模 式 识 别

第3章：聚类分析

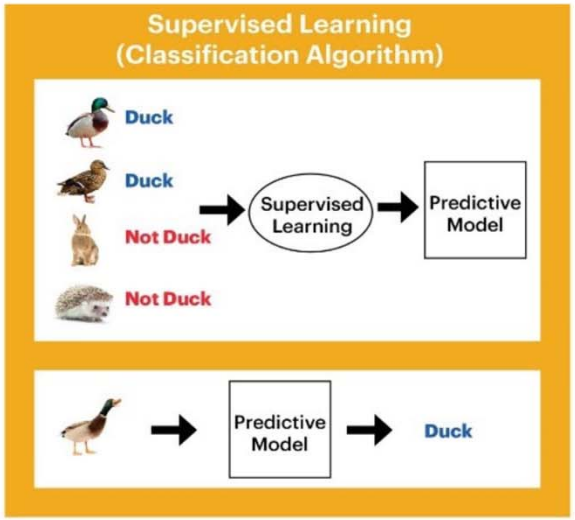
主讲人：张治国
zhiguozhang@hit.edu.cn

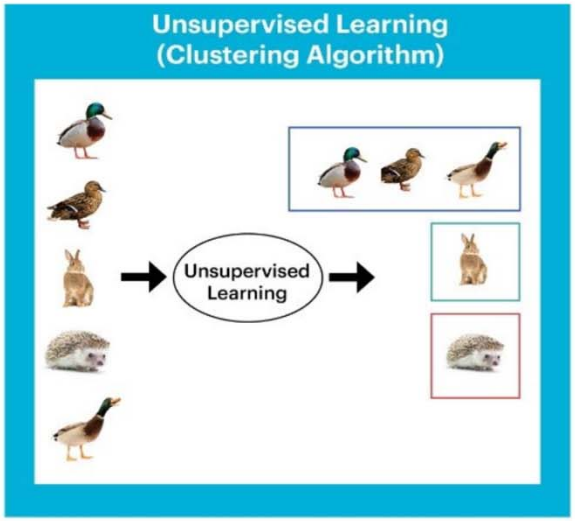
本章内容

- 无监督学习与聚类的基本概念
  - 聚类分析的原理、应用、过程和描述
- 简单聚类方法
  - 顺序聚类
  - 最大最小距离聚类
- 谱系聚类
  - 谱系聚类的合并与分裂算法
- K-均值聚类
  - 基本算法和改进算法
- 聚类检验
  - 聚类结果的检验
  - 聚类数的间接和直接选择

模式识别方法分类

- 根据训练样本有无类别标号，模式识别方法分为有监督学习（分类）和无监督学习（聚类）。
- 有监督（supervised）学习：预先已知训练样本集中每个样本的类别标号。
- 无监督（unsupervised）学习：预先不知道训练集中样本的类别标号，甚至不知道类别的数量。





无监督学习/聚类

- 聚类是人类学习的重要方式，是在实践过程中通过自身的经验累积和总结发现的事物规律。

上面有几种鱼？它们有什么相同点和不同点？

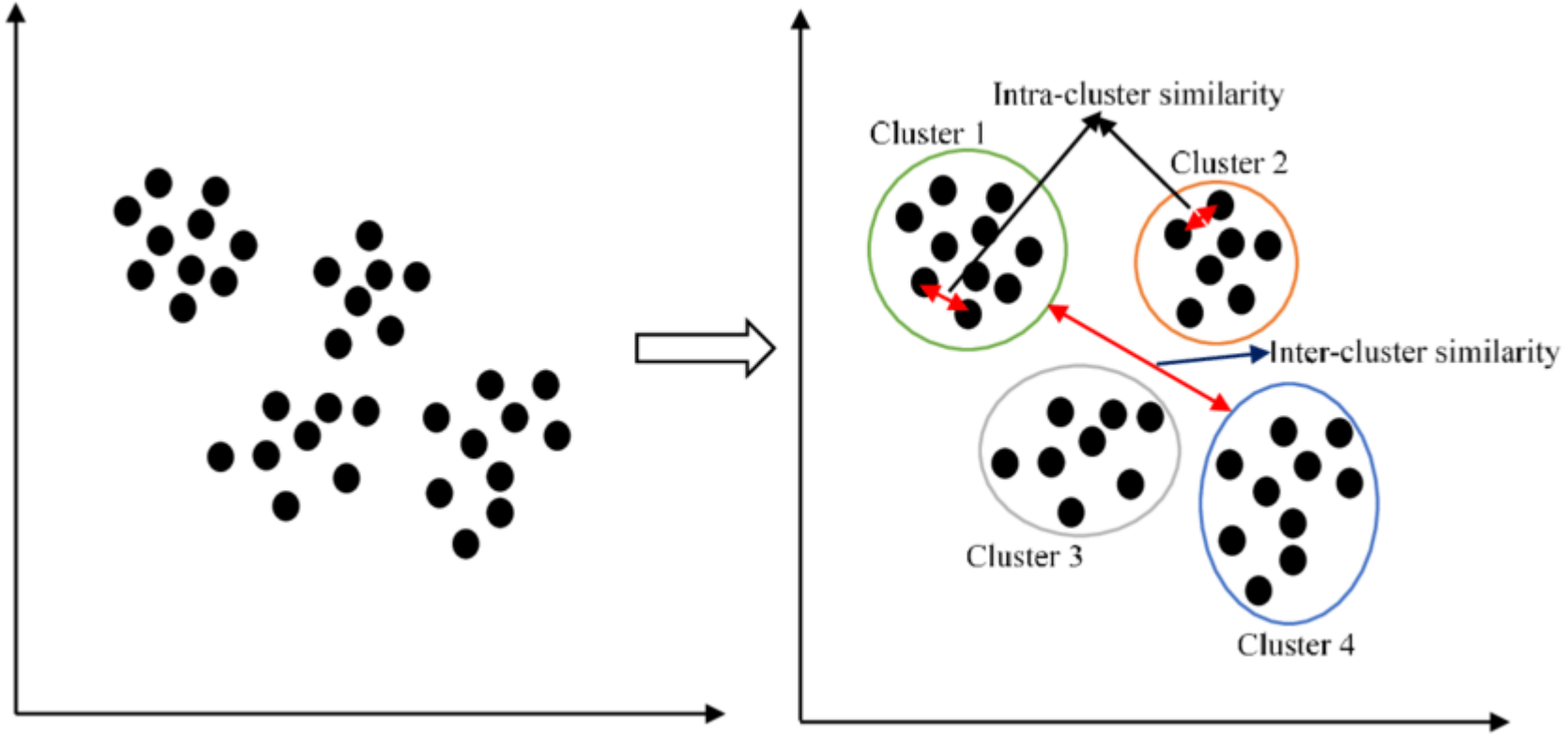
无监督学习/聚类

左侧这些外星生物分为几类?

根据什么原则划分?

聚类

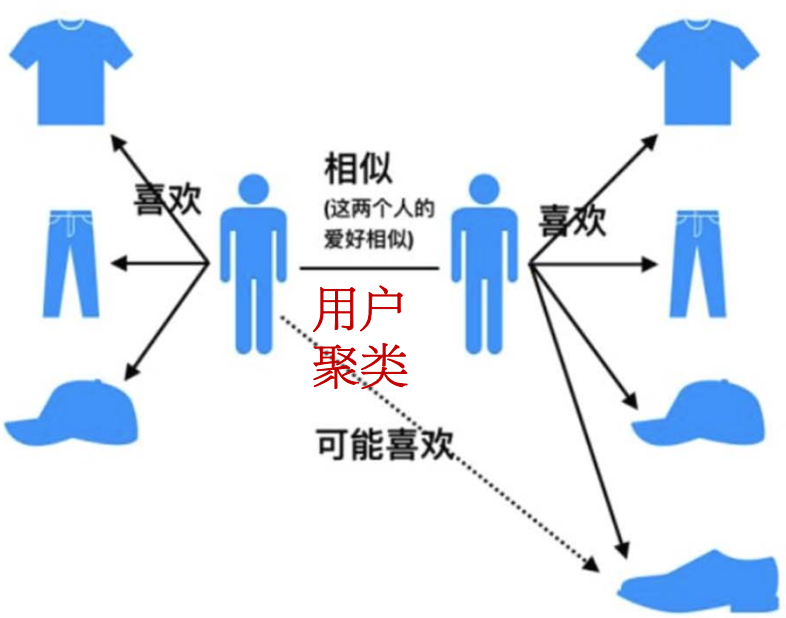
- 聚类：将样本集合划分为若干簇（Cluster）或子集的过程， 使得同一个簇中的样本具有最大相似性，不同簇间的样本具有最大的相异性 。



