6.1 间隔与支持向量机

6.1.1 问题的提出

给定训练样本集 $D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\ldots,(\boldsymbol{x}_m,y_m)\}$, $y_i\in\{-1,+1\}$, 分类学习最基本的想法就是基于训练集 D 在样本空间中找到一个划分超平面, 将不同类别的样本分开. 能将训练样本分开的划分超平面有很多. 如下图.

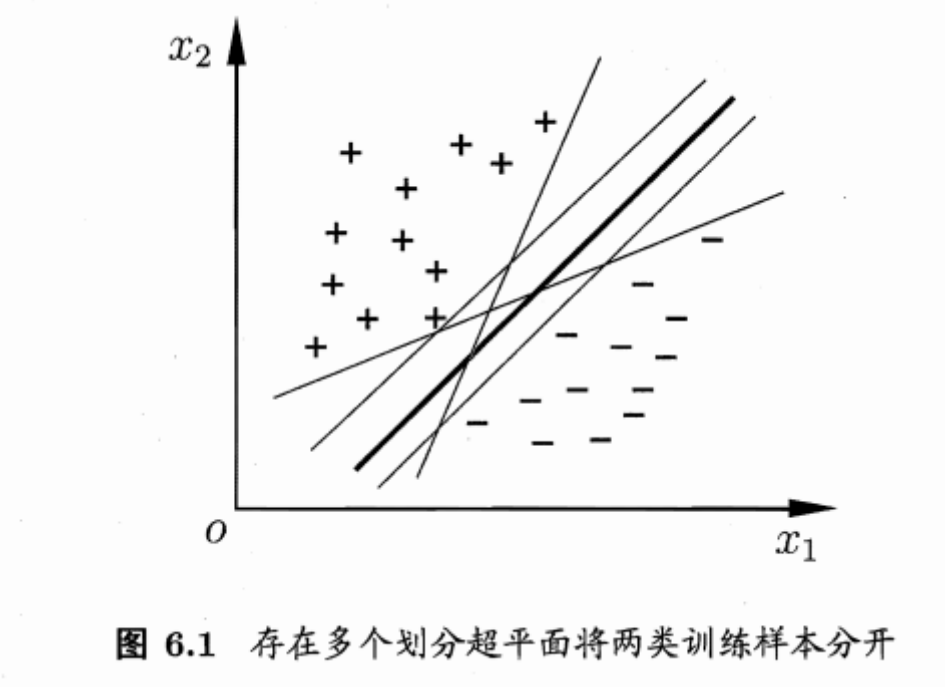


图 6.1 存在多个划分超平面将两类训练样本分开

可以看出, 粗线条的划分超平面"容忍性"最好. 换句话说, 这个划分超平面所产生的分类结果是最鲁棒的, 对未见示例的泛化能力最强.

6.1.2 数学表示

在样本空间中, 划分超平面可通过如下线性方程来描述:

$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b=0 \tag{6.1}$$

其中, $\boldsymbol{w}=(w_1;w_2;\ldots;w_d)$ 是法向量, 决定了超平面的方向; b 为位移项, 决定了超平面与原点之间的距离, 划分超平面可被法向量 $\boldsymbol{w}$ 和位移 b 确定. 将法向量记为 $(\boldsymbol{w},b)$.

样本空间中任意点 $\boldsymbol{x}$ 到超平面 $(\boldsymbol{w},b)$ 的距离可以写成

$$r=\frac{\left|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b\right|}{\|\boldsymbol{w}\|} \tag{6.2}$$

其中, $\|\boldsymbol{w}\|$ 是向量 $\boldsymbol{w}$ 的 L_2 范数, 也就是 $\|\boldsymbol{w}\|_2$ 的简写. 向量的 L_2 范数就是向量的模.

假设超平面 $(\boldsymbol{w},b)$ 能将训练样本正确分类, 即对于 $(\boldsymbol{x}_i,y_i)\in D$, 若 $y_i=+1$, 则有 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b>0$; 若 $y_i=-1$, 则有 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b<0$.

令

$$\begin{cases} \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b \geqslant +1, & y_i = +1 \\ \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b \leqslant -1, & y_i = -1 \end{cases} \tag{6.3}$$

> **注: 为何是令大于1和小于-1?**
>
> 因为这样标记方便我们将上述 (6.3) 变成如下的形式:
>
> $$y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) \geqslant +1,$$
>
> 正是因为标签为1和-1, 才方便我们将约束条件变成一个约束方程, 从而方便我们的计算

如图 (6.2)

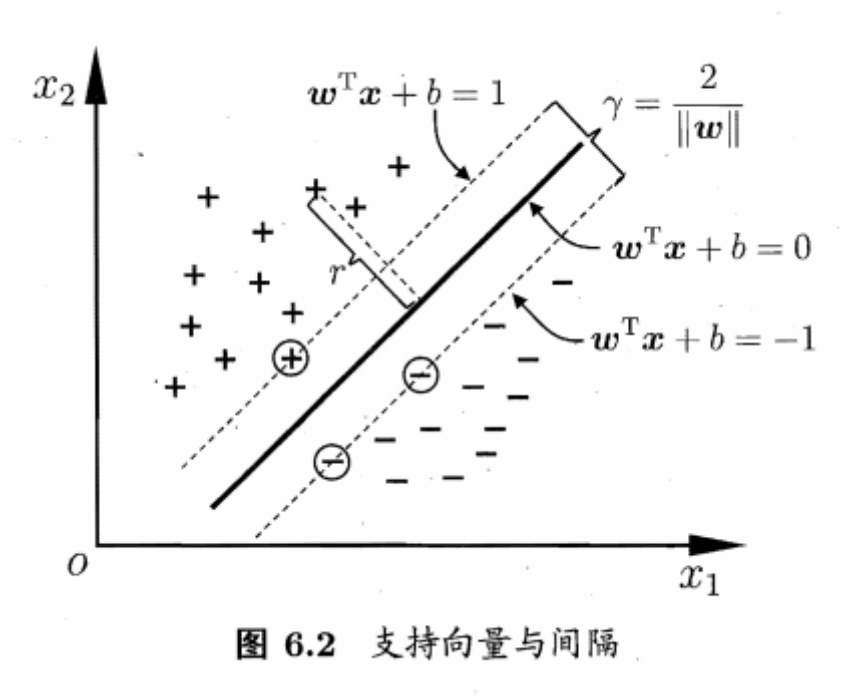


图 6.2　支持向量与间隔

如图 (6.2) , 距离超平面最近的这几个训练样本点使式 (6.3) 的等号成立, 它们被称为"支持向量机" (support vector) , 两个异类支持向量到超平面的距离之和为:

$$\gamma = \frac{2}{\|\boldsymbol{w}\|} \tag{6.4}$$

称为**"间隔"(margin)**

欲找到具有"最大间隔"(maximum margin)的划分超平面, 也就是找到能满足式(6.3)中约束的参数 $\boldsymbol{w}$ 和 b , 使得 γ 最大, 即

$$\begin{aligned} &\max_{\boldsymbol{w},b} \frac{2}{\|\boldsymbol{w}\|} \\ &\text{s.t. } y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \geqslant 1, \quad i = 1, 2, \ldots, m \end{aligned} \tag{6.5}$$

最大化间隔, 即最大化 $\|\boldsymbol{w}\|^{-1}$, 也就是等价于最小化 $\|\boldsymbol{w}\|^2$. 则 (6.5) 可重写为

$$\begin{aligned} &\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 \\ &\text{s.t. } y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \geqslant 1, \quad i = 1, 2, \ldots, m \end{aligned} \tag{6.6}$$

这就是**支持向量机**(support vector machine, 简称SVM)的**基本型**.

6.2 对偶问题

6.2.1 对偶问题

我们的目的就是根据式 (6.6) 来求得最大间隔划分超平面所对应的模型

$$f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b \tag{6.7}$$

其中 $\boldsymbol{w}$ 和 b 是模型参数. (6.6)本身是一个凸二次规划问题, 可以直接求解, 但是有更高效的方法.

