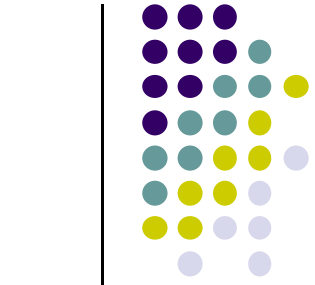
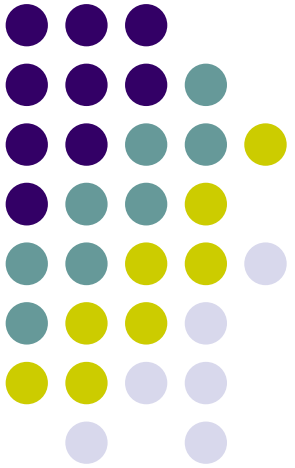


聚类问题

哈尔滨工业大学计算学部 刘远超



聚类问题

本章内容简介

- 什么是聚类
- 聚类算法评价指标
- 常见的聚类算法

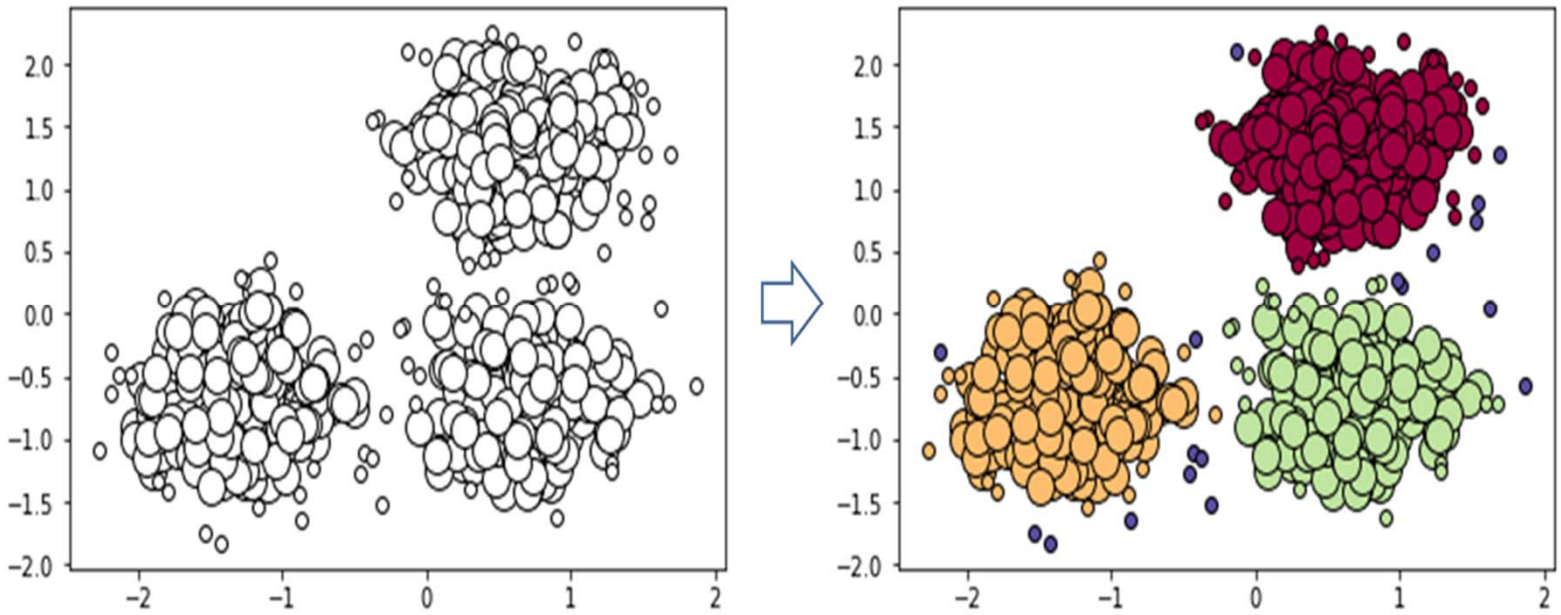
聚类问题

本章内容简介

- **什么是聚类**
- 聚类算法评价指标
- 常见的聚类算法

什么是聚类?

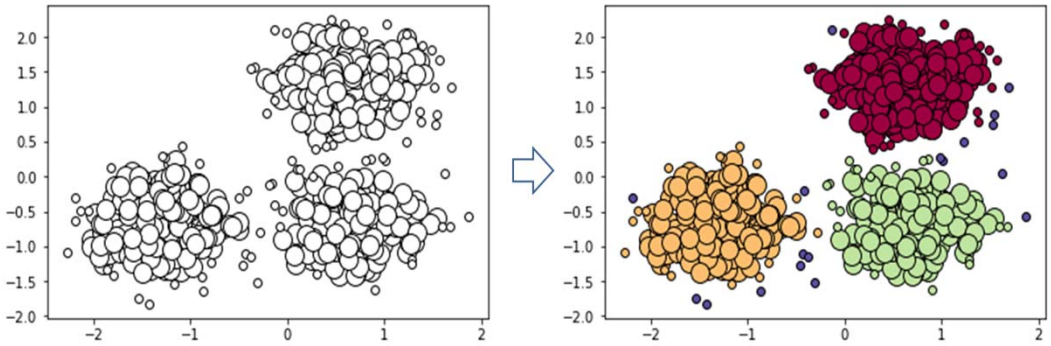
本质：簇的划分



聚类的定义：将物理或抽象对象的集合分成由类似的对象组成的多个类的过程被称为聚类。由聚类所生成的簇是一组数据对象的集合，这些对象与同一个簇中的对象彼此相似，与其他簇中的对象相异。

什么是聚类?

聚类和分类的区别



- **从类别上看**，分类问题中的类别数目是已知的，而聚类问题所要划分得到的簇的数目通常是未知的。
- **从数据格式上看**，分类问题中的每条样本数据包括特征和标签两大部分。 而聚类问题处理的样本通常只有特征，需要根据某一标准将其划分到不同的簇中。
- **从方法上看**，分类是一种有监督的机器学习方法，侧重在学习如何将样本特征映射为类别标签的函数；而聚类侧重在划分，以便将相似或者距离近的样本归为同一个簇。

聚类问题

本章内容简介

- 什么是聚类
- 聚类算法评价指标
- 常见的聚类算法

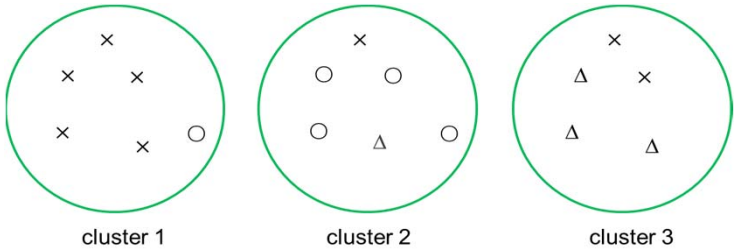
聚类问题

本章内容简介

- 什么是聚类
- **聚类算法评价指标**
  - 纯度
  - 兰德指数
  - 聚类精确率、聚类召回率、聚类F值
  - 调整互信息
  - 轮廓系数
  - 同质性、完整性
- 常见的聚类算法

聚类算法评价指标

纯度



- 假设聚类算法将样本集合分成K 个簇，则找到每个簇的主类别（该类别样本在该簇中的出现次数比其它类别多），将所有K个簇中的主类别样本数求和并归一化。计算公式为：

$$purity(\Omega, C) = \frac{1}{N}\sum_k \max_j |\Omega_k \cap c_j|$$

则对于如图的聚类结果，聚类纯度的计算过程为：

1) 首先计量每个簇的主类别样本数：
   - cluster 1 中 class × 数目最多（共有5个）
   - cluster 2中 class ○ 数目最多（共有4个）
   - cluster 3 中 class Δ 数目最多（共有3个）
2) 基于上述计数结果计算综合纯度，即

$$purity(\Omega, C) = \frac{1}{N}\sum_k \max_j |\Omega_k \cap c_j| = \frac{1}{17} \times (5 + 4 + 3) = 0.7059$$

其中17为输入样本集合中的样本总数。

聚类算法评价指标

兰德指数

（一）基本思想：兰德指数（Rand index，RI）[Rand, 1971]将聚类看成是对样本集每个样本对的决策过程：当且仅当两个样本相似时，才应该将二者都归入到同一簇中。

因此正确决策的数量为TP和TN：

- TP 表示两个同类样本点被划分到同一个簇中的次数（即“同→同”）；
- TN 表示两个非同类样本点被划分到两个不同簇的次数（即“不同→不同”）。

错误决策的数量为FP和FN:

- FP表示两个非同类的样本点被划分到同簇中的次数（即“不同→同”）；
- FN 表示两个同类的样本点被划分到两个不同簇的次数（即“同→不同”）。

（二）基于以上四个统计数值，兰德指数计算为正确决策的比率：

$$RI = \frac{TP+TN}{TP+FP+TF+FN} = \frac{TP+TN}{C_N^2} \qquad (5\text{-}2)$$

其中，C_N^2表示数据集中所有N个样本可以自由组合形成的样本对的个数。

兰德指数的取值范围为$RI \in [0,1]$，取值越大表示聚类的划分结果与真实的类别划分越一致。

兰德指数的计算举例。假设某聚类算法的聚类结果仍然如图5.2所示。则兰德指数的计算过程为：

1）先计算同一簇中任取两个样本点形成的总的组合数，其包含了同类和非同类两种情况：即$TP + FP = C_6^2 + C_6^2 + C_5^2$=15+15+10=40。

