第3章

分类问题

本章学习要点

- 了解分类问题的基本概念与常用术语。
- 了解并掌握基于后验概率的贝叶斯分类器设计原理与常用方法。
- 了解并掌握基于特征直接建立判别函数的非贝叶斯分类器设计原理与常用方法。
- 了解多分类问题，以及将其拆分成二分类问题的策略。
- 了解常用的分类模型评价指标。
- 了解特征降维的基本原理及常用方法。

为了认识客观世界，人们按相似程度将客观事物划分成**类别**(class)。从不同角度观察客观事物会得到不同的属性，这种属性被称作客观事物的**特征**(feature)。可以基于特征定义**距离**(distance)，用于度量事物间的相似程度，距离的定义可以有各种形式。同类事物之间的相似度高，相互间的特征距离就小；反之，不同类的事物之间相似度低，特征距离就大。

给定样本特征，将其划归某个类别，这就是**分类**(classification)问题，或称为**识别**(recognition)问题。与回归分析相比，回归是一种定量分析，而分类则是一种定性分析。分类是人类最基本的智力活动之一，分类问题也因此成为机器学习中最基本、研究也最为广泛的问题之一。如何让计算机具备学习能力，解决分类问题呢？首先需要对分类问题进行抽象、建模，然后基于特征建立**判别函数**(discriminant function，或称决策函数)，并根据判别函数进行分类决策。机器学习将分类所用的判别函数称作**分类器**(classifier)。

可以借助统计学相关的理论与方法来解决分类问题。类别是一个集合概念，例如苹果、香蕉是两类水果，分别代表两类水果的集合。从概率论角度，类别是一种离散型随机变量，其值域是一个离散的类别集合，例如{苹果，香蕉}；特征也是随机变量(离散型或连续型)，不同类别的特征服从不同的概率分布，例如苹果与香蕉的颜色就服从不同的概率分布。各类别的特征概率分布被称为该类别的**模式**(pattern)，基于概率分布进行分类的问题也因此被称为**模式识别**(pattern recognition)问题。**贝叶斯**(Bayes)决策是统计决策中一种建立判

别函数、解决分类问题的基本方法。

也可以借助计算机科学中的算法设计思想,直接基于特征(而不是特征的概率分布)来设计分类算法,建立判别函数,从而解决分类问题,例如 k 近邻方法、线性判别分析、决策树等。本章将这些分类器统称为**非贝叶斯**决策。

3.1 贝叶斯分类器

在分类问题中,分类错误率(或正确率)是一个重要的评价标准。所谓贝叶斯分类器,就是给定特征 X,然后基于条件概率 $P(Y=c_i \mid X)$ 进行决策分类,将类别 Y 判定为条件概率最大的 c_i。这种分类规则能使分类错误率最小(即正确率最大),因此贝叶斯分类器是一种错误率最小的分类器。应用贝叶斯分类器解决分类问题,首先需通过样本训练集建立起问题的概率模型。

3.1.1 贝叶斯决策

由已知条件推出未知结论,这就是逻辑推理。如果已知条件、未知结论都是随机的,需要由已知条件推出未知结论的概率,这就是概率推理。贝叶斯公式是概率论中一个基本的概率推理公式。

1. 贝叶斯公式与贝叶斯决策

贝叶斯公式 设离散型随机变量 Y 的值域为 $\Omega_Y=\{c_1,c_2,\cdots,c_n\}$ 且 $P(Y=c_k)>0$,$k=1,2,\cdots,n$,则对任意的随机变量 X,$P(X)>0$,有

$$P(Y=c_k \mid X)=\frac{P(X,Y=c_k)}{P(X)}$$

$$=\frac{P(X \mid Y=c_k)P(Y=c_k)}{\sum_{k=1}^{n}P(X \mid Y=c_k)P(Y=c_k)},\quad k=1,2,\cdots,n \tag{3-1}$$

或将 $Y=c_k$ 简写成 Y_k,有

$$P(Y_k \mid X)=\frac{P(X,Y_k)}{P(X)}=\frac{P(X \mid Y_k)P(Y_k)}{\sum_{k=1}^{n}P(X \mid Y_k)P(Y_k)},\quad k=1,2,\cdots,n \tag{3-2}$$

式(3-1)或式(3-2)被称为贝叶斯公式。贝叶斯公式有什么用处呢?假设用 Y 表示原因,X 表示结果,则式(3-2)可改写为

$$P(\text{原因}_k \mid \text{结果})=\frac{P(\text{结果} \mid \text{原因}_k)P(\text{原因}_k)}{\sum_{k=1}^{n}P(\text{结果} \mid \text{原因}_k)P(\text{原因}_k)},\quad k=1,2,\cdots,n$$

贝叶斯公式的意义在于它建立了概率 $P(\text{结果}\mid\text{原因})$ 与 $P(\text{原因}\mid\text{结果})$ 之间的桥梁。如果已知什么原因可能导致什么结果,则通过贝叶斯公式就可以在已知结果的情况下反推其背后的原因。这一点非常有用,例如医学知识告诉我们什么疾病(病因)可能有什么症状(结果),利用贝叶斯公式就可以反过来通过症状去诊断疾病。在分类问题中,特征是可观测的结果,

类别是不可观测的原因，分类就是依据特征去推测类别。

依据随机事件的概率进行推理与决策，这就是统计决策。基于贝叶斯公式进行概率推理与决策，这就是贝叶斯决策。

贝叶斯决策　设离散型随机变量 Y 的值域为 $\Omega_Y=\{c_1,c_2,\cdots,c_n\}$ 且 $P(Y_k)>0$，$k=1,2,\cdots,n$，对任意的随机变量 X，$P(X)>0$，如果

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

则判定：给定 X 时 $Y=c_{k^*}$。

将贝叶斯决策规则应用于分类问题，式(3-3)中的随机变量 Y 表示类别(共 n 个类别)，X 表示分类特征。假设已求得 $P(Y=c_k|X)$，$k=1,2,\cdots,n$，则给定特征 X 时可基于式(3-3)的贝叶斯决策规则判定，特征 X 所描述事物的类别为 c_{k^*}。这时，式(3-3)就是一个分类判别函数，它被称为**贝叶斯分类器**(Bayes classifier)。可以证明，在已知概率分布的情况下，贝叶斯分类器是错误率最小的分类器。因为式(3-2)中 $P(Y_k|X)$ 的计算公式具有共同的分母 $P(X)$，所以式(3-3)等价于

$$k^* = \underset{k=1,2,\cdots,n}{\operatorname{argmax}} P(X,Y_k) = \underset{k=1,2,\cdots,n}{\operatorname{argmax}} P(X \mid Y_k)P(Y_k) \tag{3-4}$$

设计贝叶斯分类器，最主要的工作就是通过样本数据获得(估计)分类所需的概率分布。通过样本数据能比较容易地估计出特征概率分布 $P(X)$、类别概率分布 $P(Y_k)$，以及各类的特征条件概率 $P(X|Y_k)$。而类别条件概率 $P(Y_k|X)$ 则需要使用式(3-2)的贝叶斯公式，通过 $P(X)$、$P(Y_k)$ 和 $P(X|Y_k)$ 计算出来，在此基础上再使用式(3-3)进行分类决策。如果改用式(3-4)的判别函数，则不需要知道 $P(X)$，只需要 $P(Y_k)$ 和 $P(X|Y_k)$ 就可以进行分类决策，这样可以有效简化分类决策的计算过程。

式(3-2)中，$P(Y_k|X)$ 为类别条件概率，$P(X,Y_k)$ 为联合概率分布，$P(X)$ 为特征概率分布，$P(Y_k)$ 为类别概率分布，$P(X|Y_k)$ 为各类的特征条件概率。其中，类别概率分布 $P(Y_k)$ 是给定样本特征之前的先验知识，贝叶斯决策将其称为类别分布的**先验概率**(prior probability)；而类别条件概率 $P(Y_k|X)$ 是获得样本特征信息 X 之后对先验概率 $P(Y_k)$ 的更新，因此被称为类别分布的**后验概率**(posterior probability)。贝叶斯公式可以被看作一个将先验概率更新为后验概率的公式。

2. 贝叶斯分类器举例

下面就以一个苹果分类的例子，详细讲解贝叶斯决策的过程。红富士和国光是两个非常知名的苹果品种，如图 3-1 所示。

(a) 红富士苹果　　(b) 国光苹果

图 3-1　两个知名的苹果品种

红富士苹果(见图 3-1(a))为大型果,单果重 180～400g;底色淡黄或黄绿,着暗红或鲜红色霞;外形呈扁圆或近似圆形;口感甜脆。国光苹果(见图 3-1(b))为中型果,平均果重 150g 左右;底色黄绿或绿色;果实为扁圆形;口感酸甜。假设随机采集到如表 3-1 所示的 10 个样本苹果,希望能基于这个样本数据集建立起苹果的品种分类模型。本例只有两个类别(红富士或国光),机器学习将只有两个类别的分类问题称作**二分类**(binary classification)问题。相应地,将具有多个类别的分类问题称作**多分类**(multi-class classification)问题。

表 3-1 红富士和国光苹果样本(样本容量为 10)

编号	底色	外形	口感	果重/g	品种
1	黄	圆	甜	190	红富士
2	黄绿	扁圆	酸甜	260	红富士
3	绿	扁圆	酸甜	150	国光
4	黄绿	圆	甜	200	红富士
5	绿	扁圆	酸甜	210	国光
6	黄绿	扁圆	酸甜	170	国光
7	黄	圆	酸甜	200	红富士
8	黄绿	扁圆	酸甜	230	红富士
9	绿	扁圆	甜	180	国光
10	黄绿	扁圆	酸甜	240	红富士

表 3-1 是原始苹果样本数据,其中的底色、外形、口感和果重是 4 个可用于分类的特征数据 X,品种则是分类的目标 Y。建立分类模型,需要先估计概率分布 $P(X)$、$P(Y)$和 $P(X|Y)$,然后使用式(3-2)的贝叶斯公式计算出后验概率 $P(Y|X)$,最后再使用式(3-3)的贝叶斯分类器进行分类。

1) 使用单个离散型特征的贝叶斯分类器

假设选取表 3-1 中的“口感”作为苹果品种的分类特征 X,这是一个离散型随机变量,其值域为{甜,酸甜},只有两个取值。记“甜”为 x_1,“酸甜”为 x_2,特征 X 的概率分布包含两项,即 $P(X=x_1)$、$P(X=x_2)$。可以统计样本数据集“口感”一列“甜”和“酸甜”出现的次数来估计特征 X 的概率分布 $P(X)$。记分类目标“品种”为 Y,其值域为{红富士,国光},记“红富士”为 c_1,“国光”为 c_2,同理可以通过样本数据集估计出类别概率分布 $P(Y)$。估计概率分布 $P(X)$、$P(Y)$的过程及结果如表 3-2 所示。

表 3-2 特征概率分布 $P(X)$与类别概率分布 $P(Y)$

类　别	口感：X		品种：Y	
	甜：x_1	酸甜：x_2	红富士：c_1	国光：c_2
出现次数	$X=x_1$：3		$Y=c_1$：6	
	$X=x_2$：7		$Y=c_2$：4	
概率分布	$P(X=x_1)=3/10$		$P(Y=c_1)=6/10$	
	$P(X=x_2)=7/10$		$P(Y=c_2)=4/10$	

估计特征条件概率 $P(X|Y)$时需要先给定类别 Y(红富士或国光),然后将该类别的样本数据筛选出来并生成子集。估计特征 X 在子集上的概率分布,其结果就是类别 Y 的特征

条件概率 $P(X|Y)$。表 3-3 给出了红富士、国光苹果特征条件概率 $P(X|Y)$的估计过程及结果。

表 3-3　红富士与国光苹果的特征条件概率 $P(X|Y)$

类　　别	品种 Y			
	口感 $X\|Y=$红富士 c_1		口感 $X\|Y=$国光 c_2	
	甜：x_1	酸甜：x_2	甜：x_1	酸甜：x_2
出现次数	$X=x_1$：2 $X=x_2$：4		$X=x_1$：1 $X=x_2$：3	
概率分布	$P(X=x_1\|Y=c_1)=2/6$ $P(X=x_2\|Y=c_1)=4/6$		$P(X=x_1\|Y=c_2)=1/4$ $P(X=x_2\|Y=c_2)=3/4$	

表 3-2、表 3-3 的 $P(X)$、$P(Y)$和 $P(X|Y)$，就构成了一个完整的苹果分类问题的概率分布模型。任意给定新样本的口感特征 X，例如口感“酸甜”，可表示为($X=x_2$)，可以使用式(3-2)所示的贝叶斯公式计算出后验概率 $P(Y|X)$，即

$$P(Y_1 \mid X)=\frac{P(X,Y_1)}{P(X)}=\frac{P(X=x_2 \mid Y_1)P(Y_1)}{P(X=x_2)}=\frac{\frac{4}{6}\times\frac{6}{10}}{\frac{7}{10}}=\frac{4}{7}$$

或

$$P(Y_1 \mid X)=\frac{P(X,Y_1)}{P(X)}=\frac{P(X=x_2 \mid Y_1)P(Y_1)}{\sum_{i=1}^{2}P(X=x_2 \mid Y_i)P(Y_i)}=\frac{\frac{4}{6}\times\frac{6}{10}}{\frac{4}{6}\times\frac{6}{10}+\frac{3}{4}\times\frac{4}{10}}=\frac{4}{7}$$

同理，可得

$$P(Y_2 \mid X)=\frac{P(X,Y_2)}{P(X)}=\frac{P(X=x_2 \mid Y_2)P(Y_2)}{P(X=x_2)}=\frac{\frac{3}{4}\times\frac{4}{10}}{\frac{7}{10}}=\frac{3}{7}$$

因为 $P(Y_1|X)>P(Y_2|X)$，根据式(3-3)所示的贝叶斯决策，将口感酸甜的苹果判定为 c_1(红富士)。

如果改用式(3-4)所示的贝叶斯决策，则不需要知道特征概率分布 $P(X)$，只需计算

$$P(X=x_2 \mid Y_1)P(Y_1)=\frac{4}{6}\times\frac{6}{10}=\frac{4}{10}$$

$$P(X=x_2 \mid Y_2)P(Y_2)=\frac{3}{4}\times\frac{4}{10}=\frac{3}{10}$$

因为 $P(X=x_2|Y_1)P(Y_1)>P(X=x_2|Y_2)P(Y_2)$，所以将口感酸甜的苹果判定为 c_1(红富士)。这与式(3-3)所示的贝叶斯决策的结论相同，但计算过程简化了。

2) 使用两个离散型特征的贝叶斯分类器

假设选取表 3-1 中的“口感”和“外形”两项作为苹果品种的分类特征，这是两个离散型随机变量，可记作 $X=(X_1,X_2)$，其值域是{甜，酸甜}和{圆，扁圆}的组合，即{(甜，圆)，

(甜,扁圆),(酸甜,圆),(酸甜,扁圆)},共有 4 种取值。记“甜”为 x_{11},“酸甜”为 x_{12};再记“圆”为 x_{21},“扁圆”为 x_{22},特征 X 的概率分布包含 4 项,即 $P(X=(x_{11},x_{21}))$、$P(X=(x_{11},x_{22}))$、$P(X=(x_{12},x_{21}))$和 $P(X=(x_{12},x_{22}))$。

可以统计样本数据集的数据,估计出特征 X 的概率分布 $P(X_1,X_2)$、类别概率分布 $P(Y)$,还有特征条件概率 $P(X_1,X_2|Y)$,上述估计过程及结果如表 3-4、表 3-5 所示。

表 3-4 特征概率分布 $P(X_1,X_2)$与类别概率分布 $P(Y)$

类 别	(口感,外形):$X=(X_1,X_2)$	品种:Y	
	甜 x_{11},酸甜 x_{12};圆 x_{21},扁圆 x_{22}	红富士:c_1	国光:c_2
出现次数	$X=(x_{11},x_{21})$:2 $X=(x_{11},x_{22})$:1 $X=(x_{12},x_{21})$:1 $X=(x_{12},x_{22})$:6	$Y=c_1$:6 $Y=c_2$:4	
概率分布	$P(X=(x_{11},x_{21}))=2/10$ $P(X=(x_{11},x_{22}))=1/10$ $P(X=(x_{12},x_{21}))=1/10$ $P(X=(x_{12},x_{22}))=6/10$	$P(Y=c_1)=6/10$ $P(Y=c_2)=4/10$	

表 3-5 红富士与国光苹果的特征条件概率 $P(X_1,X_2|Y)$

类 别	品种 Y	
	(口感,外形)$X|Y=$红富士 c_1	(口感,外形)$X|Y=$国光 c_2
	甜 x_{11},酸甜 x_{12};圆 x_{21},扁圆 x_{22}	甜 x_{11},酸甜 x_{12};圆 x_{21},扁圆 x_{22}
出现次数	$X=(x_{11},x_{21})$:2 $X=(x_{11},x_{22})$:0 $X=(x_{12},x_{21})$:1 $X=(x_{12},x_{22})$:3	$X=(x_{11},x_{21})$:0 $X=(x_{11},x_{22})$:1 $X=(x_{12},x_{21})$:0 $X=(x_{12},x_{22})$:3
概率分布	$P(X=(x_{11},x_{21})|Y=c_1)=2/6$ $P(X=(x_{11},x_{22})|Y=c_1)=0/6$ $P(X=(x_{12},x_{21})|Y=c_1)=1/6$ $P(X=(x_{12},x_{22})|Y=c_1)=3/6$	$P(X=(x_{11},x_{21})|Y=c_2)=0/4$ $P(X=(x_{11},x_{22})|Y=c_2)=1/4$ $P(X=(x_{12},x_{21})|Y=c_2)=0/4$ $P(X=(x_{12},x_{22})|Y=c_2)=3/4$

表 3-4、表 3-5 的 $P(X_1,X_2)$、$P(Y)$和 $P(X_1,X_2|Y)$,就构成了一个完整的苹果分类问题的概率分布模型,其中使用了两个离散型分类特征,需计算它们的联合概率分布或联合条件概率分布。任意给定新样本的口感、外形特征 X,例如(酸甜,扁圆),可表示为($X=(x_{12},x_{22})$)。使用式(3-4)所示的贝叶斯决策,有

$$P(X=(x_{12},x_{22})\mid Y_1)P(Y_1)=\frac{3}{6}\times\frac{6}{10}=\frac{3}{10}$$

$$P(X=(x_{12},x_{22})\mid Y_2)P(Y_2)=\frac{3}{4}\times\frac{4}{10}=\frac{3}{10}$$

因为 $P(X=(x_{12},x_{22})|Y_1)P(Y_1)$、$P(X=(x_{12},x_{22})|Y_2)P(Y_2)$两者相等，所以可以将具有(酸甜，扁圆)特征的苹果判定为 c_1(红富士)，也可以判定为 c_2(国光)。

3) 使用离散型、连续型混合特征的贝叶斯分类器

假设选取表 3-1 中的“口感”和“果重”两项作为苹果品种的分类特征，“口感”特征是离散型随机变量，而“果重”特征是连续型随机变量。统计样本数据集的数据，可以很容易地估计出离散型随机变量的概率分布，而估计连续型随机变量的概率分布则相对比较复杂。

估计连续型随机变量概率分布常用的方法：先假设随机变量服从某种概率分布，例如正态分布 $N(\mu,\sigma^2)$，然后使用极大似然估计方法估计出其中的参数 μ 和 σ，最终求得随机变量的概率密度函数。表 3-6、表 3-7 给出了估计离散型、连续型混合特征联合概率分布 $P(X_1,X_2)$、类别概率分布 $P(Y)$和混合特征联合条件概率分布 $P(X_1,X_2|Y)$的大致过程。习惯上，大写字母 P 表示离散型随机变量的概率分布或连续型随机变量的分布函数，小写字母 p 表示连续型随机变量的概率密度函数。

表 3-6　特征概率分布 $P(X_1,X_2)$与类别概率分布 $P(Y)$

类　　别	(口感，果重)：$X=(X_1,X_2)$	品种：Y	
	甜 x_{11}，酸甜 x_{12}；果重 $x_2\in R$	红富士：c_1	国光：c_2
出现次数	$X=(x_{11},x_2)$：3 $X=(x_{12},x_2)$：7	$Y=c_1$：6 $Y=c_2$：4	
概率分布	$P(X_1=x_{11})=3/10$ $P(X_1=x_{12})=7/10$ $p_1(x_2\|X_1=x_{11})\sim N(\mu_1,\sigma_1^2)$ $p_2(x_2\|X_1=x_{12})\sim N(\mu_2,\sigma_2^2)$ $P(X_1,X_2)=P(X_2\|X_1)P(X_1)$	$P(Y=c_1)=6/10$ $P(Y=c_2)=4/10$	

表 3-7　红富士与国光苹果的特征条件概率 $P(X_1,X_2|Y)$

类　　别	品种 Y	
	(口感，果重)$X|Y=$红富士 c_1	(口感，果重)$X|Y=$国光 c_2
	甜 x_{11}，酸甜 x_{12}；果重 $x_2\in R$	甜 x_{11}，酸甜 x_{12}；果重 $x_2\in R$
出现次数	$X=(x_{11},x_2)$：2 $X=(x_{12},x_2)$：4	$X=(x_{11},x_2)$：1 $X=(x_{12},x_2)$：3
概率分布	$P(X_1=x_{11})=2/6$ $P(X_1=x_{12})=4/6$ $p_1(x_2\|X_1=x_{11})\sim N(\mu_1,\sigma_1^2)$ $p_2(x_2\|X_1=x_{12})\sim N(\mu_2,\sigma_2^2)$ $P(X_1,X_2\|Y=c_1)=P(X_2\|X_1)P(X_1)$	$P(X_1=x_{11})=1/4$ $P(X_1=x_{12})=3/4$ $p_3(x_2\|X_1=x_{11})\sim N(\mu_3,\sigma_3^2)$ $p_4(x_2\|X_1=x_{12})\sim N(\mu_4,\sigma_4^2)$ $P(X_1,X_2\|Y=c_2)=P(X_2\|X_1)P(X_1)$

估计表 3-6 的混合特征联合概率分布 $P(X_1,X_2)$时，需先按离散型特征“口感”(X_1，甜或酸甜)将样本数据拆分成两个子集；然后使用极大似然估计方法估计出连续型特征“果重”X_2 在子集上的概率密度函数 $p(x_2|X_1)$，例如服从正态分布 $N(\mu,\sigma^2)$时需估计其参数 μ 和 σ；最后再计算混合特征的联合概率分布 $P(X_1,X_2)=P(X_2|X_1)P(X_1)$。对于表 3-7 的

混合特征联合条件概率分布 $P(X_1,X_2|Y)$,可以使用类似的方法进行估计。

表 3-6、表 3-7 的 $P(X_1,X_2)$、$P(Y)$和 $P(X_1,X_2|Y)$,就构成了一个完整的苹果分类问题的概率分布模型,其中使用了一个离散型特征和一个连续型特征。可以看出,随着特征数的增加、离散型与连续型的混合,特征联合概率密度的估计难度不断加大。贝叶斯分类器的难点在于概率分布的估计。

在已知概率分布的情况下,贝叶斯分类器在理论上是错误率最小的分类器。但实际应用中,由于估计多个特征之间联合概率分布的难度非常大,因此贝叶斯分类器难以实施。贝叶斯分类器通常被作为研究分类器性能时的一种基准模型。

3.1.2 朴素贝叶斯与参数估计

贝叶斯分类器的难点在于概率分布估计,特别是高维特征的联合概率分布,其估计难度非常大。一个 d 维特征的联合概率分布 $P(X_1,X_2,\cdots,X_d)$,可以使用贝叶斯公式分解成如下形式,

$$\begin{aligned}P(X_1,X_2,\cdots,X_d)&=P(X_1\mid X_2,\cdots,X_d)P(X_2,\cdots,X_d)\\&=P(X_1\mid X_2,\cdots,X_d)P(X_2\mid X_3,\cdots,X_d)P(X_3,\cdots,X_d)\\&=P(X_1\mid X_2,\cdots,X_d)P(X_2\mid X_3,\cdots,X_d)\cdots P(X_{d-1}\mid X_d)P(X_d)\end{aligned}$$

假设所有特征之间相互独立,则联合概率分布可简化为

$$P(X_1,X_2,\cdots,X_d)=P(X_1)P(X_2)\cdots P(X_d)=\prod_{i=1}^{d}P(X_i) \tag{3-5}$$

同理,如果假设给定类别条件下所有特征之间相互独立,则

$$P(X_1,X_2,\cdots,X_d\mid Y)=P(X_1\mid Y)P(X_2\mid Y)\cdots P(X_d\mid Y)=\prod_{i=1}^{d}P(X_i\mid Y) \tag{3-6}$$

这样,d 维特征联合概率分布的估计问题就简化为 d 个一维特征概率分布的估计问题。这种简化具有非常大的应用价值,可以有效降低概率估计的难度。

所有特征之间相互独立,这是一个限制性很强的假设,基于这种假设所设计的贝叶斯分类器被称作**朴素贝叶斯**(naive Bayes)分类器。除了特征联合条件概率分布 $P(X_1,X_2,\cdots,X_d|Y)$的估计方法不同之外,朴素贝叶斯分类器与贝叶斯分类器在其他方面都是一样的。

在朴素贝叶斯分类器中,d 维联合概率分布的估计问题被简化为一维特征概率分布的估计问题。下面对一维特征概率分布的估计问题做必要讲解,然后再给出一个关于乳腺癌诊断的分类器设计与编程实例。

1. 特征条件概率的估计

设计朴素贝叶斯分类器,最主要的工作就是通过样本数据获得(估计)分类所需的特征条件概率,即给定类别 $Y=y$ 的条件下各项特征 X_i 的概率分布 $P(X_i|Y=y)$。下面给出三种常用概率分布形式,以及它们的参数估计方法。

1) 0-1 分布与二项分布(离散型)

给定类别 $Y=y$,如果离散型特征 X(随机变量)只有两种取值(可表示成 0 或 1),其概

率分布为

$$P(X=1\mid Y=y)=p,\quad P(X=0\mid Y=y)=1-p,\quad 0<p<1$$

则称随机变量 X 服从参数为 p 的 **0-1** 分布，又称**伯努利**(Bernoulli)分布。如果独立重复 m 次试验，每次试验均服从参数为 p 的 **0-1** 分布，用 X 表示 m 次试验中 1 发生的次数，则称随机变量 X 服从参数为(m,p)的**二项**(binomial)分布。

给定类别 $Y=y$ 时的特征样本数据集$\{x_1,x_2,\cdots,x_m\}$，可以估计出参数 p。数据集的样本容量为 m，统计其中取值为 1 的样本点个数 m_1，则 $p=m_1/m$，即

$$P(X=1\mid Y=y)=\frac{m_1}{m},\quad P(X=0\mid Y=y)=1-\frac{m_1}{m} \tag{3-7}$$

例 3-1　证明 $p=m_1/m$ 是二项分布的极大似然估计。

证明：给定类别 $Y=y$ 时的特征样本数据集$\{x_1,x_2,\cdots,x_m\}$，其中各样本点服从相同的 0-1 分布，联合概率服从二项分布。该数据集的似然函数为

$$\begin{aligned}L(p)&=P(X=x_1\mid Y=y)P(X=x_2\mid Y=y)\cdots P(X=x_m\mid Y=y)\\&=\prod_{i=1}^{m_1}P(X=1\mid Y=y)\prod_{i=1}^{m-m_1}P(X=0\mid Y=y)\\&=p^{m_1}(1-p)^{m-m_1}\end{aligned}$$