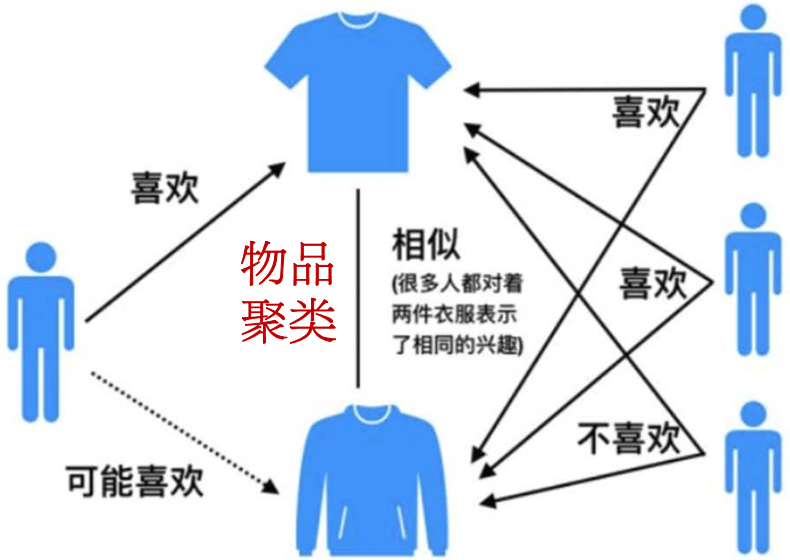
聚类的应用

- 聚类的应用：信息检索、推荐系统、图像分割、数据压缩、精准治疗等等

推荐系统中的用户或物品聚类



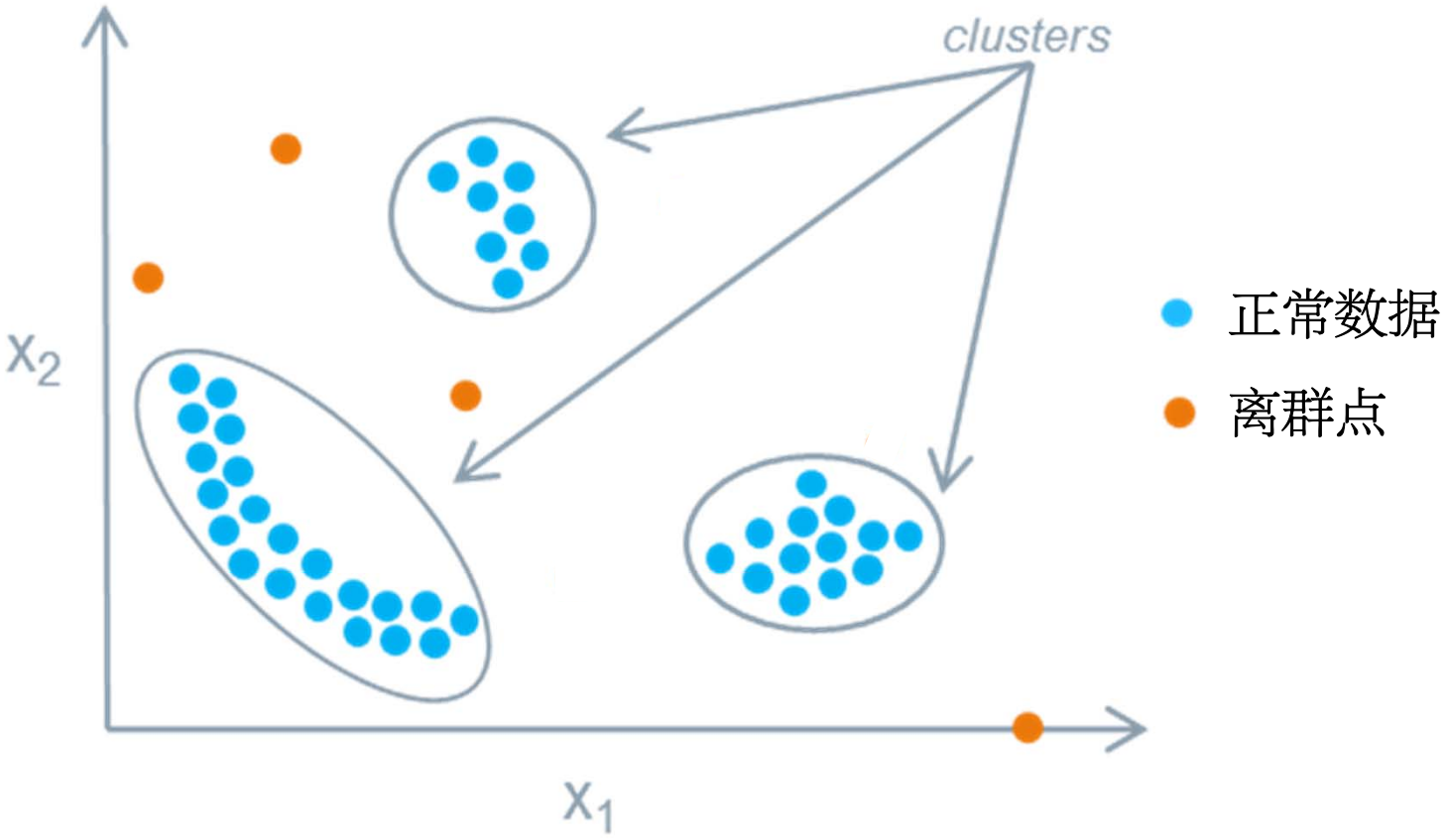
“人以群分”的基于用户的协同过滤



“物以类聚”的基于物品的协同过滤

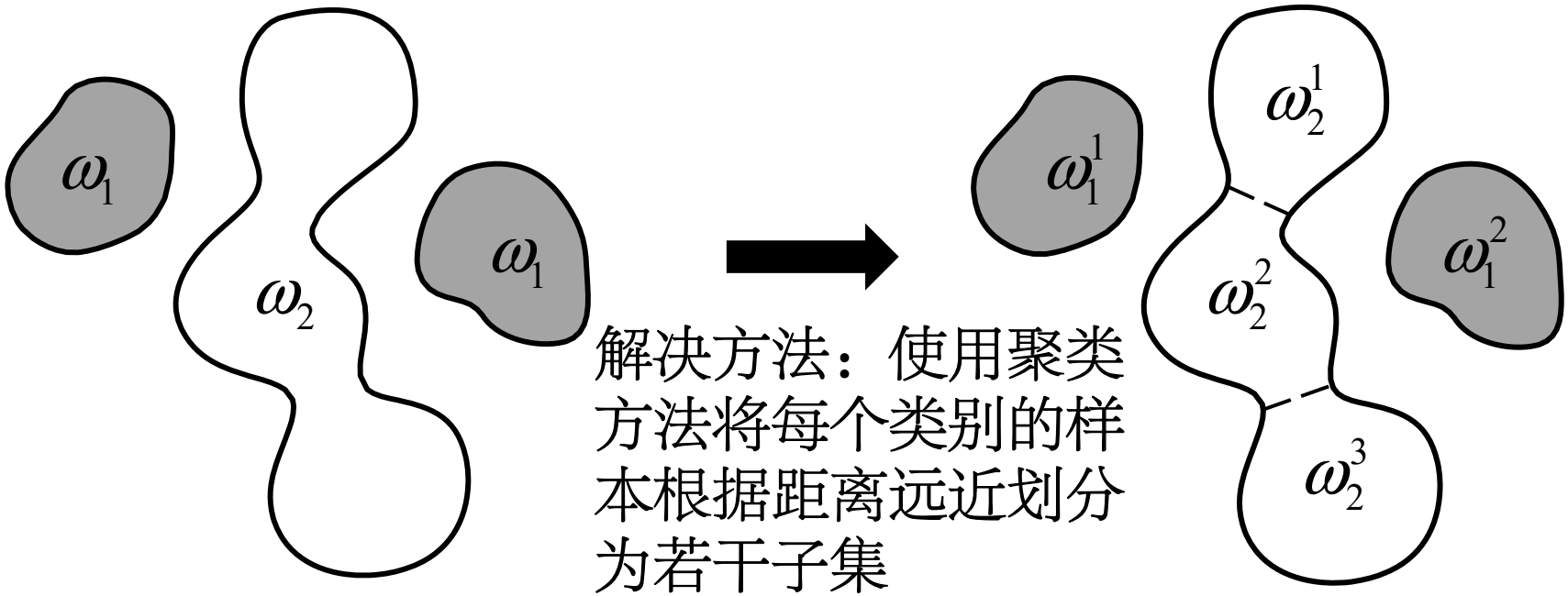
聚类的应用

- 聚类可以在模式识别系统中的数据预处理步骤中使用，以检测离群点。



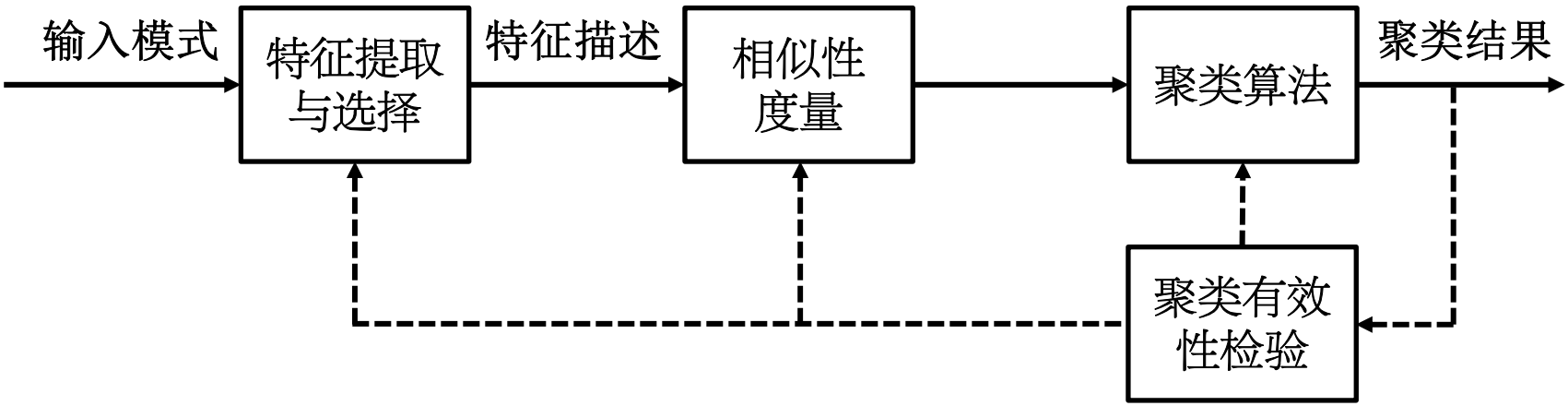
聚类的应用

- 聚类可以作为有监督学习的前处理步骤，提供样本数据的结构信息。
- 例：聚类解决最近邻分类中样本分布不规则问题（见第2讲相关内容）



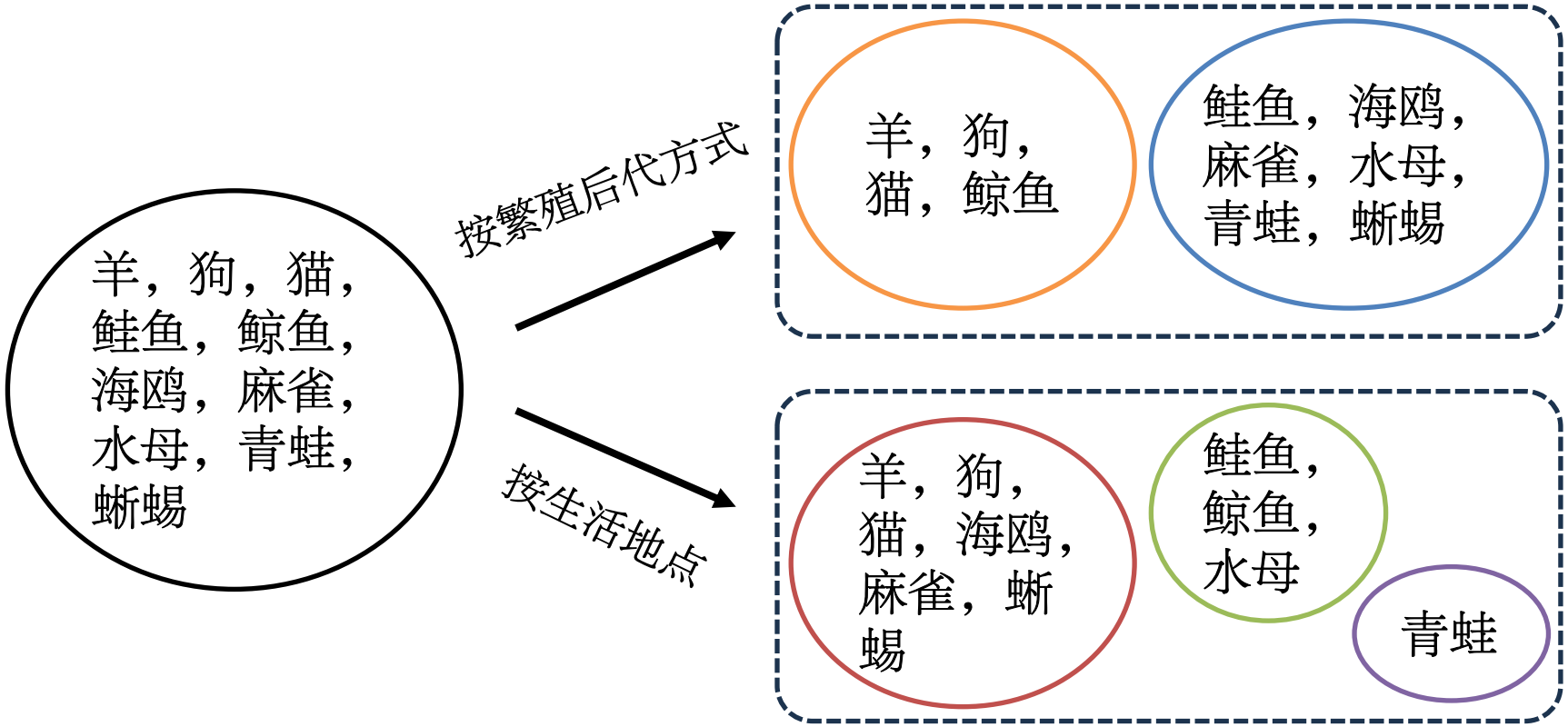
解决方法：使用聚类方法将每个类别的样本根据距离远近划分为若干子集

聚类分析的过程



聚类分析的过程

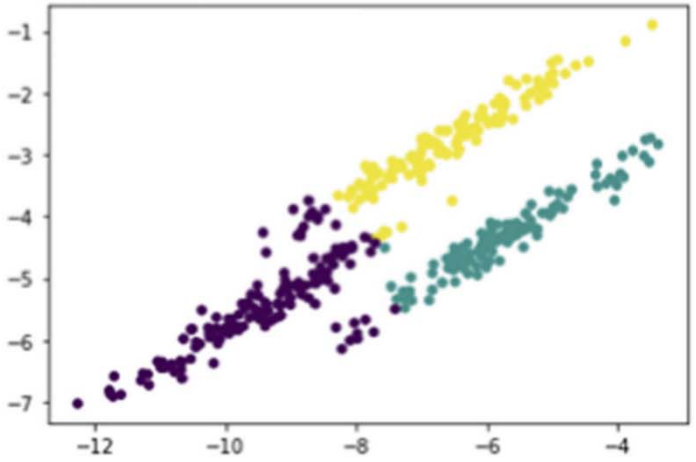
- 特征提取与选择：选择提取何种特征是聚类分析的基础，与问题需求直接相关。



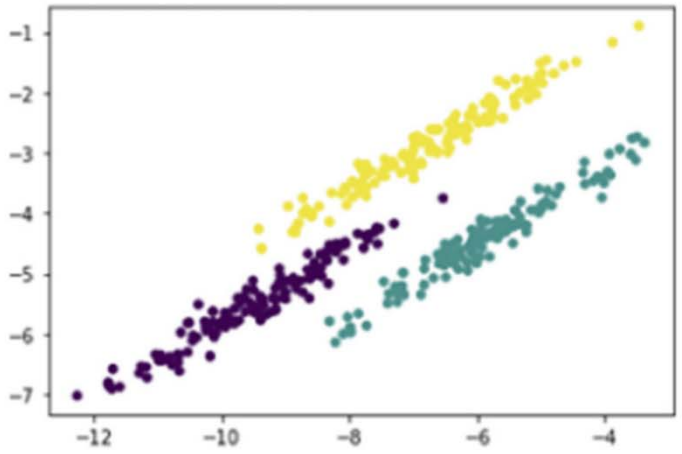
聚类分析的过程

- 相似性度量：聚类分析依赖距离（相似性）的度量，不同的距离测度有不同的聚类结果。
- 距离包括：（1）样本之间的距离，（2）样本与聚类之间的距离，（3）聚类之间的距离

基于欧氏距离的聚类结果



基于马氏距离的聚类结果

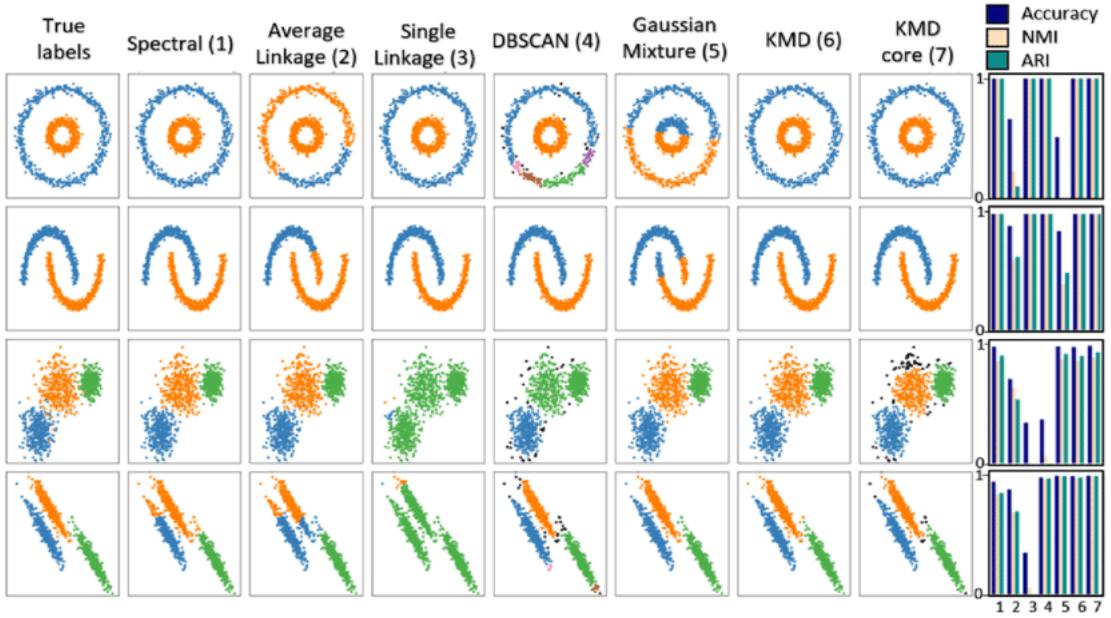


聚类分析的过程

- 聚类算法：按照某种准则无监督地将样本集合划分为若干簇/子类。
- 根据需求不同，聚类的输出结果可以是每个子类的中心、样本集中每个样本所属的子类标签或层次化的样本聚类结构。
- 常用聚类算法：
  - 基于连接的聚类：谱系聚类
  - 基于中心的聚类：K均值
  - 基于密度的聚类：DBSCAN
  - 基于分布的聚类：高斯混合模型

聚类分析的过程

- 聚类有效性检验：特征的选择、相似性度量的选择、算法的选择及其参数都会影响聚类结果；因此需要设定指标检验聚类结果是否准确合理。
- 有效性检验可以协助调整以上聚类分析中的各环节，并重新聚类以获得好结果。



不同聚类方法在不同数据集上的比较

聚类的数学描述

- 待聚类的样本集合 $D=\{\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n\}$ 中包含 n 个样本和 k 个子集/簇/类（k 可以预先设定，也可在聚类过程中确定）。
- k 个子类 $C_1,\cdots,C_k$ 需要满足三个条件：

1. $C_i \neq \varnothing, i=1,\cdots,k;$ 每一类都至少有一个样本
2. $\bigcup_{i=1}^{k} C_i = D;$ 任何一个样本都属于某一类
3. $C_i \cap C_j = \varnothing, i \neq j, i, j=1,\cdots,k$。 任何一个样本都只属于一个类

- 聚类的目标：类内样本距离小，类间样本距离大

聚类准则

- 类内距离准则：每个样本与其所属的类别中心间距离平方之和

$$J_W(C_1,\cdots,C_k)=\frac{1}{n}\sum_{j=1}^{k}\sum_{\boldsymbol{x}\in C_j}\left\|\boldsymbol{x}-\boldsymbol{m}_j\right\|^2,\ \text{其中}\ \boldsymbol{m}_j=\frac{1}{n_j}\sum_{\boldsymbol{x}\in C_j}\boldsymbol{x}$$

$\boldsymbol{m}_j$ 是第 j 类的样本中心，n_j 是第 j 类的样本量。

- 类间距离准则：每个类别中心到样本整体中心之间的加权距离平方之和

$$J_B(C_1,\cdots,C_k)=\sum_{j=1}^{k}\frac{n_j}{n}\left\|\boldsymbol{m}_j-\boldsymbol{m}\right\|^2,\ \text{其中}\ \boldsymbol{m}=\frac{1}{n}\sum_{\boldsymbol{x}\in D}\boldsymbol{x}$$

