第5章

聚类问题

本章学习要点

- 了解聚类问题的基本概念、概率模型以及与分类问题的异同点。
- 了解并掌握含隐变量概率模型及其 EM 求解算法。
- 了解并掌握高斯混合模型、三硬币模型这两种常用含隐变量的概率模型。
- 了解 k 均值聚类、DBSCAN 聚类这两种常用的聚类方法。

聚类(clustering)是一种特殊的分类问题。分类是根据有标注数据集来训练模型,学习人们预先设定的类别概念,属于**有监督学习**;而聚类则是根据无标注(仅包含特征)数据集训练模型,即根据数据自身的分布特性或结构,自动将数据聚集成**簇**(cluster),形成类别概念,它属于**无监督学习**。

聚类问题可形式化描述为:给定无标注数据集 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$,其中 $\boldsymbol{x}_i$ 是 d 维样本特征,m 为样本容量,然后设计聚类算法训练模型,将数据集 D 划分成 k 个不相交的簇 $\{C_1,C_2,\cdots,C_k\}$,其中

$$D=\bigcup_{i=1}^{k} C_i, \quad \text{且若 } i\neq j,\text{则 } C_i\cap C_j=\Phi$$

今后任给新的样本特征 $\boldsymbol{x}$,也可以通过聚类模型将其划归某一类簇。

5.1 聚类问题的提出

在分类问题中,如果因为标注训练集的工作量过大,或问题本身并没有预设的类别概念,则分类问题就演变成了聚类问题。为便于后续讲解,这里先对概率符号做一下简化,将离散型概率分布 $P(X=\boldsymbol{x})$或连续型概率密度 $p(\boldsymbol{x})$统一记作 $P(\boldsymbol{x})$,例如

$$P(X=\boldsymbol{x})\equiv P(\boldsymbol{x}),\quad P(X=\boldsymbol{x},Y)\equiv P(\boldsymbol{x},Y),\quad P(Y\mid X=\boldsymbol{x})\equiv P(Y\mid \boldsymbol{x})$$
$$P(Y=y)\equiv P(y),\quad P(X,Y=y)\equiv P(X,y),\quad P(Y=y\mid X)\equiv P(y\mid X)$$

$$P(X)=\int_{\Omega_Y} p(X,y;\boldsymbol{\theta})\mathrm{d}y \equiv \sum_{Y} P(X,Y;\boldsymbol{\theta}) \equiv \sum_{y\in\Omega_Y} P(X,y;\boldsymbol{\theta})$$

请读者根据上下文加以理解。下面分别给出分类问题、聚类问题的形式化描述。

5.1.1　分类问题概述

在分类问题中，假设有 n 个类别 $\{c_1,c_2,\cdots,c_n\}$，将分类特征记作随机变量 X，类别记作随机变量 Y。类别 Y 是离散型随机变量，其值域为 $\Omega_Y=\{c_1,c_2,\cdots,c_n\}$，概率分布（先验概率）为多项分布，可记作 $P(Y;\boldsymbol{\alpha})$，即

$$P(Y=c_k)=\alpha_k,\quad \text{记作 } P(Y_k;\alpha_k),\quad k=1,2,\cdots,n \tag{5-1}$$

其中 $\boldsymbol{\alpha}=(\alpha_1,\alpha_2,\cdots,\alpha_n)$ 为未知参数，且 $\sum_{k=1}^{n}\alpha_k=1$。再假设各类特征具有相同的概率分布形式，但所取参数 $\boldsymbol{\beta}_k$ 不同，将它们记作

$$P(X\mid Y_k;\boldsymbol{\beta}_k),\quad k=1,2,\cdots,n \tag{5-2}$$

其中，$\boldsymbol{\beta}=(\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\cdots,\boldsymbol{\beta}_n)$ 为未知参数。

可以根据类别的先验概率分布 $P(Y;\boldsymbol{\alpha})$ 和各类的特征概率分布 $P(X\mid Y_k;\boldsymbol{\beta}_k)$ 设计贝叶斯分类器。给定特征 X，将类别 Y 判定为后验概率 $P(Y_k\mid X)$ 最大的 c_k，其分类判别函数为

$$k^{*}=\underset{k=1,2,\cdots,n}{\operatorname{argmax}} P(Y_k\mid X) \tag{5-3}$$

或其等价形式

$$k^{*}=\underset{k=1,2,\cdots,n}{\operatorname{argmax}} P(X,Y_k)=\underset{k=1,2,\cdots,n}{\operatorname{argmax}} P(X\mid Y_k)P(Y_k) \tag{5-4}$$

为明确未知参数，可在式(5-4)中标出 $\boldsymbol{\alpha}$、$\boldsymbol{\beta}$，即

$$k^{*}=\underset{k=1,2,\cdots,n}{\operatorname{argmax}} P(X,Y_k;\alpha_k,\boldsymbol{\beta}_k)=\underset{k=1,2,\cdots,n}{\operatorname{argmax}} P(X\mid Y_k;\boldsymbol{\beta}_k)P(Y_k;\alpha_k) \tag{5-5}$$

可以看出，分类问题的关键是如何确定未知参数 $\boldsymbol{\alpha}$、$\boldsymbol{\beta}$。

给定数据集 $D_{\text{train}}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\}$，其样本容量为 m，可以使用极大似然估计方法分别估计出参数 $\boldsymbol{\alpha}$ 和 $\boldsymbol{\beta}$。对于参数 $\boldsymbol{\alpha}$，若 D_{train} 中类别取值为 c_k 的样本点个数等于 m_k，则使用极大似然估计方法即可估计出最优参数 $\boldsymbol{\alpha}^{*}$ 为

$$\alpha_k^{*}=\frac{m_k}{m},\quad P(Y_k)=\alpha_k^{*},\quad k=1,2,\cdots,n \tag{5-6}$$

对于参数 $\boldsymbol{\beta}$，记 D_{train} 中类别取值为 c_k 样本点所构成的子集为 D_k。基于子集 D_k，使用极大似然估计方法可以估计第 k 类特征概率分布（离散型）或概率密度函数（连续型）$P(X\mid Y_k;\boldsymbol{\beta}_k)$ 中的最优参数 $\boldsymbol{\beta}_k^{*}$（例如正态分布中的均值 μ_k 和方差 σ_k^2），其似然函数为

$$L(\boldsymbol{\beta}_k)=\prod_{j=1}^{m_k} P(\boldsymbol{x}_j\mid Y_k;\boldsymbol{\beta}_k),\quad k=1,2,\cdots,n \tag{5-7}$$

最大化式(5-7)的对数似然函数 $\ln L(\boldsymbol{\beta}_k)$，得

$$\boldsymbol{\beta}_k^{*}=\underset{\boldsymbol{\beta}_k}{\operatorname{argmax}} \ln L(\boldsymbol{\beta}_k)=\underset{\boldsymbol{\beta}_k}{\operatorname{argmax}} \sum_{j=1}^{m_k} \ln P(\boldsymbol{x}_j\mid Y_k;\boldsymbol{\beta}_k),\quad k=1,2,\cdots,n \tag{5-8}$$

在估计出最优参数 $\boldsymbol{\alpha}^{*}=(\alpha_1^{*},\alpha_2^{*},\cdots,\alpha_n^{*})$、$\boldsymbol{\beta}^{*}=(\boldsymbol{\beta}_1^{*},\boldsymbol{\beta}_2^{*},\cdots,\boldsymbol{\beta}_n^{*})$ 之后，给定任意新样本特

征 $\boldsymbol{x}$，可以按式(5-5)的分类判别函数进行分类，即

$$k^* = \underset{k=1,2,\cdots,n}{\operatorname{argmax}} P(\boldsymbol{x} \mid Y_k; \boldsymbol{\beta}_k^*) P(Y_k; \alpha_k^*) \tag{5-9}$$

总结如下，分类问题就是使用包含类别标注的样本数据集 $D_{\text{train}}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\}$来训练概率模型中的参数$\boldsymbol{\alpha}$、$\boldsymbol{\beta}$；然后按照最大后验概率原则设计贝叶斯分类器；今后任给新的样本特征 $\boldsymbol{x}$ 都可以使用该分类器进行分类。

5.1.2 聚类问题概述

设计式(5-5)的贝叶斯分类器，其关键是如何确定概率模型中的未知参数$\boldsymbol{\alpha}$、$\boldsymbol{\beta}$。如果给定数据集做过标注(通常为人工标注)，即数据集 $D_{\text{train}}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\}$中的类别标注 Y 已知，则可以使用极大似然估计方法分别估计出最优参数$\boldsymbol{\alpha}^*$和$\boldsymbol{\beta}^*$。

如果数据集未做标注(通常是因为工作量过大或难以标注)，即数据集 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$只包含各样本点分类特征 X，其对应的类别标注 Y 未知，称分类特征 X 是**可观测**的变量，而类别 Y 则是**不可观测**的变量(一般称为**隐变量**，hidden variable；或**潜变量**，latent variable)。如果概率模型含有不可观测的隐变量，这时机器学习该如何确定模型中的未知参数呢？因为数据集未做标注，无法采用式(5-6)、式(5-8)的极大似然估计方法来估计最优参数$\boldsymbol{\alpha}^*$、$\boldsymbol{\beta}^*$。

对于 n 个类别$\{c_1,c_2,\cdots,c_n\}$的分类问题，特征 X 的概率分布 $P(X;\boldsymbol{\alpha},\boldsymbol{\beta})$是联合概率分布 $P(X,Y;\boldsymbol{\alpha},\boldsymbol{\beta})$关于特征 X 的边缘概率，即

$$P(X;\boldsymbol{\alpha},\boldsymbol{\beta}) = \sum_{k=1}^{n} P(X \mid Y_k;\boldsymbol{\beta}_k) P(Y_k;\alpha_k) = \sum_{k=1}^{n} P(X,Y_k;\alpha_k,\boldsymbol{\beta}_k) \tag{5-10}$$

其中，$\boldsymbol{\alpha}=(\alpha_1,\alpha_2,\cdots,\alpha_n)$、$\boldsymbol{\beta}=(\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\cdots,\boldsymbol{\beta}_n)$为未知参数。为了进行更一般化的讨论，记$\boldsymbol{\theta}=(\boldsymbol{\alpha},\boldsymbol{\beta})$，类别 Y 的值域为Ω_Y，则式(5-10)可改写为

$$P(X;\boldsymbol{\theta}) = \sum_{Y} P(X,Y;\boldsymbol{\theta}) \equiv \sum_{y\in\Omega_Y} P(X,y;\boldsymbol{\theta}) \tag{5-11}$$

给定未做标注的数据集 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$，可将该数据集的似然函数定义为

$$L(\boldsymbol{\theta}) = \prod_{j=1}^{m} P(\boldsymbol{x}_j;\boldsymbol{\theta}) = \prod_{j=1}^{m} \Big(\sum_{y\in\Omega_Y} P(\boldsymbol{x}_j,y;\boldsymbol{\theta}) \Big) \tag{5-12}$$

对式(5-12)取自然对数，则数据集 D 的对数似然函数为

$$\ln L(\boldsymbol{\theta}) = \ln\Big(\prod_{j=1}^{m} P(\boldsymbol{x}_j;\boldsymbol{\theta}) \Big) = \sum_{j=1}^{m} \ln\Big(\sum_{y\in\Omega_Y} P(\boldsymbol{x}_j,y;\boldsymbol{\theta}) \Big) \tag{5-13}$$

最大化式(5-13)的对数似然函数，求解最优参数$\boldsymbol{\theta}^*$，即

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \ln L(\boldsymbol{\theta}) \tag{5-14}$$

在确定出最优参数$\boldsymbol{\theta}^*=(\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*)$之后，即可确定问题的概率模型，并根据式(5-9)的分类判别函数将数据集 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$中的 m 个样本数据划分成不同的类，这就是**聚类**。

聚类与分类的区别在于：聚类问题中没有预设类别，它是由机器学习算法按照数据自身(即样本特征)的分布特性自动提炼出来的；而分类问题中的类别通常是由人工指定的。

类别是一种“概念”，是人们在长期的生产、生活实践中总结出来的。进入信息化时代，很多问题通常是先有数据或只有数据，可通过聚类方法对问题进行研究，这就是聚类分析。例如，网上消费产生了很多数据，这时可以通过聚类方法对消费者或消费行为进行研究分析。

在聚类问题中，因为数据集 D 不能提供各样本点的类别标注，即式(5-14)的对数似然函数 $\ln L(\boldsymbol{\theta})$ 包含隐变量 Y，这是一种含隐变量的最优化问题。如何设计求解含隐变量的最优化算法，这是聚类分析的关键。

总结如下，聚类问题就是使用未做标注的样本数据集 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$ 来训练概率模型中的参数 $\boldsymbol{\alpha}$、$\boldsymbol{\beta}$，即含隐变量的参数估计；然后再按最大后验概率原则设计贝叶斯分类器，将样本数据集中的样本数据划分成不同的类。

5.1.3　混合概率模型及其参数估计问题

对于 n 个类别 $\{c_1,c_2,\cdots,c_n\}$ 的分类或聚类问题，将分类特征记作随机变量 X，类别记作随机变量 Y，则 X、Y 的联合概率分布 $P(X,Y)$ 即为分类或聚类问题的概率模型。联合概率分布 $P(X,Y)$ 还可表示为

$$P(X,Y)\equiv P(X,Y;\boldsymbol{\alpha},\boldsymbol{\beta})=P(X\mid Y;\boldsymbol{\beta})P(Y;\boldsymbol{\alpha}) \tag{5-15}$$

