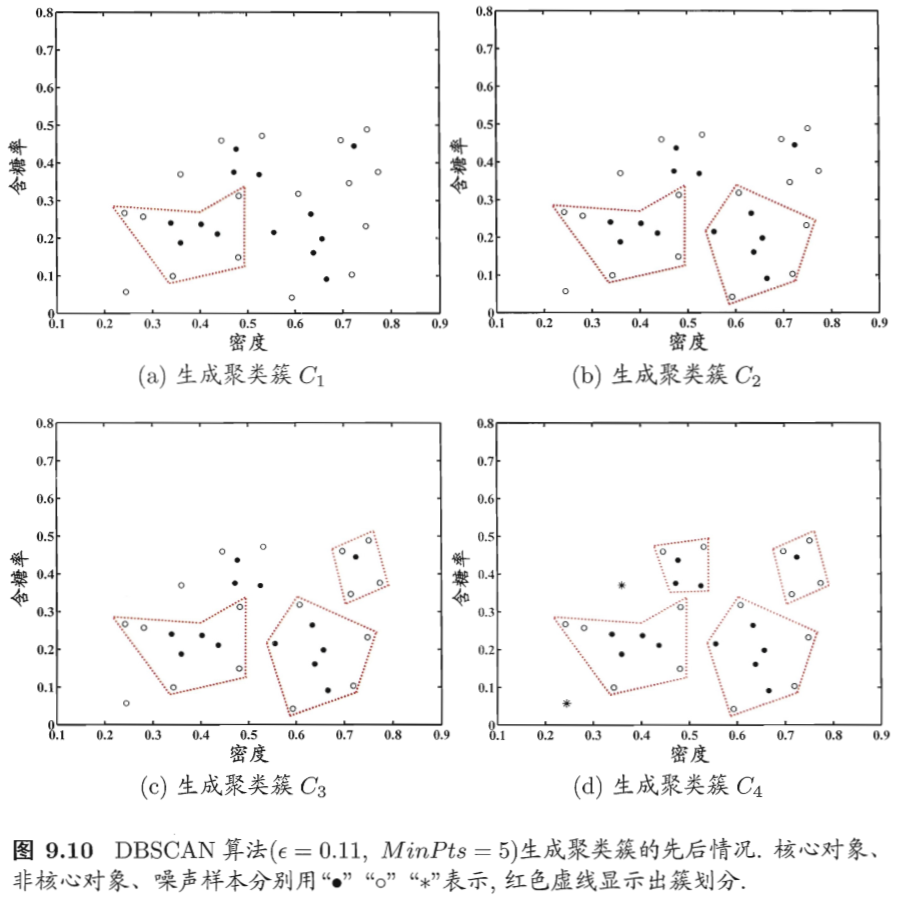
算法：根据给定的领域参数找出所有核心对象，然后以每个核心对象为出发点，找出由其密度可达的样本生成聚类簇。

Q：核心对象O的邻域。

a\b ：从a中踢出b。

第14-20行：从总的样本集中去掉与Q内所有元素密度可达的元素。

16 17 18 不懂



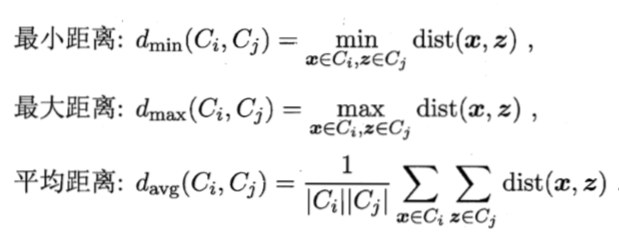
可能会存在一些数据是不属于任何簇的，这些样本被认为是噪声或异常样本。在个别应用场景下，这部分样本将具有较高的价值，比如在反欺诈场景下，这部分不合群的样本数据是欺诈行为数据的概率较高。