两边取对数，得对数似然函数

$$\ln L(p)=m_1\ln p+(m-m_1)\ln(1-p)$$

求对数似然函数 $\ln L(p)$对参数 p 的导数，并令其等于零，即

$$\frac{\partial\ln L(p)}{\partial p}=\frac{m_1}{p}-\frac{m-m_1}{1-p}=0$$

解方程得 $p=m_1/m$，这就是使似然函数最大的参数取值，即二项分布的极大似然估计。

2) 多项分布(离散型)

给定类别 $Y=y$，如果离散型特征 X(随机变量)有 n 种取值(可表示为 $1\sim n$，$n>2$)，其概率分布为

$$P(X=i\mid Y=y)=p_i,\quad 0<p_i<1,\quad \sum_{i=1}^{n}p_i=1$$

则称 $P(X\mid Y=y)$服从**多项**(multinomial)分布。

给定类别 $Y=y$ 时的特征样本数据集$\{x_1,x_2,\cdots,x_m\}$，可以估计出参数$\{p_1,p_2,\cdots,p_n\}$。数据集的样本容量为 m，统计其中取值为 i 的样本点个数 m_i，则 $p_i=m_i/m$，即

$$P(X=i\mid Y=y)=\frac{m_i}{m},\quad i=1,2,\cdots,n \tag{3-8}$$

为防止样本欠采样造成概率 p_i 为零的情况，通常采用**拉普拉斯平滑**(Laplace smoothing)技术对 p_i 的计算公式进行修正。平滑后，$p_i=(m_i+1)/(m+n)>0$，即

$$P(X=i\mid Y=y)=\frac{m_i+1}{m+n},\quad i=1,2,\cdots,n$$

3) 高斯分布(连续型)

给定类别 $Y=y$,如果特征 X(随机变量)取连续值,值域 $\Omega_X=\{x:x\in R\}$,且其概率密度函数为

$$p(x\mid Y=y)=\frac{1}{\sqrt{2\pi}\sigma}\mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

则称 $P(X|Y=y)$服从参数为(μ,σ^2)的**正态**(normal)分布,又称**高斯**(Gaussian)分布。

给定类别 $Y=y$ 时的特征样本数据集$\{x_1,x_2,\cdots,x_m\}$,可以估计出参数 μ、σ^2。使用极大似然估计可得

$$\mu=\bar{x}=\frac{1}{m}\sum_{i=1}^{m}x_i,\quad \sigma^2=\frac{1}{m}\sum_{i=1}^{m}(x_i-\bar{x})^2 \tag{3-9}$$

例 3-2 证明式(3-9)中的 μ、σ^2 是高斯分布的极大似然估计。

证明:给定类别 $Y=y$ 时的特征样本数据集$\{x_1,x_2,\cdots,x_m\}$,其似然函数为

$$L(\mu,\sigma^2)=p(x_1\mid Y=y)p(x_2\mid Y=y)\cdots p(x_m\mid Y=y)=\prod_{i=1}^{m}\frac{1}{\sqrt{2\pi}\sigma}\mathrm{e}^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

两边取对数,得对数似然函数

$$\begin{aligned}\ln L(\mu,\sigma^2)&=-m\ln\sqrt{2\pi}-m\ln\sigma-\frac{1}{2\sigma^2}\sum_{i=1}^{m}(x_i-\mu)^2\\&=-m\ln\sqrt{2\pi}-\frac{m}{2}\ln\sigma^2-\frac{1}{2\sigma^2}\sum_{i=1}^{m}(x_i-\mu)^2\end{aligned}$$

求对数似然函数 $\ln L(p)$对参数 μ 和σ^2 的偏导数,并令其等于 0,即

$$\begin{cases}\dfrac{\partial\ln L(\mu,\sigma^2)}{\partial\mu}=\dfrac{1}{\sigma^2}\displaystyle\sum_{i=1}^{m}(x_i-\mu)=0\\[2ex]\dfrac{\partial\ln L(\mu,\sigma^2)}{\partial\sigma^2}=-\dfrac{m}{2\sigma^2}+\dfrac{1}{2\sigma^4}\displaystyle\sum_{i=1}^{m}(x_i-\mu)^2=0\end{cases}$$

解方程可得式(3-9),这就是使似然函数最大的参数取值,即高斯分布的极大似然估计。

2. 使用 scikit-learn 库提供的样本数据集练习编程

scikit-learn 为机器学习提供了很多练习用的数据集,例如练习回归任务的 boston house-prices dataset、diabetes dataset;练习分类任务使用的 iris dataset、digits dataset、breast cancer wisconsin dataset 等。本章选用其中的乳腺癌数据集 breast cancer wisconsin dataset 来讲解朴素贝叶斯分类器的设计过程及编程实现。

breast cancer wisconsin dataset 是一个乳腺癌诊断的数据集,其中包含 569 个病例,每个病例包含 30 项特征数据(均为连续型特征)和一项诊断结果(1 表示恶性(malignant),0 表示良性(benign))。诊断乳腺癌就是根据各项临床特征(即检查指标)判定乳腺癌是恶性肿瘤还是良性肿瘤。这是一个分类问题,更准确地说是一个二分类问题。

图 3-2 给出了一个下载乳腺癌诊断数据集的示例代码,它将数据集及其说明文档分别保存到 data 目录下的 breast_cancer.csv 文件和 breast_cancer.txt 中。

3. 使用 scikit-learn 库中的朴素贝叶斯模型

scikit-learn 库根据分类特征的概率分布形式,将朴素贝叶斯分类器模型封装成不同的

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

        #下载乳腺癌数据集，保存到本地文件breast_cancer.csv中
        from sklearn.datasets import load_breast_cancer
        cancer = load_breast_cancer()
        print(cancer.data.shape)

        df = pd.DataFrame(cancer.data, columns=cancer.feature_names)
        df['malignant'] = cancer['target']
        df.to_csv("./data/breast_cancer.csv", index=None)

        #将数据集的说明文档保存到本地文件breast_cancer.txt中
        file = open("./data/breast_cancer.txt", 'w')
        file.write(cancer.DESCR)
        file.close()

        (569, 30)
```

图 3-2　下载乳腺癌诊断数据集的示例代码

类，常用的有三个：**GaussianNB**（高斯分布特征）、**MultinomialNB**（多项分布特征）、**BernoulliNB**（伯努利分布特征，即二项分布），并将它们存放在 sklearn.naive_bayes 模块中。这些朴素贝叶斯类都实现了学习算法 **fit**()、预测算法 **predict**()和评价算法 **score**()。下面通过乳腺癌诊断数据集来具体讲解朴素贝叶斯类的使用方法。为了进行本章的分类器编程实战，这里新建一个 Jupyter 记事本文件，后面将在该文件中正式编写程序代码。

乳腺癌诊断数据集的各项临床检查指标都是连续型特征，因此需要选用高斯分布的朴素贝叶斯类 GaussianNB。使用 GaussianNB 类建立乳腺癌诊断模型（即分类模型）主要分为 3 步。

1）加载数据集

从下载到本地 data 目录下的 breast_cancer.csv 文件中加载乳腺癌诊断数据集，得到一个 DataFrame 类的二维表格 **cancer**，显示其形状（shape，即表格的行数和列数）。取出数据集中的特征（0～29 列），生成特征集 **X**；再取出诊断结果（第 30 列，列名为 malignant），生成目标集 **Y**。图 3-3 给出了示例代码。

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

        cancer = pd.read_csv("./data/breast_cancer.csv")
        print("shape=", cancer.shape)

        X = cancer.iloc[:, :30]
        Y = cancer["malignant"]  # 或：Y = cancer.iloc[:, 30]

        shape= (569, 31)
```

图 3-3　加载数据集的示例代码

2）拆分训练集和测试集

使用 sklearn.model_selection 模块中的函数 train_test_split()，将特征集、目标集再分别拆分成训练集和测试集，得到训练集的特征集 **X_train** 和目标集 **Y_train**、测试集的特征集 **X_test** 和目标集 **Y_test**，共 4 个子集。图 3-4 给出了示例代码。

```
In [2]: from sklearn.model_selection import train_test_split
        X_train,X_test,Y_train,Y_test = train_test_split(
                                        X, Y, test_size=0.2, random_state=2020)
        print("X_train:", X_train.shape, "Y_train:", Y_train.shape)
        print("X_test:", X_test.shape, "Y_test:", Y_test.shape)

        X_train: (455, 30) Y_train: (455,)
        X_test: (114, 30) Y_test: (114,)
```

图 3-4 拆分训练集和测试集的示例代码

3) 训练并测试模型

图 3-5 给出了使用 GaussianNB 类建立并训练乳腺癌诊断模型的示例代码。其中,诊断模型(即分类模型)是基于高斯分布的朴素贝叶斯分类器,训练模型使用的是训练集(X_train、Y_train);然后用训练好的模型对测试集 X_test 前两个病例样本进行分类预测,将预测结果 predict 与 Y_test 中的真实诊断结果 malignant 进行比对,1 表示恶性,0 表示良性;最后使用函数 score()计算模型在训练集(X_train、Y_train)和测试集(X_test、Y_test)上的**平均正确率**(mean accuracy)。

```
In [3]: from sklearn.naive_bayes import GaussianNB
        gnb = GaussianNB()
        gnb.fit(X_train, Y_train)

        Y1 = gnb.predict(X_test[:2])
        print(X_test.values[:2]); print()
        print("predict:", Y1," malignant:", Y_test.values[:2])
        print("mean accuracy on train:", gnb.score(X_train, Y_train))
        print("mean accuracy on test:", gnb.score(X_test, Y_test))

        [[2.321e+01 2.697e+01 1.535e+02 1.670e+03 9.509e-02 1.682e-01 1.950e-01
          1.237e-01 1.909e-01 6.309e-02 1.058e+00 9.635e-01 7.247e+00 1.558e+02
          6.428e-03 2.863e-02 4.497e-02 1.716e-02 1.590e-02 3.053e-03 3.101e+01
          3.451e+01 2.060e+02 2.944e+03 1.481e-01 4.126e-01 5.820e-01 2.593e-01
          3.103e-01 8.677e-02]
         [1.164e+01 1.833e+01 7.517e+01 4.125e+02 1.142e-01 1.017e-01 7.070e-02
          3.485e-02 1.801e-01 6.520e-02 3.060e-01 1.657e+00 2.155e+00 2.062e+01
          8.540e-03 2.310e-02 2.945e-02 1.398e-02 1.565e-02 3.840e-03 1.314e+01
          2.926e+01 8.551e+01 5.217e+02 1.688e-01 2.660e-01 2.873e-01 1.218e-01
          2.806e-01 9.097e-02]]

        predict: [0 1]  malignant: [0 1]
        mean accuracy on train: 0.9406593406593406
        mean accuracy on test: 0.9736842105263158
```

图 3-5 使用 GaussianNB 类建立并训练乳腺癌诊断模型的示例代码

从图 3-5 可以看出,使用 GaussianNB 类建立并训练乳腺癌诊断模型,其在训练集、测试上的诊断正确率分别为 94%和 97%左右。

3.1.3 逻辑斯谛回归与牛顿法

贝叶斯分类器需要先估计特征概率分布 $P(X)$、类别概率分布 $P(Y_i)$,以及各类的特征条件概率 $P(X|Y_i)$,然后使用贝叶斯公式计算出后验概率 $P(Y_i|X)$,最后再进行分类决策。能不能通过特征 X 直接估计出后验概率 $P(Y_i|X)$呢?对于二分类问题来说,答案是肯定的。给定样本数据集,可以使用逻辑斯谛回归方法直接估计后验概率 $P(Y_1|X)$、$P(Y_0|X)$。注:$P(Y_1|X)$、$P(Y_0|X)$分别是 $P(Y=1|X)$和 $P(Y=0|X)$的简写。

在二分类问题中，类别 Y 的取值只有两个，$\Omega_Y=\{1,0\}$，其后验概率 $P(Y|X)$ 服从 0-1 分布，即

$$P(Y_1 \mid X)=p,\quad P(Y_0 \mid X)=1-p,\quad 0<p<1$$
$$P(Y_1 \mid X)+P(Y_0 \mid X)=1$$

将 $P(Y_1|X)$ 与 $P(Y_0|X)$ 之比称作 0-1 分布的**几率**(odds)，将几率的自然对数称作**对数几率**(logit，记作 z)，其数学形式可表示为

$$\text{几率}=\frac{P(Y_1 \mid X)}{P(Y_0 \mid X)}=\frac{p}{1-p} \tag{3-10}$$

$$\text{对数几率 } z=\ln\frac{P(Y_1 \mid X)}{P(Y_0 \mid X)}=\ln\frac{p}{1-p} \tag{3-11}$$

整理式(3-11)可得

$$p=\frac{e^z}{1+e^z}=\frac{1}{1+e^{-z}} \tag{3-12}$$

式(3-12)的函数形式被称为**逻辑斯谛**函数，它是一种 sigmoid 函数(即 S 形函数)。式(3-11)、式(3-12)描述了 0-1 分布中对数几率 z 与概率 p 之间的函数关系。

针对二分类问题，给定样本特征 $X=\boldsymbol{x}$，可以使用逻辑斯谛回归方法预测后验概率分布 $P(Y|X=\boldsymbol{x})$ 的对数几率 z，然后使用式(3-12)计算出概率 p。逻辑斯谛回归有一个重要假设：0-1 分布的对数几率 z 与特征 $\boldsymbol{x}$ 之间存在线性关系，即

$$z=\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x} \tag{3-13}$$

其中，$\boldsymbol{\omega}$ 是未知参数。这个假设来源于自然界的生物增长模型，具有一定的合理性(参见 2.5.2 节)。将式(3-13)代入式(3-12)，得

$$p=\frac{e^z}{1+e^z}=\frac{e^{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}}}{1+e^{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}}} \tag{3-14}$$

由此可得后验概率 $P(Y_1|X=\boldsymbol{x})$、$P(Y_0|X=\boldsymbol{x})$ 的回归模型，即

$$P(Y_1 \mid X=\boldsymbol{x})=p=\frac{e^{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}}}{1+e^{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}}},\quad P(Y_0 \mid X=x)=1-p=\frac{1}{1+e^{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}}} \tag{3-15}$$

有了式(3-15)的后验概率回归模型，再结合式(3-3)的贝叶斯决策，这样所设计出的分类器被称为**逻辑斯谛回归**(logistic regression)分类器，或**对数几率回归**(logit regression)分类器。式(3-15)的回归模型中，$\boldsymbol{\omega}$ 是未知参数，需通过样本训练集并使用极大似然估计进行训练，确定出最优参数 $\boldsymbol{\omega}^*$。

1. 构造似然函数

给定二分类的样本数据训练集 $D_{\text{train}}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\}$，其中 $\boldsymbol{x}_i$ 是样本特征，y_i 是其对应的真实类别(0 或 1)，可以构造式(3-15)所示回归模型的似然函数，将使似然函数最大的参数取值 $\boldsymbol{\omega}^*$ 作为最优参数。

式(3-15)所示回归模型的似然函数可以定义为

$$L(\boldsymbol{\omega})=\prod_{i=1}^{m}P(Y=y_i \mid \boldsymbol{x}_i;\boldsymbol{\omega})$$

两边取自然对数，得到对数似然函数

$$\ln L(\boldsymbol{\omega})=\ln\prod_{i=1}^{m}P(Y=y_i\mid\boldsymbol{x}_i;\boldsymbol{\omega})=\sum_{i=1}^{m}\ln P(Y=y_i\mid\boldsymbol{x}_i;\boldsymbol{\omega}) \tag{3-16}$$

其中,后验概率 $P(Y=y_i|\boldsymbol{x}_i;\boldsymbol{\omega})$按 y_i 的取值可分为如下两种情况。

(1) 如果 $y_i=1$,则

$$P(Y=y_i\mid\boldsymbol{x}_i;\boldsymbol{\omega})=P(Y=1\mid\boldsymbol{x}_i;\boldsymbol{\omega})=\frac{e^{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i}}{1+e^{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i}}$$

$$\ln P(Y=1\mid\boldsymbol{x}_i;\boldsymbol{\omega})=\ln\frac{e^{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i}}{1+e^{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i}}=\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i-\ln(1+e^{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i}) \tag{3-17}$$

(2) 如果 $y_i=0$,则

$$P(Y=y_i\mid\boldsymbol{x}_i;\boldsymbol{\omega})=P(Y=0\mid\boldsymbol{x}_i;\boldsymbol{\omega})=\frac{1}{1+e^{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i}}$$

$$\ln P(Y=0\mid\boldsymbol{x}_i;\boldsymbol{\omega})=\ln\frac{1}{1+e^{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i}}=-\ln(1+e^{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i}) \tag{3-18}$$

可以构造一个新函数,将式(3-17)、式(3-18)的两个对数概率函数合并成一个,即

$$\begin{aligned}\ln P(Y=y_i\mid\boldsymbol{x}_i;\boldsymbol{\omega})=&\,y_i\ln P(Y=1\mid\boldsymbol{x}_i;\boldsymbol{\omega})\\&+(1-y_i)\ln P(Y=0\mid\boldsymbol{x}_i;\boldsymbol{\omega})\end{aligned} \tag{3-19}$$

其中,y_i、$1-y_i$ 必然一个是 0,另一个是 1,这相当于对等号右边两个对数概率项进行开关选择。

式(3-19)是第 i 个样本点的对数似然函数,其负数形式

$$-\ln P(Y=y_i\mid\boldsymbol{x}_i;\boldsymbol{\omega})=-[y_i\ln P(Y=1\mid\boldsymbol{x}_i;\boldsymbol{\omega})+(1-y_i)\ln P(Y=0\mid\boldsymbol{x}_i;\boldsymbol{\omega})]$$

被称为第 i 个样本点的**交叉熵**(cross entropy)。整个训练集 D_{train} 上的交叉熵为各样本点交叉熵之和,即

$$-\sum_{i=1}^{m}\ln P(Y=y_i\mid\boldsymbol{x}_i;\boldsymbol{\omega})$$

最大化似然函数等价于最小化交叉熵。交叉熵的作用相当于训练逻辑斯谛回归模型时的损失函数。

将式(3-17)、式(3-18)代入式(3-19),则

$$\ln P(Y=y_i\mid\boldsymbol{x}_i;\boldsymbol{\omega})=y_i[\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i-\ln(1+e^{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i})]+(1-y_i)[-\ln(1+e^{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i})]$$

整理可得

$$\ln P(Y=y_i\mid\boldsymbol{x}_i;\boldsymbol{\omega})=y_i\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i-\ln(1+e^{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i}) \tag{3-20}$$

将式(3-20)代入式(3-16)的对数似然函数,则

$$\ln L(\boldsymbol{\omega})=\sum_{i=1}^{m}[y_i\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i-\ln(1+e^{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i})] \tag{3-21}$$

最大化对数似然函数 $\ln L(\boldsymbol{\omega})$,等价于最小化其负值(即交叉熵),记作 $L_1(\boldsymbol{\omega})$,即

$$L_1(\boldsymbol{\omega})=-\ln L(\boldsymbol{\omega})=\sum_{i=1}^{m}[-y_i\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i+\ln(1+e^{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i})] \tag{3-22}$$

式(3-22)是关于$\boldsymbol{\omega}$ 的连续且高阶可导凸函数,可以使用梯度下降法、牛顿法等迭代算法求解

出最优参数$\boldsymbol{\omega}^*$，即

$$\boldsymbol{\omega}^* = \underset{\boldsymbol{\omega}}{\operatorname{argmin}} L_1(\boldsymbol{\omega}) \tag{3-23}$$

如果使用梯度下降法，其参数迭代公式为

$$\boldsymbol{\omega}^k = \boldsymbol{\omega}^{k-1} - a \cdot \nabla L_1(\boldsymbol{\omega}^{k-1})$$

如果使用牛顿法，则参数迭代公式为

$$\boldsymbol{\omega}^k = \boldsymbol{\omega}^{k-1} - \boldsymbol{H}^{-1}(\boldsymbol{\omega}^{k-1}) \times \nabla L_1(\boldsymbol{\omega}^{k-1}) \tag{3-24}$$

其中，

$$\nabla L_1(\boldsymbol{\omega}) = \begin{bmatrix} \dfrac{\partial L_1}{\partial \omega_1} \\ \dfrac{\partial L_1}{\partial \omega_2} \\ \vdots \\ \dfrac{\partial L_1}{\partial \omega_d} \end{bmatrix}, \quad \boldsymbol{H}(\boldsymbol{\omega}) = \begin{bmatrix} \dfrac{\partial^2 L_1}{\partial \omega_1 \omega_1} & \dfrac{\partial^2 L_1}{\partial \omega_1 \omega_2} & \cdots & \dfrac{\partial^2 L_1}{\partial \omega_1 \omega_d} \\ \dfrac{\partial^2 L_1}{\partial \omega_2 \omega_1} & \dfrac{\partial^2 L_1}{\partial \omega_2 \omega_2} & \cdots & \dfrac{\partial^2 L_1}{\partial \omega_2 \omega_d} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial^2 L_1}{\partial \omega_d \omega_1} & \dfrac{\partial^2 L_1}{\partial \omega_d \omega_2} & \cdots & \dfrac{\partial^2 L_1}{\partial \omega_d \omega_d} \end{bmatrix}$$

2. 牛顿法

梯度下降法、牛顿法都属于数值迭代算法。梯度下降法利用一阶导数（或一阶偏导数）确定下一个迭代点的搜索方向，牛顿法在此基础上进一步利用二阶导数（或二阶偏导数）确定出下一个迭代点的大致位置，因此牛顿法能够提高迭代算法的收敛速度。

先讨论一元函数的无约束最小化问题：$x^* = \min\limits_{x \in R} f(x)$。假设从 x^{k-1} 处出发，该如何搜索下一个迭代点 x^k，使得 $f(x^k) \leqslant f(x^{k-1})$呢？一元函数 $f(x)$在 x^{k-1} 处的二阶泰勒展开式为

$$f(x) = f(x^{k-1}) + f'(x^{k-1})(x - x^{k-1}) + \frac{1}{2} f''(x^{k-1})(x - x^{k-1})^2 + o((x - x^{k-1})^2)$$

令

$$\varphi(x) = f(x^{k-1}) + f'(x^{k-1})(x - x^{k-1}) + \frac{1}{2} f''(x^{k-1})(x - x^{k-1})^2 \approx f(x)$$

则函数 $\varphi(x)$是 $f(x)$的近似函数，且在 x^{k-1} 处的一阶导数、二阶导数与 $f(x)$相同，可以将 $\varphi(x)$的极值点作为最小化 $f(x)$的下一个迭代点。求 $\varphi(x)$的一阶导数，并令其等于 0，即

$$\varphi'(x) = f'(x^{k-1}) + f''(x^{k-1})(x - x^{k-1}) = 0$$

解得

$$x = x^{k-1} - \frac{f'(x^{k-1})}{f''(x^{k-1})}$$

将 $\varphi(x)$的极值点 x 作为下一个迭代点 x^k，即

$$x^k = x^{k-1} - \frac{f'(x^{k-1})}{f''(x^{k-1})} \tag{3-25}$$

式(3-25)就是牛顿法的迭代公式。

使用牛顿法求解一元函数 $f(x)$无约束最小化问题的算法过程如下：首先任意选取一个参数初值 x^0，然后按照式(3-25)的迭代公式，不断更新迭代，得到新的参数 $x^1, x^2, \cdots, x^k, \cdots$，使 $f(x)$的函数值逐步下降，即 $f(x^k) \leqslant f(x^{k-1})$；重复迭代过程，直到相邻两次迭代计算出的函数值基本没有变化，则 $f(x)$收敛至最小值，迭代结束，将当前参数 x^k 作为最优参数 x^*。

将一元函数的无约束最小化问题推广到多元函数，例如 d 元函数，即

$$\boldsymbol{x}^* = \min_{x \in R^d} f(\boldsymbol{x})$$

其中，$\boldsymbol{x}=(x_1, x_2, \cdots, x_d)^{\mathrm{T}}$ 是一个 d 维向量。从一元函数推广到 d 元函数，需将一阶、二阶导数分别替换成梯度向量(一阶偏导向量)$\nabla f(\boldsymbol{x})$和黑塞矩阵(二阶偏导矩阵)$\boldsymbol{H}(\boldsymbol{x})$。对于 d 元函数，牛顿法迭代公式的推导过程如下。

$$f(\boldsymbol{x}) = f(\boldsymbol{x}^{k-1}) + \nabla f(\boldsymbol{x}^{k-1})(\boldsymbol{x} - \boldsymbol{x}^{k-1}) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^{k-1})^{\mathrm{T}} \boldsymbol{H}(\boldsymbol{x}^{k-1})(\boldsymbol{x} - \boldsymbol{x}^{k-1}) + o((\boldsymbol{x} - \boldsymbol{x}^{k-1})^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{x}^{k-1}))$$

其中，

$$\nabla f(\boldsymbol{x}) = \begin{bmatrix} \dfrac{\partial f}{\partial x_1} \\ \dfrac{\partial f}{\partial x_2} \\ \vdots \\ \dfrac{\partial f}{\partial x_d} \end{bmatrix}, \quad \boldsymbol{H}(\boldsymbol{x}) = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1 x_1} & \dfrac{\partial^2 f}{\partial x_1 x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_1 x_d} \\ \dfrac{\partial^2 f}{\partial x_2 x_1} & \dfrac{\partial^2 f}{\partial x_2 x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_2 x_d} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial^2 f}{\partial x_d x_1} & \dfrac{\partial^2 f}{\partial x_d x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_d x_d} \end{bmatrix}$$

令

$$\varphi(\boldsymbol{x}) = f(\boldsymbol{x}^{k-1}) + \nabla f(\boldsymbol{x}^{k-1})(\boldsymbol{x} - \boldsymbol{x}^{k-1}) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^{k-1})^{\mathrm{T}} \boldsymbol{H}(\boldsymbol{x}^{k-1})(\boldsymbol{x} - \boldsymbol{x}^{k-1}) \approx f(\boldsymbol{x})$$

则函数 $\varphi(\boldsymbol{x})$是 $f(\boldsymbol{x})$的近似函数，可以将 $\varphi(\boldsymbol{x})$的极值点作为最小化 $f(\boldsymbol{x})$的下一个迭代点。求 $\varphi(\boldsymbol{x})$的一阶偏导数，并令其等于零，即

$$\nabla \varphi(\boldsymbol{x}) = \nabla f(\boldsymbol{x}^{k-1}) + \boldsymbol{H}(\boldsymbol{x}^{k-1})(\boldsymbol{x} - \boldsymbol{x}^{k-1}) = 0$$

如果 $\boldsymbol{H}(\boldsymbol{x})$可逆，解方程得

$$\boldsymbol{x} = \boldsymbol{x}^{k-1} - \boldsymbol{H}^{-1}(\boldsymbol{x}^{k-1})\, \nabla f(\boldsymbol{x}^{k-1})$$

将 $\varphi(\boldsymbol{x})$的极值点 $\boldsymbol{x}$ 作为下一个迭代点 $\boldsymbol{x}^k$，即

$$\boldsymbol{x}^k = \boldsymbol{x}^{k-1} - \boldsymbol{H}^{-1}(\boldsymbol{x}^{k-1})\, \nabla f(\boldsymbol{x}^{k-1}) \tag{3-26}$$