对式 (6.6) 使用拉格朗日乘子法可得到其"对偶问题"(dual problem). 具体来说, 对式 (6.6) 的每条约束添加拉格朗日乘子 $\alpha_i \geqslant 0$, 则该问题的拉格朗日函数可以写成

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{m} \alpha_i \left(1 - y_i \left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right)\right) \tag{6.8}$$

> **注1: 关于拉格朗日对偶性详解**
>
> 令 $\theta(w) = \max_{\alpha_i \geq 0} L(w, b, \alpha)$, 易验证,当某个约束条件不满足时, 例如 $y_i \left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) < 1$, 那么显然有 $\theta(w) = \infty$ (只要令 $\alpha_i = \infty$ 即可). 而当所有约束条件都满足时, 则有 $\theta(w) = \frac{1}{2}\|\boldsymbol{w}\|^2$, 亦即最初要最小化的量.
>
> 因此, 在要求约束条件得到满足的情况下最小化 $\frac{1}{2}\|\boldsymbol{w}\|^2$, 实际上等价于直接最小化 (当然, 这里也有约束条件, 就是 $\alpha_i \geq 0, i = 1, \ldots, n$) , 因为如果约束条件没有得到满足, $\theta(w)$会等于无穷大, 自然不会是我们所要求的最小值
>
> 具体写出来, 目标函数变成了:
>
> $$\min_{\boldsymbol{w},b} \theta(\boldsymbol{w}) = \min_{\boldsymbol{w},b} \max_{\alpha_i \geq 0} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = p^*$$
>
> 把最小和最大的位置交换一下, 变成:
>
> $$\max_{\alpha_i \geq 0} \min_{\boldsymbol{w},b} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = d^*$$

其中, $\boldsymbol{\alpha} = (\alpha_1; \alpha_2; \ldots; \alpha_m)$. 令 $L(\boldsymbol{w}, b, \boldsymbol{\alpha})$ 对 $\boldsymbol{w}$ 和 b 的偏导为零可得

$$\boldsymbol{w} = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i \tag{6.9}$$

$$0 = \sum_{i=1}^{m} \alpha_i y_i \tag{6.10}$$

> **注2: (6.9)和 (6.10) 推导**
>
> 对 $\boldsymbol{w}$ 求偏导有:
>
> $\frac{\partial L}{\partial w} = \frac{1}{2} \times 2 \times \boldsymbol{w} + 0 - \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i - 0 = 0 \Longrightarrow \boldsymbol{w} = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i$ (1. $\frac{\partial \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i}{\partial \boldsymbol{w}} = \boldsymbol{x}_i$ 2. $\frac{\partial \boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}}{\partial \boldsymbol{w}} = 2\boldsymbol{w}$)
>
> $\frac{\partial L}{\partial b} = 0 + 0 - 0 - \sum_{i=1}^{m} \alpha_i y_i = 0 \Longrightarrow \sum_{i=1}^{m} \alpha_i y_i = 0$

将式 (6.9)带入式 (6.8) , 即可将 $L(\boldsymbol{w}, b, \boldsymbol{\alpha})$ 中的 $\boldsymbol{w}$ 和 b 消去, 再考虑式 (6.10) 的约束, 就得到式 **(6.6) 的对偶问题**

> **注3: 对偶式推导过程**
>
> $$\begin{aligned} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) &= \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \sum_{i=1}^{m}\alpha_i - \sum_{i=1}^{m}\alpha_i y_i \boldsymbol{w}^T\boldsymbol{x}_i - \sum_{i=1}^{m}\alpha_i y_i b \\ &= \frac{1}{2}\boldsymbol{w}^T\sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x}_i - \boldsymbol{w}^T\sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x}_i + \sum_{i=1}^{m}\alpha_i - b\sum_{i=1}^{m}\alpha_i y_i \\ &= -\frac{1}{2}\boldsymbol{w}^T\sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x}_i + \sum_{i=1}^{m}\alpha_i - b\sum_{i=1}^{m}\alpha_i y_i \end{aligned}$$
>
> 又 $0 = \sum_{i=1}^{m}\alpha_i y_i$, 进一步化简得到
>
> $$\begin{aligned} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) &= -\frac{1}{2}\boldsymbol{w}^T\sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x}_i + \sum_{i=1}^{m}\alpha_i \\ &= -\frac{1}{2}(\sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x}_i)^T(\sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x}_i) + \sum_{i=1}^{m}\alpha_i \\ &= -\frac{1}{2}\sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x}_i^T\sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x}_i + \sum_{i=1}^{m}\alpha_i \\ &= \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^T\boldsymbol{x}_j \end{aligned}$$
>
> 所以,$\max_{\alpha_i \geq 0}\min_{\boldsymbol{w},b} L(\boldsymbol{w}, b, \boldsymbol{\alpha})$, 最后 $L(\boldsymbol{w}, b, \boldsymbol{\alpha})$ 的只有参数 α_i (α_j 一样). 因此
>
> $\max_{\alpha_i \geq 0}\min_{\boldsymbol{w},b} L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \max_{\boldsymbol{\alpha}} \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j$

(6.6) 的对偶问题

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j \\ &s.t. \sum_{i=1}^{m}\alpha_i y_i = 0 \\ &\quad \alpha_i \geqslant 0, \quad i = 1, 2, \ldots, m \end{aligned} \tag{6.11}$$

解出 $\boldsymbol{\alpha}$ 后, 求出 $\boldsymbol{w}$ 和b 即可得到模型

$$\begin{aligned} f(\boldsymbol{x}) &= \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b \\ &= \sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x} + b \end{aligned} \tag{6.12}$$

6.2.2 KKT条件

从对偶问题 (6.11)解出的 α_i 是式 (6.8) 中的拉格朗日乘子, 它恰对应着训练样本$(\boldsymbol{x}_i, y_i)$. 注意到式 (6.6) 中有不等式约束, 因此上述过程需满足 KKT(Karush-Kuhn-Tucker) 条件, 即要求

$$\begin{cases} \alpha_i \geqslant 0 \\ y_i f(\boldsymbol{x}_i) - 1 \geqslant 0 \\ \alpha_i (y_i f(\boldsymbol{x}_i) - 1) = 0 \end{cases}$$

> **注4:** 关于拉格朗日对偶性以及KKT条件, 详细知识参见统计学习方法附录C,

具体含义:

于是, 对任意训练样本 $(\boldsymbol{x}_i, y_i)$, 总有 $\alpha_i = 0$ 或 $y_i f(\boldsymbol{x}_i) = 1$.

- 若 $\alpha_i = 0$, 则该样本将不会在式 (6.12) 的求和中出现, 也就不会对 $f(\boldsymbol{x})$ 有任何影响;
- 若 $\alpha_i > 0$, 则必有 $y_i f(\boldsymbol{x}_i) = 1$, 所对应的样本点位于最大间隔边界上, 是一个支持向量.
- 这显示出支持向量机的一个重要性质:训练完成后, 大部分的训练样本都不需保留, 最终模型仅与支持向量有关

6.2.3 SMO算法简介

1 SMO简介

不难发现, (6.11) 是一个二次规划问题, 可使用通用的二次规划算法来求解; 然而, 该问题的规模正比于训练样本数 , 这会在实际任务中造成很大的开销. 为了避开这个障碍, 人们通过利用问题本身的特性, 提出了很多高效算法, SMO (Sequential Minimal Optimization) 是其中一个著名的代表 [Platt , 1998].

SMO 的基本思路是先固定 α_i 之外的所有参数, 然后求 α_i 上的极值. 由于存在约束$\sum_{i=1}^{m} \alpha_i y_i = 0$, 若固定 α_i 之外的其他变量, 则 α_i 可由其他变量导出. 于是, SMO 每次选择两个变量 α_i 和 α_j , 并固定其他参数. 这样, 在参数初始化后, SMO 不断执行如下两个步骤直至收敛:

- 选取一对需更新的变量 α_i 和 α_j ;
- 固定 α_i 和 α_j 以外的参数, 求解式 (6.11) 获得更新后的 α_i 和 α_j

注意到只需选取的 α_i 和 α_j 中有一个不满足 KKT 条件 (6.13), 目标函数就会在选代后减小. 直观来看, KKT 条件违背的程度越大, 则变量更新后可能导致的目标函数值减幅越大. 于是, SMO 先选取违背 KKT 条件程度最大的变量. 第二个变量应选择一个使目标函数值减小最快的变量, 但由于比较各变量所对应的目标函数值减幅的复杂度过高, 因此 SMO 采用了一个启发式: **使选取的两变量所对应样本之间的间隔最大**. 一种直观的解释是, 这样的两个变量有很大的差别, 与对两个相似的变量进行更新相比, 对它们进行更新会带给目标函数值更大的变化.

