百度智能云

无监督学习

聚类

什么是聚类

百度智能云

- 有监督学习
  - 回归
    - 线性回归
    - 岭回归
  - 分类
    - 朴素贝叶斯
    - svm
- 无监督学习
  - 聚类
    - k-means
  - 降维
    - PCA降维

k-means

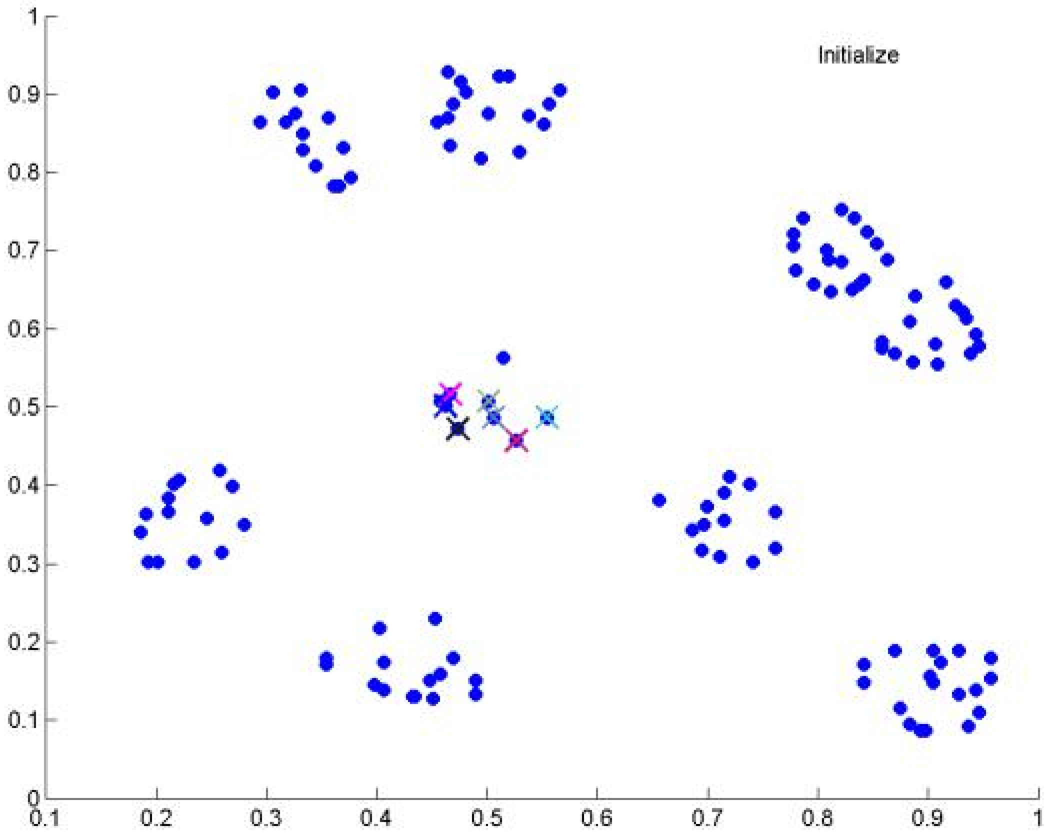
k-means算法运算步骤

- 假定我们要对N个样本观测做聚类，要求聚为K类:
  - 1.初始化：选择K个点作为初始中心点
  - 2.计算所有样本到所有中心点的距离（中心点的数量为k）
  - 3.把样本归为距离中心点最近的类别（共k个类别）
  - 4.计算每个类别内的样本均值
    - 该均值作为新的中心点
  - 5.重复第2-4步，直到结束。
    - 结束条件：中心点不再改变或达到指定的迭代次数

