$$-\frac{\partial L(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^{(j)}}=\sum_{i=1}^{n}\left[\left(\pi\left(x_i\right)-y_i\right)x_i^{(j)}\right]$$

所以有如下梯度下降法的迭代公式：

$$\boldsymbol{\theta}^{(j)}=\boldsymbol{\theta}^{(j)}-\alpha\sum_{i=1}^{n}\left[\left(\pi\left(x_i\right)-y_i\right)x_i^{(j)}\right]$$

4.7.2 支持向量机

支持向量机（support vector machine, SVM）是监督学习中最有影响力的方法之一(Boser et al., 1992; Cortes and Vapnik, 1995)。类似于逻辑回归，这个模型也是基于线性函数$\boldsymbol{w}^{\top}\boldsymbol{x}+b$的。不同于逻辑回归的是，支持向量机不输出概率，只输出类别。当$\boldsymbol{w}^{\top}\boldsymbol{x}+b$为正时，支持向量机预测属于正类。类似地，当$\boldsymbol{w}^{\top}\boldsymbol{x}+b$为负时，支持向量机预测属于负类。

支持向量机的一个重要创新是核技巧（kernel trick）。核技巧观察到许多机器学习算法都可以写成样本间点积的形式。例如，支持向量机中的线性函数可以重写为

$$\boldsymbol{w}^{\top}\boldsymbol{x}+b=b+\sum_{i=1}^{m}\alpha_i\boldsymbol{x}^{\top}\boldsymbol{x}^{(i)}$$

其中，$\boldsymbol{x}^{(i)}$是训练样本，α是系数向量。学习算法重写为这种形式允许我们将$\boldsymbol{x}$替换为特征函数$\boldsymbol{\phi}(\boldsymbol{x})$的输出，点积替换为被称为核函数（kernel function）的函数$k\left(\boldsymbol{x},\boldsymbol{x}^{(i)}\right)=\boldsymbol{\phi}(\boldsymbol{x})\cdot\boldsymbol{\phi}\left(\boldsymbol{x}^{(i)}\right)$。运算符$\cdot$表示类似于$\boldsymbol{\phi}(\boldsymbol{x})^{\top}\boldsymbol{\phi}\left(\boldsymbol{x}^{(i)}\right)$的点积。对于某些特征空间，我们可能不会书面地使用向量内积。在某些无限维空间中，我们需要使用其他类型的内积，如基于积分而非加和的内积。

使用核估计替换点积之后，我们可以使用如下函数进行预测

$$f(\boldsymbol{x})=b+\sum_{i}\alpha_i k\left(\boldsymbol{x},\boldsymbol{x}^{(i)}\right)$$

这个函数关于$\boldsymbol{x}$是非线性的，关于$\boldsymbol{\phi}(\boldsymbol{x})$是线性的。$\alpha$和$f(\boldsymbol{x})$之间的关系也是线性的。核函数完全等价于用$\boldsymbol{\phi}(\boldsymbol{x})$预处理所有的输入，然后在新的转换空间学习线性模型。

核技巧十分强大有两个原因。首先，它使我们能够使用保证有效收敛的凸优化技术来学习非线性模型（关于$\boldsymbol{x}$的函数）。这是可能的，因为我们可以认为$\boldsymbol{\phi}$是固定的，仅优化α，即优化算法可以将决策函数视为不同空间中的线性函数。其二，核函数k的实现

方法通常比直接构建 $\phi(\boldsymbol{x})$ 再算点积高效很多。

最常用的核函数是高斯核（Gaussian kernel），

$$k(\boldsymbol{u},\boldsymbol{v})=\mathcal{N}\left(\boldsymbol{u}-\boldsymbol{v};\boldsymbol{0},\sigma^2 I\right)$$

其中 $\mathcal{N}(x;\boldsymbol{\mu},\boldsymbol{\Sigma})$ 是标准正态密度。这个核也被称为径向基函数（radial basis function, RBF）核，因为其值沿 $\boldsymbol{v}$ 中从 $\boldsymbol{u}$ 向外辐射的方向减小。高斯核对应于无限维空间中的点积，但是该空间的推导没有整数上最小核的示例那么直观。