2 SMO的数学简要解释

SMO 算法之所以高效, 恰由于在固定其他参数后, 仅优化两个参数的过程能做到非常高效.具体来说, 仅考虑 α_i 和 α_j 时, 式 (6.11) 中的约束可重写为

$$\alpha_i y_i + \alpha_j y_j = c, \quad \alpha_i \geqslant 0, \quad \alpha_j \geqslant 0 \tag{6.14}$$

其中,

$$c = -\sum_{k \neq i,j} \alpha_k y_k \tag{6.15}$$

是使 $\sum_{i=1}^{m} \alpha_i y_i = 0$ 成立的常数. 用

$$\alpha_i y_i + \alpha_j y_j = c \tag{6.16}$$

消去式 (6.11) 中的变量 α_j , 那么就得到一个关于 α_i 的单变量二次规划问题, 仅有的约束是 $\alpha_i \geqslant 0$. 不难发现, 这样的二次规划问题具有闭式解, 于是不必调用数值优化算法即可高效地计算出更新后的 α_i 和 α_j .

接下来确定便宜项 b . 注意到对任意支持向量 $(\boldsymbol{x}_s, y_s)$ (也即是$\alpha_i > 0$ 时, 为支持向量) , 都有 $y_s f(\boldsymbol{x}_s) = 1$, 根据式 (6.12) 进一步有:

$$y_s \left(\sum_{i \in S} \alpha_i y_i \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_s + b \right) = 1 \tag{6.17}$$

其中, $S = \{i | \alpha_i > 0, i = 1, 2, \ldots, m\}$ 为所有支持向量的下标集. 理论上, 可选任意支持向量并通过求解式 (6.17) 获得 b , 但现实任务中常采用一种更鲁棒的做法: 使用所有支持向量求解的平均值

$$b = \frac{1}{|S|} \sum_{s \in S} \left(\frac{1}{y_s} - \sum_{i \in S} \alpha_i y_i \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_s \right) \tag{6.18}$$

其中, $|S|$ 表示集合 S 中元素的数量.

6.3 核函数

6.3.1 线性不可分的情况

前面的都是假设训练样本是线性可分的, 即存在一个划分超平面能将训练样本正确分类.然而在现实任务中, 原始样本空间内也许并不存在一个能正确划分两类样本的超平面. 如图 6.3 中的" 异或 " 问题就不是线性可分的.

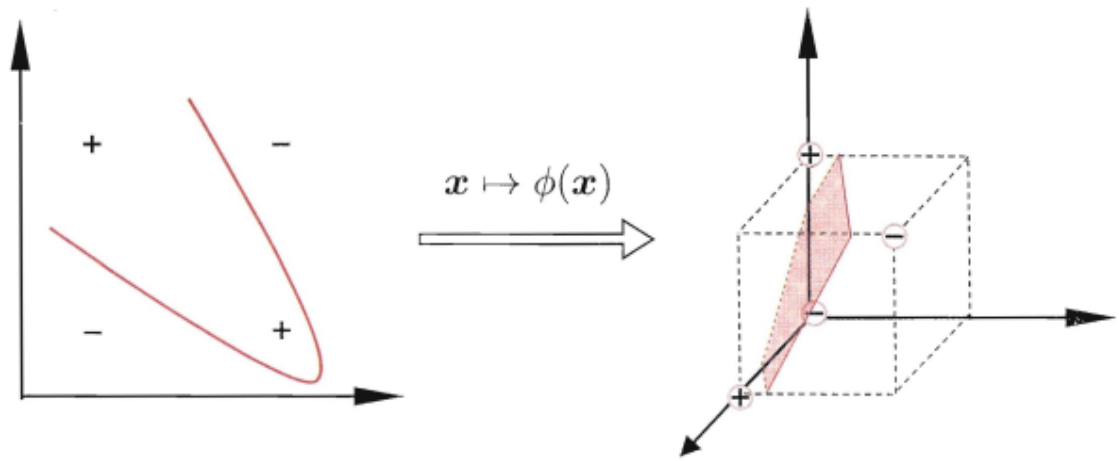


图 6.3 异或问题与非线性映射

对这样的问题, **可将样本从原始空间映射到一个更高维的特征空间, 使得样本在这个特征空间内线性可分.**

如在图 6.3 中, 若将原始的二维空间映射到一个合适的三维空间, 就能找到一个合适的划分超平面.

同时, **如果原始空间是有限维, 即属性数有限, 那么一定存在一个高维特征空间使样本可分**

6.3.2 数学表示

令 $\phi(\boldsymbol{x})$ 表示将 $\boldsymbol{x}$ 映射后的特征向量, 于是, 在特征空间中划分超平面所对应的模型可表示为:

$$f(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x})+b \tag{6.19}$$

其中, $\boldsymbol{w}$ 和 b 是模型参数. 类似于式 (6.6), 有

$$\begin{aligned}&\min_{\boldsymbol{w},b}\ \frac{1}{2}\|\boldsymbol{w}\|^2\\&\text{s.t. } y_i\left(\boldsymbol{w}^{\mathrm{T}}\phi\left(\boldsymbol{x}_i\right)+b\right)\geqslant 1,\quad i=1,2,\ldots,m\end{aligned} \tag{6.20}$$

其对偶问题是

$$\max_{\boldsymbol{\alpha}}\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j\phi(\boldsymbol{x}_i)^{\mathrm{T}}\phi\left(\boldsymbol{x}_j\right) \tag{6.21}$$

$$\begin{aligned}s.t.\ &\sum_{i=1}^{m}\alpha_i y_i=0\\&\alpha_i\geqslant 0,\quad i=1,2,\ldots,m\end{aligned}$$

$\phi(\boldsymbol{x}_i)^{\mathrm{T}}\phi\left(\boldsymbol{x}_j\right)$ 表示样本 $\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$ 映射到特征空间之后的内积. 计算较为困难. 为避开, 设想这样一个函数:

$$\kappa\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right)=\left\langle\phi\left(\boldsymbol{x}_i\right),\phi\left(\boldsymbol{x}_j\right)\right\rangle=\phi(\boldsymbol{x}_i)^{\mathrm{T}}\phi\left(\boldsymbol{x}_j\right) \tag{6.22}$$

即 $\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$ 在特征空间的内积等于它们在原始样本空间中通过函数 $\kappa(\cdot,\cdot)$ 计算的结果.

于是, 式 (6.21) 可以重写为:

$$\max_{\boldsymbol{\alpha}}\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j\kappa\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right) \tag{6.23}$$

$$\begin{aligned}s.t.\ &\sum_{i=1}^{m}\alpha_i y_i=0\\&\alpha_i\geqslant 0,\quad i=1,2,\ldots,m\end{aligned}$$

求解后即可得到

$$\begin{aligned}f(\boldsymbol{x})&=\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x})+b\\&=\sum_{i=1}^{m}\alpha_i y_i\phi(\boldsymbol{x}_i)^{\mathrm{T}}\phi(\boldsymbol{x})+b\\&=\sum_{i=1}^{m}\alpha_i y_i\kappa\left(\boldsymbol{x},\boldsymbol{x}_i\right)+b\end{aligned} \tag{6.24}$$

这里的函数 $\kappa(\cdot,\cdot)$ 就是**"核函数"**(kernel function)

式 (6.24) 显示出模型最优解可通过训练样本的函数展开, 这一展开式亦称 "支持向量展式"

6.3.3 有关核函数的定理

(一) 定理6.1 (核函数)

定理6.1 (核函数) 令 $\mathcal{X}$ 为输入空间, $\kappa(\cdot,\cdot)$ 是定义在 $\mathcal{X}\times\mathcal{X}$ 上的对称函数, 则 k 是核函数当且仅当对于任意数据 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_m\}$, "核矩阵" (kernel matrix) $\mathbf{K}$ 总是本正定的:

$$\mathbf{K}=\begin{bmatrix}\kappa(\boldsymbol{x}_1,\boldsymbol{x}_1) & \cdots & \kappa(\boldsymbol{x}_1,\boldsymbol{x}_j) & \cdots & \kappa(\boldsymbol{x}_1,\boldsymbol{x}_m)\\ \vdots & \ddots & \vdots & \ddots & \vdots\\ \kappa(\boldsymbol{x}_i,\boldsymbol{x}_1) & \cdots & \kappa(\boldsymbol{x}_i,\boldsymbol{x}_j) & \cdots & \kappa(\boldsymbol{x}_i,\boldsymbol{x}_m)\\ \vdots & \ddots & \vdots & \ddots & \vdots\\ \kappa(\boldsymbol{x}_m,\boldsymbol{x}_1) & \cdots & \kappa(\boldsymbol{x}_m,\boldsymbol{x}_j) & \cdots & \kappa(\boldsymbol{x}_m,\boldsymbol{x}_m)\end{bmatrix}$$

定理 6.1 表明, **只要一个对称函数所对应的核矩阵半正定, 它就能作为核函数使用.** 事实上, **对于一个半正定核矩阵, 总能找到一个与之对应的映射** ϕ . 换言之, 任何一个核函数都隐式地定义了一个称为"再生核希尔伯特空间" (Reproducing Kernel Hilbert Space, 简称 RKHS) 的特征空间.