k-means算法运算步骤

百度智能云

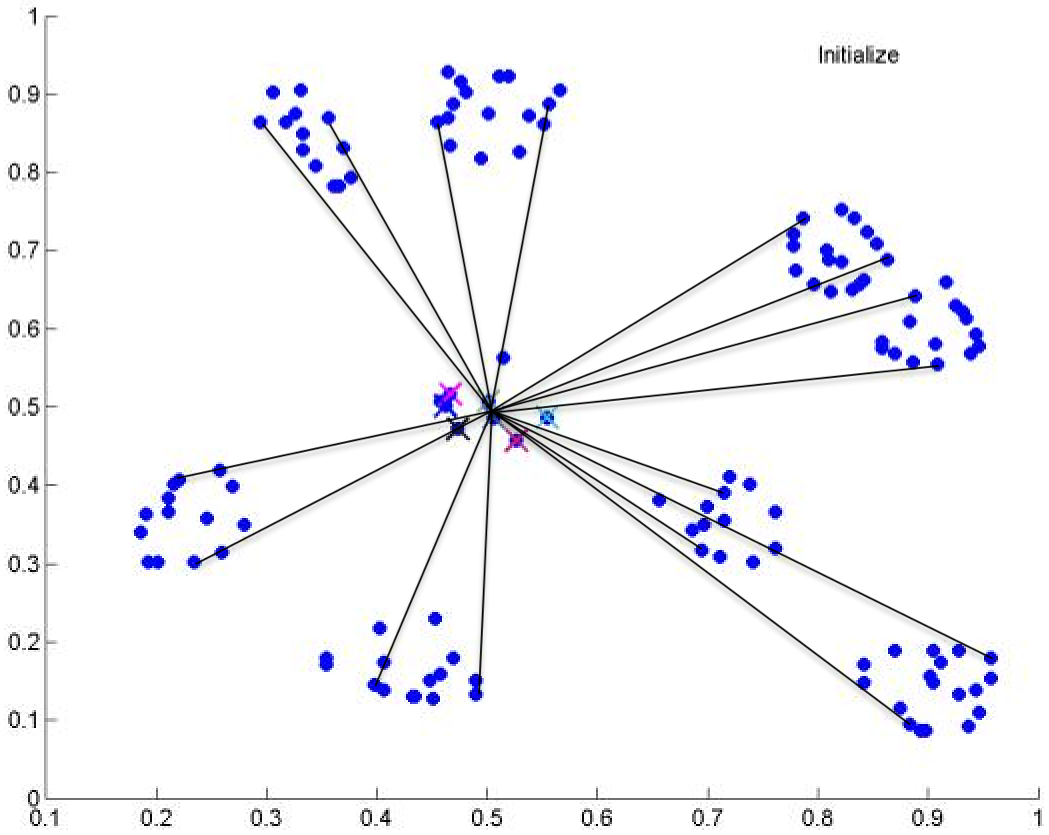
- 假定我们要对N个样本观测做聚类，要求聚为K类：
  - 1.初始化：选择K个点作为初始中心点
  - 2.计算所有样本到所有中心点的距离（中心点的数量为k）
  - 3.把样本归为距离中心点最近的类别（共k个类别）
  - 4.计算每个类别内的样本均值
    - 该均值作为新的中心点
  - 5.重复第2-4步，直到结束。
    - 结束条件：中心点不再改变或达到指定的迭代次数