是样本整体中心。

聚类准则

- 类内、类间的距离平方和可以用样本的散布矩阵计算。
- 第 j 类的类内散布矩阵：$S_W^j = \frac{1}{n_j} \sum_{\boldsymbol{x} \in C_j} (\boldsymbol{x} - \boldsymbol{m}_j)(\boldsymbol{x} - \boldsymbol{m}_j)^T$
- 总的类内散布矩阵：$S_W(C_1, \cdots, C_k) = \sum_{j=1}^{k} \frac{n_j}{n} S_W^j$
- 类间散布矩阵：$S_B = \sum_{j=1}^{k} \frac{n_j}{n} (\boldsymbol{m}_j - \boldsymbol{m})(\boldsymbol{m}_j - \boldsymbol{m})^T$
- 可证明 $J_W(C_1, \cdots, C_k) = tr(S_W)$

  $J_B(C_1, \cdots, C_k) = tr(S_B)$

聚类准则

- 利用类内和类间散布矩阵可定义类内类间距离准则，综合考虑类内聚集程度和类间离散程度：

$$J_{WB}(C_1,\cdots,C_k)=tr(S_W^{-1}S_B)$$

- 基于准则函数，聚类可转化为优化问题：

$$\max_{C_1,\cdots,C_k} J_{WB}(C_1,\cdots,C_k)$$

- 但以上的优化问题难以求解，因此一般寻找近似最优解。

顺序聚类

- 顺序聚类：算法简单，无需设定聚类数目
  - 每次输入一个样本，计算该样本与当前已形成的各个类间的距离；
  - 如果所有距离都大于某个设定的阈值 θ，则生成一个新的类，否则加入最近的类；
  - 也可以设定最大聚类数 M，达到最大聚类数后不再新增聚类。

顺序聚类

- 顺序聚类需计算样本 $\boldsymbol{x}$ 与聚类C间的距离$d(\boldsymbol{x}, C)$：

1) 最大距离： $\boldsymbol{x}$ 与 C 中最远样本的距离

$$d(\boldsymbol{x}, C) = \max_{\boldsymbol{y} \in C} d(\boldsymbol{x}, \boldsymbol{y})$$

2) 最小距离： $\boldsymbol{x}$ 与 C 中最近样本的距离

$$d(\boldsymbol{x}, C) = \min_{\boldsymbol{y} \in C} d(\boldsymbol{x}, \boldsymbol{y})$$

3) 平均距离： $\boldsymbol{x}$ 与 C 中所有样本的距离平均值

$d(\boldsymbol{x}, C) = \frac{1}{n_C} \sum_{\boldsymbol{y} \in C} d(\boldsymbol{x}, \boldsymbol{y})$，其中$n_C$是聚类$C$的样本数

4) 中心距离： $\boldsymbol{x}$ 与 C 中样本均值间的距离

$d(\boldsymbol{x}, C) = d(\boldsymbol{x}, \boldsymbol{m}_C)$，其中 $\boldsymbol{m}_C = \frac{1}{n_C} \sum_{\boldsymbol{y} \in C} \boldsymbol{y}$是聚类$C$的均值

顺序聚类算法

- 初始化：第一个样本 $\boldsymbol{x}_1$ 作为第一个聚类，$C_1 = \{\boldsymbol{x}_1\}$ ，聚类数 $l = 1$，距离阈值θ，最大聚类数 M；
- 顺序输入每个训练样本 $\boldsymbol{x}_i$：
  - 计算 $\boldsymbol{x}_i$ 距离最近的类别C_k：$d(\boldsymbol{x}_i, C_k) = \min\limits_{1 \le j \le l} d(\boldsymbol{x}_i, C_j)$
  - 如果 $d(\boldsymbol{x}_i, C_k) > \theta$ 并且 $l < M$，则 $l = l + 1$，$C_l = \{\boldsymbol{x}_i\}$；
  - 否则 $C_k = C_k \bigcup \{\boldsymbol{x}_i\}$；
- 输出：聚类 $\{C_1, \cdots, C_l\}$，聚类数 l。

顺序聚类

- 例：用顺序聚类将下列样本分类，阈值 $\theta = 2.5$，最大聚类数 $M = 5$，样本间使用欧氏距离，样本与聚类间的相似度以中心距离度量。

$\boldsymbol{x}_1 = (1, 1)^T$

$\boldsymbol{x}_2 = (4, 5)^T$

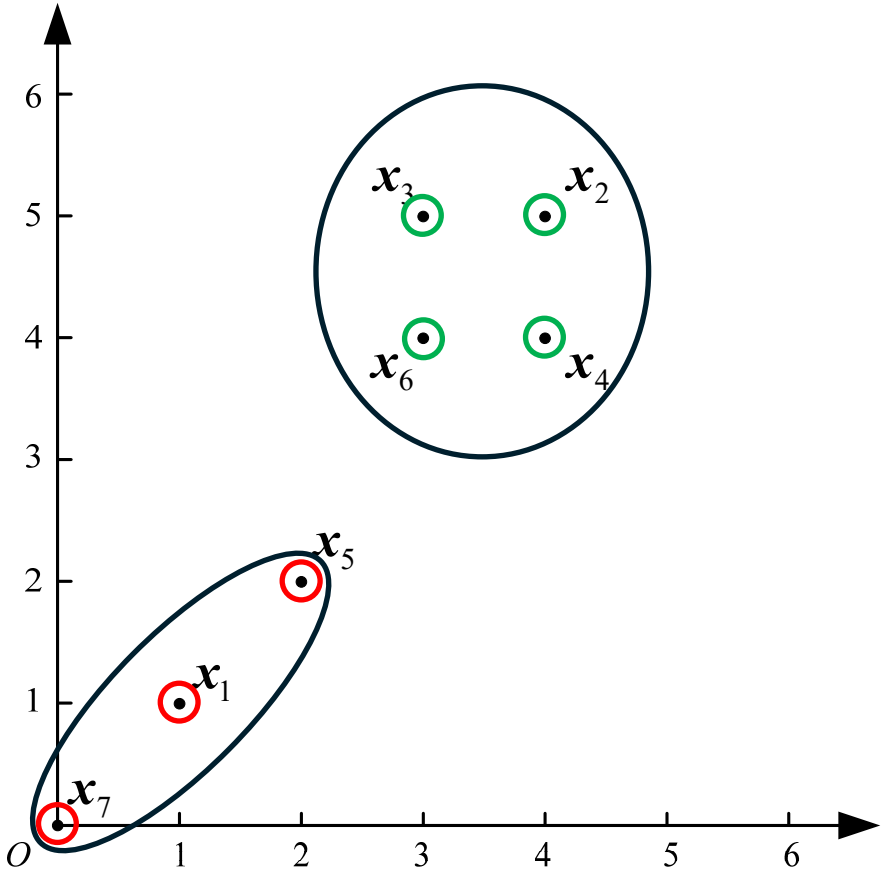
$\boldsymbol{x}_3 = (3, 5)^T$

$\boldsymbol{x}_4 = (4, 4)^T$

$\boldsymbol{x}_5 = (2, 2)^T$

$\boldsymbol{x}_6 = (3, 4)^T$

$\boldsymbol{x}_7 = (0, 0)^T$



顺序聚类

- 顺序聚类的结果受样本输入顺序的影响：

$\boldsymbol{x}_1 = (1, 1)^T$

$\boldsymbol{x}_2 = (4, 5)^T$

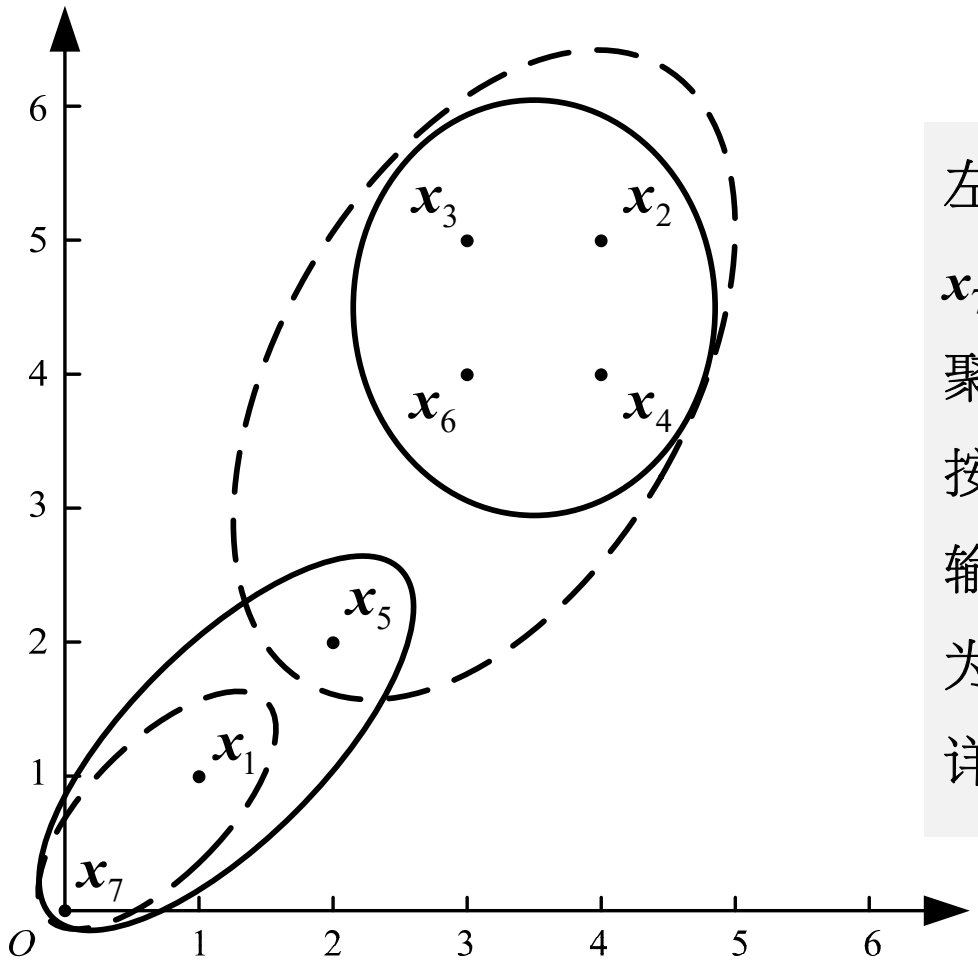
$\boldsymbol{x}_3 = (3, 5)^T$

$\boldsymbol{x}_4 = (4, 4)^T$

$\boldsymbol{x}_5 = (2, 2)^T$

$\boldsymbol{x}_6 = (3, 4)^T$

$\boldsymbol{x}_7 = (0, 0)^T$

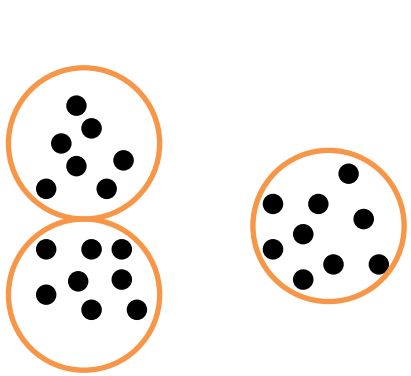


左例中，按照从 $\boldsymbol{x}_1$ 到 $\boldsymbol{x}_7$ 的顺序输入样本，聚类结果为实线圈；按照 $\boldsymbol{x}_7$ 到 $\boldsymbol{x}_1$ 的顺序输入样本，聚类结果为虚线圈。

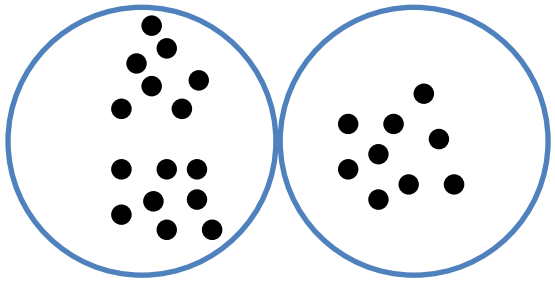
详见44−45页例3.1。

顺序聚类

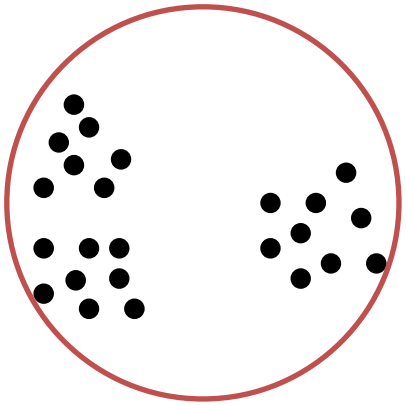
- 顺序聚类的结果受距离阈值参数 θ 的影响：



较小阈值



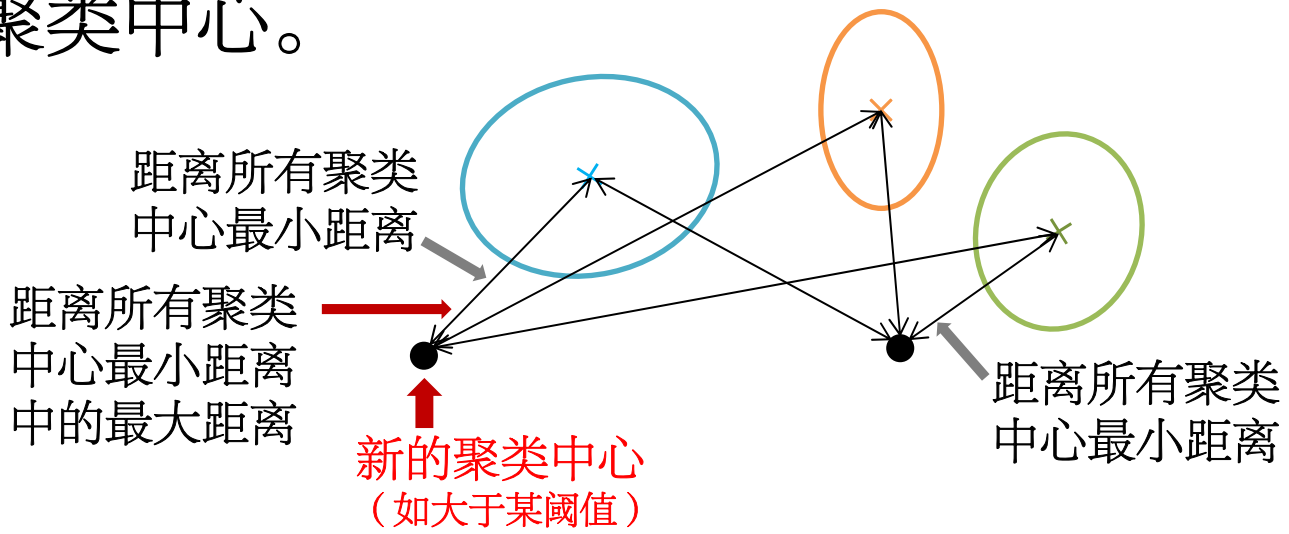
适中阈值



较大阈值

最大最小距离聚类

- 最大最小距离聚类（Max-Min Distance）：
  每次循环中寻找距离当前所有聚类中心最远的样本（每个训练样本距离所有聚类中心最小距离中的最大距离），如果该样本与最近的聚类中心之间的距离大于一定的阈值，则增加此样本为一个新的聚类中心。