式(3-26)就是多元函数时的牛顿法迭代公式。

3. 使用 scikit-learn 库中的逻辑斯谛回归分类器模型

scikit-learn 库将逻辑斯谛回归分类器模型封装成一个类(类名为 **LogisticRegression**)，并将其存放在 sklearn.linear_model 模块中。LogisticRegression 类实现了逻辑斯谛回归分类器模型的学习算法 **fit**()、预测算法 **predict**()和评价算法 **score**()。

应用逻辑斯谛回归分类器，需要对样本的特征数据进行标准化(z-score 或 Min-Max)，

然后再将特征集、目标集进一步拆分成训练集和测试集。对乳腺癌诊断问题，图 3-6 给出了特征标准化和数据集拆分的示例代码。示例代码先对 3.1.2 节得到的特征集 X 进行标准化，得到新的特征集 X1，然后将 X1 和目标集 Y 按照 8:2 的比例拆分成训练集和测试集。拆分得到训练集的特征集 **X1_train** 和目标集 **Y_train**、测试集的特征集 **X1_test** 和目标集 **Y_test**，共 4 个子集。

```
In [4]: from sklearn.preprocessing import StandardScaler, MinMaxScaler

        scaler = StandardScaler()
        #scaler = MinMaxScaler()
        scaler.fit(X)
        X1 = scaler.transform(X)
        # X1 = scaler.fit_transform(X)

        from sklearn.model_selection import train_test_split
        X1_train, X1_test, Y_train, Y_test = train_test_split(
                                             X1, Y, test_size=0.2, random_state=2020)
        print("X1_train:", X1_train.shape, "Y_train:", Y_train.shape)
        print("X1_test:", X1_test.shape, "Y_test:", Y_test.shape)

           X1_train: (455, 30) Y_train: (455,)
           X1_test: (114, 30) Y_test: (114,)
```

图 3-6　特征标准化和数据集拆分的示例代码

图 3-7 给出了使用 LogisticRegression 类建立并训练乳腺癌诊断模型的示例代码。其中，诊断模型（即分类模型）采用的是逻辑斯谛回归分类器，训练模型使用的是训练集（X1_train、Y_train）；然后用训练好的模型对测试集 X1_test 前两个病例样本进行分类预测，将预测结果 predict 与 Y_test 中的真实诊断结果 malignant 进行比对，1 表示恶性，0 表示良性；最后使用函数 score()计算模型在训练集（X1_train、Y_train）和测试集（X1_test、Y_test）上的平均正确率。

```
In [5]: from sklearn.linear_model import LogisticRegression
        lr = LogisticRegression(penalty='l2', C=1.0, random_state=2020)
        lr.fit(X1_train, Y_train)

        Y1 = lr.predict(X1_test[:2])
        print(X1_test[:2]); print()
        print("predict:", Y1," malignant:", Y_test.values[:2])
        print("mean accuracy on train:", lr.score(X1_train, Y_train))
        print("mean accuracy on test:", lr.score(X1_test, Y_test))

           [[ 2.57961809  1.78726935  2.53447284  2.88707955 -0.09040012  1.21022282
              1.33334652  1.92889603  0.35553387  0.04144935  2.35619316 -0.45967024
              2.16869979  2.54038166 -0.20433459  0.17615607  0.43357064  0.87007041
             -0.56208259 -0.28062613  3.05256427  1.43836286  2.94101757  3.62730675
              0.6896002   1.00723154  1.48632804  2.20319388  0.32718689  0.1565044
           ]
            [-0.70642616 -0.22331665 -0.69195555 -0.68937948  1.26957147 -0.05005056
             -0.22723639 -0.3628993  -0.03876801  0.3405638  -0.35793278  0.79857725
             -0.35199608 -0.43380945  0.49969396 -0.13291308 -0.08102636  0.35424367
             -0.59235221  0.01705767 -0.64800054  0.58343296 -0.64787811 -0.63088521
              1.59700315  0.07465072  0.07249786  0.10953674 -0.15329395  0.3892508
           3]]

           predict: [0 1]  malignant: [0 1]
           mean accuracy on train: 0.989010989010989
           mean accuracy on test: 0.9736842105263158
```

图 3-7　使用 LogisticRegression 类建立并训练乳腺癌诊断模型的示例代码

4. 将逻辑斯谛回归推广至多分类

可以将逻辑斯蒂回归由二分类推广至多分类（假设为 N 分类），这时类别后验概率由二分类的 0-1 分布推广至多项分布。相应地，后验概率的函数形式也由逻辑斯谛函数推广至

归一化指数函数(或称作 **softmax** 函数),其数学形式为

$$z_k = \boldsymbol{\omega}_k^{\mathrm{T}}\boldsymbol{x}, \quad k=1,2,\cdots,N$$

$$z_{\text{total}} = \sum_{k=1}^{N} \mathrm{e}^{z_k} = \sum_{k=1}^{N} \mathrm{e}^{\boldsymbol{\omega}_k^{\mathrm{T}}\boldsymbol{x}}$$

$$P(Y_k \mid \boldsymbol{x}) = \mathrm{softmax}(z_k) = \frac{\mathrm{e}^{z_k}}{z_{\text{total}}}, \quad k=1,2,\cdots,N$$

其中,$\boldsymbol{x}$ 为样本特征,$\boldsymbol{\omega}_k$ 为第 k 类的回归系数;$P(Y_k \mid \boldsymbol{x})$为第 k 类的后验概率,$0<P(Y_k \mid \boldsymbol{x})<1$,且 $\sum_{k=1}^{N} P(Y_k \mid \boldsymbol{x})=1$。

给定一个多分类的样本特征 $\boldsymbol{x}$,使用逻辑斯谛回归方法分别估计出各类的后验概率 $P(Y_k \mid \boldsymbol{x})$,$k=1,2,\cdots,N$,然后将使 $P(Y_k \mid \boldsymbol{x})$最大的 Y_k 作为分类结果。

二分类时,第 i 个样本点的交叉熵(记作 H)为

$$\begin{aligned} H &= -\ln P(Y=y_i \mid \boldsymbol{x}_i;\boldsymbol{\omega}) \\ &= -[y_i \ln P(Y=1 \mid \boldsymbol{x}_i;\boldsymbol{\omega}) + (1-y_i)\ln P(Y=0 \mid \boldsymbol{x}_i;\boldsymbol{\omega})] \end{aligned}$$

而多分类时,通常会对样本类别进行 N 位 one-hot 编码,例如类别 1 的 one-hot 编码为 10…0,类别 2 的 one-hot 编码为 01…0,…,类别 N 的 one-hot 编码为 00…1。这时第 i 个样本点类别 y_i 的 one-hot 编码可记为 $y_i^1 y_i^2 \cdots y_i^N$,其交叉熵可写成

$$H = -\ln P(Y=y_i \mid \boldsymbol{x}_i;\boldsymbol{\omega}) = -\sum_{k=1}^{N} y_i^k \ln P(y_i=k \mid \boldsymbol{x}_i;\boldsymbol{\omega}), \quad y_i^k \in \{0,1\}$$

基于逻辑斯谛回归进行分类决策时,交叉熵的作用相当于训练模型时的损失函数。

5. 进一步理解交叉熵

设 $p(y)$、$q(y)$是随机变量 Y 上的两个概率分布,则 $q(y)$相对于 $p(y)$的交叉熵被定义为

$$\begin{cases} H(q,p) = -\sum\limits_{y\in\Omega_Y} q(y)\ln p(y), & \text{离散型} \\ H(q,p) = -\int_{\Omega_Y} q(y)\ln p(y)\mathrm{d}y, & \text{连续型} \end{cases}$$

其中,Ω_Y 为随机变量 Y 的值域。交叉熵描述了两种不同概率分布之间的差异程度,因此常被分类模型用作度量真实概率分布与预测概率分布之间误差的损失函数。注:交叉熵不是对称的,即 $H(q,p)\neq H(p,q)$。

例如,使用逻辑斯谛回归方法进行二分类时,给定样本 $\boldsymbol{x}_i$,可以估计出类别 $y\in\{0,1\}$ 的预测概率分布 $p(y)$,可记作 p_1、p_0。另外,可以将样本 $\boldsymbol{x}_i$ 的真实类别 y_i 也表示成概率分布的形式并记作 q_1、q_0。例如,$y_i=1$ 时的概率分布 $q(y)$为:$q_1=1$,$q_0=0$;$y_i=0$ 时的概率分布 $q(y)$则为:$q_1=0$,$q_0=1$。这时度量预测概率分布 $p(y)$与真实概率分布 $q(y)$的差异,可以使用交叉熵的形式,即

$$H(q,p) = -\sum_{y\in\{0,1\}} q(y)\ln p(y) = -(q_1\ln p_1 + q_0 \ln p_0)$$

对比式(3-19)可以看出,交叉熵与对数似然函数互为相反数。使用逻辑斯谛回归方法进行分类时,最大化对数似然函数就等价于最小化交叉熵损失函数。

3.2　非贝叶斯分类器

计算机科学是一门新兴学科，它善于从其他学科汲取营养，善于运用算法来解决问题。针对分类问题，本节介绍三种借助算法思想所设计的非贝叶斯分类器模型，它们分别是 k 近邻方法、线性判别分析和决策树。这三种方法都是直接基于特征（而不是特征的概率分布）来设计分类算法，建立判别函数，从而解决分类问题。

3.2.1　k 近邻分类器与距离度量

k 近邻（k-Nearest Neighbor，KNN）分类器是一种**基于实例**（instance-based）的分类器。这种分类器不建立任何模型，只是将训练集作为样本实例保存起来。假设给定训练集

$$D_{\text{train}}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\}$$

其中，$\boldsymbol{x}_i$ 是样本特征，$y_i\in\{c_1,c_2,\cdots,c_n\}$ 是其对应的实际类别，直接将训练集 D_{train} 以某种**数据结构**（data structure）保存起来。

给定新样本 $\boldsymbol{x}$，按照某种**距离度量**（distance metric）查找训练集中 k 个（例如 1 个、3 个或 5 个等）距离最近（称作**最近邻**，nearest neighbor）的样本实例，然后统计这 k 个样本实例的类别；最后的分类决策规则是，将出现次数最多的类别判定为新样本 $\boldsymbol{x}$ 的类别，这种分类决策规则被称为**简单多数表决**（simple majority vote）规则。k 近邻分类器直接基于训练集实例进行分类，没有显式的学习过程，也不需要学习算法。

1. 距离度量

如何对样本点特征之间的相似度（距离）进行度量，这是 k 近邻分类器最核心的内容。度量两个 d 维特征 $\boldsymbol{x}_i=\{x_{i1},x_{i2},\cdots,x_{id}\}$ 和 $\boldsymbol{x}_j=\{x_{j1},x_{j2},\cdots,x_{jd}\}$ 之间的相似度，通常使用**欧氏**（Euclidean）距离或**曼哈顿**（Manhattan）距离。**闵可夫斯基**（Minkowski）距离则给出了更一般形式的定义。将闵可夫斯基距离记作 L_p，其定义形式为

$$L_p(\boldsymbol{x}_i,\boldsymbol{x}_j)=\left(\sum_{l=1}^{d}|x_{il}-x_{jl}|^p\right)^{\frac{1}{p}},\quad p\geqslant 1 \tag{3-27}$$

如果 $p=1$，则闵可夫斯基距离就是曼哈顿距离，记作 $L_1(\boldsymbol{x}_i,\boldsymbol{x}_j)$ 或 $\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_1$。

$$L_1(\boldsymbol{x}_i,\boldsymbol{x}_j)=\sum_{l=1}^{d}|x_{il}-x_{jl}|$$

如果 $p=2$，则闵可夫斯基距离就是欧氏距离，记作 $L_2(\boldsymbol{x}_i,\boldsymbol{x}_j)$ 或 $\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2$。

$$L_2(\boldsymbol{x}_i,\boldsymbol{x}_j)=\sqrt{\sum_{l=1}^{d}|x_{il}-x_{jl}|^2}=\sqrt{\sum_{l=1}^{d}(x_{il}-x_{jl})^2}$$

如果 $p=+\infty$，则闵可夫斯基距离就是**切比雪夫**（Chebyshev）距离，记作 L_∞。

$$L_\infty(\boldsymbol{x}_i,\boldsymbol{x}_j)=\max_{l=1,2,\cdots,d}|x_{il}-x_{jl}|$$

2. 数据结构与查找算法

给定新样本 $\boldsymbol{x}$，k 近邻分类器需要逐个计算与训练集里各实例 $\boldsymbol{x}_i$ 的距离。如果使用顺

序表(例如数组或链表)存储训练集 D_{train},则查找算法的平均复杂度为 $O(N)$。

可以改用 **kd 树**(k-dimensional tree)来存储训练集 D_{train},这样能将查找 $\boldsymbol{x}$ 最近邻算法的平均复杂度降到 $O(\log N)$,有效提高算法的查找速度。kd 树由"数据结构"中的**二叉搜索树**(Binary Search Tree,BST)演变而来,常用于高维数据的存储与查找。如果想详细了解 kd 树及其查找算法,读者可进一步查阅相关资料。

3. 简单多数表决规则

给定新样本 $\boldsymbol{x}$,使用查找算法查找出训练集 D_{train} 中与 $\boldsymbol{x}$ 距离最近的 k 个实例。假设这 k 个实例为 $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_k, y_k)$,其中实例类别 $y_i \in \{c_1, c_2, \cdots, c_n\}$。统计 n 个类别在 k 个实例中出现的次数,将出现次数最多的类别判定为新样本 $\boldsymbol{x}$ 的类别。

4. 使用 scikit-learn 库中的 k 近邻分类器模型

scikit-learn 库将 k 近邻分类器模型封装成一个类(类名为 **KNeighborsClassifier**),并将其存放在 sklearn.neighbors 模块中。KNeighborsClassifier 类实现了 k 近邻分类器模型的学习算法 **fit**()、预测算法 **predict**()和评价算法 **score**()。

图 3-8 给出了使用 KNeighborsClassifier 类建立并训练乳腺癌诊断模型的示例代码。其中,诊断模型(即分类模型)采用的是 k 近邻分类器;数据集使用的是 3.1.3 节标准化后的乳腺癌诊断数据集,其中的训练集(X1_train、Y_train)被用作样本实例;然后用训练好的模型对测试集 X1_test 前两个病例样本进行分类预测,将预测结果 predict 与 Y_test 中的真实诊断结果 malignant 进行比对,1 表示恶性,0 表示良性;最后使用函数 score()计算模型在训练集(X1_train、Y_train)和测试集(X1_test、Y_test)上的平均正确率。

```
In [6]: from sklearn.neighbors import KNeighborsClassifier
        knc = KNeighborsClassifier(n_neighbors=3, p=2, metric='minkowski')
        knc.fit(X1_train, Y_train)

        Y1 = knc.predict(X1_test[:2])
        print(X1_test[:2]);  print()
        print("predict:", Y1," malignant:", Y_test.values[:2])
        print("mean accuracy on train:", knc.score(X1_train, Y_train))
        print("mean accuracy on test:", knc.score(X1_test, Y_test))

[[ 2.57961809  1.78726935  2.53447284  2.88707955 -0.09040012  1.21022282
   1.33334652  1.92889603  0.35553387  0.04144935  2.35619316 -0.45967024
   2.16869979  2.54038166 -0.20433459  0.17615607  0.43357064  0.87007041
  -0.56208259 -0.28062613  3.05256427  1.43836286  2.94101757  3.62730675
   0.6896002   1.00723154  1.48632804  2.20319388  0.32718689  0.1565044
]
 [-0.70642616 -0.22331665 -0.69195555 -0.68937948  1.26957147 -0.05005056
  -0.22723639 -0.3628993  -0.03876801  0.3405638  -0.35793278  0.79857725
  -0.35199608 -0.43380945  0.49969396 -0.13291308 -0.08102636  0.35424367
  -0.59235221  0.01705767 -0.64800054  0.58343296 -0.64787811 -0.63088521
   1.59700315  0.07465072  0.07249786  0.10953674 -0.15329395  0.3892508
3]]

predict: [0 1]  malignant: [0 1]
mean accuracy on train: 0.989010989010989
mean accuracy on test: 0.9385964912280702
```

图 3-8 使用 KNeighborsClassifier 类建立并训练乳腺癌诊断模型的示例代码

5. 超参数 k 的选择

k 近邻分类器模型中的 k 是一个超参数,其取值对分类结果有很大影响。图 3-9 给出一个例子,对于新样本 $\boldsymbol{x}$,不同 k 值会得到不同的分类结果。当 $k=1$ 时,k 近邻分类器也被称作最近邻分类器。

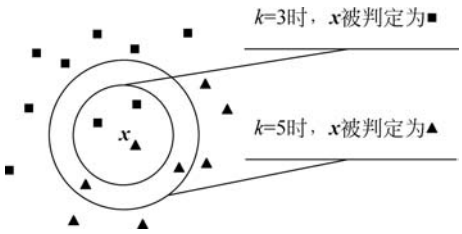


图 3-9　不同 k 值会得到不同的分类结果

对于乳腺癌诊断模型所采用的 k 近邻分类器，图 3-10、图 3-11 分别给出对比不同 k 值(1～9)时模型分类性能(其中采用了 5 折交叉验证，即 cv＝5)的示例代码和折线图。图 3-11 的折线图可以为超参数 k 值的选择提供参考。

```
In [7]: from sklearn.neighbors import KNeighborsClassifier
        from sklearn.model_selection import cross_val_score

        scores = []
        for k in range(1,10):
            knc = KNeighborsClassifier(n_neighbors=k,p=2,metric='minkowski')
            cv_results = cross_val_score(knc, X1_train, Y_train, cv=5)
            scores.append( np.mean(cv_results) )

        fig = plt.figure( figsize=(4, 3), dpi=100 )
        plt.rcParams['font.sans-serif'] = ['SimHei']
        plt.rcParams['axes.unicode_minus'] = False

        plt.plot(range(1,10), scores, 'r--', linewidth=2)
        plt.xlabel(r'$k$')
        plt.ylabel(r'平均正确率')
        plt.title(r'对比不同$k$值时模型的分类性能')
        plt.show()
```

图 3-10　对比不同 k 值(1～9)时模型分类性能的示例代码

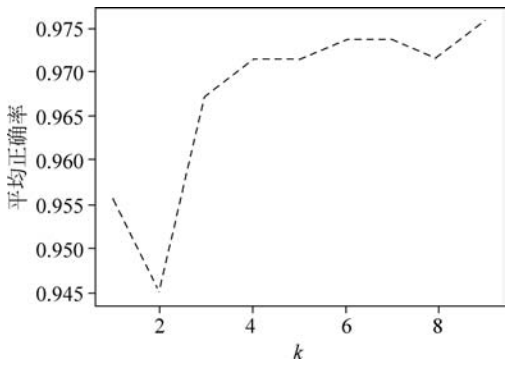


图 3-11　对比不同 k 值(1～9)时模型分类性能的折线图

3.2.2　线性判别分析与特征空间

贝叶斯分类器需要先估计特征概率分布 $P(X)$、类别概率分布 $P(Y_k)$，以及各类的特征条件概率 $P(X|Y_k)$，然后使用贝叶斯公式计算出后验概率 $P(Y_k|X)$，最后再进行分类决策。估计概率分布需要有足够多的样本数据。如果特征 X 包含的属性个数很多，即高维特征，则估计特征概率分布 $P(X)$、各类特征条件概率 $P(X|Y_k)$所需样本集的容量要求很

大，否则就会因样本稀疏造成概率估计的偏差。

对于二分类问题，**线性判别分析**(Linear Discriminant Analysis，LDA)方法的思路是：给定训练集，设法将其中的高维特征压缩到一维，然后基于一维特征来设计分类器，这样就能降低对样本集容量的要求。

1. 特征空间与向量投影

给定训练集 $D_{\text{train}}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\}$，其中 $\boldsymbol{x}_i$ 是样本特征，$y_i\in\{c_1,c_2,\cdots,c_n\}$是其对应的实际类别，$i=1,2,\cdots,m$。可以将每个样本点的特征看作向量空间里的一个向量，这种向量空间被称为**特征空间**(feature space)，其中的向量被称为**特征向量**(feature vector，这个特征向量与矩阵的特征向量不是一回事)。图 3-12 给出一个二维特征空间的示意图。

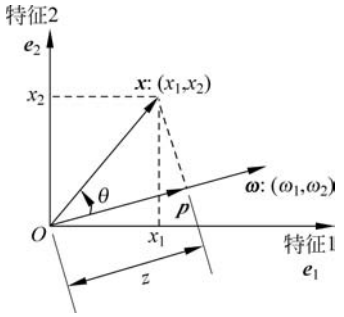


图 3-12 将每个样本点的特征看作特征空间中的一个向量

在图 3-12 所示的特征空间中，每一项特征属性都是一个**维度**(dimension)，d 项特征属性就构成一个 d 维空间。样本点 $\boldsymbol{x}$ 在特征空间中的坐标$(x_1,x_2,\cdots,x_d)$就是各属性的取值。例如，假设特征 1 表示苹果的“口感”，特征 2 表示“果重”，苹果样本 $\boldsymbol{x}$ 的坐标(甜，190)就表示 $\boldsymbol{x}$ 的口感是“甜”，果重为 190g。

样本点 $\boldsymbol{x}$ 的坐标实际上是向量 $\boldsymbol{x}$ 在各基向量(坐标轴)上的投影。例如图 3-12 中，样本点 $\boldsymbol{x}$ 的坐标(x_1,x_2)就是其在基向量 $\boldsymbol{e}_1$、$\boldsymbol{e}_2$ 上的投影 x_1、x_2。可以将样本点 $\boldsymbol{x}$ 投影到其他向量(例如$\boldsymbol{\omega}$)上。假设向量$\boldsymbol{\omega}$ 的坐标是$(\omega_1,\omega_2,\cdots,\omega_d)$，向量 $\boldsymbol{x}$ 在$\boldsymbol{\omega}$ 上的投影可分为**标量投影**(scalar projection，记作 z)与**向量投影**(vector projection，记作 $\boldsymbol{p}$)两种形式(参见图 3-12)，其计算公式分别为

$$z=\frac{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}}{\|\boldsymbol{\omega}\|} \tag{3-28}$$

$$\boldsymbol{p}=z\,\frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|} \tag{3-29}$$

其中，向量 $\boldsymbol{x}=(x_1,x_2,\cdots,x_d)^{\mathrm{T}}$，$\boldsymbol{\omega}=(\omega_1,\omega_2,\cdots,\omega_d)^{\mathrm{T}}$；$z$ 是标量，为 $\boldsymbol{x}$ 在$\boldsymbol{\omega}$ 上的标量投影；$\boldsymbol{p}$ 是向量，为 $\boldsymbol{x}$ 在$\boldsymbol{\omega}$ 上的向量投影；$\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}$ 是向量$\boldsymbol{\omega}$ 与向量 $\boldsymbol{x}$ 的内积(或称点积)，$\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}=\omega_1x_1+\omega_2x_2+\cdots+\omega_dx_d$；$\|\boldsymbol{\omega}\|$是向量$\boldsymbol{\omega}$ 的模(即长度)，$\|\boldsymbol{\omega}\|=\sqrt{\omega_1^2+\omega_2^2+\cdots+\omega_d^2}=\sqrt{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\omega}}$。

可以证明，$\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}=\|\boldsymbol{\omega}\|\,\|\boldsymbol{x}\|\cos\theta$($\theta$ 为$\boldsymbol{\omega}$ 与 $\boldsymbol{x}$ 之间的夹角)，由此可推导出式(3-28)。如果$\boldsymbol{\omega}$ 为单位向量，即$\|\boldsymbol{\omega}\|=1$，则式(3-28)可简化为

$$z=\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}=\omega_1x_1+\omega_2x_2+\cdots+\omega_dx_d \tag{3-30}$$

式(3-28)、式(3-30)的标量投影，实际上是通过线性组合方法将 d 维向量 $\boldsymbol{x}=(x_1,x_2,\cdots,x_d)^{\mathrm{T}}$ 压缩成$\boldsymbol{\omega}$ 方向上的一维标量 z，线性组合中的各项系数分别对应向量$\boldsymbol{\omega}$ 的各个分量。

2. 选择投影方向

对于二分类问题，线性判别分析方法希望在选择投影方向$\boldsymbol{\omega}$ 时，能够让样本特征投影点“类内方差最小，类间方差最大”。换句话说，投影后同类样本点越聚集越好，不同类的样本点

越远离越好。图 3-13 给出一个二维特征空间的投影示意图，其中的●、▲分别表示类 0 和类 1 的样本点，$\boldsymbol{v}$、$\boldsymbol{\omega}$ 分别表示两个不同的投影方向，$\boldsymbol{\mu}_0$、$\boldsymbol{\mu}_1$ 分别表示投影后类 0 和类 1 样本投影点的均值。可以看出，$\boldsymbol{\omega}$ 方向的投影效果比 $\boldsymbol{v}$ 方向要好，因为类 0 和类 1 在 $\boldsymbol{v}$ 方向上的投影点混杂在一起，很难分类。通过投影对原始特征进行变换，这是线性判别分析方法最核心的内容。

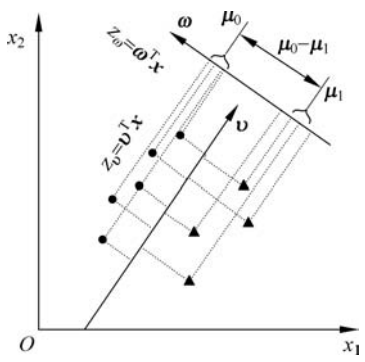


图 3-13　二维特征空间的投影示意图

下面用数学语言来描述选择最优投影方向$\boldsymbol{\omega}$ 的过程，整个过程分 4 步完成。给定训练集 $D_{\text{train}}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\}$，其中 $\boldsymbol{x}_i=(x_{i1},x_{i2},\cdots,x_{id})^{\mathrm{T}}$ 是 d 维样本特征，$y_i\in\{0,1\}$是其对应的实际类别。假设训练集 D_{train} 中 0 类样本有 m_0 个，1 类样本有 m_1 个，将数据集按类别拆分成子集 $D_k=\{(\boldsymbol{x}_1,k),(\boldsymbol{x}_2,k),\cdots,(\boldsymbol{x}_{m_k},k)\},k=0,1$。

1）定义投影前 d 维特征空间的统计量

分别定义各类特征的均值$\boldsymbol{\mu}_k$ 和离散度（类似于协方差）矩阵 $\boldsymbol{S}_k,k=0,1$，

$$\boldsymbol{\mu}_k=\frac{1}{m_k}\sum_{i=1}^{m_k}\boldsymbol{x}_i \tag{3-31}$$

$$\boldsymbol{S}_k=\sum_{i=1}^{m_k}(\boldsymbol{x}_i-\boldsymbol{\mu}_k)(\boldsymbol{x}_i-\boldsymbol{\mu}_k)^{\mathrm{T}} \tag{3-32}$$

其中，$\boldsymbol{x}_i\in D_k$。式(3-31)、式(3-32)的展开式为

$$\boldsymbol{\mu}_k=\begin{bmatrix}\mu_{k1}\\ \mu_{k2}\\ \vdots\\ \mu_{kd}\end{bmatrix},\quad \mu_{kl}=\frac{1}{m_k}\sum_{i=1}^{m_k}x_{il},\quad l=1,2,\cdots,d$$

$$\boldsymbol{S}_k=\begin{bmatrix}\sigma_{11} & \sigma_{12} & \cdots & \sigma_{1d}\\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2d}\\ \vdots & \vdots & \ddots & \vdots\\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_{dd}\end{bmatrix},\quad \sigma_{jl}=\sum_{i=1}^{m_k}(x_{ij}-\mu_{kj})(x_{il}-\mu_{kl}),\quad j,l=1,2,\cdots,d$$