我们可以认为高斯核在执行一种模板匹配(template matching)。训练标签 y 相关的训练样本 x 变成了类别 y 的模版。当测试点 x' 到 x 的欧几里得距离很小，对应的高斯核响应很大时，表明 x' 和模版 x 非常相似。该模型进而会赋予相对应的训练标签 y 较大的权重。总的来说，预测将会组合很多这种通过训练样本相似度加权的训练标签。

支持向量机不是唯一可以使用核技巧来增强的算法。许多其他的线性模型也可以通过这种方式来增强。使用核技巧的算法类别被称为核机器（kernel machine）或核方法（kernel method）(Williams and Rasmussen, 1996; Schölkopf et al., 1999)。

核机器的一个主要缺点是计算决策函数的成本关于训练样本的数目是线性的。因为第 i 个样本贡献 $\alpha_i k\left(\boldsymbol{x},\boldsymbol{x}^{(i)}\right)$ 到决策函数。支持向量机能够通过学习主要包含零的向量 α，以缓和这个缺点。那么判断新样本的类别仅需要计算非零 α_i 对应的训练样本的核函数。这些训练样本被称为支持向量（support vector）。

花书中的这部分内容写的很简单，我们结合视频教程以及其他学习资源（比较好的应该是周志华的《西瓜书》和邱锡鹏的《神经网络与深度学习》），将支持向量机思想、推导过程细细地讲一下。

4.7.2.1 间隔与支持向量

给定训练样本集 $D=\left\{\left(\boldsymbol{x}_1,y_1\right),\left(\boldsymbol{x}_2,y_2\right),\ldots,\left(\boldsymbol{x}_m,y_m\right)\right\}, y_i\in\{-1,+1\}$，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但能将训练样本分开的划分超平面可能有很多，如图 33 所示，我们应该努力去找到哪一个呢?

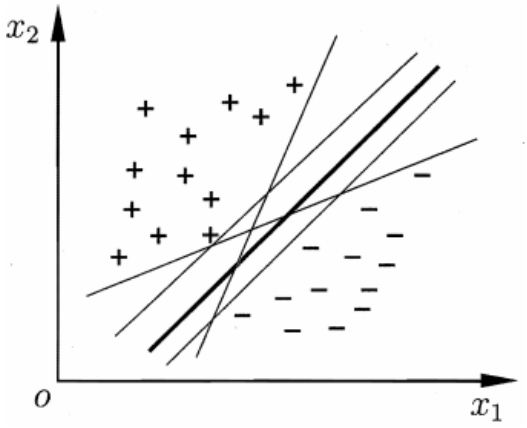


图 33 存在多个划分超平面将两类训练样本分开

直观上看，应该去找位于两类训练样本"正中间"的划分超平面，即图 33 深色的那个，因为该划分超平面对训练样本局部扰动的“容忍”性最好。（正确分类对应经验误差，最好的分类对应结构误差），所以 SVM 需要优化两个误差，从而使得分类结果更加鲁棒。

在样本空间中，划分超平面可通过如下线性方程来描述:

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

其中 $\boldsymbol{w}=\left(w_1;w_2;\ldots;w_d\right)$ 为法向量，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。显然，划分超平面可被法向量 $\boldsymbol{w}$ 和位移 b 确定，下面我们将其记为 $(\boldsymbol{w},b)$ 。样本空间中任意点 $\boldsymbol{x}$ 到超平面 $(\boldsymbol{w},b)$ 的距离可写为

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

假设超平面 $(\boldsymbol{w},b)$ 能将训练样本正确分类，即对于 $\left(\boldsymbol{x}_i,y_i\right)\in D$ ，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