k-means算法运算步骤

百度智能云

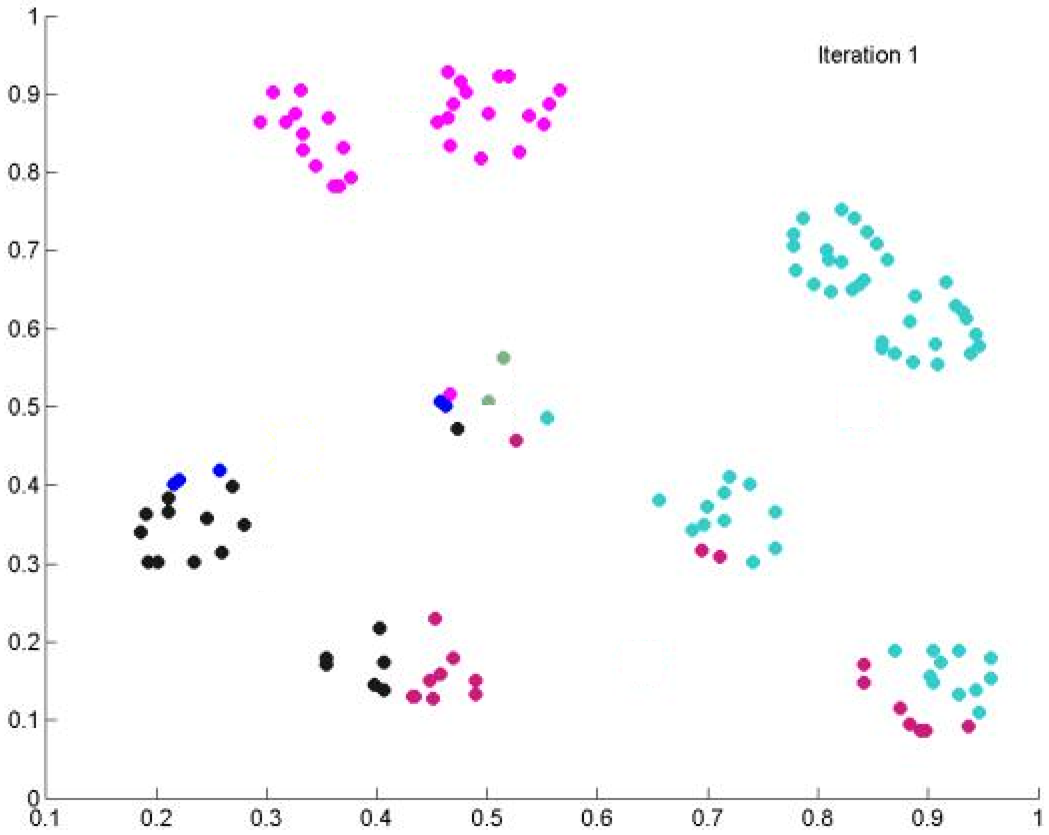
- 假定我们要对N个样本观测做聚类，要求聚为K类：
  - 1.初始化：选择K个点作为初始中心点
  - 2.计算所有样本到所有中心点的距离（中心点的数量为k）
  - 3.把样本归为距离中心点最近的类别（共k个类别）
  - 4.计算每个类别内的样本均值
    - 该均值作为新的中心点
  - 5.重复第2-4步，直到结束。
    - 结束条件：中心点不再改变或达到指定的迭代次数