2) TP为同类（class）中的样本对出现在同一簇（cluster） 中的次数。因此需要统计每个簇（cluster）中样本个数大于等于2的同类点，并分别求其自由组合数：本例中$TP = C_5^2 + C_4^2 + C_3^2 + C_2^2 = 20$ （后两项是因为cluster 3中×和Δ的数目都大于或者等于2个）

3）综上，有$FP = 40 - 20 = 20$。

4) TP+FN 实际上表示的是两个同类样本被划分到在同一簇和不同簇的所有情况总和，所以有$TP + FN = C_8^2 + C_5^2 + C_4^2 = 44$，因此$FN$=24。

5）对于本例中输入的17个样本，所有可能出现情况的总和为：$TP + FP + TN + FN = C_{17}^2 = 136$

6）综上，可以求得$\mathrm{TP} = 20, \mathrm{FP} = 20, \mathrm{FN} = 24, \mathrm{TN} = 72$，如表5. 1所示。

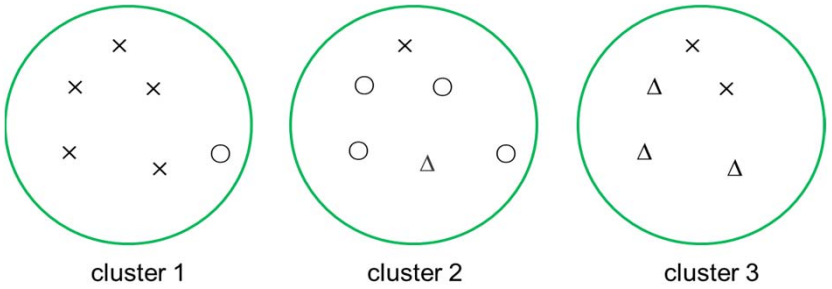


表5.1兰德指数的四个要素数值

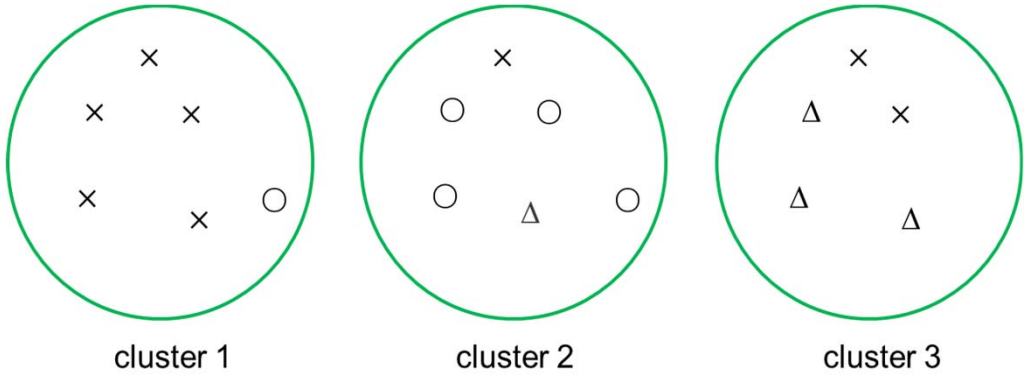
	相同簇	不同簇
相同类	TP=20	FN=24
不同类	FP=20	TN=72

从而，该例子中的兰德指数计算为

$$RI = \frac{TP+TN}{TP+FP+TF+FN} = \frac{TP+TN}{C_N^2} = \frac{20+72}{20+20+24+72} \approx 0.68。$$

聚类算法评价指标

聚类的精确率、召回率和F1值



- 聚类结果的精确率、召回率和F1值的计算方法与用于评价分类时的度量方法计算类似，**只是其中的四个要素即TP、FP、FN、TN表示各种组合的数量取值**，其含义参照上一节兰德指数，因而与分类不同。
- **计算举例。**假设某聚类算法的聚类结果仍然如图5.2所示。根据表5.1中所示的四个要素数值，则有

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}) = 20/40 = 0.5$$

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}) = 20/44 = 0.455$$

$$\text{F1} = 2 \times \text{Recall} \times \frac{\text{Precision}}{\text{Recall+Precision}} = (2 * 0.5 * 0.455)/(0.5 + 0.455) = 0.48。$$

聚类算法评价指标

基于互信息的评价方法（借鉴了信息论中互信息的计算方法）

- 假设有N个样本，对于两种划分U和V：$U=\{U_1,U_2,\dots,U_R\}$，$V=\{V_1,V_2,\dots,V_C\}$，$n_{ij}=|U_i\cap V_j|$表示U_i和V_j的公共元素的数目。则**U和V之间的互信息计算方法如下**：

  1）随机选择一个样本，其落在U_i中的概率为$P(i)=\frac{|U_i|}{N}$，因此划分U的熵计算为：$H(U)=-\sum_{i=1}^{R}P(i)logP(i)$

  2）随机选择一个样本，其落在V_j中的概率为$P'(j)=\frac{|V_j|}{N}$，因此划分V的熵计算为$H(V)=-\sum_{j=1}^{C}P'(j)log\ P'(j)$

  3）因此，两种划分U和V之间的互信息为$MI(U,V)=\sum_{i=1}^{R}\sum_{j=1}^{C}P(i,j)log\frac{P(i,j)}{P(i)P'(j)}$，其中$P(i,j)=\frac{|U_i\cap V_j|}{N}$表示样本同时落在$U_i$和$V_j$的概率。

- 由于其中一种划分例如U可以作为参考的真实划分，V为聚类算法得到的另外一种划分结果，则以上方法可以用于评价聚类的性能。
- **利用互信息评价聚类结果的一个不足之处是**：如果划分的簇数量较多，其互信息取值可能会更大，特别地当K=N时，互信息达到最大值。因此互信息和纯度类似，其难以衡量聚类质量与簇的数量K之间的关系。

聚类算法评价指标

轮廓系数

- 上述各种聚类结果的评价方法假设输入的样本集合存在一个可供参考的标准划分。
- 而**轮廓系数（Silhouette Coefficient）[Rousseeuw, 1987]适用于每个样本的实际类别归属信息未知的情况，其通过结合内聚度和分离度两种因素来评价聚类结果。**
- **轮廓系数的计算方法为**：
  1) 对于每个样本i：
     a) 计算$a(i)$。$a(i)$为样本i到同簇其它样本的平均距离，其值越小，则样本 i 越应该被聚类到该簇；
     b) 计算$b(i)$。$b(i)$为样本i到每个其它簇（除了自己所属的簇以外）的所有样本的平均距离的最小值。
     c) 计算样本 i 的轮廓系数$s(i)$：$s(i) = \frac{b(i)-a(i)}{\max\{a(i),\ b(i)\}}$ (5-7)

     $s(i)$取值范围为$-1 \le s(i) \le +1$。$s(i)$接近+1，则说明样本i聚类合理；$s(i)$接近-1，则说明样本i更应该分类到另外的簇。
  2) 所有样本的$s(i)$的均值即为聚类结果的轮廓系数
- **轮廓系数取值越大，表示同簇内的样本之间距离较小，不同簇间的距较离大**，这实际上正是聚类的定义和目标。

聚类算法评价指标

同质性与完整性（1/2）

- **同质性（Homogeneity）的评价目标**是认为聚类算法得到的每个簇越纯越好[Rosenberg, 2007]。例如如果每个簇只包含一个类的样本，则同质性取值为1。
- 假设有N个数据点，标准的参考划分为$C = \{c_i | i = 1, \dots, n\}$，算法划分的簇为$K = \{k_i | 1, \dots, m\}$。假设$A$为邻接表即$A = \{a_{ij}\}$且其中$a_{ij}$表示同时属于类$c_i$和簇 k_j的元素的个数。

  则同质性的计算方法为：

$$h = \begin{cases} 1, & if\ H(C|K) = 0 \\ 1 - \frac{H(C|K)}{H(C)}, & else \end{cases} \tag{5-8}$$

其中，$H(C|K) = -\sum_{k=1}^{|K|}\sum_{c=1}^{|C|}\frac{a_{ck}}{N} log \frac{a_{ck}}{\sum_{c=1}^{|C|} a_{ck}}$，$H(C) = -\sum_{c=1}^{|C|}\frac{\sum_{k=1}^{|K|} a_{ck}}{n} log \frac{\sum_{k=1}^{|K|} a_{ck}}{n}$。

聚类算法评价指标

同质性与完整性（2/2）

- **完整性（Completeness）的评价目标是认为聚类算法得到的每个簇应该尽可能包括某个类的全部样本。**完整性的计算方法为：

$$c = \begin{cases} 1, & if\ H(K|C) = 0 \\ 1 - \frac{H(K|C)}{H(K)}, & else \end{cases} \tag{5-9}$$

