第 5 章

CHAPTER 5

朴素贝叶斯分类器

朴素贝叶斯分类器是一种有监督的统计学过滤器，在垃圾邮件过滤、信息检索等领域十分常用。通过本章的介绍，读者将会看到朴素贝叶斯分类器因何得名、其与贝叶斯公式的联系，以及其与极大似然估计的关系。

5.1 极大似然估计

对于工厂生产的某一批灯泡，质检部门希望检测其合格率。设 m 表示产品总数，随机变量 $X_i \in \{0,1\}$ 表示编号为 i 的产品是否合格。由于这些产品都是同一批生产的，不妨假设

$$X_1, X_2, \cdots, X_m \overset{i.i.d.}{\sim} \text{Bern}(p) \tag{5-1}$$

其中，p 表示产品合格的概率，也就是质检部门希望得到的数据。根据经典概率模型有

$$p \approx \frac{1}{m}\sum_{i=1}^{m} X_i \tag{5-2}$$

但是式(5-2)为什么成立？这就需要使用极大似然估计来证明了。

极大似然估计的思想是：找到这样一个参数 p，它使所有随机变量的联合概率最大。例中，联合概率表示为

$$P(X_1 = x_1, X_2 = x_2, \cdots, X_m = x_m) = \prod_{i=1}^{m} P(X_i = x_i) = \prod_{i=1}^{m} p^{x_i}(1-p)^{1-x_i} \tag{5-3}$$

最大化联合概率等价于求

$$\begin{aligned} p^* &= \arg\max_p \log \prod_{i=1}^{m} p^{x_i}(1-p)^{1-x_i} \\ &= \arg\max_p \sum_{i=1}^{m} (x_i \log p + (1-x_i)\log(1-p)) \\ &= \arg\max_p m\log(1-p) + \log\frac{p}{1-p}\sum_{i=1}^{m} x_i \end{aligned} \tag{5-4}$$

根据微积分知识容易证明式(5-2)

$$p^{*}=\frac{1}{m}\sum_{i=1}^{m}X_i \tag{5-5}$$

形式化地说，已知整体的概率分布模型 $f(x;\theta)$，但是模型的参数 θ 未知时，可以使用极大似然估计来估计 θ 的值。这里的概率分布模型既可以是连续的(概率密度函数)也可以是离散的(概率质量函数)。假设在一次随机实验中，我们独立同分布地抽到了 m 个样本 x_1，x_2，…，x_m 组成的样本集合。似然函数，也就是联合概率分布

$$L(\theta)=f_m(x_1,x_2,\cdots,x_m;\theta)=\prod_{i=1}^{m}f(x_i;\theta) \tag{5-6}$$

表示当前样本集合出现的可能性。令似然函数 $L(\theta)$对参数 θ 的导数为 0，可以得到 θ 的最优解。但是运算中涉及乘法运算及乘法的求导等，往往计算上存在不便性。而对似然函数取对数并不影响似然函数的单调性，即

$$L(\theta_1)>L(\theta_2)\Rightarrow\log L(\theta_1)>\log L(\theta_2) \tag{5-7}$$

所以最大化对数似然函数

$$l(\theta)=\log L(\theta)=\log\prod_{i=1}^{m}p(x_i;\theta)=\sum_{i=1}^{m}\log(p(x_i;\theta)) \tag{5-8}$$

可以在保证最优解与似然函数相同的条件下，大大减少计算量。

极大似然估计通过求解参数 θ 使得 $f_N(x_1,x_2,\cdots,x_N;\theta)$最大，这是一种很朴素的思想：既然从总体中随机抽样得到了当前样本集合，那么当前样本集合出现的可能性极大。

5.2　朴素贝叶斯分类

在概率论中，贝叶斯公式的描述如下

$$P(Y_i\mid X)=\frac{P(X,Y_i)}{P(X)}=\frac{P(Y_i)P(X\mid Y_i)}{\sum_{j=1}^{K}P(Y_j)P(X\mid Y_j)} \tag{5-9}$$

