支持向量机

- 一种二类分类模型

1 线性可分支持向量机与硬间隔最大化

1.1 线性可分支持向量机

假设给定一个特征空间上的训练数据集$T=\{(x_1,y_1),(x_2,y_2),\dots,(x_N,y_N)\}$。其中，$x_i\in\chi=\mathbb{R}^n$，$y_i\in\gamma=\{+1,-1\}$，$i=1,2,\dots,N$。

> 学习目标是在特征空间上找到一个分离超平面。分离超平面的方程$w\cdot x+b=0$。

其相应的分类决策函数为

$$f(x)=\text{sign}(w^*\cdot x+b^*)$$

称为线性可分支持向量机。

1.2 函数间隔和几何间隔

函数间隔就是利用类标记符号y和实例点x与$w\cdot x+b=0$之间的距离来判定分类的正确性和确信度。

$$\hat{\gamma}_i=y_i(w\cdot x_i+b)$$

虽然间隔函数能够表示分类预测的正确性和确信度。但是当模型参数变成$2w$和$2b$时，得到的超平面没有变化但函数间隔是原来的两倍了。

因此，我们应该对分离超平面的w进行约束，例如规范化（$||w||=1$），此时的间隔函数会变成几何函数。

$$\gamma_i=y_i(\frac{w}{||w||}\cdot x+\frac{b}{||w||})$$

其中，$||W||$为w的L_2范数。

1.3 间隔最大化

支持向量机学习的基本想法就是

1. 能够正确划分训练数据集
2. 几何间隔最大分离超平面

壹、最大间隔分离超平面

从几何间隔能发现，最大化的$\frac{\hat{\gamma}}{||w||}$可以找到最优模型，因此

$$\max_{w,b}\frac{\hat{\gamma}}{||w||}$$

$$s.t.\quad y_i(w\cdot x_i+b)\geq\hat{\gamma},\quad i=1,2,\dots,N$$

而其中取$\hat{\gamma}=1$代入上面的最优化问题，可以注意到最大化$\frac{1}{||w||}$和最小化$\frac{1}{2}||w||^2$是等价的，于是得到下面的线性可分支持向量机学习的最优化问题。

$$\min_{w,b} \frac{1}{2}||w||^2$$

$$s.t. \quad y_i(w \cdot x_i + b) - 1 \geq 0, \quad i = 1, 2, \ldots, N$$

而这个就是一个凸二次规划问题

$$\min_{w} f(w)$$

$$s.t. \quad \begin{aligned} g_i(w) &\leq 0, \quad i = 1, 2, \ldots, k \\ h_i(w) &= 0, \quad i = 1, 2, \ldots, l \end{aligned}$$

其中，目标函数$f(w)$和约束函数$g_i(w)$都是$\mathbb{R}^n$上的连续可微分的凸函数，约束函数$h_i(w)$是$\mathbb{R}^n$上的仿射函数。

> 仿射函数就是在保持原有函数的特征的情况下进行的线性变换，但是仿射函数是一个1阶函数（其中是由矩阵和向量组成的函数）

1. 什么是仿射函数？
2. 仿射函数、线性函数的区别？
3. 策略算法工程师之路-凸二次优化

贰、最大间隔分离超平面的存在唯一性

> 若训练数据集T线性可分，则可将训练数据集的样本点完全正确分开的最大间隔分离超平面存在且唯一

（1）存在性

上述已证明

（2）唯一性

1. 首先，证明w^*的唯一性。
   假设问题拥有两个解，分别为(w_1^*, b_1^*)和(w_2^*, b_2^*)。
   显然$||w_1^*|| = ||w_2^*|| = c$，其中$c$是一个常数。令$w = \frac{w_1^* + w_2^*}{2}$，$b = \frac{b_1^* + b_2^*}{2}$。从而有，

$$c \leq ||w|| \leq \frac{1}{2}||w_1^*|| + \frac{1}{2}||w_2^*|| = c$$

   上述表达式中的不等号能变成等号，即为$||w|| = \frac{1}{2}||w_1^*|| + \frac{1}{2}||w_2^*||$，从而有$w_1^* = \lambda w_2^*$，$|\lambda| = 1$。若$\lambda = -1$，则$w = 0$，解便不存在。因此必有$\lambda = 1$，即

$$w_1^* = w_2^*$$

2. 其次，证明$b*$的唯一性
   设x_1'和x_2'是集合$\{x_i | y_i = +1\}$和x_1''和x_2''是集合$\{x_i | y_i = -1\}$。

