### **聚类的定义**

聚类(Clustering)是按照某个特定标准(如距离)把一个[数据集分割](https://so.csdn.net/so/search?q=%E6%95%B0%E6%8D%AE%E9%9B%86%E5%88%86%E5%89%B2&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/pearl8899/article/details/_blank)成不同的类或簇，使得****同一个簇内的数据对象的相似性尽可能大，同时不在同一个簇中的数据对象的差异性也尽可能地大****。**组内相似性越大，组间差距越大，说明聚类效果越好。**

**聚类和分类的区别**

**聚类(Clustering)**：是指把相似的数据划分到一起，具体划分的时候并不关心这一类的标签，目标就是把相似的数据聚合到一起，聚类是一种**无监督**学习(Unsupervised Learning)方法。

**分类(Classification)**：是把不同的数据划分开，其过程是通过训练数据集获得一个分类器，再通过分类器去预测未知数据，分类是一种**监督**学习(Supervised Learning)方法。

聚类得到的簇可以用**聚类中心、簇大小、簇密度**和**簇描述**等来表示

聚类中心是一个簇中所有样本点的均值(质心)

簇大小表示簇中所含样本的数量

簇密度表示簇中样本点的紧密程度

簇描述是簇中样本的业务特征

/\*业务特征通常是指与具体业务场景相关联的属性或变量，这些特征能够反映数据在实际业务中的性质、状态或行为

假设我们对一家银行的客户进行聚类分析，得到以下结果：

簇1：客户年龄在20-30岁之间，平均存款余额较低，但信用卡消费频繁。

簇2：客户年龄在40-50岁之间，平均存款余额较高，信用卡消费较少，但定期投资理财产品。

在簇描述中，业务特征可以是：

簇1：年轻消费活跃群体（业务特征）。

簇2：中年理财偏好群体（业务特征）。\*/

**聚类分析的要求**

不同的聚类算法有不同的应用背景，有的适合于大数据集，可以发现任意形状的聚簇；有的算法思想简单，适用于小数据集。总的来说，数据挖掘中针对聚类的典型要求包括：

（1）**可伸缩性**：当数据量从几百上升到几百万时，聚类结果的准确度能一致。

（2）**处理不同类型属性的能力**：许多算法针对的数值类型的数据。但是，实际应用场景中，会遇到二元类型数据，分类/标称类型数据，序数型数据。

（3）**发现任意形状的类簇**：许多聚类算法基于距离（欧式距离或曼哈顿距离）来量化对象之间的相似度。基于这种方式，我们往往只能发现相似尺寸和密度的球状类簇或者凸型类簇。但是，实际中类簇的形状可能是任意的。

（4）**初始化参数的需求最小化**：很多算法需要用户提供一定个数的初始参数，比如期望的类簇个数，类簇初始中心点的设定。聚类的结果对这些参数十分敏感，调参数需要大量的人力负担，也非常影响聚类结果的准确性。