其中，$H(K|C) = -\sum_{c=1}^{|C|}\sum_{k=1}^{|K|}\frac{a_{ck}}{N} log\frac{a_{ck}}{\sum_{k=1}^{|K|}a_{ck}}$，$H(K) = -\sum_{k=1}^{|K|}\frac{\sum_{c=1}^{|C|}a_{ck}}{n} log\frac{\sum_{c=1}^{|C|}a_{ck}}{n}$。

- 可见，上述**同质性和完整性两个评价指标侧重聚类的不同目标，因而可以互相弥补**，相应地，V-measure则综合考虑二者，即V-measure是均一性和完整性的调和平均值：

$$V_\beta = \frac{(1+\beta)*h*c}{(\beta*h)+c} \tag{5-10}$$

如果β大于1，则完整性的权重更大，反之，则同质性的权重更大。

聚类问题

本章内容简介

- 什么是聚类
- 聚类算法评价指标
- 常见的聚类算法
  - K-means聚类算法
  - 吸引子传播聚类算法
  - 层次聚类算法
  - DBSCAN密度聚类
  - 均值偏移聚类算法
  - 自组织映射聚类算法
  - 高斯混合模型聚类算法

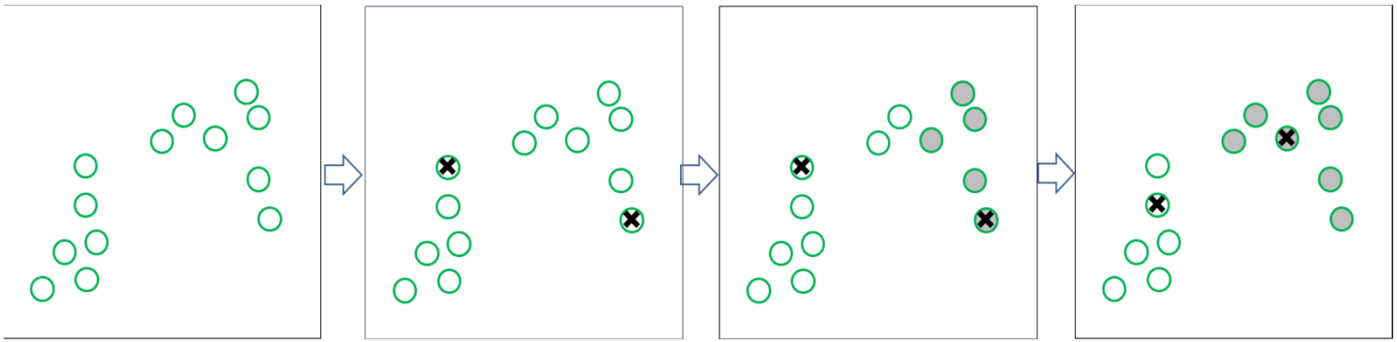
聚类问题

本章内容简介

- 什么是聚类
- 聚类算法评价指标
- 常见的聚类算法
  - **K-means聚类算法**
  - 吸引子传播聚类算法
  - 层次聚类
  - DBSCAN密度聚类
  - 均值偏移聚类算法
  - 自组织映射聚类算法
  - 高斯混合模型聚类算法

常见的聚类方法

K均值聚类算法

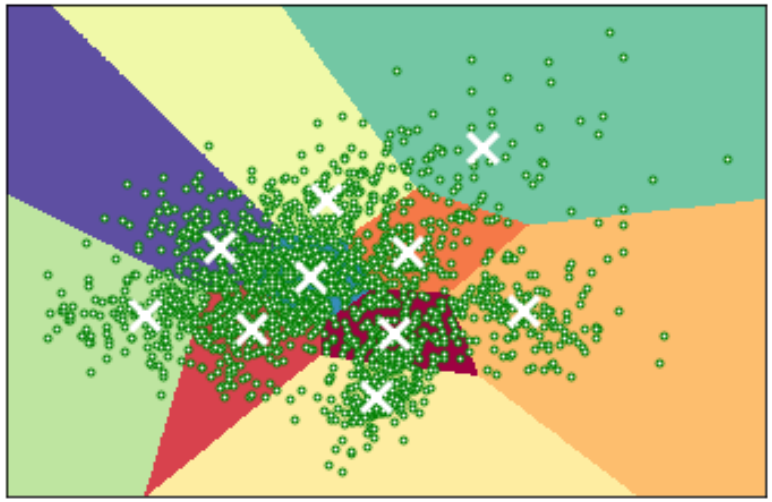


K均值聚类算法的基本步骤为：

1. **初始化。**设定聚类的簇数目K，从待聚类样本中随机选取K个对象作为初始的聚类中心；
2. **进行一次聚类迭代。**计算每个对象与当前各个聚类中心之间的距离，把每个对象分配给距离它最近的聚类中心（**寻找组织**）。每分配一个样本，所在聚类中心的位置根据聚类中现有的对象重新计算；
3. **判断是否满足聚类终止条件。**终止条件可以是没有（或最小数目）对象被重新分配给不同的簇，没有（或最小数目）聚类中心发生变化，或者聚类的误差平方和局部最小等。如果不满足终止条件，则返回步骤2继续迭代；
4. 算法结束。

常见的聚类方法

高维样本聚类结果的可视化



- 为了将高维空间（例如图像、文本等）上样本点的聚类结果在二维平面上展示，可以借助主成分分析法或者自编码器等方法进行降维处理，然后再利用K均值聚类算法聚类并进行二维可视化。
- 例如，图5.4 给出了在手写体数字图像数据集上的K均值聚类结果可视化效果（K=10）。其中“×”表示聚类中心，不同颜色表示不同的簇。该例子中利用主成分分析法将原始数字图像维度压缩到2维然后进行聚类，以便在二维平面上可视化每个点。

常见的聚类方法

K均值聚类算法中的K值确定问题

- K均值聚类算法中的K值可以根据经验来指定，也可以通过某种计算手段来自动辅助选择恰当的K值。
  - 例如一种方法是**基于前文介绍的轮廓系数来辅助选择K值**：其基本思路是设定K可能的取值区间，计算K的每个取值的轮廓系数，最后选择轮廓系数取值最大的K值进行聚类。
  - ……

常见的聚类方法

K-means++聚类算法

- K-means++[Arthur，2007]是K-means算法的变体。其针对标准K-Means聚类算法对初始样本点选择比较敏感的问题，**对初始聚类中心样本点的选择策略进行了改进，其选取原则是使初始聚类中心样本点之间的相互距离尽可能的远。**
- **K-means++聚类算法中初始聚类中心点的选择步骤**：
  1. 随机选取一个样本作为第一个聚类中心 C1；
  2. 计算剩余每个样本i与当前所有的簇中心的最短距离d_i；
  3. 使用**轮盘法**从剩余样本中选出下一个聚类中心，即 d_i越大，则轮盘法中该样本被选中作为下一个聚类中心的概率越大；
  4. 重复步骤2，直到选出 K 个聚类中心为止。
- 通过上述步骤选出初始中心样本点后，**K-means++聚类算法的其余聚类步骤**与标准的 K-means 算法相同。
- 与K-means相比，K-means++可以显著提高聚类质量。尽管K-means++计算初始聚类中心额外多花费了时间，但其能找到较好的初始中心，往往能快速收敛。

常见的聚类方法

Mini Batch K-Means

- Mini Batch K-Means聚类算法[Fitriyani, 2016]也是K-Means算法的变体，主要**适用于大量样本数据的聚类场景，以提高聚类效率。**
- **其基本思想是**针对参与聚类样本数量特别大的应用场景，采取随机抽选小部分样本来代替整体的策略进行聚类中心的计算和迭代更新。
- **其与K-Means的不同之处在于**：每次迭代更新聚类的中心点时，是利用所有样本中随机选取的一个小集合（即mini-batch）中的样本来更新中心。这种过程多次进行，直到收敛或者达到结束条件。形成各个簇以后，在最后步骤中将所有样本分配到各个簇。

聚类问题

本章内容简介

- 什么是聚类
- 聚类算法评价指标
- 常见的聚类算法
  - K-means聚类算法
  - 吸引子传播聚类算法
  - 层次聚类
  - DBSCAN密度聚类
  - 均值偏移聚类算法
  - 自组织映射聚类算法
  - 高斯混合模型聚类算法

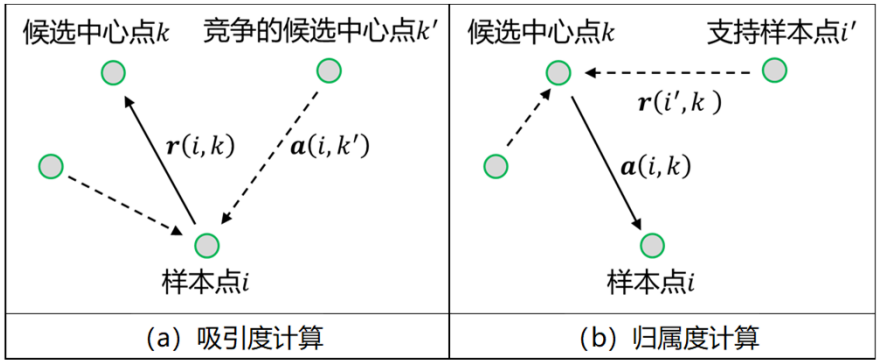
聚类问题

本章内容简介

- 什么是聚类
- 聚类算法评价指标
- 常见的聚类算法
  - K-means聚类算法
  - **吸引子传播聚类算法**
  - 层次聚类
  - DBSCAN密度聚类
  - 均值偏移聚类算法
  - 自组织映射聚类算法
  - 高斯混合模型聚类算法