则由$b_1^* = -\frac{1}{2}(w^* \cdot x_1' + w^* \cdot x_1'')$, $b_2^* = -\frac{1}{2}(w^* \cdot x_2' + w^* \cdot x_2'')$，得

$$b_1^* - b_2^* = -\frac{1}{2}[w^* \cdot (x_1' - x_2') + w^* \cdot (x_2'' - x_2'')]$$

又因为

$$w^* \cdot x_2' + b_1^* \geq 1 = w^* \cdot x_1' + b_1'$$
$$w^* \cdot x_1' + b_2^* \geq 1 = w^* \cdot x_2' + b_2'$$

所以，$w^* \cdot (x_1' - x_2') = 0$。同理有$w^* \cdot (x_1'' - x_2'') = 0$。因此，

$$b_1^* - b_2^* = 0$$

由此可以得出w和b都是唯一的，解的唯一性得证

叁、支持向量和间隔边界

在线性可分情况下，训练数据集的样本中与分离超平面距离最近的样本的点实例称为支持向量。支持向量是使约束条件式等号成立的点，即：

$$y_i(w \cdot x_i + b) - 1 = 0$$

对$y_i = +1$的正例点，支持向量在超平面

$$H_1 = w \cdot x + b = 1$$

对$y_i = -1$的正例点，支持向量在超平面

$$H_2 = w \cdot x + b = -1$$

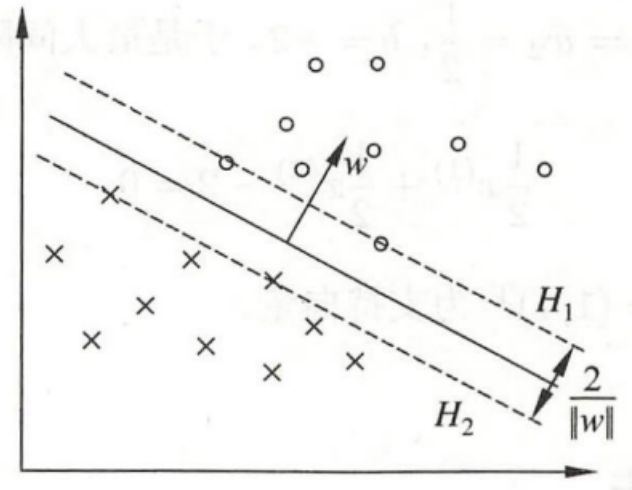

图 7.3　支持向量

1.4 学习的对偶算法

为了求解线性可分支持向量机的求优化问题，将它作为原始最优化问题，应用拉格朗日对偶性，通过求解对偶问题得到原始问题的最优解。
这样做的优点有：

1. 对偶问题更容易求解

2. 自然引入核函数，进而推广到非线性分类问题

首先，构建拉格朗日函数。为此，对每一个不等式约束引入拉格朗日函数乘子$\alpha_i \geq 0, i=1,2,\ldots,N$，得到拉格朗日函数：

$$L(w,b,\alpha)=\frac{1}{2}||w||^2-\sum_{i=1}^{N}\alpha_i y_i(w\cdot x_i+b)+\sum_{i=1}^{N}\alpha_i$$

其中，$\alpha=(\alpha_1,\alpha_2,\ldots,\alpha_N)^T$为拉格朗日乘子向量。

此时的最优化问题就变成了：

$$\max_{\alpha}\min_{w,b}L(w,b,\alpha)$$

分开解这个最优化问题：
（1）求$\min_{w,b}L(w,b,\alpha)$
将拉格朗日函数对w,b进行偏导并令其等于0

$$\nabla_w L(w,b,\alpha)=w-\sum_{i=1}^{N}\alpha_i y_i x_i=0$$

$$\nabla_b L(w,b,\alpha)=-\sum_{i=1}^{N}\alpha_i y_i=0$$

得

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

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

带入回原拉格朗日函数，可得：

$$\begin{aligned}L(w,b,\alpha)&=\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(x_i\cdot x_j)-\sum_{i=1}^{N}\alpha_i y_i((\sum_{j=1}^{N}\alpha_j y_j x_j)\cdot x_i+b)+\sum_{i=1}^{N}\\&=-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(x_i\cdot x_j)+\sum_{i=1}^{N}\alpha_i\end{aligned}$$