（5）**处理噪声数据的能力**：噪声数据通常可以理解为影响聚类结果的干扰数据，包含孤立点，错误数据等，一些算法对这些噪声数据非常敏感，会导致低质量的聚类。

（6）**增量聚类和对输入次序的不敏感**：一些算法不能将新加入的数据快速插入到已有的聚类结果中，还有一些算法针对不同次序的数据输入，产生的聚类结果差异很大。

（7）**高维性**：有些算法只能处理2到3维的低纬度数据，而处理高维数据的能力很弱，高维空间中的数据分布十分稀疏，且高度倾斜。

（8）**可解释性和可用性**：我们希望得到的聚类结果都能用特定的语义、知识进行解释，和实际的应用场景相联系。

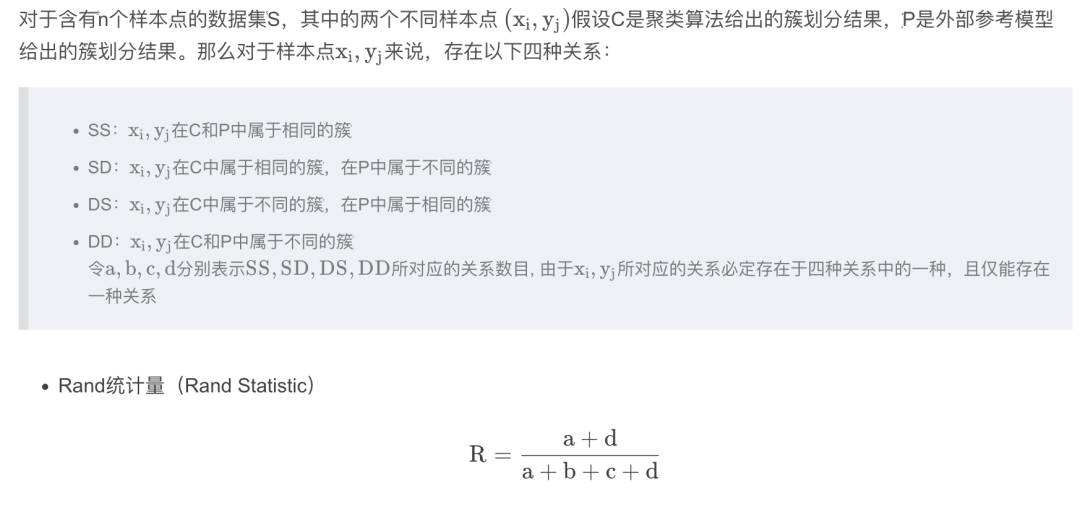
**聚类分析的度量**

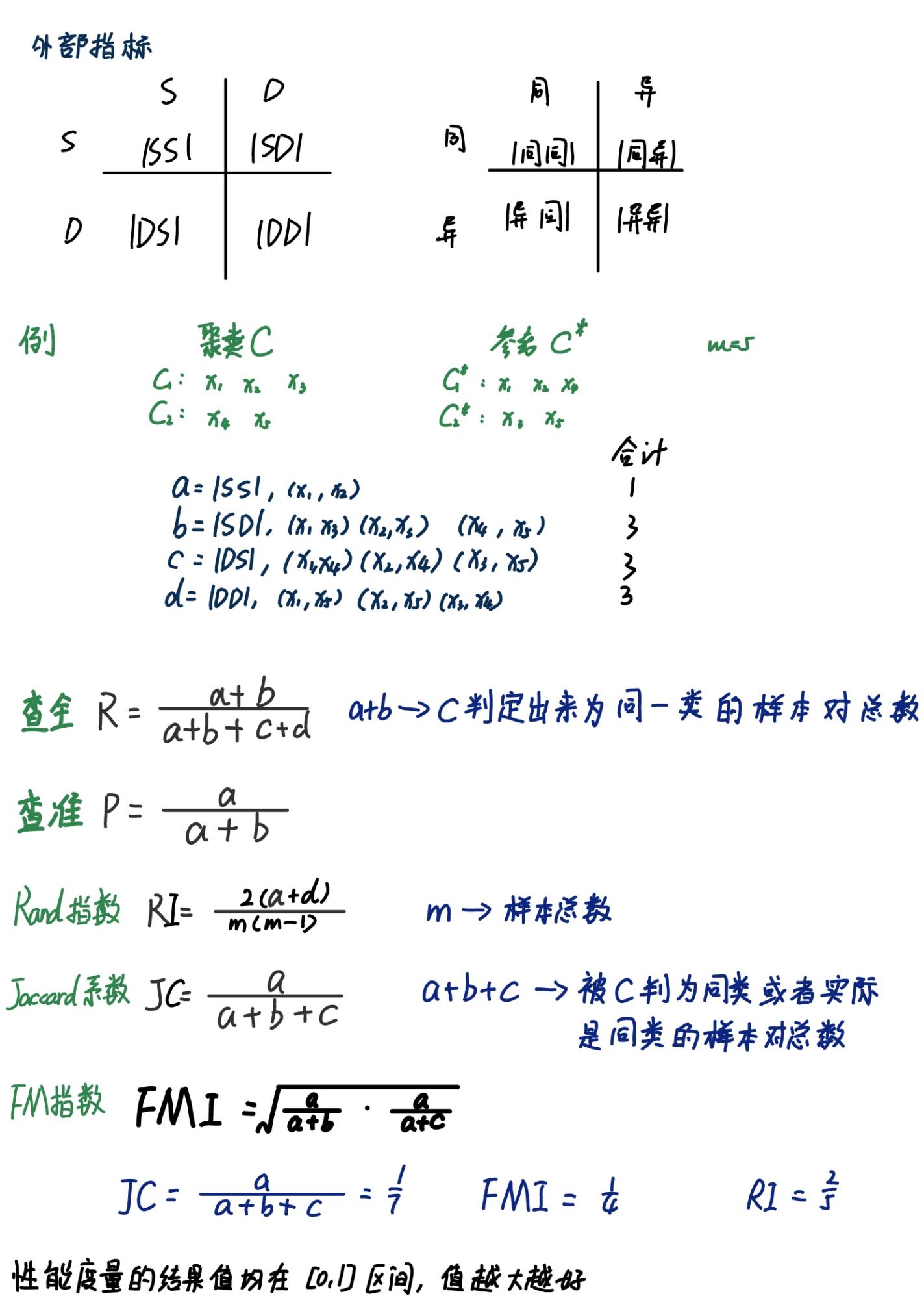
聚类分析的度量指标用于对聚类结果进行评判，分为内部指标和外部指标两大类

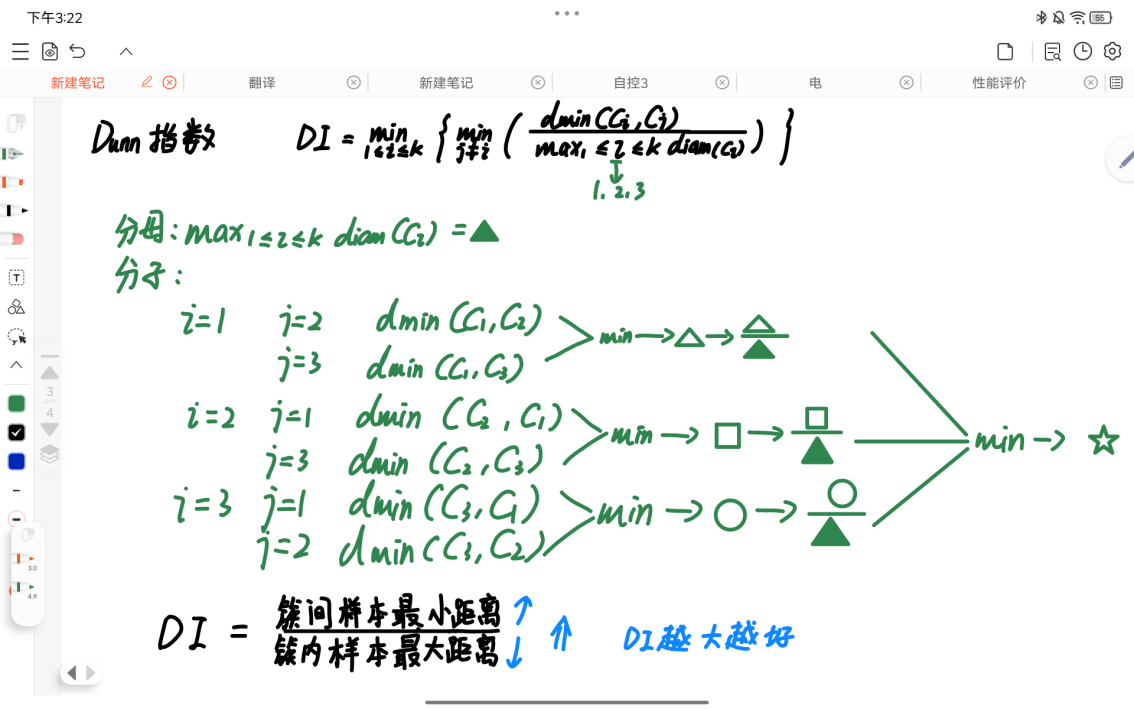
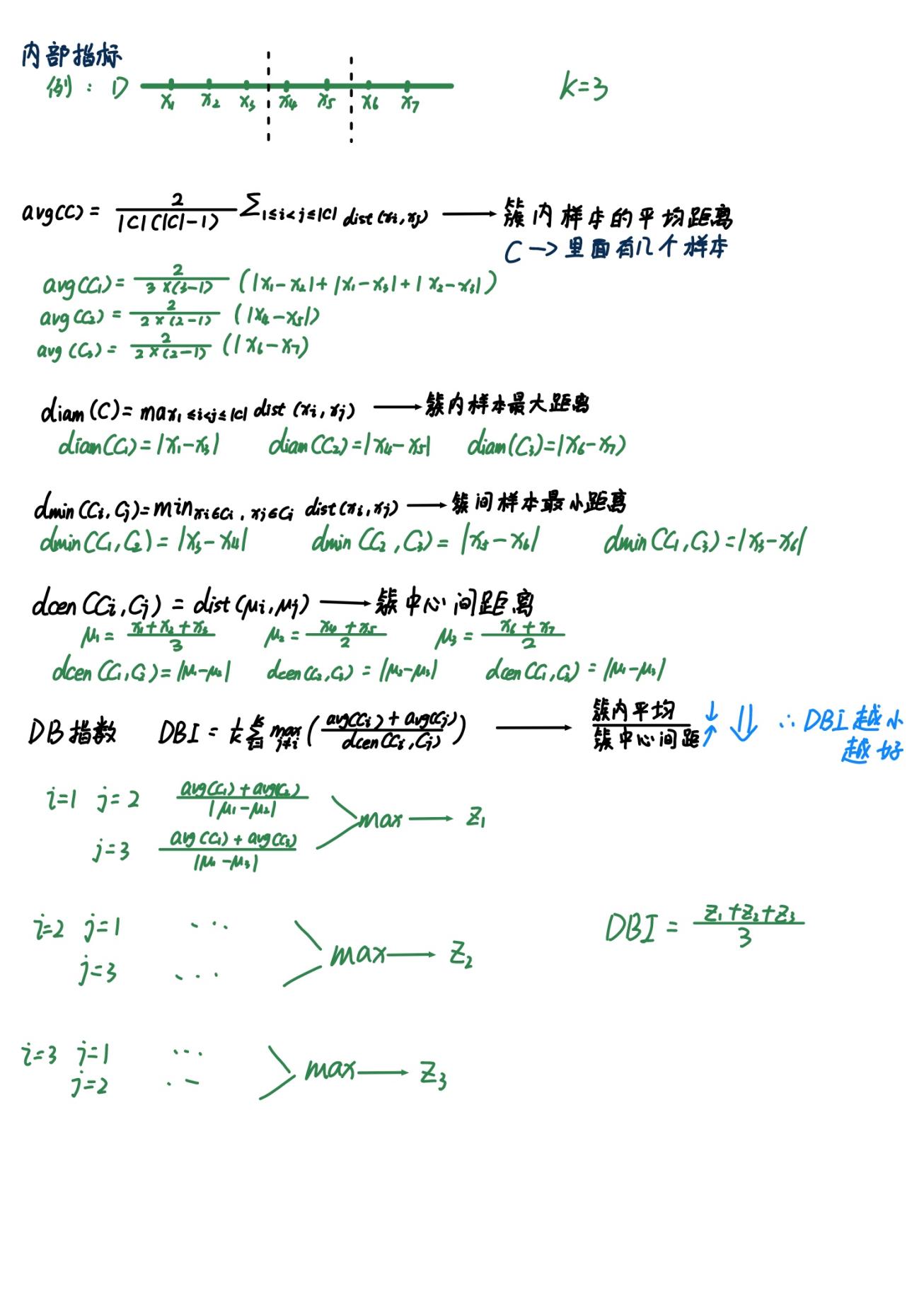
**外部指标**指用事先指定的聚类模型作为参考来评判聚类结果的好坏