k-means算法运算步骤

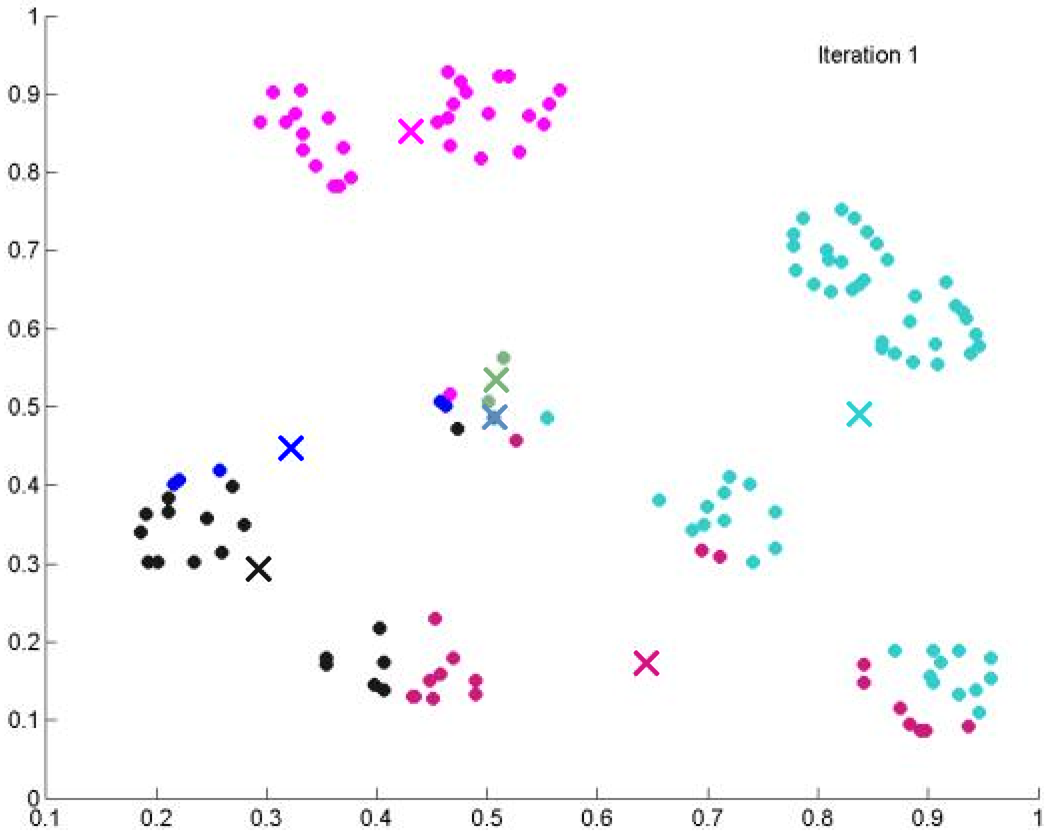
百度智能云

- 假定我们要对N个样本观测做聚类，要求聚为K类：
  - 1.初始化：选择K个点作为初始中心点
  - 2.计算所有样本到所有中心点的距离（中心点的数量为k）
  - 3.把样本归为距离中心点最近的类别（共k个类别）
  - 4.计算每个类别内的样本均值
    - 该均值作为新的中心点
  - 5.重复第2-4步，直到结束。
    - 结束条件：中心点不再改变或达到指定的迭代次数



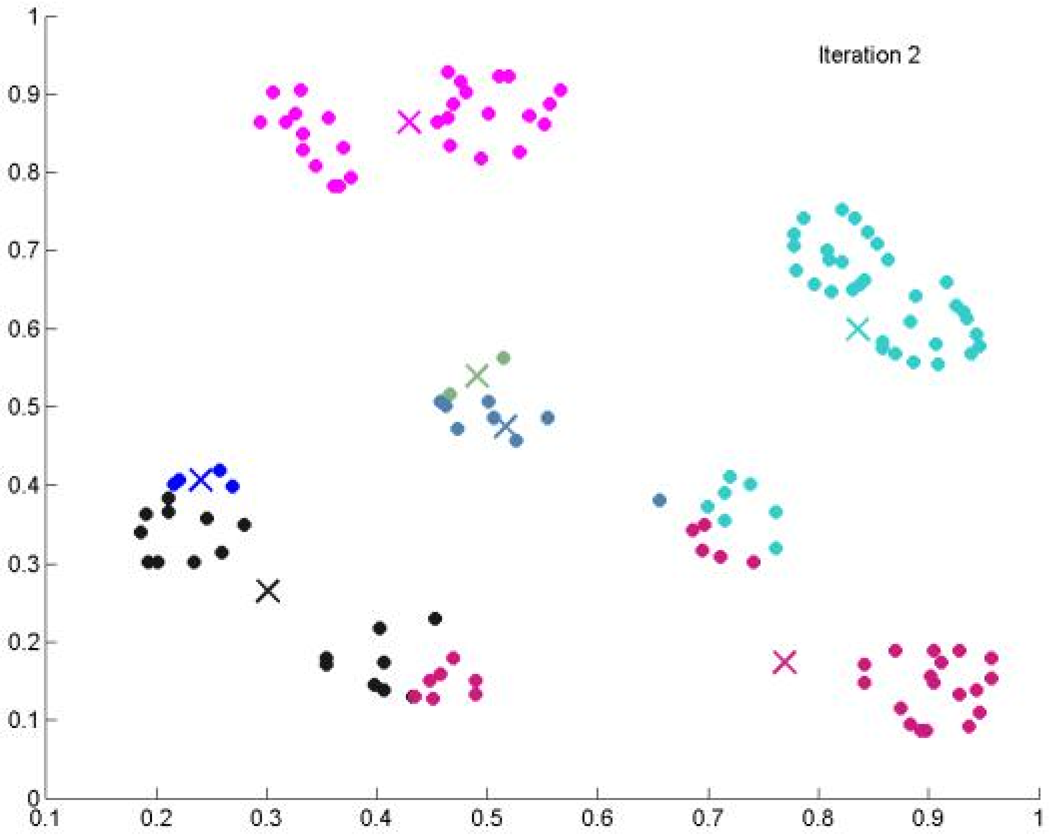
k-means算法运算步骤

- 假定我们要对N个样本观测做聚类，要求聚为K类：
  - 1.初始化：选择K个点作为初始中心点
  - 2.计算所有样本到所有中心点的距离（中心点的数量为k）
  - 3.把样本归为距离中心点最近的类别（共k个类别）
  - 4.计算每个类别内的样本均值
    - 该均值作为新的中心点
  - 5.重复第2-4步，直到结束。
    - 结束条件：中心点不再改变或达到指定的迭代次数