即

$$\min_{w,b}L(w,b,\alpha)=-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(x_i\cdot x_j)+\sum_{i=1}^{N}\alpha_i$$

（2）求$\max_{\alpha}(\min_{w,b}L(w,b,\alpha))$

$$\begin{aligned}\max_{\alpha} \quad & -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(x_i\cdot x_j)+\sum_{i=1}^{N}\alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^{N}\alpha_i y_i=0 \\ & \alpha_i\geq 0, \quad i=1,2,\dots,N\end{aligned}$$

转换成等价的对偶最优化问题：

$$\begin{aligned}\min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(x_i\cdot x_j)-\sum_{i=1}^{N}\alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^{N}\alpha_i y_i=0 \\ & \alpha_i\geq 0, \quad i=1,2,\dots,N\end{aligned}$$

定理1.2

设$\alpha^*=(\alpha_1^*,\alpha_2^*,\dots,\alpha_l^*)^T$ 是对偶最优化问题的的解，则存在下标j，使得$\alpha_j^*>0$，并可按下式求得原始最优化问题的解w^*,b^*：

$$w^*=\sum_{i=1}^{N}\alpha_i^* y_i x_i$$

$$b^*=y_j-\sum_{i=1}^{N}\alpha_i^* y_i(x_i\cdot x_j)$$

从支持向量机的原始公式能得出，

$$y_j(w^*\cdot x_j+b^*)-1=0$$

将w^*带入以上式子可以注意到$y_j^2=1$，即得分离超平面

$$\sum_{i=1}^{N}\alpha_i^* y_i(x\cdot x_i)+b^*=0$$

分类决策函数可以写成

$$f(x)=\text{sign}(\sum_{i=1}^{N}\alpha_i^* y_i(x\cdot x_i)+b^*)$$

这说明了分类决策函数只依赖于输入x和训练样本输入的内积。这个式子也称为线性可分支持向量机的对偶形式。

2 线性支持向量机与软间隔最大化

2.1 线性支持向量机

线性可分问题的支持向量机学习方法对于线性不可分训练数据是不适用的。为了解决这个问题需要改硬间隔最大化，使其成为软间隔最大化。

> 线性可分是数据点可以被一个超平面完全分离
> 线性不可分是数据点无法别一个超平面分离。
> 例子：XOR问题就是典型的线性不可分例子

假设给定一个特征空间上的训练数据集

$$T=\{(x_1,y_1),(x_2,y_2),\dots,(x_N,y_N)\}$$

其中，$x_i\in\mathcal{X}=\mathbb{R}^n$, $y_i\in y=\{+1,-1\}$, $i=1,2,\dots,N$, x_i为第i个特征向量, y_i为x_i的类标记。再假设训练数据集是线性不可分的
通常情况下是，训练数据集中有一些特异点，将这些特异点除去后，剩下大部分的样本点组成的集合是线性可分的。

线性不可分意味着某些样本点(x_i,y_i)不能满足函数间隔大于等于1的约束条件。为解决这个问题，可以对每个样本点(x_i,y_i)引进一个松弛变量 $\xi_i\geq 0$，使间隔函数加上松弛变量大于等于 1。这样玉树条件变为

$$y_i(w\cdot x_i+b)\geq 1-\xi_i$$

同时目标函数由原来的$\frac{1}{2}\|w\|^2$变成了

$$\frac{1}{2}\|w\|^2+C\sum_{i=1}^{N}\xi_i$$

这里，$C>0$称为惩罚函数，一般有应用问题决定，C 值大时对五分类的惩罚增大，反之惩罚减少。

为了得到最优解，线性不可分的线性支持向量问题的虚席问题编程如下凸二次规划问题：

$$\begin{aligned}
&\min_{w,b,\xi}\ \frac{1}{2}\|w\|^2+C\sum_{i=1}^{N}\xi_i\\
&\text{s.t.}\quad y_i(w\cdot x_i+b)\geq 1-\xi_i,\quad i=1,2,\dots,N\\
&\qquad \xi_i\geq 0,\quad i=1,2,\dots,N
\end{aligned}$$