**内部指标**是指不借助任何外部参考，只用参与聚类的样本评判聚类结果好坏

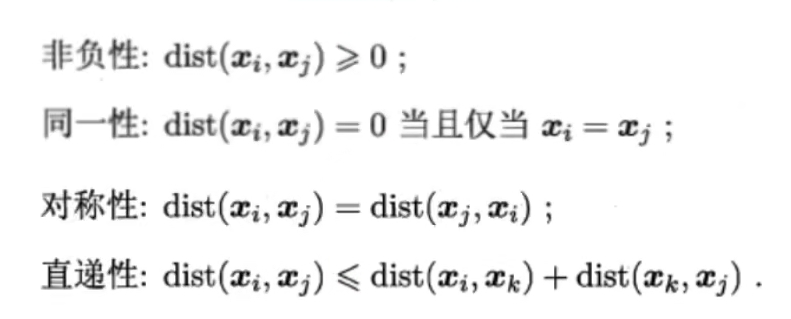
**外部指标**

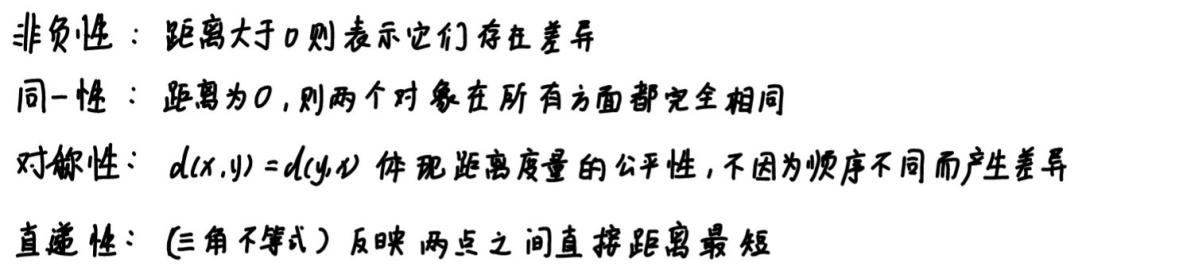


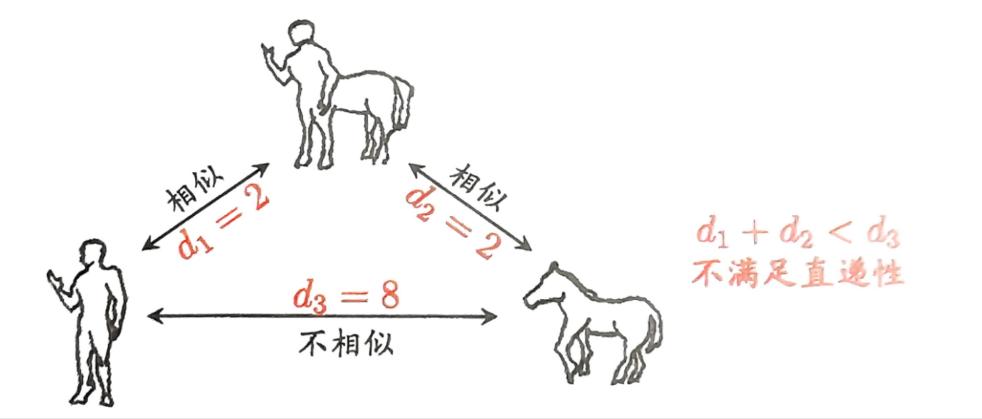


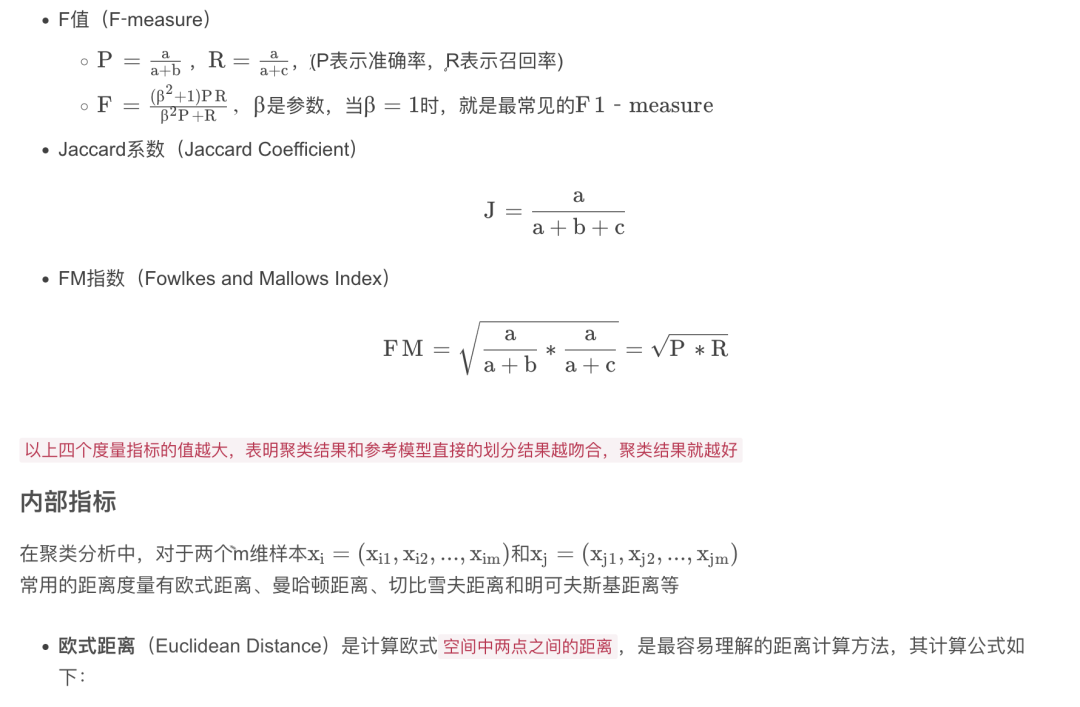


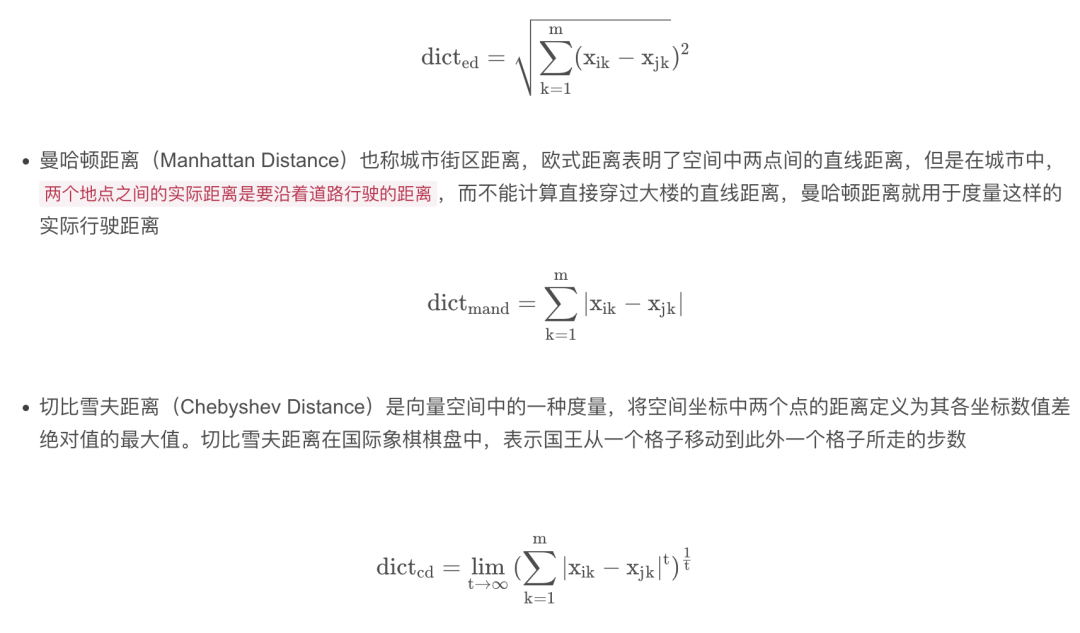
**有序属性**

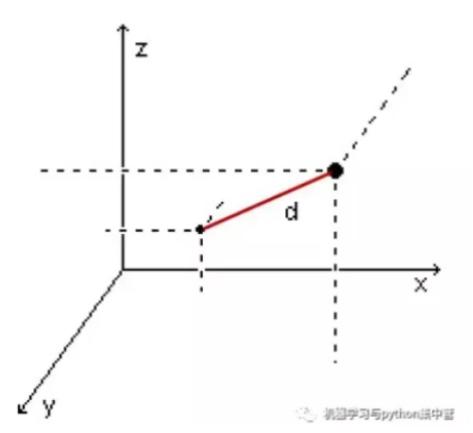
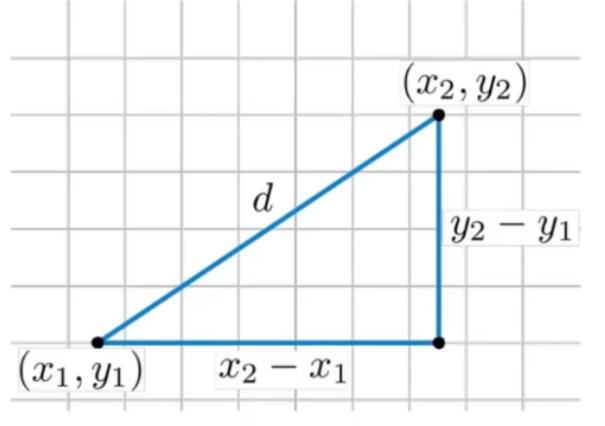