再定义 d 维特征空间的**类内离散度矩阵**（within-class scatter matrix）$\boldsymbol{S}_w$ 和 d 维特征空间的**类间离散度矩阵**（between-class scatter matrix）$\boldsymbol{S}_b$ 为

$$\boldsymbol{S}_w=P(Y=0)\boldsymbol{S}_0+P(Y=1)\boldsymbol{S}_1=\frac{m_0}{m}\boldsymbol{S}_0+\frac{m_1}{m}\boldsymbol{S}_1 \tag{3-33}$$

$$\boldsymbol{S}_b=(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)^{\mathrm{T}} \tag{3-34}$$

式(3-34)的展开式为

$$(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)=\begin{bmatrix}\mu_{01}-\mu_{11}\\ \mu_{02}-\mu_{12}\\ \vdots\\ \mu_{0d}-\mu_{1d}\end{bmatrix}$$

$$\boldsymbol{S}_b=\begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1d} \\ b_{21} & b_{22} & \cdots & b_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ b_{d1} & b_{d2} & \cdots & b_{dd} \end{bmatrix},\quad b_{jl}=(\mu_{0j}-\mu_{1j})(\mu_{0l}-\mu_{1l}),\quad j,l=1,2,\cdots,d$$

式(3-33)中的 $\boldsymbol{S}_w$、式(3-34)中的 $\boldsymbol{S}_b$ 都是对称半正定矩阵,且 $\boldsymbol{S}_w$ 在 $m>d$ 时一般是可逆的。

2) 定义投影后一维投影空间的统计量

在特征向量 $\boldsymbol{x}=(x_1,x_2,\cdots,x_d)^{\mathrm{T}}$ 向 $\boldsymbol{\omega}=(\omega_1,\omega_2,\cdots,\omega_d)^{\mathrm{T}}$ 投影后的一维空间,定义各类特征投影点 z 的均值 $\tilde{\mu}_k$ 和离散度(类似于方差)$\widetilde{S}_k$,$k=0,1$,

$$\tilde{\mu}_k=\frac{1}{m_k}\sum_{i=1}^{m_k}z_i=\frac{1}{m_k}\sum_{i=1}^{m_k}\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i=\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\mu}_k \tag{3-35}$$

$$\widetilde{S}_k=\sum_{i=1}^{m_k}(z_i-\tilde{\mu}_k)^2=\sum_{i=1}^{m_k}(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i-\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\mu}_k)(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i-\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\mu}_k)^{\mathrm{T}}=\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{S}_k\boldsymbol{\omega} \tag{3-36}$$

式(3-35)、式(3-36)的展开式为

$$\tilde{\mu}_k=\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\mu}_k=(\omega_1,\omega_2,\cdots,\omega_d)\begin{bmatrix}\mu_{k1}\\ \mu_{k2}\\ \vdots\\ \mu_{kd}\end{bmatrix}$$

$$\widetilde{S}_k=\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{S}_k\boldsymbol{\omega}=(\omega_1,\omega_2,\cdots,\omega_d)\begin{bmatrix}\sigma_{11} & \sigma_{12} & \cdots & \sigma_{1d}\\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2d}\\ \vdots & \vdots & \ddots & \vdots\\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_{dd}\end{bmatrix}\begin{bmatrix}\omega_1\\ \omega_2\\ \vdots\\ \omega_d\end{bmatrix}$$

再定义一维投影空间的**类内方差**(within-class variance)$\widetilde{S}_w$ 和一维投影空间的**类间方差**(between-class variance)$\widetilde{S}_b$ 为

$$\begin{aligned}\widetilde{S}_w&=P(Y=0)\widetilde{S}_0+P(Y=1)\widetilde{S}_1=\frac{m_0}{m}\widetilde{S}_0+\frac{m_1}{m}\widetilde{S}_1\\&=\frac{m_0}{m}\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{S}_0\omega+\frac{m_1}{m}\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{S}_1\omega=\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{S}_w\boldsymbol{\omega}\end{aligned} \tag{3-37}$$

$$\begin{aligned}\widetilde{S}_b&=(\tilde{\mu}_0-\tilde{\mu}_1)^2=(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\mu}_0-\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\mu}_1)^2=(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\mu}_0-\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\mu}_1)(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\mu}_0-\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\mu}_1)^{\mathrm{T}}\\&=\boldsymbol{\omega}^{\mathrm{T}}(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\omega}=\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{S}_b\boldsymbol{\omega}\end{aligned} \tag{3-38}$$

式(3-37)中的 $\widetilde{S}_w$、式(3-38)中的 $\widetilde{S}_b$ 都是二次型的形式,且 $\widetilde{S}_w\geqslant 0$,$\widetilde{S}_b\geqslant 0$。

3) 构造准则函数

对于二分类问题,线性判别分析方法希望在选择投影方向 $\boldsymbol{\omega}$ 时,能够让样本特征投影点"类内方差最小,类间方差最大"。这句话的含义就是希望让一维投影空间的类内方差 $\widetilde{S}_w$ 最小,类间方差 $\widetilde{S}_b$ 最大。可以构造如下准则函数 $J(\boldsymbol{\omega})$:

$$J(\boldsymbol{\omega})=\frac{\widetilde{S}_b}{\widetilde{S}_w}=\frac{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{S}_b\boldsymbol{\omega}}{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{S}_w\boldsymbol{\omega}} \tag{3-39}$$

然后选择使准则函数 $J(\boldsymbol{\omega})$最大的$\boldsymbol{\omega}$ 作为最优投影方向。这样，选择最优投影方向问题被转换为最大化准则函数 $J(\boldsymbol{\omega})$的问题。注：式(3-39)的准则函数是 1936 年由 R. A. Fisher 提出的，因此被称作 Fisher 准则函数。

4) 最大化准则函数

式(3-39)是$\boldsymbol{\omega}$ 的两个二次型之比，因此对任意常数 c，$J(c\boldsymbol{\omega})=J(\boldsymbol{\omega})$，即准则函数 $J(\boldsymbol{\omega})$的最优参数$\boldsymbol{\omega}$ 不唯一。不失一般性，可以令分母$\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{S}_w\boldsymbol{\omega}=1$，以此作为对$\boldsymbol{\omega}$ 的约束条件，然后最大化分子$\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{S}_b\boldsymbol{\omega}$ 。这样式(3-39)的优化问题等价于

$$\max_{\boldsymbol{\omega}} \ \boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{S}_b\boldsymbol{\omega} \tag{3-40}$$

$$\text{s. t. } \ \boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{S}_w\boldsymbol{\omega}=1$$

使用拉格朗日乘子法，定义拉格朗日函数为

$$L(\boldsymbol{\omega},\lambda)=\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{S}_b\boldsymbol{\omega}-\lambda(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{S}_w\boldsymbol{\omega}-1)$$

其中，λ 为拉格朗日乘子。对$\boldsymbol{\omega}$ 求偏导数，并令其等于零，即

$$\frac{\partial L(\boldsymbol{\omega},\lambda)}{\partial\boldsymbol{\omega}}=2\boldsymbol{S}_b\boldsymbol{\omega}-2\lambda\boldsymbol{S}_w\boldsymbol{\omega}=0$$

其中，$\dfrac{\partial\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}}{\partial\boldsymbol{x}}=2\boldsymbol{A}\boldsymbol{x}$，解得

$$\boldsymbol{S}_w^{-1}\boldsymbol{S}_b\boldsymbol{\omega}=\lambda\boldsymbol{\omega} \tag{3-41}$$

将式(3-34)的 $\boldsymbol{S}_b$ 代入 $\boldsymbol{S}_b\boldsymbol{\omega}$ ，则

$$\boldsymbol{S}_b\boldsymbol{\omega}=(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\omega}$$

式中的$(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\omega}$ 是两个向量的点积，其结果为标量(记作 R)，这说明 $\boldsymbol{S}_b\boldsymbol{\omega}$ 是$(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)$方向上的一个向量。由式(3-41)可得

$$\lambda\boldsymbol{\omega}=\boldsymbol{S}_w^{-1}(\boldsymbol{S}_b\boldsymbol{\omega})=\boldsymbol{S}_w^{-1}(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)((\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{\omega})=\boldsymbol{S}_w^{-1}(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)R$$

整理可得

$$\boldsymbol{\omega}=\frac{R}{\lambda}\boldsymbol{S}_w^{-1}(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)$$

因为线性判别分析方法只希望找出最优的投影方向，即找出投影方向上的任意一个向量$\boldsymbol{\omega}$ 均可，所以可以忽略比例因子$\dfrac{R}{\lambda}$，最终求得最优投影向量$\boldsymbol{\omega}^*$ 为

$$\boldsymbol{\omega}^*=\boldsymbol{S}_w^{-1}(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1) \tag{3-42}$$

式(3-42)中的 $\boldsymbol{S}_w$、$\boldsymbol{\mu}_0$、$\boldsymbol{\mu}_1$ 是已知的，可通过训练集样本数据直接计算出来。在求矩阵 $\boldsymbol{S}_w$ 的逆矩阵时，可以使用奇异值分解方法来简化求解，即先对 $\boldsymbol{S}_w$ 做奇异值分解，则

$$\boldsymbol{S}_w=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}$$

其中 $\boldsymbol{U}$、$\boldsymbol{V}$ 是正交矩阵(即 $\boldsymbol{U}^{-1}=\boldsymbol{U}^{\mathrm{T}}$，$\boldsymbol{V}^{-1}=\boldsymbol{V}^{\mathrm{T}}$)，$\boldsymbol{\Sigma}$ 是对角阵(其对角元素是 $\boldsymbol{S}_w$ 的奇异值)；然后求出 $\boldsymbol{S}_w$ 的逆矩阵，则

$$\boldsymbol{S}_w^{-1}=\boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^{\mathrm{T}}$$

3. 线性判别分析分类器

对于二分类问题，给定训练集 $D_{\text{train}}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\}$，其中 $\boldsymbol{x}_i=(x_{i1},x_{i2},\cdots,x_{id})$是 d 维样本特征，$y_i\in\{0,1\}$是其对应的实际类别。线性判别分析方法使

用式(3-31)～式(3-33)分别计算出 $\boldsymbol{S}_w$、$\boldsymbol{\mu}_0$ 和 $\boldsymbol{\mu}_1$，再用式(3-42)即可求出最优投影向量 $\boldsymbol{\omega}^*$。

在确定了最优投影向量 $\boldsymbol{\omega}^*$ 之后，使用式(3-28)的标量投影将训练集 D_{train} 的 d 维特征向量 $\boldsymbol{x}_i$ 转换为一维特征标量 z_i，即

$$z_i = \frac{(\boldsymbol{\omega}^*)^{\mathrm{T}}\boldsymbol{x}_i}{\|\boldsymbol{\omega}^*\|}, \quad i = 1,2,\cdots,m$$

这样就能得到一维特征的训练集 $D'_{\text{train}} = \{(z_1, y_1),(z_2, y_2),\cdots,(z_m, y_m)\}$。

对于二分类问题，可以基于一维特征训练集 D'_{train} 直接设定阈值 T，例如 $T = \frac{1}{2}(\tilde{\mu}_0 + \tilde{\mu}_1)$，然后将分类决策规则定为：给定新样本 $\boldsymbol{x}$，如果

$$z = \frac{(\boldsymbol{\omega}^*)^{\mathrm{T}}\boldsymbol{x}}{\|\boldsymbol{\omega}^*\|} \leqslant T$$

则将新样本 $\boldsymbol{x}$ 的类别判定为 0 或 1，这就是一个最简单的线性判别分析分类器。也可以基于一维特征训练集 D'_{train} 设计其他形式的分类器，例如贝叶斯分类器或 k 近邻分类器。设计分类器，是由于一维特征模型比高维特征模型要简单得多。

目前，解决分类问题已经很少使用线性判别分析方法了，线性判别分析被更多地用于特征降维(详见 3.4.3 节)。

4. 使用 scikit-learn 库中的线性判别分析模型

scikit-learn 库将线性判别分析模型封装成一个类(类名为 **LinearDiscriminantAnalysis**)，并将其存放在 sklearn.discriminant_analysis 模块中。LinearDiscriminantAnalysis 类实现了线性判别分析分类器模型的学习算法 **fit**()、预测算法 **predict**() 和评价算法 **score**()。LinearDiscriminantAnalysis 类既可以用于分类，也可以用于特征降维。

图 3-14 给出了使用 LinearDiscriminantAnalysis 类建立并训练乳腺癌诊断模型的示例代码。其中，诊断模型(即分类模型)采用的是线性判别分析分类器；数据集使用的是 3.1.3 节标准化后的乳腺癌诊断数据集，其中的训练集(X1_train、Y_train)被用于训练模型；然后用训练好的模型对测试集 X1_test 前两个病例样本进行分类预测，将预测结果 predict 与

```
In [8]: from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
        lda = LinearDiscriminantAnalysis(solver='svd')
        lda.fit(X1_train, Y_train)

        Y1 = lda.predict(X1_test[:2])
        print(X1_test[:2]); print()
        print("predict:", Y1," malignant:", Y_test.values[:2])
        print("mean accuracy on train:", lda.score(X1_train, Y_train))
        print("mean accuracy on test:", lda.score(X1_test, Y_test))

[[ 2.57961809  1.78726935  2.53447284  2.88707955 -0.09040012  1.21022282
   1.33334652  1.92889603  0.35553387  0.04144935  2.35619316 -0.45967024
   2.16869979  2.54038166 -0.20433459  0.17615607  0.43357064  0.87007041
  -0.56208259 -0.28062613  3.05256427  1.43836286  2.94101757  3.62730675
   0.6896002   1.00723154  1.48632804  2.20319388  0.32718689  0.1565044
]
 [-0.70642616 -0.22331665 -0.69195555 -0.68937948  1.26957147 -0.05005056
  -0.22723639 -0.3628993  -0.03876801  0.3405638  -0.35793278  0.79857725
  -0.35199608 -0.43380945  0.49969396 -0.13291308 -0.08102636  0.35424367
  -0.59235221  0.01705767 -0.64800054  0.58343296 -0.64787811 -0.63088521
   1.59700315  0.07465072  0.07249786  0.10953674 -0.15329395  0.3892508
3]]

predict: [0 1]  malignant: [0 1]
mean accuracy on train: 0.9648351648351648
mean accuracy on test: 0.9649122807017544
```

图 3-14 使用 LinearDiscriminantAnalysis 类建立并训练乳腺癌诊断模型的示例代码

Y_test 里的真实诊断结果 malignant 进行比对，1 表示恶性，0 表示良性；最后使用函数 score 计算模型在训练集(X1_train、Y_train)和测试集(X1_test、Y_test)上的平均正确率。

3.2.3　决策树

如何区分流感与普通感冒？一般来说，普通感冒的症状较轻，多数患者仅出现咳嗽、喉咙痛、流鼻涕等上呼吸道症状，但发高烧的人却不多；而流感除上呼吸道症状以外，同时还会伴随高烧、头痛、明显肌肉酸痛等全身症状。

区分流感与普通感冒可以使用不同的特征项，其中的某些特征项对于区分流感与普通感冒比较有效(例如是否高烧或全身酸痛等)，某些特征项的区分效果不明显(例如咳嗽、流鼻涕等)。

1. 决策树分类模型

决策树(decision tree)分类模型是一种按照特征有效性，先主要特征，后次要特征，逐步递进，最终完成分类决策的模型。决策树模型的分类决策过程分多步完成，每步选用一个特征项，生成若干个分支，将其绘制成图形非常像一棵倒立的树，故被称作“决策树”。图 3-15 给出一个对苹果进行分类的决策树示意图，它根据口感、外形和底色三个特征项来区分苹果是红富士还是国光。

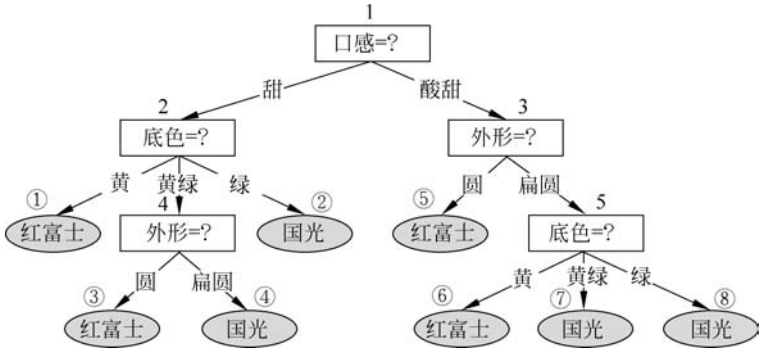


图 3-15　一个苹果分类的决策树示意图

一棵决策树包含一个**根结点**、若干**内部结点**和若干**叶子结点**。根结点和每个内部结点都分别代表一步决策(例如图 3-15 中的方形结点 1～5)，决策内容是根据某项特征生成若干分支，其中根结点(图 3-15 中的方形结点 1)是决策的起点；每个叶子结点代表决策的一个结论，即分类结果(例如图 3-15 中的椭圆形结点①～⑧)。

给定一个样本苹果，假设其特征为(酸甜，圆，黄绿)，根据图 3-15 的决策树分类模型，分类决策过程从根结点 1 开始，根据“口感＝酸甜”这项特征做出第一步决策，进入内部结点 3；再根据“外形＝圆”这项特征做出第二步决策，进入叶子结点⑤；进入叶子结点表示决策已得出结论，叶子结点⑤的结论是：样本苹果的品种为“红富士”。从根结点到叶子结点的路径对应决策过程的步骤序列，步骤的每一步可被描述成一个 if-then 规则，例如：

```
if (口感 = 酸甜)..... 第一步
then
```

```
        if (外形 = 圆)..... 第二步
        then
            品种 = 红富士 ..... 结论
        if (外形 = 扁圆)
        then
            …
    if (口感 = 甜)..... 第一步
    then
        …
```

整个决策树就是一个 if-then 规则集合,所描述的是根据历史观测提炼出来的某种一般性知识。基于 if-then 规则进行分类决策,非常类似于人们基于知识的**演绎推理**(deductive reasoning),即从一般性知识推及某个特定的个体(从一般推及个别)。但这些 if-then 规则是怎么来的呢?它们是通过对样本数据进行**归纳推理**(inductive reasoning,从个别推及一般)得来的。决策树的归纳推理过程就是基于训练集和学习算法来习得知识、建立决策树模型的过程。

2. 决策树学习算法

给定训练集 $D_{\text{train}}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\}$,其中 $\boldsymbol{x}_i=(x_{i1},x_{i2},\cdots,x_{id})$ 是 d 维样本特征,$y_i\in\{c_1,c_2,\cdots,c_n\}$ 是其对应的类别,$i=1,2,\cdots,m$。决策树学习算法就是要通过训练集来建立一个决策树分类模型。

决策树的学习过程从建立根结点开始,选择某项特征并根据其取值将训练集划分成不同子集,每个取值生成一个子集,然后为每个子集生成一个内部结点;剔除已使用过的特征项,再对所有子集重复"选择特征-划分子集"的过程,直到不可划分为止;将不可划分子集的结点设为叶子结点,并将其标记为某个类别。可以看出,决策树的学习过程是一个递归过程。

下面就以 3.1.1 节表 3-1 的 10 个苹果样本数据为训练集,具体讲解决策树的学习过程。苹果样本数据有 4 项特征,X=(底色,外形,口感,果重);类别有两个,即红富士和国光,它属于二分类问题。

首先建立根结点(记作 1 号结点),其对应的初始训练集包含 10 个样本数据(记作 D_1),对样本数据按顺序进行编号,$D_1=\{1,2,\cdots,10\}$。对 1 号结点进行"选择特征-划分子集"过程。假设选用特征"底色"来划分子集(参见表 3-8),该项特征有 3 个不同取值,分别为黄、黄绿和绿。

表 3-8 红富士和国光苹果样本(10 个)

编号	底色	外形	口感	果重/g	品种
1	黄	圆	甜	190	红富士
2	黄绿	扁圆	酸甜	260	红富士
3	绿	扁圆	酸甜	150	国光
4	黄绿	圆	甜	200	红富士
5	绿	扁圆	酸甜	210	国光
6	黄绿	扁圆	酸甜	170	国光
7	黄	圆	酸甜	200	红富士
8	黄绿	扁圆	酸甜	230	红富士
9	绿	扁圆	甜	180	国光
10	黄绿	扁圆	酸甜	240	红富士

按特征“底色”的取值，对 1 号结点数据集 D_1 进行划分，可划分出 3 个子集：

“底色＝黄”的子集 $D_2=\{1,7\}$；

“底色＝黄绿”的子集 $D_3=\{2,4,6,8,10\}$；

“底色＝绿”的子集 $D_4=\{3,5,9\}$。

为每个子集建立一个内部结点，分别记作 2 号结点、3 号结点、4 号结点，此时所建立的决策树形状如图 3-16(a)所示。

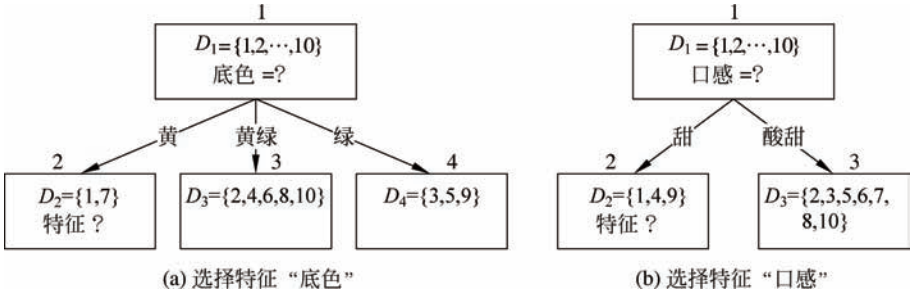


图 3-16　选择特征与划分子集的过程

如果改用特征“口感”来划分子集，该项特征有两个不同取值(甜、酸甜)，按“口感”的取值对 1 号结点数据集 D_1 进行划分，可划分出 2 个子集：

“口感＝甜”的子集 $D_2=\{1,4,9\}$；

“口感＝酸甜”的子集 $D_3=\{2,3,5,6,7,8,10\}$。

为每个子集建立一个内部结点，分别记作 2 号结点、3 号结点，此时所建立的决策树形状如图 3-16(b)所示。

可以看出，选用不同特征项会生成不同的决策树。决策树模型是按特征有效性(先主后次)分步建立决策树的。特征有效性指的是特征对分类是否有效。该如何度量特征有效性呢？

表 3-8 包含 10 个苹果样本数据，它是训练决策树模型的初始集合 D_1，其中既包含红富士苹果，也包含国光苹果。决策树模型依据某项特征将 D_1 划分成子集，希望每个子集尽可能属于同一类别，也就是将 D_1 划分成**纯度**(purity)较高的子集。机器学习借用信息论中的信息熵或统计学中的基尼指数来度量数据集的纯度，然后根据所划分子集的纯度来选择特征项。所划分子集的纯度越高，则特征项越有效。如何选择特征项、如何度量数据集的纯度，这是决策树分类模型最核心的内容。

3. 随机变量的信息熵与基尼指数

将数据集 D 中样本的类别 Y 看作一个离散型随机变量，其值域 $\Omega=\{c_1,c_2,\cdots,c_n\}$，共 n 个类别；再假设 D 中样本类别的概率分布为 $P(Y=c_k)=p_k$，$\sum_{k=1}^{n}p_k=1$，则

(1) 数据集 D 的**信息熵**(information entropy)被定义为

$$H(D)=-\sum_{k=1}^{n}p_k\mathrm{lb}p_k \tag{3-43}$$

其中，$0\leqslant H(D)\leqslant \mathrm{lb}n$。注：计算信息熵时若 $p_k=0$，则约定 $0\times\mathrm{lb}0=0$。

(2) 数据集 D 的**基尼指数**(Gini index)被定义为

$$\mathrm{Gini}(D)=\sum_{k=1}^{n}\sum_{j\neq k}p_k p_j=\sum_{k=1}^{n}p_k(1-p_k)=1-\sum_{k=1}^{n}p_k^2 \tag{3-44}$$

其中,$0\leqslant\mathrm{Gini}(D)\leqslant\dfrac{n-1}{n}$。注:基尼指数可理解为从数据集 D 中随机抽取两个样本点,其类别不一致的概率(它反映了数据集的混乱程度),或数据集 D 中样本点被错分概率的数学期望。

例如表 3-8 所示的初始数据集 D_1,其中有两个苹果类别{红富士,国光},属于 0-1 分布。由样本数据可以估计出苹果类别 Y_1 的概率分布为

$$P(Y_1=\text{红富士})=\frac{6}{10}=0.6,\quad P(Y_1=\text{国光})=\frac{4}{10}=0.4$$

分别计算 D_1 的信息熵和基尼指数,有

$$H(D_1)=-(0.6\times\mathrm{lb}0.6+0.4\times\mathrm{lb}0.4)=0.970\,95$$

$$\mathrm{Gini}(D_1)=1-(0.6^2+0.4^2)=0.48$$

假设将数据集 D_1 划分成子集 D_2、D_3,D_2 只包含红富士苹果,D_3 只包含国光苹果,即

$$P(Y_2=\text{红富士})=1,\quad P(Y_2=\text{国光})=0$$

$$P(Y_3=\text{红富士})=0,\quad P(Y_3=\text{国光})=1$$

则 D_2、D_3 都属于纯度最高的数据集,分别计算它们的信息熵和基尼指数,计算结果都将为零。

$$H(D_2)=-(1\times\mathrm{lb}1+0\times\mathrm{lb}0)=0,\quad \mathrm{Gini}(D_2)=1-(1^2+0^2)=0$$

$$H(D_3)=-(0\times\mathrm{lb}0+1\times\mathrm{lb}1)=0,\quad \mathrm{Gini}(D_3)=1-(0^2+1^2)=0$$

可以看出,数据集的纯度越高,则信息熵和基尼指数越小;纯度越低,则信息熵和基尼指数越大。如果数据集只包含一个类别,则纯度最高,其信息熵和基尼指数最小(都为 0);如果数据集包含全部类别(假设为 n)且服从均匀分布,则纯度最低,其信息熵和基尼指数最大,信息熵最大值为 $\mathrm{lb}n$,基尼指数最大值为$\dfrac{n-1}{n}$。借用信息熵或基尼指数,可以度量数据集的纯度。

4. 特征选择准则

给定数据集 $D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\}$,其中包含 m 个样本数据。假设

- 数据集 D 的样本特征为 d 维,记作 $X=\{X_1,X_2,\cdots,X_d\}$,其中 X_i 表示第 i 项特征;
- 所有特征项都是离散型的,第 i 项特征 X_i 有 V_i 个可能的取值。

则按第 i 项特征 X_i 的取值,可以将数据集 D 划分成 V_i 个子集$\{D_1,D_2,\cdots,D_{V_i}\}$。将每个子集包含的样本数据个数记作 $m_1,m_2,\cdots,m_{V_i}$,$m_1+m_2+\cdots+m_{V_i}=m$。将数据集 D 及其子集$\{D_1,D_2,\cdots,D_{V_i}\}$中样本的类别看作离散型随机变量,它们具有共有的值域 $\Omega=\{c_1,c_2,\cdots,c_n\}$,共 n 个类别。特征选择就是根据所划分子集的纯度来选择特征项,子集纯度越高,则该项特征越有效。