2.2 学习的对偶算法

原始问题的对偶形式是

$$\begin{aligned}
&\min_{\alpha}\ \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(x_i\cdot x_j)-\sum_{i=1}^{N}\alpha_i\\
&\text{s.t.}\quad \sum_{i=1}^{N}\alpha_i y_i=0\\
&\qquad 0\leq\alpha_i\leq C,\quad i=1,2,\dots,N
\end{aligned}$$

原始最优化问题的拉格朗日函数是

$$L(w,b,\xi,\alpha,\mu)\equiv\frac{1}{2}\|w\|^2+C\sum_{i=1}^{N}\xi_i-\sum_{i=1}^{N}\alpha_i(y_i(w\cdot x_i+b)-1+\xi_i)-\sum_{i=1}^{N}\mu_i\xi_i$$

其中，$\alpha_i\geq 0,\mu_i\geq 0$**。**

对偶问题时拉格朗日函数的极大极小问题。首先求$L(w,b,\xi,\alpha,\mu)$**对**w,b,ξ **的极小，由**

$$\nabla_w L(w,b,\xi,\alpha,\mu)=w-\sum_{i=1}^{N}\alpha_i y_i x_i=0$$

$$\nabla_b L(w,b,\xi,\alpha,\mu)=-\sum_{i=1}^{N}\alpha_i y_i=0$$

$$\nabla_{\xi_i} L(w,b,\xi,\alpha,\mu)=C-\alpha_i-\mu_i=0$$

得

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

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

$$C-\alpha_i-\mu_i=0$$

带入原始最优化问题的拉格朗日函数得

$$\min_{w,b,\xi}L(w,b,\xi,\alpha,\mu)=-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(x_i\cdot x_j)+\sum_{i=1}^{N}\alpha_i$$

再对$\min_{w,b,\xi}L(w,b,\xi,\alpha,\mu)$**求**$\alpha$**的极大，即对偶问题：**

$$\max_{\alpha}-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(x_i\cdot x_j)+\sum_{i=1}^{N}\alpha_i$$

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

$$C-\alpha_i-\mu_i=0$$

$$\alpha_i\geq 0$$

$$\mu_i\geq 0,\quad i=1,2,\dots,N$$

从中消去μ_i**，从而只留下变量**α_i**，并将约束写成**

$$0\leq\alpha_i\leq C$$

再将对目标函数求极大转换为求极小，于是得到对偶问题。

2.3 支持向量

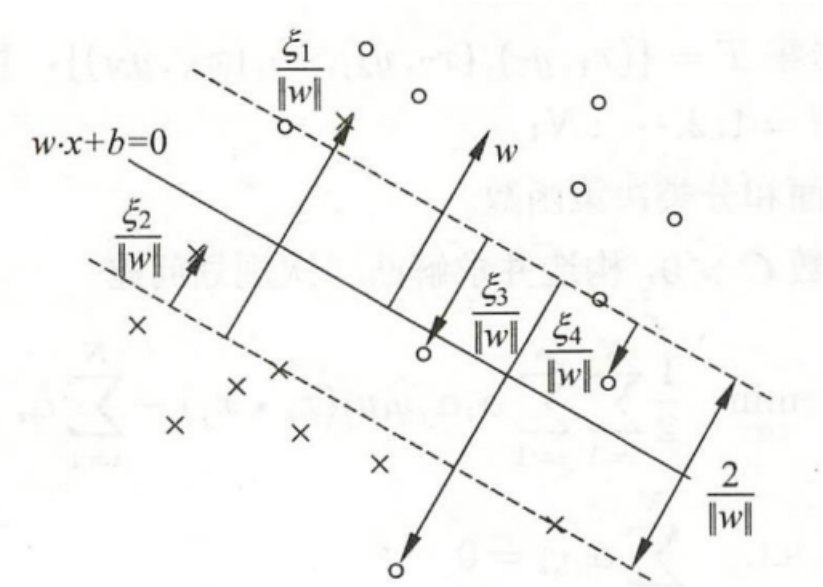

图 7.5　软间隔的支持向量

2.4 合页损失函数

对于线性支持向量机学习来说，其模型为分离超平面$w^* \cdot x + b^* = 0$及决策函数$f(x) = \text{sign}(w^* \cdot + b^*)$，其学习策略为软间隔最大化，学习算法为凸二次规划。