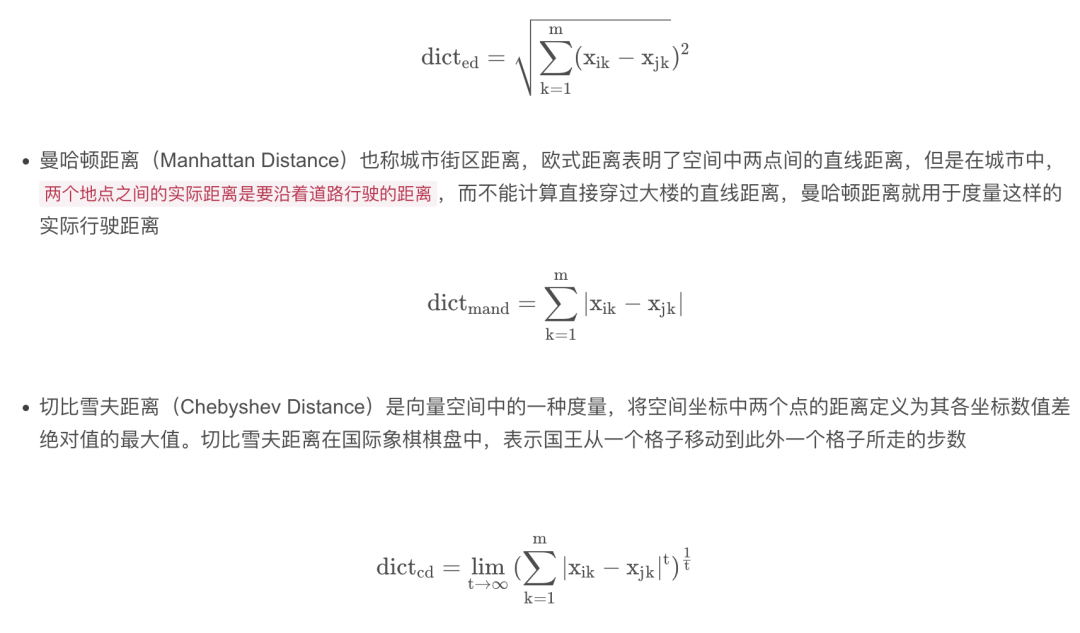


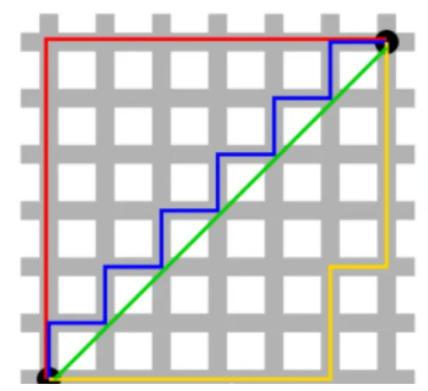


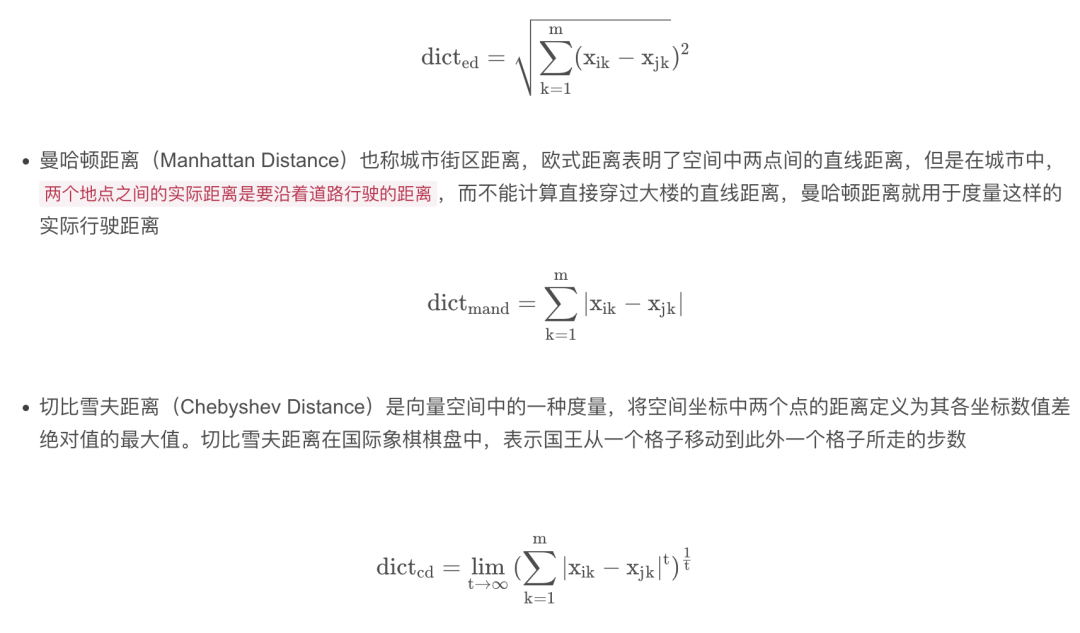


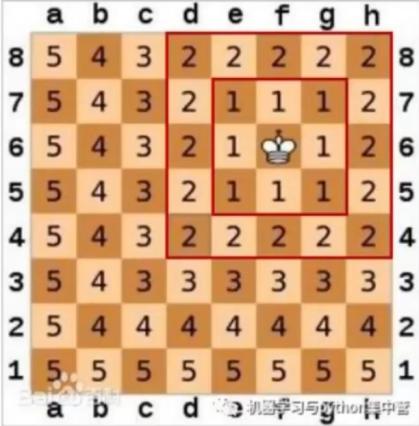


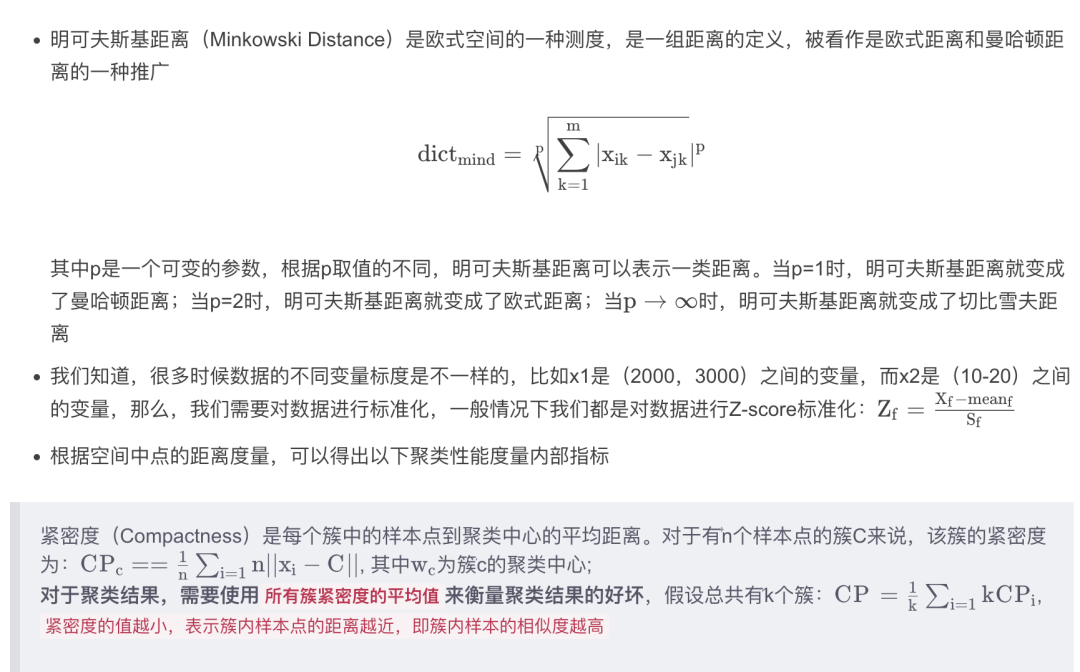












有序属性量化数据点间的相似性和差异性