(二) 常用的核函数及核函数的相关组合

表 6.1 列出了集中常用的核函数

表 6.1 常用核函数

名称	表达式	参数
线性核	$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)=\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j$	
多项式核	$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)=(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j)^d$	$d\geqslant 1$ 为多项式的次数
高斯核	$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)=\exp\left(-\frac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2}{2\sigma^2}\right)$	$\sigma>0$ 为高斯核的带宽(width)
拉普拉斯核	$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)=\exp\left(-\frac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|}{\sigma}\right)$	$\sigma>0$
Sigmoid 核	$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)=\tanh(\beta\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j+\theta)$	tanh 为双曲正切函数, $\beta>0,\theta<0$

此为, 还可以通过函数组合得到相关的核函数.

- 若 k_1 和 k_2 为核函数, 则对于任意正数 γ_1,γ_2 , 其线性组合

$$\gamma_1\kappa_1+\gamma_2\kappa_2 \tag{6.25}$$

  也是核函数

- 若 k_1 和 k_2 为核函数, 则核函数的直积

$$\kappa_1\otimes\kappa_2(\boldsymbol{x},\boldsymbol{z})=\kappa_1(\boldsymbol{x},\boldsymbol{z})\kappa_2(\boldsymbol{x},\boldsymbol{z}) \tag{6.26}$$

  也是核函数

- 若 k_1 为核函数, 则对于任意函数 $g(\boldsymbol{x})$,

$$\kappa(\boldsymbol{x},\boldsymbol{z})=g(\boldsymbol{x})\kappa_1(\boldsymbol{x},\boldsymbol{z})g(\boldsymbol{z})$$

  也是函数

6.4 软间隔与正则化

6.4.1 软间隔的概念

前面我们假定训练样本在**样本空间**或**特征空间**中是**线性可分**的, 即存在一个超平面能将不同类的样本完全划分开. 然而, 在现实任务中往往很难确定合适的核函数使得训练样本在特征空间中线性可分; 即使恰好找到了 某个核函数使训练集在特征空间中线性可分, 也很难断定这个貌似线性可分的结果不是由于**过拟合**所造成的.

缓解该问题的一个办法是**允许支持向量机在一些样本上出错**.为此, 要引入"**软间隔**" (soft margin)的概念, 如图6.4所示.

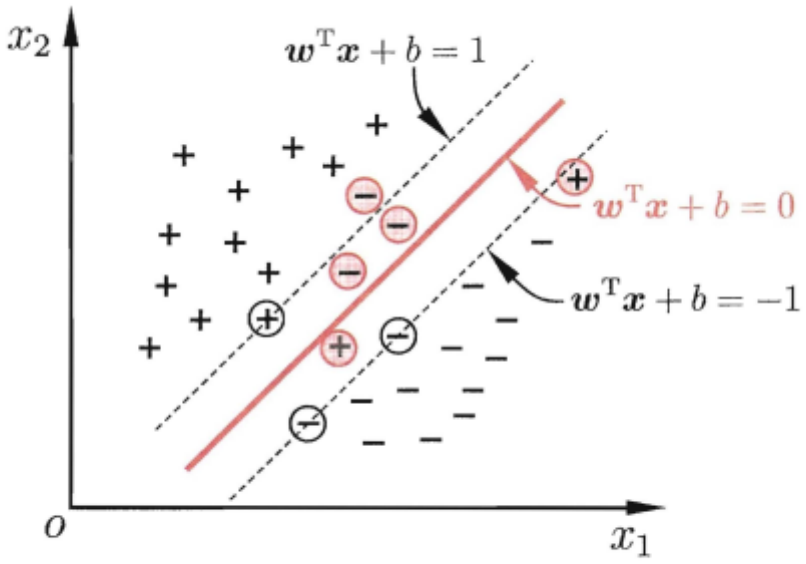


图 6.4 软间隔示意图. 红色圈出了一些不满足约束的样本.

6.4.2 数学表示

前面介绍的支持向量机形式是**要求所有样本均满足约束 (6.3)**, 即**所有样本都必须划分正确**, 这称为" **硬间隔**" (hard margin), 而**软间隔则是允许某些样本不满足约束**

$$y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\right)\geqslant 1 \tag{6.28}$$

在**最大化间隔的同时, 不满足约束的样本应尽可能的少**. 则**优化目标**可以写为

$$\min_{\boldsymbol{w},b}\frac{1}{2}\|\boldsymbol{w}\|^2+C\sum_{i=1}^{m}\ell_{0/1}\left(y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\right)-1\right) \tag{6.29}$$

其中 $C>0$ 是一个常数, $\ell_{0/1}$ 是 "0/1损失函数"

$$\ell_{0/1}(z)=\begin{cases}1, & \text{if } z<0\\ 0, & \text{otherwise}\end{cases} \tag{6.30}$$

显然, 当 C 为无穷大时, 式 (6.29) 迫使所有样本均满足约束 (6.28) , 于是式 (6.29) 等价于 式 (6.6) ; 当 C 取有限值时, 式 (6.29) 允许一些样本不满足约束.

然而, $\ell_{0/1}$ 非凸、非连续, 数学性质不太好, 使得式 (6.29) 不易直接求解. 人们通常用其他一些函数来代替 $\ell_{0/1}$, 称为"**替代损失**" (surrogate loss). 替代损失函数一般具有较好的数学性质, 如它们通常是凸的连续函数且是 $\ell_{0/1}$ 的上界. 图 6.5 给出了三种常用的替代损失函数:

- hinge 损失: $\ell_{\text{hinge}}(z)=\max(0,1-z)$ (6.31)
- 指数损失(exponential loss): $\ell_{exp}(z)=\exp(-z)$ (6.32)
- 对率损失(logistic loss): $\ell_{\exp}(z)=\exp(-z)$ (6.33)

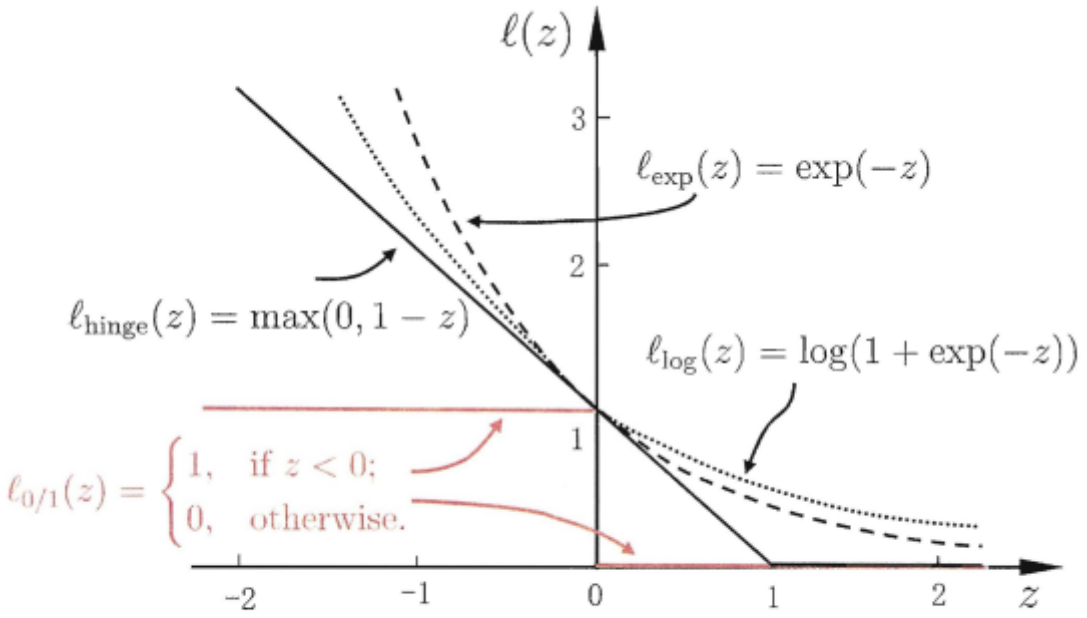


图 6.5 三种常见的替代损失函数: hinge损失、指数损失、对率损失

若采用hinge损失, 则式 (6.29) 变成

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} \max\left(0, 1 - y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right)\right) \tag{6.34}$$