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

如图 34 所示，距离超平面最近的这几个训练样本点使上式的等号成立，它们被称为"支持向量"(support vector)，两个异类支持向量到超平面的距离之和为

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

它被称为"间隔"(margin)。

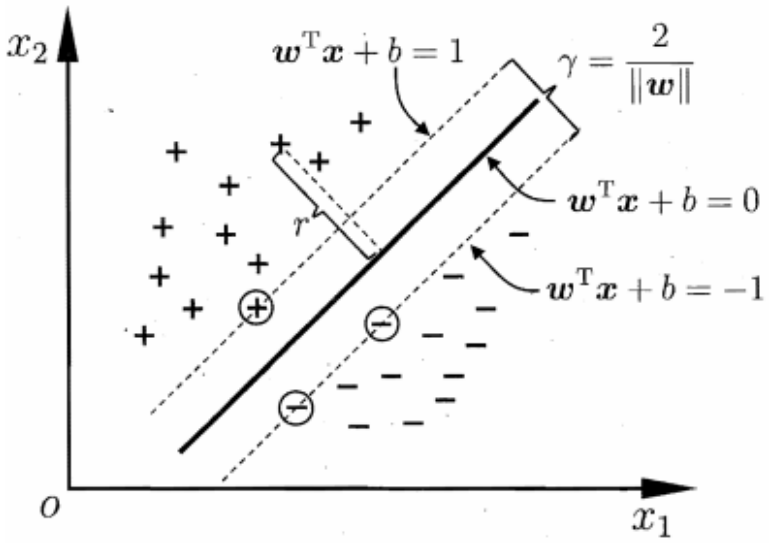


图 34 支持向量与间隔

这里需要注意的是，边界上的=1 是怎么来的呢？对于中间的直线为$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b=0$，通过平移使得该直线通过其中一个类的第一个样本，此时线性方程变为$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b=c$，两边同时除以$c$得到$\boldsymbol{w}'^{\mathrm{T}}\boldsymbol{x}+b'=1$，对于$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b=0$两边也同时除以$c$得到$\boldsymbol{w}'^{\mathrm{T}}\boldsymbol{x}+b'=0$，根据直线关于直线的对称得到另一条直线$\boldsymbol{w}'^{\mathrm{T}}\boldsymbol{x}+b'=-1$，最后用$\boldsymbol{w}$和$b$代替$\boldsymbol{w}'$和$b'$就得到了图中的形式。根据解析几何可知，两条平行直线间的距离为$d=\frac{|c_1-c_2|}{\|\boldsymbol{w}\|}$，所以得到间隔为$\gamma=\frac{|1-(-1)|}{\|\boldsymbol{w}\|}=\frac{2}{\|\boldsymbol{w}\|}$。

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

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

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

显然，为了最大化间隔，仅需最大化$\|\boldsymbol{w}\|^{-1}$，这等价于最小化$\|\boldsymbol{w}\|^2$。于是，上式可重写为

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

这就是支持向量机(Support Vector Machine，简称 SVM)的基本型。

例 7.1 数据与例 2.1 相同. 已知一个如图 7.4 所示的训练数据集，其正例点是 $x_1=(3,3)^T$， $x_2=(4,3)^T$，负例点是 $x_3=(1,1)^T$，试求最大间隔分离超平面.

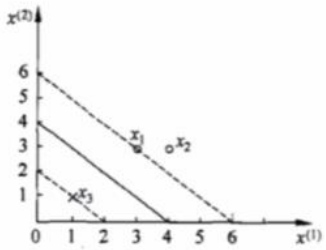


图 7.4 间隔最大分离超平面示例

解 按照算法 7.1，根据训练数据集构造约束最优化问题：

$$\begin{aligned}\min_{w,b}\ & \frac{1}{2}(w_1^2+w_2^2)\\ \text{s.t.}\ & 3w_1+3w_2+b\geqslant 1\\ & 4w_1+3w_2+b\geqslant 1\\ & -w_1-w_2-b\geqslant 1\end{aligned}$$

正好是 KKT 的标准形式

$$\nabla f(x)=\begin{pmatrix}w_1\\w_2\end{pmatrix}\quad \nabla g_1(x)=\begin{pmatrix}3\\3\end{pmatrix}\quad \nabla g_2(x)=\begin{pmatrix}4\\3\end{pmatrix}\quad \nabla g_3(x)=\begin{pmatrix}-1\\-1\end{pmatrix}$$

$$\nabla f(x)-\lambda_1\nabla g_1(x)-\lambda_2\nabla g_2(x)-\lambda_3\nabla g_3(x)=0$$