其中，$\boldsymbol{\alpha}=(\alpha_1,\alpha_2,\cdots,\alpha_n)$、$\boldsymbol{\beta}=(\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\cdots,\boldsymbol{\beta}_n)$ 为概率模型中的未知参数；$P(Y;\boldsymbol{\alpha})$ 表示类别概率分布，即 $P(Y=c_k)=\alpha_k$，$k=1,2,\cdots,n$；$P(X|Y;\boldsymbol{\beta})$ 表示各类的特征概率分布，它们的分布形式相同但所取参数不同，可记作 $P(X|Y_k;\boldsymbol{\beta}_k)$，$k=1,2,\cdots,n$。

在联合概率分布 $P(X,Y;\boldsymbol{\alpha},\boldsymbol{\beta})$ 中，边缘概率分布 $P(X;\boldsymbol{\alpha},\boldsymbol{\beta})$ 可看作分类特征 X 的概率模型，它可表示为

$$\begin{aligned}P(X;\boldsymbol{\alpha},\boldsymbol{\beta})&=\sum_{k=1}^{n}P(X\mid Y_k;\boldsymbol{\beta}_k)P(Y_k;\alpha_k)\\&=\alpha_1P(X\mid Y_1;\boldsymbol{\beta}_1)+\alpha_2P(X\mid Y_2;\boldsymbol{\beta}_2)+\cdots+\alpha_nP(X\mid Y_n;\boldsymbol{\beta}_n)\end{aligned} \tag{5-16}$$

可以看出，分类特征 X 的概率模型 $P(X;\boldsymbol{\alpha},\boldsymbol{\beta})$ 是由 n 个概率模型 $P(X|Y_1;\boldsymbol{\beta}_1)$、$P(X|Y_2;\boldsymbol{\beta}_2)$、……、$P(X|Y_n;\beta_n)$ 组成的一个混合概率模型，概率 α_1、α_2、……、α_n 也被称作**混合系数**，$\sum_{k=1}^{n}\alpha_k=1$。

如果已知参数 $\boldsymbol{\alpha}=(\alpha_1,\alpha_2,\cdots,\alpha_n)$、$\boldsymbol{\beta}=(\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\cdots,\boldsymbol{\beta}_n)$，则按照式(5-16)的混合概率模型产生样本 $\boldsymbol{x}$ 的过程是：首先按照概率 $\alpha_1,\alpha_2,\cdots,\alpha_n$ 选择模型，假设选择第 k 个模型；然后按第 k 个模型的概率分布 $P(X|Y_k;\boldsymbol{\beta}_k)$ 生成样本 $\boldsymbol{x}$。反过来，如果给定样本数据集 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$，希望估计概率模型 $P(X;\boldsymbol{\alpha},\boldsymbol{\beta})$ 中的未知参数 $\boldsymbol{\alpha}$、$\boldsymbol{\beta}$，这就是混合概率模型的参数估计问题。

混合概率模型的参数估计问题与聚类问题非常相似，其概率模型在本质上是一样的，而且它们用于参数估计的样本数据集 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$ 都不包含类别标注。所不同的是，聚类问题在估计出模型参数后，还需进一步设计贝叶斯分类器，将数据集 D 中的样本数据划分成不同的类。聚类问题、混合概率模型的参数估计问题都属于含隐变量的最优化问题，求解这样的问题通常使用 EM 算法。

5.2 EM 算法

EM 算法是一种迭代算法，主要用于求解含隐变量的最优化问题。任给初始参数θ^0，EM 算法的关键步骤是：第 i 次迭代时如何将参数从θ^{i-1} 更新到θ^i，使得对数似然函数 $\ln l(\theta)=\ln P(X;\theta)$的函数值逐步上升，即 $\ln l(\theta^i)\geqslant\ln l(\theta^{i-1})$，最终收敛至最大值。

5.2.1 EM 算法原理

1. 问题描述

假设随机变量 X、Y 服从参数为θ 的概率分布 $P(X,Y;\theta)$，其中 Y 为不可观测或未被观测的隐变量。将随机变量 Y(例如类别)的值域记作 Ω_Y，则随机变量 X(例如分类特征)的边缘概率为

$$P(X;\theta)=\sum_Y P(X,Y;\theta)\equiv\sum_{y\in\Omega_Y}P(X,y;\theta)$$

或

$$P(X;\theta)=\sum_Y P(X\mid Y;\theta)P(Y;\theta)\equiv\sum_{y\in\Omega_Y}P(X\mid y;\theta)P(y;\theta)$$

给定观测样本 $X=\boldsymbol{x}$，其似然函数 $l(\theta)$可定义为

$$l(\theta)=P(\boldsymbol{x};\theta)=\sum_{y\in\Omega_Y}P(\boldsymbol{x},y;\theta)$$

其中，$P(\boldsymbol{x};\theta)$被称作**不完全数据**(incomplete-data)$\boldsymbol{x}$ 的似然函数，$P(\boldsymbol{x},y;\theta)$被称作**完全数据**(complete-data)$(\boldsymbol{x},y)$的似然函数。将观测样本 $X=\boldsymbol{x}$ 的对数似然函数定义为

$$\ln l(\theta)=\ln P(\boldsymbol{x};\theta)=\ln\Big(\sum_Y P(\boldsymbol{x},Y;\theta)\Big)\equiv\ln\Big(\sum_{y\in\Omega_Y}P(\boldsymbol{x},y;\theta)\Big)\tag{5-17}$$

最大化式(5-17)的对数似然函数，求解最优参数θ^*，即

$$\theta^*=\arg\max_\theta\ln l(\theta)\tag{5-18}$$

因为对数似然函数 $\ln l(\theta)$包含隐变量 Y，因此式(5-18)是一种含隐变量的极大似然估计问题，需通过特殊的 EM 算法进行求解。

2. 算法准备：Jensen 不等式

根据 Jensen 不等式，设 X 为随机变量，$E(X)$为 X 的期望，$E(f(X))$为 $f(X)$的期望，则

(1) 对任意凸函数 f，有 $f(E(X))\leqslant E(f(X))$。

(2) 对任意凹函数 f，有 $f(E(X))\geqslant E(f(X))$。

(3) 如果 X 为常量，则上述两个不等式均取等号，即如果 $X=c$，则 $f(X)=f(c)$也为常量，因此有 $f(E(X))=f(c)=E(f(X))$。

Jensen 不等式有多种不同的表述形式，这里再给出其中的一种：若函数 f 为凸集 D 上的凹函数，则对于任意 n 个点 $x_j\in D$ 及实数 $p_j(0\leqslant p_j\leqslant 1)$，$j=1,2,\cdots,n$，且 $\sum_{j=1}^{n}p_j=1$，都有

$$f\left(\sum_{j=1}^{n} p_j x_j\right) \geqslant \sum_{j=1}^{n} p_j f(x_j) \tag{5-19}$$

如果 $x_1=x_2=\cdots=x_n=c$，则上述不等式取等号。

若将式(5-19)应用于随机变量 X、Y 的联合概率分布，则

- 当取 $p_j=P(Y|X)$，$x_j=P(X,Y)$，f 为自然对数(凹函数)时

$$\ln\left(\sum_{Y} P(Y \mid X)P(X,Y)\right) \geqslant \sum_{Y} P(Y \mid X)\ln P(X,Y) \tag{5-20}$$

- 当取 $p_j=P(Y|X)$，$x_1=x_2=\cdots=x_n=P(X)$，f 为自然对数(凹函数)时

$$\ln\left(\sum_{Y} P(Y \mid X)P(X)\right) = \sum_{Y} P(Y \mid X)\ln P(X) \tag{5-21}$$

其中，边缘概率 $P(X)$ 是与 Y 取值无关的量(相当于常量 c)。

3. 设计迭代算法

任给初始参数 θ^0，设计求解式(5-18)的迭代算法，其关键步骤是：第 i 次迭代时如何将参数从 θ^{i-1} 更新到 θ^i，使得对数似然函数 $\ln l(\theta)$ 的函数值上升，即 $\ln l(\theta^i)\geqslant\ln l(\theta^{i-1})$。

1) 根据参数 θ^{i-1} 确定概率模型

因为上次迭代的参数 θ^{i-1} 为已知，据此可确定联合概率分布 $P(X,Y;\theta^{i-1})$、边缘概率分布 $P(X;\theta^{i-1})$ 和后验概率 $P(Y|X;\theta^{i-1})$。这三者之间的关系为

$$P(X,Y;\theta^{i-1}) = P(Y \mid X;\theta^{i-1})P(X;\theta^{i-1}) \tag{5-22}$$

2) 确定对数似然函数 $\ln l(\theta)$ 的下界

对式(5-17)的对数似然函数 $\ln l(\theta)$ 做如下等价变换：

$$\ln l(\theta) = \ln\left(\sum_{Y} P(X,Y;\theta)\right) = \ln\left(\sum_{Y} P(Y \mid X;\theta^{i-1})\left(\frac{P(X,Y;\theta)}{P(Y \mid X;\theta^{i-1})}\right)\right) \tag{5-23}$$

根据式(5-19)的 Jensen 不等式，如果取

$$p_j = P(Y \mid X;\theta^{i-1}), \quad x_j = \left(\frac{P(X,Y;\theta)}{P(Y \mid X;\theta^{i-1})}\right)$$

则有

$$\ln l(\theta) \geqslant \sum_{Y} P(Y \mid X;\theta^{i-1})\ln\left(\frac{P(X,Y;\theta)}{P(Y \mid X;\theta^{i-1})}\right) \tag{5-24}$$

不等式(5-24)右边的项是对数似然函数 $\ln l(\theta)$ 的下界，可记作**下界函数 B**(θ,θ^{i-1})，或称作**证据下界目标**(Evidence Lower Bound Objective，ELBO)，即

$$B(\theta,\theta^{i-1}) = \sum_{Y} P(Y \mid X;\theta^{i-1})\ln\left(\frac{P(X,Y;\theta)}{P(Y \mid X;\theta^{i-1})}\right) \tag{5-25}$$

将优化目标由最大化式(5-18)的对数似然函数改为最大化式(5-25)的下界函数 $B(\theta,\theta^{i-1})$，求解最优参数 θ^*，即

$$\theta^* = \arg\max_{\theta} B(\theta,\theta^{i-1}) \tag{5-26}$$

其中，θ^{i-1} 是上一轮迭代的参数，为已知量。将 θ^* 作为迭代算法的新参数 θ^i，则有

$$\theta^i = \theta^*, \quad \ln l(\theta^i) = \max_{\theta} B(\theta,\theta^{i-1}) \tag{5-27}$$

3) 证明 $\ln l(\theta^i)\geqslant\ln l(\theta^{i-1})$

根据式(5-23),若取$\theta=\theta^{i-1}$,则

$$\ln l(\theta^{i-1})=\ln\Big(\sum_Y P(X,Y;\theta^{i-1})\Big)=\ln\Big(\sum_Y P(Y\mid X;\theta^{i-1})P(X;\theta^{i-1})\Big) \tag{5-28}$$

根据式(5-19)的 Jensen 不等式,如果取

$$p_j=P(Y\mid X;\theta^{i-1}),\quad x_1=x_2=\cdots=x_n=P(X;\theta^{i-1})$$

则 Jensen 不等式取等号,即

$$\begin{aligned}\ln l(\theta^{i-1})&=\ln\Big(\sum_Y P(Y\mid X;\theta^{i-1})P(X;\theta^{i-1})\Big)\\&=\sum_Y P(Y\mid X;\theta^{i-1})\ln P(X;\theta^{i-1})\\&=\sum_Y P(Y\mid X;\theta^{i-1})\ln\Big(\frac{P(X,Y;\theta^{i-1})}{P(Y\mid X;\theta^{i-1})}\Big)=B(\theta^{i-1},\theta^{i-1})\end{aligned} \tag{5-29}$$

由式(5-27)可知:

$$\ln l(\theta^i)=\max_\theta B(\theta,\theta^{i-1})\geqslant B(\theta^{i-1},\theta^{i-1})=\ln l(\theta^{i-1}) \tag{5-30}$$

4) Q 函数

对于式(5-26)的下界函数 $B(\theta,\theta^{i-1})$的最优化问题做进一步整理:

$$\begin{aligned}\theta^*&=\arg\max_\theta B(\theta,\theta^{i-1})=\arg\max_\theta\sum_Y P(Y\mid X;\theta^{i-1})\ln\Big(\frac{P(X,Y;\theta)}{P(Y\mid X;\theta^{i-1})}\Big)\\&=\arg\max_\theta\Big[\sum_Y P(Y\mid X;\theta^{i-1})\ln P(X,Y;\theta)-\sum_Y P(Y\mid X;\theta^{i-1})\ln P(Y\mid X;\theta^{i-1})\Big]\\&=\arg\max_\theta\sum_Y P(Y\mid X;\theta^{i-1})\ln P(X,Y;\theta)\end{aligned} \tag{5-31}$$

记

$$Q(\theta,\theta^{i-1})=\sum_Y P(Y\mid X;\theta^{i-1})\ln P(X,Y;\theta) \tag{5-32}$$