其中 $Y_1,Y_2,\cdots,Y_K$ 为一个完备事件组，$P(Y_i)$称为先验概率，$P(Y_i\mid X)$称为后验概率。设 $X=(X^1,X^2,\cdots,X^n)$表示 n 维(离散)样本特征，$Y\in\{c_1,c_2,\cdots,c_K\}$表示样本类别。由于一个样本只能属于这 K 个类别中的一个，所以 $Y_1,Y_2,\cdots,Y_K$ 一定是完备的。

给定样本集合 $D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\}$，我们希望估计 $P(Y\mid X)$。根据贝叶斯公式，对于任意样本 $x=(x^1,x^2,\cdots,x^n)$，其标签为 c_k 的概率为

$$P(Y=c_k\mid X=x)=\frac{P(Y=c_k)P(X=x\mid Y=c_k)}{P(X=x)} \tag{5-10}$$

假设随机变量 $X^1,X^2,\cdots,X^n$ 相互独立，则有

$$\begin{aligned}P(X=x\mid Y=c_k)&=P(X^1=x^1,X^2=x^2,\cdots,X^n=x^n\mid Y=c_k)\\&=\prod_{i=1}^{n}P(X^i=x^i\mid Y=c_k)\end{aligned} \tag{5-11}$$

带入式(5-10)得

$$P(Y=c_k\mid X=x)=\frac{P(Y=c_k)\prod_{i=1}^{n}P(X^i=x^i\mid Y=c_k)}{P(X=x)} \tag{5-12}$$

在实际进行分类任务时，不需要计算出 $P(Y|X)$ 的精确值，只需要求出 k^* 即可

$$k^* = \arg\max_k P(Y=c_k \mid X=x) \tag{5-13}$$

不难看出，式(5-12)右侧的分母部分与 k 无关。因此

$$k^* = \arg\max_k P(Y=c_k)\prod_{i=1}^{n} P(X^i=x^i \mid Y=c_k) \tag{5-14}$$

式中所有项都可以用频率代替概率在样本集合上进行估计

$$\begin{cases} P(Y=c_k) \approx \dfrac{N_k}{m} \\ P(X^i=x^i \mid Y=c_k) \approx \dfrac{\sum\limits_{j=1}^{m} I\{x_j^i=x^i, y_j=c_k\}}{N_k} \end{cases} \tag{5-15}$$

其中，N_k 表示 D 中标签为 c_k 的样本数量。

5.3 拉普拉斯平滑

当样本集合不够大时，可能无法覆盖特征的所有可能取值。也就是说，可能存在某个 c_k 和 x^i 使

$$P(X^i=x^i \mid Y=c_k)=0 \tag{5-16}$$

此时，无论其他特征分量的取值为何，都一定有

$$P(Y=c_k)\prod_{i=1}^{n} P(X^i=x^i \mid Y=c_k)=0 \tag{5-17}$$

为了避免这样的问题，实际应用中常采用平滑处理。典型的平滑处理就是拉普拉斯平滑

$$\begin{cases} P(Y=c_k) \approx \dfrac{N_k+1}{m+K} \\ P(X^i=x^i \mid Y=c_k) \approx \dfrac{\sum\limits_{j=1}^{m} I\{x_j^i=x^i, y_j=c_k\}+1}{N_k+A_i} \end{cases} \tag{5-18}$$

其中，A_i 表示 X^i 的所有可能取值的个数。

基于上述讨论，完整的朴素贝叶斯分类器的算法描述见算法 5-1。

算法 5-1　朴素贝叶斯分类器

输入：样本集合 $D=\{(x_1,y_1),(x_2,y_2),\cdots,(x_m,y_m)\}$；待预测样本 x；样本标记的所有可能取值 $\{c_1,c_2,\cdots,c_K\}$；样本输入变量 X 的每个属性变量 X^i 的所有可能取值 $\{a_{i1},a_{i2},\cdots,a_{iA_i}\}$