**层次聚类**：

层次聚类试图在不同层次对数据集进行划分，从而形成树形的聚类结构。数据集的划分可采用“自底向上”的聚合策略，也可采用“自顶向下”的分拆策略。

自底向上：AGNES（AGglomerative NESting）是一种采用自底向上聚合策略的层次聚类算法，它的算法十分简单，那就是先将每个样本看作一个初始聚类簇，然后在算法运行的每一步找出距离最近的两个聚类簇进行合并，该过程不断重复，直到达到预设的聚类簇个数。

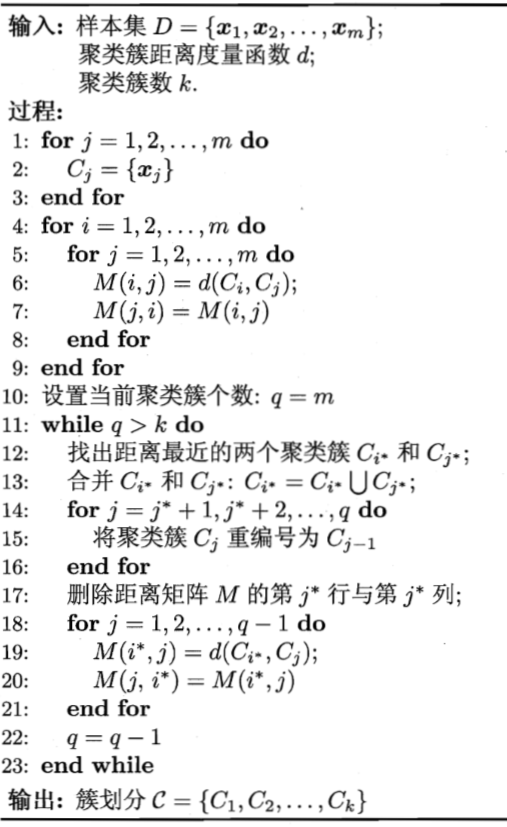
距离怎么算？

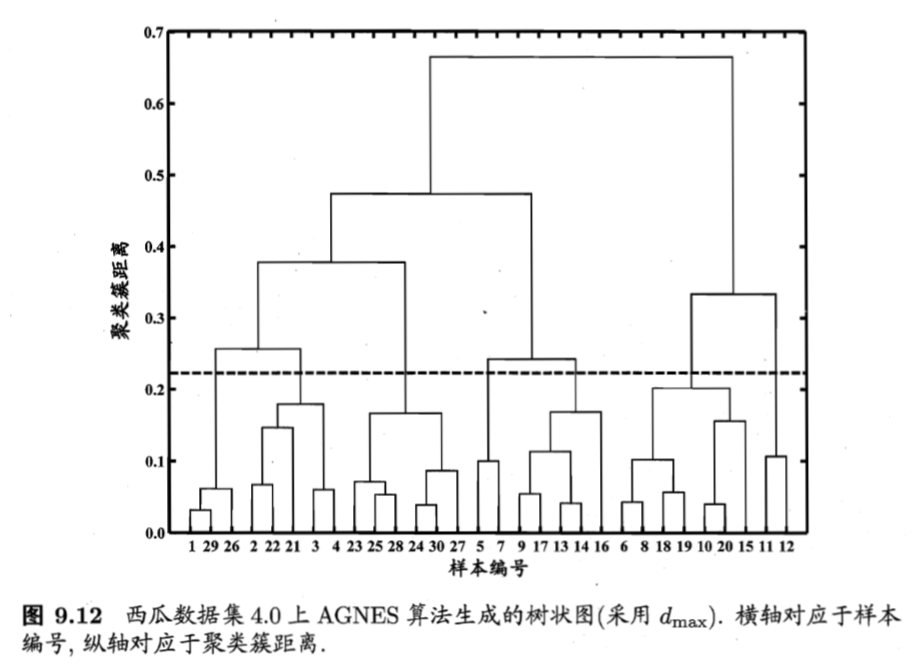


［1］最小距离：两个聚类簇的样本之间的最小距离，使用该距离定义的算法被称为“单链接”算法

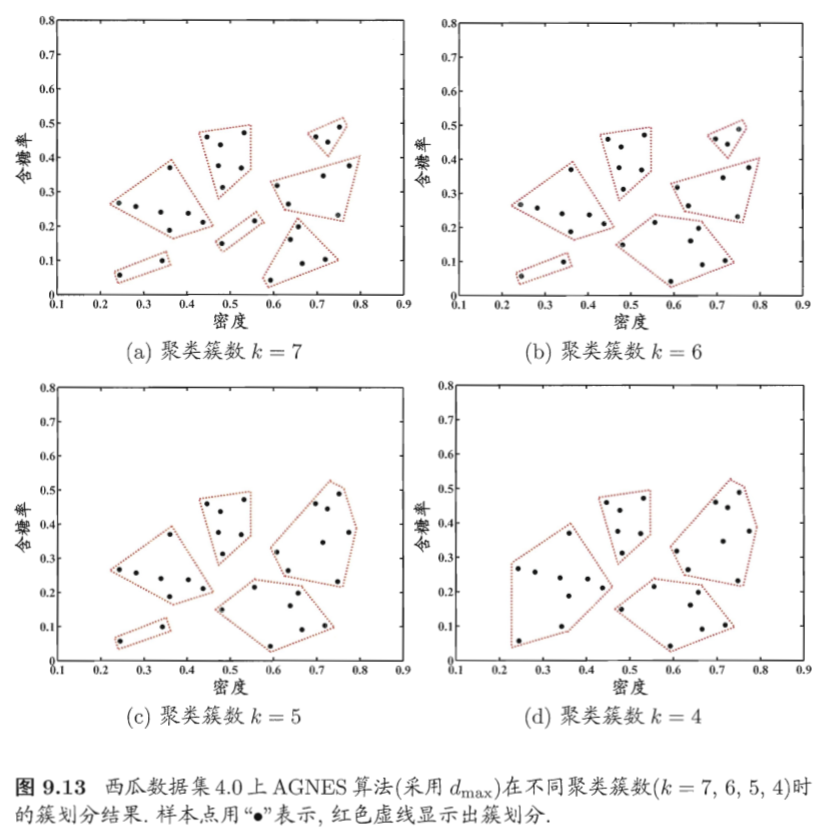
［2］最大距离：两个聚类簇的样本之间的最大距离，使用该距离定义的算法被称为“全链接”算法

［3］平均距离：两个聚类簇的样本之间的距离之和的平均，使用该距离定义的算法被称为“均链接”算法