> 注5: 同6.29一样的思路来理解

接着, 引入"**松弛变量**"(slack variable) $\xi_i \geqslant 0$, 可将式 (6.34) 重写为

$$\min_{\boldsymbol{w},b,\xi_i} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} \xi_i \tag{6.35}$$

$$s.t. \quad y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) \geqslant 1 - \xi_i$$

$$\xi_i \geqslant 0, i = 1, 2, \ldots, m$$

这就是常用的**"软间隔支持向量机"**的基本式

> 注6: 式(6.35)是式(6.34)的上限, 最小化式(6.35)的同时也会最化小式(6.34), 这是因为:
>
> 由式(6.35)的约束条件 $y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) \geqslant 1 - \xi_i$ 可得 $\xi_i \geqslant 1 - y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right)$, 再加上约束条件 $\xi_i \geqslant 0$, 即 $\xi_i \geqslant \max\left(0, 1 - y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right)\right)$, 因此式(6.35)是式(6.34)的上限

显然, 式 (6.35) 中每个样本都有一个对应的松弛变量, 用以表征该样本不满足约束 (6.28) 的程度.

与式 (6.6) 相似, 这还是一个二次规划问题. 于是, 类似于式 (6.8) , 通过拉格朗日乘子法可得到式 (6.35) 的拉格朗日函数

$$
\begin{aligned}
L(\boldsymbol{w}, b, \boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\mu}) =& \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m}\xi_i \\
&+ \sum_{i=1}^{m}\alpha_i\left(1 - \xi_i - y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right)\right) - \sum_{i=1}^{m}\mu_i\xi_i
\end{aligned} \tag{6.36}
$$

其中, $\alpha_i \geqslant 0, \mu_i \geqslant 0$ 是拉格朗日乘子.

> 注7: **拉格朗日基本形式**为
>
> $$\min_{x\in\mathbf{R}^n} f(x)$$
>
> $$s.t. \quad c_i(x) \leqslant 0, \quad i = 1, 2, \cdots, k$$
>
> $$h_j(x) = 0, \quad j = 1, 2, \cdots, l$$
>
> 引入拉格朗日函数有
>
> $$L(x, \alpha, \beta) = f(x) + \sum_{i=1}^{k}\alpha_i c_i(x) + \sum_{j=1}^{l}\beta_j h_j(x)$$

令 $L(\boldsymbol{w}, b, \boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\mu})$ 对 $\boldsymbol{w}, b, \xi_i$ 的偏导为零可得

$$\boldsymbol{w} = \sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x}_i \tag{6.37}$$

$$0 = \sum_{i=1}^{m}\alpha_i y_i \tag{6.38}$$

$$C = \alpha_i + \mu_i \tag{6.39}$$

将式 (6.37)-(6.39) 代入式 (6.36) 即可得到**软间隔支持向量机的对偶问题**

$$
\begin{aligned}
&\max_{\boldsymbol{\alpha}} \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j \\
&s.t. \sum_{i=1}^{m}\alpha_i y_i = 0 \\
&0 \leqslant \alpha_i \leqslant C, \quad i = 1, 2, \ldots, m
\end{aligned} \tag{6.40}
$$

软间隔的对偶问题式 (6.40) 与硬间隔下的对偶问题 (6.11) 对比可看出, 两者唯一的差别就再与对偶变量的约束不同: **前者是** $0 \leqslant \alpha_i \leqslant C$ **, 后者是** $0 \leqslant \alpha_i$ **.** 于是, 可采用 6.2节中同样的算法求解式 (6.40); 在引入**核函数**后能得到与式 (6.24) 同样的**支持向量展式**.

类似于式 (6.13), 对软间隔支持向量机, KKT条件要求

$$
\begin{cases}
\alpha_i \geqslant 0, \quad \mu_i \geqslant 0 \\
y_i f(\boldsymbol{x}_i) - 1 + \xi_i \geqslant 0 \\
\alpha_i\left(y_i f(\boldsymbol{x}_i) - 1 + \xi_i\right) = 0 \\
\xi_i \geqslant 0, \quad \mu_i\xi_i = 0
\end{cases} \tag{6.41}
$$

对任意训练样本 $(\boldsymbol{x}_i, y_i)$, 总有 $\alpha_i = 0$ 或 $y_i f(\boldsymbol{x}_i) = 1 - \xi_i$.

- 若 $\alpha_i = 0$, 则该样本不会对 $f(\boldsymbol{x})$ 有任何影响;
- 若 $\alpha_i > 0$, 则必有 $y_i f(\boldsymbol{x}_i) = 1 - \xi_i$, 即该样本是支持向量:
  - 由式 (6.39)可知, 若 $\alpha_i < C$, 则 $\mu_i > 0$, 进而有 $\xi_i = 0$, 即该样本恰在最大间隔边界上;
  - 若 $\alpha_i = C$, 则有 $\mu_i = 0$,
    - 此时若 $\xi_i \leqslant 1$, 则该样本落在最大间隔内部,
    - 若 $\xi_i > 1$, 则该样本被错误分类.

由此可看出, 软间隔支持向量机的最终模型**仅与支持向量有关**, 即通过采用hinge损失函数仍**保持了稀疏性**.

6.5 SMO算法详解

6.5.1 概论

(一) 概念

支持向量机的学习问题可以形式化为求解凸二次规划问题. 这样的凸二次规划问题具有全局最优解, 并且有许多最优化算法可以用于这一问题的求解. 但是当训练样本容量很大时, 这些算法往往变得非常低效, 以致无法使用.

所以, 如何高效地实现支持向量机学习就成为一个重要的问题. 目前人们已提出许多快速实现算法. 本节讲述其中的序列最小最优化 (sequential minimal optimization, SMO) 算法, 这种算法1998年由Platt提出.

(二) 目标问题--软间隔对偶问题

首先, 软间隔的对偶问题前面已经说过了, 也就是 (6.40)

$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j \tag{6.40}$$

$$s.t. \sum_{i=1}^{m} \alpha_i y_i = 0$$

$$0 \leqslant \alpha_i \leqslant C, \quad i = 1, 2, \ldots, m$$

同时, 对软间隔支持向量机, KKT条件要求

$$\begin{cases} \alpha_i \geqslant 0, \quad \mu_i \geqslant 0 \\ y_i f\left(\boldsymbol{x}_i\right)-1+\xi_i \geqslant 0 \\ \alpha_i\left(y_i f\left(\boldsymbol{x}_i\right)-1+\xi_i\right)=0 \\ \xi_i \geqslant 0, \quad \mu_i \xi_i=0 \end{cases} \tag{6.41}$$

接下来, 我们变换一下, 先把max变换为min, 然后把 $\boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j$ 用核函数表示为 $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, 关于核函数可参考 6.3.2 小节知识.

那么, 就可以变换为:

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)-\sum_{i=1}^{N} \alpha_i \tag{7.98}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} \alpha_i y_i=0 \tag{7.99}$$

$$0 \leqslant \alpha_i \leqslant C, \quad i=1,2, \cdots, N \tag{7.100}$$

在这个问题中, 变量时拉格朗日乘子, 一个变量 α_i 对应一个样本点 $(\boldsymbol{x}_i, y_i)$; 变量的总数等于训练样本容量.

(三) SMO算法的思路

SMO算法是一种启发式算法, 其基本思路是: **如果所有变量的解都满足此最优化问题的KKT条件 (Karush-Kuhn-Tucker conditions) , 那么这个最优化问题的解就得到了**. **因为KKT条件是该最优化问题的充分必要条件**. 否则, 选择**两个变量**, **固定其他变量**, **针对这两个变量构建一个二次规划问题**. 这个二次规划问题关于这两个变量的解应该更接近原始二次规划问题的解, 因为这会使得原始二次规划问题的目标函数值变得更小. 重要的是, 这时子问题可以通过解析方法求解, 这样就可以大大提高整个算法的计算速度. 子问题有两个变量, **一个是违反KKT条件最严重的那一个, 另一个由约束条件自动确定**. 如此, SMO算法将原问题不断分解为子问题并对子问题求解, 进而达到求解原问题的目的.

注意, **子问题的两个变量中只有一个是自由变量**, 假设 α_1, α_2 为两个变量, $\alpha_3, \alpha_4, \cdots, \alpha_N$ 固定, 那么由等式约束 (7.99) 可知:

$$\alpha_1=-y_1 \sum_{i=2}^{N} \alpha_i y_i$$

如果 α_2 确定, 那么 α_1 也随之确定. 所以子问题中同时更新两个变量.