输出：待预测样本 x 所属的类别

1. 计算标记为 c_k 的样本出现的概率：

$$P(Y=c_k)=\frac{N_k+1}{m+K},\quad k=1,2,\cdots,K$$

2. 计算标记为 c_k 的样本，其 X^i 分量的属性值为 a_{ip} 的概率

$$P(X^i = a_{ip} \mid Y = c_k) = \frac{\sum_{j=1}^{N_k} I(x_j^i = a_{ip}, y_j = c_k) + 1}{N_k + A_i}$$

3. 根据上面的估计值计算 x 属于所有 y_k 的概率值，并选择概率最大的作为输出

$$y = \arg\max_{k=1,2,\cdots,K} (P(Y = c_k \mid X = x))$$

Return

$$= \arg\max_{k=1,2,\cdots,K} \left(P(Y = c_k) \prod_{i=1}^{n} P(X^i = x^i \mid Y = c_k)\right)$$

5.4　朴素贝叶斯分类器的极大似然估计解释

朴素贝叶斯思想的本质是极大似然估计，$P(Y=c_k)$ 和 $P(X^i=x^i \mid Y=c_k)$ 是我们要估计的概率值。以 $P(Y=c_k)$ 为例，令 $\theta_k = P(Y=c_k)$，则似然函数为

$$L(\theta) = \prod_{i=1}^{m} P(Y = y_i) = \prod_{k=1}^{K} \theta_k^{N_k} \tag{5-19}$$

根据极大似然估计，求 θ 等价于求解下面的优化问题：

$$\begin{aligned} &\max_{\theta} \quad l(\theta) = \sum_{k=1}^{K} N_k \ln\theta_k \\ &\text{s.t.} \quad \sum_{k=1}^{K} \theta_k = 1 \end{aligned} \tag{5-20}$$

使用拉格朗日乘子法求解。首先构造拉格朗日乘数为

$$\mathrm{Lag}(\theta) = \sum_{k=1}^{K} N_k \ln\theta_k + \lambda\left(\sum_{k=1}^{K} \theta_k - 1\right) \tag{5-21}$$

令拉格朗日函数对 θ_k 的偏导为 0，有

$$\frac{\partial \mathrm{Lag}}{\partial \theta_k} = \frac{N_k}{\theta_k} + \lambda = 0 \Rightarrow N_k = -\lambda\theta_k \tag{5-22}$$

于是

$$\sum_{k=1}^{K} N_k = -\lambda\left(\sum_{k=1}^{K} \theta_k\right) = -\lambda \tag{5-23}$$

解得

$$\begin{cases} \lambda = -m \\ \theta_k = \dfrac{N_k}{\lambda} = \dfrac{N_k}{m} \end{cases} \tag{5-24}$$

这样便得到了 $P(Y=c_k)$ 的极大似然估计。对 $P(X^i=x^i \mid Y=c_k)$ 的极大似然估计求解过程类似，留给读者自行推导。

5.5 实例：基于朴素贝叶斯实现垃圾短信分类

本节以一个例子来阐述朴素贝叶斯分类器在垃圾短信分类中的应用。SMS Spam Collection Data Set是一个垃圾短信分类数据集，包含了5574条短信，其中有747条垃圾短信。数据集以纯文本的形式存储，其中每行对应于一条短信。每行的第一个单词是spam或ham，表示该行的短信是不是垃圾短信。随后记录了短信的内容，内容和标签之间以制表符分隔。

该数据集没有收录进sklearn.datasets，所以我们需要自行加载，如代码清单5-1所示。

代码清单5-1 加载SMS垃圾短信数据集

```
with open('./SMSSpamCollection.txt', 'r', encoding = 'utf8') as f:
    sms = [line.split('\t') for line in f]
y, x = zip(*sms)
```

加载完成后，x 和 y 分别是长为5574的字符串列表。其中 y 的每个元素只可能是spam或ham，分别表示垃圾短信和正常短信。x 的每个元素表示对应短信的内容。在训练贝叶斯分类器以前，需要首先将 x 和 y 转换成适于训练的数值表示形式，这个过程称为特征提取，如代码清单5-2所示。