其中,函数 $Q(\theta,\theta^{i-1})$可看作完全数据(X,Y)的对数似然函数 $\ln P(X,Y;\theta)$在概率分布 $P(Y|X;\theta^{i-1})$下的期望,简称 **Q 函数**。

式(5-26)的下界函数 $B(\theta,\theta^{i-1})$的最优化问题与函数 $Q(\theta,\theta^{i-1})$的最优化问题等价,即

$$\theta^*=\arg\max_\theta B(\theta,\theta^{i-1})=\arg\max_\theta Q(\theta,\theta^{i-1}) \tag{5-33}$$

式(5-33)是对式(5-26)的简化,将其最优参数θ^*作为迭代算法的新参数θ^i,同样有

$$\ln l(\theta^i)\geqslant\ln l(\theta^{i-1})$$

4. EM 算法步骤

假设随机变量 X、Y 服从参数为θ 的概率分布 $P(X,Y;\theta)$,其中 Y 为不可观测或未被观测的隐变量。如果已知联合概率 $P(X,Y;\theta)$的分布形式,或其等价形式,例如已知边缘概率 $P(Y;\theta)$、条件概率 $P(X|Y;\theta)$的分布形式,但分布中的参数θ 未知。这时可以根据不完全数据 X 的样本,使用 EM 算法估计参数θ,从而建立起随机变量 X、Y 的概率模型 $P(X,Y;\theta)$。

给定未做标注的数据集 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$，其对数似然函数为

$$\ln L(\boldsymbol{\theta})=\ln\Big(\prod_{j=1}^{m}P(\boldsymbol{x}_j;\boldsymbol{\theta})\Big)=\sum_{j=1}^{m}\ln\Big(\sum_{y\in\Omega_Y}P(\boldsymbol{x}_j,y;\boldsymbol{\theta})\Big) \tag{5-34}$$

使用 EM 算法最大化式(5-34)的对数似然函数 $\ln L(\boldsymbol{\theta})$，求解最优参数 $\boldsymbol{\theta}^*$，其算法步骤为：首先选择初始参数 $\boldsymbol{\theta}^0$，然后迭代执行如下的 E 步和 M 步(EM 算法的名称正是来源于此)，直至收敛(例如迭代后参数 $\boldsymbol{\theta}^i$ 与 $\boldsymbol{\theta}^{i-1}$ 无明显变化)。

E 步：根据上次迭代的参数 $\boldsymbol{\theta}^{i-1}$，求完全数据 (X,Y) 的对数似然函数 $\ln P(X,Y;\boldsymbol{\theta})$ 在概率分布 $P(Y|X;\boldsymbol{\theta}^{i-1})$ 下的期望。这里的 E 指的就是**期望**(expectation)，也即 Q 函数。数据集 D 上的 Q 函数为

$$Q(\boldsymbol{\theta},\boldsymbol{\theta}^{i-1})=\sum_{j=1}^{m}\Big(\sum_{y\in\Omega_Y}P(y\mid\boldsymbol{x}_j;\boldsymbol{\theta}^{i-1})\ln P(\boldsymbol{x}_j,y;\boldsymbol{\theta})\Big) \tag{5-35}$$

其中，$\boldsymbol{\theta}^{i-1}$ 为已知参数，$\boldsymbol{\theta}$ 为待求解参数。

M 步：最大化期望。这里的 M 指的就是**最大化**(maximization)期望，即最大化 Q 函数，然后将其最优参数作为迭代后的新参数 $\boldsymbol{\theta}^i$，即

$$\boldsymbol{\theta}^i=\arg\max_{\boldsymbol{\theta}}Q(\boldsymbol{\theta},\boldsymbol{\theta}^{i-1}) \tag{5-36}$$

算法收敛后，将最后一次迭代的参数 $\boldsymbol{\theta}^i$ 作为最优参数 $\boldsymbol{\theta}^*$。联合概率 $P(X,Y;\boldsymbol{\theta})$ 的分布形式是已知的，因此将 $\boldsymbol{\theta}^*$ 代入其中，这就建立起了随机变量 X、Y 的概率模型 $P(X,Y;\boldsymbol{\theta}^*)$。所建立的概率模型 $P(X,Y;\boldsymbol{\theta}^*)$ 今后可以用于聚类、分类或其他用途。

5.2.2　高斯混合模型

高斯混合模型(Gaussian Mixture Model，GMM)是一种广泛使用的概率模型。该模型由多个高斯分布(即正态分布)混合而成，需使用 EM 算法来估计模型参数(或称训练模型)。

1. GMM 模型描述

高斯混合模型假设有 n 个类别 $\{c_1,c_2,\cdots,c_n\}$，将分类特征记作随机变量 X，类别记作随机变量 Y。其中，类别 Y 是不可观测的隐变量且服从多项分布，其概率分布 $P(Y;\boldsymbol{\alpha})$ 为

$$P(Y=c_k)=\alpha_k,\quad \text{记作 } P(Y_k;\alpha_k),\quad k=1,2,\cdots,n \tag{5-37}$$

其中，$\boldsymbol{\alpha}=(\alpha_1,\alpha_2,\cdots,\alpha_n)$ 为未知参数，且 $\sum_{k=1}^{n}\alpha_k=1$。再假设各类的特征 X 都服从高斯分布(即正态分布)，但所取参数 $\boldsymbol{\beta}_k=(\mu_k,\sigma_k^2)$ 不同，它们的概率密度函数可记作

$$p(x\mid Y_k;\boldsymbol{\beta}_k)=\frac{1}{\sqrt{2\pi}\sigma_k}\mathrm{e}^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}},\quad k=1,2,\cdots,n \tag{5-38}$$

其中，$\boldsymbol{\beta}=(\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\cdots,\boldsymbol{\beta}_n)$ 为未知参数。

由类别概率分布(先验概率)$P(Y;\boldsymbol{\alpha})$ 和各类的特征概率分布 $P(X|Y_k;\boldsymbol{\beta}_k)$，可以计算出联合概率 $P(X,Y;\boldsymbol{\alpha},\boldsymbol{\beta})$、分类特征 X 的边缘概率 $P(X;\boldsymbol{\alpha},\boldsymbol{\beta})$、类别的条件概率(后验概率)$P(Y|X;\boldsymbol{\alpha},\boldsymbol{\beta})$，即

$$P(X,Y_k;\alpha_k,\boldsymbol{\beta}_k)=P(X\mid Y_k;\boldsymbol{\beta}_k)P(Y_k;\alpha_k),\quad k=1,2,\cdots,n \tag{5-39}$$

$$P(X;\boldsymbol{\alpha},\boldsymbol{\beta})=\sum_{k=1}^{n}P(X\mid Y_k;\boldsymbol{\beta}_k)P(Y_k;\alpha_k)$$
$$=\alpha_1P(X\mid Y_1;\boldsymbol{\beta}_1)+\alpha_2P(X\mid Y_2;\boldsymbol{\beta}_2)+\cdots+\alpha_nP(X\mid Y_n;\boldsymbol{\beta}_n) \tag{5-40}$$

$$P(Y_k\mid X;\boldsymbol{\alpha},\boldsymbol{\beta})=\frac{P(X,Y_k;\alpha_k,\boldsymbol{\beta}_k)}{P(X;\boldsymbol{\alpha},\boldsymbol{\beta})},\quad k=1,2,\cdots,n \tag{5-41}$$

给定未做标注的数据集 $D=\{x_1,x_2,\cdots,x_m\}$,其中只包含分类特征 X,未包含类别标注 Y,因此是一个不完全数据集。现在希望根据数据集 D 建立特征 X 的概率模型 $P(X;\boldsymbol{\alpha},\boldsymbol{\beta})$,这是一个含隐变量的混合概率模型(即 GMM),其概率分布形式已知,但参数$\boldsymbol{\alpha}$、$\boldsymbol{\beta}$未知。

2. GMM 模型参数估计

给定未做标注的数据集 $D=\{x_1,x_2,\cdots,x_m\}$,其对数似然函数为

$$\ln L(\boldsymbol{\alpha},\boldsymbol{\beta})=\ln\Big(\prod_{j=1}^{m}p(x_j;\boldsymbol{\alpha},\boldsymbol{\beta})\Big)$$
$$=\ln\Big(\prod_{j=1}^{m}\Big(\sum_{k=1}^{n}p(x_j\mid Y_k;\boldsymbol{\beta}_k)P(Y_k;\alpha_k)\Big)\Big) \tag{5-42}$$

将式(5-37)、式(5-38)代入式(5-42),整理可得

$$\ln L(\boldsymbol{\alpha},\boldsymbol{\beta})=\sum_{j=1}^{m}\ln\Big(\sum_{k=1}^{n}\Big(\alpha_k\cdot\frac{1}{\sqrt{2\pi}\sigma_k}\mathrm{e}^{-\frac{(x_j-\mu_k)^2}{2\sigma_k^2}}\Big)\Big) \tag{5-43}$$

最大化式(5-43)的对数似然函数 $\ln L(\boldsymbol{\alpha},\boldsymbol{\beta})$,需使用 EM 算法求解最优参数$\boldsymbol{\alpha}^*$、$\boldsymbol{\beta}^*$,从而建立起 GMM 的概率模型 $P(X;\boldsymbol{\alpha}^*,\boldsymbol{\beta}^*)$。

3. 应用 EM 算法

记$\boldsymbol{\theta}=(\boldsymbol{\alpha},\boldsymbol{\beta})$,首先选择初始参数$\boldsymbol{\theta}^0=(\boldsymbol{\alpha}^0,\boldsymbol{\beta}^0)$,然后迭代执行如下的 E 步和 M 步,直至收敛(例如迭代后参数无变化)。

E 步:根据上次迭代的参数$\boldsymbol{\theta}^{i-1}$,求数据集 D 上的 Q 函数(即期望)。

$$Q(\boldsymbol{\theta},\boldsymbol{\theta}^{i-1})=\sum_{j=1}^{m}\Big(\sum_{k=1}^{n}P(Y_k\mid x_j;\boldsymbol{\theta}^{i-1})\ln P(x_j,Y_k;\boldsymbol{\theta})\Big) \tag{5-44}$$

其中

$$P(Y_k\mid x_j;\boldsymbol{\theta}^{i-1})=\frac{P(x_j,Y_k;\boldsymbol{\theta}^{i-1})}{P(x_j;\boldsymbol{\theta}^{i-1})}=\frac{\alpha_k^{i-1}\cdot\frac{1}{\sqrt{2\pi}\sigma_k^{i-1}}\mathrm{e}^{-\frac{(x_j-\mu_k^{i-1})^2}{2(\sigma_k^{i-1})^2}}}{\sum_{l=1}^{n}\Big(\alpha_l^{i-1}\cdot\frac{1}{\sqrt{2\pi}\sigma_l^{i-1}}\mathrm{e}^{-\frac{(x_j-\mu_l^{i-1})^2}{2(\sigma_l^{i-1})^2}}\Big)} \tag{5-45}$$

$$\ln P(x_j,Y_k;\boldsymbol{\theta})=\ln\Big(\alpha_k\cdot\frac{1}{\sqrt{2\pi}\sigma_k}\mathrm{e}^{-\frac{(x_j-\mu_k)^2}{2\sigma_k^2}}\Big)$$
$$=\ln\alpha_k+\ln\frac{1}{\sqrt{2\pi}}-\frac{1}{2}\ln\sigma_k^2-\frac{1}{2\sigma_k^2}(x_j-\mu_k)^2 \tag{5-46}$$

为便于后续演算，记 $\gamma_{jk}\equiv P(Y_k\mid x_j;\theta^{i-1})$，$\gamma_{jk}$ 满足 $\sum_{k=1}^{n}\gamma_{jk}=\gamma_{j1}+\gamma_{j2}+\cdots+\gamma_{jn}=1$。将 γ_{jk} 和式(5-46)代入式(5-44)可得

$$Q(\theta,\theta^{i-1})=\sum_{j=1}^{m}\left(\sum_{k=1}^{n}\gamma_{jk}\left(\ln\alpha_k+\ln\frac{1}{\sqrt{2\pi}}-\frac{1}{2}\ln\sigma_k^2-\frac{1}{2\sigma_k^2}(x_j-\mu_k)^2\right)\right) \tag{5-47}$$

E 步结束。

M 步：最大化 Q 函数(即最大化期望)，然后将其最优参数作为迭代后的新参数 θ^i。

对式(5-47)的 Q 函数求参数 μ_k 的偏导并令其等于 0。

$$\frac{\partial Q(\theta,\theta^{i-1})}{\partial\mu_k}=\sum_{j=1}^{m}\gamma_{jk}\left(\frac{1}{\sigma_k^2}(x_j-\mu_k)\right)=0 \tag{5-48}$$

求得最优参数 μ_k 为

$$\mu_k=\frac{\sum_{j=1}^{m}\gamma_{jk}x_j}{\sum_{j=1}^{m}\gamma_{jk}},\quad k=1,2,\cdots,n \tag{5-49}$$

再对 Q 函数求参数 σ_k^2 的偏导并令其等于 0。

$$\frac{\partial Q(\theta,\theta^{i-1})}{\partial\sigma_k^2}=\sum_{j=1}^{m}\gamma_{jk}\left(-\frac{1}{2\sigma_k^2}+\frac{1}{2\sigma_k^4}(x_j-\mu_k)^2\right)=0 \tag{5-50}$$