整个SMO算法包括两个部分: **求解两个变量二次规划的解析方法**和**选择变量的启发式方法**

6.5.2 两个变量二次规划的求解方法

(一) 优化问题的改写

不失一般性, 假设选择的两个变量是 α_1, α_2 , 其他变量 $\alpha_i(i=3,4, \cdots, N)$ 是固定的. 于是SMO的最优化问题(7.98) ~ (7.100)的子问题可以写成:

$$\begin{aligned} \min_{\alpha_1, \alpha_2} \quad W\left(\alpha_1, \alpha_2\right)= & \frac{1}{2} K_{11} \alpha_1^2+\frac{1}{2} K_{22} \alpha_2^2+y_1 y_2 K_{12} \alpha_1 \alpha_2 \\ & -\left(\alpha_1+\alpha_2\right)+y_1 \alpha_1 \sum_{i=3}^{N} y_i \alpha_i K_{i 1}+y_2 \alpha_2 \sum_{i=3}^{N} y_i \alpha_i K_{i 2} \end{aligned} \tag{7.101}$$

$$\text{s.t.} \quad \alpha_1 y_1 + \alpha_2 y_2 = -\sum_{i=3}^{N} y_i \alpha_i = \zeta \tag{7.102}$$

$$0 \leqslant \alpha_i \leqslant C, \quad i = 1, 2, \cdots, N \tag{7.103}$$

其中, $K_{ij} = K\left(x_i, x_j\right), i, j = 1, 2, \cdots, N$, ζ 是常数, 目标函数式 (7.101) 中省略了不含 α_1, α_2 的常数项.

> 注: (7.101)的推导过程
>
> 直接带入即可化简, 可得到结果, 但是需要注意以下两个计算过程:
>
> - 但是要注意一个是 $K_{12} = K_{21}$,
> - $y_1\alpha_1 \sum_{i=3}^{N} y_i\alpha_i K_{i1}$ =$y_1\alpha_1 \sum_{j=3}^{N} y_j\alpha_i K_{1j}$, 所以可以合并在一起.

(二) 约束条件

为了求解两个变量的二次规划问题(7.101)~(7.103), 首先分析约束条件, 然后在此约束条件下求极小.

由于只有两个变量 α_1, α_2 , 约束可以用二维空间中的图形表示（如图7.8所示）

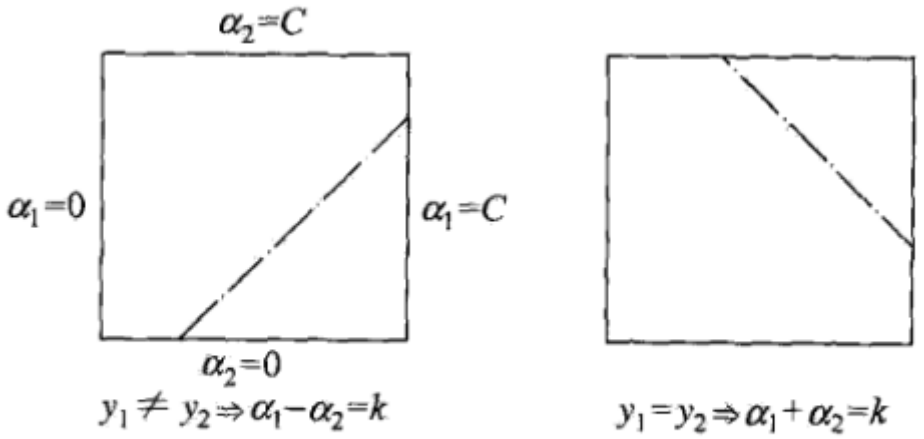


图 7.8　二变量优化问题图示

> 注: 观察约束条件 (7.102) 和 (7.103) , 因为 y_1 和 y_2 的取值只有{-1,1}, 所以当 $y_1 \neq y_2$ 时, 也就变为图 7.8 左边图, $y_1 = y_2$ 时, 也就时图 7.8 右边图

不等式约束 (7.103) 使得 (α_1, α_2) 在盒子 $[0, C] \times [0, C]$ 内, 等式约束 (7.102) 使 (α_1, α_2) 在平行于盒子 $[0, C] \times [0, C]$ 的对角线的直线上. **因此要求的是目标函数在一条平行于对角线的线段上的最优值**. 这使得两个变量的最优化问题成为**实质上的单变量的最优化问题**, 不妨考虑为变量 α_2 的最优化问题.

假设问题 (7.101)~(7.103) 的初始可行解为 α_1^{old} , α_2^{old} , 最优解为 α_1^{new} , α_2^{new} , 并且假设在沿着约束方向未经剪辑时 α_2 的最优解为 $\alpha_2^{\text{new,unc}}$.

由于 α_2^{new} 需满足不等式约束 (7.103) , 所以最优值 α_2^{new} 的取值范围必须满足条件

$$L \leqslant \alpha_2^{\text{new}} \leqslant H$$

其中, L 和 H 是 α_2^{new} 所在的对角线段端点的界. 如果 $y_1 \neq y_2$, 即图 7.8 左边图, 则有:

$$L = \max\left(0, \alpha_2^{\text{old}} - \alpha_1^{\text{old}}\right), \quad H = \min\left(C, C + \alpha_2^{\text{old}} - \alpha_1^{\text{old}}\right)$$

如果 $y_1 = y_2$, 即图 7.8 右边图, 则

$$L = \max\left(0, \alpha_2^{\text{old}} + \alpha_1^{\text{old}} - C\right), \quad H = \min\left(C, \alpha_2^{\text{old}} + \alpha_1^{\text{old}}\right)$$

注: **L 和 H 的推导过程**

首先根据原问题的约束条件和初始解,最优解有:

$$\alpha_1^{new} y_1 + \alpha_2^{new} y_2 = \alpha_1^{old} y_1 + \alpha_2^{old} y_2 = \zeta$$
$$0 \leqslant \alpha_i \leqslant C, \quad i = 1, 2, \cdots, N$$

第一种情况, 当 $y_1 \neq y_2$, 即图 7.8 左边图, 那么有:

$$\alpha_1^{old} - \alpha_2^{old} = \alpha_1^{new} - \alpha_2^{new} = \zeta$$

进行如下推导:

$$\alpha_2^{new} = \alpha_1^{new} - (\alpha_1^{old} - \alpha_2^{old}) \tag{I}$$

这里需要注意的一点是, α_2^{new} 是待求解的, α_1^{new} 是变化的.

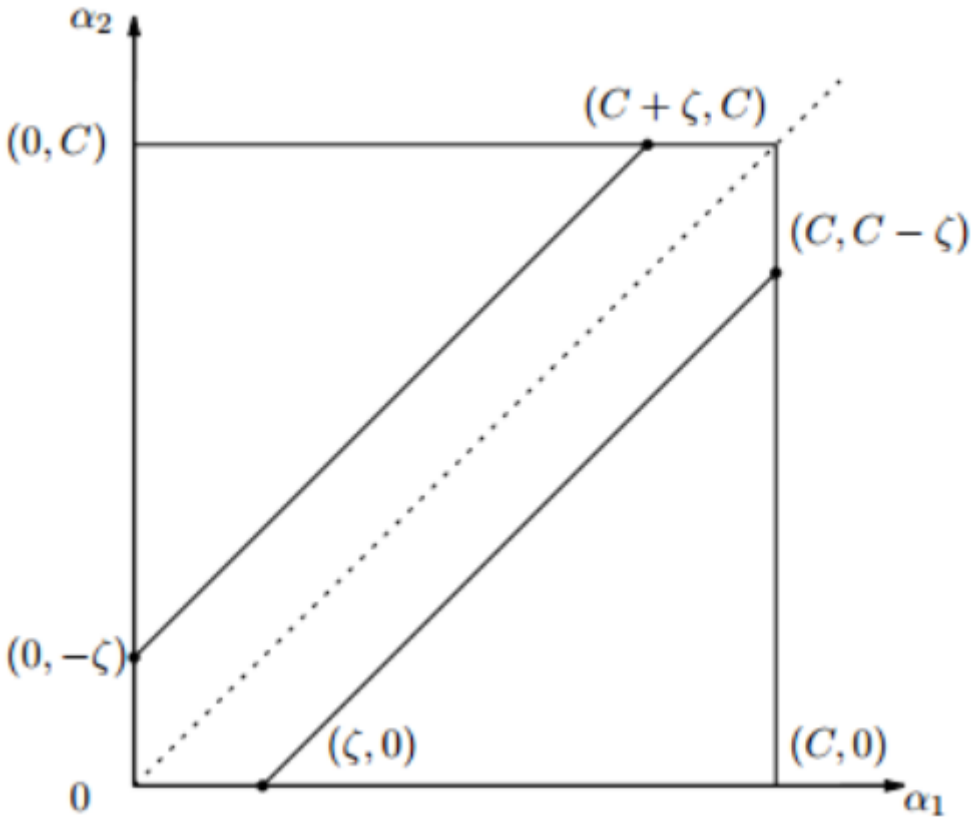
又 $0 \leqslant \alpha_2^{new} \leqslant C$, $0 \leqslant \alpha_1^{new} \leqslant C$, 那么 α_2^{new} 的最小值最小只能到0 , 什么时候取 0 呢, 就是 $(\alpha_1^{old} - \alpha_2^{old}) < 0$ 时, 当 $(\alpha_1^{old} - \alpha_2^{old}) > 0$ 时, 最小值就是 在 $\alpha_1^{new} = 0$ 时, I 式变为:

$\alpha_2^{new} = -(\alpha_1^{old} - \alpha_2^{old})$, 因此, L的取值范围就是 $L = \max\left(0, \alpha_2^{\text{old}} - \alpha_1^{\text{old}}\right)$

同理可以求得 H 的取值范围:

$$H = \min\left(C, C + \alpha_2^{\text{old}} - \alpha_1^{\text{old}}\right)$$

具体可以参见下图:

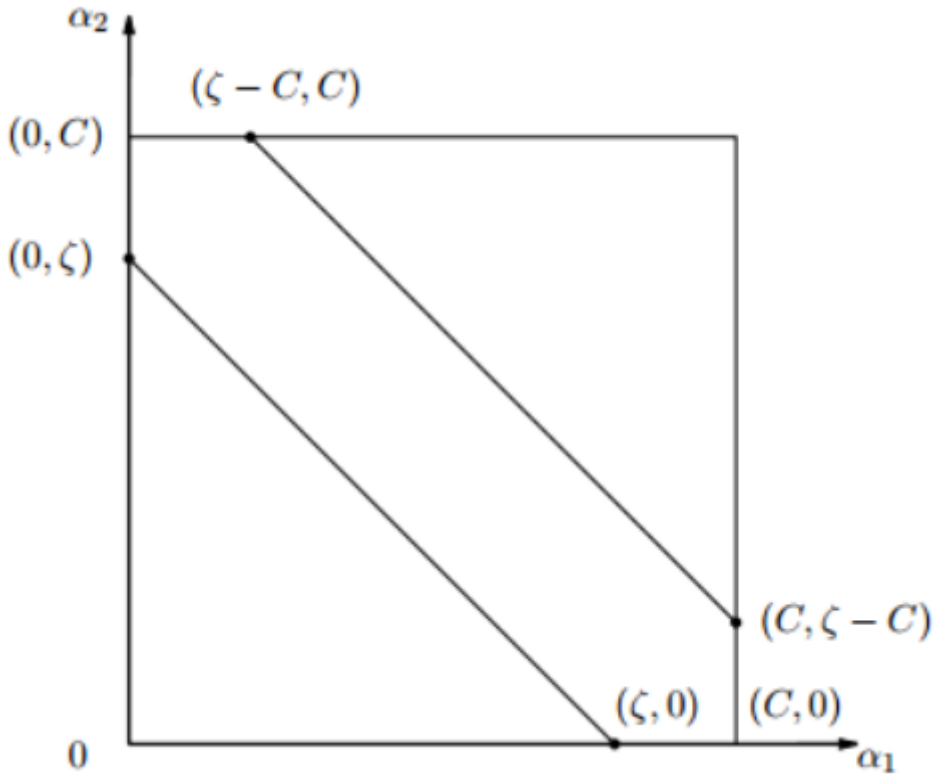


第二种情况, 如果 $y_1 = y_2$, 即图 7.8 右边图,

可以根据同样的方法,推导得到 L 和 H 的取值范围:

$$L = \max\left(0, \alpha_2^{\text{old}} + \alpha_1^{\text{old}} - C\right), \quad H = \min\left(C, \alpha_2^{\text{old}} + \alpha_1^{\text{old}}\right)$$

取值范围如下图:



(三) 两个变量的解

首先求沿着约束方向未经剪辑**即未考虑不等式约束(7.103)时** α_2 的最优解 $\alpha_2^{\text{new,unc}}$, 然后再求剪辑后 α_2 的解 α_2^{new}

为了后面公式的简洁, 记:

$$g(x) = \sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b \tag{7.104}$$

令

$$E_i = g(x_i) - y_i = \left(\sum_{j=1}^{N} \alpha_j y_j K(x_j, x_i) + b\right) - y_i, \quad i = 1, 2 \tag{7.105}$$

当 $i = 1, 2$ 时, $g(x)$ 为 x 的预测值, E_i 为函数 $g(x)$ 对输入 x_i 的预测值与真实输出 y_i 之差.

定理 7.6 两个变量的解

最优化问题 (7.101) ~ (7.103) 沿着约束方向**未经剪辑时的解是** :

$$\alpha_2^{\text{new, unc}} = \alpha_2^{\text{old}} + \frac{y_2(E_1 - E_2)}{\eta} \tag{7.106}$$

其中,

$$\eta = K_{11} + K_{22} - 2K_{12} = \left\| \Phi\left(x_1\right) - \Phi\left(x_2\right) \right\|^2 \tag{7.107}$$

$\Phi\left(x_1\right)$ 是输入空间到特征空间的映射, $E_i, \quad i = 1, 2$, 由式 (7.105) 给出.

经剪辑后 α_2 的解是

$$\alpha_2^{\text{new}} = \begin{cases} H, & \alpha_2^{\text{new,unc}} > H \\ \alpha_2^{\text{new,unc}}, & L \leqslant \alpha_2^{\text{new,unc}} \leqslant H \\ L, & \alpha_2^{\text{new,unc}} < L \end{cases} \tag{7.108}$$

由 α_2^{new} 求得 α_1^{new} 是:

$$\alpha_1^{\text{new}} = \alpha_1^{\text{old}} + y_1 y_2 \left(\alpha_2^{\text{old}} - \alpha_2^{\text{new}}\right) \tag{7.109}$$

> **注: 关于定理的推导过程.**
>
> 1. 关于**未经剪辑时的解的推导**过程:
>
> 请参考李航<统计学习方法> p127-128的证明
>
> 2. 经剪辑后的解 (7.108) 的解释:
>
> 要使其满足不等式约束必须将其限制在区间 $[L, H]$内, 从而得到 α_2^{new} 的表达式 (7.108)
>
> 3. α_1^{new} 的解 (7.109) 的解释:
>
> 由等式约束 (7.102) , 得到 α_1^{new} 的表达式 (7.109)

6.5.3 变量的选择方法

SMO算法在每个子问题中选择两个变量优化, 其中至少一个变量是违反KKT条件的.

(一) 第 1 个变量的选择

SMO称选择第1个变量的过程为外层循环. 外层循环在训练样本中选取违反 KKT 条件最严重的样本点, 并将其对应的变量作为第1个变量. 具体地, 检验训练样本点 $\left(x_i, y_i\right)$ 是否满足KKT条件, 即

$$\alpha_i = 0 \Leftrightarrow y_i g\left(x_i\right) \geqslant 1 \tag{7.111}$$

$$0 < \alpha_i < C \Leftrightarrow y_i g\left(x_i\right) = 1 \tag{7.112}$$

$$\alpha_i = C \Leftrightarrow y_i g\left(x_i\right) \leqslant 1 \tag{7.113}$$

其中, $g\left(x_i\right) = \sum_{j=1}^{N} \alpha_j y_j K\left(x_i, x_j\right) + b$.