代码清单5-2 SMS垃圾短信数据集特征提取

```
from sklearn.feature_extraction.text import CountVectorizer as CV
from sklearn.model_selection import train_test_split
y = [label == 'spam' for label in y]
x_train, x_test, y_train, y_test = train_test_split(x, y)
counter = CV(token_pattern = '[a-zA-Z]{2,}')
x_train = counter.fit_transform(x_train)
x_test = counter.transform(x_test)
```

特征提取的结果存储在(x_train, y_train)以及(x_test, y_test)中。其中x_train和x_test分别是4180×6595和1394×6595的稀疏矩阵。不难看出，两个矩阵的行数之和等于5574，也就是完整数据集的大小。因此两个矩阵的每行应该代表一个样例，那么每列代表什么呢？查看counter的vocabulary_属性就会发现，其大小恰好是6595，也就是所有短信中出现过的不同单词的个数。例如短信“Go until jurong point, go”中一共有5个单词，但是由于go出现了两次，所以不同单词的个数只有4个。x_train和x_test中的第(i,j)个元素就表示第 j 个单词在第 i 条短信中出现的次数。

代码清单5-3 朴素贝叶斯分类器的构造与训练

```
from sklearn.naive_bayes import MultinomialNB as NB
model = NB()
```

```
model.fit(x_train, y_train)
train_score = model.score(x_train, y_train)
test_score = model.score(x_test, y_test)
print("train score:", train_score)
print("test score:", test_score)
```

最后就是朴素贝叶斯分类器的构造与训练，如代码清单 5-3 所示。我们首先基于训练集训练朴素贝叶斯分类器，然后分别在训练集和测试集上进行测试。测试结果显示，模型在训练集上的分类准确率达到 0.993，在测试集上的分类准确率为 0.986。可见朴素贝叶斯分类器达到了良好的分类效果。

朴素贝叶斯分类器假设样本特征之间相互独立。这一假设非常强，以至于几乎不可能满足。但是在实际应用中，朴素贝叶斯分类器往往表现良好，特别是在垃圾邮件过滤、信息检索等场景下往往表现优异。

第 6 章

CHAPTER 6

支持向量机

支持向量机[5]是一种功能强大的机器学习算法。典型的支持向量机是一种二分类算法，其基本思想是：对于空间中的样本点集合，可用一个超平面将样本点分成两部分，一部分属于正类，一部分属于负类。支持向量机的优化目标就是找到这样一个超平面，使得空间中距离超平面最近的点到超平面的几何间隔尽可能大，这些点称为支持向量。

6.1 最大间隔及超平面

给定样本集合 $D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_m,y_m)\}$，设 $y_i\in\{-1,+1\}$。设输入空间中的一个超平面表示为

$$\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}+b=0 \tag{6-1}$$

其中，$\boldsymbol{\omega}$ 称为法向量，决定超平面的方向；b 为偏置，决定超平面的位置。根据点到直线距离公式的扩展，空间中一点 $\boldsymbol{x}_i$ 到超平面$\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}+b=0$ 的欧氏距离为

$$r_i=\frac{|\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i+b|}{\|\boldsymbol{\omega}\|} \tag{6-2}$$

如果超平面能将正确所有样本点分类，则点到直线的距离可以写成分段函数的形式为

$$r_i=\begin{cases}\dfrac{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i+b}{\|\boldsymbol{\omega}\|}, & y_i=+1\\[2ex] -\dfrac{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i+b}{\|\boldsymbol{\omega}\|}, & y_i=-1\end{cases} \tag{6-3}$$

式(6-3)也可用一个方程来表示

$$r_i=\frac{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i+b}{\|\boldsymbol{\omega}\|}y_i \tag{6-4}$$

6.2 线性可分支持向量机

线性可分支持向量机的目标是，通过求解$\boldsymbol{\omega}$ 和 b 找到一个超平面$\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}+b=0$。在保证超平面能够正确将样本进行分类的同时，使得距离超平面最近的点到超平面的距离尽可能