决策树学习算法的关键是如何定义纯度的度量形式,即特征选择准则。目前,常用的决策树学习算法有 **ID3**、**C4.5** 和 **CART**,它们分别提出了三种不同形式的特征选择准则。

1) ID3(信息增益)

假设按第 i 项特征 X_i 将数据集 D 划分成 V_i 个子集$\{D_1,D_2,\cdots,D_{V_i}\}$,ID3 算法首先按式(3-43)计算数据集 D 及其各子集 $D_1,D_2,\cdots,D_{V_i}$ 的信息熵,然后定义特征 X_i 对数据集 D 进行划分所获得的**信息增益**(information gain)为

$$\mathrm{Gain}(D,X_i)=H(D)-\sum_{k=1}^{V_i}\frac{m_k}{m}H(D_k),\quad i=1,2,\cdots,d \tag{3-45}$$

其中,等式右边的第二项可看作子集$\{D_1,D_2,\cdots,D_{V_i}\}$的平均信息熵。最终,将信息增益最大(即对纯度提升最大)的特征项选作对数据集 D 进行划分的最优特征,即

$$X_{\mathrm{optimal}}=\underset{i=1,2,\cdots,d}{\mathrm{argmax}}\,\mathrm{Gain}(D,X_i) \tag{3-46}$$

信息增益最大,这意味着子集的平均信息熵最小,也就是子集的平均纯度最大。基于信息增益选择特征,往往会偏向选择取值较多(即子集较多)的特征。为消除这种偏向,决策树学习算法引入了信息增益率的概念,基于信息增益率选择特征会更加客观。

2) C4.5(信息增益率)

假设按第 i 项特征 X_i 将数据集 D 划分成 V_i 个子集$\{D_1,D_2,\cdots,D_{V_i}\}$,C4.5 算法在式(3-45)信息增益的基础上,再定义特征 X_i 对数据集 D 进行划分所获得的**信息增益率**(information gain ratio)为

$$\mathrm{Gain\_ratio}(D,X_i)=\frac{\mathrm{Gain}(D,X_i)}{H_{X_i}(D)},\quad i=1,2,\cdots,d \tag{3-47}$$

其中,$H_{X_i}(D)$表示数据集 D 中特征 X_i 的信息熵(将特征 X_i 看作随机变量),其计算公式为

$$H_{X_i}(D)=-\sum_{k=1}^{V_i}\frac{m_k}{m}\mathrm{lb}\frac{m_k}{m},\quad i=1,2,\cdots,d \tag{3-48}$$

最终,将信息增益率最大的特征项选作对数据集 D 进行划分的最优特征,即

$$X_{\mathrm{optimal}}=\underset{i=1,2,\cdots,d}{\mathrm{argmax}}\,\mathrm{Gain\_ratio}(D,X_i) \tag{3-49}$$

3) CART(基尼指数)

假设按第 i 项特征 X_i 将数据集 D 划分成 V_i 个子集$\{D_1,D_2,\cdots,D_{V_i}\}$,CART 算法按式(3-44)计算各子集 $D_1,D_2,\cdots,D_{V_i}$ 的基尼指数,然后定义特征 X_i 对数据集 D 进行划分所得子集的**平均基尼指数**(average Gini index)为

$$\mathrm{Gini\_average}(D,X_i)=\sum_{k=1}^{V_i}\frac{m_k}{m}\mathrm{Gini}(D_k),\quad i=1,2,\cdots d \tag{3-50}$$

最终,将平均基尼指数最小(即子集平均纯度最大)的特征项选作对数据集 D 进行划分的最优特征,即

$$X_{\mathrm{optimal}}=\underset{i=1,2,\cdots,d}{\mathrm{argmin}}\,\mathrm{Gini\_average}(D,X_i) \tag{3-51}$$

5. 将不可划分子集设为叶子结点

决策树学习算法从建立根结点开始,依据某种准则(ID3、C4.5 或 CART)选择最优特征项,并按该特征项的取值将初始训练集划分成若干子集,为每个子集生成一个内部结点;剔除已使用过的特征项,再对所有子集重复"选择特征-划分子集"的过程,直到不可划分为止;将不可划分子集的结点设为叶子结点,并将其标记为某个类别。决策树学习算法是一个递归算法,停止递归的条件是子集不可划分。

什么样的子集是不可划分子集呢？不可划分子集有 4 种情况。

(1) 当前子集的样本属于同一类别(假设为 c_k)，无须划分。将当前子集的结点设为叶子结点，并将其类别标记为 c_k。

(2) 当前子集为空集，无法划分。将当前子集的结点设为叶子结点，并将该结点的类别标记为其父结点数据集中所含样本最多的类别。注：按照某个特征项的取值划分数据集，如果数据集没有覆盖该特征项的全部取值，则未被覆盖取值所划分出的子集为空集。

(3) 选择特征项划分当前子集，如果所有特征项都已使用过，没有任何新特征项可供选择，则无法继续划分。将当前子集的结点设为叶子结点，并将该结点的类别标记为当前子集所含样本最多的类别。注：对子集重复“选择特征-划分子集”的过程时，需要剔除之前已经使用过的特征项，只在剩下的新特征项中选择。

(4) 选择某个特征项划分当前子集，如果当前子集包含的所有样本点在该特征项下的取值都相同，则无法继续划分。将当前子集的结点设为叶子结点，并将该结点的类别标记为当前子集所含样本最多的类别。

总结一下，决策树模型就是将初始训练集逐步划分为纯度更高的子集，并在此过程中学习到分类规则，即决策树。今后，可以使用决策树对任意新样本进行分类。这里进一步对分类器模型做一下总结。给定特征 X，建立分类器模型有两种不同的思路：一是贝叶斯分类器，即基于概率模型并依最大后验概率 $P(Y|X)$来判定类别 Y，例如朴素贝叶斯分类器、逻辑斯蒂回归等，其中最值得借鉴的是如何基于严谨的概率模型与概率推理来求解问题；二是非贝叶斯分类器，即直接基于特征 X 来建立类别 Y 的判别函数，例如 k 近邻分类器、线性判别分析、决策树等，其中最值得借鉴的是如何提出创新思想并将其形式化为可被计算机执行的算法，也即将只可“意会”的思想转变为可以“言传”的算法。

6. 使用 scikit-learn 库中的决策树分类器模型

scikit-learn 库将决策树分类器模型封装成一个类(类名为 **DecisionTreeClassifier**)，并将其存放在 sklearn.tree 模块中。DecisionTreeClassifier 类实现了决策树分类器模型的学习算法 **fit**()、预测算法 **predict**()和评价算法 **score**()。

图 3-17 给出了使用 DecisionTreeClassifier 类建立并训练乳腺癌诊断模型的示例代码。

```
In [9]: from sklearn.tree import DecisionTreeClassifier
        dtc = DecisionTreeClassifier(criterion='entropy', random_state=2020)
        dtc.fit(X1_train, Y_train)

        Y1 = dtc.predict(X1_test[:2])
        print(X1_test[:2]); print()
        print("predict:", Y1," malignant:", Y_test.values[:2])
        print("mean accuracy on train:", dtc.score(X1_train, Y_train))
        print("mean accuracy on test:", dtc.score(X1_test, Y_test))
```

```
[[ 2.57961809  1.78726935  2.53447284  2.88707955 -0.09040012  1.21022282
   1.33334652  1.92889603  0.35553387  0.04144935  2.35619316 -0.45967024
   2.16869979  2.54038166 -0.20433459  0.17615607  0.43357064  0.87007041
  -0.56208259 -0.28062613  3.05256427  1.43836286  2.94101757  3.62730675
   0.6896002   1.00723154  1.48632804  2.20319388  0.32718689  0.1565044
]
 [-0.70642616 -0.22331665 -0.69195555 -0.68937948  1.26957147 -0.05005056
  -0.22723639 -0.3628993  -0.03876801  0.3405638  -0.35793278  0.79857725
  -0.35199608 -0.43380945  0.49969396 -0.13291308 -0.08102636  0.35424367
  -0.59235221  0.01705767 -0.64800054  0.58343296 -0.64787811 -0.63088521
   1.59700315  0.07465072  0.07249786  0.10953674 -0.15329395  0.3892508
3]]

predict: [0 1]  malignant: [0 1]
mean accuracy on train: 1.0
mean accuracy on test: 0.9385964912280702
```

图 3-17　使用 DecisionTreeClassifier 类建立并训练乳腺癌诊断模型的示例代码

其中诊断模型(即分类模型)采用的是决策树分类器；数据集使用的是 3.1.3 节标准化后的乳腺癌诊断数据集，其中的训练集(X1_train、Y_train)被用于训练模型；然后用训练好的模型对测试集 X1_test 前 2 个病例样本进行分类预测，将预测结果 predict 与 Y_test 中的真实诊断结果 malignant 进行比对，1 表示恶性，0 表示良性；最后使用函数 score()计算模型在训练集(X1_train、Y_train)和测试集(X1_test、Y_test)上的平均正确率。

3.3　多分类问题与分类模型评价

某些分类器既可以处理二分类，也可以处理多分类；而某些分类器只能处理二分类。本节首先讲解如何将多分类问题转换为二分类问题，然后使用二分类的方法来处理多分类。

机器学习在训练好模型之后，更关注模型对新样本的预测能力，即泛化能力。模型的泛化能力强，这才是机器学习意义上的好模型。为此，机器学习在训练好模型之后还需要再用测试集进行测试，评价模型的泛化性能。评价回归模型泛化能力的主要指标有均方误差 MSE 和决定系数 R 方(参见 2.3.3 节)，而评价分类模型泛化能力则会使用完全不同的指标。本节将介绍评价分类模型泛化能力的主要指标。

3.3.1　二分类与多分类

对“是/否”“真/假”“好/坏”等问题进行分类属于二分类问题。例如，根据临床特征判定乳腺癌是否恶性肿瘤就是一个二分类问题。在二分类问题中，类别的取值只有两个。通常将感兴趣的类别记作 1，并将其称作**正类**(positive class)；另外那个类别记作 0 或 −1，并将其称作**反类**(negative class)。例如乳腺癌诊断问题所关注的是恶性肿瘤，可以将恶性肿瘤看作正类，良性肿瘤看作反类。对于数据集中的样本数据，正类的样本数据被称作**正例**(positive example)，反类的样本数据被称作**反例**(negative example)。

实际应用中还有一些多分类问题，例如判断苹果品种是红富士、国光或黄元帅(三分类)，人脸识别和语音识别(N 分类)等。某些分类器既可以处理二分类，也可以处理多分类，例如朴素贝叶斯、k 近邻、决策树等。某些分类器本来只能处理二分类，但略加修改即可推广至多分类，例如逻辑斯谛回归、线性判别分析等。

处理多分类问题的另一种思路是将其转换为二分类问题。可以将多分类问题拆分为若干个二分类问题，为每个二分类问题设计一个分类器，然后汇总它们的二分类结果，最终得出多分类的结果。

假设给定训练集 $D_{\text{train}}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\}$，其中 $\boldsymbol{x}_i$ 是样本特征，$y_i\in\{c_1,c_2,\cdots,c_N\}$是其对应的实际类别，共 N 个类别。将这样的 N 分类问题拆分为二分类问题，常用的拆分策略有**一对一**(One vs One，OvO)或**一对其余**(One vs Rest，OvR)。

1. 一对一

按照一对一拆分策略，将 N 个类别两两组合，总共拆分出 $N(N-1)/2$ 个二分类问题。每个类别 c_k 与另外 $N-1$ 个类别按一对一方式，分别设计 $N-1$ 个分类器。给定新的样本特征，每个分类器会得到一个是或不是 c_k 的二分类结果；统计是 c_k 的结果个数(记作 n_k)，

然后将 n_k 最大的类别作为多分类结果。

2. 一对其余

按照一对其余拆分策略，每次将 N 个类别中的一个作为正类，其余 $N-1$ 个合起来作为反类，总共拆分出 N 个二分类问题。设计 N 个分类器，每个类别 c_k 对应一个以 c_k 为正类的分类器。给定新的样本特征，每个分类器会得到一个是否正类的二分类结果；如果只有一个分类器判定样本为正类，则将该分类器对应的正类 c_k 作为多分类结果；如果有多个分类器判定样本为正类，则将正类概率(或称 score)最大的那个 c_k 作为多分类结果。

3.3.2 分类模型的评价指标

分类模型在训练好之后还需要再用测试集进行测试，评价其泛化能力。注意，训练模型时会使用损失函数或准则函数来评价模型在训练集上的性能，并以此作为选择最优参数的标准。不同模型在训练时会使用不同的损失函数或准则函数，但评价泛化能力时则应使用统一的评价指标，这样才有可比性。

1. 二分类模型的评价指标

给定某个样例$(\boldsymbol{x},y)$，将分类器模型 f 的分类结果记作 $f(\boldsymbol{x})$。如果样例$(\boldsymbol{x},y)$的分类结果 $f(\boldsymbol{x})$与真实类别 y 一致，即 $f(\boldsymbol{x})=y$，则称样例$(\boldsymbol{x},y)$分类正确，否则就是分类错误。

式(3-52)定义了一个机器学习中常用的函数 $I(A)$，其中 A 表示一个事件。该函数被称作事件 A 的**指示函数**(indicator function)，或事件 A 的指示随机变量(服从 0-1 分布)。

$$I(A)=\begin{cases}1, & \text{如果事件 } A \text{ 发生了}\\ 0, & \text{如果事件 } A \text{ 未发生}\end{cases} \tag{3-52}$$

例如，

$$I(f(\boldsymbol{x})=y)=\begin{cases}1, & \text{如果 } f(\boldsymbol{x})=y\\ 0, & \text{如果 } f(\boldsymbol{x})\neq y\end{cases}$$

针对二分类问题，假设给定测试集 $D_{\text{test}}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\}$，其中 $\boldsymbol{x}_i$ 是样本特征，$y_i\in\{0,1\}$是其对应的真实类别。测试集 D_{test} 共有 m 个测试样例，假设其中正例有 m^+个，反例有 m^-个，$m^++m^-=m$。下面给出评价二分类模型泛化能力的主要指标。

1) 正确率

正确率(accuracy)就是分类器模型 f 对测试集 D_{test} 分类正确的样例数占总样例数的比例，即

$$\text{accuracy}=\frac{\sum_{i=1}^{m}I(f(\boldsymbol{x}_i)=y_i)}{m} \tag{3-53}$$

在正类、反类分布不平衡的情况下，正确率并不能客观反映分类器模型的性能。例如，乳腺癌分类器模型可以将恶性肿瘤看作正类，良性肿瘤看作反类。假设良性肿瘤占全部肿瘤的 99%，恶性肿瘤只占 1%。设计一个最简单的分类器，给定任意肿瘤样本，将分类结果都固定判定为良性肿瘤，则其分类正确率可高达 99%。这显然是不合理的，或者说不符合

人们对分类器模型的评价需求。

2）混淆矩阵

在二分类问题中，被分类器判定为正类的样例，如果其真实类别确实是正类，则称为**真正例**（True Positive，TP）；如果其真实类别是反类，则称为**假正例**（False Positive，FP）。同理，被分类器判定为反类的样例，如果其真实类别确实是反类，则称为**真反例**（True Negative，TN）；如果其真实类别是正类，则称为**假反例**（False Negative，FN）。

使用测试集对分类器模型进行测试，二分类结果可以表示成**混淆矩阵**（confusion matrix）的形式，其中包括 TP、FP、TN、FN 这 4 种情形的样例数（参见表 3-9）。

表 3-9　二分类结果的混淆矩阵

真实类别	分类结果	
	正例	反例
正例	TP（真正例）	FN（假反例）
反例	FP（假正例）	TN（真反例）

假设测试集有 m 个测试样例，其中正例有 m^+ 个，反例有 m^- 个，则

$$\mathrm{TP}+\mathrm{FN}=m^+;\quad \mathrm{TN}+\mathrm{FP}=m^-;\quad \mathrm{TP}+\mathrm{FN}+\mathrm{TN}+\mathrm{FP}=m$$

3）精确率与召回率

在乳腺癌分类器模型中，人们可能会更关注恶性肿瘤（正类）的分类性能，这时可以使用正类的精确率或召回率来评价分类模型。

精确率（precision，记作 P）就是在被分类器判定为正类的样例（TP＋FP）中，有多大比例属于真正例（TP）；**召回率**（recall，记作 R）就是在数据集的全部正例（TP＋FN）中，有多大比例能被分类器判定为正例（即真正例 TP）。基于混淆矩阵可以计算出精确率 P 和召回率 R。

$$P=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{3-54}$$

$$R=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{3-55}$$

例如乳腺癌分类器模型将恶性肿瘤看作正类，良性肿瘤看作反类，则精确率 P 表示被分类器判定为恶性肿瘤的样例中，有多大比例属于真的恶性肿瘤；召回率 R 表示数据集的全部恶性肿瘤样例中，有多大比例能被分类器找出来（即确实被判定为恶性肿瘤）。

设计分类器时，精确率和召回率通常很难同时兼顾到。精确率高时，召回率往往偏低；召回率高时，精确率就会偏低。例如，设计乳腺癌分类器模型时，严格恶性肿瘤的诊断标准可以提高精确性，但会造成某些恶性肿瘤的漏诊（召回率降低）；放松诊断标准虽能提高召回率，但会将某些良性肿瘤误诊为恶性肿瘤（精确率降低）。不同分类问题对精确率、召回率的关注程度不同，某些问题关注精确率，另外一些问题可能更关注召回率。

可以将精确率和召回率结合起来，构造一个综合性评价指标。F1 值就是这样一种比较常用的综合性指标，其定义形式为

$$\frac{1}{\mathrm{F1}}=\frac{1}{2}\cdot\left(\frac{1}{P}+\frac{1}{R}\right)$$

整理可得，

$$F1 = \frac{2 \times P \times R}{P + R} \tag{3-56}$$

2. 使用 scikit-learn 库中的度量函数

scikit-learn 库为分类模型评价指标提供了相应的计算函数，它们被统一存放在 sklearn.metrics 模块中。

accuracy_score()：正确率计算函数；

precision_score()：精确率计算函数；

recall_score()：召回率计算函数；

f1_score()：F1 值计算函数。

图 3-18 给出了计算乳腺癌诊断模型评价指标的示例代码。其中图 3-18(a)使用的诊断模型是决策树分类器，图 3-18(b)使用的则是逻辑斯谛回归分类器；数据集使用的都是 3.1.3 节标准化后的乳腺癌诊断数据集，训练集(X1_train、Y_train)被用于训练模型；然后用测试集(X1_test、Y_test)对模型进行测试，分别计算并显示模型在测试集上的正确率、精确率、召回率和 F1 值。

```
In [10]: from sklearn.metrics import accuracy_score
         from sklearn.metrics import precision_score
         from sklearn.metrics import recall_score
         from sklearn.metrics import f1_score

         from sklearn.tree import DecisionTreeClassifier
         dtc = DecisionTreeClassifier(criterion='entropy', random_state=2020)
         dtc.fit(X1_train, Y_train)
         Y_pred = dtc.predict(X1_test)

         print("accuracy_score=", accuracy_score(Y_test, Y_pred))
         print("precision_score=", precision_score(Y_test, Y_pred, average='binary'))
         print("recall_score=", recall_score(Y_test, Y_pred, average='binary'))
         print("f1_score=", f1_score(Y_test, Y_pred, average='binary'))

         accuracy_score= 0.9385964912280702
         precision_score= 0.9538461538461539
         recall_score= 0.9393939393939394
         f1_score= 0.9465648854961831
```

(a) 决策树分类器

```
In [11]: from sklearn.metrics import accuracy_score
         from sklearn.metrics import precision_score
         from sklearn.metrics import recall_score
         from sklearn.metrics import f1_score

         from sklearn.linear_model import LogisticRegression
         lr = LogisticRegression(penalty='l2', C=1.0, random_state=2020)
         lr.fit(X1_train, Y_train)
         Y_pred = lr.predict(X1_test)

         print("accuracy_score=", accuracy_score(Y_test, Y_pred))
         print("precision_score=", precision_score(Y_test, Y_pred, average='binary'))
         print("recall_score=", recall_score(Y_test, Y_pred, average='binary'))
         print("f1_score=", f1_score(Y_test, Y_pred, average='binary'))

         accuracy_score= 0.9736842105263158
         precision_score= 0.9846153846153847
         recall_score= 0.9696969696969697
         f1_score= 0.9770992366412214
```

(b) 逻辑斯谛回归分类器

图 3-18　计算乳腺癌诊断模型评价指标的示例代码

3. 多分类模型的评价指标

多分类问题可以看作由多个二分类问题组成，测试结果可以表示成多个二分类混淆矩阵。对这些二分类混淆矩阵求平均，将所得到的平均指标作为多分类模型的评价指标。

第一种求平均的方法：先计算各混淆矩阵的精确率、召回率和 F1 值，然后对计算结果分别求平均，所得到的平均值被称为**宏精确率**(macro-P)、**宏召回率**(macro-R)和**宏 F1 值**(macro-F1)。

第二种求平均的方法：先对混淆矩阵的 TP、FP、TN、FN 样例数求平均，得到一个平均混淆矩阵，基于这个平均混淆矩阵所求出的精确率、召回率和 F1 值被称为**微精确率**(micro-P)、**微召回率**(micro-R)和**微 F1 值**(micro-F1)。

3.3.3　P-R 曲线与 ROC 曲线

本节再介绍两种评价分类模型的可视化曲线，即 P-R 曲线与 ROC 曲线。针对二分类问题，一个理想分类器应当将正例样本判定为正类，将反例样本判定给反例。给定样本特征，分类器会生成一个属于正类(或反类)的后验概率，或一个模拟的后验概率值。使用测试集对分类器模型进行测试，按后验概率(或模拟后验概率)对测试样本进行降序排序，如图 3-19 所示。

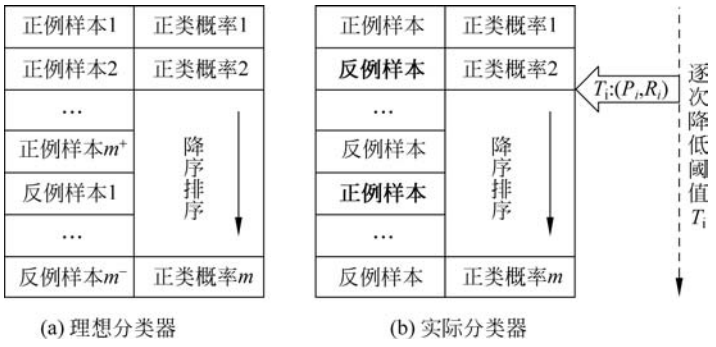


图 3-19　按后验概率(或模拟后验概率)排序后的测试样本

如果是理想分类器，则所有正例样本属于正类的后验概率都应该比反例样本高，会排在反例样本的前面(见图 3-19(a))。但实际的分类器很难做到这一点，某些反例样本属于正类的后验概率比正例样本高，反而排在正例样本的前面(见图 3-19(b))。正例样本和反例样本的排列次序能在更深层次上反映分类器性能的优劣。

1. P-R 曲线

为后验概率(或模拟后验概率)设定一个**阈值**(threshold)T，将高于或等于阈值的样本判为正类，低于阈值的判为反类，这样可以模拟一次分类决策。按从高到低的顺序逐次降低阈值 T_i，模拟出一组分类决策，分别计算每次分类结果的精确率和召回率，得到一组(P_i，R_i)数据。将这组(P_i，R_i)数据绘制成图形，以精确率 P 为纵轴，召回率 R 为横轴，就得到一条“精确率-召回率”曲线，它被称作 P-R 曲线(P-R curve)。

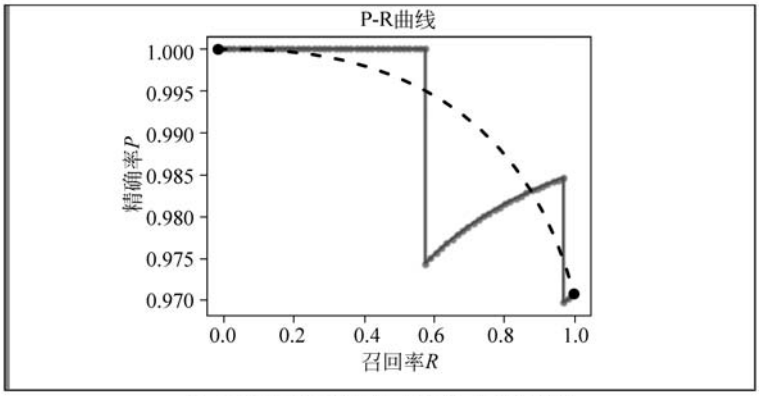
scikit-learn 库为计算 P-R 曲线提供了一个函数 **precision_recall_curve**()，它可以自动设置阈值，并计算出一组(P_i，R_i)数据，该函数被存放在 sklearn.metrics 模块中。

图 3-20 给出了计算并绘制乳腺癌诊断模型 P-R 曲线的示例代码(见图 3-20(a))及其运行结果(见图 3-20(b))。其中的诊断模型是逻辑斯谛回归分类器，可以调用其 **predict_proba**()函数获取测试样本的后验概率；数据集使用的是 3.1.3 节标准化后的乳腺癌诊断数据集，训练集(X1_train、Y_train)被用于训练模型；然后用测试集(X1_test、Y_test)对模

型进行测试,计算并显示测试结果的 P-R 曲线。

```
In [12]: from sklearn.linear_model import LogisticRegression
         lr = LogisticRegression(penalty='l2', C=1.0, random_state=2020)
         lr.fit(X1_train, Y_train)
         #Y_pred = lr.predict(X1_test)
         Y_pred_proba = lr.predict_proba(X1_test)[:, 1]

         from sklearn.metrics import precision_recall_curve
         precision, recall, thresholds = \
             precision_recall_curve(Y_test, Y_pred_proba, pos_label=1)
         print(recall.shape, precision.shape, thresholds.shape)

         fig = plt.figure( figsize=(4, 3), dpi=100 )
         plt.rcParams['font.sans-serif'] = ['SimHei']
         plt.rcParams['axes.unicode_minus'] = False

         plt.plot(recall, precision)
         plt.scatter(recall, precision, s=10, alpha=0.5)
         plt.title("P-R曲线")
         plt.xlabel("召回率 - R"); plt.ylabel("精确率 - P")
         plt.show()

         (69,) (69,) (68,)
```

(a) 示例代码



(b) 示例代码所绘制的P-R曲线(实线部分)

图 3-20 计算并绘制乳腺癌诊断模型 P-R 曲线的示例代码和运行结果

因为乳腺癌数据集比较小,正例(恶性)样本更少,因此图 3-20(b)所绘制的 P-R 曲线(实线部分)很不平滑。在样本数据足够多的情况下,P-R 曲线应该像图 3-20(b)中虚线部分那样比较平滑。从 P-R 曲线可以更直观地看到,精确率高时,召回率往往偏低;召回率高时,精确率就会偏低。P-R 曲线越偏向右上角,说明分类器的性能越好。