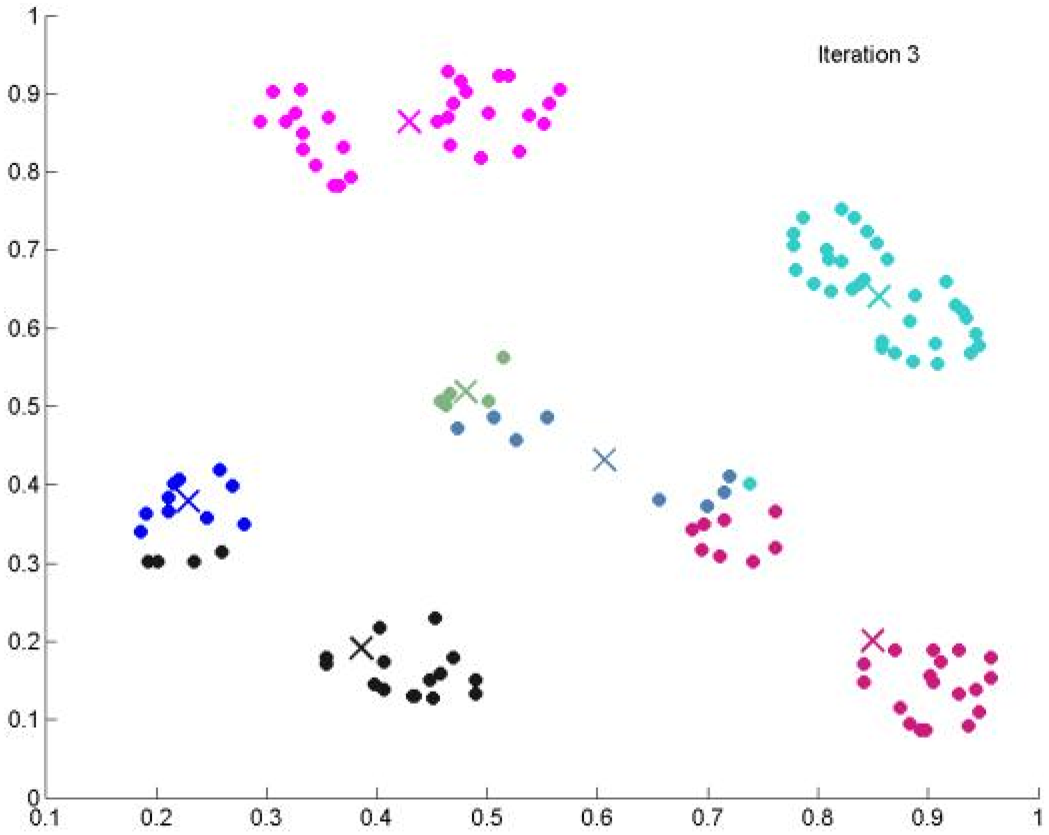
k-means算法运算步骤

- 假定我们要对N个样本观测做聚类，要求聚为K类:
  - 1.初始化：选择K个点作为初始中心点
  - 2.计算所有样本到所有中心点的距离（中心点的数量为k）
  - 3.把样本归为距离中心点最近的类别（共k个类别）
  - 4.计算每个类别内的样本均值
    - 该均值作为新的中心点
  - 5.重复第2-4步，直到结束。
    - 结束条件：中心点不再改变或达到指定的迭代次数