大。这是一个典型的带有约束条件的优化问题，约束条件是超平面能将样本集合中的点正确分类。将距离超平面最近的点与超平面之间的距离记为

$$r = \min_{i=1,2,\cdots,m} r_i \tag{6-5}$$

最优化问题可写做

$$\begin{aligned} &\max_{\boldsymbol{\omega},b} \quad r \\ &\text{s.t.} \quad r_i = \frac{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i + b}{\|\boldsymbol{\omega}\|} y_i \geqslant r, \quad i = 1,2,\cdots,m \end{aligned} \tag{6-6}$$

对于超平面$\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}+b=0$，可以为等式两边同时乘以相同的不为0的实数，超平面不发生变化。所以对任一支持向量$\boldsymbol{x}^*$可以通过对超平面公式进行缩放使得$(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}^*+b)y^*=1$，$\boldsymbol{x}^*$到超平面的距离可表示为$\frac{1}{\|\boldsymbol{\omega}\|}$，则优化问题可写作

$$\begin{aligned} &\max_{\boldsymbol{\omega},b} \quad \frac{1}{\|\boldsymbol{\omega}\|} \\ &\text{s.t.} \quad r_i = \frac{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i + b}{\|\boldsymbol{\omega}\|} y_i \geqslant \frac{1}{\|\boldsymbol{\omega}\|}, \quad i = 1,2,\cdots,m \end{aligned} \tag{6-7}$$

最大化$\frac{1}{\|\boldsymbol{\omega}\|}$也即最小化$\frac{1}{2}\|\boldsymbol{\omega}\|^2$，使用$\frac{1}{2}\|\boldsymbol{\omega}\|^2$作为优化目标是为了计算方便。

$$\begin{aligned} &\min_{\boldsymbol{\omega},b} \quad \frac{1}{2}\|\boldsymbol{\omega}\|^2 \\ &\text{s.t.} \quad r_i = (\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i + b)y_i - 1 \geqslant 0, \quad i = 1,2,\cdots,m \end{aligned} \tag{6-8}$$

可以证明，支持向量机的超平面存在唯一性，本书略。支持向量机中的支持向量至少为两个，由超平面分割成的正负两个区域至少各存在一个支持向量，且超平面的位置仅由这些支持向量决定，与支持向量外的其他样本点无关。在这两个区域，过支持向量，可以分别做一个与支持向量机分割超平面平行的平面H_1和H_2，两个超平面之间的距离为$\frac{2}{\|\boldsymbol{\omega}\|}$，如图6-1所示。

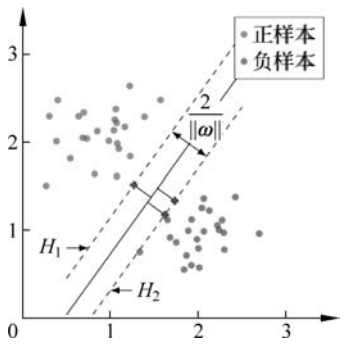

图6-1　线性可分支持向量机一(见彩插)

在感知机模型中，优化的目标是：在满足模型能够正确分类的约束条件下，使得样本集合中的所有点到分割超平面的距离最小，这样的超平面可能存在无数个。一个简单的例子，假如二维空间中样本集合中正负样本个数点均为一个，那么垂直于两者所连直线，且位于两者之间的所有直线都将是符合条件的解。由于优化目标不同，造成解的个数不同，这是支持向量机与感知机模型很大的区别。

式(6-8)中的最优化问题，可使用拉格朗日乘子法进行求解。拉格朗日函数为

$$\mathrm{Lag}(\boldsymbol{\omega},b,\boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{\omega}\|^2 + \sum_{i=1}^{m}\alpha_i(1-(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}_i+b)y_i) \tag{6-9}$$

其中，$\boldsymbol{\alpha}=(\alpha_1,\alpha_2,\cdots,\alpha_m)$，$\alpha_i \geqslant 0$表示拉格朗日乘子。令Lag对$\boldsymbol{\omega}$和$b$的偏导为0

$$\begin{cases}\dfrac{\partial \mathrm{Lag}(\boldsymbol{\omega},b,\boldsymbol{\alpha})}{\partial \boldsymbol{\omega}}=\boldsymbol{\omega}-\sum\limits_{i=1}^{m}\alpha_i y_i \boldsymbol{x}_i=0\\ \dfrac{\partial \mathrm{Lag}(\boldsymbol{\omega},b,\boldsymbol{\alpha})}{\partial b}=-\sum\limits_{i=1}^{m}\alpha_i y_i=0\end{cases} \tag{6-10}$$