$$\begin{cases}\begin{pmatrix}w_1\\w_2\end{pmatrix}-\lambda_1\begin{pmatrix}3\\3\end{pmatrix}-\lambda_2\begin{pmatrix}4\\3\end{pmatrix}-\lambda_3\begin{pmatrix}-1\\-1\end{pmatrix}=0\\ \left.\begin{aligned}\lambda_1(3w_1+3w_2+b-1)=0\\ \lambda_2(4w_1+3w_2+b-1)=0\\ \lambda_3(-w_1-w_2-b-1)=0\end{aligned}\right\}\ \text{KKT条件}\ \lambda_i g_i(x)=0\\ \lambda_1\geq 0\\ \lambda_2\geq 0\\ \lambda_3\geq 0\end{cases}$$

$$\Rightarrow\quad w_1=w_2=\frac{1}{2},b=-2$$

$$\lambda_1=\frac{1}{3},\lambda_2=0,\lambda_3=\frac{1}{2}$$

求得此最优化问题的解 $w_1=w_2=\frac{1}{2}$， $b=-2$. 于是最大间隔分离超平面为

$$\frac{1}{2}x^{(1)}+\frac{1}{2}x^{(2)}-2=0$$

其中， $x_1=(3,3)^T$ 与 $x_3=(1,1)^T$ 为支持向量.

图 35 SVM 样例

4.7.2.2 对偶问题

（1） 原始问题

1. 原始问题

假设 $f(x)$，$c_i(x)$，$h_j(x)$ 是定义在 $\mathbf{R}^n$ 上的连续可微函数．考虑约束最优化问题

$$\min_{x\in\mathbf{R}^n} f(x) \tag{C.1}$$

$$\text{s.t. } c_i(x)\leqslant 0,\quad i=1,2,\cdots,k \tag{C.2}$$

$$h_j(x)=0,\quad j=1,2,\cdots,l \tag{C.3}$$

称此约束最优化问题为原始最优化问题或原始问题．

首先，引进广义拉格朗日函数（generalized Lagrange function）

$$L(x,\alpha,\beta)=f(x)+\sum_{i=1}^{k}\alpha_i c_i(x)+\sum_{j=1}^{l}\beta_j h_j(x) \tag{C.4}$$

这里，$x=(x^{(1)},x^{(2)},\cdots,x^{(n)})^{\mathrm{T}}\in\mathbf{R}^n$，$\alpha_i,\beta_j$ 是拉格朗日乘子，$\alpha_i\geqslant 0$．考虑 x 的函数：

$$\theta_P(x)=\max_{\alpha,\beta,\alpha_i\geqslant 0} L(x,\alpha,\beta) \tag{C.5}$$

最大值是 x 的函数

这里，下标 P 表示原始问题．

假设给定某个 x．如果 x 违反原始问题的约束条件，即存在某个 i 使得 $c_i(x)>0$ 或者存在某个 j 使得 $h_j(x)\neq 0$，那么就有

因为是取最大值，所以可以选取不同的 α 和 β 使得最大值为正无穷。

$$\theta_P(x)=\max_{\alpha,\beta,\alpha_i\geqslant 0}\left[f(x)+\sum_{i=1}^{k}\alpha_i c_i(x)+\sum_{j=1}^{l}\beta_j h_j(x)\right]=+\infty \tag{C.6}$$

因为若某个 i 使约束 $c_i(x)>0$，则可令 $\alpha_i\to+\infty$，若某个 j 使 $h_j(x)\neq 0$，则可令 β_j 使 $\beta_j h_j(x)\to+\infty$，而将其余各 α_i，β_j 均取为 0．

相反地，如果 x 满足约束条件式 (C.2) 和式 (C.3)，则由式 (C.5) 和式 (C.4) 可知，$\theta_P(x)=f(x)$．因此，

满足条件时，为 0，小于等于 0，所以最大为 f（x）。

$$\theta_P(x)=\begin{cases}f(x), & x\text{满足原始问题约束}\\ +\infty, & \text{其他}\end{cases} \tag{C.7}$$

所以如果考虑极小化问题

$$\min_x \theta_P(x)=\min_x \max_{\alpha,\beta,\alpha_i\geqslant 0} L(x,\alpha,\beta) \tag{C.8}$$