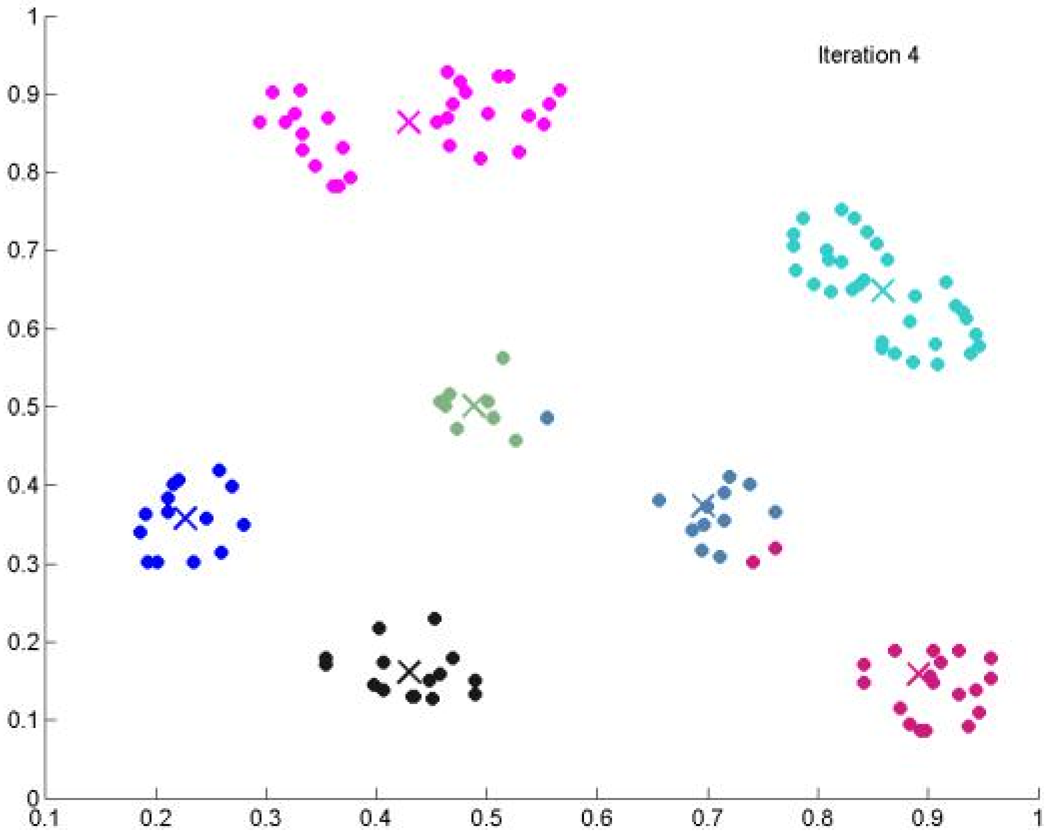
k-means算法运算步骤

- 假定我们要对N个样本观测做聚类，要求聚为K类：
  - 1.初始化：选择K个点作为初始中心点
  - 2.计算所有样本到所有中心点的距离（中心点的数量为k）
  - 3.把样本归为距离中心点最近的类别（共k个类别）
  - 4.计算每个类别内的样本均值
    - 该均值作为新的中心点
  - 5.重复第2-4步，直到结束。
    - 结束条件：中心点不再改变或达到指定的迭代次数