2. ROC 曲线

将 P-R 曲线的纵轴由精确率 P 改为**真正例率**(True Positive Rate,TPR),横轴由召回率 R 改为**假正例率**(False Positive Rate,FPR),这样所得到的曲线被称为 ROC(Receiver Operating Characteristic)曲线。真正例率 TPR、假正例率 FPR 的定义分别为

$$\mathrm{TPR}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{3-57}$$

$$\mathrm{FPR}=\frac{\mathrm{FP}}{\mathrm{TN}+\mathrm{FP}} \tag{3-58}$$

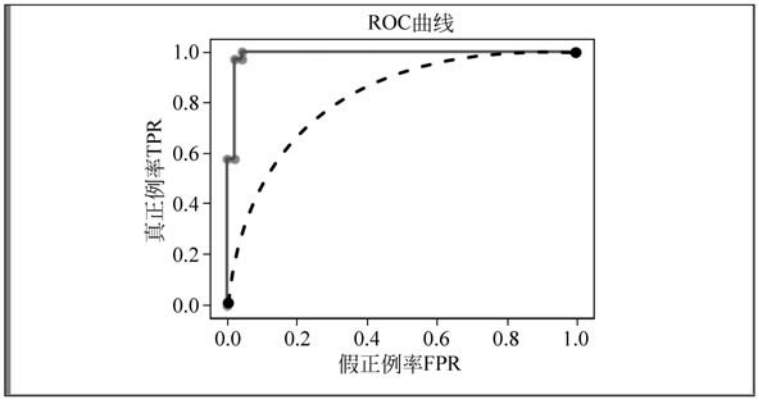
ROC 曲线的计算与绘制过程与 P-R 曲线类似。使用测试集对分类器模型进行测试,按

从高到低的次序为测试样本后验概率设定一组阈值 T_i，模拟出一组分类决策，分别计算每次分类结果的真正例率和假正例率，得到一组(TPR_i，FPR_i)数据。将这组(TPR_i，FPR_i)数据绘制成图形，以真正例率 TPR 为纵轴，假正例率 FPR 为横轴，就得到一条 ROC 曲线。

scikit-learn 库为 ROC 曲线提供了一个函数 **roc_curve**()，它可以自动设置阈值，并计算出一组(TPR_i，FPR_i)数据，该函数被存放在 sklearn. metrics 模块中。图 3-21 给出了计算并绘制乳腺癌诊断模型 ROC 曲线的示例代码(见图 3-21(a))及其运行结果(见图 3-21(b))。

```
In [13]: from sklearn.linear_model import LogisticRegression
         lr = LogisticRegression(penalty='l2', C=1.0, random_state=2020)
         lr.fit(X1_train, Y_train)
         #Y_pred = lr.predict(X1_test)
         Y_pred_proba = lr.predict_proba(X1_test)[:, 1]

         from sklearn.metrics import roc_curve
         fpr, tpr, thresholds = \
             roc_curve(Y_test, Y_pred_proba, pos_label=1)
         print(fpr.shape, tpr.shape, thresholds.shape)

         fig = plt.figure( figsize=(4, 3), dpi=100 )
         plt.rcParams['font.sans-serif'] = ['SimHei']
         plt.rcParams['axes.unicode_minus'] = False

         plt.plot(fpr, tpr)
         plt.scatter(fpr, tpr, s=20, alpha=0.5)
         plt.title("ROC曲线")
         plt.xlabel("假正例率 - FPR"); plt.ylabel("真正例率 - TPR")
         plt.show()

         (8,) (8,) (8,)
```

(a) 示例代码



(b) 示例代码所绘制的ROC曲线（实线部分）

图 3-21　计算并绘制乳腺癌诊断模型 ROC 曲线的示例代码和运行结果

同样，因为乳腺癌数据集比较小，所以图 3-21(b)所绘制的 ROC 曲线(实线部分)不太平滑。在样本数据足够多的情况下，ROC 曲线应该像图 3-21(b)中虚线部分那样，比较平滑。ROC 曲线越偏向左上角，说明分类器的性能越好。

3.4　特征降维

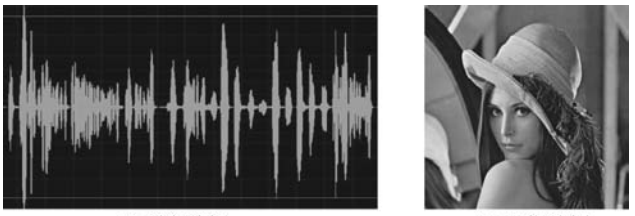
给定训练集 $D_{\text{train}}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(x_m,y_m)\}$，其中 $\boldsymbol{x}_i$ 是 d 维样本特征，$y_i\in\{c_1,c_2,\cdots,c_n\}$是其对应的实际类别，$i=1,2,\cdots,m$。可以将样本特征 $\boldsymbol{x}_i$ 看作特征空

间里的向量。在特征空间中,每个特征项是一个维度并对应一个标准基向量,d 个特征项就构成一个 d 维特征空间,其标准基向量可记作$\{\boldsymbol{e}_1,\boldsymbol{e}_2,\cdots,\boldsymbol{e}_d\}$。每个样本数据 $\boldsymbol{x}_i$ 对应特征空间里的一个点,其坐标$\{x_{i1},x_{i2},\cdots,x_{id}\}$就是样本在各特征项上的取值。

例如表 3-1 的苹果样本数据,其样本特征包含底色、外形、口感和果重 4 个特征项,以这 4 个特征项为坐标轴就构成一个四维特征空间。表 3-1 的苹果样本特征是经过人的加工整理,然后提炼出来的,这个加工、提炼过程被称为**特征提取**(feature extraction)。

经过特征提取的数据比较精炼、规范,这样的数据被称为**结构化**(structured)数据。实际应用中还有大量**非结构化**(unstructured)数据,例如文本数据、音频数据、图像数据、视频数据等。提取非结构化数据的特征需专门领域的知识,例如自然语言处理、语音信号处理、数字图像处理等。

可以先对非结构化的原始数据进行特征提取,然后基于所提取的特征数据进行机器学习。也可以直接以非结构化原始数据为特征进行机器学习,其优点是不会因特征提取而丢失任何信息,缺点是维数太多。维数很多的数据被称为**高维**(high-dimension)数据。例如,在 22kHz 采样率情况下,一段 10ms 语音采样数据(见图 3-22(a))大约有 $22\times10=220$(个),其维数是 220 维;一幅 $64\times64=4096$(像素)灰度图像(见图 3-22(b))的维数是 4096 维,而彩色图像的维数则是 $4096\times3=12\,288$ 维。



(a) 语音数据　　(b) 图像数据

图 3-22　高维数据示例

在机器学习中,特征维数过高很容易导致距离计算困难、对样本需求量迅速加大(否则难以体现联合概率分布)等问题,这种现象被称为**维数灾难**(curse of dimensionality)。**降维**(dimension reduction)是缓解维数灾难的重要途径,其中会用到线性代数相关的知识,例如坐标变换及其矩阵表示、矩阵的特征值分解与奇异值分解等。

3.4.1　线性代数基础

1. 坐标变换及其矩阵表示

一个二维特征空间(见图 3-23),其两个标准基$\{\boldsymbol{e}_1,\boldsymbol{e}_2\}$分别对应样本特征的两个特征项。可以将样本特征数据 $\boldsymbol{x}$ 看作特征空间中的一个点,其坐标(x_1,x_2)为样本在特征项 1、特征项 2 上的取值;$\boldsymbol{x}$ 也被称作特征空间中的一个特征向量,记作 $\boldsymbol{x}=(x_1,x_2)^{\mathrm{T}}$。

假设将特征向量 $\boldsymbol{x}$ 在 $\boldsymbol{e}_1$、$\boldsymbol{e}_2$ 方向上的坐标分别拉伸 λ_1、λ_2 倍(见图 3-23(a)),则**拉伸**(或称缩放)**变换**后新向量 $\boldsymbol{z}$ 的坐标为

$$\boldsymbol{z}=\begin{bmatrix}z_1\\z_2\end{bmatrix}=\begin{bmatrix}\lambda_1x_1\\\lambda_2x_2\end{bmatrix}=\begin{bmatrix}\lambda_1&0\\0&\lambda_2\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}=\begin{bmatrix}\lambda_1&0\\0&\lambda_2\end{bmatrix}\boldsymbol{x}$$

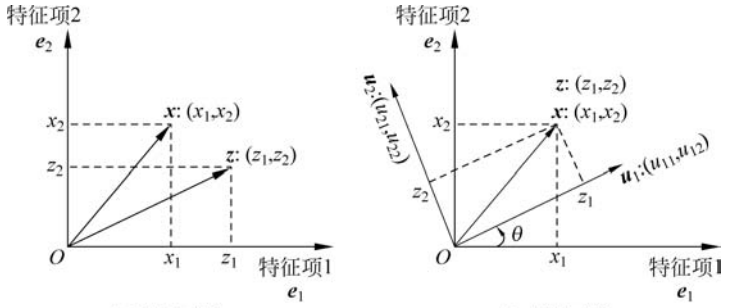


图 3-23　二维特征空间的坐标变换

记 $\boldsymbol{D}=\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{bmatrix}$，则 $\boldsymbol{z}=\boldsymbol{D}\boldsymbol{x}$。一般地，$d$ 维特征空间的拉伸变换可以用一个 $d\times d$ 对角矩阵 $\boldsymbol{D}$ 表示，其对角元素就是在各坐标轴方向的拉伸倍数。换句话说，对角矩阵对应的坐标变换为拉伸变换。

假设将两个标准基$\{\boldsymbol{e}_1,\boldsymbol{e}_2\}$旋转某个角度 θ（见图 3-23(b)），得到一组新的基$\{\boldsymbol{u}_1,\boldsymbol{u}_2\}$：$\boldsymbol{u}_1=\begin{bmatrix}u_{11}\\ u_{12}\end{bmatrix}$，$\boldsymbol{u}_2=\begin{bmatrix}u_{21}\\ u_{22}\end{bmatrix}$，或写成 $\boldsymbol{u}_1=u_{11}\boldsymbol{e}_1+u_{12}\boldsymbol{e}_2$，　$\boldsymbol{u}_2=u_{21}\boldsymbol{e}_1+u_{22}\boldsymbol{e}_2$。旋转变换后，向量 $\boldsymbol{x}=(x_1,x_2)^{\mathrm{T}}$ 在基$\{\boldsymbol{u}_1,\boldsymbol{u}_2\}$下的坐标 $\boldsymbol{z}$ 为

$$\boldsymbol{z}=\begin{bmatrix}z_1\\ z_2\end{bmatrix}=\begin{bmatrix}u_{11} & u_{12}\\ u_{21} & u_{22}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}=\begin{bmatrix}\boldsymbol{u}_1^{\mathrm{T}}\\ \boldsymbol{u}_2^{\mathrm{T}}\end{bmatrix}\boldsymbol{x} \tag{3-59}$$

记 $\boldsymbol{U}=\begin{bmatrix}\boldsymbol{u}_1^{\mathrm{T}}\\ \boldsymbol{u}_2^{\mathrm{T}}\end{bmatrix}$，则 $\boldsymbol{z}=\boldsymbol{U}\boldsymbol{x}$。坐标 $\boldsymbol{z}$ 实际上就是向量 $\boldsymbol{x}$ 在新基向量$\{\boldsymbol{u}_1,\boldsymbol{u}_2\}$上的标量投影（被称作**投影变换**），旋转变换就是一种投影变换。一般地，d 维特征空间从一组基，例如标准基$\{\boldsymbol{e}_1,\boldsymbol{e}_2,\cdots,\boldsymbol{e}_d\}$，到另一组基$\{\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_d\}$的投影变换可以用一个 $d\times d$ 矩阵 $\boldsymbol{U}$ 表示，其行向量分别对应新的基向量。将向量 $\boldsymbol{x}=(x_1,x_2,\cdots,x_d)^{\mathrm{T}}$ 在基$\{\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_d\}$下的坐标记作 $\boldsymbol{z}=(z_1,z_2,\cdots,z_d)^{\mathrm{T}}$，则

$$\boldsymbol{z}=\boldsymbol{U}\boldsymbol{x},\quad \text{其中}\ \boldsymbol{U}_{d\times d}=\begin{bmatrix}\boldsymbol{u}_1^{\mathrm{T}}\\ \boldsymbol{u}_2^{\mathrm{T}}\\ \vdots\\ \boldsymbol{u}_d^{\mathrm{T}}\end{bmatrix} \tag{3-60}$$

矩阵 $\boldsymbol{U}$ 称作从基$\{\boldsymbol{e}_1,\boldsymbol{e}_2,\cdots,\boldsymbol{e}_d\}$到基$\{\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_d\}$的**转移矩阵**（transition matrix）。如果新的基向量$\{\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_d\}$是规范正交基（由相互正交单位向量构成的基），即

$$\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{u}_j=\begin{cases}1, & i=j\\ 0, & i\neq j\end{cases},\quad i,j=1,2,\cdots,d$$

则转移矩阵 $\boldsymbol{U}_{d\times d}$ 为**正交矩阵**（orthogonal matrix），即 $\boldsymbol{U}^{-1}=\boldsymbol{U}^{\mathrm{T}}$，或 $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}=\boldsymbol{I}$。任意一组基（可能不正交或不规范）到规范正交基的投影变换，其转移矩阵均为正交矩阵。

对样本数据进行坐标变换，实际上是对样本做特征变换。变换前，坐标 $\boldsymbol{x}=(x_1,x_2,\cdots,x_d)^{\mathrm{T}}$ 是样本在原特征项上的取值；变换后，坐标 $\boldsymbol{z}=(z_1,z_2,\cdots,z_d)^{\mathrm{T}}$ 是样本在新特征项上的取值，因此坐标变换就是将原特征 $\boldsymbol{x}$ 变换成新特征 $\boldsymbol{z}$。

式(3-59)、式(3-60)坐标变换所得到的新特征 $\boldsymbol{z}$ 是原特征 $\boldsymbol{x}$ 的线性组合，即线性变换。线性变换可表示成矩阵形式，拉伸变换和投影变换都属于线性变换。线性变换可以利用线性代数相关的理论、方法进行处理。实际应用中还会存在非线性的情况，某些非线性变换可以通过**核函数**(kernel function)转化为线性变换。

2. 特征值、特征向量与特征值分解

定义 3-1 一个 $n\times n$ 矩阵 $\boldsymbol{A}$，如果存在一个非零向量 $\boldsymbol{x}$ 使得 $\boldsymbol{Ax}=\lambda\boldsymbol{x}$，则称标量 λ 为矩阵 $\boldsymbol{A}$ 的**特征值**(eigenvalue)，向量 $\boldsymbol{x}$ 为矩阵 $\boldsymbol{A}$ 的属于特征值 λ 的特征**向量**(eigenvector)。

定理 3-1 一个 $n\times n$ 矩阵 $\boldsymbol{A}$ 是可对角化的，当且仅当 $\boldsymbol{A}$ 有 n 个线性无关的特征向量。

证明：若 $\boldsymbol{A}$ 有 n 个线性无关特征向量 $\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n$，其对应的特征值分别为 $\lambda_1,\lambda_2,\cdots,\lambda_n$，则以这 n 个特征向量为列向量的矩阵 $\boldsymbol{X}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n)$ 可逆并能将 $\boldsymbol{A}$ 对角化，即

$$\boldsymbol{AX}=\boldsymbol{A}(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n)=(\boldsymbol{Ax}_1,\boldsymbol{Ax}_2,\cdots,\boldsymbol{Ax}_n)=(\lambda_1\boldsymbol{x}_1,\lambda_2\boldsymbol{x}_2,\cdots,\lambda_n\boldsymbol{x}_n),$$

$$\boldsymbol{AX}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n)\begin{bmatrix}\lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n\end{bmatrix}=\boldsymbol{XD} \tag{3-61}$$

因为 $\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n$ 线性无关，所以 $\boldsymbol{X}$ 可逆，因而

$$\boldsymbol{X}^{-1}\boldsymbol{AX}=\boldsymbol{D},\quad \text{或 } \boldsymbol{A}=\boldsymbol{XDX}^{-1} \tag{3-62}$$

其中的 $\boldsymbol{X}$、$\boldsymbol{D}$ 不唯一，因为将各列重排或乘以一个非零系数，新的 $\boldsymbol{X}$ 及其对应的 $\boldsymbol{D}$ 仍能将 $\boldsymbol{A}$ 对角化。矩阵 $\boldsymbol{A}$ 可对角化也可以说成是矩阵 $\boldsymbol{A}$ 可按特征值分解为 $\boldsymbol{XDX}^{-1}$。

定理 3-2(谱定理) 若 $\boldsymbol{A}$ 是 $n\times n$ 实对称矩阵，则存在一个正交矩阵 $\boldsymbol{U}$ 可以对角化 $\boldsymbol{A}$，即 $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{AU}=\boldsymbol{D}$，其中 $\boldsymbol{U}$ 的列向量是 $\boldsymbol{A}$ 的 n 个线性无关特征向量(单位向量)，$\boldsymbol{D}$ 是对角矩阵(对角元素是特征向量对应的特征值)。或者说，实对称矩阵 $\boldsymbol{A}$ 可分解为一个正交矩阵 $\boldsymbol{U}$ 和一个对角矩阵 $\boldsymbol{D}$，即

$$\boldsymbol{A}=\boldsymbol{UDU}^{\mathrm{T}}=(\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_n)\begin{bmatrix}\lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n\end{bmatrix}\begin{bmatrix}\boldsymbol{u}_1^{\mathrm{T}}\\ \boldsymbol{u}_2^{\mathrm{T}}\\ \vdots\\ \boldsymbol{u}_n^{\mathrm{T}}\end{bmatrix}$$

$$=\lambda_1\boldsymbol{u}_1\boldsymbol{u}_1^{\mathrm{T}}+\lambda_2\boldsymbol{u}_2\boldsymbol{u}_2^{\mathrm{T}}+\cdots+\lambda_n\boldsymbol{u}_n\boldsymbol{u}_n^{\mathrm{T}}$$

定理 3-2 的意义在于，如果将 $n\times n$ 实对称矩阵 $\boldsymbol{A}$ 看作一个坐标变换，则该变换可分解成两次到规范正交基的投影变换和一次拉伸变换，其中投影变换的基对应矩阵 $\boldsymbol{A}$ 的各特征向量，拉伸变换的倍数对应矩阵 $\boldsymbol{A}$ 的各特征值。或者说，实对称矩阵 $\boldsymbol{A}$ 可分解成一组分量(即 $\boldsymbol{u}_1\boldsymbol{u}_1^{\mathrm{T}}$、$\boldsymbol{u}_2\boldsymbol{u}_2^{\mathrm{T}}$、$\cdots$、$\boldsymbol{u}_n\boldsymbol{u}_n^{\mathrm{T}}$)的线性组合，各分量的系数(或称作“谱”)就是矩阵 $\boldsymbol{A}$ 的特征值。

定义 3-2 一个 $n\times n$ 实对称矩阵 $\boldsymbol{A}$，若对任意非零向量 $\boldsymbol{x}\in R^n$，有 $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Ax}\geqslant 0$，则称 $\boldsymbol{A}$ 为**半正定的**(positive semidefinite)。

定理 3-3　半正定矩阵的所有特征值都为非负实数。

证明：对于 $n\times n$ 半正定矩阵 $\boldsymbol{A}$，若 λ 是 $\boldsymbol{A}$ 的特征值，$\boldsymbol{x}$ 是属于 λ 的特征向量，则对任意非零向量 $\boldsymbol{x}\in R^n$，有 $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}=\lambda\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}=\lambda\|\boldsymbol{x}\|^2$。

因为 $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}\geqslant 0$，所以

$$\lambda=\frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}}{\|\boldsymbol{x}\|^2}\geqslant 0$$

定理 3-4　若 $\boldsymbol{A}$ 为一 $m\times n$ 实矩阵，则 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 为 $n\times n$ 实对称半正定矩阵，$\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$ 为 $m\times m$ 实对称半正定矩阵，且 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$、$\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$ 的所有特征值均为非负实数。

证明：若 $\boldsymbol{A}$ 为一 $m\times n$ 实矩阵，则 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 为 $n\times n$ 实对称矩阵。对任意非零向量 $\boldsymbol{x}\in R^n$，有 $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x}=(\boldsymbol{A}\boldsymbol{x})^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x})=\|\boldsymbol{A}\boldsymbol{x}\|^2\geqslant 0$，所以 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 为 $n\times n$ 半正定矩阵，其所有特征值均为非负实数。

同理可证，$\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$ 为 $m\times m$ 实对称半正定矩阵，其所有特征值也都为非负实数。

定理 3-4 的几何意义在于，如果将 $n\times n$ 实对称矩阵 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 看作一个坐标变换，则该变换可分解成两次到规范正交基的投影变换和一次拉伸变换（谱定理），其中投影变换的基对应矩阵 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 的各特征向量，拉伸变换的倍数对应矩阵 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 的各特征值，它们均为非负实数。同理，$\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$ 与此类似。

3. 矩阵的奇异值分解

定理 3-5（SVD 定理）　若 $\boldsymbol{A}$ 为一 $m\times n$ 实矩阵（假设 $m>n$），则 $\boldsymbol{A}$ 有一个奇异值分解，即 $\boldsymbol{A}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}$，其中 $\boldsymbol{U}$ 是一 $m\times m$ 正交矩阵，$\boldsymbol{V}$ 是一 $n\times n$ 正交矩阵，$\boldsymbol{\Sigma}$ 是一 $m\times n$ 矩阵。$\boldsymbol{\Sigma}$ 的矩阵形式为

$$\boldsymbol{\Sigma}=\begin{bmatrix}\boldsymbol{\Sigma}_1\\ \boldsymbol{0}\end{bmatrix},\quad \text{其中}\boldsymbol{\Sigma}_1=\begin{bmatrix}\sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n\end{bmatrix} \tag{3-63}$$

其对角线元素满足

$$\sigma_1\geqslant\sigma_2\geqslant\cdots\geqslant\sigma_n\geqslant 0\text{ 且 }\sigma_1^2,\sigma_2^2,\cdots,\sigma_n^2\text{ 为 }\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\text{ 的特征值}$$

证明：略（分别用 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 的特征向量构造 $\boldsymbol{V}$，用 $\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$ 的特征向量构造 $\boldsymbol{U}$）。

定理 3-6　$\boldsymbol{A}$ 为一 $m\times n$ 实矩阵，若 $\boldsymbol{A}$ 的奇异值分解为 $\boldsymbol{A}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}$，则

（1）$\boldsymbol{U}=(\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_m)$ 的列向量为 $\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$ 的特征向量，且 $\boldsymbol{U}$ 可对角化 $\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$，即

$$\boldsymbol{U}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{U}=\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{\mathrm{T}}$$

（2）$\boldsymbol{V}=(\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_n)$ 的列向量为 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$ 的特征向量，且 $\boldsymbol{V}$ 可对角化 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$，即

$$\boldsymbol{V}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}=\boldsymbol{\Sigma}^{\mathrm{T}}\boldsymbol{\Sigma}$$

其中 $\boldsymbol{\Sigma}$ 如式（3-63）所示，$\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{\mathrm{T}}$（或 $\boldsymbol{\Sigma}^{\mathrm{T}}\boldsymbol{\Sigma}$）为对角矩阵，其对角元素为 $\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$（或 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$）的特征值。

证明：略（将 $\boldsymbol{A}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}$ 代入 $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{U}$、$\boldsymbol{V}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{V}$ 即可）。

4. 样本特征的协方差矩阵

给定训练集 $D_{\text{train}}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\}$，其中 $\boldsymbol{x}_i$ 是 d 维样本特征，则定义训练集特征矩阵 $\boldsymbol{X}$ 为

$$\boldsymbol{X}_{d\times m}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m)=\begin{bmatrix}x_{11} & x_{21} & \cdots & x_{m1}\\ x_{12} & x_{22} & \cdots & x_{m2}\\ \vdots & \vdots & \ddots & \vdots\\ x_{1d} & x_{2d} & \cdots & x_{md}\end{bmatrix} \tag{3-64}$$

其中的每一列对应一个样本数据 $\boldsymbol{x}_i$，$i=1,2,\cdots,m$。

换个角度，可以将 d 维样本特征的每个特征项看作一个随机变量 X_k，$k=1,2,\cdots,d$，则特征矩阵 $\boldsymbol{X}$ 的第 k 行就是随机变量 X_k 的样本数据。每个特征随机变量 X_k 有 m 个样本数据，计算其均值，

$$\boldsymbol{\mu}=\begin{bmatrix}\mu_1\\ \mu_2\\ \vdots\\ \mu_d\end{bmatrix},\quad \mu_k=\frac{1}{m}\sum_{i=1}^{m}x_{ik},\quad k=1,2,\cdots,d$$

再将特征随机变量的协方差矩阵(或称**特征内积矩阵**)记作$\boldsymbol{\Sigma}$(与式(3-63)中的$\boldsymbol{\Sigma}$没有关系，仅仅是用了相同的记号)，其计算公式为

$$\begin{aligned}\boldsymbol{\Sigma}_{d\times d}&=\frac{1}{m-1}(\boldsymbol{x}_1-\boldsymbol{\mu},\boldsymbol{x}_2-\boldsymbol{\mu},\cdots,\boldsymbol{x}_m-\boldsymbol{\mu})(\boldsymbol{x}_1-\boldsymbol{\mu},\boldsymbol{x}_2-\boldsymbol{\mu},\cdots,\boldsymbol{x}_m-\boldsymbol{\mu})^{\mathrm{T}}\\ &=\frac{1}{m-1}\begin{bmatrix}x_{11}-\mu_1 & x_{21}-\mu_1 & \cdots & x_{m1}-\mu_1\\ x_{12}-\mu_2 & x_{22}-\mu_2 & \cdots & x_{m2}-\mu_2\\ \vdots & \vdots & \ddots & \vdots\\ x_{1d}-\mu_d & x_{2d}-\mu_d & \cdots & x_{md}-\mu_d\end{bmatrix}\begin{bmatrix}x_{11}-\mu_1 & x_{21}-\mu_1 & \cdots & x_{m1}-\mu_1\\ x_{12}-\mu_2 & x_{22}-\mu_2 & \cdots & x_{m2}-\mu_2\\ \vdots & \vdots & \ddots & \vdots\\ x_{1d}-\mu_d & x_{2d}-\mu_d & \cdots & x_{md}-\mu_d\end{bmatrix}^{\mathrm{T}}\end{aligned}$$

由定理 3-4 可知，协方差矩阵$\boldsymbol{\Sigma}$是一个 $d\times d$ 实对称半正定矩阵。记

$$\boldsymbol{\Sigma}_{d\times d}=\begin{bmatrix}\sigma_{11}^2 & \sigma_{12}^2 & \cdots & \sigma_{1d}^2\\ \sigma_{21}^2 & \sigma_{22}^2 & \cdots & \sigma_{2d}^2\\ \vdots & \vdots & \ddots & \vdots\\ \sigma_{d1}^2 & \sigma_{d2}^2 & \cdots & \sigma_{dd}^2\end{bmatrix}$$

其中，$\sigma_{kl}^2=\frac{1}{m-1}\sum_{i=1}^{m}(x_{ki}-\mu_k)(x_{li}-\mu_l)$，$k,l=1,2,\cdots,d$，并且$\boldsymbol{\Sigma}$中的对角元素为各特征项的方差，非对角元素为不同特征项之间的协方差。

如果先对样本特征做**去中心化**(zero-centered，或称零均值化)处理，即将各特征项样本数据分别减去其均值，这样可以简化协方差矩阵的计算。去中心化后每个特征随机变量 X_k 的均值$\mu_k=0$，特征的协方差矩阵$\boldsymbol{\Sigma}$因此可简化为

$$\boldsymbol{\Sigma}_{d\times d}=\frac{1}{m-1}\begin{bmatrix}x_{11} & x_{21} & \cdots & x_{m1}\\ x_{12} & x_{22} & \cdots & x_{m2}\\ \vdots & \vdots & \ddots & \vdots\\ x_{1d} & x_{2d} & \cdots & x_{md}\end{bmatrix}\begin{bmatrix}x_{11} & x_{21} & \cdots & x_{m1}\\ x_{12} & x_{22} & \cdots & x_{m2}\\ \vdots & \vdots & \ddots & \vdots\\ x_{1d} & x_{2d} & \cdots & x_{md}\end{bmatrix}^{\mathrm{T}}=\frac{1}{m-1}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} \tag{3-65}$$