最大最小距离聚类

- 确定聚类数量和聚类中心：
  - 初始化：第一个样本 $\boldsymbol{x}_1$ 作为第一个聚类中心，$\boldsymbol{m}_1 = \boldsymbol{x}_1$；
  - 寻找距离 $\boldsymbol{m}_1$ 最远的样本作为第二个聚类中心

    $$i_{\max} = \arg\max_{1 \le i \le n} d(\boldsymbol{x}_i, \boldsymbol{m}_1),\ \boldsymbol{m}_2 = \boldsymbol{x}_{i_{\max}},\ l = 2\ ;$$
  - 循环，直到没有新的聚类中心产生为止：
    - 计算每个样本 $\boldsymbol{x}_i$ 到当前 l 个聚类中心的距离，寻找最小值

      $$d_i = \max_{1 \le k \le l} d(\boldsymbol{x}_i, \boldsymbol{m}_k),$$
    - 寻找所有样本到聚类中心最小距离中的最大距离：

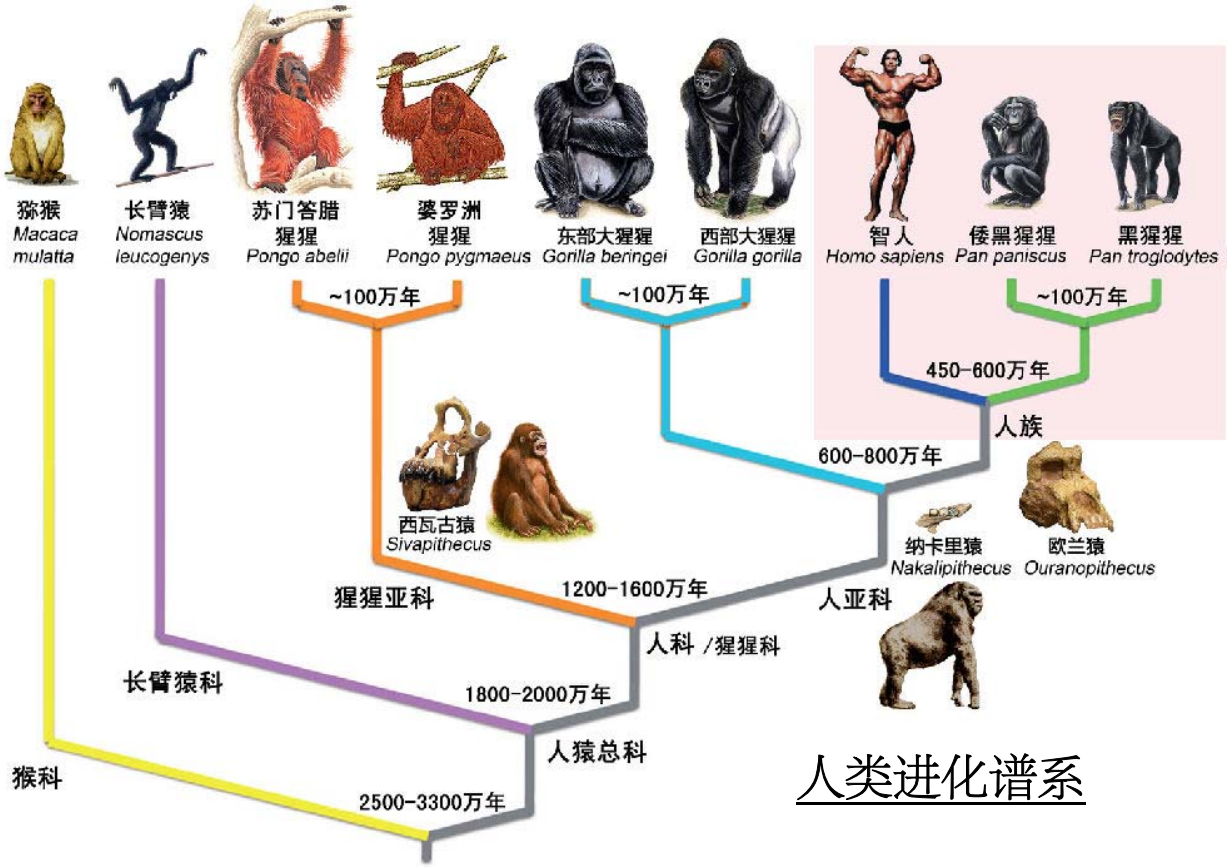
      $$d_{\max} = \max_{1 \le i \le n} d_i,\ i_{\max} = \arg\max_{1 \le i \le n} d_i$$
    - 如果 $d_{\max} > \theta \| \boldsymbol{m}_1 - \boldsymbol{m}_2 \|$，则产生新的聚类中心 $\boldsymbol{m}_{l+1} = \boldsymbol{x}_{i_{\max}}$，$l = l + 1$
- 分类训练样本：
  - 初始化各个聚类：$C_k = \varnothing,\ 1 \le k \le l$
  - 顺序输入每个训练样本 $\boldsymbol{x}_i$：
    - 计算 $\boldsymbol{x}_i$ 距离最近的聚类：$k = \arg\min_{1 \le t \le l} d(\boldsymbol{x}_i, \boldsymbol{m}_t)$
    - 分类 $\boldsymbol{x}_i$：$C_k = C_k \cup \{\boldsymbol{x}_i\}$
- 输出：聚类 $\{C_1, \cdots, C_l\}$，聚类数 l。

最大最小距离聚类

- 顺序聚类中样本的分类和新的聚类产生过程同时进行；最大最小距离聚类中样本的分类和新的聚类产生在两个过程中进行。
- 最大最小距离聚类的结果只和第一个聚类中心及阈值有关，可以缓解顺序聚类受样本顺序影响的缺陷。
- 最大最小距离聚类算法的计算量较大。

谱系聚类

- 谱系聚类（层次聚类）：不仅产生出样本的不同聚类，而且要生成一个完整的层次分类谱系图（Dendrogram）。



人类进化谱系

谱系聚类

- 谱系聚类算法分为两类：合并法和分裂法。
- 合并法：初始将每个样本作为一类，每一轮迭代选择最相近的两类合并，经过若干轮后将所有样本合并为一类。
- 分裂法：首先将所有样本作为一类，每一轮迭代选择一个现有聚类分裂为两类，经过若干轮后将每个样本单独分为一类。

谱系聚类合并算法

- 初始化：每个样本作为单独一类，$C_i = \{\boldsymbol{x}_i\}$，$i = 1, \cdots, n$；
- 循环，直到所有样本属于一个聚类为止：
  - 寻找当前聚类中最相近的两个聚类：

$$d(C_i, C_j) = \min_{r,s} d(C_r, C_s)$$

  - 删除聚类 C_i 和 C_j，增加新的聚类 $C_q = C_i \bigcup C_j$。
- 输出：样本的合并过程，形成层次化谱系。

谱系聚类合并算法

- 将所有样本聚为一类并无意义，一般设立某些终止条件以提前输出合并聚类的结果。
- 常见的终止条件
  - 预定类别数：设定一个目标聚类数，合并过程中达到预定类别数时停止。
  - 距离阈值：设定一个距离阈值，当最近两类的距离大于该阈值，则停止合并。
  - 最优聚类数：根据某种判断聚类结果的准则（下节课介绍）确定最优的聚类数。

谱系聚类合并算法

- 每一轮迭代选择合并的依据是类间的相似程度。
- 常用的聚类之间的距离度量

前文已定义的距离包括：
样本间距离，样本和类之间距离

  - 最大距离法：两类间相距最远的两个样本之间的距离

$$d(C_i, C_j) = \max_{\boldsymbol{x}\in C_i, \boldsymbol{y}\in C_j} d(\boldsymbol{x}, \boldsymbol{y})$$

  - 最小距离法：两类间相距最近的两个样本之间的距离

$$d(C_i, C_j) = \min_{\boldsymbol{x}\in C_i, \boldsymbol{y}\in C_j} d(\boldsymbol{x}, \boldsymbol{y})$$

  - 平均距离法：两类间任意一对样本距离的平均值

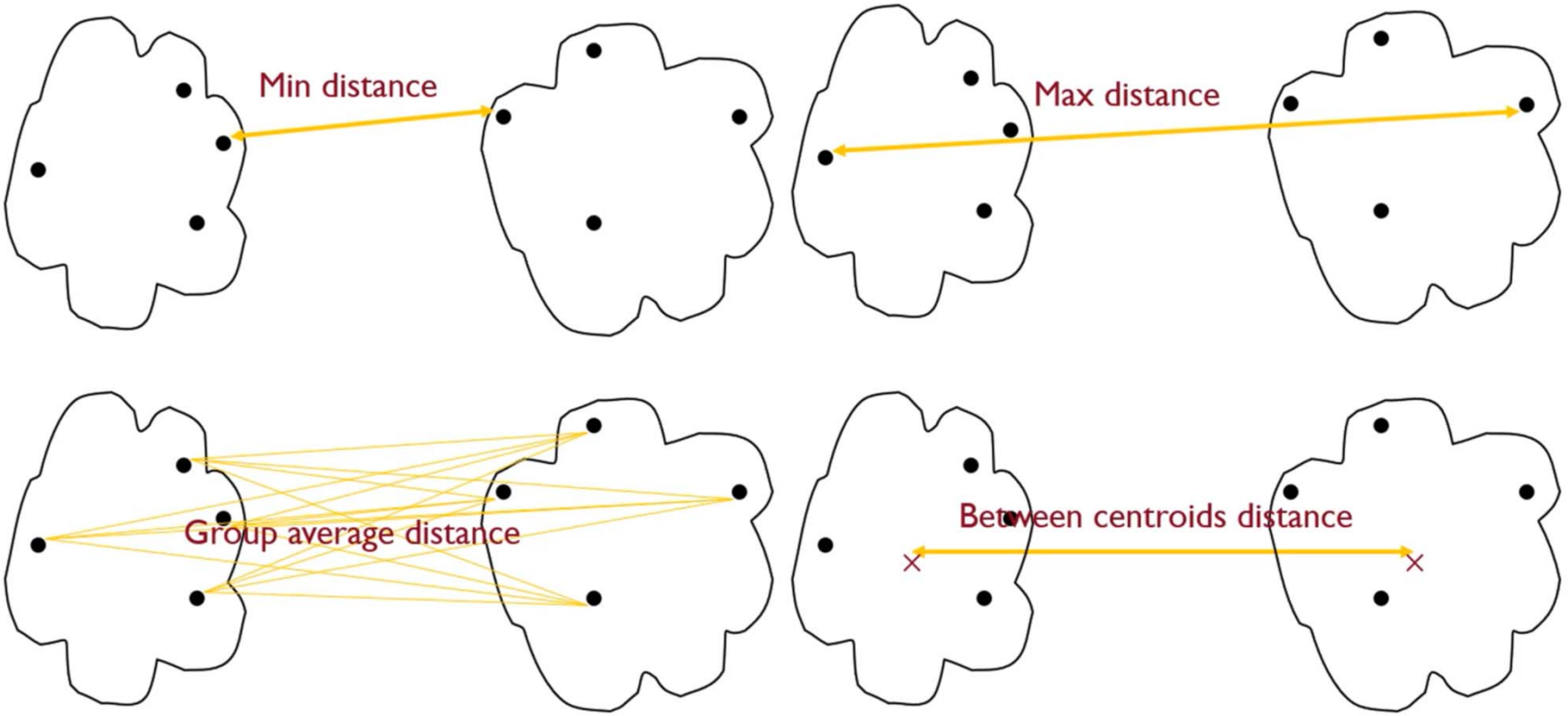
$d(C_i, C_j) = \dfrac{1}{n_i n_j} \sum_{\boldsymbol{x}\in C_i, \boldsymbol{y}\in C_j} d(\boldsymbol{x}, \boldsymbol{y})$，其中 n_i 和 n_j 是两类的样本数

  - 平均样本法：两类样本均值之间的距离

$d(C_i, C_j) = d(\boldsymbol{m}_i, \boldsymbol{m}_j)$，其中 $\boldsymbol{m}_i$ 和 $\boldsymbol{m}_j$ 是两类的样本均值

谱系聚类合并算法

- 常用的聚类之间的距离度量



谱系聚类合并算法

- 谱系聚类算法在第 k 轮合并前需要计算 $n-k+1$ 个聚类间的距离，生成整个谱系需要 n 轮。所以总的距离计算次数是 $(n^3-n)/6$。 运算量极大

- 距离计算可以缩减：

  1) 除了被合并的聚类之外，其他聚类之间的距离没有变化，只需重新计算与新生成聚类有关的距离；

  2) 新生成的聚类与原有聚类的距离可以由被合并的两个聚类与其他聚类间的距离进行推算（注意：不同距离度量有不同推算方式，见课本49-50页）。

谱系聚类合并算法

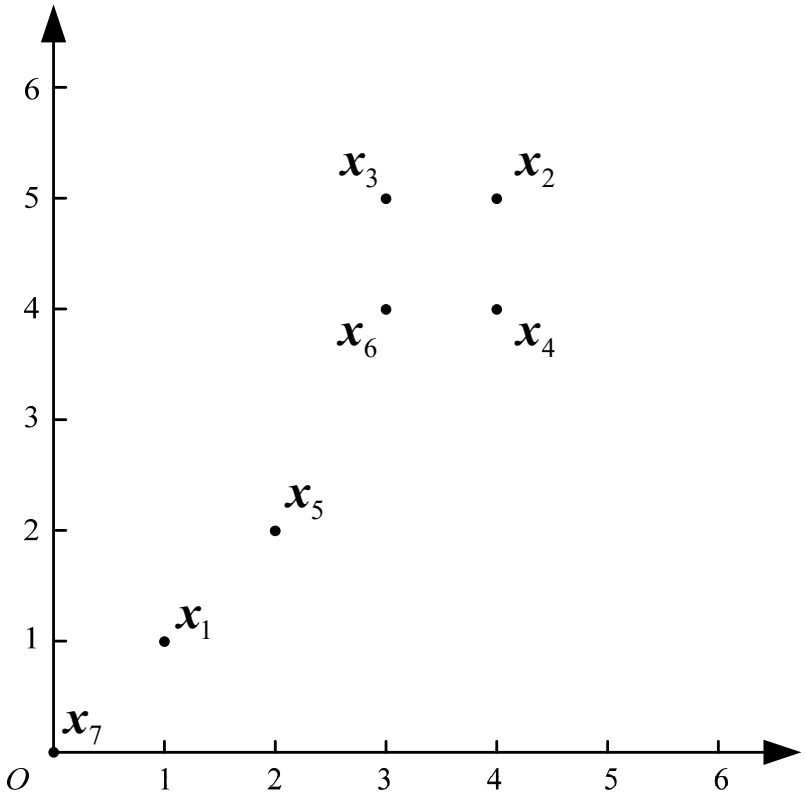
- 初始化：每个样本作为单独一类，$C_i = \{\boldsymbol{x}_i\}$，$i = 1, \cdots, n$ ，每个聚类的样本数 $n_i = 1$，计算任意两个样本间的距离，构成距离矩阵 $\boldsymbol{D} = (D_{i,j} = d(\boldsymbol{x}_i, \boldsymbol{x}_j))_{n \times n}$ ，聚类数 $l = n$；
- 循环，直到满足聚类终止条件为止：
  - 寻找距离矩阵 $\boldsymbol{D}$ 中上三角矩阵元素的最小值 D_{ij}；
  - 删除聚类 C_i 和 C_j，增加新的聚类 $C_q = C_i \cup C_j$，$n_q = n_i + n_j$，$l = l - 1$；
  - 更新距离矩阵 $\boldsymbol{D}$：
    - 最大距离： $D_{kq} = D_{qk} = \max(D_{ik}, D_{jk})$
    - 最小距离： $D_{kq} = D_{qk} = \min(D_{ik}, D_{jk})$
    - 平均距离： $D_{kq} = D_{qk} = \dfrac{n_i}{n_i + n_j} D_{ik} + \dfrac{n_j}{n_i + n_j} D_{jk}$
    - 平均样本法： $D_{kq} = D_{qk} = \sqrt{\dfrac{n_i}{n_i + n_j} D_{ki}^2 + \dfrac{n_j}{n_i + n_j} D_{kj}^2 - \dfrac{n_i n_j}{(n_i + n_j)^2} D_{ij}^2}$
- 输出：聚类 $\{C_1, \cdots, C_l\}$，聚类数 l。