k-means算法运算步骤

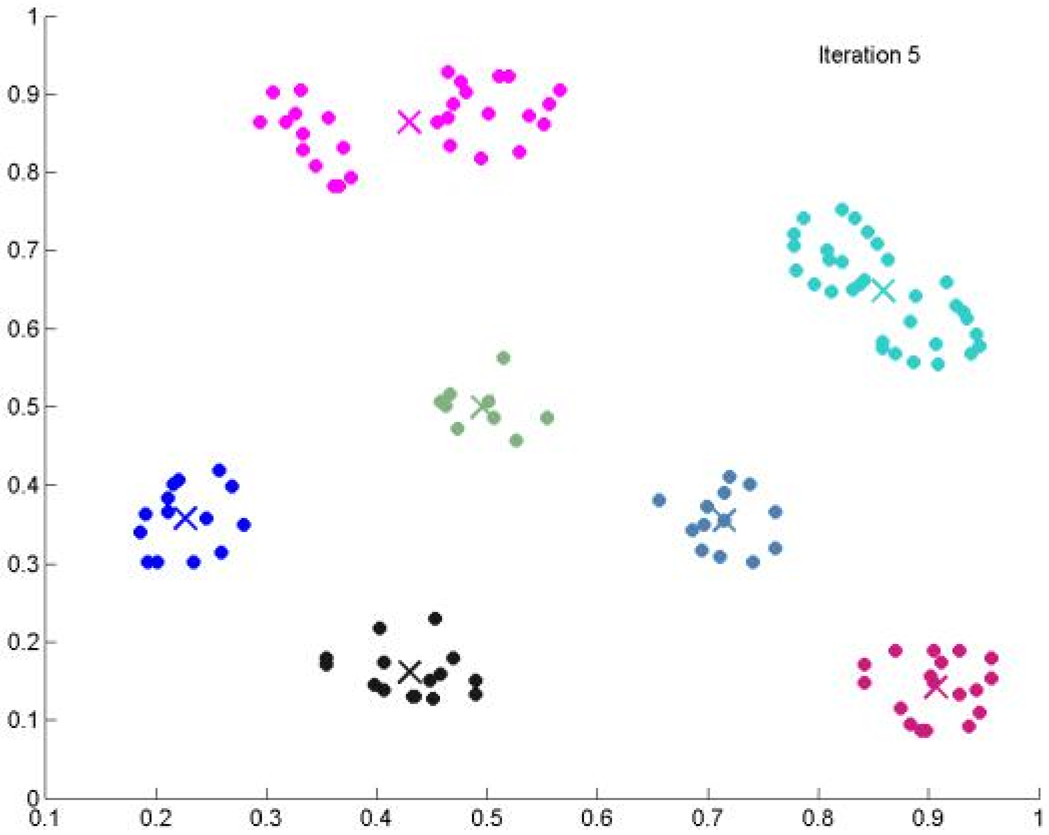
- 假定我们要对N个样本观测做聚类，要求聚为K类：
  - 1.初始化：选择K个点作为初始中心点
  - 2.计算所有样本到所有中心点的距离（中心点的数量为k）
  - 3.把样本归为距离中心点最近的类别（共k个类别）
  - 4.计算每个类别内的样本均值
    - 该均值作为新的中心点
  - 5.重复第2-4步，直到结束。
    - 结束条件：中心点不再改变或达到指定的迭代次数