它是与原始最优化问题 (C.1) ～ (C.3) 等价的，即它们有相同的解．问题 $\min_x \max_{\alpha,\beta,\alpha_i\geqslant 0} L(x,\alpha,\beta)$ 称为广义拉格朗日函数的极小极大问题．这样一来，就把原始最优化问题表示为广义拉格朗日函数的极小极大问题．为了方便，定义原始问题的最优值

$$p^*=\min_x \theta_P(x) \tag{C.9}$$

称为原始问题的值．

图 36 原始问题

（2） 对偶问题

定义

$$\theta_D(\alpha,\beta)=\min_x L(x,\alpha,\beta)$$ 最小值是 α 和 β 的函数

再考虑最大化 $\theta_D(\alpha,\beta)=\min_x L(x,\alpha,\beta)$，即

$$\max_{\alpha,\beta;\alpha_i\geq 0}\theta_D(\alpha,\beta)=\max_{\alpha,\beta;\alpha_i>0}\min_x L(x,\alpha,\beta)$$

问题 $\max_{\alpha,\beta;\alpha_i>0}\min_x L(x,\alpha,\beta)$ 称为广义拉格朗日函数的极大极小问题。

（3） SMO 算法(序列最小优化算法)

$$\begin{cases}\min\dfrac{1}{2}w^2\\ (wx_i+b)y_i\geq 1\end{cases}$$ 标准形式

$$\min_{x\in\mathbb{R}^n} f(x)$$
$$\text{s.t.}\quad c_i(x)\leq 0,\ i=1,2,\cdots,k$$
$$h_j(x)=0,\ j=1,2,\cdots,l$$

$$L(w,b,\alpha)=\frac{1}{2}w^2+\sum_{i=1}^{N}\alpha_i[1-(wx_i+b)y_i]$$

原问题：$\min\limits_{w,b}\ \max\limits_{\alpha_i\geq 0}\ L(w,b,\alpha)$

对偶问题：$\max\limits_{\alpha_i\geq 0}\ \min\limits_{w,b}\ L(w,b,\alpha)$

对 w 和 b 求导

图 37 转换为对偶问题

$$\frac{\partial L}{\partial \boldsymbol{w}}=0\Rightarrow \boldsymbol{w}-\sum_{i=1}^{N}\alpha_i\boldsymbol{x}_i y_i=0\Rightarrow \boldsymbol{w}=\sum_{i=1}^{N}\alpha_i\boldsymbol{x}_i y_i$$
$$\frac{\partial L}{\partial b}=0\Rightarrow\sum_{i=1}^{N}\alpha_i y_i=0$$

将上式代回到对偶问题中（视频中因为没有用向量表示，所以不太好推导）

$$\begin{aligned}L(w,b,\alpha)&=\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}+\sum_{i=1}^{N}\alpha_i-\sum_{i=1}^{N}\alpha_i y_i\boldsymbol{w}^T\boldsymbol{x}_i-\sum_{i=1}^{N}\alpha_i y_i b\\ &=\sum_{i=1}^{N}\alpha_i+\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}-\boldsymbol{w}^T\boldsymbol{w}\\ &=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\left(\sum_{i=1}^{N}\alpha_i y_i\boldsymbol{x}_i\right)^T\left(\sum_{j=1}^{N}\alpha_j y_j\boldsymbol{x}_j\right)\\ &=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\boldsymbol{x}_i^T\boldsymbol{x}_j\end{aligned}$$

所以原问题的对偶问题如下：

$$\begin{aligned}&\max_{\alpha} \sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j \\ &\text{s.t.} \quad \sum_{i=1}^{N}\alpha_i y_i=0 \\ &\quad\ \alpha_i\geqslant 0,\quad i=1,2,\ldots,N\end{aligned} \qquad \text{（原问题的对偶）}$$

KKT 条件为$\alpha_i\left[1-\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\right)y_i\right]=0$。

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

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

于是，对任意训练样本$\left(\boldsymbol{x}_i,y_i\right)$，总有$\alpha_i=0$或$\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\right)y_i=1$。若$\alpha_i=0$，则该样本将不会在上式的求和中出现，也就不会对$f(\boldsymbol{x})$有任何影响;若$\alpha_i>0$，则必有$\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\right)y_i=1$，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。