常见的聚类方法

吸引子传播聚类算法(1/2)

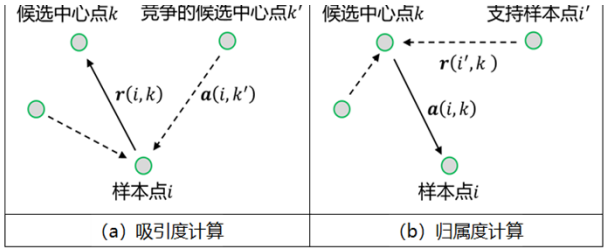


(a) 吸引度计算　　(b) 归属度计算

- **吸引子传播聚类算法**（Affinity Propagation，简称AP算法）[Frey, 2007]，也被称为近邻传播聚类或亲和力传播聚类，是一种基于图论的聚类算法。
- **其基本原理是**首先利用样本相互之间的相似度矩阵构建连接网络，每个样本为网络中的节点。然后在每次迭代中分别进行**样本之间吸引度（Responsibility）和归属度（Availability）的传递计算**，以逐步求精计算输入样本集合的聚类中心。

  样本之间吸引度和归属度的定义为：

  - **吸引度**：$r(i,k)$描述了**样本点k适合作为样本i的聚类中心**的程度，表示的是从i到k的消息，如图5.3（a）所示。
  - **归属度**：$a(i,k)$描述了**样本点i选择样本点k作为其聚类中心**的适合程度，表示从k到i的消息，如图5.3（b）所示。



吸引子传播聚类算法的基本步骤为：

1. 假设待聚类的样本集合为$\{x_1, x_2, \cdots, x_n\}$。构建样本两两之间的相似度矩阵$S$，即其每个元素$s(i,j)$为样本$x_i$和$x_j$之间的相似度；
2. 设置时间步$t=0$，初始化吸引信息矩阵R和归属信息矩阵A；
3. $t=t+1$，更新矩阵R和A：
   - 更新矩阵R：$r_{t+1}(i,k)=s(i,k)-\max\limits_{k'\neq k}\{a_t(i,k')+s(i,k')\}$ (5-11)
   - 更新矩阵A：

$$a_{t+1}(i,k)=\begin{cases} min\left(0, r_t(k,k)+\sum_{i'\notin\{i,k\}}\max\{0, r_t(i',k)\}\right), & i\neq k \\ \sum_{i'\neq k}\max\{0, r_t(i',k)\}, & i=k \end{cases} \quad (5\text{-}12)$$

4. 引入衰减系数λ进行衰减计算，即
   - 更新矩阵R：$r_{t+1}(i,k)\leftarrow(1-\lambda)r_{t+1}(i,k)+\lambda r_t(i,k)$ (5-13)
   - 更新矩阵A：$a_{t+1}(i,k)\leftarrow(1-\lambda)a_{t+1}(i,k)+\lambda a_t(i,k)$ (5-14)
5. 重复步骤3和4，直到矩阵R和A稳定或者达到预设的最大迭代次数；
6. 将$a(i,k)+r(i,k)$取值最大的若干对角线元素对应的样本k作为类别中心，每个样本点i归属于$a(i,k)+r(i,k)$取值最大的类别k。

聚类问题

本章内容简介

- 什么是聚类
- 聚类算法评价指标
- 常见的聚类算法
  - K-means聚类算法
  - 吸引子传播聚类算法
  - 层次聚类算法
  - DBSCAN密度聚类算法
  - 均值偏移聚类算法
  - 自组织映射聚类算法
  - 高斯混合模型聚类算法

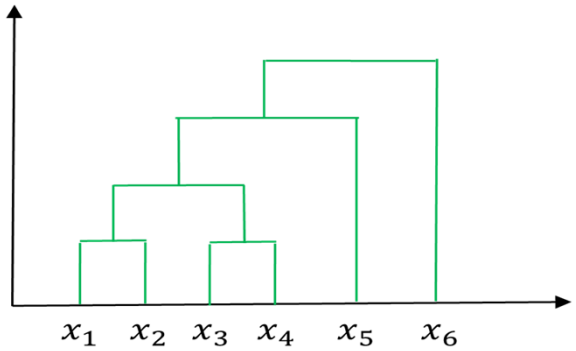
聚类问题

本章内容简介

- 什么是聚类
- 聚类算法评价指标
- 常见的聚类算法
  - K-means聚类算法
  - 吸引子传播聚类算法
  - **层次聚类算法**
  - DBSCAN密度聚类算法
  - 均值偏移聚类算法
  - 自组织映射聚类算法
  - 高斯混合模型聚类算法

常见的聚类方法

层次聚合聚类算法



- 层次聚合聚类算法在计算两簇间的距离时可以使用如下三种聚类准则之一：
  - **最大距离**（也称为complete-linkage聚类）。即将两个簇中距离最远的样本之间的距离作为两类之间的距离。
  - **最小距离**（也称为single-linkage聚类）。即将两个簇中距离最近的样本之间的距离作为两类之间的距离。
  - **平均距离**（也称为average-linkage聚类）。即两个簇中所有样本的距离的平均值作为两类之间的距离。
- 层次聚合聚类算法具有距离定义简单、可以发现簇间的层次关系等优点；缺点是计算量大，对异常值敏感等。

聚类问题

本章内容简介

- 什么是聚类
- 聚类算法评价指标
- 常见的聚类算法
  - K-means聚类算法
  - 吸引子传播聚类算法
  - 层次聚类
  - DBSCAN密度聚类算法
  - 均值偏移聚类算法
  - 自组织映射聚类算法
  - 高斯混合模型聚类算法

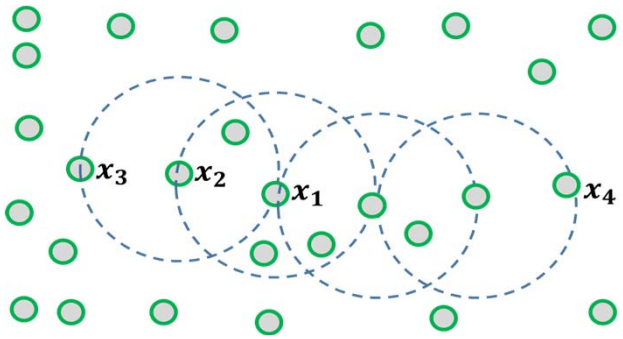
聚类问题

本章内容简介

- 什么是聚类
- 聚类算法评价指标
- 常见的聚类算法
  - K-means聚类算法
  - 吸引子传播聚类算法
  - 层次聚类
  - **DBSCAN密度聚类算法**
  - 均值偏移聚类算法
  - 自组织映射聚类算法
  - 高斯混合模型聚类算法

常见的聚类方法

DBSCAN密度聚类——相关定义



DBSCAN聚类算法的基本原理图

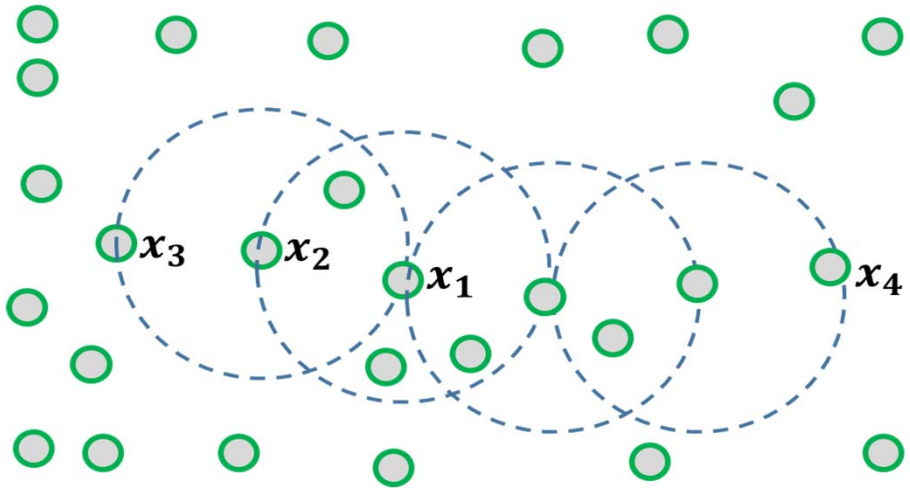
- **MinPts=3；**
- **核心对象为x_1；**
- **x_2由x_1密度直达；**
- **x_3由x_1密度可达；**
- **x_3与x_4密度相连**

结合图5.7，设样本集$D=(x_1,x_2,\ldots,x_m)$，DBSCAN聚类算法中的若干基本概念定义如下：

- **ε-邻域：** 样本x_j的ε-邻域定义为D中与x_j的距离不大于ϵ的子样本集$N\varepsilon(x_j)=\{x_j\in D|distance(x_i,x_j)\leq\varepsilon\}$；
- **核心对象**：对于任一样本x_j，如果其ϵ-邻域中至少包含$MinPts$个样本，即如果$|N\epsilon(xj)|\geq MinPts$，则$x_j$为核心对象；
- **密度直达**：如果x_i位于x_j的ϵ-邻域中，且x_j是核心对象，则称x_i可由x_j密度直达。注意反之不一定成立，即密度直达不满足对称性；
- **密度可达**：对于样本x_i和x_j，如果存在核心样本对象序列$p_1,p_2,\ldots,p_T$，满足$p_1=x_i,p_T=x_j$，且p_{t+1}由p_t密度直达，则称x_j由x_i密度可达。密度可达满足传递性，但不满足对称性（可以由密度直达的不对称性得出）。
- **密度相连**：对于样本x_i和x_j，如果存在核心对象x_k，使x_i和x_j均由x_k密度可达，则称x_i和x_j密度相连。密度相连关系满足对称性。