谱系聚类的特点

- 谱系聚类的复杂度主要是初始计算距离矩阵时的 $n(n-1)/2$ 次距离计算，后续迭代计算量较小。
- 谱系聚类的优点：结果与先后次序无关；除聚类结果外，也可以产生谱系聚类过程和聚类结构。
- 谱系聚类的缺点：当 n 较大时，运算和存储量仍较大；两个样本一旦被合并在一个聚类中之后，不会再被分开。

谱系聚类合并法

- 例：用谱系聚类将下列样本聚为两类，样本间使用曼哈顿距离，聚类间距离使用最小距离。



$$x_1 = (1, 1)^T$$
$$x_2 = (4, 5)^T$$
$$x_3 = (3, 5)^T$$
$$x_4 = (4, 4)^T$$
$$x_5 = (2, 2)^T$$
$$x_6 = (3, 4)^T$$
$$x_7 = (0, 0)^T$$

1	7	6	6	2	5	2
	2	1	1	5	2	9
		3	2	4	1	8
			4	4	1	8
				5	3	4
					6	7
						7

谱系聚类合并算法

第1轮

1	7	6	6	2	5	2
	2	1	1	5	2	9
		3	2	4	1	8
			4	4	1	8
				5	3	4
					6	7
						7

第2轮

1	6	6	2	5	2
	2,3	1	4	1	8
		4	4	1	8
			5	3	4
				6	7
					7

第3轮

1	6	2	5	2
	2,3,4	4	1	8
		5	3	4
			6	7
				7

第4轮

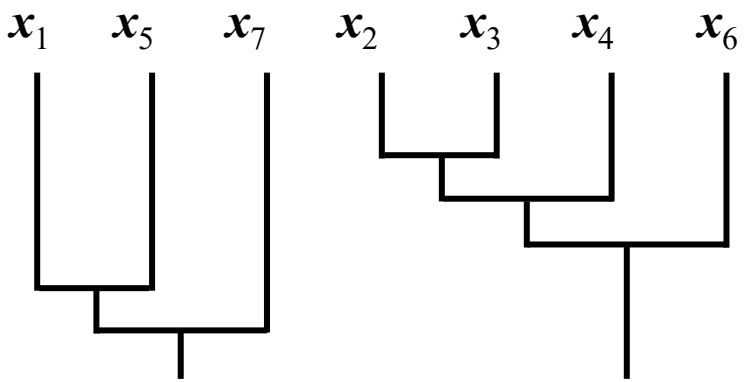
1	5	2	2
	2,3,4,6	3	7
		5	4
			7

第5轮

1,5	3	2
	2,3,4,6	7
		7

第6轮

1,5,7	3
	2,3,4,6



谱系聚类分裂算法

- 分裂算法初始将所有样本作为一类，然后用一个最优方式将一个聚类分为两类；每一轮都照此选择最优方式，从所有分裂中选择最优者，将此聚类按照最优方式分为两类。每轮增加一个聚类，n 轮后得到 n 个样本（或满足某条件后停止）。
- 分裂法计算量很大，较少使用。
- 分裂法可以和其他聚类法结合，将每一聚类按某聚类法分为两类。

K-均值聚类

- K-均值聚类（K-means）：最经典和最常用的聚类算法之一。
- K-均值聚类的目标是将 n 个样本依据最小化类内距离的准则分到 K 个聚类中：

$$\min_{C_1,\cdots,C_K} J_W(C_1,\cdots,C_K)=\frac{1}{n}\sum_{j=1}^{K}\sum_{\boldsymbol{x}\in C_j}\|\boldsymbol{x}-\boldsymbol{m}_j\|^2$$

其中 $\boldsymbol{m}_j=\frac{1}{n_j}\sum_{\boldsymbol{x}\in C_j}\boldsymbol{x}$ 是第 j 类的均值，n_j 是第 j 类的样本数。

- 直接优化求解以上类内距离准则有困难。

K–均值聚类

- K–均值的难度之一在于无法预先知道聚类均值。
- 如果已知聚类均值，则为了最小化类内距离，对于每一个样本 $\boldsymbol{x}$，显然应将 $\boldsymbol{x}$ 加入一个聚类 C_j，其中

$$j = \arg\min_{1 \le i \le K} \| \boldsymbol{x} - \boldsymbol{m}_i \|^2$$

- K–均值算法的基本思想：首先假设每类均值的猜想值 $\hat{\boldsymbol{m}}_1, \cdots, \hat{\boldsymbol{m}}_K$，根据均值猜想值确定每个样本的类别 $\hat{C}_1, \cdots, \hat{C}_K$；然后逐步迭代估计均值和确定样本类别，直到结果收敛。

K–均值聚类

- 初始化：随机选择 K 个聚类均值 $\boldsymbol{m}_j,\ j=1,\cdots,K$；
- 循环，直到 K 个均值都不再变化为止：
  - $C_j=\varnothing,\ j=1,\cdots,K$；
  - for $i=1$ to n

$$k=\arg\min_{1\le j\le K}\|\boldsymbol{x}_i-\boldsymbol{m}_j\|,\ \ C_k=C_k\bigcup\{\boldsymbol{x}_i\}$$

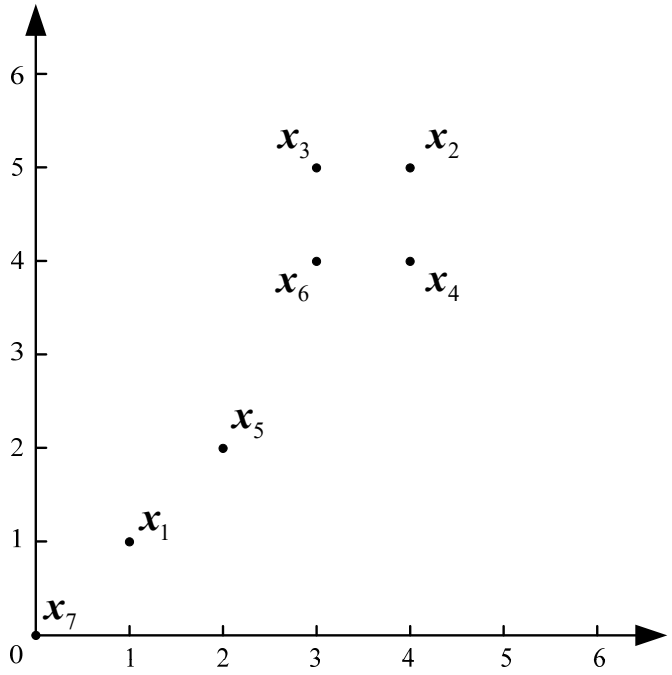
  - end for
  - 更新 K 个聚类的均值：$\boldsymbol{m}_j=\dfrac{1}{n_j}\sum_{\boldsymbol{x}\in C_j}\boldsymbol{x},\ j=1,\cdots,K$
- 输出：聚类 $\{C_1,\cdots,C_K\}$。

K–均值聚类

- 例：使用K均值聚类将下列样本聚成两个类别，选择 $\boldsymbol{x}_6$ 和 $\boldsymbol{x}_7$ 作为初始聚类均值，使用欧式距离

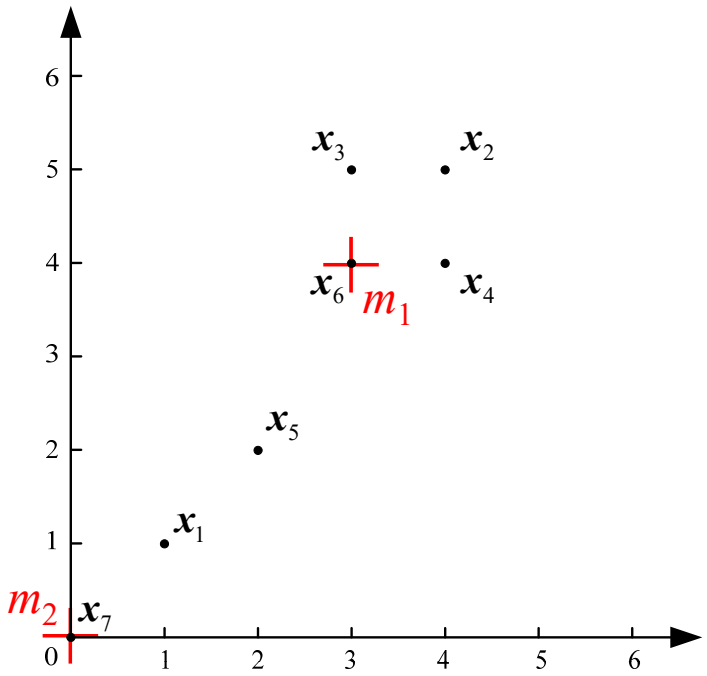
$$\boldsymbol{x}_1 = (1, 1)^T, \boldsymbol{x}_2 = (4, 5)^T, \boldsymbol{x}_3 = (3, 5)^T, \boldsymbol{x}_4 = (4, 4)^T,$$

$$\boldsymbol{x}_5 = (2, 2)^T, \boldsymbol{x}_6 = (3, 4)^T, \boldsymbol{x}_7 = (0, 0)^T$$



K–均值聚类

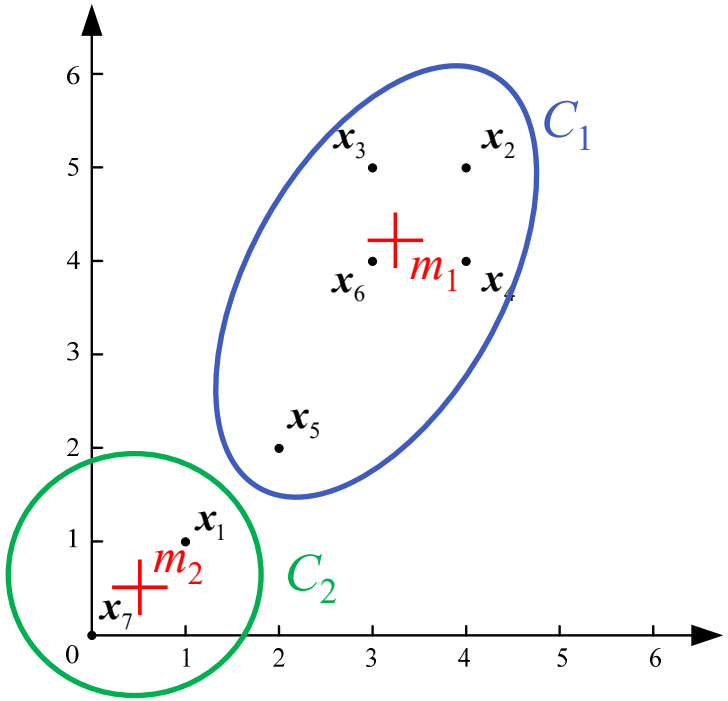
- 例：使用K均值聚类将下列样本聚成两个类别，选择 $\boldsymbol{x}_6$ 和 $\boldsymbol{x}_7$ 作为初始聚类均值，使用欧式距离
  - 初始：　$m_1 = (3, 4)^T, m_2 = (0, 0)^T$



K-均值聚类

- 例：使用K均值聚类将下列样本聚成两个类别，选择 $\boldsymbol{x}_6$ 和 $\boldsymbol{x}_7$ 作为初始聚类均值，使用欧式距离
  - 第一轮： $C_1=\{\boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5, \boldsymbol{x}_6\}$, $C_2=\{\boldsymbol{x}_1, \boldsymbol{x}_7\}$

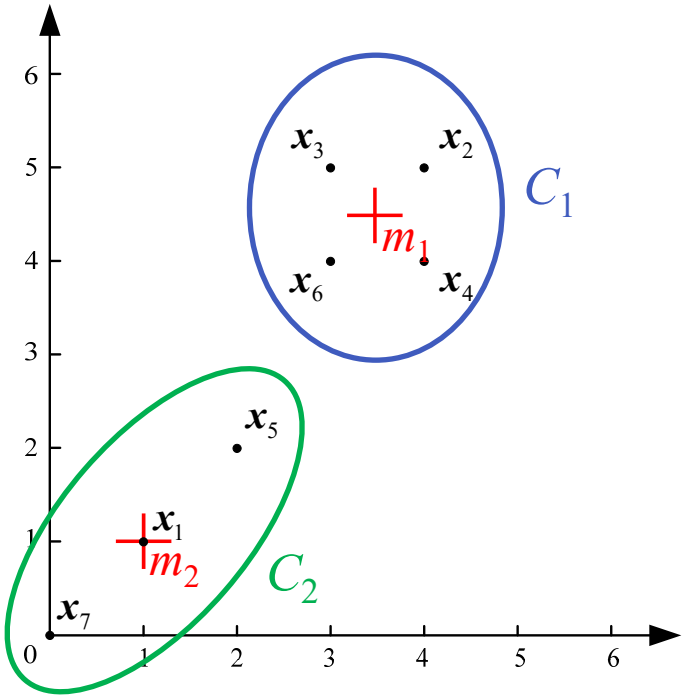
$$m_1 = (3.25, 4.25)^T, m_2 = (0.5, 0.5)^T$$



K-均值聚类

- 例：使用K均值聚类将下列样本聚成两个类别，选择 $\boldsymbol{x}_6$ 和 $\boldsymbol{x}_7$ 作为初始聚类均值，使用欧式距离
  - 第二轮： $C_1=\{\boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_6\}$, $C_2=\{\boldsymbol{x}_1, \boldsymbol{x}_7, \boldsymbol{x}_5\}$