那么如何求解式对偶问题呢？不难发现，这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向上的极值。由于存在约束$\sum_{i=1}^{N}\alpha_i y_i=0$，若固定$\alpha_i$之外的其他变量，则$\alpha_i$可由其他变量导出。于是，SMO 每次选择两个变量$\alpha_i$和$\alpha_j$，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（原问题的对偶）获得更新后的α_i和α_j。

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（原问题的对偶）中的约束可重写为

$$\alpha_i y_i + \alpha_j y_j = c, \quad \alpha_i \geqslant 0, \quad \alpha_j \geqslant 0$$

其中

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

如何确定偏移项 b 呢？注意到对任意支持向量$(\boldsymbol{x}_s, y_s)$都有$y_s f(\boldsymbol{x}_s) = 1$，即

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

其中，$S = \{i \mid \alpha_i > 0, i = 1, 2, \ldots, m\}$为所有支持向量的下标集。理论上，可选择任意支持向量并通过上式获得 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

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

4.7.2.3 核函数

在前面的讨论中，我们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 38 所示。

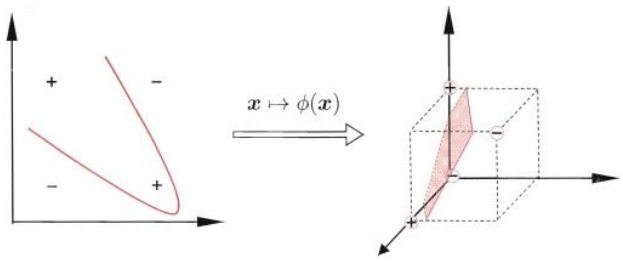


图 38 异或问题

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。例如在图 38 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。幸运的是，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。

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

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

其中 $\boldsymbol{w}$ 和 b 是模型参数。则有

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

其对偶问题是

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

（注意这里使用 m 表示样本数，但不影响说明）

求解式上式涉及到计算 $\phi\left(\boldsymbol{x}_i\right)^{\mathrm{T}}\phi\left(\boldsymbol{x}_j\right)$，这是样本 $\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$ 映射到特征空间之后的内积。由于特征空间维数可能很高，甚至可能是无穷维，因此直接计算 $\phi\left(\boldsymbol{x}_i\right)^{\mathrm{T}}\phi\left(\boldsymbol{x}_j\right)$ 通常是困难的。为了避开这个障碍，可以设想这样一个函数：

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

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 κ(，)计算的结果。有了这样的函数，我们就不必直接去计算高维甚至无穷维特征空间中的内积，于是式可重写为

$$\begin{aligned}&\max_{\alpha}\quad\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j\kappa\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right)\\&\text{s.t.}\quad\sum_{i=1}^{m}\alpha_i y_i=0\\&\qquad\alpha_i\geqslant 0,\quad i=1,2,\ldots,m\end{aligned}$$

求解后即可得到

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

显然，若已知合适映射 ϕ 的具体形式，则可写出核函数 κ(,)。但在现实任务中我们通常不知道 ϕ 是什么形式，那么，合适的核函数是否一定存在呢？什么样的函数能做核函数呢？我们有下面的定理：

定理（核函数）：令 χ 为输入空间，κ(,)是定义在 χ x χ 上的对称函数，则 κ 是核函数当且仅当对于任意数据 $D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m\}$，"核矩阵"(kernel matrix) K 总是半正定的：