记

$$\boldsymbol{\Sigma}_{d\times d}=\begin{bmatrix}\sigma_{11}^2 & \sigma_{12}^2 & \cdots & \sigma_{1d}^2\\ \sigma_{21}^2 & \sigma_{22}^2 & \cdots & \sigma_{2d}^2\\ \vdots & \vdots & \ddots & \vdots\\ \sigma_{d1}^2 & \sigma_{d2}^2 & \cdots & \sigma_{dd}^2\end{bmatrix}$$

其中，$\sigma_{kl}^2=\dfrac{1}{m-1}\sum_{i=1}^{m}x_{ki}x_{li}$，$k,l=1,2,\cdots,d$。

3.4.2　主成分分析

样本特征由多个特征项组成，可以将每个特征项看作样本特征的一个**成分**（component，或称分量），每个成分（即每个特征项）都是一个随机变量。**主成分分析**（Principal Component Analysis，PCA）是一种常用的特征降维方法，其核心思想是：

- 两个特征项之间的协方差反映它们的相关性。协方差越大，则特征项之间的相关性就越大，数据冗余也越大。机器学习应尽可能消除数据冗余，也即降低特征项之间的相关性，让它们的协方差越接近于零越好。
- 特征项的方差反映所携带的信息量。方差越大，则特征项所携带的信息量就越大。方差大的特征项属于样本特征的主要成分，方差小的则属于次要成分。特征降维时应保留样本特征的主要成分，丢弃次要成分。
- 通过对样本特征进行投影变换，尽可能降低不同特征项之间的相关性，然后仅保留样本特征的主要成分即可实现降维。PCA 降维的关键是选择投影方向，找出最优的投影基向量。

1. 标准化样本特征

给定训练集 $D_{\text{train}}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\}$，或 $D_{\text{train}}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$，其中 $\boldsymbol{x}_i$ 是 d 维样本特征。PCA 降维要求先对原始样本特征做去中心化和归一化处理（例如 z-score 标准化），去中心化是为了简化协方差矩阵的计算，归一化是为了让不同特征之间的方差可比较；然后构造样本的特征矩阵 $\boldsymbol{X}_{d\times m}$（见式(3-64)），并计算其协方差矩阵 $\boldsymbol{\Sigma}_{d\times d}$（见式(3-65)）。为便于后续讲解，这里将特征矩阵 $\boldsymbol{X}_{d\times m}$ 的协方差矩阵 $\boldsymbol{\Sigma}_{d\times d}$ 改记为 $\boldsymbol{\Sigma}_{d\times d}^{\boldsymbol{X}}$。

2. 选择最优特征变换方向

给定 d 维样本特征数据 $\boldsymbol{x}$ 或样本特征矩阵 $\boldsymbol{X}_{d\times m}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m)$，从原始基向量，例如标准基 $\{\boldsymbol{e}_1,\boldsymbol{e}_2,\cdots,\boldsymbol{e}_d\}$，到另一组规范正交基 $\{\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_d\}$ 对样本特征做投影变换，变换后的坐标可表示为如下矩阵形式：

$$\boldsymbol{z}=\boldsymbol{W}\boldsymbol{x}=\begin{bmatrix}\boldsymbol{w}_1^{\mathrm{T}}\\ \boldsymbol{w}_2^{\mathrm{T}}\\ \vdots\\ \boldsymbol{w}_d^{\mathrm{T}}\end{bmatrix}\boldsymbol{x}\tag{3-66a}$$

或

$$\boldsymbol{Z}_{d\times m}=\boldsymbol{W}\boldsymbol{X}_{d\times m}=\begin{bmatrix}\boldsymbol{w}_1^{\mathrm{T}}\\ \boldsymbol{w}_2^{\mathrm{T}}\\ \vdots\\ \boldsymbol{w}_d^{\mathrm{T}}\end{bmatrix}(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m) \tag{3-66b}$$

其中，$\boldsymbol{W}$ 为 $d\times d$ 转移矩阵，$\boldsymbol{z}$ 为单个样本特征 $\boldsymbol{x}$ 投影变换后的新特征向量；$\boldsymbol{Z}_{d\times m}$ 为样本特征矩阵 $\boldsymbol{X}_{d\times m}$ 投影变换后的新特征矩阵，其展开式为

$$\boldsymbol{Z}_{d\times m}=(\boldsymbol{z}_1,\boldsymbol{z}_2,\cdots,\boldsymbol{z}_m)=\begin{bmatrix}z_{11} & z_{21} & \cdots & z_{m1}\\ z_{12} & z_{22} & \cdots & z_{m2}\\ \vdots & \vdots & \ddots & \vdots\\ z_{1d} & z_{2d} & \cdots & z_{md}\end{bmatrix}$$

因为样本特征矩阵 $\boldsymbol{X}_{d\times m}$ 经过去中心化处理，所以每一行的均值或累加和都为 0。投影变换所得到新的样本特征矩阵 $\boldsymbol{Z}_{d\times m}$，其每一行的均值与累加和也保持为零。

如果将新样本特征的每个特征项（$\boldsymbol{Z}_{d\times m}$ 中的每一行）看作一个随机变量 Z_k，$k=1,2,\cdots,d$，则新特征随机变量的协方差矩阵为

$$\boldsymbol{\Sigma}_{d\times d}^{\boldsymbol{Z}}=\frac{1}{m-1}\begin{bmatrix}z_{11} & z_{21} & \cdots & z_{m1}\\ z_{12} & z_{22} & \cdots & z_{m2}\\ \vdots & \vdots & \ddots & \vdots\\ z_{1d} & z_{2d} & \cdots & z_{md}\end{bmatrix}\begin{bmatrix}z_{11} & z_{21} & \cdots & z_{m1}\\ z_{12} & z_{22} & \cdots & z_{m2}\\ \vdots & \vdots & \ddots & \vdots\\ z_{1d} & z_{2d} & \cdots & z_{md}\end{bmatrix}^{\mathrm{T}}=\frac{1}{m-1}\boldsymbol{Z}\boldsymbol{Z}^{\mathrm{T}}$$

PCA 希望新的特征随机变量 Z_k 之间相关性越小越好，这样可以消除数据冗余。理想情况下，协方差矩阵 $\boldsymbol{\Sigma}_{d\times d}^{\boldsymbol{Z}}$ 应为对角阵，即各特征项之间的协方差都为 0：

$$\boldsymbol{\Sigma}_{d\times d}^{\boldsymbol{Z}}=\begin{bmatrix}\sigma_{11}^2 & & & \\ & \sigma_{22}^2 & & \\ & & \ddots & \\ & & & \sigma_{dd}^2\end{bmatrix}$$

因为

$$\boldsymbol{\Sigma}_{d\times d}^{\boldsymbol{Z}}=\frac{1}{m-1}\boldsymbol{Z}\boldsymbol{Z}^{\mathrm{T}}=\frac{1}{m-1}(\boldsymbol{W}\boldsymbol{X})(\boldsymbol{W}\boldsymbol{X})^{\mathrm{T}}=\boldsymbol{W}\left(\frac{1}{m-1}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\right)\boldsymbol{W}^{\mathrm{T}} \tag{3-67}$$

将式(3-65)代入式(3-67)得

$$\boldsymbol{\Sigma}_{d\times d}^{\boldsymbol{Z}}=\boldsymbol{W}\left(\frac{1}{m-1}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\right)\boldsymbol{W}^{\mathrm{T}}=\boldsymbol{W}(\boldsymbol{\Sigma}_{d\times d}^{\boldsymbol{X}})\boldsymbol{W}^{\mathrm{T}} \tag{3-68}$$

可以看出，如果 $\boldsymbol{W}(\boldsymbol{\Sigma}_{d\times d}^{\boldsymbol{X}})\boldsymbol{W}^{\mathrm{T}}$ 是对角阵，则 $\boldsymbol{\Sigma}_{d\times d}^{\boldsymbol{Z}}$ 就是对角阵。那么该如何选择转移矩阵 $\boldsymbol{W}$，使得 $\boldsymbol{W}(\boldsymbol{\Sigma}_{d\times d}^{\boldsymbol{X}})\boldsymbol{W}^{\mathrm{T}}$ 为对角阵呢？

$\boldsymbol{\Sigma}_{d\times d}^{\boldsymbol{X}}$ 是实对称半正定矩阵，因此存在一个正交矩阵 $\boldsymbol{U}$ 可对角化 $\boldsymbol{\Sigma}_{d\times d}^{\boldsymbol{X}}$（谱定理），即

$$\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{\Sigma}_{d\times d}^{\boldsymbol{X}})\boldsymbol{U}=\begin{bmatrix}\boldsymbol{u}_1^{\mathrm{T}}\\ \boldsymbol{u}_2^{\mathrm{T}}\\ \vdots\\ \boldsymbol{u}_d^{\mathrm{T}}\end{bmatrix}(\boldsymbol{\Sigma}_{d\times d}^{\boldsymbol{X}})(\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_d)=\begin{bmatrix}\lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_d\end{bmatrix} \tag{3-69}$$

其中，$\boldsymbol{U}$ 为 $d\times d$ 正交矩阵，$\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_d$ 为矩阵 $\boldsymbol{\Sigma}^{\boldsymbol{X}}_{d\times d}$ 的 d 个线性无关特征向量（单位向量）；$\boldsymbol{x}=\mathrm{diag}(\lambda_1,\lambda_2,\cdots,\lambda_d)$ 为对角矩阵，对角元素分别为特征向量 $\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_d$ 对应的特征值 $\lambda_1,\lambda_2,\cdots,\lambda_d$。如果以 $\{\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_d\}$ 作为投影变换的基向量 $\{\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_d\}$，则转移矩阵 $\boldsymbol{W}_{d\times d}=\boldsymbol{U}^{\mathrm{T}}_{d\times d}$，即

$$\boldsymbol{W}_{d\times d}=\boldsymbol{U}^{\mathrm{T}}_{d\times d}=\begin{bmatrix}\boldsymbol{u}_1^{\mathrm{T}}\\ \boldsymbol{u}_2^{\mathrm{T}}\\ \vdots\\ \boldsymbol{u}_d^{\mathrm{T}}\end{bmatrix} \tag{3-70}$$

将式(3-70)、式(3-69)代入式(3-68)得

$$\boldsymbol{\Sigma}^{\boldsymbol{Z}}_{d\times d}=\boldsymbol{W}(\boldsymbol{\Sigma}^{\boldsymbol{X}}_{d\times d})\boldsymbol{W}^{\mathrm{T}}=\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{\Sigma}^{\boldsymbol{X}}_{d\times d})\boldsymbol{U}=\begin{bmatrix}\lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_d\end{bmatrix} \tag{3-71}$$

从式(3-71)可以看出，选择协方差矩阵 $\boldsymbol{\Sigma}^{\boldsymbol{X}}_{d\times d}$ 的单位特征向量构造一组规范正交基 $\{\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_d\}$，则对样本特征从原始基向量到新基向量 $\{\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_d\}$ 做投影变换，变换得到的新样本特征具有如下特性。

(1) 各特征项之间相互独立，协方差为零。**结论 1**：由协方差矩阵 $\boldsymbol{\Sigma}^{\boldsymbol{X}}_{d\times d}$ 各单位特征向量所构造的规范正交基 $\{\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_d\}$，就是 PCA 所要选择的最优投影基向量，也即最优的变换方向。

(2) 第 k 个特征项 Z_k 的方差为对应基向量 $\boldsymbol{u}_k$ 的特征值 λ_k。**结论 2**：对各特征值排序，$\lambda_1\geqslant\lambda_2\geqslant\cdots\geqslant\lambda_d\geqslant 0$，则特征值大的基向量代表样本特征的主要成分，其所对应的特征项应当保留；特征值小的基向量代表样本特征的次要成分，其所对应的特征项可以丢弃。

3. PCA 算法步骤

给定训练集 $D_{\mathrm{train}}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\}$，或 $D_{\mathrm{train}}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$，其中 $\boldsymbol{x}_i$ 是 d 维样本特征。

1) 样本特征标准化

构造训练集特征矩阵 $\boldsymbol{X}^{\boldsymbol{o}}$，计算各特征项均值 μ_k、标准差 σ_k，对 $\boldsymbol{X}^{\boldsymbol{o}}$ 进行 z-score 标准化，得到标准化特征矩阵 $\boldsymbol{X}$。

$$\boldsymbol{X}^{\boldsymbol{o}}_{d\times m}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m)=\begin{bmatrix}x_{11} & x_{21} & \cdots & x_{m1}\\ x_{12} & x_{22} & \cdots & x_{m2}\\ \vdots & \vdots & \ddots & \vdots\\ x_{1d} & x_{2d} & \cdots & x_{md}\end{bmatrix}$$

$$\mu_k=\frac{1}{m}\sum_{i=1}^{m}x_{ik},\quad \sigma_k=\sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(x_{ik}-\mu_k)^2},\quad k=1,2,\cdots,d$$

$$\boldsymbol{X}_{d\times m}=\begin{bmatrix}(x_{11}-\mu_1)/\sigma_1 & (x_{21}-\mu_1)/\sigma_1 & \cdots & (x_{m1}-\mu_1)/\sigma_1\\ (x_{12}-\mu_2)/\sigma_2 & (x_{22}-\mu_2)/\sigma_2 & \cdots & (x_{m2}-\mu_2)/\sigma_2\\ \vdots & \vdots & \ddots & \vdots\\ (x_{1d}-\mu_d)/\sigma_d & (x_{2d}-\mu_d)/\sigma_d & \cdots & (x_{md}-\mu_d)/\sigma_d\end{bmatrix} \tag{3-72}$$

2) 对协方差矩阵$\boldsymbol{\Sigma}^{\boldsymbol{X}}_{d\times d}$做特征值分解

按照式(3-65)计算协方差矩阵$\boldsymbol{\Sigma}^{\boldsymbol{X}}_{d\times d}$,然后进行特征值分解,求得特征值$\lambda_1,\lambda_2,\cdots,\lambda_d$及其对应的特征向量$\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_d$。

3) 构造转移矩阵$\boldsymbol{W}$

假设将d维样本特征降到d'维,则先对特征值排序,按从大到小的顺序选取前d'个特征值所对应的特征向量$\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_{d'}$,以它们为行向量构造出投影变换的转移矩阵$\boldsymbol{W}_{d'\times d}$。

4) 特征降维

给定d维样本特征$\boldsymbol{x}$,按照计算公式$\boldsymbol{z}=\boldsymbol{W}_{d'\times d}\boldsymbol{x}$,求得降维后的$d'$维样本特征$\boldsymbol{z}$。

注:算法第2步可以将特征值分解改为奇异值分解。这时可以不计算协方差矩阵$\boldsymbol{\Sigma}^{\boldsymbol{X}}_{d\times d}$,而是直接对训练集特征矩阵$\boldsymbol{X}$做奇异值分解,即$\boldsymbol{X}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}$,求得奇异值$\sigma_1,\sigma_2,\cdots,\sigma_d$。奇异值$\sigma_i$与协方差矩阵$\boldsymbol{\Sigma}^{\boldsymbol{X}}_{d\times d}$特征值$\lambda_i$之间的关系是:$\sigma_i^2=\lambda_i$,奇异值$\sigma_i$在$\boldsymbol{U}$中的左奇异向量(列向量)也恰好对应着协方差矩阵$\boldsymbol{\Sigma}^{\boldsymbol{X}}_{d\times d}$的特征向量$\boldsymbol{u}_i$。按训练集特征矩阵$\boldsymbol{X}$奇异值大小,选择$d'$个单位左奇异向量为行向量构造出投影变换的转移矩阵$\boldsymbol{W}_{d'\times d}$,它与按协方差矩阵$\boldsymbol{\Sigma}^{\boldsymbol{X}}_{d\times d}$特征值分解求得的转移矩阵完全一致。

4. 核主成分分析

样本特征是被观测到的数据,它反映了事物的某种内在规律。这种内在规律可能是在低维空间上展开,却在高维空间被观测到并记录成高维空间的特征。例如,单摆上的小球在一维曲线上做往复摆动(见图3-24),但观测记录的特征却是二维(甚至是三维)位置坐标。这种情况被描述为,具有低维结构的特征被**嵌入**(embedding)高维空间中,简称"低维嵌入"。

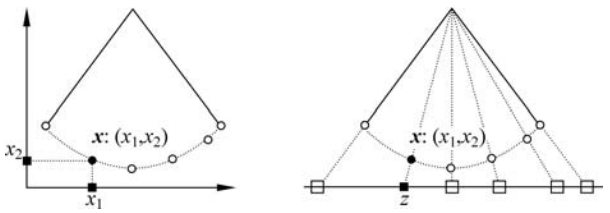


图3-24 单摆小球在一维曲线上做往复摆动

协方差矩阵是很多机器学习算法的基础,它反映了特征(即随机变量)之间的相关性或特征之间的某种结构。PCA降维就是在构造标准化特征矩阵$\boldsymbol{X}$之后,先按式(3-65)计算其协方差矩阵$\boldsymbol{\Sigma}^{\boldsymbol{X}}_{d\times d}$,然后再基于协方差矩阵进行降维。

核主成分分析(Kernel PCA,KPCA)的核心思想:在低维嵌入的情况下,应当基于低维特征协方差而不是高维特征协方差来进行降维,否则会丢失特征的低维结构。

如果知道低维特征是如何嵌入到高维空间中去的,例如已知嵌入函数ϕ:低维特征→高维特征(或ϕ^{-1}:高维特征→低维特征),则可以根据嵌入函数ϕ找出计算低维特征协方差的办法。但多数情况下,嵌入函数ϕ是未知的。

在嵌入函数ϕ未知的情况下,如何根据观测到的高维特征去计算低维特征的协方差呢?协方差计算是基于向量内积的。可以基于高维特征定义一个函数来模拟低维特征的内积,这样的函数被称作**核函数**(kernel function),通常记作$K(\boldsymbol{x}_i,\boldsymbol{x}_j)$。其中$\boldsymbol{x}_i,\boldsymbol{x}_j$是两个

高维特征向量，而函数值则是它们在低维空间对应特征向量的内积（即 $\phi^{-1}(\boldsymbol{x}_i)\cdot\phi^{-1}(\boldsymbol{x}_j)$）。

至于如何定义或选择核函数 $K(\boldsymbol{x}_i,\boldsymbol{x}_j)$，目前还没有统一的方法。常用的核函数有**径向基函数**（Radial Basis Function，RBF）核、**sigmoid** 核（sigmoid kernel）、**多项式**核（polynomial kernel）等。**径向基函数核**（也称 RBF 核、高斯核）是一种常用的核函数，其函数形式为

$$K(\boldsymbol{x}_i,\boldsymbol{x}_j)=\mathrm{e}^{-\gamma\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2}$$

定义核函数之后，就可以按照核函数计算低维特征的协方差矩阵，记作$\boldsymbol{\Sigma}^{\boldsymbol{X}}_{\mathrm{KPCA}}$。给定训练集样本特征矩阵

$$\boldsymbol{X}_{d\times m}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m)=\begin{bmatrix}x_{11} & x_{21} & \cdots & x_{m1}\\ x_{12} & x_{22} & \cdots & x_{m2}\\ \vdots & \vdots & \ddots & \vdots\\ x_{1d} & x_{2d} & \cdots & x_{md}\end{bmatrix}$$

将样本特征的每个特征项（$\boldsymbol{X}_{d\times m}$ 的每一行）看作一个随机变量 X_k，$k=1,2,\cdots,d$，特征随机变量的低维协方差矩阵$\boldsymbol{\Sigma}^{\boldsymbol{X}}_{\mathrm{KPCA}}$ 为

$$\boldsymbol{\Sigma}^{\boldsymbol{X}}_{\mathrm{KPCA}}=\begin{bmatrix}K(X_1,X_1) & K(X_1,X_2) & \cdots & K(X_1,X_d)\\ K(X_2,X_1) & K(X_2,X_2) & \cdots & K(X_2,X_d)\\ \vdots & \vdots & \ddots & \vdots\\ K(X_d,X_1) & K(X_d,X_2) & \cdots & K(X_d,X_d)\end{bmatrix}\tag{3-73}$$

其中，$K(X_k,X_l)=\phi^{-1}(X_k)\cdot\phi^{-1}(X_l)$，$k,l=1,2,\cdots,d$。

KPCA 将基于这个低维特征协方差矩阵$\boldsymbol{\Sigma}^{\boldsymbol{X}}_{\mathrm{KPCA}}$进行降维，其后续算法步骤与 PCA 完全一样：即先对$\boldsymbol{\Sigma}^{\boldsymbol{X}}_{\mathrm{KPCA}}$做特征值分解，求出最优的特征投影方向，然后构造投影变换的转移矩阵 $\boldsymbol{W}$。需要注意的是，核函数 $K(\boldsymbol{x}_i,\boldsymbol{x}_j)$应具有对称性，并使所构造的协方差矩阵$\boldsymbol{\Sigma}^{\boldsymbol{X}}_{\mathrm{KPCA}}$为半正定，这样后续的特征值分解等算法步骤才能成立。

5. 使用 scikit-learn 库中的 PCA 降维模型

scikit-learn 库将 PCA、KPCA 降维模型封装成类（类名分别为 **PCA** 和 **KernelPCA**），并将其存放在 sklearn.decomposition 模块中。PCA 和 KernelPCA 类实现了降维模型的学习算法 **fit**()和降维转换算法 **transform**()。

图 3-25 给出了使用 PCA 类进行特征降维，然后再使用逻辑斯谛回归分类器 LogisticRegression 建立并训练乳腺癌诊断模型的示例代码。其中数据集使用的是 3.1.3 节 z-score 标准化后的乳腺癌诊断数据集，先将数据集（X1_train、X1_test）从 30 维降到 5 维，查看这 5 个新特征的方差贡献率；再使用降维后的 5 个特征训练模型，并对测试集 X1_test 前 2 个病例样本进行分类预测，将预测结果 predict 与 Y_test 中的真实诊断结果 malignant 进行比对，1 表示恶性，0 表示良性；最后使用函数 score()计算模型在训练集（X1_train、Y_train）和测试集（X1_test、Y_test）上的平均正确率。

3.4.3　线性判别分析

3.2.2 节曾介绍过使用线性判别分析（LDA）解决二分类问题，其思路是先将高维特征

```
In [14]: from sklearn.decomposition import PCA
         pca = PCA(n_components=5)
         pca.fit(X1_train)
         print("explained_variance_ratio_=", pca.explained_variance_ratio_)
         X2_train = pca.transform(X1_train)
         X2_test = pca.transform(X1_test)

         from sklearn.linear_model import LogisticRegression
         lr = LogisticRegression(penalty='l2', C=1.0, random_state=2020)
         lr.fit(X2_train, Y_train)

         Y2 = lr.predict(X2_test[:2])
         print(X2_test[:2]); print()
         print("predict:", Y2," malignant:", Y_test.values[:2])
         print("mean accuracy on train:", lr.score(X2_train, Y_train))
         print("mean accuracy on test:", lr.score(X2_test, Y_test))

         explained_variance_ratio_= [0.4376296  0.1880184  0.09455378 0.06845665
         0.05663576]
         [[ 8.55763176 -4.09563401 -0.10790408 -0.65646783 -0.28506859]
          [-0.66950192  1.78227379 -0.6457208  -0.73212841 -0.99798661]]

         predict: [0 1]   malignant: [0 1]
         mean accuracy on train: 0.978021978021978
         mean accuracy on test: 0.9736842105263158
```

图 3-25 使用 PCA 类进行特征降维,然后再使用逻辑斯谛回归分类器 LogisticRegression 并建立乳腺癌诊断模型的示例代码

投影到一维,然后基于一维特征来设计分类器。对于多分类问题,可以对高维特征做若干次投影,得到一组新特征,然后再设计分类器。LDA 多次投影的过程实际上就是消除特征冗余,选择主要成分的降维过程,这与 PCA 非常类似。LDA 既可作为分类器模型,也可作为降维模型。

PCA 不考虑类别标注,其投影原则是将高维特征投影到协方差最小的方向。而 LDA 使用带类别标注的训练集,其投影原则是将高维特征投影到"类内方差最小,类间方差最大"的方向。PCA 不考虑类别标注,属于无监督降维;而 LDA 考虑类别标注,属于有监督降维。这是两者的根本区别,它们后续的投影原则及学习算法也因此有很大不同。

1. LDA 降维算法

下面用数学方法来描述 LDA 选择最优投影方向$\boldsymbol{\omega}$的过程,整个过程分 4 步完成。给定训练集 $D_{\text{train}}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\}$,其中 $\boldsymbol{x}_i=(x_{i1},x_{i2},\cdots,x_{id})$是 d 维样本特征,$y_i\in\{c_1,c_2,\cdots,c_N\}$是其对应的实际类别,共 N 个类别。假设训练集 D_{train} 中第 k 类样本有 m_k 个,将其从数据集中拆分出来形成子集 D_k,即

$$D_k=\{(\boldsymbol{x}_1,c_k),(\boldsymbol{x}_2,c_k),\cdots,(\boldsymbol{x}_{m_k},c_k)\},\quad k=1,2,\cdots,N$$

分别定义各类特征的均值$\boldsymbol{\mu}_k$和离散度(类似于协方差)矩阵 $\boldsymbol{S}_k$,

$$\boldsymbol{\mu}_k=\frac{1}{m_k}\sum_{i=1}^{m_k}\boldsymbol{x}_i,\quad \boldsymbol{S}_k=\sum_{i=1}^{m_k}(\boldsymbol{x}_i-\boldsymbol{\mu}_k)(\boldsymbol{x}_i-\boldsymbol{\mu}_k)^{\mathrm{T}},\quad k=1,2,\cdots,N$$

再定义 d 维特征空间的**类内离散度矩阵**(within-class scatter matrix)$\boldsymbol{S}_w$,

$$\boldsymbol{S}_w=\sum_{k=1}^{N}P(Y=c_k)\boldsymbol{S}_k=\sum_{k=1}^{N}\frac{m_k}{m}\boldsymbol{S}_k$$

和 d 维特征空间的**类间离散度矩阵**(between-class scatter matrix)$\boldsymbol{S}_b$ 为

$$\boldsymbol{\mu}=\frac{1}{m}\sum_{i=1}^{m}\boldsymbol{x}_i,\quad \boldsymbol{S}_b=\sum_{k=1}^{N}\frac{m_k}{m}(\boldsymbol{\mu}_k-\boldsymbol{\mu})(\boldsymbol{\mu}_k-\boldsymbol{\mu})^{\mathrm{T}}$$

LDA 最多可以做 $N-1$ 次投影，并希望各投影方向$\boldsymbol{\omega}$ 都尽量让样本特征投影点“类内方差最小，类间方差最大”，因此构造如下优化目标：

$$\max_{\boldsymbol{W}} \frac{\boldsymbol{W}^{\mathrm{T}}\boldsymbol{S}_b\boldsymbol{W}}{\boldsymbol{W}^{\mathrm{T}}\boldsymbol{S}_w\boldsymbol{W}}, \quad \text{其中} \quad \boldsymbol{W}_{d\times(N-1)}=(\boldsymbol{\omega}_1,\boldsymbol{\omega}_2,\cdots,\boldsymbol{\omega}_{N-1})$$