> 注1 : 关于 (7.111)~(7.113) 的推导:
>
> 1. $\alpha_i = 0$
>
> 由(6.39) 知: $C = \alpha_i + \mu_i$, 可得:
>
> $$\mu_i = C$$
>
> 再由对偶问题的 kkt 条件 (6.41) 中的 $\mu_i \xi_i = 0$ 可知:
>
> $$\xi_i = 0$$

> 再由 kkt 条件中的 $y_i f\left(\boldsymbol{x}_i\right)-1+\xi_i \geqslant 0$ (或者原始问题的约束条件, 是一样的), 有:
>
> $$y_i g\left(x_i\right) \geqslant 1$$
>
> 2. $0<\alpha_i<C$
>
> 若 $\alpha_i>0$, 则必有 $y_i f\left(\boldsymbol{x}_i\right)=1-\xi_i$, 即该样本是支持向量, 由式 (6.39), 即 $C=\alpha_i+\mu_i$ 可知, 若 $\alpha_i<C$, 则 $\mu_i>0$, 根据 $\mu_i \xi_i=0$, 进而有 $\xi_i=0$, 即该样本恰在最大间隔边界上; 所以有, $y_i f\left(\boldsymbol{x}_i\right)=1$, 即 $y_i g\left(x_i\right)=1$
>
> 3. $\alpha_i=C$
>
> 首先, $\xi_i \geqslant 0$, 同时, 由于 $\alpha_i=C$, 那么由 $\alpha_i\left(y_i f\left(\boldsymbol{x}_i\right)-1+\xi_i\right)=0$ 可得, $\left(y_i f\left(\boldsymbol{x}_i\right)-1+\xi_i\right)=0$, 所以 $y_i f(\boldsymbol{x}_i) \leqslant 1$

> 注2: 其实 (7.111)~(7.113) 就是 kkt 条件 (6.41) 的充要条件, 两者可以互相推出.

检验是在精度 ε 范围内进行的. 在检验过程中, 外层循环首先遍历所有满足条件 $0<\alpha_i<C$ 的样本点, 即在间隔边界上的支持向量点, 检验它们是否满足KKT条件. 如果这些样本点都满足KKT条件, 那么遍历整个训练集, 检验它们是否满足KKT条件.

(二) 第 2 个变量的选择

SMO称选择第2个变量的过程为内层循环. 假设在外层循环中已经找到第1个变量 α_1 , 现在要在内层循环中找第2个变量 α_2 . 第2个变量选择的标准是希望能使 α_2 有足够大的变化. 由式 (7.106) 和式(7.108) 可知, 是依赖于$\left|E_1-E_2\right|$ 的, 为了加快计算速度, 一种简单的做法是选择 α_2 , 使其对应的 $\left|E_1-E_2\right|$ 最大. 因为 α_1 已定, E_1 也确定了. 如果 E_1 是正的, 那么选择最小的 E_i 作为E_2; 如果 E_1 是负的, 那么选择最大的 E_i作为E_2. 为了节省计算时间, 将所有 E_i 值保存在一个列表中. 在特殊情况下, 如果内层循环通过以上方法选择的 α_2 不能使目标函数有足够的下降, 那么采用以下启发式规则继续选择 α_2 . 遍历在间隔边界上的支持向量点, 依次将其对应的变量作为 α_2 试用, 直到目标函数有足够的下降. 若找不到合适的 α_2 , 那么遍历训练数据集; 若仍找不到合适的 α_2 , 则放弃第1个 α_1 , 再通过外层循环寻求另外的 α_1

(三) 计算阈值 b 和差值 E_i

在每次完成两个变量的优化后, 都要重新计算阈值b。 当 $0<\alpha_1^{\text{new}}<\mathrm{C}$ 时, 由 KKT 条件 (7.112) 可知:

$$\sum_{i=1}^N \alpha_i y_i K_{i 1}+b=y_1$$

> 注: $\left(x_1, y_1\right)$ 也满足(7.112), 两边同乘以 y_1 , 有:
>
> $$y_1^2 g\left(x_i\right)=y_1$$
>
> 又 $y_1^2=1$, 即可得到上述结论

于是, 可得:

$$b_1^{\text{new}}=y_1-\sum_{i=3}^N \alpha_i y_i K_{i 1}-\alpha_1^{\text{new}} y_1 K_{11}-\alpha_2^{\text{new}} y_2 K_{21} \tag{7.114}$$

由 E_1 的定义式 (7.105) 有:

$$E_1 = \sum_{i=3}^{N} \alpha_i y_i K_{i1} + \alpha_1^{\text{old}} y_1 K_{11} + \alpha_2^{\text{old}} y_2 K_{21} + b^{\text{old}} - y_1$$

式 (7.114) 的前两项可以通过 E_1 改写为:

$$y_1 - \sum_{i=3}^{N} \alpha_i y_i K_{i1} = -E_1 + \alpha_1^{\text{old}} y_1 K_{11} + \alpha_2^{\text{old}} y_2 K_{21} + b^{\text{old}}$$

带入式 (7.114) , 可得:

$$b_1^{\text{new}} = -E_1 - y_1 K_{11} \left(\alpha_1^{\text{new}} - \alpha_1^{\text{eld}}\right) - y_2 K_{21} \left(\alpha_2^{\text{new}} - \alpha_2^{\text{old}}\right) + b^{\text{old}} \tag{7.115}$$

那么, 同样的, 如果 $0 < \alpha_2^{\text{new}} < C$, 则有:

$$b_2^{\text{new}} = -E_2 - y_1 K_{12} \left(\alpha_1^{\text{new}} - \alpha_1^{\text{old}}\right) - y_2 K_{22} \left(\alpha_2^{\text{new}} - \alpha_2^{\text{old}}\right) + b^{\text{old}} \tag{7.116}$$

- **如果 α_1^{new} , α_2^{new} 同时满足条件 $0 < \alpha_i^{\text{new}} < C, i = 1, 2$ (也就是 b_1^{new} 和 b_2^{new} 都有效的时候), 他们是相等的, 即 $b^{\text{new}} = b_1^{\text{new}} = b_2^{\text{new}}$**
- **如果 α_1^{new} , α_2^{new} 是 0 或者 C , 那么 b_1^{new} 和 b_2^{new} 以及他们两者之间的数都是符合 KKT 条件的阈值, 这时选择它们的中点作为 $b^{new} = \frac{b_1^{new} + b_2^{new}}{2}$**

6.5.4 SMO算法总结

算法 7.5 (SMO算法)

输入: 训练数据集 $T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$, 其中, $x_i \in \mathcal{X} = \mathbf{R}^n$, $y_i \in \mathcal{Y} = \{-1, +1\}, \quad i = 1, 2, \cdots, N$, 精度 $\mathcal{E}$

输出: 近似解 $\hat{\alpha}$

- (1) 取初值 $\alpha^{(0)} = 0$, 令 $k = 0$;
- (2) 按照 6.5.3 变量的选择方法中第一个变量选择, 选择第一个变量 $\alpha_1^{(k)}$, 按照第二个变量选择方法选择第二个变量 $\alpha_2^{(k)}$, 根据式 (7.106) , 求出新的 $\alpha_2^{\text{new, unc}}$,

$$\alpha_2^{\text{new, unc}} = \alpha_2^{(k)} + \frac{y_2 \left(E_1 - E_2\right)}{\eta}$$

- (3) 按照下式 (即式 (7.108)) 求出 $\alpha_2^{(k+1)}$

$$\alpha_2^{(k+1)} = \begin{cases} H, & \alpha_2^{\text{new,unc}} > H \\ \alpha_2^{\text{new,unc}}, & L \leqslant \alpha_2^{\text{new,unc}} \leqslant H \\ L, & \alpha_2^{\text{new,unc}} < L \end{cases}$$

- (4) 利用 $\alpha_2^{(k+1)}$ 和 $\alpha_1^{(k+1)}$ 的关系(即式 (7.109)) , 求出 $\alpha_1^{(k+1)}$.

$$\alpha_1^{(k+1)} = \alpha_1^{(k)} + y_1 y_2 \left(\alpha_2^{(k)} - \alpha_2^{(k+1)}\right)$$

- (5) 按照 6.5.3 变量的选择方法中的 (三) 计算阈值 b 和差值 E_i , 计算 b^{k+1} 和 E_i
- (6) 在精度 $\mathcal{E}$ 范围内检查是否满足如下的终止条件:

$$\sum_{i=1}^{N} \alpha_i y_i = 0$$

$$0 \leqslant \alpha_i \leqslant C, \quad i=1,2,\cdots,N$$

$$\alpha_i^{k+1}=0 \Rightarrow y_i g\left(x_i\right) \geq 1$$

$$0<\alpha_i^{k+1}<C \Rightarrow y_i g\left(x_i\right)=1$$

$$\alpha_i^{k+1}=C \Rightarrow y_i g\left(x_i\right) \leq 1$$

- (7) 如果满足则结束, 返回 α_i^{k+1}, 否则转到步骤 (2)