在树状图不同的层次上进行切割，则可得到相应的簇划分结果。



基础知识

1）无监督学习（unsupervised learning）

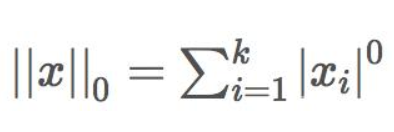
训练样本的标记信息是未知的，即没有正确答案，目标是通过对无标记训练样本的学习来揭示数据的内在性质和规律，为进一步的数据分析提供基础。

2）监督学习（supervised learning）

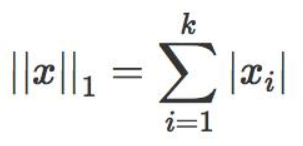
顾名思义，监督学习与无监督学习相反，即在标记好的训练样本上进行训练和学习的机器学习算法

3）范数：L0范数、L1范数、L2范数

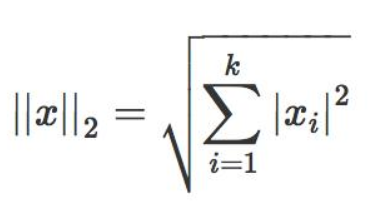
L0范数： （非0元素个数）



L1范数： （向量中各个元素绝对值之和）



L2范数： （欧氏距离）



（具体可看https://baijiahao.baidu.com/s?id=1607333156323286278&wfr=spider&for=pc）