**聚类的过程**

数据准备：包括特征标准化和降维；

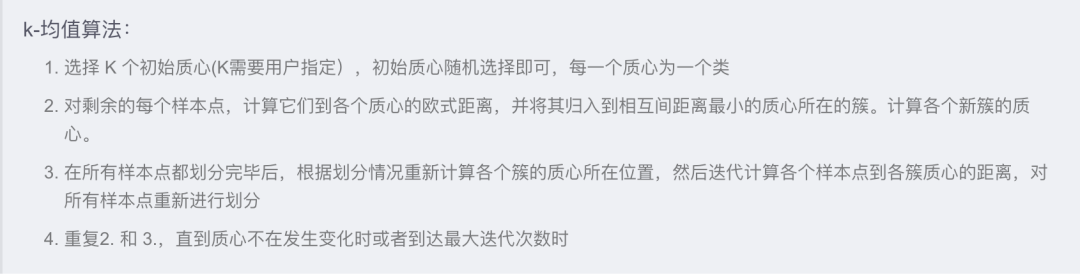
特征选择：从最初的特征中选择最有效的特征,并将其存储于向量中；

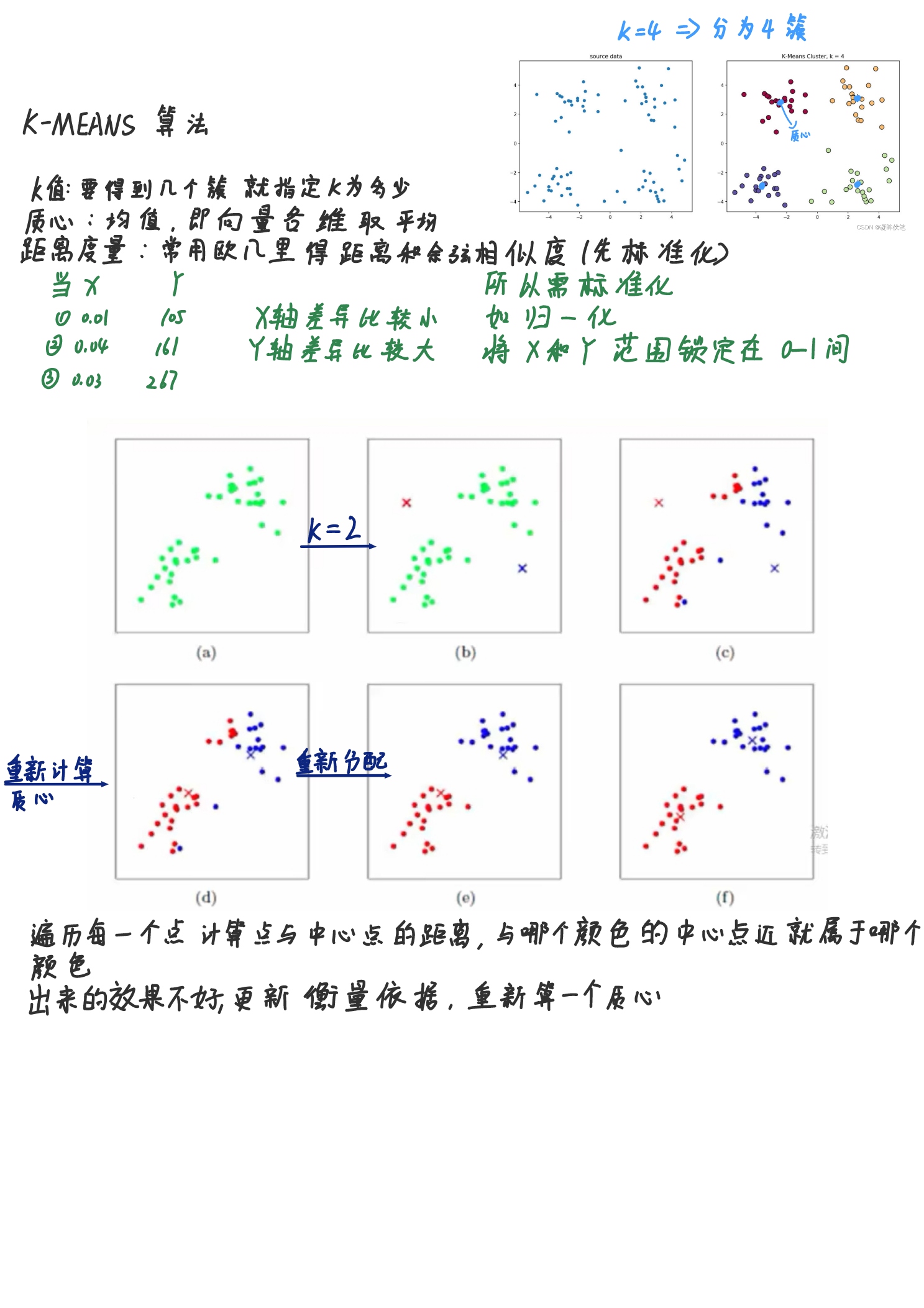
特征提取：通过对所选择的特征进行转换形成新的突出特征；

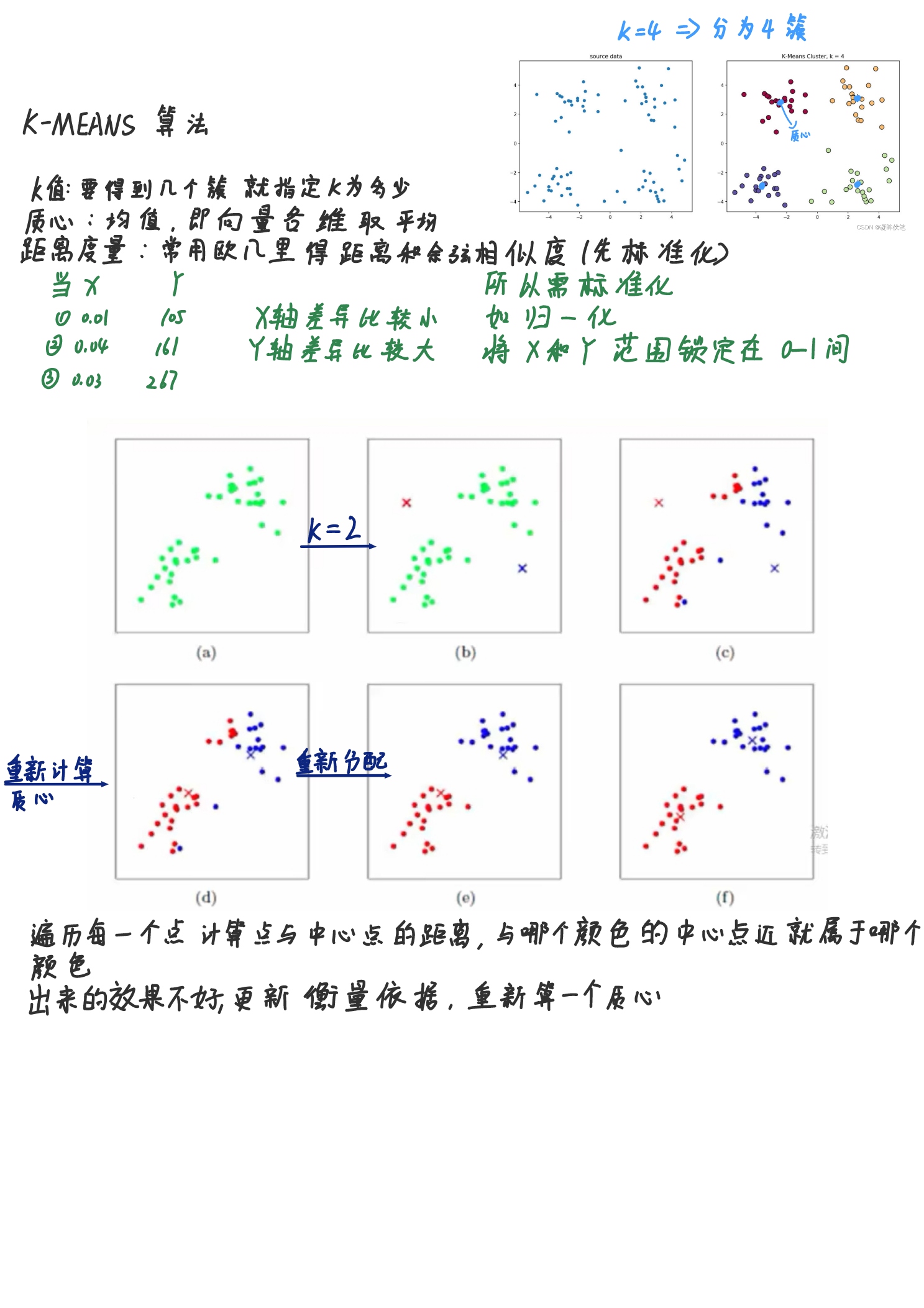
聚类(或分组)：首先选择合适特征类型的某种距离函数(或构造新的距离函数)进行接近程度的度量，而后执行聚类或分组；