求解最优参数 σ_k^2 为

$$\sum_{j=1}^{m}\gamma_{jk}\left(-1+\frac{1}{\sigma_k^2}(x_j-\mu_k)^2\right)=0$$

$$\sigma_k^2=\frac{\sum_{j=1}^{m}\gamma_{jk}(x_j-\mu_k)^2}{\sum_{j=1}^{m}\gamma_{jk}},\quad k=1,2,\cdots,n \tag{5-51}$$

再根据约束条件 $\sum_{k=1}^{n}\alpha_k=1$，使用拉格朗日乘子法求解最优参数 α_k。首先构造 Q 函数的拉格朗日函数 $\mathrm{LQ}(\theta,\theta^{i-1})$，则

$$\mathrm{LQ}(\theta,\theta^{i-1})=\sum_{j=1}^{m}\left(\sum_{k=1}^{n}\gamma_{jk}\left(\ln\alpha_k+\ln\frac{1}{\sqrt{2\pi}}-\frac{1}{2}\ln\sigma_k^2-\frac{1}{2\sigma_k^2}(x_j-\mu_k)^2\right)\right)+\lambda\left(\sum_{k=1}^{n}\alpha_k-1\right) \tag{5-52}$$

然后对 $\mathrm{LQ}(\theta,\theta^{i-1})$ 求参数 α_k 的偏导并令其等于 0，即

$$\frac{\partial LQ(\theta,\theta^{i-1})}{\partial\alpha_k}=\sum_{j=1}^{m}\frac{\gamma_{jk}}{\alpha_k}+\lambda=0 \tag{5-53}$$

$$\lambda\alpha_k=-\sum_{j=1}^{m}\gamma_{jk},\quad k=1,2,\cdots,n \tag{5-54}$$

将 n 个 $\lambda\alpha_k$ 累加起来，有

$$\lambda\alpha_1+\lambda\alpha_2+\cdots+\lambda\alpha_n=-\left(\sum_{j=1}^{m}\gamma_{j1}+\sum_{j=1}^{m}\gamma_{j2}+\cdots+\sum_{j=1}^{m}\gamma_{jn}\right)$$

$$\lambda(\alpha_1+\alpha_2+\cdots+\alpha_n)=-\left(\sum_{j=1}^{m}(\gamma_{j1}+\gamma_{j2}+\cdots+\gamma_{jn})\right)$$

因为 $\sum_{k=1}^{n}\alpha_k=1,\sum_{k=1}^{n}\gamma_{jk}=1$，所以 $\lambda=-m$，将 λ 代入式(5-54)可得

$$\alpha_k=\frac{1}{m}\sum_{j=1}^{m}\gamma_{jk},\quad k=1,2,\cdots,n \tag{5-55}$$

M 步结束。综合式(5-49)、式(5-51)和式(5-55)，取新的迭代参数 θ^i 为

$$(\alpha_k,\mu_k,\sigma_k^2),\quad k=1,2,\cdots,n$$

检查迭代条件，如果参数 θ^i 与 θ^{i-1} 无明显变化，或 i 达到最大迭代次数，则停止迭代；否则返回 E 步，继续下次迭代。迭代结束后，将最后一次迭代的参数 θ^i 作为最优参数 θ^*。

5.2.3 三硬币模型

高斯混合模型是连续型混合概率模型的代表，三硬币模型则是离散型混合概率模型的代表。假设有三枚硬币，分别记作 A、B、C，它们出现正面的概率分别为 α、p、q。将硬币正面记作 1，反面记作 0，然后进行如下三硬币实验：先掷硬币 A，根据结果选择硬币 B 或 C，正面选 B，反面选 C；再掷所选出的硬币 B 或 C，若为正面则记录 1，反面则记录 0；独立重复 m 次实验，所记录结果为一个 0、1 序列，记作 $\{x_1,x_2,\cdots,x_m\}$。

三硬币模型假设硬币的投掷过程不可见，只能看到最终的记录结果 $\{x_1,x_2,\cdots,x_m\}$。

1. 模型描述

将硬币 A 的投掷结果记作随机变量 Y(可看作聚类问题中不可观测的类别)，所选出硬币(B 或 C)的投掷结果记作随机变量 X(可看作聚类问题中可观测的分类特征)。随机变量 Y、X 均服从 0-1 分布，记 $Y=1$ 为 Y_1，$Y=0$ 为 Y_0；$X=1$ 为 X_1，$X=0$ 为 X_0；再记参数 $\theta=(\alpha,p,q)$，则

$$\begin{cases}P(Y_1)=\alpha,P(Y_0)=1-\alpha, & \text{记作 } P(Y_k;\theta),k=0,1\\ P(X_1\mid Y_1)=p,P(X_0\mid Y_1)=1-p, & \text{记作 } P(X\mid Y_1;\theta)\\ P(X_1\mid Y_0)=q,P(X_0\mid Y_0)=1-q, & \text{记作 } P(X\mid Y_0;\theta)\end{cases} \tag{5-56}$$

其中，$\theta=(\alpha,p,q)$为未知参数。

由概率分布 $P(Y_k;\theta)$、$P(X|Y_1;\theta)$和 $P(X|Y_0;\theta)$，可以计算出联合概率 $P(X,Y;\theta)$、X 的边缘概率 $P(X;\theta)$、Y 的条件概率 $P(Y|X;\theta)$，即

$$P(X,Y;\theta)=\begin{cases}P(X_1,Y_1;\theta)=P(X_1\mid Y_1;\theta)P(Y_1;\theta)=\alpha p\\ P(X_0,Y_1;\theta)=P(X_0\mid Y_1;\theta)P(Y_1;\theta)=\alpha(1-p)\\ P(X_1,Y_0;\theta)=P(X_1\mid Y_0;\theta)P(Y_0;\theta)=(1-\alpha)q\\ P(X_0,Y_0;\theta)=P(X_0\mid Y_0;\theta)P(Y_0;\theta)=(1-\alpha)(1-q)\end{cases} \tag{5-57}$$

$$P(X;\theta)=\begin{cases}P(X_1;\theta)=\sum_{k=0}^{1}P(X_1\mid Y_k;\theta)P(Y_k;\theta)=\alpha p+(1-\alpha)q\\ P(X_0;\theta)=\sum_{k=0}^{1}P(X_0\mid Y_k;\theta)P(Y_k;\theta)=\alpha(1-p)+(1-\alpha)(1-q)\end{cases}\tag{5-58}$$

$$P(Y\mid X;\theta)=\begin{cases}P(Y_1\mid X_1;\theta)=\dfrac{P(X_1,Y_1;\theta)}{P(X_1;\theta)}=\dfrac{\alpha p}{\alpha p+(1-\alpha)q}\\ P(Y_0\mid X_1;\theta)=1-P(Y_1\mid X_1;\theta)\\ P(Y_1\mid X_0;\theta)=\dfrac{P(X_0,Y_1;\theta)}{P(X_0;\theta)}=\dfrac{\alpha(1-p)}{\alpha(1-p)+(1-\alpha)(1-q)}\\ P(Y_0\mid X_0;\theta)=1-P(Y_1\mid X_0;\theta)\end{cases}\tag{5-59}$$

将掷硬币实验所记录的0、1序列看作数据集$D=\{x_1,x_2,\cdots,x_m\}$，$x_j\in\{0,1\}$，$j=1,2,\cdots,m$，其中只包含第二枚硬币(B或C)的投掷结果X。现在希望根据数据集D建立三硬币的概率模型$P(X,Y;\theta)$，这是一个含隐变量的概率模型，其概率分布形式已知，但参数$\theta=(\alpha,p,q)$未知。

2. 模型参数估计

给定掷硬币实验的数据集$D=\{x_1,x_2,\cdots,x_m\}$，其对数似然函数为

$$\ln L(\theta)=\ln\Big(\prod_{j=1}^{m}P(x_j;\theta)\Big)=\sum_{j=1}^{m}\ln P(x_j;\theta)\tag{5-60}$$

最大化式(5-60)的对数似然函数$\ln L(\theta)$，需使用EM算法求解最优参数$\theta^*=(\alpha^*,p^*,q^*)$，从而建立起三硬币的概率模型。

3. 应用EM算法

首先选择初始参数$\theta^0=(\alpha^0,p^0,q^0)$，然后迭代执行如下的E步和M步，直至收敛(例如迭代后参数无变化)。

E步：根据上次迭代的参数θ^{i-1}，求数据集D上的Q函数(即期望)。

首先，将参数$\theta^{i-1}=(\alpha^{i-1},p^{i-1},q^{i-1})$代入式(5-59)，求得$P(Y|X;\theta^{i-1})$。为便于后续演算，这里对式(5-59)做进一步改写。记$P(Y_1\mid X_1;\theta)$为γ_1，$P(Y_1\mid X_0;\theta)$为γ_0，则

$$P(Y\mid X;\theta)=\begin{cases}P(Y_1\mid X_1;\theta)=\gamma_1=\dfrac{\alpha p}{\alpha p+(1-\alpha)q}\\ P(Y_0\mid X_1;\theta)=1-\gamma_1\\ P(Y_1\mid X_0;\theta)=\gamma_0=\dfrac{\alpha(1-p)}{\alpha(1-p)+(1-\alpha)(1-q)}\\ P(Y_0\mid X_0;\theta)=1-\gamma_0\end{cases}\tag{5-61}$$

其中，γ_1、γ_0分别是第二枚硬币(B或C)投掷结果X为正面或反面时，硬币A投掷结果为正面的概率。代入参数$\theta^{i-1}=(\alpha^{i-1},p^{i-1},q^{i-1})$，求得$\gamma_1^{i-1}$和$\gamma_0^{i-1}$。

$$\begin{cases}\gamma_1^{i-1}=\dfrac{\alpha^{i-1}p^{i-1}}{\alpha^{i-1}p^{i-1}+(1-\alpha^{i-1})q^{i-1}}\\ \gamma_0^{i-1}=\dfrac{\alpha^{i-1}(1-p^{i-1})}{\alpha^{i-1}(1-p^{i-1})+(1-\alpha^{i-1})(1-q^{i-1})}\end{cases}\tag{5-62}$$

再假设数据集 $D=\{x_1,x_2,\cdots,x_m\}$ 中正面样本点有 m_1 个,反面样本点有 m_0 个,$m_1+m_0=m$,则数据集 D 上的 Q 函数为

$$\begin{aligned}Q(\theta,\theta^{i-1})&=\sum_{j=1}^{m}\Big(\sum_{k=0}^{1}P(Y_k\mid x_j;\theta^{i-1})\ln P(x_j,Y_k;\theta)\Big)\\&=m_1Q_1(\theta,\theta^{i-1})+m_0Q_0(\theta,\theta^{i-1})\end{aligned}\tag{5-63}$$

其中

$$\begin{aligned}Q_1(\theta,\theta^{i-1})&=\sum_{k=0}^{1}P(Y_k\mid X_1;\theta^{i-1})\ln P(X_1,Y_k;\theta)\\&=P(Y_1\mid X_1;\theta^{i-1})\ln P(X_1,Y_1;\theta)+P(Y_0\mid X_1;\theta^{i-1})\ln P(X_1,Y_0;\theta)\\&=\gamma_1^{i-1}\ln(\alpha p)+(1-\gamma_1^{i-1})\ln((1-\alpha)q)\end{aligned}\tag{5-64}$$

$$\begin{aligned}Q_0(\theta,\theta^{i-1})&=\sum_{k=0}^{1}P(Y_k\mid X_0;\theta^{i-1})\ln P(X_0,Y_k;\theta)\\&=(Y_1\mid X_0;\theta^{i-1})\ln P(X_0,Y_1;\theta)+P(Y_0\mid X_0;\theta^{i-1})\ln P(X_0,Y_0;\theta)\\&=\gamma_0^{i-1}\ln(\alpha(1-p))+(1-\gamma_0^{i-1})\ln((1-\alpha)(1-q))\end{aligned}\tag{5-65}$$

E 步结束。

M 步:最大化 Q 函数(即最大化期望),然后将其最优参数作为迭代后的新参数 θ^i。

对式(5-63)的 Q 函数求参数 α 的偏导并令其等于 0,即

$$\frac{\partial Q(\theta,\theta^{i-1})}{\partial\alpha}=m_1\Big(\frac{\gamma_1^{i-1}}{\alpha}-\frac{(1-\gamma_1^{i-1})}{(1-\alpha)}\Big)+m_0\Big(\frac{\gamma_0^{i-1}}{\alpha}-\frac{(1-\gamma_0^{i-1})}{(1-\alpha)}\Big)=0\tag{5-66}$$

求得最优参数 α 为

$$\alpha=\frac{1}{m_1+m_0}(m_1\gamma_1^{i-1}+m_0\gamma_0^{i-1})=\frac{1}{m}\sum_{k=0}^{1}m_k\gamma_k^{i-1}\tag{5-67}$$

再对 Q 函数求参数 p 的偏导并令其等于 0,即

$$\frac{\partial Q(\theta,\theta^{i-1})}{\partial p}=m_1\frac{\gamma_1^{i-1}}{p}-m_0\frac{\gamma_0^{i-1}}{1-p}=0\tag{5-68}$$

求得最优参数 p 为

$$p=\frac{m_1\gamma_1^{i-1}}{m_1\gamma_1^{i-1}+m_0\gamma_0^{i-1}}=\frac{m_1\gamma_1^{i-1}}{\sum_{k=0}^{1}m_k\gamma_k^{i-1}}\tag{5-69}$$