求解得 $\boldsymbol{S}_b\boldsymbol{W}=\lambda\boldsymbol{S}_w\boldsymbol{W}$。这是一个**广义特征值问题**(generalized eigenvalue problem)。

如果需要将 d 维样本特征降到 d' 维，则取 $\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$ 的 d' 个最大广义特征值所对应的特征向量，将其作为行向量构造出投影变换的转移矩阵 $\boldsymbol{W}$。需要注意的是，假设有 N 个类别，原始样本特征为 d 维，则 d' 应满足：$d'\leqslant\min(N-1,d)$。

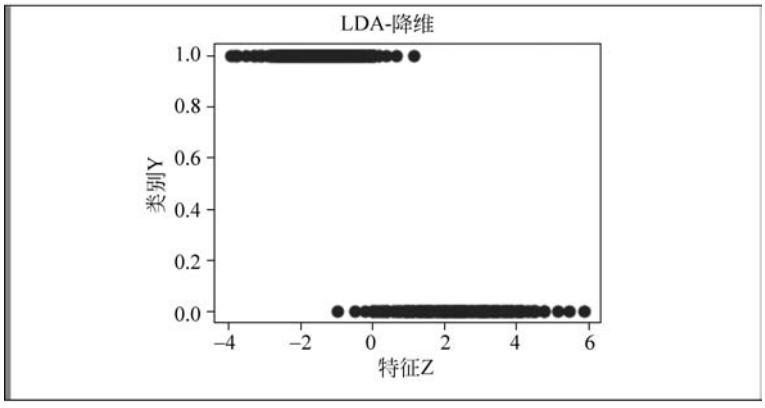
2. 使用 scikit-learn 库中的 LDA 降维模型

scikit-learn 库将 LDA 的降维模型与分类模型合在一起，封装成同一个类(类名为**LinearDiscriminantAnalysis**)，并将其存放在 sklearn. discriminant _ analysis 模块中。LinearDiscriminantAnalysis 类实现了降维/分类器模型的学习算法 **fit**()、降维转换算法 **transform**()和预测算法 **predict**()。

图 3-26(a)给出了使用 LinearDiscriminantAnalysis 类对乳腺癌诊断数据集进行特征降维的示例代码。其中数据集使用的是 3.1.3 节标准化后的乳腺癌诊断数据集，将数据集(X1_train、X1_test)从 30 维降到 1 维(二分类问题只能被降到一维)；然后使用散点图对其进行可视化(见图 3-26(b))。

```
In [15]: from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
         lda = LinearDiscriminantAnalysis(n_components=1)
         lda.fit(X, Y)

         fig = plt.figure( figsize=(4, 3), dpi=100 )
         plt.rcParams['font.sans-serif'] = ['SimHei']
         plt.rcParams['axes.unicode_minus'] = False

         plt.scatter(lda.transform(X), Y)
         plt.title("LDA - 降维")
         plt.xlabel("特征 - Z"); plt.ylabel("类别 - Y")
         plt.show()
```

(a) 示例代码



(b) 降维后的特征-类别散点图

图 3-26　使用 LinearDiscriminantAnalysis 类对乳腺癌诊断数据集进行特征降维

3.4.4 非线性降维

PCA 和 LDA 降维使用线性变换,将低维特征表示成高维特征的线性组合,这属于线性降维。线性降维的核心思想是降维时尽可能保持高维特征的方差或类间方差,以便下一步进行回归或分类。而某些降维问题则希望降维时能尽量保持样本在高维空间的某种低维分布结构,以便下一步进行可视化与数据分析。通常,高维数据必须降到二维或三维才便于可视化。

给定训练集 $D_{\text{train}}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\}$,或 $D_{\text{train}}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$,其中 $\boldsymbol{x}_i$ 是 d 维样本特征,构造训练集特征矩阵 $\boldsymbol{X}$ 为

$$\boldsymbol{X}_{d\times m}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m)=\begin{bmatrix}x_{11} & x_{21} & \cdots & x_{m1}\\ x_{12} & x_{22} & \cdots & x_{m2}\\ \vdots & \vdots & \ddots & \vdots\\ x_{1d} & x_{2d} & \cdots & x_{md}\end{bmatrix}$$

其中的每一列对应一个样本数据 $\boldsymbol{x}_i$,$i=1,2,\cdots,m$。

现在希望将 d 维样本特征 $\boldsymbol{x}_i$(高维特征)降到 d'维样本特征 $\boldsymbol{z}_i$(低维特征,$d'\leqslant d$),降维时能保持样本在高维空间的某种低维分布结构(参见图 3-27 中沿 S 形结构分布的 4 个样本点)。所保持的分布结构可以是样本间的距离(例如图 3-27 中 $\boldsymbol{x}_1$ 与其他样本点之间的直线距离或曲线距离),或样本间的局部线性关系(例如图 3-27 中 $\boldsymbol{x}_1$ 与 $\boldsymbol{x}_2$、$\boldsymbol{x}_3$ 在局部近似具有线性关系),或样本的局部概率分布等。

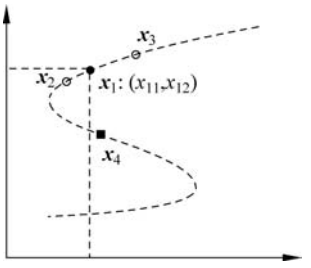


图 3-27 样本在高维空间的分布结构

降维后让低维特征继续保持样本的分布结构,这就需要使用非线性降维方法,而且保持不同分布结构需要不同的降维模型与算法。

1. MDS——保持样本间欧氏距离

多维缩放(Multiple Dimensional Scaling,MDS)降维,就是降维后低维空间中样本点之间的**欧氏距离**(Euclidean distance)与高维空间保持一致。

首先根据训练集特征矩阵 $\boldsymbol{X}$,计算高维空间的**距离矩阵** $\boldsymbol{D}$,

$$\boldsymbol{D}_{m\times m}=\begin{bmatrix}\text{dist}_{11} & \text{dist}_{12} & \cdots & \text{dist}_{1m}\\ \text{dist}_{21} & \text{dist}_{22} & \cdots & \text{dist}_{2m}\\ \vdots & \vdots & \ddots & \vdots\\ \text{dist}_{m1} & \text{dist}_{m2} & \cdots & \text{dist}_{mm}\end{bmatrix} \tag{3-74}$$

其中,dist_{ij} 为样本点 $\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$ 之间的欧氏距离:$\text{dist}_{ij}=\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2$,$i,j=1,2,\cdots,m$。

假设 MDS 降维后,高维空间样本点 $\boldsymbol{x}_i$ 被转换为低维(d'维)空间的 $\boldsymbol{z}_i$,则低维特征矩阵 $\boldsymbol{Z}$ 为

$$\boldsymbol{Z}_{d'\times m}=(\boldsymbol{z}_1,\boldsymbol{z}_2,\cdots,\boldsymbol{z}_m)=\begin{bmatrix}z_{11} & z_{21} & \cdots & z_{m1}\\ z_{12} & z_{22} & \cdots & z_{m2}\\ \vdots & \vdots & \ddots & \vdots\\ z_{1d'} & z_{2d'} & \cdots & z_{md'}\end{bmatrix} \tag{3-75}$$

其中的每一列对应一个低维样本数据 $\boldsymbol{z}_i, i=1,2,\cdots,m$。再构造低维特征矩阵 $\boldsymbol{Z}$ 的**样本内积矩阵**，

$$\boldsymbol{B}_{m\times m}=(\boldsymbol{Z}_{d'\times m})^{\mathrm{T}}\boldsymbol{Z}_{d'\times m}=\begin{bmatrix}\boldsymbol{z}_1^{\mathrm{T}}\\ \boldsymbol{z}_2^{\mathrm{T}}\\ \vdots\\ \boldsymbol{z}_m^{\mathrm{T}}\end{bmatrix}(\boldsymbol{z}_1,\boldsymbol{z}_2,\cdots,\boldsymbol{z}_m)=\begin{bmatrix}\boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_1 & \boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_2 & \cdots & \boldsymbol{z}_1^{\mathrm{T}}\boldsymbol{z}_m\\ \boldsymbol{z}_2^{\mathrm{T}}\boldsymbol{z}_1 & \boldsymbol{z}_2^{\mathrm{T}}\boldsymbol{z}_2 & \cdots & \boldsymbol{z}_2^{\mathrm{T}}\boldsymbol{z}_m\\ \vdots & \vdots & \ddots & \vdots\\ \boldsymbol{z}_m^{\mathrm{T}}\boldsymbol{z}_1 & \boldsymbol{z}_m^{\mathrm{T}}\boldsymbol{z}_2 & \cdots & \boldsymbol{z}_m^{\mathrm{T}}\boldsymbol{z}_m\end{bmatrix}\tag{3-76}$$

依据定理 3-4，样本内积矩阵 $\boldsymbol{B}$ 是 $m\times m$ 实对称半正定矩阵。这里注意样本内积矩阵与协方差矩阵(特征的内积矩阵)的区别。

因为 MDS 降维后，低维空间中样本点 $\boldsymbol{z}_i$ 与 $\boldsymbol{z}_j$ 之间的欧氏距离 $\|\boldsymbol{z}_i-\boldsymbol{z}_j\|_2$ 与其在高维空间中的欧氏距离 dist_{ij} 一致，所以

$$\|\boldsymbol{z}_i-\boldsymbol{z}_j\|_2^2=\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{z}_i+\boldsymbol{z}_j^{\mathrm{T}}\boldsymbol{z}_j-2\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{z}_j=\mathrm{dist}_{ij}^2\tag{3-77}$$

不失一般性，令降维后各低维特征项均被去中心化，即低维特征矩阵 $\boldsymbol{Z}$ 每行元素的均值为 0，则其累加和也为 0，$\sum_{i=1}^{m}z_{ik}=0, k=1,2,\cdots,d'$，由式(3-76)、式(3-77)可推导出

$$\boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{z}_j=-\frac{1}{2}\left(\mathrm{dist}_{ij}^2-\frac{1}{m}\sum_{k=1}^{m}\mathrm{dist}_{ik}^2-\frac{1}{m}\sum_{k=1}^{m}\mathrm{dist}_{kj}^2+\frac{1}{m^2}\sum_{k=1}^{m}\sum_{l=1}^{m}\mathrm{dist}_{kl}^2\right)\tag{3-78}$$

式(3-78)表明，MDS 通过降维前后距离矩阵 $\boldsymbol{D}$ 保持不变即可求得低维特征的样本内积矩阵 $\boldsymbol{B}$。下面再通过矩阵 $\boldsymbol{B}$ 求解低维特征矩阵 $\boldsymbol{Z}$。

样本内积矩阵 $\boldsymbol{B}$ 是 $m\times m$ 实对称半正定矩阵，因此对 $\boldsymbol{B}$ 进行特征值分解可得

$$\boldsymbol{B}=\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^{\mathrm{T}}\tag{3-79}$$

其中，$\boldsymbol{D}=\mathrm{diag}(\lambda_1,\lambda_2,\cdots,\lambda_m)$ 为特征值构造的对角矩阵，且 $\lambda_1\geqslant\lambda_2\geqslant\cdots\geqslant\lambda_m\geqslant 0$，$\boldsymbol{U}=(\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_m)$ 是由特征值对应的单位特征向量所构成的正交矩阵。式(3-79)可展开为

$$\begin{aligned}\boldsymbol{B}_{m\times m}&=(\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_m)\begin{bmatrix}\lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_m\end{bmatrix}\begin{bmatrix}\boldsymbol{u}_1^{\mathrm{T}}\\ \boldsymbol{u}_2^{\mathrm{T}}\\ \vdots\\ \boldsymbol{u}_m^{\mathrm{T}}\end{bmatrix}\\&=\lambda_1\boldsymbol{u}_1\boldsymbol{u}_1^{\mathrm{T}}+\lambda_2\boldsymbol{u}_2\boldsymbol{u}_2^{\mathrm{T}}+\cdots+\lambda_m\boldsymbol{u}_m\boldsymbol{u}_m^{\mathrm{T}}\end{aligned}$$

可以看出，特征值为零的项可直接丢弃，特征值小的项对样本内积矩阵 $\boldsymbol{B}_{m\times m}$ 的影响也比较小。如需降到 d' 维，则保留前 d' 个最大特征值及其对应的特征向量：

$$\boldsymbol{B}'_{m\times m}=(\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_{d'})\begin{bmatrix}\lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_{d'}\end{bmatrix}\begin{bmatrix}\boldsymbol{u}_1^{\mathrm{T}}\\ \boldsymbol{u}_2^{\mathrm{T}}\\ \vdots\\ \boldsymbol{u}_{d'}^{\mathrm{T}}\end{bmatrix}\approx\boldsymbol{B}_{m\times m}$$

令 $\boldsymbol{U}'_{m\times d'}=(\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_{d'})$，$\boldsymbol{D}'_{d'\times d'}=\mathrm{diag}(\sqrt{\lambda_1},\sqrt{\lambda_2},\cdots,\sqrt{\lambda_{d'}})$，则

$$\boldsymbol{B}'_{m\times m}=\boldsymbol{U}'_{m\times d'}(\boldsymbol{D}'_{d'\times d'}\boldsymbol{D}'_{d'\times d'})(\boldsymbol{U}'_{m\times d'})^{\mathrm{T}}=(\boldsymbol{U}'_{m\times d'}\boldsymbol{D}'_{d'\times d'})(\boldsymbol{U}'_{m\times d'}\boldsymbol{D}'_{d'\times d'})^{\mathrm{T}}$$

又

$$\boldsymbol{B}'_{m\times m}\approx\boldsymbol{B}_{m\times m}=(\boldsymbol{Z}_{d'\times m})^{\mathrm{T}}\boldsymbol{Z}_{d'\times m}$$

则降到 d' 维后的低维特征矩阵近似为

$$\boldsymbol{Z}_{d'\times m}=(\boldsymbol{U}'_{m\times d'}\boldsymbol{D}'_{d'\times d'})^{\mathrm{T}}=\boldsymbol{D}'_{d'\times d'}(\boldsymbol{U}'_{m\times d'})^{\mathrm{T}}\tag{3-80}$$

矩阵 $\boldsymbol{Z}_{d'\times m}$ 的列向量 $\boldsymbol{z}_i$ 就是高维特征 $\boldsymbol{x}_i$ 经 MDS 降维后的低维特征，$i=1,2,\cdots,m$。

2. 流形降维方法

如果高维空间的样本特征具有某种分布结构，例如特征分布在某个低维曲线或曲面上，这时度量样本点距离就不能用欧氏距离(即直线距离，参见图 3-27)，而应改用非欧形式的距离，例如曲线距离或曲面上的**测地距离**(geodesic distance)。如何计算高维特征空间中的非欧距离呢？

机器学习借鉴了数学和物理学中**流形**(manifold)的概念，认为高维特征空间虽然整体上与欧氏空间不同，但局部仍具有欧氏空间的性质。流形就是局部具有欧氏空间性质的空间，术语称这样的空间在局部与欧氏空间"同胚"。可以借助流形的思想，在降维时尽量保持样本在高维空间的局部结构，使得降维后对低维数据进行可视化时能观察到这些结构。

1) Isomap

给定训练集 $D_{\text{train}}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\}$，或 $D_{\text{train}}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$，其中 $\boldsymbol{x}_i$ 是 d 维样本特征。借助流形的思想，可以在每个样本点 $\boldsymbol{x}_i$ 的邻域内利用欧氏距离找出其 k 个近邻样本，为样本集建立起一个邻接图(或邻接矩阵)。邻接图上的每个顶点对应一个样本点，边表示两个顶点间存在邻接关系，其权重则表示这两个邻接点之间的欧氏距离。将图上顶点之间的最短路径作为样本点间的非欧距离，这样就把非欧距离的计算问题转换为一个求最短路径问题，可以使用图论算法(例如 Dijkstra 算法)进行求解。

利用上述方法，最终可求得高维特征空间的距离矩阵 $\boldsymbol{D}$。有了距离矩阵 $\boldsymbol{D}$，下一步就可以使用 MDS 降维方法将 d 维样本特征降到 d' 维，$d'\leqslant d$。这种借助流形求解距离矩阵，然后进行降维的方法被称为**等度量映射**(Isometric Mapping，Isomap)。

2) LLE 和 SNE

高维空间的样本特征具有某种分布结构，可以将这种分布结构看作被嵌入高维空间的低维结构，降维时应当予以保持。"分布结构"可以用样本点之间的"距离"来描述，例如 MDS、Isomap 认为降维时应当保持"距离"，也可以使用其他概念来描述这种分布结构。

如果使用样本点之间的"线性关系"来描述分布结构，则降维时应尽量保持这种线性关系，由此就产生了**局部线性嵌入**(Locally Linear Embedding，LLE)降维方法。假设原始 d 维样本点 $\boldsymbol{x}_i$ 可以表示成其 k 个近邻点的线性组合(参见图 3-27)，

$$\boldsymbol{x}_i=\omega_{i1}\boldsymbol{x}_1+\omega_{i2}\boldsymbol{x}_2+\cdots+\omega_{ik}\boldsymbol{x}_k$$

LLE 将 d 维样本特征降到 d' 维($d'\leqslant d$)后，希望所对应的低维特征点 $\boldsymbol{z}_i$ 能尽量保持这种线性关系，即

$$\boldsymbol{z}_i=\omega_{i1}\boldsymbol{z}_1+\omega_{i2}\boldsymbol{z}_2+\cdots+\omega_{ik}\boldsymbol{z}_k$$

采用 LLE 降维，低维特征 $\boldsymbol{z}_i$ 能尽量保持其高维特征 $\boldsymbol{x}_i$ 存在的线性关系(即分布结构)。可视化低维特征 $\boldsymbol{z}_i$(通常取二维或三维)，就能看到其高维特征 $\boldsymbol{x}_i$ 的分布结构，这非常有利于数据的分析。

也可以使用样本点之间的"条件概率"来描述分布结构，降维时尽量保持这种概率分布，由此就产生了**随机近邻嵌入**(Stochastic Neighbor Embedding，SNE)降维方法。假设将样

本点看作随机变量，原始 d 维样本点 $\boldsymbol{x}_i$ 邻域内近邻点 $\boldsymbol{x}_k$ 出现的条件概率 $p(\boldsymbol{x}_k \mid \boldsymbol{x}_i)$ 服从高斯分布。SNE 将 d 维样本特征降到 $d'(d' \leqslant d)$ 维后，希望所对应的低维特征点 $\boldsymbol{z}_i$ 能尽量保持这种概率分布，即

$$p(\boldsymbol{z}_k \mid \boldsymbol{z}_i) = p(\boldsymbol{x}_k \mid \boldsymbol{x}_i)$$

如果将低维空间概率分布 $p(\boldsymbol{z}_k \mid \boldsymbol{z}_i)$ 由高斯分布改为 t-分布，则这样的 SNE 就被称为 t-SNE。

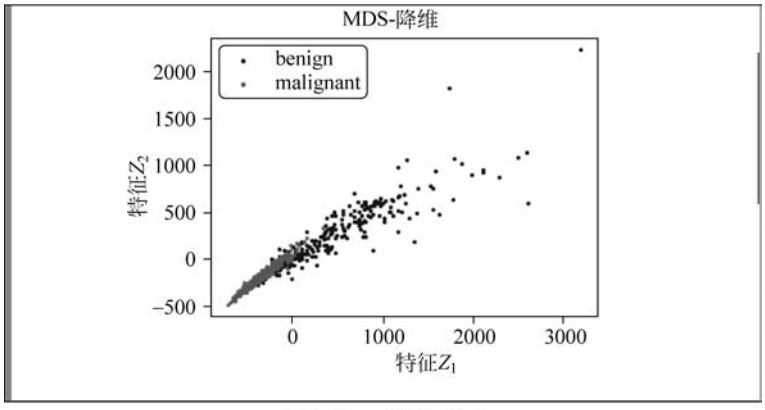
3. 使用 scikit-learn 库中的非线性降维模型

scikit-learn 库将常用的非线性降维模型封装成类，并将它们集中存放在 sklearn.manifold 模块中，其中包括 MDS、Isomap、LocallyLinearEmbedding 和 TSNE，共 4 个类。这些类都实现了降维模型的学习算法 **fit**()和降维转换算法 **transform**()。

图 3-28(a)给出了使用 MDS 类对乳腺癌诊断数据集进行特征降维的示例代码。其中，数据集使用的是 3.1.2 节加载的原始乳腺癌诊断数据集，将其从 30 维降到 2 维；然后使用散点图进行可视化(见图 3-28(b))，其中 benign 表示良性，malignant 表示恶性。

```
In [16]: from sklearn.manifold import MDS
         mds = MDS(n_components=2)
         X_transformed = mds.fit_transform(X)
         L = X_transformed.shape[0]
         z01 = []; z02 = []
         z11 = []; z12 = []
         for i in range(L):
             if (Y[i] == 0):
                 z01.append(X_transformed[i][0])
                 z02.append(X_transformed[i][1])
             else:
                 z11.append(X_transformed[i][0])
                 z12.append(X_transformed[i][1])

         fig = plt.figure( figsize=(4, 3), dpi=100 )
         plt.rcParams['font.sans-serif'] = ['SimHei']
         plt.rcParams['axes.unicode_minus'] = False

         plt.scatter(z01, z02, s=2, label="benign")
         plt.scatter(z11, z12, s=2, label="malignant")
         plt.title("MDS - 降维")
         plt.xlabel("特征 - $Z_1$"); plt.ylabel("特征 - $Z_2$")
         plt.legend(loc='best')
         plt.show()
```

(a) 示例代码



(b) 降维后的二维特征散点图

图 3-28　使用 MDS 类对乳腺癌诊断数据集进行特征降维

3.5 本章习题

一、单选题

1. 下列关于分类问题的描述中,错误的是(　　)。
   A. 给定样本特征,将其划归某一类别,这就是分类问题,或称为识别问题
   B. 分类问题必须基于特征的概率分布建立判别函数
   C. 机器学习将分类所用的判别函数称作分类器
   D. 贝叶斯决策是统计决策中解决分类问题的基本方法
2. 在已知概率分布的情况下,分类错误率最小的分类器是(　　)。
   A. 贝叶斯　　B. 朴素贝叶斯
   C. k 近邻　　D. 决策树
3. 给定特征 X,贝叶斯决策是基于下列概率中的(　　)最大来判定类别 Y 的。
   A. $P(X)$　　B. $P(Y)$
   C. $P(X|Y)$　　D. $P(Y|X)$
4. 下列关于朴素贝叶斯的描述中,错误的是(　　)。
   A. 朴素贝叶斯是针对高维特征联合概率分布难以估计问题而提出的
   B. 朴素贝叶斯假设各特征项之间相互独立
   C. 朴素贝叶斯假设各特征项与类别之间相互独立
   D. 设计朴素贝叶斯分类器需估计给定类别条件下各特征项的概率分布
5. 给定类别 Y,如果特征 X 有 3 个不同选项,则 $P(X|Y)$服从下列概率分布中的(　　)。
   A. 0-1 分布　　B. 伯努利分布
   C. 多项分布　　D. 高斯分布
6. 下列关于牛顿法的描述中,错误的是(　　)。
   A. 牛顿法是一种数值迭代算法
   B. 应用牛顿法的一个前提条件是目标函数二阶可导
   C. 牛顿法利用二阶导数确定下一迭代点的搜索方向
   D. 牛顿法的收敛速度比梯度下降法快
7. 将逻辑斯蒂回归应用于多分类,其后验概率的函数形式为(　　)。
   A. logistic 函数　　B. softmax 函数
   C. sigmoid 函数　　D. Gaussian 函数
8. 下列关于 k 近邻分类器的描述中,错误的是(　　)。
   A. k 近邻分类器直接基于训练集实例进行分类,没有显式的学习过程
   B. 给定新样本 $\boldsymbol{x}$,k 近邻分类器会查找训练集中的 k 个最近邻并统计它们的类别
   C. k 近邻分类器采用简单多数表决规则判定新样本的类别
   D. k 近邻分类器必须使用欧氏距离来度量样本点间的相似度
9. 下列关于线性判别分析的描述中,错误的是(　　)。
   A. 对于二分类问题,线性判别分析的核心思想是设法将高维特征压缩至一维
   B. 线性判别分析选择投影方向的标准是“类内方差最大,类间方差最小”

C. 线性判别分析在将高维特征压缩到一维之后再基于一维特征设计分类器

D. 线性判别分析更多地用于特征降维

10. 下列关于决策树分类器的描述中，错误的是（　　）。

A. 决策树分类器按照特征有效性，先使用主要特征，后使用次要特征

B. 决策树的分类决策过程分多步完成，每步选用一个特征项

C. 决策树选择特征项的标准是“所划分子集纯度越高，则特征项越有效”

D. 使用训练集建立决策树的过程是一个演绎推理的过程

11. 决策树模型的 3 种特征选择准则中不包括（　　）。

A. ID3　　B. C4.5

C. CART　　D. MSE

12. 二分类模型的评价指标中不包括（　　）。

A. 正确率　　B. 精确率

C. 召回率　　D. R 方

13. 设 $\boldsymbol{U}_{d\times d}$ 为正交矩阵，则下列等式不成立的是（　　）。

A. $\boldsymbol{U}=\boldsymbol{U}^{\mathrm{T}}$　　B. $\boldsymbol{U}^{-1}=\boldsymbol{U}^{\mathrm{T}}$

C. $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}=\boldsymbol{I}$　　D. $\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}}=\boldsymbol{I}$

14. 下列方法中属于线性特征降维方法的是（　　）。

A. 主成分分析　　B. 等度量映射

C. 局部线性嵌入　　D. 多维缩放

15. 下列降维方法中能够保证降维后样本点之间的欧氏距离不变的是（　　）。

A. 主成分分析　　B. 等度量映射

C. 局部线性嵌入　　D. 多维缩放

二、讨论题

1. 假设，记表 3-1 中的分类特征“底色”为 X，分类目标“品种”为 Y，根据表 3-1 的样本数据分别计算特征概率分布 $P(X)$、类别概率分布 $P(Y)$、特征条件概率 $P(X|Y)$，然后使用贝叶斯公式计算出类别条件概率 $P(Y|X)$。

2. 假设表 3-1 中的分类特征“底色”“口感”相互独立且分别记为 X_1 和 X_2，再记分类目标“品种”为 Y，根据表 3-1 的样本数据计算特征条件概率 $P(X_1,X_2|Y)$。

3. 假设随机变量 X 服从正态分布，给定样本数据集 $\{x_1,x_2,\cdots,x_n\}$，使用极大似然估计求解其均值 μ 和方差 σ^2。

4. 如果一元函数 $f(x)$ 二阶可导，则可使用牛顿法求解其极值点，试推导其第 k 次迭代时 x^{k-1} 到 x^k 的迭代公式。

5. 给定样本数据集 $D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\}$，其中 $\boldsymbol{x}_i$ 是 d 维样本特征，试写出样本特征的协方差矩阵。

6. 什么是矩阵的特征值、奇异值，并简述它们的应用。

三、编程实践题

使用 scikit-learn 库提供的鸢尾花数据集（iris plants dataset）设计一个逻辑斯蒂回归分类器。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) 使用函数 sklearn.model_selection.train_test_split()将数据集按 8:2 的比例拆分成训练集和测试集；

(4) 使用 sklearn.linear_mode.LogisticRegression 类建立逻辑斯蒂回归模型，并使用训练集进行训练；

(5) 分别计算模型在训练集和测试集上的平均正确率。