常见的聚类方法

DBSCAN密度聚类



- DBSCAN（Density-Based Spatial Clustering of Applications with Noise）是一种**基于密度**的聚类算法。
- **其基本原理为**：从任意未被访问的样本点开始，如果其ε-邻域内有**足够数目（minPts）**的点，则将其视为**核心对象**并建立一个新簇，然后找到该核心对象**能够密度可达**的样本集合以扩充该簇。之后重复这一过程，直到所有核心对象都找到归属的簇为止，如图5.7所示。
- 基于这一原理，核心对象点的连通以及核心对象的ε-邻域（即图中的虚线圆圈），将样本对象点分成若干个簇。其它的点为噪声点，即不属于任何一个簇。

算法 5.1: DBSCAN 聚类算法

Input: 样本集合 $D = \{x_1, x_2, ..., x_m\}$, 邻域参数 $(\varepsilon, MinPts)$, 样本距离度量方式

Output: 簇划分 C

```
将数据集 D 的所有样本点标记为未处理状态;
for 数据集 D 中每个样本点 p do
    if (p 已经归入某个簇或标记为噪声) then
        continue;
    else
        if (p 的 ε 邻域中的样本点数小于 MinPts) then
            标记样本点 p 为边界点或噪声点;
        else
            标记样本点 p 为核心点，建立新簇 C, 并将 p 的 ε 邻域中所有样本点加入簇
             C 中;
            for p 的 ε 邻域中所有未被处理的样本点 q do
                如果样本点 q 的 ε 邻域包含的样本点数大于或等于 MinPts，则将其中
                 未归入任何一个簇的样本点加入簇 C 中;
            end
        end
    end
end
```

常见的聚类方法

高维样本聚类结果的可视化

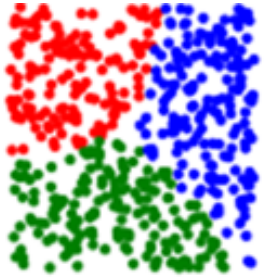
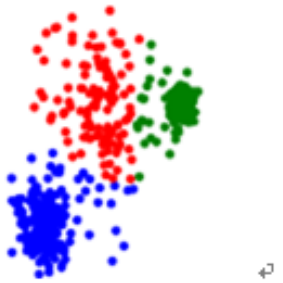
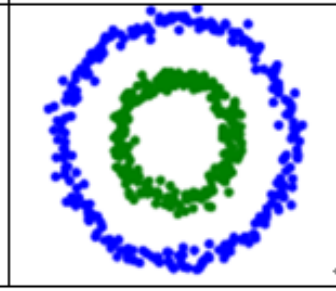
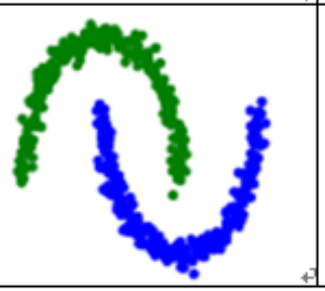
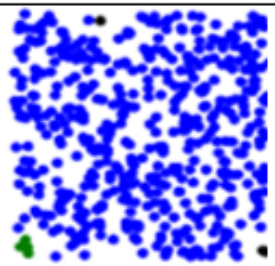
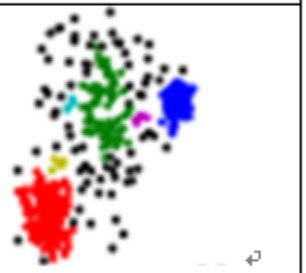
K-means				
DBSCAN				

图 5.8 DBSCAN 与 K 均值聚类算法聚类结果的对比

聚类问题

本章内容简介

- 什么是聚类
- 聚类算法评价指标
- 常见的聚类算法
  - K-means聚类算法
  - 吸引子传播聚类算法
  - 层次聚类
  - DBSCAN密度聚类算法
  - 均值偏移聚类算法
  - 自组织映射聚类算法
  - 高斯混合模型聚类算法

聚类问题

本章内容简介

- 什么是聚类
- 聚类算法评价指标
- 常见的聚类算法
  - K-means聚类算法
  - 吸引子传播聚类算法
  - 层次聚类
  - DBSCAN密度聚类算法
  - **均值偏移聚类算法**
  - 自组织映射聚类算法
  - 高斯混合模型聚类算法

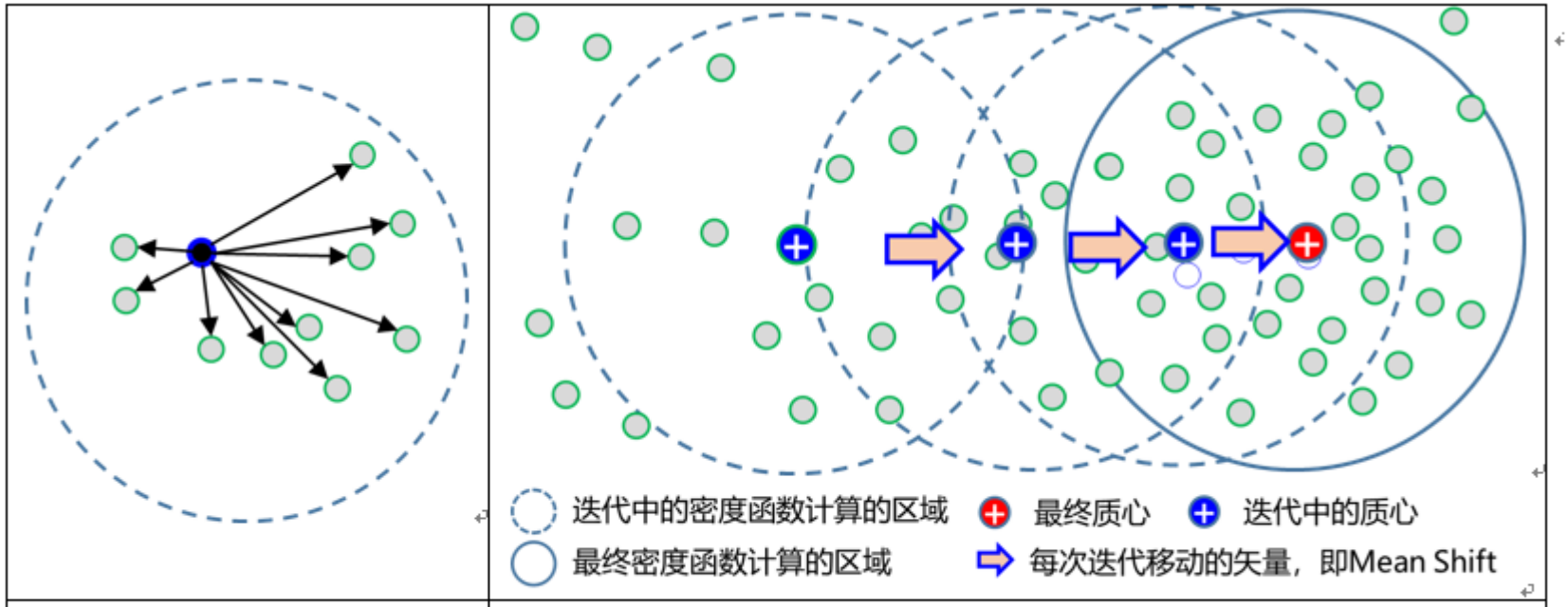
常见的聚类方法

均值偏移聚类算法

- 首先给出**均值偏移向量（Mean Shift Vector）的定义。**对于样本集中的某个点x，其均值偏移向量为：

$$M_h(x) = \frac{1}{k}\sum_{x_i \in S_h}(x_i - x) \qquad (5\text{-}15)$$

其中S_h是以样本点x为中心的半径为h的高维球区域（例如图5.9（a）所示的二维圆形区域示例），k为该区域的点数。



（a）均值偏移向量　　（b）均值偏移的目的是找到最密集的区域

算法 5.2: 均值偏移聚类算法

Input: 待聚类样本集 D、半径 r、迁移长度 $(shift)$ 阈值 T_s、中心距离阈值 T_c

Output: 簇的划分

```
将数据集 D 的所有样本点标记为未处理状态;
do
    从 D 中随机选择一个未标记样本 x，作为起始中心点 center 并创建新簇 C;
    do
        将以 center 为中心、半径为 r 的区域中所有样本点放入簇 C，并记录这些点在簇 C 中的出现次数为 1;
        以 center 为中心，计算从 center 开始到半径 r 区域内每个样本点的向量并累加得到 shift;
        center = center + shift //即 center 沿着 shift 方向移动，将移动过程中遇到的点都归类到簇 C;
    while shift 的长度不小于预定的阈值 T_s;
    保留簇 C 当前的中心位置 center;
    if 当前簇 C 的中心位置 center 与其它已经存在的簇 C' 的中心之间的距离小于预定的阈值 T_c, then
        将两个簇 C 与簇 C' 合并，并归并其中每个数据点的出现次数;
    end
while 不是所有点都被标记为已访问;
对于每个样本点 x，将其分配给访问频率最大的簇
```