最后对 Q 函数求参数 q 的偏导并令其等于 0,即

$$\frac{\partial Q(\theta,\theta^{i-1})}{\partial q}=m_1\frac{1-\gamma_1^{i-1}}{q}-m_0\frac{1-\gamma_0^{i-1}}{1-q}=0\tag{5-70}$$

求得最优参数 q 为

$$q=\frac{m_1(1-\gamma_1^{i-1})}{(m_1+m_0)-(m_1\gamma_1^{i-1}+m_0\gamma_0^{i-1})}=\frac{m_1(1-\gamma_1^{i-1})}{m-\sum_{k=0}^{1}m_k\gamma_k^{i-1}} \tag{5-71}$$

综合式(5-67)、式(5-69)和式(5-71),取新的迭代参数 θ^i 为(α,p,q),M 步结束。检查迭代条件,如果参数 θ^i 与 θ^{i-1} 无明显变化,或 i 达到最大迭代次数,则停止迭代;否则返回 E 步,继续下次迭代。迭代结束后,将最后一次迭代的参数 θ^i 作为最优参数 θ^*。

三硬币模型是基于 0-1(二项)分布的概率模型,其求解算法可推广至多项分布。

5.3 k 均值聚类

与之前基于概率的聚类方法不同,k **均值聚类**(k-means clustering)是一种基于距离的聚类方法,其中 k 表示类别的个数。假设有 k 个类,每个类有一个中心点。直观上看,样本点离哪个类的中心点距离近就应该划归哪个类。

k 均值聚类是一种无监督学习算法。给定未做标注的数据集 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$,$k$ 均值聚类需要将其划分成 k 个不相交的**簇**,可记作$\{C_1,C_2,\cdots,C_k\}$。首先为每个簇建立一个能够代表该簇的**原型**(prototype,即中心点),然后将各样本点划归距离最近原型所代表的簇。k 均值聚类以簇中样本的均值作为该簇的原型,每个簇构成一类,共 k 个。将 k 个类的均值记作均值向量$\boldsymbol{\mu}=(\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\cdots,\boldsymbol{\mu}_k)$,它们是聚类模型的未知参数,需基于数据集 D 并设计聚类算法来进行学习。

k 均值聚类算法主要包括数据预处理、距离度量、均值初始化、均值迭代和数据集聚类等五步,其中距离度量和均值迭代是算法的关键。

5.3.1 k 均值聚类算法

1. 数据预处理

数据集 D 通常包含多个特征项,不同特征项的取值范围可能存在较大差异,这就会造成特征项之间的不平等。为防止这种不平等现象被代入后续的距离度量,k 均值聚类算法需通过预处理对数据集 D 进行标准化,也即消除量纲,统一不同特征项的取值范围。常用的数据标准化方法有 Min-Max、z-score 等。

2. 距离度量

对于数值型特征,度量样本点之间的距离通常采用欧氏距离,即

$$\mathrm{dist}(\boldsymbol{x}_i,\boldsymbol{x}_j)=\sqrt{\sum_{l=1}^{d}(x_{il}-x_{jl})^2}\equiv\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2 \tag{5-72}$$

其中,d 表示特征的维数;样本点 $\boldsymbol{x}_i$、$\boldsymbol{x}_j$ 之间的欧氏距离通常也被记作 $\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2$。

对于非数值型特征,目前还没有很好的距离度量方法。例如,如何度量苹果、香蕉、桔子之间的距离,或如何度量马、牛、羊之间的距离?k 均值聚类方法主要适用于数值型特征的聚类问题。可以将非数值型特征转换为数值型,然后按数值型特征进行处理。

在确定了距离度量之后，k 均值聚类算法将聚类模型在数据集 D 上的损失函数定义为各样本点与其所属类均值之间距离的平方和，即

$$L(\boldsymbol{\mu})=\sum_{j=1}^{m}\mathrm{dist}^2(\boldsymbol{x}_j,\boldsymbol{\mu}_l)=\sum_{j=1}^{m}\|\boldsymbol{x}_j-\boldsymbol{\mu}_l\|_2^2 \tag{5-73}$$

其中，$\boldsymbol{\mu}_l$ 表示样本点 $\boldsymbol{x}_j$ 属于类 C_l，即 $\boldsymbol{x}_j\in C_l$，$\boldsymbol{\mu}_l$ 为该类的均值。可以看出损失函数 $L(\boldsymbol{\mu})$正比于各类方差的总和(即类内方差)，即

$$L(\boldsymbol{\mu})=\sum_{j=1}^{m}\|\boldsymbol{x}_j-\boldsymbol{\mu}_l\|_2^2\propto\sum_{l=1}^{k}\boldsymbol{\sigma}_l^2 \tag{5-74}$$

最小化式(5-73)的损失函数 $L(\boldsymbol{\mu})$，相当于最小化类内方差。损失函数 $L(\boldsymbol{\mu})$中各样本点 $\boldsymbol{x}_j$ 的所属类别 C_l 为隐变量，求解最优参数 $\boldsymbol{\mu}^*$ 需通过类似 EM 的迭代算法进行求解。

3. 均值初始化

记样本均值 $\boldsymbol{\mu}=(\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\cdots,\boldsymbol{\mu}_k)$，从标准化后的数据集 D 中随机选取 k 个样本点，分别作为 k 个类的初始均值，将它们记作$\boldsymbol{\mu}^0=(\boldsymbol{\mu}_1^0,\boldsymbol{\mu}_2^0,\cdots,\boldsymbol{\mu}_k^0)$。

4. 均值迭代

有了初始均值$\boldsymbol{\mu}^0$，下面从 $i=1$ 开始迭代执行如下的**标注样本**和**更新均值**两步，直至收敛(例如迭代后均值无变化)。

1) 标注样本

对数据集 D 中的样本数据进行标注。根据上次迭代的均值$\boldsymbol{\mu}^{i-1}=(\boldsymbol{\mu}_1^{i-1},\boldsymbol{\mu}_2^{i-1},\cdots,\boldsymbol{\mu}_k^{i-1})$，将各样本点标注为距离最近均值所代表的类，即

$$c_j^{i-1}=\underset{l=1,2,\cdots,k}{\operatorname{argmin}}\|\boldsymbol{x}_j-\boldsymbol{\mu}_l^{i-1}\|_2^2,\quad j=1,2,\cdots,m \tag{5-75}$$

其中，$\boldsymbol{x}_j$ 表示第 j 个样本点；c_j^{i-1} 表示该样本点本次迭代的标注结果，$c_j^{i-1}\in\{1,2,\cdots,k\}$。然后再根据标注结果，将标注相同的样本点划归一类，这样数据集 D 就被划分成 k 个类，记作 $C^{i-1}=\{C_1^{i-1},C_2^{i-1},\cdots,C_k^{i-1}\}$。

2) 更新均值

根据上一步的分类结果 $C^{i-1}=\{C_1^{i-1},C_2^{i-1},\cdots,C_k^{i-1}\}$，重新计算各类的均值，即更新均值。

$$\boldsymbol{\mu}_l^i=\frac{1}{|C_l^{i-1}|}\sum_{\boldsymbol{x}_j\in C_l^{i-1}}\boldsymbol{x}_j,\quad l=1,2,\cdots,k \tag{5-76}$$

其中，$|C_l^{i-1}|$表示类 C_l^{i-1} 中样本点的个数。更新均值后检查迭代条件，如果参数 $\boldsymbol{\mu}^i$ 与 $\boldsymbol{\mu}^{i-1}$ 无明显变化，或 i 达到最大迭代次数，则停止迭代，将参数$\boldsymbol{\mu}^i$ 作为聚类模型的最优参数 $\boldsymbol{\mu}^*$；否则返回上一步，继续迭代。

5. 数据集聚类

通过均值迭代求解出的最优参数$\boldsymbol{\mu}^*=(\boldsymbol{\mu}_1^*,\boldsymbol{\mu}_2^*,\cdots,\boldsymbol{\mu}_k^*)$，它们是最终聚类模型中各类的样本均值(即原型)。按照与式(5-75)相同的距离最近原则，对数据集 D 中样本数据做最终标注，将样本数据 $\boldsymbol{x}_j$ 标注为 y_j，即

$$y_j=\underset{l=1,2,\cdots,k}{\operatorname{argmin}}\|\boldsymbol{x}_j-\boldsymbol{\mu}_l^*\|_2^2,\quad j=1,2,\cdots,m$$

这样就完成了对数据集的聚类过程。今后任给新样本 $\boldsymbol{x}$，同样也可以按上述方法进行标注(即分类)。

5.3.2 关于 k 均值聚类的讨论

1. k 均值聚类算法的收敛性

k 均值聚类算法即最小化式(5-73)的损失函数 $L(\boldsymbol{\mu})$，其中各样本点 $\boldsymbol{x}_j$ 的所属类别 C_l 相当于隐变量 Y，其求解过程实际上是对常规含隐变量概率模型和 EM 算法的一种简化。

k 均值聚类算法中标注样本的操作(见式(5-75))相当于 EM 算法中的 E 步：根据上次迭代的参数 $\boldsymbol{\mu}^{i-1}$，求数据集 D 上的 Q 函数(即期望)。

$$Q(\boldsymbol{\mu},\boldsymbol{\mu}^{i-1})=\sum_{j=1}^{m}\left(\sum_{Y}P(Y\mid\boldsymbol{x}_j;\boldsymbol{\mu}^{i-1})\ln P(\boldsymbol{x}_j,Y;\boldsymbol{\mu})\right) \tag{5-77}$$

其中，$\boldsymbol{\mu}^{i-1}$ 为已知参数，$\boldsymbol{\mu}$ 为待求解参数，且

$$P(Y\mid\boldsymbol{x};\boldsymbol{\mu}^{i-1})=\begin{cases}P(Y_c\mid\boldsymbol{x};\boldsymbol{\mu}^{i-1})=1, & c=\underset{l=1,2,\cdots,k}{\operatorname{argmin}}\|\boldsymbol{x}_j-\boldsymbol{\mu}_l^{i-1}\|_2^2\\ P(Y_c\mid\boldsymbol{x};\boldsymbol{\mu}^{i-1})=0, & c\neq\underset{l=1,2,\cdots,k}{\operatorname{argmin}}\|\boldsymbol{x}_j-\boldsymbol{\mu}_l^{i-1}\|_2^2\end{cases} \tag{5-78}$$

$$\ln P(\boldsymbol{x},Y;\boldsymbol{\mu})=\begin{cases}\ln P(\boldsymbol{x},Y_c;\boldsymbol{\mu})=-\|\boldsymbol{x}-\boldsymbol{\mu}_c\|_2^2, & c=\underset{l=1,2,\cdots,k}{\operatorname{argmin}}\|\boldsymbol{x}_j-\boldsymbol{\mu}_l^{i-1}\|_2^2\\ \ln P(\boldsymbol{x},Y_c;\boldsymbol{\mu})=0, & c\neq\underset{l=1,2,\cdots,k}{\operatorname{argmin}}\|\boldsymbol{x}_j-\boldsymbol{\mu}_l^{i-1}\|_2^2\end{cases} \tag{5-79}$$

$$Q(\boldsymbol{\mu},\boldsymbol{\mu}^{i-1})=\sum_{j=1}^{m}(-\|\boldsymbol{x}_j-\boldsymbol{\mu}_c\|_2^2)=-L(\boldsymbol{\mu}),\quad \boldsymbol{\mu}_c\text{ 为 }\boldsymbol{x}_j\text{ 所属类的均值} \tag{5-80}$$

k 均值聚类算法中更新均值的操作(见式(5-76))相当于 EM 算法中的 M 步：最大化 Q 函数，实际上就是最小化损失函数 $L(\boldsymbol{\mu})$，即

$$\boldsymbol{\mu}^{i}=\underset{\boldsymbol{\mu}}{\operatorname{argmax}}\,Q(\boldsymbol{\mu},\boldsymbol{\mu}^{i-1})=\underset{\boldsymbol{\mu}}{\operatorname{argmin}}\,L(\boldsymbol{\mu}) \tag{5-81}$$

式(5-81)的最优解就是式(5-76)的最优解。

由 EM 算法可知，第 i 次迭代时将参数从 $\boldsymbol{\mu}^{i-1}$ 更新到 $\boldsymbol{\mu}^{i}$，可使损失函数 $L(\boldsymbol{\mu})$ 的函数值下降，即 $L(\boldsymbol{\mu}^{i})\leqslant L(\boldsymbol{\mu}^{i-1})$。如果损失函数 $L(\boldsymbol{\mu})$ 是凸函数(例如式(5-73))，则 k 均值聚类算法可以收敛至最小值。

2. 超参数 k 值的选择

k 均值聚类中的 k 是一个需要人工预设的超参数，表示类别的个数。可以通过可视化观察数据分布情况，给出一个相对合理的 k 值；也可以为 k 选取一组候选值，然后使用数据集逐个训练并计算各候选 k 值最终的损失(见式(5-73))，从中选取损失最小的候选值作为最优 k 值。

一个更加灵活的方法是仅将预设的 k 值看作预估值，聚类时再根据数据的实际分布进行调整。聚类过程中，当簇内样本点数量过少或两个簇距离过近时，对簇进行**合并**操作；当

簇内样本点数量过多且方差较大时,对簇进行**分裂**操作。这种根据数据实际分布动态调整 k 值的方法被称作 **ISODATA**(Iterative Self-Organizing Data Analysis Techniques Algorithm)聚类。