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

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。

表 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, $\beta>0,\theta<0$

此外，还可通过函数组合得到，例如：

若 κ1 和 κ2 为核函数，则对于任意正数 γ_1、γ_2，其线性组合也是核函数；

$$\gamma_1\kappa_1+\gamma_2\kappa_2$$

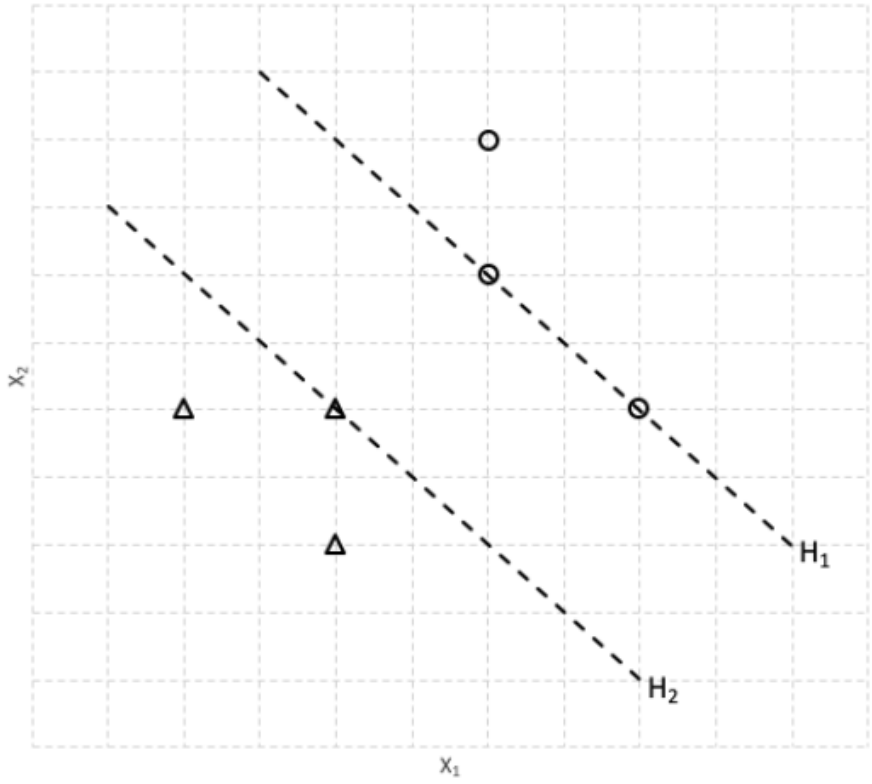
若 κ1 和 κ2 为核函数，则核函数的直积也是核函数；

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

若 κ1 为核函数，则对于任意函数 g(x)， $\kappa(\boldsymbol{x},\boldsymbol{z}) = g(\boldsymbol{x})\kappa_1(\boldsymbol{x},\boldsymbol{z})g(\boldsymbol{z})$ 也是核函数。

4.7.2.4 作业

假设我们要学习一个硬间隔 SVM，其线性决策函数为$\mathbf{w}^{\intercal}\mathbf{x} + b = 0$。输入特征为$x_1, x_2$，标签$y \in \{-1, +1\}$（分别用△和○表示）。训练数据如下图所示。

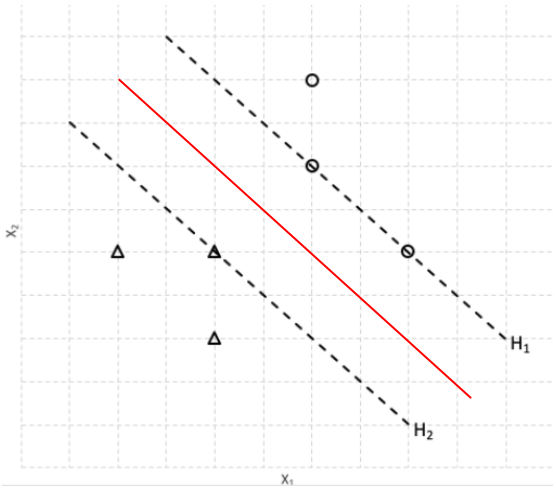


i. 根据最大间隔原则，在图中标出支持向量，并大致画出对应的分割超平面。
ii. 若超平面H_1的表达式为$\mathbf{w}^{\intercal}\mathbf{x} + b = 1$，写出超平面$H_2$的表达式。
iii. 线性硬间隔支持向量机的限制可以写作
$y_i(\mathbf{w}^{\intercal}\mathbf{x_i} + b) \geq 1, \forall i \in \{1, ..., N\}$，解释其原因。
iv. 当训练集数据满足什么要求时，存在一个可行的$\mathbf{W}$？
v. 推导超平面H_1与H_2间距离（间隔）的表达式。
vi. 根据以上问题的答案，写出对应硬间隔支持向量机的优化问题的表达式。
vii. 对于以下数据点$\mathbf{x_1} = (1, 2, 3), \mathbf{x_2} = (4, 1, 2), \mathbf{x_3} = (-1, 2, -1)$及其