聚类问题

本章内容简介

- 什么是聚类
- 聚类算法评价指标
- 常见的聚类算法
  - K-means聚类算法
  - 吸引子传播聚类算法
  - 层次聚类
  - DBSCAN密度聚类算法
  - 均值偏移聚类算法
  - 自组织映射聚类算法
  - 高斯混合模型聚类算法

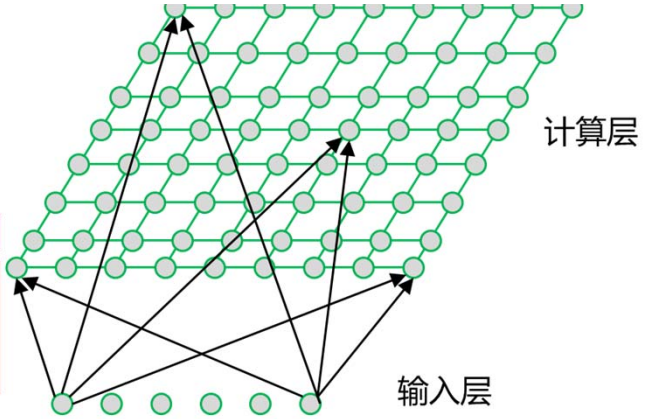
聚类问题

本章内容简介

- 什么是聚类
- 聚类算法评价指标
- 常见的聚类算法
  - K-means聚类算法
  - 吸引子传播聚类算法
  - 层次聚类
  - DBSCAN密度聚类算法
  - 均值偏移聚类算法
  - **自组织映射聚类算法**
  - 高斯混合模型聚类算法

常见的聚类方法

自组织映射网络聚类



算法 5.3: 自组织映射神经网络聚类算法

Input: 待聚类样本集 D、学习速率 η

Output: 簇的划分

初始化。对输出层各节点的向量随机初始化，初始化学习速率 η;

do

  接受输入。//从训练集中随机选取样本作为网络输入;

  寻找获胜神经元。//计算输入样本向量与输出层中每个节点向量之间的相似度，并选出相似度最大的节点作为获胜节点;

  确定获胜节点为中心的邻域。//邻域范围一般随着训练时间的增加而逐渐收缩;

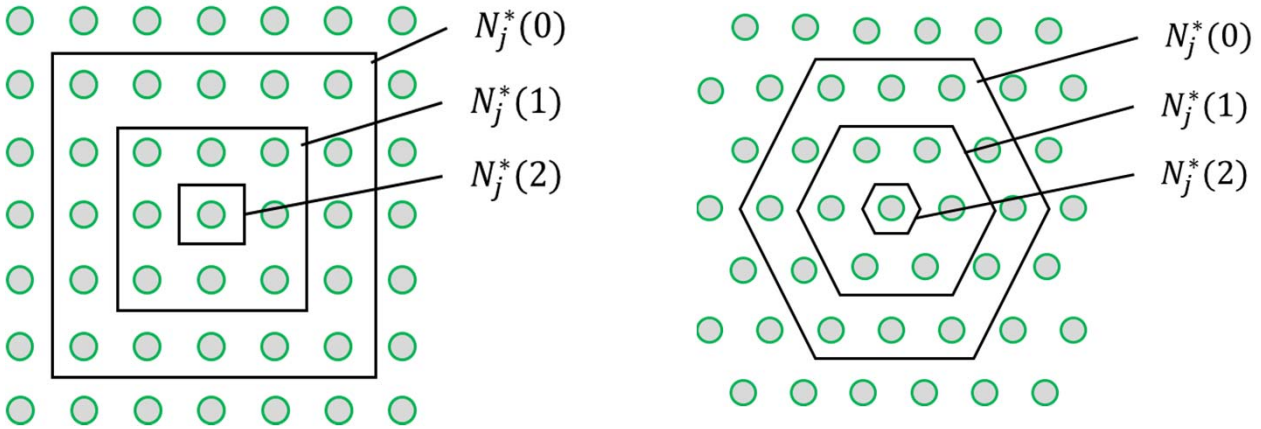
  调整权值：$w_j(t+1) = w_j(t) + \eta(t, N)[x - w_j(t)]$

while 不满足训练结束条件（例如学习速率 η 衰减到某个预定的阈值);

输出聚类。将输入的每个样本映射到与之最相似的输出层节点，完成聚类。

常见的聚类方法

自组织映射网络聚类



优胜邻域的范围随着训练的进行逐渐缩小

- 算法中，调整权值是通过对包括优胜节点及其邻域在内的每个节点 j所对应的向量w_j进行调整，向输入样本向量x靠近。
- $\eta(t,N)$是与训练时间t，以及邻域内节点j与获胜神经元j^*之间拓扑距离N的函数，其一般应该满足：$t\uparrow\rightarrow\eta\downarrow$，$N\uparrow\rightarrow\eta\downarrow$，即随着训练时间的增加或者距离获胜神经元拓扑距离的增加，函数取值呈现递减的规律。例如可构造为$\eta(t,N)=\eta(t)e^{-N}$。**优胜邻域的范围随着训练的进行逐渐缩小，如图所示。**

聚类问题

本章内容简介

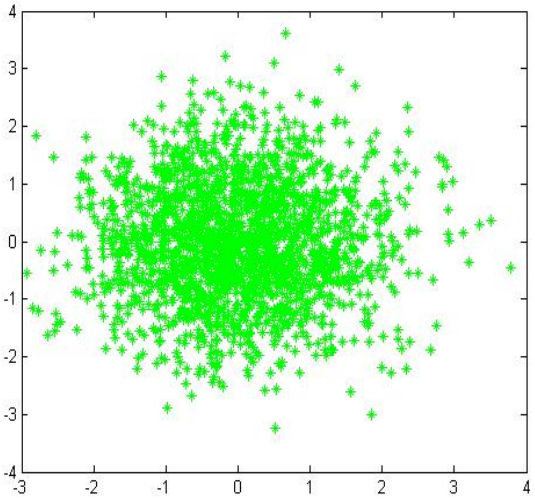
- 什么是聚类
- 聚类算法评价指标
- 常见的聚类算法
  - K-means聚类算法
  - 吸引子传播聚类算法
  - 层次聚类
  - DBSCAN密度聚类算法
  - 均值偏移聚类算法
  - 自组织映射聚类算法
  - 高斯混合模型聚类算法

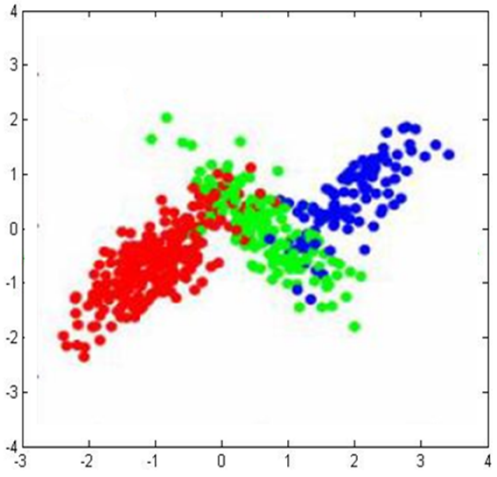
聚类问题

本章内容简介

- 什么是聚类
- 聚类算法评价指标
- 常见的聚类算法
  - K-means聚类算法
  - 吸引子传播聚类算法
  - 层次聚类
  - DBSCAN密度聚类算法
  - 均值偏移聚类算法
  - 自组织映射聚类算法
  - **高斯混合模型聚类算法**

高斯混合模型聚类算法





- **高斯混合模型（Gaussian Mixture Model，GMM）的基本思想**：任意形状的概率分布都可以用多个高斯分布函数去近似。每个高斯分布函数称为一个“Component”。这些"Component"组成了 GMM 的概率密度函数。
- 因此，高斯混合模型就是把数据看作是从K个高斯分布中生成出来的。K 值需要事先指定。
- 与其它聚类算法采用相似度或者距离作为判断样本归属的依据不同的是，GMM采用概率作为判断依据，即通过属于某一类的概率大小来判断类别归属。

高斯混合模型的定义

- 定义：假设观测数据$y(y_1, y_2, \ldots, y_N)$由高斯混合模型生成，则高斯混合模型是具有如下形式的概率分布模型：

  $P(y|\theta) = \sum_{k=1}^{K} \alpha_k \phi(y|\theta_k)$。 其中，

  - $\phi(y|\theta_k)$是高斯分布密度，$\phi(y|\theta_k) = \frac{1}{\sqrt{2\pi}\sigma_k} exp\left(-\frac{(y-\mu_k)^2}{2\sigma_k^2}\right)$ 称为第k个分模型。 $\theta_k = (\mu_k, \sigma_k^2)$，
  - α_k是系数， $\alpha_k \geq 0$， $\sum_{k=1}^{K} \alpha_k = 1$；
  - $\theta = (\alpha_1, \alpha_2, \ldots, \alpha_K; \theta_1, \theta_2, \ldots, \theta_K)$。
  - 可以用EM算法估计高斯混合模型的参数θ。