3. 初始均值的选择

k 均值聚类的 k 个初始均值是从数据集 D 中随机选取的。从直观上看,这 k 个初始均值应**尽量散开**,这样可以加快收敛速度。随机选取第一个初始均值 $\boldsymbol{\mu}_1^0$,然后在选择下一个均值 $\boldsymbol{\mu}_2^0$ 时尽量选择距 $\boldsymbol{\mu}_1^0$ 较远的样本点,或者说距 $\boldsymbol{\mu}_1^0$ 越远的样本点被选作 $\boldsymbol{\mu}_2^0$ 的概率应当越大;重复这个过程,在选择下一个均值 $\boldsymbol{\mu}_i^0$ 时,距前 $i-1$ 个 $\{\boldsymbol{\mu}_1^0,\boldsymbol{\mu}_2^0,\cdots,\boldsymbol{\mu}_{i-1}^0\}$ 越远的样本点被选作 $\boldsymbol{\mu}_i^0$ 的概率应当越大,直至选出全部 k 个均值。按照上述方法选取初始均值的 k 均值聚类方法被称为 ***k*-means++**聚类。

4. 聚类模型的评价指标

评价聚类模型的好坏,**轮廓系数**(silhouette coefficient)是一个常用的评价指标。对于聚类模型在数据集 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$ 的聚类结果,首先计算各样本点 $\boldsymbol{x}_i$ 距本簇中其他样本的平均距离 a_i(越小越好),以及距其他最近一个簇中样本间的平均距离 b_i(越大越好),然后定义样本点 $\boldsymbol{x}_i$ 的轮廓系数 s_i 为

$$s_i=\frac{b_i-a_i}{\max(a_i,b_i)},\quad -1\leqslant s_i\leqslant 1 \tag{5-82}$$

最终,聚类模型在整个数据集 D 上的轮廓系数 s 被定义为所有样本点轮廓系数的平均,即

$$s=\frac{1}{m}\sum_{i=1}^{m}s_i,\quad -1\leqslant s\leqslant 1 \tag{5-83}$$

轮廓系数 s 越接近于 1,则说明聚类的效果越好。

5.3.3 使用 scikit-learn 库中的 *k* 均值聚类模型

练习 k 均值聚类编程,可以使用 scikit-learn 库提供的葡萄酒数据集 wine recognition dataset 和 k 均值聚类模型 **KMeans**。

1. 葡萄酒数据集 wine recognition dataset

wine recognition dataset 是一个包含人工标注的葡萄酒数据集,其中包含 178 个样例(产自 3 个不同的种植园),每个样例包含 13 项特征数据(表示 13 种化学成分含量,均为数值型特征)和一项人工标注(标注来自哪个种植园,分别用 0、1、2 表示)。

图 5-1 给出一个下载葡萄酒数据集的示例代码,它将数据集及其说明文档分别保存到 data 目录下的 wine.csv 文件和 wine.txt 中。

仅仅依据产自哪个种植园来标注葡萄酒,这种分类方式是科学的吗?可以根据葡萄酒的化学成分进行聚类,对比一下聚类的标注结果与人工标注是否一致。如果两者比较接近,那就说明根据产地对葡萄酒进行分类是科学的。

2. 使用 scikit-learn 库中的 *k* 均值聚类模型

scikit-learn 库将 k 均值聚类模型封装成一个类(类名为 **KMeans**),并将其存放在

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        %matplotlib inline

        #下载葡萄酒数据集，保存到本地文件wine.csv中
        from sklearn.datasets import load_wine
        w3 = load_wine()
        print(w3.keys());  print(w3.data.shape)
        print('wine: ', w3.target[0]);  print(w3.data[0])

        df = pd.DataFrame( w3.data )
        df['target'] = w3.target
        df.to_csv("./data/wine.csv", index=None)
        #将数据集的说明文档保存到本地文件wine.txt中
        file = open("./data/wine.txt", 'w')
        file.write(w3.DESCR);  file.close()

    dict_keys(['data', 'target', 'target_names', 'DESCR', 'feature_names'])
    (178, 13)
    wine:  0
    [1.423e+01 1.710e+00 2.430e+00 1.560e+01 1.270e+02 2.800e+00 3.060e+00
     2.800e-01 2.290e+00 5.640e+00 1.040e+00 3.920e+00 1.065e+03]
```

图 5-1　下载葡萄酒数据集的示例代码

sklearn.cluster 模块中。KMeans 类实现了 k 均值聚类(含 k-means++)模型的聚类算法 **fit**(),另外还提供一个预测算法 **predict**(),可用于对新样本的分类。

首先从下载到本地 data 目录下的 wine.csv 文件中加载葡萄酒数据集,得到一个 DataFrame 类的二维表格 w3,显示其形状(shape,即表格的行数和列数)。取出数据集中的特征(0～12 列),对其进行标准化(例如使用 scikit-learn 库提供的 MinMaxScaler 类实现 Min-Max 标准化),生成一个仅包含特征的数据集 X;再取出人工标注(第 13 列,列名为 target),生成标注集 Y。图 5-2 给出了加载本地葡萄酒数据集 wine.csv 并进行标准化的示例代码。

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        %matplotlib inline

        w3 = pd.read_csv("./data/wine.csv")
        print("shape=", w3.shape)

        from sklearn.preprocessing import MinMaxScaler
        scaler = MinMaxScaler()
        scaler.fit(w3.iloc[:, :13])
        X = scaler.transform(w3.iloc[:, :13])
        print(w3.iloc[:2, :5]);  print(X[:2, :5])

        Y = w3["target"]

    shape= (178, 14)
          0     1     2     3      4
    0  14.23  1.71  2.43  15.6  127.0
    1  13.20  1.78  2.14  11.2  100.0
    [[0.84210526 0.1916996  0.57219251 0.25773196 0.61956522]
     [0.57105263 0.2055336  0.4171123  0.03092784 0.32608696]]
```

图 5-2　加载本地葡萄酒数据集并进行标准化的示例代码

图 5-3 给出了使用 KMeans 类对葡萄酒数据集进行聚类的示例代码。其中聚类所使用的数据集是图 5-2 生成的特征数据集 X,k 值取 3;调用 fit()方法得到聚类结果,其中包括三个类的样本均值 cluster_centers_,以及数据集 X 中各样例的聚类标注 labels_(0、1 或 2)。可以同时显示图 5-2 标注集 Y 中的人工标注,将其与 KMeans 类标注进行比对,如图 5-4 所示。

```
In  [2]: from sklearn.cluster import KMeans
         km = KMeans(n_clusters=3, init='random', random_state=2020)
         #km = KMeans(n_clusters=3, init='k-means++', random_state=2020)

         km.fit(X)
         print("cluster_centers_:");  print(km.cluster_centers_)
         print("labels_:");  print(km.labels_)
         print("label_Y:");  print(Y.values)

    cluster_centers_:
    [[0.31137521 0.23689915 0.47291703 0.49991686 0.2477209  0.45305895
      0.38240098 0.4117468  0.39742546 0.14773478 0.47351167 0.58897554
      0.15640099]
     [0.544689   0.47844053 0.56013612 0.53833177 0.31146245 0.24476489
      0.10713464 0.61852487 0.22827646 0.4826404  0.19254989 0.16090576
      0.24739982]
     [0.70565142 0.24842869 0.58490401 0.3444313  0.41072701 0.64211419
      0.55467939 0.30034024 0.47727155 0.35534046 0.47780888 0.69038612
      0.59389397]]
```

图 5-3　使用 KMeans 类对葡萄酒数据集进行聚类的示例代码

```
labels_:
[2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 1 1 0 0 0 0 0 0 1 0 1 0 0 2
 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
label_Y:
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2]
```

图 5-4　比对 KMeans 类标注 labels_与人工标注 labels_Y

图 5-4 中 KMeans 类标注 labels_的 2、0、1(仅仅是一种类别编号)分别对应人工标注 labels_Y 的 0、1、2。可以看出,两者的分类结果(即哪些样例是属于同一类的)很接近,这说明不同产地葡萄酒的化学成分确实不一样,以产地来衡量葡萄酒质量是有一定道理的。

3. 使用 scikit-learn 库中的轮廓系数函数

scikit-learn 库为评价聚类模型提供了一个计算轮廓系数的函数 **silhouette_score**(),它被存放在 sklearn.metrics 模块中。图 5-5 给出了一段示例代码,用于计算并显示 k 均值聚类模型在葡萄酒数据集上的轮廓系数。

```
In  [3]: from sklearn.metrics import silhouette_score
         sc = silhouette_score(X, km.labels_, metric='euclidean')
         print(sc)

    0.3008938518500134
```

图 5-5　k 均值聚类模型在葡萄酒数据集上的轮廓系数

5.4　密度聚类 DBSCAN

与 k 均值聚类基于样本距离的方法不同,**DBSCAN**(Density-Based Spatial Clustering of Applications with Noise)是一种基于**样本密度**(density)的聚类方法。从直观上看,同类的样本应集中分布在同一区域内,区域内部的样本密度高,边缘的样本密度低。聚类时可以从

某个内部点出发逐步向周边扩展，直至遇到边缘点时停止；重复这个扩展过程，最终扩展至类的整个分布区域。

如果有多个类，不同类的样本应分布在不同区域，区域之间应该有间隙，或者只有少量被称作**噪声**(noise)或**离群点**(outlier)的样本。图 5-6 给出一个样本分布示意图，其中有三个类，分别用●、▲、╋表示，另外还用★表示相对孤立的噪声样本。

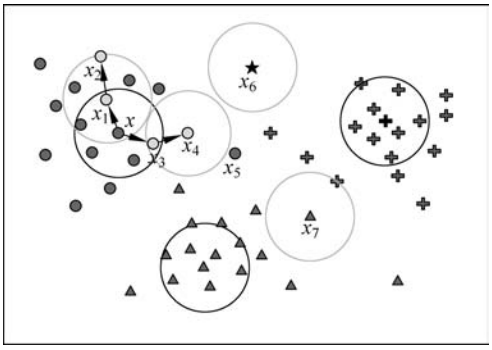

图 5-6　三个类的样本分布示意图

DBSCAN 是一种典型的基于密度的聚类算法，它能在具有噪声的数据集中发现任意形状的簇，每个簇构成一类。下面结合图 5-6 来具体讲解 DBSCAN 聚类的主要术语及算法实现。

5.4.1　DBSCAN 聚类术语

给定未做标注的数据集 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$，下面定义一下 DBSCAN 聚类用到的主要术语。

1. ε-邻域

对于样本点 $\boldsymbol{x}_j\in D$，数据集 D 中与 $\boldsymbol{x}_j$ 距离不超过 ε 的样本点集合被称作 $\boldsymbol{x}_j$ 的 ε-**邻域**(neighborhood)，记作 $N_\varepsilon(\boldsymbol{x}_j)$，

$$N_\varepsilon(\boldsymbol{x}_j)=\{\boldsymbol{x}_i \mid \boldsymbol{x}_i\in D,\text{distance}(\boldsymbol{x}_i,\boldsymbol{x}_j)\leqslant\varepsilon\} \tag{5-84}$$

这个 ε-邻域中样本点的个数 $|N_\varepsilon(\boldsymbol{x}_j)|$ 被称作 $\boldsymbol{x}_j$ 的样本密度。例如，图 5-6 中 $\boldsymbol{x}$ 的邻域内有 5 个样本点(不含 $\boldsymbol{x}$)，则 $\boldsymbol{x}$ 的样本密度为 5。式(5-84)中的 ε 是一个需要人工预设的超参数；距离 distance($\boldsymbol{x}_i,\boldsymbol{x}_j$)可以使用不同的度量形式，常用的是欧氏距离。

2. 核心对象

若 $\boldsymbol{x}_j$ 的样本密度不小于 **MinPts** 个样本点，即 $|N_\varepsilon(\boldsymbol{x}_j)|\geqslant$MinPts，则 $\boldsymbol{x}_j$ 被称作**核心对象**(core object)。其中，MinPts 是核心对象的阈值，它是一个需要人工预设的超参数。如果 $\boldsymbol{x}_j$ 是一个核心对象，则说明 $\boldsymbol{x}_j$ 一定属于某个类且位于该类样本分布区域的内部。例如，假设 MinPts=3(下同)，则图 5-6 中的 $\boldsymbol{x}$、$\boldsymbol{x}_1$、$\boldsymbol{x}_3$ 等就是核心对象，但 $\boldsymbol{x}_4$、$\boldsymbol{x}_6$、$\boldsymbol{x}_7$ 等就不是核心对象。

3. 密度直达

若 $\boldsymbol{x}_i$ 位于核心对象 $\boldsymbol{x}_j$ 的邻域内，则称 $\boldsymbol{x}_i$ 可由 $\boldsymbol{x}_j$ **密度直达**(directly density-reachable)。