解得

$$\begin{cases}\boldsymbol{\omega}=\sum\limits_{i=1}^{m}\alpha_i y_i \boldsymbol{x}_i\\ 0=\sum\limits_{i=1}^{m}\alpha_i y_i\end{cases} \tag{6-11}$$

将式(6-11)带入式(6-9)中,得到

$$\min_{\boldsymbol{\omega},b}\mathrm{Lag}(\boldsymbol{\omega},b,\boldsymbol{\alpha})=\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j \tag{6-12}$$

求$\min\limits_{\boldsymbol{\omega},b}\mathrm{Lag}(\boldsymbol{\omega},b,\boldsymbol{\alpha})$对$\boldsymbol{\alpha}$的极大,等价于求$-\min\limits_{\boldsymbol{\omega},b}\mathrm{Lag}(\boldsymbol{\omega},b,\boldsymbol{\alpha})$对$\boldsymbol{\alpha}$的极小。因此式(6-8)的对偶问题为

$$\begin{aligned}\min_{\boldsymbol{\alpha}}\quad & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j-\sum_{i=1}^{m}\alpha_i\\ \text{s.t.}\quad & \sum_{i=1}^{m}\alpha_i y_i=0\\ & \alpha_i\geqslant 0,\quad i=1,2,\cdots,m\end{aligned} \tag{6-13}$$

求解优化问题(6-13),即可得到$\boldsymbol{\alpha}^*=(\alpha_1^*,\alpha_2^*,\cdots,\alpha_m^*)$。根据KKT(Karush-Kuhn-Tucker)条件,$(\boldsymbol{\omega}^*,b^*)$是原始问题(6-8)的最优解,且$\boldsymbol{\alpha}^*$是对偶问题(6-13)的最优解的充要条件是:$(\boldsymbol{\omega}^*,b^*)$,$\boldsymbol{\alpha}^*$满足KKT条件,即

$$\begin{cases}\alpha_i^*\geqslant 0, & i=1,2,\cdots,m\\ (\boldsymbol{\omega}^{*\mathrm{T}}\boldsymbol{x}_i+b^*)y_i-1\geqslant 0, & i=1,2,\cdots,m\\ \alpha_i^*((\boldsymbol{\omega}^{*\mathrm{T}}\boldsymbol{x}_i+b^*)y_i-1)=0, & i=1,2,\cdots,m\end{cases} \tag{6-14}$$

由式(6-11)有

$$\boldsymbol{\omega}^*=\sum_{i=1}^{m}\alpha_i^* y_i \boldsymbol{x}_i \tag{6-15}$$

考察KKT条件的第三条,可以发现要么$\alpha_i^*=0$,要么$(\boldsymbol{\omega}^{*\mathrm{T}}\boldsymbol{x}_i+b^*)y_i-1=0$。假设$\alpha_j>0$,则必有$(\boldsymbol{\omega}^{*\mathrm{T}}\boldsymbol{x}_j+b^*)y_j-1=0$,于是

$$b^*=y_j-\sum_{i=1}^{m}\alpha_i^* y_i \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j \tag{6-16}$$

由此得到分割超平面

$$\boldsymbol{\omega}^*\boldsymbol{x}+b^*=0 \tag{6-17}$$

当$\alpha_i^*=0$时,即式(6-15)、式(6-16)与样本$(\boldsymbol{x}_i,y_i)$无关。也就是说,只有当$\alpha_i^*>0$时,样本$(\boldsymbol{x}_i,y_i)$才对最终的结果产生影响,此时样本的输入即为支持向量。求得支持向量机的参数后,即可根据式(6-18)判断任意样本的类别

$$f(\boldsymbol{x})=\mathrm{sgn}(\boldsymbol{\omega}^{*\mathrm{T}}\boldsymbol{x}+b^*) \tag{6-18}$$