线性支持向量机学习还有另一个解释，就是最小化以下目标函数：

$$\sum_{i=1}^{N}[1 - y_i(w \cdot x_i + b)]_+ + \lambda \|w\|^2$$

目标函数的第1项是经验风险，函数

$$L(y(w \cdot x + b)) = [1 - y(w \cdot x + b)]$$

称为合页损失函数（Hinge Loss Function）。下标“+”表示以下正值的函数。

$$[z]_+ = \begin{cases} z, & z > 0 \\ 0, & z \leq 0 \end{cases}$$

定理1.4

线性支持向量机原始最优化问题：

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.} \quad & y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \ldots, N \\ & \xi_i \geq 0, \quad i = 1, 2, \ldots, N \end{aligned}$$

其等价最优化问题为

$$\min_{w,b}\sum_{i=1}^{N}[1-y_i(w\cdot x_i+b)]_+ +\lambda\|w\|^2$$

3 非线性支持向量机与核函数

在处理线性分类问题，线性分类支持向量机是一种非常有效的方法。但是，在面对到非线性数据集的时候就会分棘手，这是可以使用非线性支持向量机。

这里叙述的非线性支持向量机的特点是利用了核函数。

3.1 核函数

> 核函数（Kernel Function）是在机器学习和统计学中经常用到的一种数学函数。它用于在高维空间中进行低维数据的非线性映射，从而使得数据在高维空间中变得更易处理。核函数通常用来在支持向量机（Support Vector Machines）和核岭回归（Kernel Ridge Regression）等算法中进行非线性特征映射。
>
> 核函数的主要作用是将输入数据映射到一个高维空间，使得在这个高维空间中的数据线性可分或者更易处理。这样一来，在原始空间中线性不可分的问题，在高维空间中可能就是线性可分的了

1. 非线性分类问题

非线性分类问题是指通过利用非线性模型才能很好地进行分类的问题

假设给定一个训练数据集$T=\{(x_1,y_1),(x_2,y_2),\dots,(x_n,y_n)\}$，其中实例$x_i$属于输入空间，$s_i\in\mathcal{X}=\mathbf{R}^n$，对应的标记有两类$y_i\in\mathcal{Y}=\{-1,+1\},i=1,2,\dots,N$。如果能用$\mathbf{R}^n$中的一个超曲面将正负例正确分开，则称这个问题为非线性可分问题。

非线性问题往往不好求解，所以希望能用线性分类问题的方法解决这个问题。因此采取的方法是进行一个非线性变换，将非线性问题变成线性问题。

设原空间为$\mathcal{X}\subset\mathbf{R}^2$，$x=(x^{(1)},x^{(2)})^T\in\mathcal{X}$，新空间为$\mathcal{Z}\subset\mathbf{R}^2$，$z=(z^{(1)},z^{(2)})^T\in\mathcal{Z}$，定义从元空间到新空间的变换（映射）：

$$z=\phi(x)=((x^{(1)})^2,(x^{(1)})^2)^T$$

经过变换$z=\phi(x)$，原空间$\mathcal{X}\subset\mathbf{R}^2$变换为新空间$\mathcal{Z}\subset\mathbf{R}^2$，由

$$w_1(x^{(1)})^2+w_2(x^{(2)})^2+b=0$$

变换成了新空间的直线

$$w_1z^{(1)}+w_2z^{(2)}+b=0$$

2. 核函数的定义

设 $\mathcal{X}$ 是输入空间，又设 $\mathcal{H}$ 为特征空间，如果存在一个从 $\mathcal{X}$ 到 $\mathcal{H}$ 的映射