****K-means聚类****

k‐均值聚类是基于划分的聚类算法，计算样本点与类簇质心的距离，与类簇质心相近的样本点划分为同一类簇。k‐均值通过样本间的距离来衡量它们之间的相似度，两个样本距离越远，则相似度越低，否则相似度越高。

****

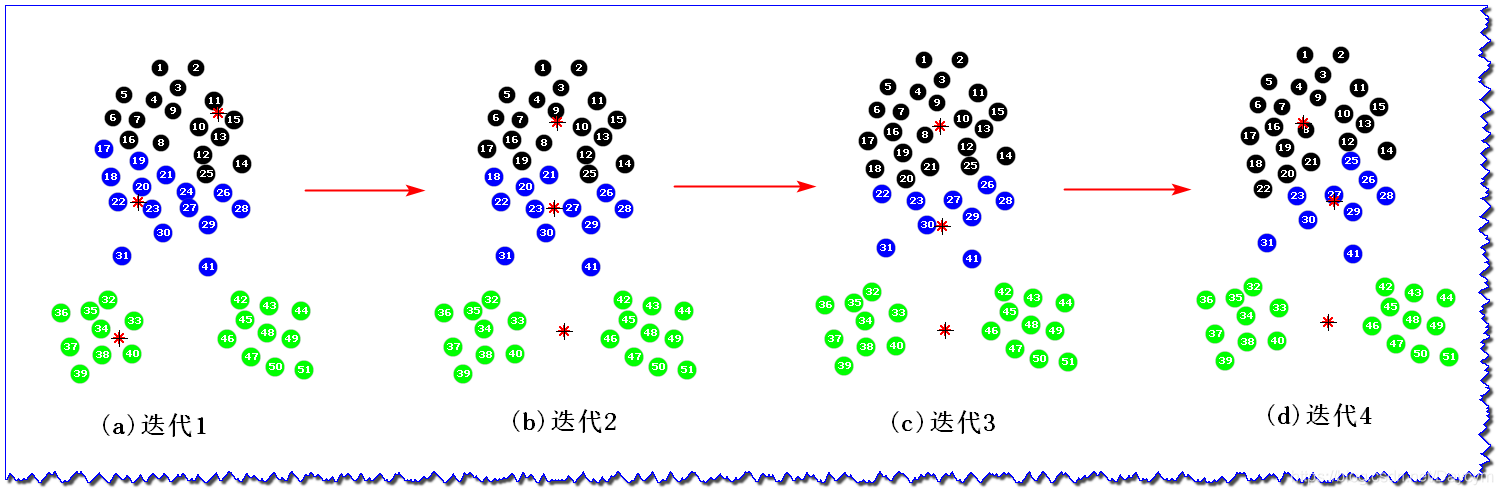
****

****

**下面是 K均值 算法涉及的几个问题：**

**选择初始质心**

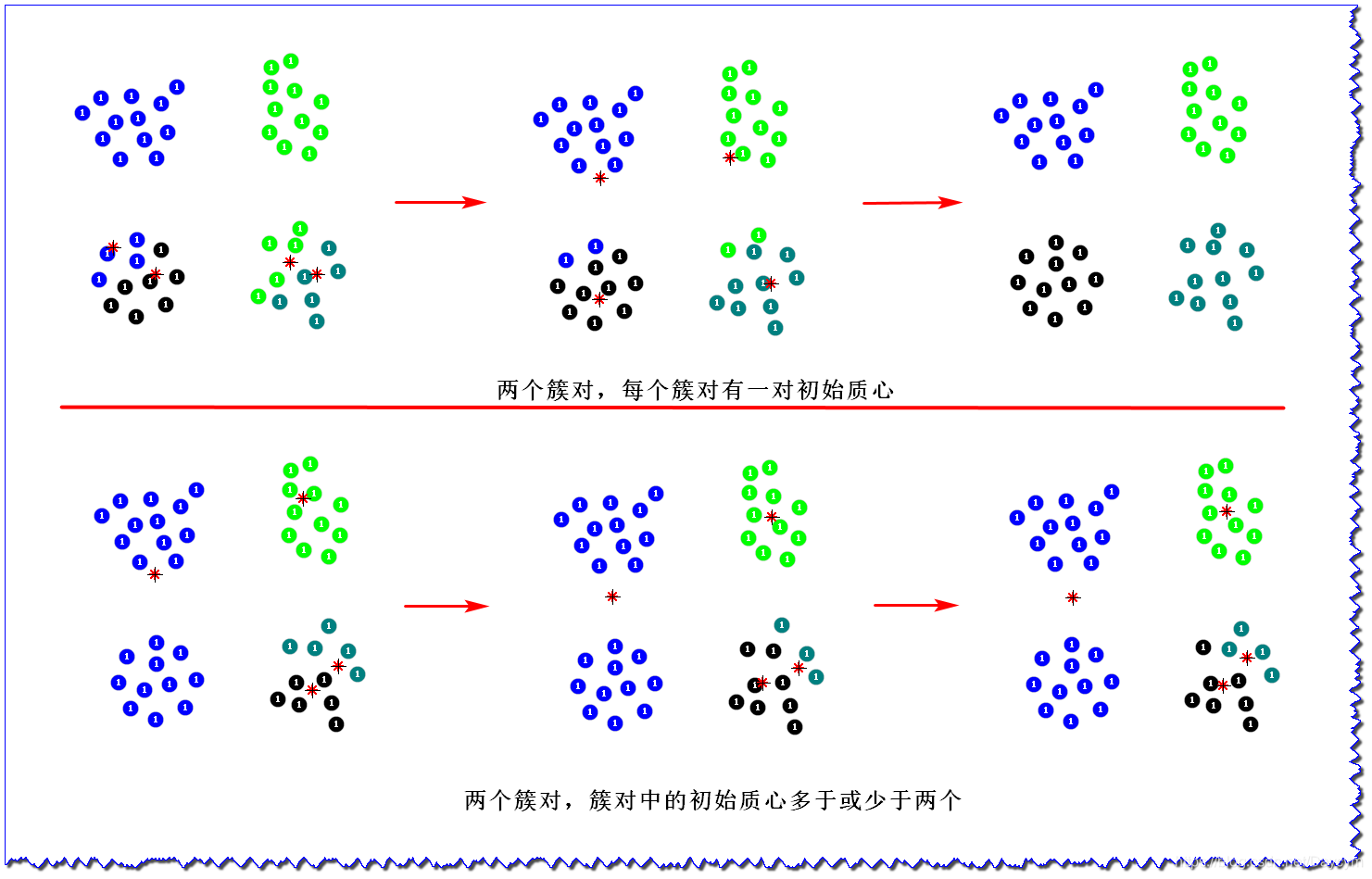
当质心随机初始化时，K均值的不同运行将产生不同的总SSE，选择适当的初始化质心是基本K均值过程的关键步骤，常见的方法是随机地选择初始化质心，但是簇的质量常常很差。