例如,图 5-6 中的 $\boldsymbol{x}_1$、$\boldsymbol{x}_3$ 可由核心对象 $\boldsymbol{x}$ 密度直达,另外 $\boldsymbol{x}_2$ 也可由核心对象 $\boldsymbol{x}_1$ 密度直达。若 $\boldsymbol{x}_i$、$\boldsymbol{x}_j$ 均为核心对象,则属于**双向**密度直达;若其中一个为核心对象,另一个为非核心对象,则属于**单向**密度直达;若两者均为非核心对象,则不可能密度直达。例如,$\boldsymbol{x}_3$ 与 $\boldsymbol{x}$ 之间属于双向密度直达(两者均为核心对象);而 $\boldsymbol{x}_3$ 与 $\boldsymbol{x}_4$ 之间属于单向密度直达($\boldsymbol{x}_4$ 为非核心对象);$\boldsymbol{x}_4$ 与 $\boldsymbol{x}_6$ 之间则不可能密度直达(两者均为非核心对象)。

4. 密度相连

若存在样本点序列 $\boldsymbol{x},\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n,\boldsymbol{y}$,其中 $\boldsymbol{x}_1$ 可由 $\boldsymbol{x}$ 密度直达,$\boldsymbol{y}$ 可由 $\boldsymbol{x}_n$ 密度直达,$\boldsymbol{x}_i$ 均为核心对象且与 $\boldsymbol{x}_{i+1}$ 之间双向密度直达,则称 $\boldsymbol{x}$ 与 $\boldsymbol{y}$ 之间经由 $\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n$ **密度相连**(density-connected)。例如,$\boldsymbol{x}_1$ 与 $\boldsymbol{x}_3$ 之间经由 $\boldsymbol{x}$ 密度相连,$\boldsymbol{x}_2$ 与 $\boldsymbol{x}_4$ 之间经由 $\boldsymbol{x}_1,\boldsymbol{x},\boldsymbol{x}_3$ 密度相连。密度直达也属于密度相连,是密度相连的特例。

5. 簇

给定超参数(ε,MinPts),数据集 D 中每个密度相连的最大子集即构成一个簇。换句话说,假设数据集 D 中的某个核心对象 $\boldsymbol{x}_j$ 属于某个簇 C,如果 $\boldsymbol{x}_i \in D$ 且 $\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$ 密度相连,则 $\boldsymbol{x}_i \in C$。可以选择一个核心对象,然后将所有与之密度相连的样本点聚集在一起形成簇,并将它们标注为一类,这就是 **DBSCAN** 聚类。

5.4.2 DBSCAN 聚类算法

给定需要聚类的数据集 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$,并设定好超参数(ε,MinPts),DBSCAN 聚类算法的步骤如下。

1. 找出核心对象

遍历数据集 D,找出其中所有的核心对象,并记作核心对象集合 Ω,则

$$\Omega=\{\boldsymbol{x}_j \mid \boldsymbol{x}_j \in D, \mid N_\varepsilon(\boldsymbol{x}_j)\mid \geqslant \text{MinPts}\} \tag{5-85}$$

2. 选取核心对象进行扩展

(1) 初始化当前类别标注 $k=0$。

(2) 遍历集合 Ω,从中随机选取一个核心对象作为簇的种子(记作 s),然后进行扩展:将 s 标记为 visited(已访问或已处理)并分别创建一个新簇 $C_k=\{s\}$ 和一个扩展队列 $Q=N_\varepsilon(s)$,转下一步。

(3) 遍历扩展队列 Q,依次取出其中的样本点(记作 q)并进行扩展:将 q 标记为 visited 并将其追加到簇 C_k 中,然后检查 q 是否为核心对象,若是核心对象则将 $N_\varepsilon(q)$中所有未被访问过(无 visited 标记)的样本点追加到队列 Q 的队尾。

(4) 重复第(3)步,直到扩展队列 Q 为空。当扩展队列 Q 为空时,簇 C_k 完成聚类,将其中所有核心对象从集合 Ω 中删除(因为已被处理),然后将 k 加 1,转第(5)步。

(5) 检查集合 Ω,如果所有核心对象都已被处理,则集合 Ω 为空,聚类结束;否则返回第(2)步,继续选取核心对象并扩展下一个簇。

3. 标注数据集

聚类结束后,根据所扩展出的 k 个簇 $C_0,C_1,\cdots,C_{k-1}$ 对数据集 D 中的样本进行标注。

将数据集 D 中属于簇 C_i 的样本点全部标注为 i，剩余样本点（不属于任何簇）标注为噪声（例如标注为－1）。整个聚类算法结束。

与 k 均值聚类相比，DBSCAN 聚类不需要指定 k 值，但需指定两个阈值（ε，MinPts），这两个超参数对聚类结果的影响比较大。另外，DBSCAN 聚类时没有建立模型，只能用于当前数据集 D 的聚类，不能再对其他新样本进行分类。DBSCAN 聚类最大的优点是可以对非凸样本集（见图 5-7）进行聚类，而 k 均值聚类只适用于凸样本集（例如服从高斯分布的样本集）。DBSCAN 聚类对凸样本集和非凸样本集都适用。

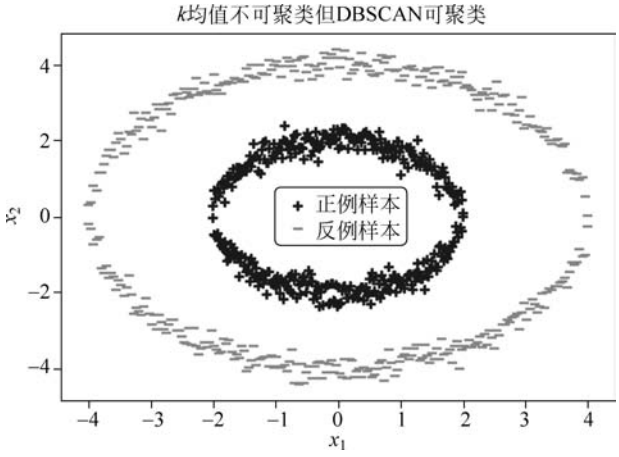

图 5-7　非凸样本集示例

5.4.3　使用 scikit-learn 库中的 DBSCAN 聚类算法

scikit-learn 库将 DBSCAN 聚类算法封装成一个类（类名也为 **DBSCAN**），并将其存放在 sklearn.cluster 模块中。DBSCAN 类的最主要方法就是聚类算法 **fit**()。由于 DBSCAN 在聚类时没有建立模型，因此 DBSCAN 类也没有对新样本进行分类的预测算法 predict()。

图 5-8 给出了使用 DBSCAN 类对葡萄酒数据集进行聚类的示例代码。其中聚类所使用的数据集是图 5-2 标准化后的特征数据集 X；调用 fit()方法得到聚类结果，其中包括数据集 X 中的核心对象 core_sample_indices_，以及各样例的聚类标注 labels_（0、1 或－1）。

```
In [4]: from sklearn.cluster import DBSCAN
        db = DBSCAN(eps=0.5, min_samples=8, metric='euclidean')

        db.fit(X)
        print("core_sample_indices_:"); print(db.core_sample_indices_)
        print("labels_:"); print(db.labels_)
        print("label_Y:"); print(Y.values)

        sc = silhouette_score(X, db.labels_, metric='euclidean')
        print(sc)

    core_sample_indices_:
    [  0   1   2   5   6   7   8   9  10  11  12  15  16  17  18  19  20  22
      23  24  26  27  28  29  30  31  32  34  35  36  37  38  40  42  44  46
      47  48  49  51  52  53  54  55  56  57  58  67  80  81  82  85  86  88
      89  91  93  97 100 101 102 103 104 106 107 108 111 113 114 116 117 119
     125 126 128 131 135 138 140 145 147 148 149 155 156 161 162 163 164 165
     166 167 170 171 172 173 174 175 176]
```

图 5-8　使用 DBSCAN 类对葡萄酒数据集进行聚类的示例代码

可以同时显示图 5-2 标注集 Y 中的人工标注，将其与 DBSCAN 聚类标注进行比对，也可以显示 DBSCAN 聚类在葡萄酒数据集上的轮廓系数，如图 5-9 所示。

```
labels_:
[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0 -1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  0  0 -1  0  0  0  0  0  0  0  0 -1 -1  1  0  0  0  0  0  0  0 -1 -1 -1 -1
  0 -1 -1  0  0  0 -1 -1  0  0  0  1 -1  0  0  0  0  0  0  0  0  0  0 -1
 -1  0 -1 -1  0  0  0  0  0 -1  0  0  0 -1 -1  0  0  0  0 -1  0  0  1  0
  0 -1 -1 -1 -1  0  0 -1  0  0  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1 -1  1  1  1  1  1 -1 -1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1]
label_Y:
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2]
0.2135398753843134
```

图 5-9　比对 DBSCAN 聚类标注 labels_与人工标注 labels_Y

5.5　向量量化

给定未做标注的数据集 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$，$k$ 均值聚类希望将其划分成 k 个不相交的簇，并以簇中样本的均值作为簇的原型。**向量量化**(Vector Quantization，VQ)与 k 均值聚类有点类似，所不同的是向量量化在聚类后将 k 个簇的均值向量$(\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\cdots,\boldsymbol{\mu}_k)$作为今后对向量进行编**码的码本**(codebook)，记作 $C=(\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\cdots,\boldsymbol{\mu}_k)$，其中每个均值$\boldsymbol{\mu}_j$ 被称作一个**码字向量**(code vector，简称码字)。编码时，任给向量 $\boldsymbol{x}$，向量量化将码本中距离最近的均值$\boldsymbol{\mu}_j$ 作为向量 $\boldsymbol{x}$ 的编码；或将距离最近均值$\boldsymbol{\mu}_j$ 的编号 j 作为向量 $\boldsymbol{x}$ 的编码，这样可以减少向量存储或传输时的数据量。向量量化被广泛应用于向量的离散化编码(称作**量化**)，或向量数据的压缩。

5.5.1　向量量化问题

向量量化分两个环节：一是**训练码本**，即给定样本数据集，通过聚类算法建立码本；二是**编码**，即任给向量，将码本中距离最近的码字(或其编号)作为向量的编码。

1. 向量量化的术语与符号

假设需要量化的向量为 d 维向量，下面给出量化过程中常用的术语和符号。

- **训练集**：用于训练码本的样本数据集，记作 $T=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$，其中每个样本数据 $\boldsymbol{x}_i$ 均为一 d 维向量。
- **训练码本**：将训练集 T 聚类成 k 个不相交的簇，将簇的集合记作 $S=\{S_1,S_2,\cdots,S_k\}$，其中 S_j 表示第 j 个簇；簇 S_j 以其样本均值作为簇的码字，将码字记作 $\boldsymbol{c}_j$(也是 d 维向量)，例如若 S_j 包含 m_j 个样本数据，则

$$\boldsymbol{c}_j=\frac{1}{m_j}\sum_{\boldsymbol{x}\in S_j}\boldsymbol{x},\quad j=1,2,\cdots,k$$

所有码字的集合被称作**码本**，记作 $C=\{\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_k\}$。对训练集进行聚类并生成码本

的算法被称作**向量量化算法**，简称 **VQ 算法**。

- **编码**：任给向量 $\boldsymbol{x}$，将码本中距离（例如欧氏距离）最近的码字 $\boldsymbol{c}_j$（或其编号 j）作为向量 $\boldsymbol{x}$ 的**编码**，记作 $q(\boldsymbol{x})$。上述编码准则被称作**最近邻准则**，其数学形式可表示为

$$q(\boldsymbol{x})=\underset{\boldsymbol{c}_j\in C}{\operatorname{argmin}}\|\boldsymbol{x}-\boldsymbol{c}_j\|_2$$

- **失真度**：聚类通常采用欧氏距离来度量向量之间的**距离**。在向量量化中，向量与其编码之间不完全相等，例如向量 $\boldsymbol{x}$ 与其编码 $q(\boldsymbol{x})$ 之间会存在误差，$q(\boldsymbol{x})\approx\boldsymbol{x}$。换句话说，对向量编码会造成**失真**（distortion）。可以用欧氏距离（或其平方）来度量编码的失真程度，即**失真度**，记作 $d(\boldsymbol{x},q(\boldsymbol{x}))\equiv\|\boldsymbol{x}-q(\boldsymbol{x})\|_2$。
- **平均失真度**：如果使用码本 C 对训练集 $T=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$ 进行编码，可以使用**平均失真度**来度量码本 C 在训练集 T 上的失真程度，记作 D_C。

$$D_C=\frac{1}{m}\sum_{i=1}^{m}d(\boldsymbol{x}_i,q(\boldsymbol{x}_i))=\frac{1}{m}\sum_{i=1}^{m}\|\boldsymbol{x}_i-q(\boldsymbol{x}_i)\|_2$$

- **量化准则**：通过训练集 T 训练码本，所训练出的码本 C^* 应遵循**最小失真准则**。其含义是，使用码本 C^* 对训练集 T 进行编码，其平均失真度应当最小。即

$$C^*=\underset{C}{\operatorname{argmin}}\,D_C \tag{5-86}$$

2. 进一步认识向量量化问题

向量量化虽然与 k 均值聚类有些相似，但向量量化问题远比聚类问题深刻。向量量化的原始问题是：给定向量空间 Ω（这里假设为连续型），其向量取值服从概率分布 $p(\boldsymbol{x})$，向量量化希望找到一个最优码本 C_Ω^*，使得向量 $\boldsymbol{x}\in\Omega$ 在编码后失真度的期望最小，即

$$C_\Omega^*=\underset{C}{\operatorname{argmin}}\,E[d(\boldsymbol{x},q(\boldsymbol{x}))]=\underset{C}{\operatorname{argmin}}\int_\Omega d(\boldsymbol{x},q(\boldsymbol{x}))p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{5-87}$$