对应标签$y_1 = +1, y_2 = +1, y_3 = -1$，正确的 SVM 决策函数 $\mathbf{w}^\top \mathbf{x} + b = 0$为以下某一选项。试找出该选项。

1. $\mathbf{w} = [0.3, 0, 0.4]^\top, b = -0.4$
2. $\mathbf{w} = [0.2, 0, 0.4]^\top, b = -0.4$
3. $\mathbf{w} = [0.1, 0, 0.4]^\top, b = -0.4$
4. $\mathbf{w} = [0.4, 0, 0.2]^\top, b = -0.4$

i、图中红色线为分割超平面。



ii、超平面 H2 的表达式为$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b=-1$。

iii、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

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

所有有限制$y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\right)\geqslant 1,\quad i=1,2,\ldots,m$。

iv、正实例点集所构成的凸集与负实例点集所构成的凸集互不相交。

v、对于中间的直线为$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b=0$，通过平移使得该直线通过其中一个类的第一个样本，此时线性方程变为$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b=c$，两边同时除以$c$得到$\boldsymbol{w}'^{\mathrm{T}}\boldsymbol{x}+b'=1$，对于$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b=0$两边也同时除以$c$得到$\boldsymbol{w}'^{\mathrm{T}}\boldsymbol{x}+b'=0$，根据直线关于直线的对称得到另一条直线$\boldsymbol{w}'^{\mathrm{T}}\boldsymbol{x}+b'=-1$，最后用$\boldsymbol{w}$和$b$代替$\boldsymbol{w}'$和$b'$就得到了图中的形式。根据解析几何可知，

两条平行直线间的距离为$d=\frac{|c_1-c_2|}{\|\boldsymbol{w}\|}$，所以得到间隔为$\gamma=\frac{|1-(-1)|}{\|\boldsymbol{w}\|}=\frac{2}{\|\boldsymbol{w}\|}$。

vi、对应的优化问题表达式为：

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

vii、根据训练数据集构造约束最优化问题：

$$\begin{aligned}&\min_{\boldsymbol{w},\boldsymbol{b}}\frac{1}{2}(w_1^2+w_2^2+w_3^2)\\&s.t.\ \ w_1+2w_2+3w_3+b\geq 1\\&\qquad 4w_1+w_2+2w_3+b\geq 1\\&\qquad w_1-2w_2+w_3-b\geq 1\end{aligned}$$

则有

$$\nabla f(\boldsymbol{w})=\begin{pmatrix}w_1\\w_2\\w_3\end{pmatrix}\quad \nabla g_1(\boldsymbol{w})=\begin{pmatrix}1\\2\\3\end{pmatrix}\quad \nabla g_2(\boldsymbol{w})=\begin{pmatrix}4\\1\\2\end{pmatrix}\quad \nabla g_3(\boldsymbol{w})=\begin{pmatrix}1\\-2\\1\end{pmatrix}$$

$$\nabla f(\boldsymbol{w})-\lambda_1\nabla g_1(\boldsymbol{w})-\lambda_2\nabla g_2(\boldsymbol{w})-\lambda_3\nabla g_3(\boldsymbol{w})=0$$

得到

$$\begin{cases}\begin{pmatrix}w_1\\w_2\\w_3\end{pmatrix}-\lambda_1\begin{pmatrix}1\\2\\3\end{pmatrix}-\lambda_2\begin{pmatrix}4\\1\\2\end{pmatrix}-\lambda_3\begin{pmatrix}1\\-2\\1\end{pmatrix}=0\\\lambda_1\left(w_1+2w_2+3w_3+b-1\right)=0\\\lambda_2\left(4w_1+w_2+2w_3+b-1\right)=0\\\lambda_3\left(w_1-2w_2+w_3-b-1\right)=0\\\\\lambda_1\geq 0\\\lambda_2\geq 0\\\lambda_3\geq 0\end{cases}$$

求得此最优化问题的解为$\begin{matrix}w_1=0.2,w_2=0,w_3=0.4,b=-0.4\\\lambda_1=0.1,\lambda_2=0,\lambda_3=0.1\end{matrix}$，于是最大间隔分离超平

面为

$$0.2x_1 + 0.4x_3 - 0.4 = 0$$