$$\phi(x):\mathcal{X}\to\mathcal{H}$$

使得对所有$x, z \in \mathcal{X}$，函数$K(x, z)$满足条件

$$K(x, z) = \phi(x) \cdot \phi(z)$$

则称$K(x, z)$为核函数，$\phi(x)$为映射函数，式中$\phi(x) \cdot \phi(z)$为$\phi(x)$和$\phi(z)$的内积

3. 核技巧在支持向量机中的应用

我们可以注意到线性支持向量机的对偶问题中，无论是目标函数还是决策函数都只涉及输入实例与实例之间的内积。

在对偶问题的目标函数中的内积$x_i \cdot x_j$可以用核函数$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$来代替。此时对偶问题的目标函数成为

$$W(\alpha) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{N} \alpha_i$$

同样，分类的决策函数中的内积也能使用核函数进行替代，成为：

$$\begin{aligned} f(x) &= \text{sign}(\sum_{i=1}^{N_s} \alpha_i^* y_i \phi(x_i) \cdot \phi(x) + b^*) \\ &= \text{sign}(\sum_{i=1}^{N_s} \alpha_i^* y_i K(x_i, x) + b^*) \end{aligned}$$

这等价于经过映射函数 ϕ 将原来的输入空间换到一个新的特征空间，将输入空间中的内积换成特征空间中内积。

也就是说，在核函数$K(x, z)$给定的条件下，可以利用解线性分类问题的方法求解非线性分类问题的支持向量机。学习是隐式地在特征空间进行的，不需要显式地定义特征空间和映射函数，这就成为核技巧。

3.2 正定核

假设 $K(x, z)$ 是定义在 $\mathcal{X} \times \mathcal{X}$ 上的对称函数，并且对任意的 $x_1, x_2, \ldots, x_m \in \mathcal{X}$，$K(x, z)$ 关于 $x_1, x_2, \ldots, x_m$ 的Gram矩阵是半正定的。可以根据函数$K(x, z)$，构成一个希尔伯特空间，其步骤是：

1. 定义映射，构成向量空间 $\mathcal{S}$

先定义映射

$$\phi : x \to K(\cdot, x)$$

根据这一映射，对任意$x_i \in \mathcal{X}$, $\alpha_i \in \mathbf{R}$, $i = 1, 2, \ldots, m$，定义线性组合

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i K(\cdot, x_i)$$

由于集合$\mathcal{S}$是由线性组合，因此其加法和乘法运算是封闭的，所以$\mathcal{S}$构成一个向量空间。

2. 在 $\mathcal{S}$ 上定义内积，使其成为内积空间

在 $\mathcal{S}$ 上定义一个运算 $*$：对任意 $f, g \in \mathcal{S}$，

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i K(\cdot, x_i)$$

$$g(\cdot) = \sum_{j=1}^{m} \beta_i K(\cdot, z_j)$$

定义运算$*$

$$f * g = \sum_{i=1}^{m} \sum_{j=1}^{l} \alpha_i \beta_l K(x_i, z_j)$$

证明... $*$ 为向量空间 $\mathcal{S}$ 的内积。因此 $\mathcal{S}$ 是一个内积空间，所以

$$f \cdot g = \sum_{i=1}^{m} \sum_{j=1}^{l} \alpha_i \beta_l K(x_i, z_j)$$

3. 将内积空间 $\mathcal{S}$ 完备化为希尔伯特空间

现在将内积空间 $\mathcal{S}$ 完备化。由内积可得范数：

$$\|f\| = \sqrt{f \cdot f}$$

因此，$\mathcal{S}$ 是一个赋范向量空间。根据泛函分析理论，对于不完备的赋范向量空间 $\mathcal{S}$，一定可以使之完备化，得到完备的赋范向量空间 $\mathcal{H}$。