为训练码本，向量量化需要先依概率分布 $p(\boldsymbol{x})$ 进行抽样，得到训练集 $T=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$，然后通过 VQ 算法求得训练集 T 上的最优码本 C^*（见式(5-86)），并将其作为整个向量空间 Ω 上最优码本 C_Ω^* 的估计。

相比较而言，k 均值聚类算法并不关心训练集 T 从哪里来，聚类结果会被推广哪里去，它只关心训练集 T 本身的聚类效果。另外，k 均值聚类算法关注的是分类效果，其算法目标是让类内方差最小，类间方差最大；而 VQ 算法关注的是编码效果，其算法目标只追求类内方差最小，并不需要类间方差最大。由于上述区别，VQ 算法与 k 均值聚类算法在初始均值的选择上存在较大不同。

5.5.2 LBG-VQ 算法

LBG-VQ 算法是向量量化中比较常用的一种码本训练算法（或称码本学习算法），它是 1980 年由 Linde、Buzo 和 Gray 三人联合提出的。LBG-VQ 算法与 k 均值聚类算法有点类似，它也是一种迭代算法。所不同的是，k 均值聚类算法是一次性选择 k 个初始均值，然后在此基础上进行迭代；而 LBG-VQ 算法则是从一个初始均值（即初始码字）开始，然后在此基础上先做码字**分裂**（split），再做聚类，重复这个**“分裂-聚类”**过程，直到码字达到指定数量（通常为 2^N 个）。

给定训练集 $T=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$，假设要训练一个包含 2^N 个码字的码本，则 LBG-VQ 算法就是一个 N 级“分裂－聚类”的迭代过程，其具体算法步骤如下。

(1) **初始化**：设置**码本大小** N(包含 2^N 个码字)，例如 $N=3(2^N=8)$；设置**失真阈值** $\varepsilon>0$，例如 $\varepsilon=0.001$；将**初始码字** $\boldsymbol{c}_1^0$ 设为训练集 T 的样本均值，并计算出初始**平均失真度** D_C^0。

$$\boldsymbol{c}_1^0=\frac{1}{m}\sum_{i=1}^{m}\boldsymbol{x}_i,\quad D_C^0=\frac{1}{m}\sum_{i=1}^{m}\|\boldsymbol{x}_i-\boldsymbol{c}_1^0\|_2$$

初始码本只包含一个码字，即 $C^0=\{\boldsymbol{c}_1^0\}$，设置**码字计数器** $k=1$；设置**迭代计数器** $t=1$，然后进入下面的“分裂-聚类”迭代过程。

(2) **分裂**(第 t 轮)：按如下方式将 $C^{t-1}=\{\boldsymbol{c}_1^{t-1},\boldsymbol{c}_2^{t-1},\cdots,\boldsymbol{c}_k^{t-1}\}$ 中的每个码字一分为二。

$$\boldsymbol{c}_j^t=(1+\varepsilon)\boldsymbol{c}_j^{t-1},\quad \boldsymbol{c}_{k+j}^t=(1-\varepsilon)\boldsymbol{c}_j^{t-1},\quad j=1,2,\cdots,k$$

并以新分裂出的 $2k$ 个码字组成新码本，将 k 乘以 2，即 $k=2k$，将新码本记作 C^t。

$$C^t=\{\boldsymbol{c}_1^t,\boldsymbol{c}_2^t,\cdots,\boldsymbol{c}_k^t\}$$

(3) **聚类**(第 t 轮)：使用新码本 C^t 对训练集 $T=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$ 进行编码和聚类，将码本 C^t 中距离最近(也即失真度最小)的码字作为样本 $\boldsymbol{x}_i$ 的编码，记作 $q(\boldsymbol{x}_i)$，即

$$q(\boldsymbol{x}_i)=\underset{\boldsymbol{c}_j^t\in C^t}{\operatorname{argmin}}\|\boldsymbol{x}_i-\boldsymbol{c}_j^t\|_2,\quad i=1,2,\cdots,m$$

并将编码相同的样本聚集成一簇，例如将编码为 $\boldsymbol{c}_j^t$ 的样本都聚集到簇 S_j 中，即

$$S_j=\{\boldsymbol{x}_i\mid \boldsymbol{x}_i\in T,q(\boldsymbol{x}_i)=\boldsymbol{c}_j^t\},\quad j=1,2,\cdots,k$$

然后计算码本 C^t 在训练集 T 上的平均失真度 D_C^t。

$$D_C^t=\frac{1}{m}\sum_{i=1}^{m}d(\boldsymbol{x}_i,q(\boldsymbol{x}_i))=\frac{1}{m}\sum_{i=1}^{m}\|\boldsymbol{x}_i-q(\boldsymbol{x}_i)\|_2$$

检查聚类迭代条件：如果第 t 轮迭代的平均失真度 D_C^t 较上一轮改进不大，即

$$\frac{D_C^{t-1}-D_C^t}{D_C^t}\leqslant\varepsilon$$

则停止聚类迭代，转第(4)步；否则重新计算各簇的样本均值并将其作为新的码本 C^{t+1}，例如若 S_j 包含 m_j 个样本数据，则

$$\boldsymbol{c}_j^{t+1}=\frac{1}{m_j}\sum_{\boldsymbol{x}\in S_j}\boldsymbol{x},\quad j=1,2,\cdots,k$$

将迭代计数加 1，即 $t=t+1$，返回第(3)步开头继续下一轮聚类迭代。

(4) **算法结束条件**：如果码本已达到指定大小，即码字数量 $k=2^N$，则算法结束，将 C^t 作为最终的码本 C^*；否则将迭代计数加 1，即 $t=t+1$，返回第(2)步继续下一轮“分裂-聚类”迭代，即先分裂，再聚类。

5.6 本章习题

一、单选题

1. 下列关于分类与聚类的描述中，错误的是(　　)。

A. 分类是根据有标注数据集来训练模型，属于有监督学习

B. 聚类是根据无标注数据集来训练模型，属于无监督学习

C. 聚类根据数据自身的分布特性或结构自动聚集成簇，形成类别概念

D. 给定无标注数据集 D，聚类算法需将数据集 D 划分成若干个可重叠的簇

2. 下列概率分布的等价表示中，错误的是(　　)。

A. $P(X=\boldsymbol{x})\equiv P(\boldsymbol{x})$　　B. $P(Y|X)\equiv P(Y|\boldsymbol{x})$

C. $P(Y=y|X)\equiv P(y|X)$　　D. $\sum\limits_{y\in\Omega_Y}P(X,y;\boldsymbol{\theta})\equiv\sum\limits_{Y}P(X,Y;\boldsymbol{\theta})$

3. 在聚类问题中，给定未做标注的数据集 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$，下列说法中错误的是(　　)。

A. 数据集 D 只包含样本的分类特征 X，未包含样本对应的类别标注 Y

B. 分类特征 X 是可观测的变量，样本类别 Y 是不可观测或未被观测的隐变量

C. 数据集 D 未包含样本类别 Y 的原因是实际应用不需要它

D. 聚类问题的关键是如何根据数据集 D 来估计含隐变量概率模型的参数

4. 关于聚类问题和混合概率模型参数估计问题，下列说法中错误的是(　　)。

A. 它们的模型在本质上都属于含隐变量的概率模型

B. 它们的样本数据集 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$ 都不包含类别标注

C. 估计它们的模型参数通常都使用 EM 算法进行求解

D. 它们最终要求解的都是分类特征 X 的概率分布 $P(X)$

5. 下列关于 EM 算法的描述中，错误的是(　　)。

A. EM 算法主要用于求解含隐变量的最优化问题

B. EM 算法是一种迭代算法

C. EM 算法的关键步骤是第 i 次迭代时如何将参数从 $\boldsymbol{\theta}^{i-1}$ 更新到 $\boldsymbol{\theta}^{i}$

D. EM 算法每次迭代应能让对数似然函数逐步下降，即 $\ln l(\boldsymbol{\theta}^{i})\leqslant\ln l(\boldsymbol{\theta}^{i-1})$

6. (　　)是 EM 算法中的 Q 函数(即期望)。

A. $\sum\limits_{Y}P(Y\mid X;\boldsymbol{\theta}^{i-1})\ln P(X,Y;\boldsymbol{\theta})$

B. $P(Y|X;\boldsymbol{\theta}^{i-1})\ln P(X,Y;\boldsymbol{\theta})$

C. $\sum\limits_{Y}P(Y\mid X;\boldsymbol{\theta}^{i-1})\ln\left(\dfrac{P(X,Y;\boldsymbol{\theta})}{P(Y\mid X;\boldsymbol{\theta}^{i-1})}\right)$

D. $P(Y|X;\boldsymbol{\theta}^{i-1})\ln\left(\dfrac{P(X,Y;\boldsymbol{\theta})}{P(Y|X;\boldsymbol{\theta}^{i-1})}\right)$

7. 下列关于 EM 算法步骤的描述中，错误的是(　　)。

A. EM 算法首先选择初始参数 $\boldsymbol{\theta}^{0}$

B. EM 算法的 E 步是根据上次迭代的参数 $\boldsymbol{\theta}^{i-1}$ 求 Q 函数(即期望)

C. EM 算法的 M 步是最小化 Q 函数，将最优参数作为迭代后的新参数 $\boldsymbol{\theta}^{i}$

D. EM 算法需重复执行 E 步和 M 步，直至收敛

8. 下列关于混合概率模型的描述中，错误的是(　　)。

A. 高斯混合模型由多个正态分布混合而成的

B. 高斯混合模型是连续型混合概率模型的代表

C. 三硬币模型是离散型混合概率模型的代表

D. 三硬币模型的硬币投掷过程是可观测的

9. 下列关于 k 均值聚类的描述中,错误的是(　　)。

A. k 均值聚类是一种基于概率的聚类方法

B. k 均值聚类是一种基于距离的聚类方法

C. k 均值聚类中的 k 指的是类别个数

D. 均值聚类是一种无监督学习算法

10. 下列关于 k 均值聚类的描述中,错误的是(　　)。

A. k 均值聚类中的 k 是需要人工预设的超参数

B. k 均值聚类可通过可视化分析给出一个相对合理的 k 值

C. 根据数据实际分布动态调整 k 值的方法被称为 ISODATA

D. 让 k 个初始均值尽量靠近的 k 均值聚类方法被称为 k-means++

11. 下列关于 DBSCAN 聚类的描述中,错误的是(　　)。

A. DBSCAN 聚类是一种基于概率的聚类方法

B. DBSCAN 聚类是一种基于样本密度的聚类方法

C. DBSCAN 聚类认为同类的样本应集中分布在某个区域内

D. DBSCAN 聚类认为不同类的样本应分布在不同区域,区域之间应该有间隙

12. 在 DBSCAN 聚类中,若 $\boldsymbol{x}_i$ 位于核心对象 $\boldsymbol{x}_j$ 的邻域内,则称 $\boldsymbol{x}_i$ 可由 $\boldsymbol{x}_j$(　　)。

A. 密度直达　　B. 密度相连

C. 密度连通　　D. 连通

13. 与 k 均值聚类相比,DBSCAN 聚类(　　)。

A. 不需要指定 k 值

B. 需要指定两个阈值(ε,MinPts)

C. 除了能对当前数据集 D 聚类之外,还能对其他新样本进行分类

D. 可以对非凸样本集进行聚类

14. 下列关于向量量化的描述中,错误的是(　　)。

A. 向量量化在聚类后将 k 个簇的均值向量($\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\cdots,\boldsymbol{\mu}_k$)作为码本

B. 均值向量($\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\cdots,\boldsymbol{\mu}_k$)中的每个均值$\boldsymbol{\mu}_j$ 被称作一个码字

C. 任给向量 $\boldsymbol{x}$,向量量化将码本中距离最近的均值(或其编号)作为其编码

D. 向量量化被广泛应用于向量数据的无失真压缩

15. 下列关于 k 均值聚类与向量量化的描述中,错误的是(　　)。

A. k 均值聚类只关心训练集本身的聚类效果

B. k 均值聚类算法的目标是让类内方差最小,类间方差最大

C. 向量量化算法关注的是编码效果

D. 向量量化算法的目标是让类内方差最大,类间方差最小

二、讨论题

1. 尝试用概率模型对聚类问题进行形式化描述。

2. 尝试对混合概率模型的参数估计问题进行形式化描述。

3. 尝试推导 EM 算法中对数似然函数 $\ln l(\boldsymbol{\theta})=\ln P(\boldsymbol{x};\boldsymbol{\theta})$的下界函数 $B(\boldsymbol{\theta},\boldsymbol{\theta}^{i-1})$。

4. 尝试用 EM 算法求解三硬币模型的参数估计问题。

5. 简述 k 均值聚类的算法思想。

6. 简述 DBSCAN 聚类的算法思想。

三、编程实践题

使用 scikit-learn 库提供的鸢尾花数据集(iris plants dataset)设计一个 k 均值聚类模型。具体的实验步骤如下。

(1) 使用函数 sklearn.datasets.load_iris()加载鸢尾花数据集。

(2) 查看数据集说明(https://scikit-learn.org/stable/datasets/toy_dataset.html#iris-dataset),并对数据集进行必要的预处理。

(3) 使用 sklearn.cluster.KMeans 类建立 k 均值聚类模型,并对鸢尾花数据集中的特征数据进行聚类处理。

(4) 对比聚类结果与人工标注,观察二者是否一致。