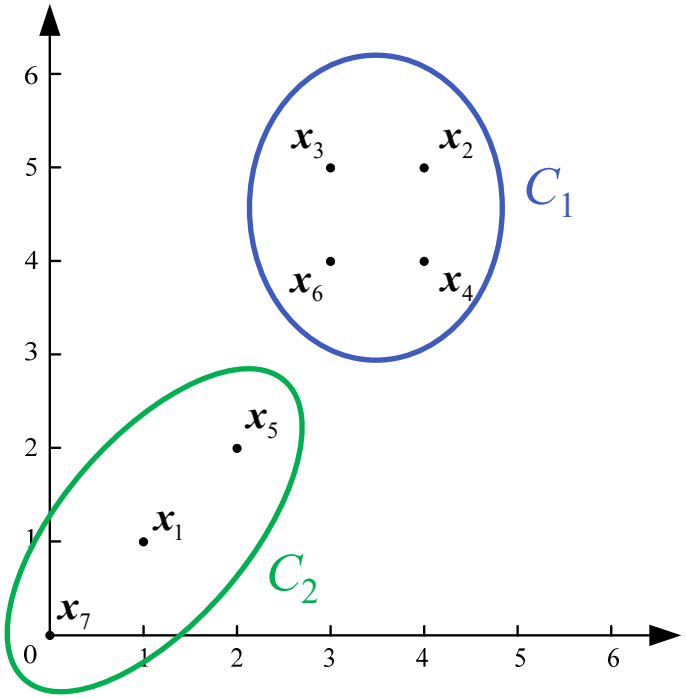
$$m_1 = (3.5, 4.5)^T, m_2 = (1, 1)^T$$



K-均值聚类

- 例：使用K均值聚类将下列样本聚成两个类别，选择 $\boldsymbol{x}_6$ 和 $\boldsymbol{x}_7$ 作为初始聚类均值，使用欧式距离
  - 第三轮： $C_1=\{\boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_6\}$, $C_2=\{\boldsymbol{x}_1, \boldsymbol{x}_7, \boldsymbol{x}_5\}$

不变，停止循环。

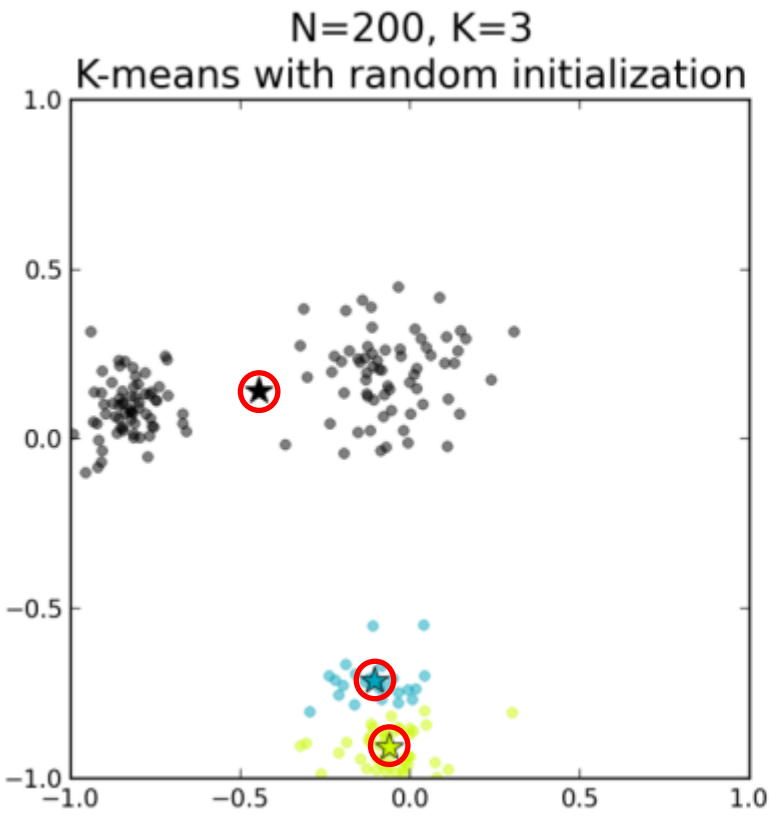


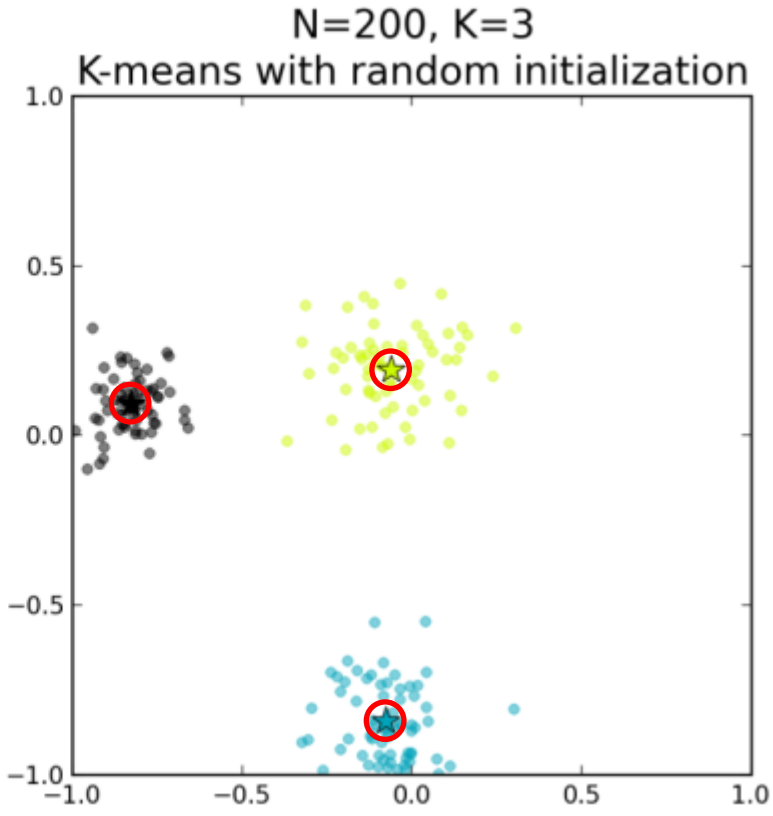
K–均值聚类

- K–均值聚类的优点：实现简单，计算复杂度低
  - 算法经过 m 次迭代收敛，需要 $m \times K \times n$ 次的样本与均值间的距离计算；一般 m 和 K 远小于 n，因此计算复杂度是 $O(n)$，远小于谱系聚类的复杂度 $O(n^2)$ 。
- K–均值聚类的问题：
  - 尽管算法收敛，但不保证收敛的解是准则的最小值；不同的初始值选择会收敛到不同的局部极小值。
  - 聚类数 K 必须预先设定。

K-均值聚类

- 初始值选择对K-均值聚类的影响：

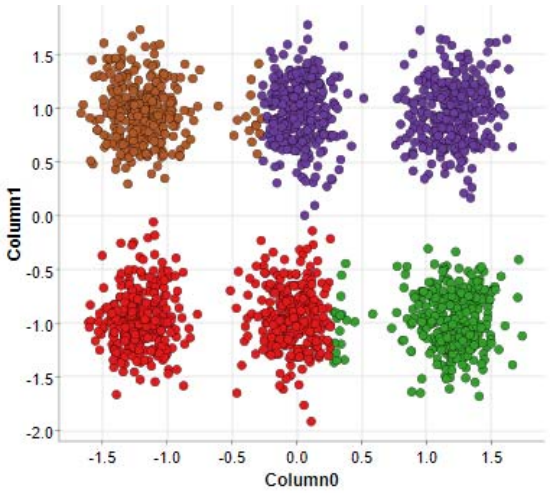




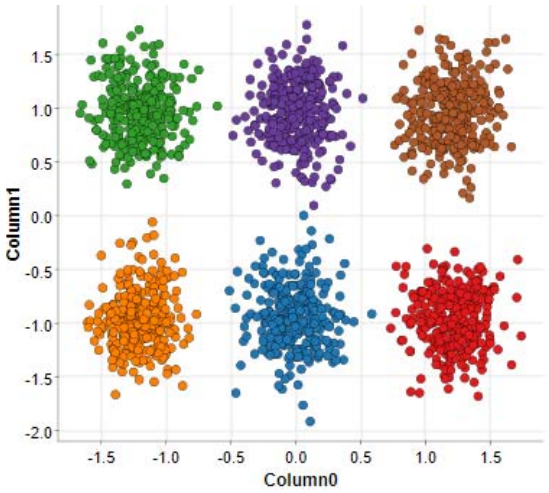
K-均值聚类

- 聚类数选择对K-均值聚类的影响：

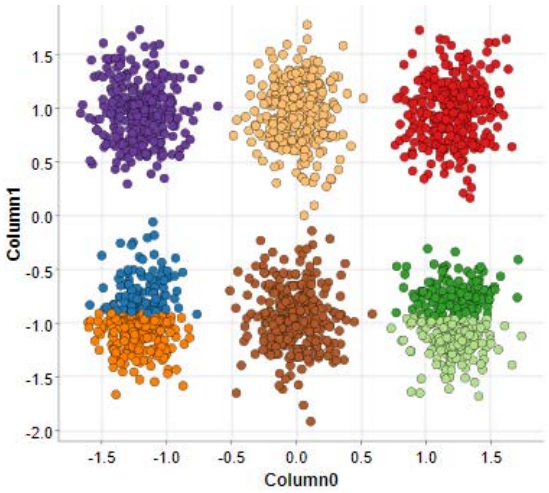
$K = 4$



$K = 6$



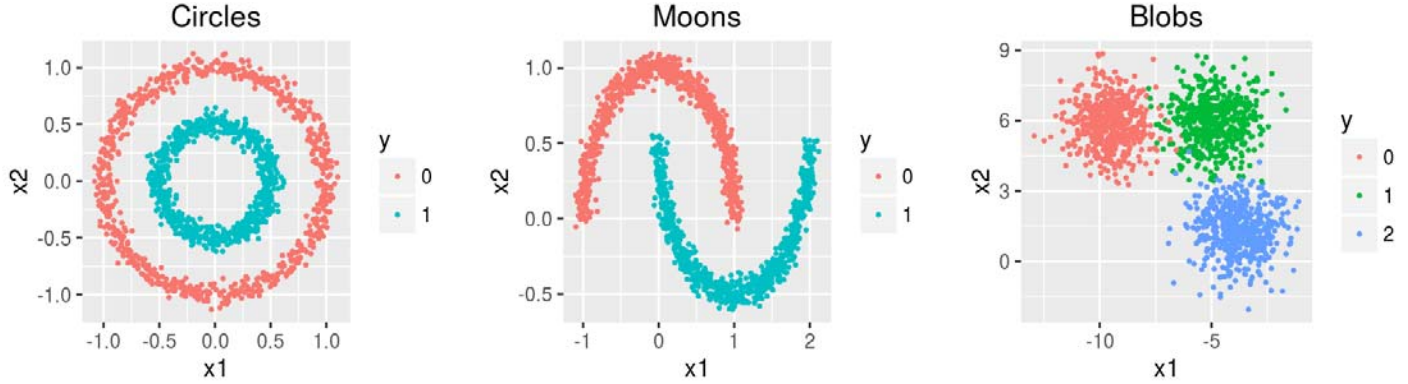
$K = 8$



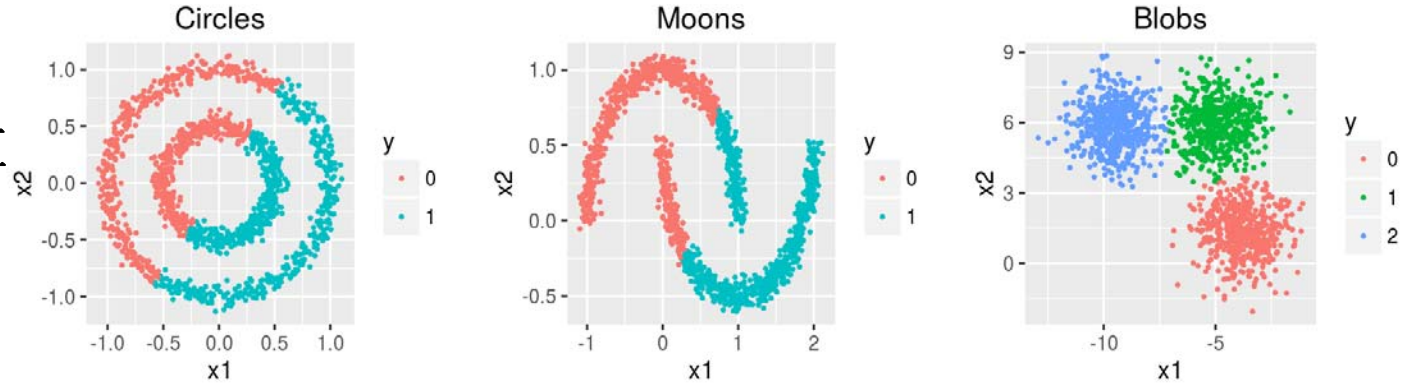
K-均值聚类

- K-均值聚类的最优解是以类内距离准则为优化目标，因此最优解不能保证是聚类的最优。
  - 如果样本类内聚集性好，大致成团形（高斯）分布，各类方差接近，则K-均值结果较好；否则效果欠佳。

真实类别

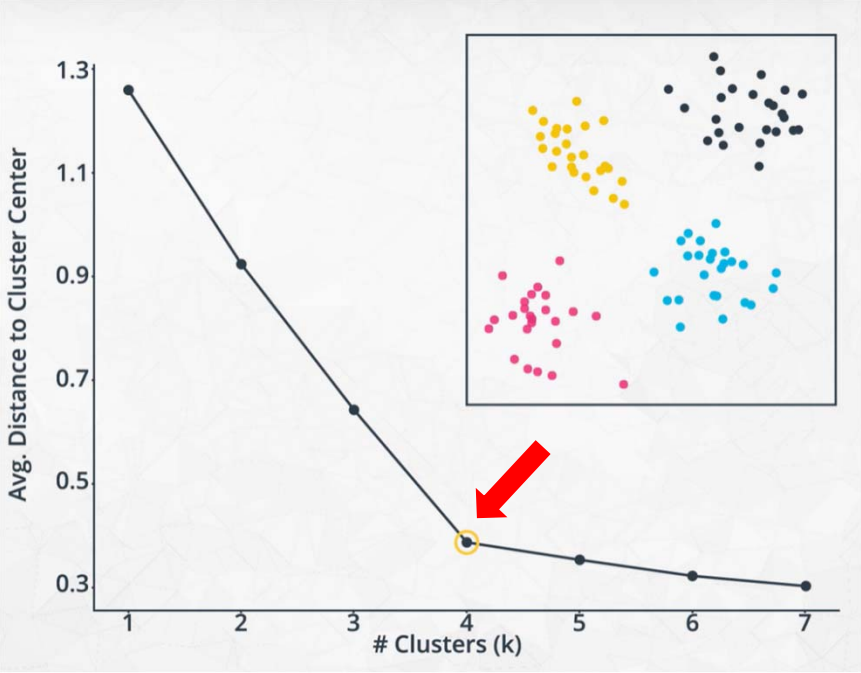


K-均值结果



K-均值的改进

- 样本中包含的聚类数尚无最优方法可以确定。
- 一种常用的方法是尝试一系列从小到大的聚类数目，由K-均值算法得到相应聚类结果，根据聚类有效性检验选出最合适的聚类数和聚类结果。