这一希尔伯特空间 $\mathcal{H}$ 称为再生希尔伯特空间。这是由核K具有再生性，既满足：

$$K(\cdot, x) \cdot f = f(x)$$

及

$$K(\cdot, x) \cdot K(\cdot, z) = K(x, z)$$

称为再生核。

4. 正定核的充要条件

设$K : \mathcal{X} \times \mathcal{X} \to \mathbf{R}$是对称函数，则$K(x, z)$为正定核函数的充要条件是对任意$x_i \in \mathcal{X}, i = 1, 2, \ldots, m$，$K(x, z)$对应的Gram矩阵：

$$K = [K(x_i, x_j)]_{m \times m}$$

是半正定矩阵（Positive Semi-Definite）。

3.3 常用核函数

1. 多项式核函数（Polynomial Kernel Function）

$$K(x, z) = (x \cdot z + 1)^p$$

对应的支持向量机是一个 p 次多项式分类器。在此情形下，分类决策函数成为

$$f(x) = \text{sign}(\sum_{i=1}^{N_s} a_i^* y_i (x_i \cdot x + 1)^p + b^*)$$

2. 高斯核函数（Gaussian Kernel Function）

$$K(x, z) = \exp(-\frac{\|x - z\|^2}{2\sigma^2})$$

对应的支持向量机是高斯径向基函数（Radial Basis Function）分类器。在此情形下，分类决策函数为

$$f(x) = \text{sign}(\sum_{i=1}^{n_s} a_i^* y_i \exp(-\frac{\|x - z\|^2}{2\sigma^2}) + b^*)$$

3. 字符串核函数（String Kernel Function）

核函数不仅可以定义在欧式空间上，还可以定义在离散数据的集合上。

考虑一个有限字符表$\sum$。字符串s是从$\sum$中取出的有限个字符的序列，包括空字符串。字符串s的长度用$|s|$表示，它的元素记作$s(1)s(2)\ldots s(|s|)$。两个字符串s和t的连接记作st。所有长度为n的字符串的集合记作$\sum^n$，所有字符串的集合记作

$$\sum^* = \cup_{n=0}^{\inf} \sum^n$$

考虑字符串s的子串u。给定一个指标序列$i = (i_1, i_2, \ldots, i_{|u|})$，$1 \le i_1 < i_2 < \ldots < i_{|u|} \le |s|$，$s$的子串定义为$u = s(i) = s(i_1)s(i_2)\ldots s(i_{|u|})$，其长度记作$l(i) = i_{|u|} - i_1 + 1$。如果$i$是连续的，则了$l(i) = |u|$；否则，$l(i) > |u|$。

假设$\mathcal{S}$是长度大于或等于n的字符串的集合，s是$\mathcal{S}$的元素。现在建立字符串集合$\mathcal{S}$到特征空间$\mathcal{H}_n = R^{\sum^n}$的映射$\phi_n(s)$。$R^{\sum^n}$表示定义在$\sum^n$上的实数空间，其每一维对应的一个字符串$u \in \sum^n$，映射$\phi_n(s)$将字符串$s$对应于空间$R^{\sum^n}$的一个向量，其在$u$维上的取值为

$$[\phi_n(s)]_u = \sum_{i:s(i)=u} \lambda^{l(i)}$$

这里，$0 < \lambda \le 1$是一个衰减参数，$l(i)$表示字符串i的长度，求和在s中所有与u相同的子串上进行。

两个字符串s和t上的字符串核函数是基于映射ϕ_n的特征空间的内积：

$$\begin{aligned} k_n(s, t) &= \sum_{u \in \sum^n} [\phi_n(s)]_u [\phi_n(t)]_u \\ &= \sum_{u \in \sum^n} \sum_{(i,j):s(i)=t(j)=u} \lambda^{l(i)} \lambda^{l(j)} \end{aligned}$$

字符串核函数$k_n(s, t)$给出了字符串s和t中长度等于n的所有子串组成的特征空间的余弦相似度（Cosine Similarity）。

直观上，两个字符串相同的子串越多，它们就越相似，字符串核函数的值就越大。字符串核函数可以由动态规划快速地计算。

3.4 非线性支持向量机分类机

综上所述，利用和函数，可以将非线性分类问题应用线性分类问题的方法解决。将线性支持向量机扩展到非线性支持向量机，只需要将线性支持向量机对偶形式中的内积换成核函数。

从非线性分类训练集，通过核函数与软间隔最大化，或凸二次规划，学习得到的分类决策函数

$$f(x) = \text{sign}(\sum_{i=1}^{N} \alpha_i^* y_i K(x, x_i) + b^*)$$

称为非线性支持向量机，$K(x, z)$是正定核函数。