参考文献：李航《统计学习方法》

用EM算法估计参数θ的推导过程

■ 第一步：明确隐变量，写出完全数据的对数似然函数

假设观测数据$y_j, j = 1,2 \ldots, N$, 这样产生：首先依概率α_k选择第k个高斯分布分模型$\phi(y|\theta_k)$；然后依该模型的概率分布生成观测数据y_j,。

此时，观测数据$y_j, j = 1,2 \ldots, N$是已知的，但是y_j具体来自哪个分模型是未知的，可以用隐变量γ_{jk}（$k = 1,2 \ldots, K$）表示：

$$\gamma_{jk} = \begin{cases} 1, & \text{第}j\text{个观测数据来自第}k\text{个分模型} \\ 0, & \text{否则} \end{cases}$$

有了观测数据y_j及未观测数据γ_{jk}，则完全数据是

$$(y_j, \gamma_{j1}, \gamma_{j2}, \cdots, \gamma_{jK}),\ j = 1,2 \ldots, N$$

于是，可以写出完全数据的似然函数：

$$P(y, \gamma|\theta) = \prod_{j=1}^{N} P(y_j, \gamma_{j1}, \gamma_{j2}, \cdots, \gamma_{jK}|\theta)$$

用EM算法估计参数θ的推导过程

$$P(y,\gamma|\theta)=\prod_{j=1}^{N}\boldsymbol{P(y_j,\gamma_{j1},\gamma_{j2},\cdots,\gamma_{jK}|\theta)}$$

$$=\prod_{j=1}^{N}\prod_{k=1}^{K}\boldsymbol{[\alpha_k\phi(y_j|\theta_k)]^{\gamma_{jk}}}$$

$$=\prod_{k=1}^{K}\prod_{j=1}^{N}[\alpha_k\phi(y_j|\theta_k)]^{\gamma_{jk}}$$

$$=\prod_{k=1}^{K}{\alpha_k}^{n_k}\prod_{j=1}^{N}[\phi(y_j|\theta_k)]^{\gamma_{jk}}$$

$$=\prod_{k=1}^{K}{\alpha_k}^{n_k}\prod_{j=1}^{N}\left[\frac{1}{\sqrt{2\pi}\sigma_k}exp\left(-\frac{(y_j-\mu_k)^2}{2\sigma_k^2}\right)\right]^{\gamma_{jk}}$$

根据指数$\boldsymbol{\gamma_{jk}}$的定义，只有观测 j 来自第 k 个分模型时，其为1，其余均为0。

对于每个分模型k，α_k要乘n_k次；$n_k=\sum_{j=1}^{N}\gamma_{jk}$，$\sum_{k=1}^{K}n_k=N$

则完全数据的对数似然函数为

$$\boldsymbol{logP(y,\gamma|\theta)}$$

$$=\sum_{k=1}^{K}\left\{n_k log\alpha_k+\sum_{j=1}^{N}\gamma_{jk}\left[log\left(\frac{1}{\sqrt{2\pi}}\right)-log\sigma_k-\frac{1}{2\sigma_k^2}(y_j-\mu_k)^2\right]\right\}$$

用EM算法估计参数θ的推导过程

■ 第二步：EM算法的E步：确定Q函数

$$Q(\theta,\theta^{(i)}) = E\left[logP(y,\gamma|\theta)\middle|y,\theta^{(i)}\right]$$

$$= E\left\{\sum_{k=1}^{K}\left\{\boldsymbol{n_k} log\alpha_k + \sum_{j=1}^{N}\gamma_{jk}\left[log\left(\frac{1}{\sqrt{2\pi}}\right) - log\sigma_k - \frac{1}{2\sigma_k^2}(y_j-\mu_k)^2\right]\right\}\right\}$$

$$= E\left\{\sum_{k=1}^{K}\left\{\left(\sum_{\boldsymbol{j=1}}^{\boldsymbol{N}}\boldsymbol{\gamma_{jk}}\right) log\alpha_k + \sum_{j=1}^{N}\gamma_{jk}\left[log\left(\frac{1}{\sqrt{2\pi}}\right) - log\sigma_k - \frac{1}{2\sigma_k^2}(y_j-\mu_k)^2\right]\right\}\right\}$$

$$= \sum_{k=1}^{K}\left\{\left(\sum_{j=1}^{N}\boldsymbol{(E\gamma_{jk})}\right) log\alpha_k + \sum_{j=1}^{N}\boldsymbol{(E\gamma_{jk})}\left[log\left(\frac{1}{\sqrt{2\pi}}\right) - log\sigma_k - \frac{1}{2\sigma_k^2}(y_j-\mu_k)^2\right]\right\}$$

记$\hat{\gamma}_{jk} = \boldsymbol{E(\gamma_{jk}|y,\theta)}$=P$(\gamma_{jk}=1|y,\theta)$，则需要求$\hat{\gamma}_{jk}$。

用EM算法估计参数θ的推导过程

$$\hat{\gamma}_{jk} = E(\gamma_{jk}|y,\theta) = \mathrm{P}(\gamma_{jk} = 1|y,\theta)$$

$$= \frac{\mathrm{P}(\gamma_{jk} = 1,\ y_j \mid \theta)}{\sum_{k=1}^{K} \mathrm{P}(\gamma_{jk} = 1,\ y_j \mid \theta)}$$

$$= \frac{\mathrm{P}(y_j|\gamma_{jk} = 1,\theta)\mathrm{P}(\gamma_{jk} = 1|\theta)}{\sum_{k=1}^{K} \mathrm{P}(y_j|\gamma_{jk} = 1,\theta)\mathrm{P}(\gamma_{jk} = 1|\theta)}$$

$$= \frac{P(y_j|\theta_k)\alpha_k}{\sum_{k=1}^{K} P(y_j|\theta_k)\alpha_k}$$

$$= \frac{\alpha_k \phi(y_j|\theta_k)}{\sum_{k=1}^{K} \alpha_k \phi(y_j|\theta_k)} \quad j = 1,2,\dots,N; k = 1,2,\dots,K$$

P(a,b|c)表示c事件发生条件下,a和b同时发生的概率：

$p(a,b|c) = p(a|b,c)p(b|c)$

- （$\gamma_{jk} = 1,\theta$）相当于θ_k
- $\mathrm{P}(\gamma_{jk} = 1|\theta)$ 等于α_k，即选中第k个模型的概率；

$\hat{\gamma}_{jk}$是在当前模型参数下第j个观测数据来自第k个分模型的概率，称为分模型$\boldsymbol{k}$对观测数据$\boldsymbol{y_j}$的响应度

用EM算法估计参数θ的推导过程

- 将$\hat{\gamma}_{jk} = \mathrm{E}\gamma_{jk}$ 及 $n_k = \sum_{j=1}^{N} E\gamma_{jk}$带入Q函数，得

$$Q\left(\theta, \theta^{(i)}\right)$$

$$= \sum_{k=1}^{K} \left\{ \left(\sum_{j=1}^{N} (E\gamma_{jk}) \right) log\alpha_k + \sum_{j=1}^{N} (E\gamma_{jk}) \left[log\left(\frac{1}{\sqrt{2\pi}}\right) - log\sigma_k - \frac{1}{2\sigma_k^2}(y_j - \mu_k)^2 \right] \right\}$$

$$= \sum_{k=1}^{K} \left\{ n_k log\alpha_k + \sum_{j=1}^{N} \hat{\gamma}_{jk} \left[log\left(\frac{1}{\sqrt{2\pi}}\right) - log\sigma_k - \frac{1}{2\sigma_k^2}(y_j - \mu_k)^2 \right] \right\}$$

■ 第三步：EM算法的M步

M步是求函数$Q\left(\theta, \theta^{(i)}\right)$对$\theta$的极大值，即求新一轮迭代的模型参数：

$$\theta^{(i+1)} = arg \max_{\theta} Q\left(\theta, \theta^{(i)}\right)$$

用$\hat{\mu}_k, \hat{\sigma}_k^2$以及$\hat{\alpha}_k$， $k = 1,2,\ldots,K$表示$\theta^{(i+1)}$的各参数。

用EM算法估计参数θ的推导过程

求 $\hat{\mu}_k, \hat{\sigma}_k^2$ 只需将$Q\left(\theta, \theta^{(i)}\right)$分别对$\mu_k, \sigma_k^2$求偏导数并令其为0，即可得到；

求$\hat{\alpha}_k$是在 $\sum_{k=1}^{K} \alpha_k = 1$条件下求偏导数并令其为0得到的。

$$\hat{\mu}_k = \frac{\sum_{j=1}^{N} \hat{\gamma}_{jk} y_j}{\sum_{j=1}^{N} \hat{\gamma}_{jk}}$$

$$\hat{\sigma}_k^2 = \frac{\sum_{j=1}^{N} \hat{\gamma}_{jk} (y_j - \mu_k)^2}{\sum_{j=1}^{N} \hat{\gamma}_{jk}}$$

$$\hat{\alpha}_k = \frac{n_k}{N} = \frac{\sum_{j=1}^{N} \hat{\gamma}_{jk}}{N} \qquad k = 1,2,\dots,K$$