****

不好的初始质心带来的效果：

尽管所有的初始质心都在自然簇中，结果也找到了最小SSE 聚类；

但是仅得到了一个**次最优聚类**，具有较高的平方误差。

****

以上数据由两个簇组成，其中簇对（上下）中的簇更靠近，而离另一对中的簇较远

随机初始化的局限：

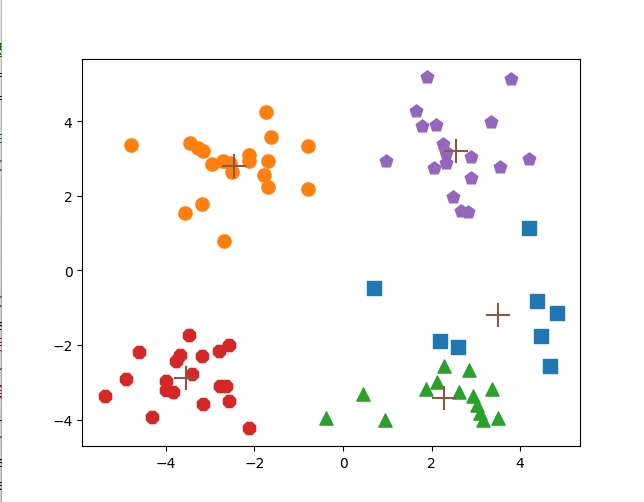
处理选取初始质心问题的一种常用技术就是：**多次运行**，每次使用一组不同的随机初始质心，然后选取具有最小SSE 的簇集；

该策略虽然简单，但是效果可能不好，这取决于数据集和寻找的簇的个数；如果我们对每个簇对用两个初始质心，则即使两个质心在同一个簇中，质心也会自己重新分布，从而找到“真正的”簇；如果一个簇只用一个初始质心，而另一个使用三个质心，则两个真正的簇将合并，而一个真正的簇将分裂

**K-means 的代码**



结果



1. **means 的优缺点**

**优点**

**简单高效**：适合大规模数据,处理大数据集时非常高效，具有良好的伸缩性。

**收敛速度快**：在适合的初始中心选择下，K-means 通常可以较快收敛。

**缺点**

**K值**难确定。

对**初始点**敏感：初始簇中心的选择对最终结果影响较大。

非凸形状簇(球形簇)：K-means 假设每个簇是凸形且大小相近，不适合发现非凸形状的簇或大小差异很大的簇,如“月牙型”数据。

对噪声敏感：离群点会影响簇的中心计算。

**局部最优**：K-means不能保证全局最优，只能达到局部最优

数据种类限制：算法要求样本存在均值，限制了数据的种类

****二分K-means聚类****

是一种改进的聚类算法，它是K-Means算法的一种变体。

与传统的K-Means算法一次性生成K个聚类不同，二分K-Means通过**递归地将一个聚类分裂成两个**，直到达到所需的聚类数目。

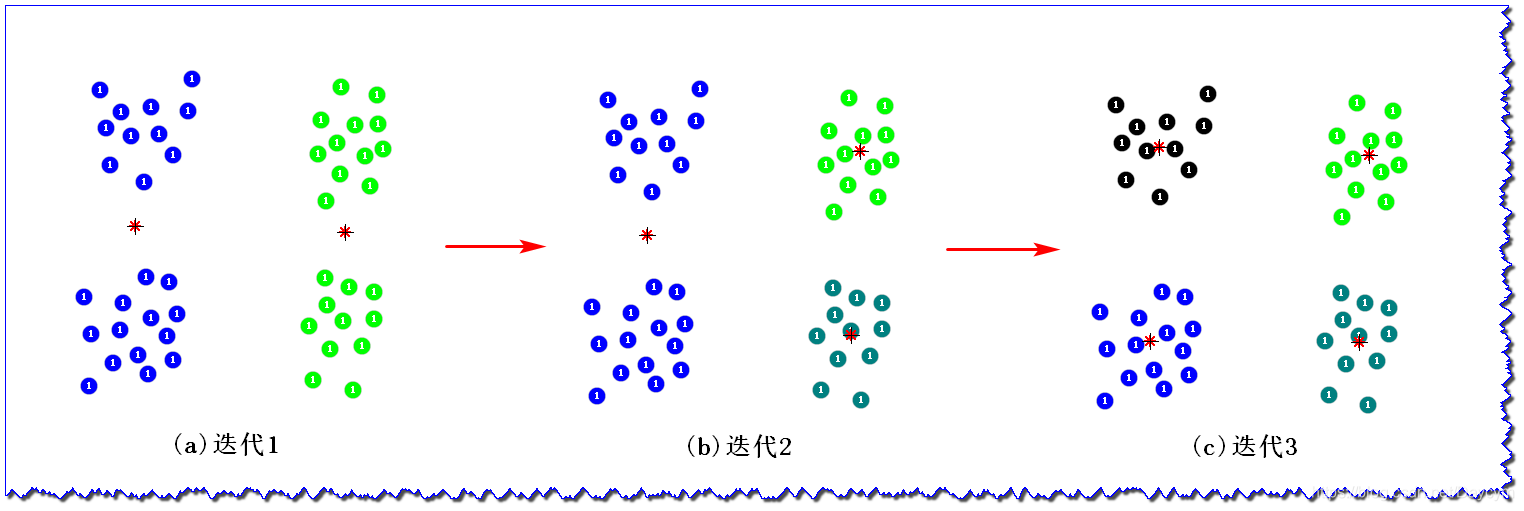
**二分K-Means算法流程：**

**初始化**：将所有数据点视为一个簇。

**分裂**：从当前的簇中选择一个簇进行分裂。选择的标准通常是选择那些分裂后可以最大程度减少**误差平方和（SSE）**的簇。

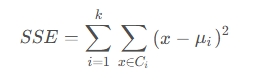
**应用K-Means**：对选定的簇应用标准的K-Means算法，将其分成两个簇。

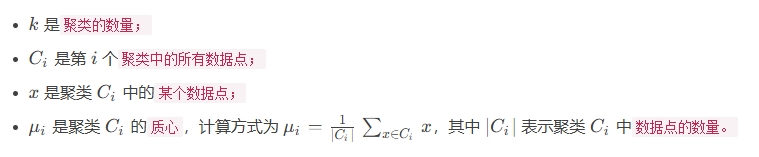
**评估**：检查是否达到了所需的聚类数目。如果没有，返回步骤2；如果达到了，算法结束。



**涉及到的公式：**

在二分K-Means中，最重要的公式是计算误差平方和（SSE），它用于决定哪个簇应该被分裂以及评估聚类的效果。SSE的公式如下：





**二分K-means 的代码**



结果