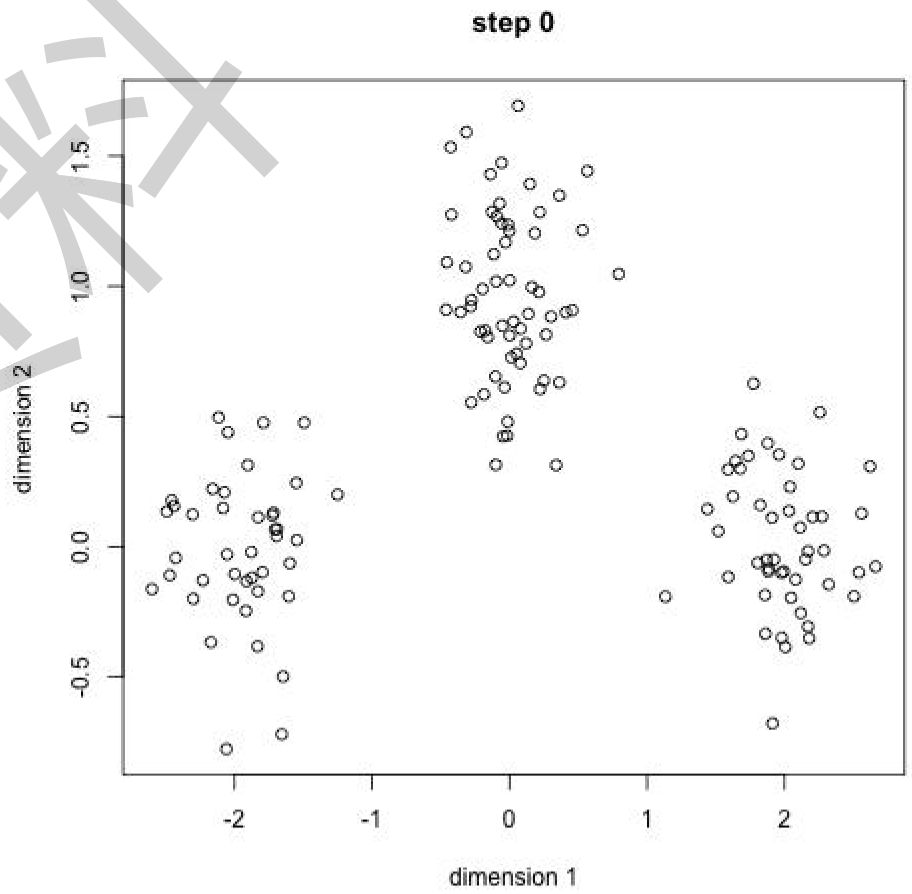
k-means算法运算步骤

百度智能云

- 假定我们要对N个样本观测做聚类，要求聚为K类：
  - 1.初始化：选择K个点作为初始中心点
  - 2.计算所有样本到所有中心点的距离（中心点的数量为k）
  - 3.把样本归为距离中心点最近的类别（共k个类别）
  - 4.计算每个类别内的样本均值
    - 该均值作为新的中心点
  - 5.重复第2-4步，直到结束。
    - 结束条件：中心点不再改变或达到指定的迭代次数



一个例子



距离

- 欧式距离

- 高维向量：
  - $A = \{a_1, a_2, a_3 \ldots a_n\}$
  - $B = \{b_1, b_2, b_3 \ldots b_n\}$

$$D = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \ldots + (a_n - b_n)^2}$$

- 每个数据分配给距离最近的类
  - $\arg\min_j ||x_n - C_j||^2$
  - 其中，C_j为聚类中心点

k-means算法效果的度量

- 误差平方和

$$SSE = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}||x_n - C_k||^2$$

- 其中， x_n表示样本， C_k表示第k个中心点， r_{nk}表示是否属于该类别，如果是则为1，否则为0

- 求极值

$$\frac{\partial}{\partial C_k}SSE = \frac{\partial}{\partial C_k}\sum_{i=1}^{K}\sum_{x\in C_i}(C_i - x)^2 = \sum_{i=1}^{K}\sum_{x\in C_i}\frac{\partial}{\partial C_k}(C_i - x)^2 = \sum_{x\in C_i} 2(C_i - x) = 0$$

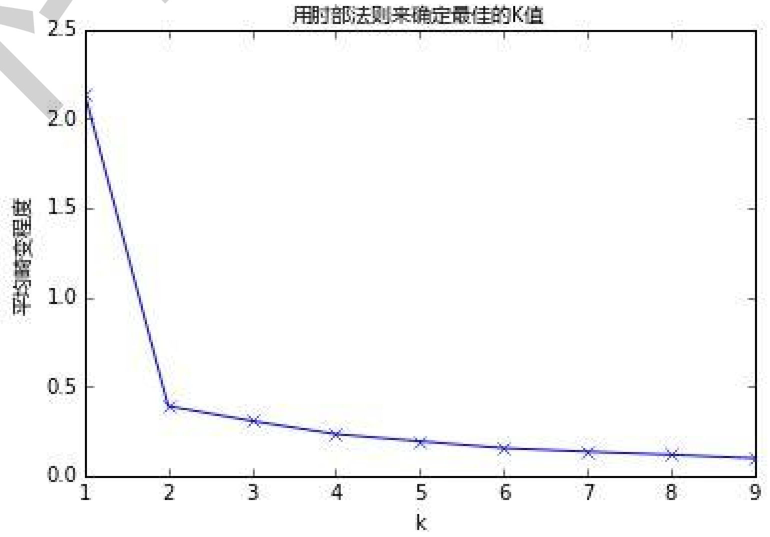
$$\sum_{x\in C_i} 2(C_i - x) = 0 \Rightarrow m_i C_i = \sum_{x\in C_i} x \Rightarrow C_i = \frac{1}{m_i}\sum_{x\in C_i} x$$

$$C = \left(\frac{x_{11}+\ldots+x_{1n}}{m}, \ldots, \frac{x_{m1}+\ldots+x_{mn}}{m}\right)$$

- 中心的选取
  - 随机
  - 多次尝试

K值怎么确定

- K设置得越大，样本划分得就越细，每个簇的聚合程度就越高，误差平方和SSE自然就越小
- 手肘法



算法优缺点

- 优点
  - 算法原理简单
  - 需要设置的超参数少（只有一个k）
  - 收敛速度快
  - 可扩展性好

- 缺点
  - K值、初始点的选取不好确定
  - 得到的结果只是局部最优
  - 受离群值影响大

百度智能云

THANK YOU

CLOUD.BAIDU.COM ABCXUEYUAN.BAIDU.COM