重复以上计算，直到对数似然函数值不再有明显的变化为止。

高斯混合模型估计的EM算法

- 输入：观测数据$y_1, y_2, \dots, y_N$，高斯混合模型；
- 输出：高斯混合模型参数

① 取参数的初始值开始迭代；

② **E步**：依据当前模型参数，计算分模型k对观测数据y_j的响应度

$$\hat{\gamma}_{jk} = \frac{\alpha_k \phi(y_j|\theta_k)}{\sum_{k=1}^{K} \alpha_k \phi(y_j|\theta_k)} \quad j = 1,2,\dots,N; k = 1,2,\dots,K$$

③ **M步**；计算新一轮迭代的模型参数

$$\hat{\mu}_k = \frac{\sum_{j=1}^{N} \hat{\gamma}_{jk} y_j}{\sum_{j=1}^{N} \hat{\gamma}_{jk}}, \quad k = 1,2,\dots,K$$

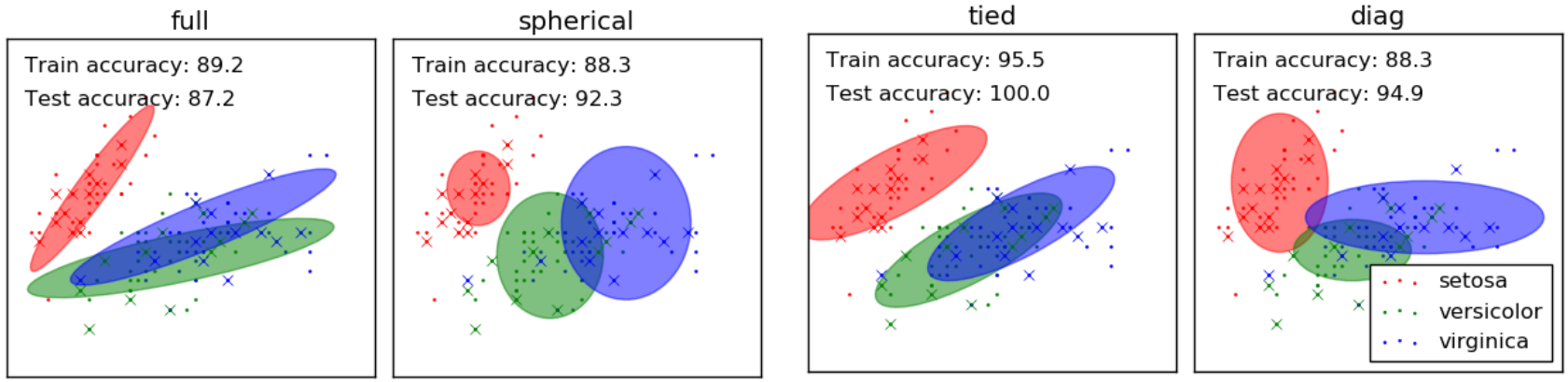
$$\hat{\sigma}_k^2 = \frac{\sum_{j=1}^{N} \hat{\gamma}_{jk} (y_j - \mu_k)^2}{\sum_{j=1}^{N} \hat{\gamma}_{jk}}, \quad k = 1,2,\dots,K$$

$$\hat{\alpha}_k = \frac{n_k}{N} = \frac{\sum_{j=1}^{N} \hat{\gamma}_{jk}}{N}, \quad k = 1,2,\dots,K$$

④ 重复第②步和第③步，直到收敛

高斯混合模型对IRIS数据聚类

- sklearn.mixture 中的 GMM 实现了期望最大化算法 expectation-maximization (EM) 用于拟合混合高斯模型.
- GMM.fit 方法用于拟合训练数据，学习Gaussian Mixture Model.
- GMM.predict 方法计算每个样本属于每个高斯类的概率.



模型选择准则之AIC和BIC

- 很多机器学习中的参数估计问题采用似然函数作为目标函数。其好处是，当训练数据足够多时，可以不断提高模型精度。但如果模型复杂度过高，容易发生过拟合。
- 所以，模型选择问题在模型复杂度与模型对数据集描述能力（即似然函数）之间寻求最佳平衡。
- 人们提出许多信息准则，通过加入模型复杂度的惩罚项来避免过拟合问题（因此，这里只是考虑模型参数数量，不涉及模型结构的选择）。这里介绍两种模型选择方法：
  - 赤池信息准则（Akaike Information Criterion，AIC）
  - 贝叶斯信息准则（Bayesian Information Criterion，BIC）。

AIC赤池信息准则（Akaike Information Criterion，AIC）

- AIC是衡量统计模型拟合优良性的一种标准（Hirotugu Akaike，1974）
- 通常情况下，AIC定义为：

$$AIC = 2k - 2ln(\hat{L})$$

其中k是模型参数个数，$\hat{L}$是似然函数的最大值。

- 目标是选取AIC最小的模型：AIC不仅要提高模型拟合度（极大似然），而且引入了惩罚项，使模型参数尽可能少，有助于降低过拟合的可能性：
  - 当两个模型之间存在较大差异时，差异主要体现在似然函数项，当似然函数差异不显著时，上式第一项，即模型复杂度则起作用，从而参数个数少的模型是较好的选择。
  - 一般而言，当模型复杂度提高（k增大）时，似然函数$\hat{L}$也会增大，从而使AIC变小，但是k过大时，似然函数增速减缓，导致AIC增大，模型过于复杂容易造成过拟合现象。

BIC（Bayesian InformationCriterion）贝叶斯信息准则

- BIC与AIC相似，用于模型选择，1978年由Schwarz提出。
- BIC的惩罚项比AIC的大，考虑了样本数量n.

$$BIC = \mathbf{ln(n)}\, k - 2\ln(\hat{L})$$

其中，

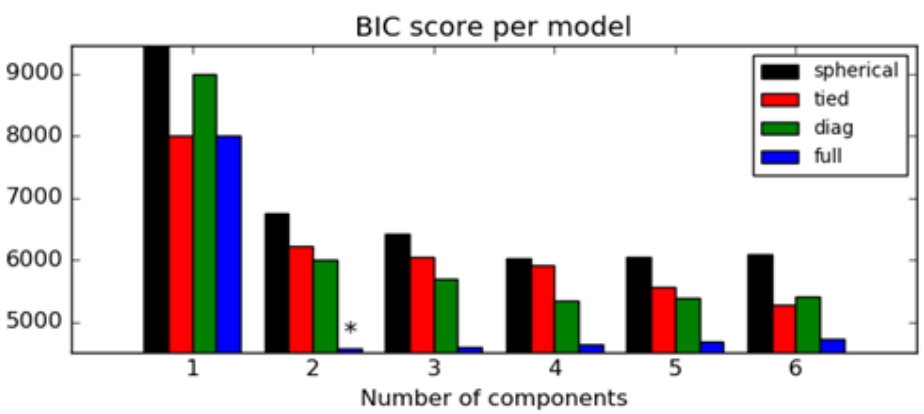
- k为模型参数个数，
- n为样本数量，
- $\hat{L}$为似然函数的最大值。即 $\hat{L} = p(x|\hat{\theta}, M)$, $\hat{\theta}$ 是是似然函数最大的参数取值。

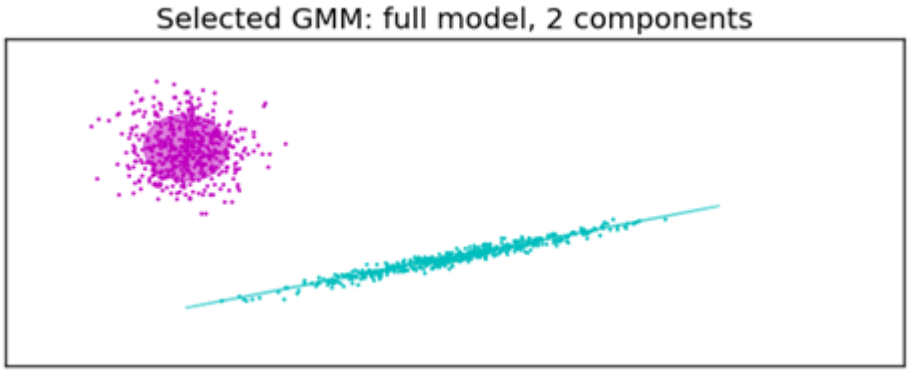
- 通常选择BIC最小的模型。虽然对于不同模型，$\mathbf{ln(n)}$都相等，但k不等，因此$\mathbf{ln(n)}\, k$还是不等，对复杂模型的惩罚更大。

Gaussian Mixture Model Selection

Python source code: plot_gmm_selection.py

- 本例显示了可以使用information-theoretic criteria (BIC)进行模型选择。
- 模型选择包括选择组件的个数，以及全协方差矩阵covariance 。本例中采用BIC，当然用AIC也可以。
- 本例中，2个组件 components ，以及全协方差矩阵的模型被选择。实际上，这也是用于生成数据的模型配置。





聚类问题

本章小结

- 什么是聚类
- 聚类算法评价指标
- 常见的聚类算法
  - K-means聚类算法
  - 吸引子传播聚类算法
  - 层次聚类
  - DBSCAN密度聚类算法
  - 均值偏移聚类算法
  - 自组织映射聚类算法
  - 高斯混合模型聚类算法