K-均值的改进

- K-均值的改进可通过有技巧地选择初始值实现

1) 根据先验知识（聚类样本的结构，如各个聚类的大概位置）设定初始的聚类均值。

2) 将样本随机划分K个聚类之后再计算初始值。

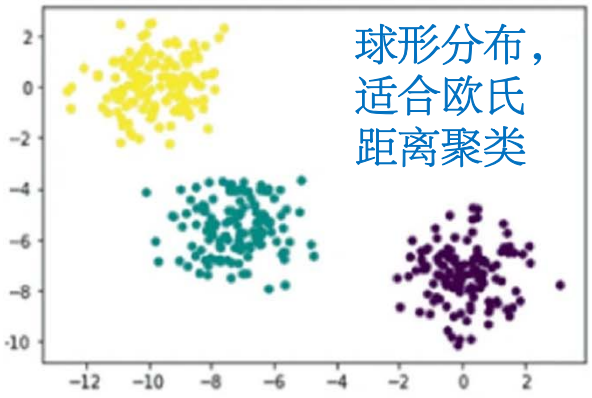
3) 选择相互之间距离最远的K个样本（可以用类似于最大最小距离法的方式），因为这些样本处于不同类的可能性较大。

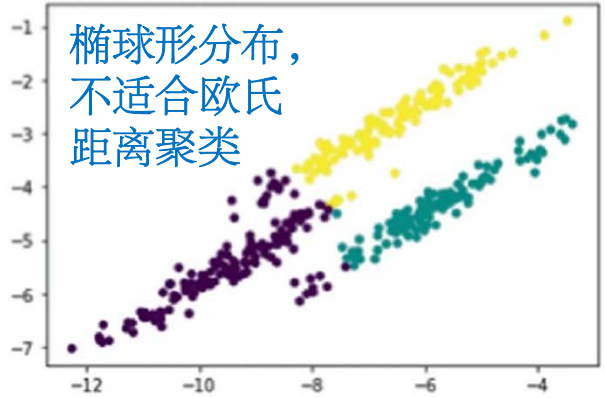
K-均值的改进

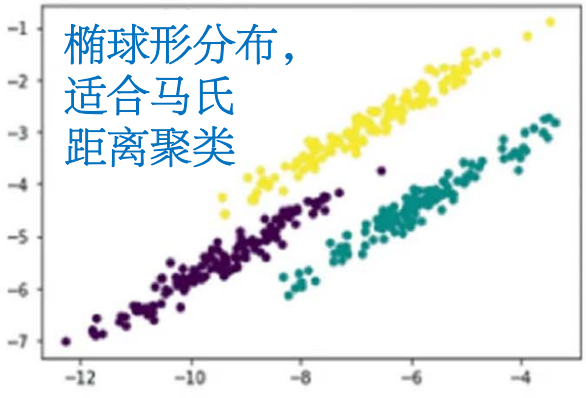
- K-均值的改进可以通过改变距离函数实现。
- K-中值（K-medians）：计算类内距离准则时采用曼哈顿距离取代欧氏距离；K-中值算法可以更好地对抗离群点的影响。
- 使用不同距离度量时，描述聚类的参数不同。K-中值算法中，描述聚类的参数是中值，每轮迭代时需计算每一维特征的中值。

K-均值的改进

- K-均值一般基于欧氏距离，因此要求每个聚类的样本大致呈团形（高斯分布）。
- 样本呈现其他分布时，需考虑其他距离度量方式（例如，椭球形分布样本适合采用马氏距离）。



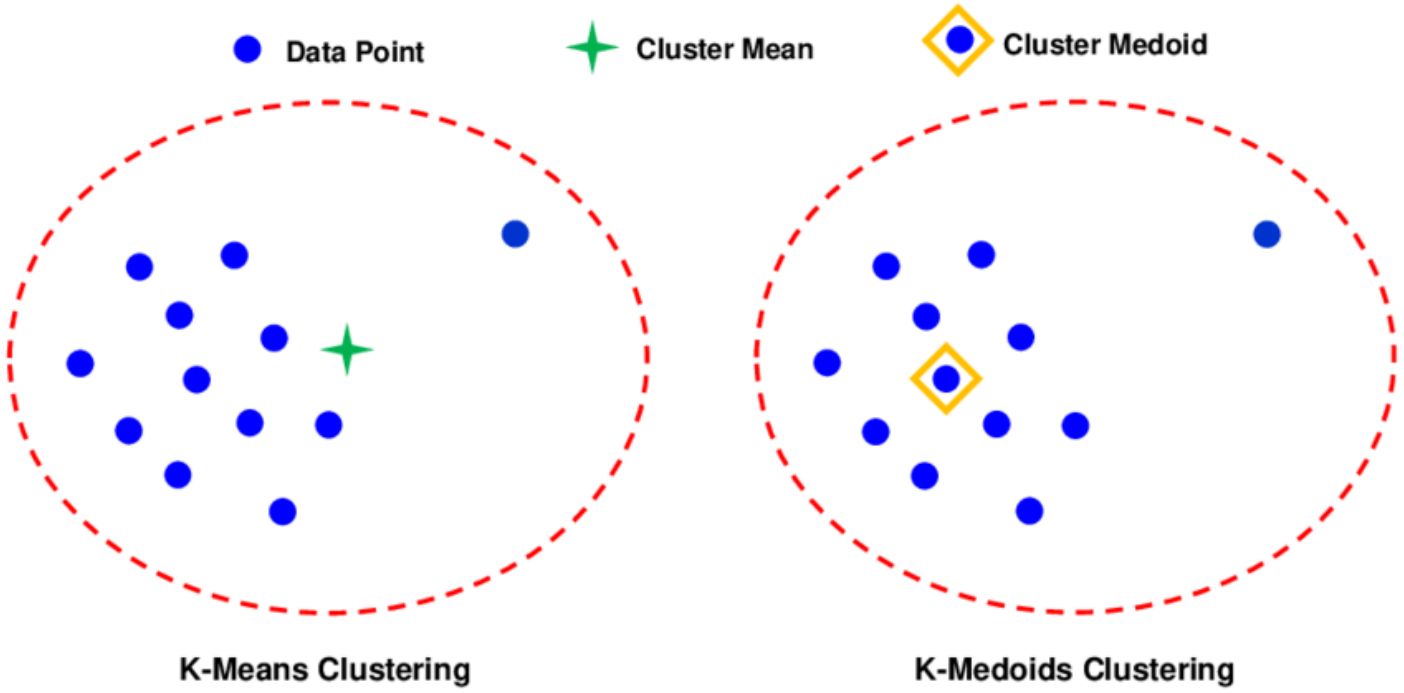




- 注意，基于马氏距离的K-均值算法需要的参数是每个聚类的均值和协方差矩阵。

K-均值的改进

- K-medoids（K-中心点）：与K-均值和K-中值类似，主要区别是，描述和代表每个聚类的不是均值或中值，而是某个样本（中心点），该样本与其他该类中的样本相似度之和最大。

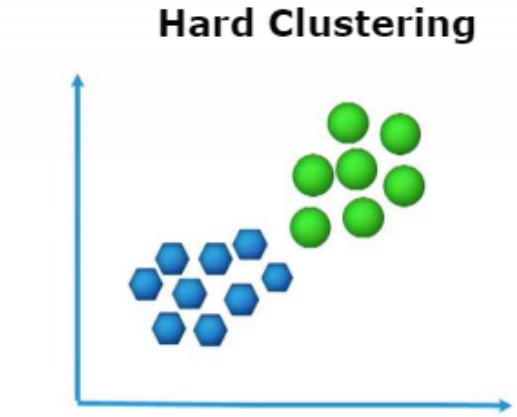


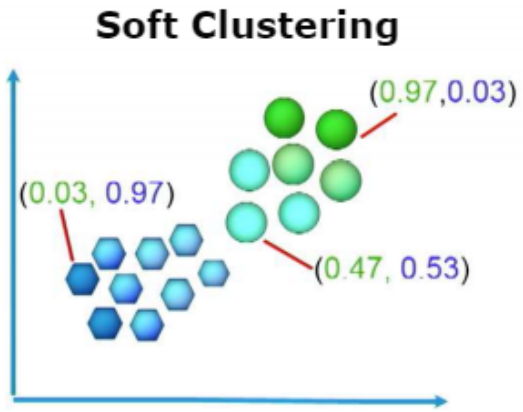
K-均值的改进

	K-均值 K-means	K-中值 K-medians	K-中心点 K-medoids
每一聚类的代表	样本均值	样本中值	类内某一个样本，它与类内其他样本的相似度之和最大
聚类优化准则	最小化样本与聚类代表之间的欧氏距离	最小化样本与聚类代表之间的曼哈顿距离	最大化样本与聚类代表之间的相似度（更通用）
特点	• 简单易算 • 对噪声和离群点敏感	• 对噪声和离群点稳健 • 计算较复杂	• 对噪声和离群点稳健 • 计算较复杂 • 结果易解释

模糊C-均值

- 需要聚类的样本集在各个聚类间未必能够严格区分，很多情况下聚类间存在交叠。
- 模糊C-均值（Fuzzy C-means，FCM）：迭代中不采用“硬分类”，而采用“模糊分类”。
  - 硬分类：严格将每个样本分到某个聚类；
  - 模糊分类/软分类：样本可能属于任何聚类，只是属于的程度不同。





模糊C-均值

- 模糊分类中，认为 $\boldsymbol{x}_i$ 属于 $C_1,\cdots,C_K$ 中的任何一个聚类，只是属于的程度不同，一般可以用隶属度 u_{ij} 表示 $\boldsymbol{x}_i$ 属于 C_j 的程度。
- 硬分类时： $u_{ij}=u_j(\boldsymbol{x}_i)=\begin{cases}1, & \boldsymbol{x}_i\in C_j\\ 0, & \boldsymbol{x}_i\notin C_j\end{cases}$
- 模糊C-均值优化的聚类准则函数是

$$J_{WF}(\boldsymbol{m}_1,\cdots,\boldsymbol{m}_K,u_{11},\cdots,u_{nK})=\frac{1}{n}\sum_{j=1}^{K}\sum_{i=1}^{n}u_{ij}^{b}\,\|\boldsymbol{x}-\boldsymbol{m}_j\|^2$$

约束条件为 $\sum_{j=1}^{K}u_{ij}=1,\ 0\le u_{ij}\le 1$，其中 $b>1$ 是控制不同聚类混合程度的可调参数。

模糊C-均值

- 上述优化问题可以由拉格朗日乘子法解决。
（延伸阅读：https://blog.csdn.net/einsdrw/article/details/37930331）
- 当聚类均值 $\boldsymbol{m}_j$ 固定时，隶属度的最优解是

$$u_{ij} = \frac{(1/\| \boldsymbol{x}_i - \boldsymbol{m}_j \|^2)^{1/(b-1)}}{\sum_{k=1}^{K} (1/\| \boldsymbol{x}_i - \boldsymbol{m}_k \|^2)^{1/(b-1)}}, \ i = 1, \cdots, n; \ j = 1, \cdots, K \quad \text{(公式*)}$$

- 当隶属度 u_{ij} 固定时，聚类均值的最优解是

$$\boldsymbol{m}_j = \frac{\sum_{i=1}^{n} u_{ij}^b \boldsymbol{x}_i}{\sum_{i=1}^{n} u_{ij}^b}, \ j = 1, \cdots, K \quad \text{(公式\#)}$$

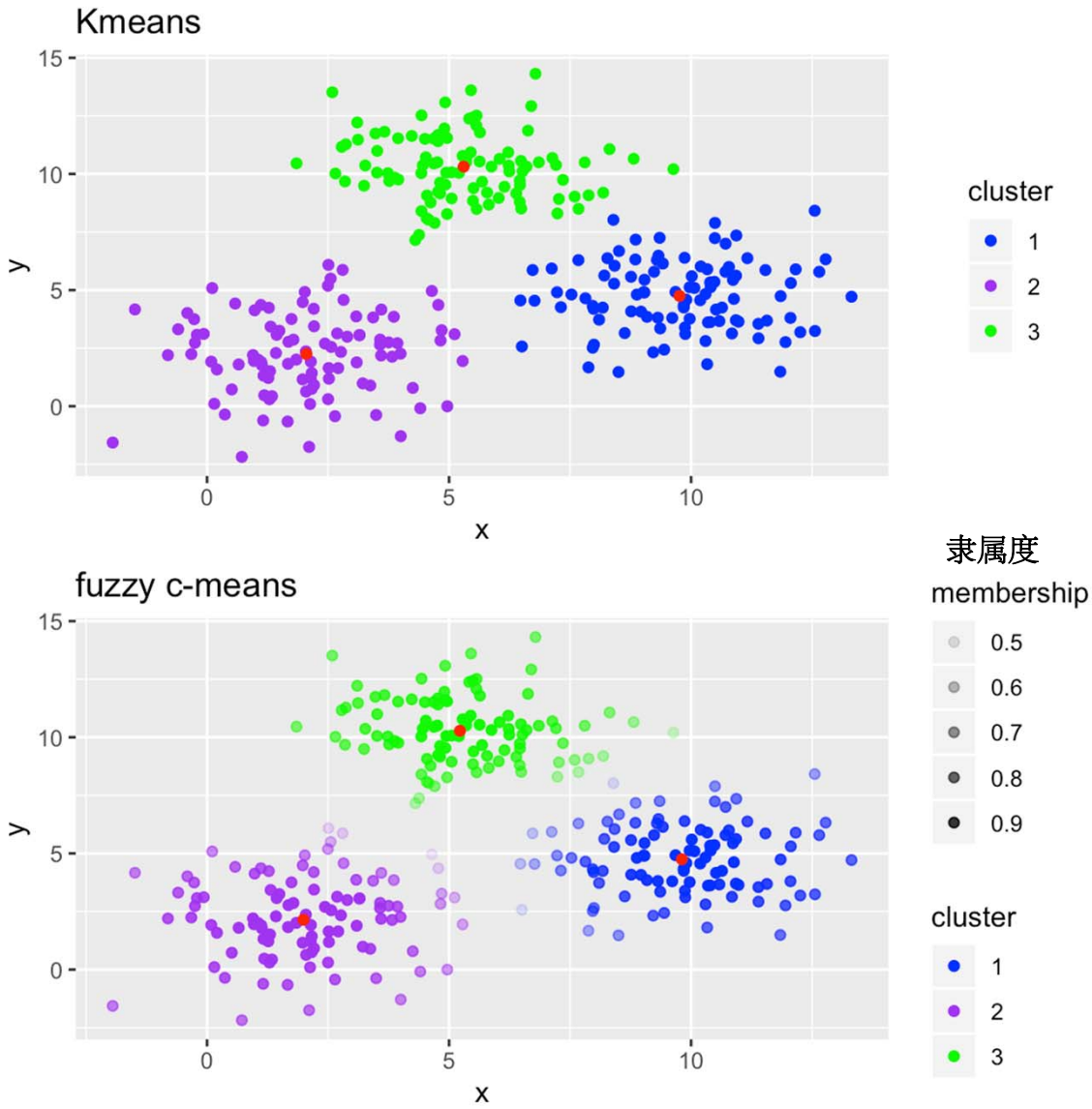
模糊C-均值

- FCM算法

- 初始化：随机选择 K 个聚类均值 $\boldsymbol{m}_j$，$j=1,\cdots,K$;
- 循环，直到两次迭代的隶属度变化很小为止：
  - 使用上页(公式*)计算每个样本对于每个聚类的隶属度
  - 使用上页(公式#)更新每个聚类的均值
- 输出：样本集的隶属度 $\{u_{ij}\}$, $i=1,\cdots,n$; $j=1,\cdots,K$。