6.3 线性支持向量机

线性可分支持向量机假设样本空间中的样本能够通过一个超平面分隔开来。但是生产环境中，我们获取到的数据往往存在噪声（正类中混入少量的负类样本，负类中混入少量的正类样本），从而使得数据变得线性不可分。这种情况就需要使用线性支持向量机求解了。

另一方面，即使样本集合线性可分，线性可分支持向量机给出的 H_1 和 H_2 之间的距离可能非常小。这种情况一般意味着模型的泛化能力降低，也就是产生了过拟合。因此我们希望 H_1 和 H_2 之间的距离尽可能大，这时同样可以使用线性支持向量机来允许部分样本点越过 H_1 和 H_2。

线性支持向量机在线性可分向量机的基础上引入了松弛变量 $\xi_i \geqslant 0$。对于样本点 $(\boldsymbol{x}_i, y_i)$，线性支持向量机允许部分样本落入越过超平面 H_1 或 H_2

$$(\boldsymbol{\omega}_i^{\mathrm{T}} \boldsymbol{x}_i + b) y_i \geqslant 1 - \xi_i \tag{6-19}$$

线性可分支持向量机中，要求所有样本都满足 $(\boldsymbol{\omega}_i^{\mathrm{T}} \boldsymbol{x}_i + b) y_i \geqslant 1$，此时 H_1 和 H_2 之间的距离 $\frac{2}{\| \boldsymbol{\omega} \|}$ 称为“硬间隔”。线性支持向量机中，$(\boldsymbol{\omega}^{\mathrm{T}} \boldsymbol{x}_i + b) y_i \geqslant 1 - \xi_i$ 允许部分样本越过超平面 H_1 或 H_2，此时 H_1 和 H_2 之间的距离 $\frac{2}{\| \boldsymbol{\omega} \|}$ 称为“软间隔”。需要注意的是，此时的支持向量不再仅仅包含位于 H_1 和 H_2 超平面上的点，还可能包含其他点。

对线性支持向量机进行优化时，我们希望“软间隔”尽量大，同时希望越过超平面 H_1 和 H_2 的样本尽可能不要远离这两个超平面，则优化的目标函数可写为

$$\frac{1}{2} \| \boldsymbol{\omega} \|^2 + C \sum_{i=1}^{m} \xi_i \tag{6-20}$$

其中，C 为惩罚系数。$\frac{1}{2} \| \boldsymbol{\omega} \|^2$ 控制最小间隔尽可能大，而 $\sum_{i=1}^{m} \xi_i$ 则控制越过超平面 H_1 或 H_2 的样本点离超平面尽量近，C 是对两者关系的权衡。线性支持向量机的优化问题可写为

$$\begin{aligned} \min_{\boldsymbol{\omega}, b, \boldsymbol{\xi}} \quad & \frac{1}{2} \| \boldsymbol{\omega} \|^2 + C \sum_{i=1}^{m} \xi_i \\ \text{s.t.} \quad & (\boldsymbol{\omega}^{\mathrm{T}} \boldsymbol{x}_i + b) y_i \geqslant 1 - \xi_i, \quad i = 1, 2, \cdots, m \\ & \xi_i \geqslant 0, \quad i = 1, 2, \cdots, m \end{aligned} \tag{6-21}$$

类似于线性可分支持向量机中的求解过程，式(6-21)的拉格朗日函数可写作

$$\begin{aligned} & \mathrm{Lag}(\boldsymbol{\omega}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) \\ = & \frac{1}{2} \| \boldsymbol{\omega} \|^2 + C \sum_{i=1}^{m} \xi_i + \sum_{i=1}^{m} \alpha_i (1 - \xi_i - (\boldsymbol{\omega}^{\mathrm{T}} x_i + b) y_i) - \sum_{i=1}^{m} \mu_i \xi_i \end{aligned} \tag{6-22}$$

其中

$$\begin{cases} \boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_m), & \alpha_i \geqslant 0 \\ \boldsymbol{\mu} = (\mu_1, \mu_2, \cdots, \mu_m), & \mu_i \geqslant 0 \end{cases} \tag{6-23}$$