➢ 可以设定一个容忍误差阈值，当两次迭代的隶属度的差值之和小于阈值，则终止迭代。

模糊C-均值



如果希望输出明确的聚类，可将样本划分到隶属度最大的聚类。

聚类检验

- 不同的算法、不同的参数、不同的初始值有不同的聚类结果。如何判断哪一个是最好的结果？
- 首先考虑聚类数相同情况下，如何检验不同初始条件得到的结果？
  - K-均值算法中聚类均值初始值不同，
  - 顺序聚类中样本顺序不同，
  - 最大最小距离算法中第一个样本选择不同，等。
- 聚类数相同情况下，一般定义某种聚类有效性准则函数以检验不同聚类结果的有效性。

聚类结果的检验

- Dunn指数：所有聚类中最近的两个聚类之间的距离与所有聚类的最大直径之比：

$$J_{Dunn}(C_1,\cdots,C_K)=\frac{\min\limits_{i,j=1,\cdots,K,i\neq j} d(C_i,C_j)}{\max\limits_{k=1,\cdots,K} diam(C_k)}$$

其中，聚类之间的距离定义为两类间最近一对样本的距离

$$d(C_i,C_j)=\min_{\boldsymbol{x}\in C_i,\boldsymbol{y}\in C_j} d(\boldsymbol{x},\boldsymbol{y})$$

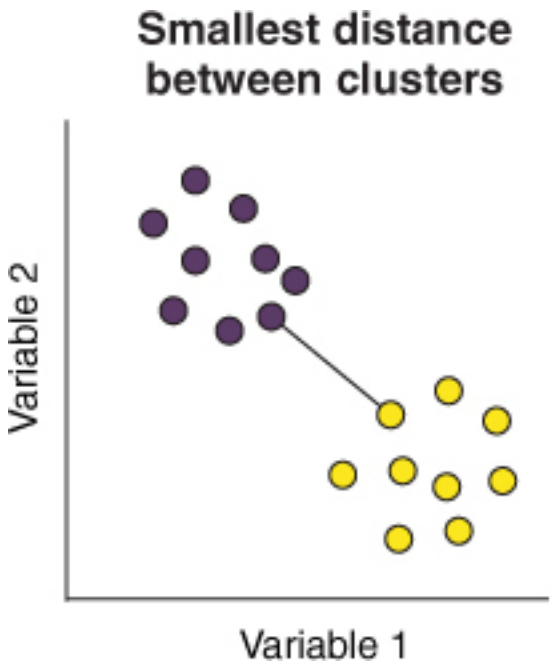
聚类样本集的直径定义为样本内距离最远的两个样本间的距离

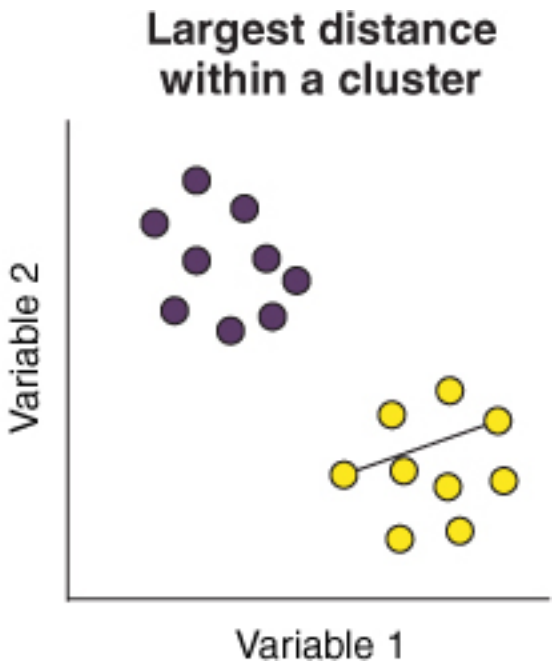
$$diam(C_i)=\max_{\boldsymbol{x},\boldsymbol{y}\in C_i} d(\boldsymbol{x},\boldsymbol{y})$$

聚类结果的检验

- Dunn指数越大表示结果越好。

$$J_{Dunn}(C_1,\cdots,C_K)=\frac{\min\limits_{i,j=1,\cdots,K,i\neq j} d(C_i,C_j)}{\max\limits_{k=1,\cdots,K} diam(C_k)}$$





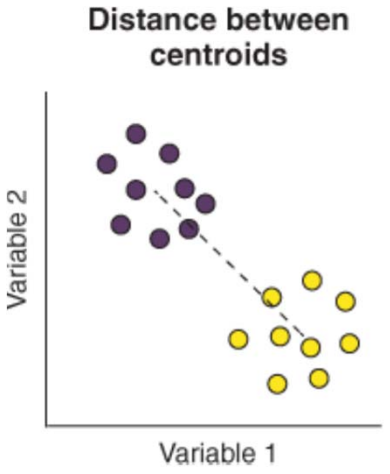
$$d(C_i,C_j)=\min_{\boldsymbol{x}\in C_i,\boldsymbol{y}\in C_j} d(\boldsymbol{x},\boldsymbol{y})$$

$$diam(C_i)=\max_{\boldsymbol{x},\boldsymbol{y}\in C_i} d(\boldsymbol{x},\boldsymbol{y})$$

聚类结果的检验

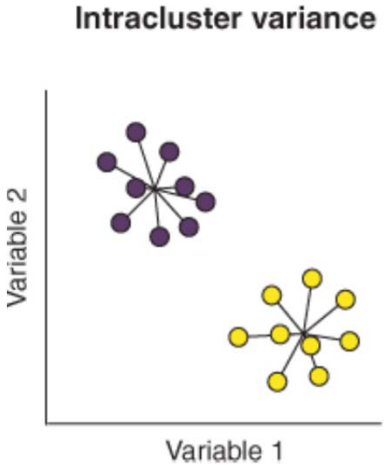
- Davies-Bouldin指数：同时考虑两类间的离散度和两类自身样本的离散度。
- 两个聚类之间的离散度可以用聚类均值之间的距离度量

$$d_{ij} = \| \boldsymbol{m}_i - \boldsymbol{m}_j \|$$



- 一个聚类的离散度可以用样本到聚类均值之间的均方距离度量

$$s_i = \sqrt{\frac{1}{n_i} \sum_{\boldsymbol{x} \in C_i} \| \boldsymbol{x} - \boldsymbol{m}_i \|^2}$$



聚类结果的检验

- 两个聚类之间的相似度为两个聚类自身的离散度值之和与两类之间的离散度之比

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}$$

- Davies-Bouldin指数为每个聚类与其他聚类之间最大相似度的平均值

$$J_{DB}(C_1, \cdots, C_K) = \frac{1}{K} \sum_{i=1}^{K} \max_{j=1,\cdots,K, i \neq j} R_{ij}$$

- Davies-Bouldin指数越小表明结果越好。

聚类结果的检验

- 如何利用准则函数（如Dunn指数或Davies-Bouldin指数）确定最佳的聚类效果?

  1. 在聚类数目一致的情况下，设置不同的初始条件分别进行聚类；
  2. 得到多个聚类结果；
  3. 选择某个准则函数评价各个聚类结果；
  4. 以最优者作为最终的聚类结果。

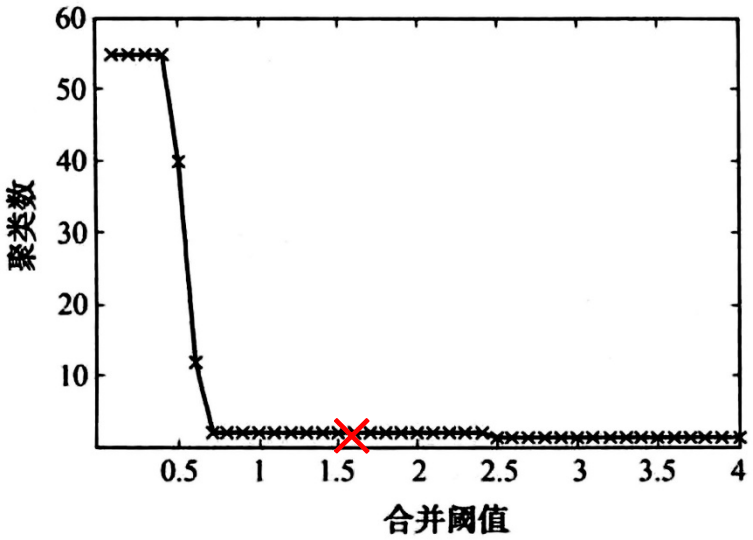
聚类数的间接选择

- 通过算法参数的设置可以间接设置聚类数目。
- 适用范围：
  - 谱系聚类：设定被合并的两个聚类之间的距离阈值作为终止条件，阈值越大则聚类数越少。
  - 顺序聚类：设定将样本合并到最近聚类的距离阈值 θ，θ越小则聚类数越多。
  - 最大最小距离聚类：确定聚类中心时比较当前的最大最小距离和 $\theta\|\boldsymbol{m}_1 - \boldsymbol{m}_2\|$以决定是否产生新的聚类，因此，$\theta$越小则聚类数越多。

聚类数的间接选择

- 通过算法参数的选择间接确定聚类数量：

1. 在可能的取值范围内设置不同的参数，并得到相应的聚类结果；
2. 建立参数与聚类数之间的对应关系；
3. 选择聚类数相同的最大参数区域；
4. 以该区域的中点作为最优参数，以对应的聚类数为最优聚类数。



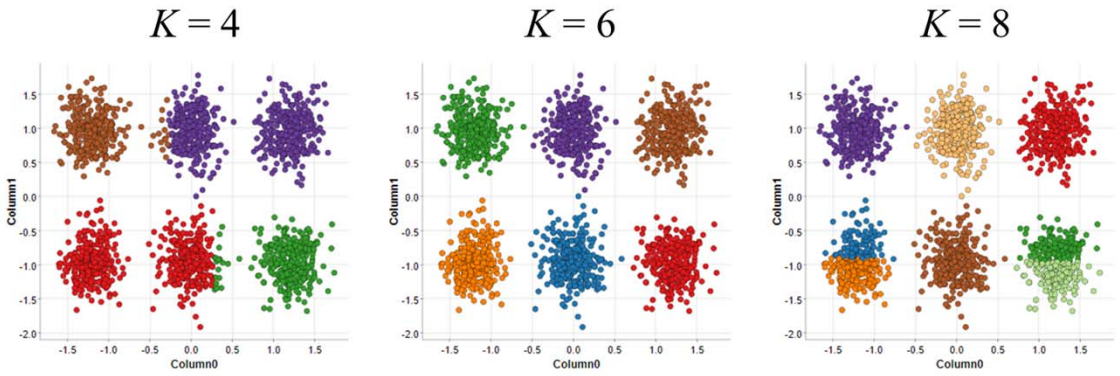
- 谱系聚类算法不同阈值得到的不同聚类数；
- 阈值在0.7-2.4之间出现较多，对应的聚类数是2；
- 选择上图中红叉点作为最优参数和聚类数。

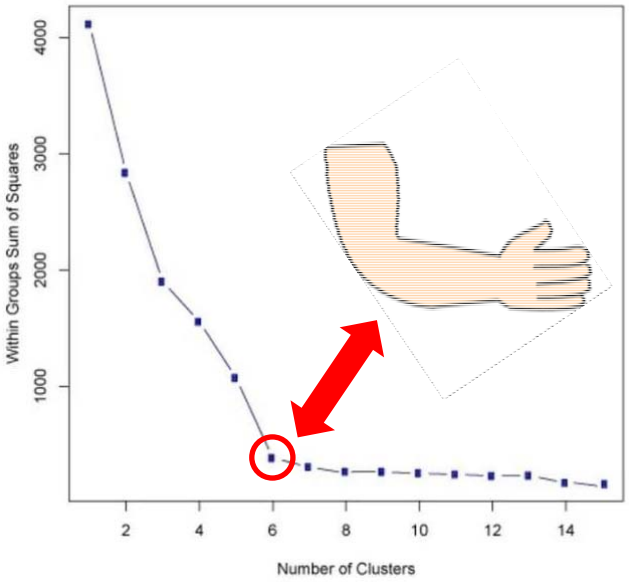
聚类数的直接选择

- 基于特定的聚类检验准则，可以根据不同聚类数的聚类结果直接找到最优的聚类数。
  1. 选择适合的聚类检验准则；
  2. 在可能的范围内逐一尝试不同的聚类数，并产生不同的聚类结果；
  3. 应用准则函数计算每个聚类结果的评价值；
  4. 如果准则函数有最优的极值（最大值或最小值），则选择对应最优值的聚类数目；*（很少出现）*
  5. 如果准则函数是聚类数目的单调函数，可以选择准则函数“拐点”处的聚类数目。

聚类数的直接选择

- 通过准则函数“拐点”寻找最优聚类数目的方法也叫“肘部法”。





- 如果样本集没有明显聚类，则准则函数较平滑，没有拐点。

聚类数的直接选择

- K-均值算法中，聚类结果由聚类数目和初始值联合确定。最优的聚类数目和初始值可以用以下方法确定：
  1. 选择适合的聚类检验准则（一般是类内距离准则，例如类内样本到聚类中心的距离平方和）；
  2. 在可能的范围内设定一系列聚类数；
  3. 对于每一个聚类数，尝试不同初始条件，从中选择可达到最优准则结果的初始条件，并将对应的最优结果作为该聚类数目下的聚类检验准则结果；
  4. 做出以聚类数目为自变量的准则函数，利用肘部法选择最优的聚类数目。

本章小结

- 介绍了聚类分析的原理、一般过程和问题描述
- 介绍了简单的聚类方法（包括顺序聚类和最大最小距离聚类）的算法和特点
- 介绍了谱系聚类（主要是合并法）的基本原理、算法和特点

本章小结

- 介绍了最常用的K-均值聚类方法
- 介绍了K-均值方法的种种改进，包括K中值、K中心点和模糊C均值
- 介绍了对聚类结果进行检验的准则函数（Dunn指数和Davies-Bouldin指数）
- 介绍了如何利用参数设置间接选择聚类数
- 介绍了如何利用肘部法直接选择聚类数