第7章 SVM

支持向量机（support vector machines,SVM）的基本模型定义是在特征空间上的间隔最大的线性分类器，它的学习策略就是**间隔最大化。**

支持向量机的模型由简到难分为：

```
线性可分支持向量机 —> 硬间隔最大化
线性支持向量机 —> 软间隔最大化
非线性支持向量机 —> 核函数
```

7.1 线性可分支持向量机

二分类问题中，这里我们用 `|w*x+b|`的大小表示点 x 离超平面的远近，即确信度； `w*x+b`的符号与类别标记 y（取 +1 或者 -1）是否一致表示正确性。

支持向量机是在特征空间中学习的，当数据线性可分时，存在多个分离超平面。感知机用误分类最小的策略，求得分离超平面，但是解有无穷多个，SVM用间隔最大化（硬间隔最大化），解就是唯一的。

函数间隔

定义式为 $y_i(w*x_i+b)$，

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

表示分类 **正确性** 及 **确信度。** 但是只使用函数间隔来选择分离超平面时，只要成比例的改变 w 和 b，例如将他们改为 2w 和 2b，这样超平面并没有改变，但是函数间隔却变成原来的 2 倍。

几何间隔

针对上述问题，我们对函数间隔加上规范化这一约束项，使`||w|| = 1`，这就成了几何间隔。

$$\gamma_i = y_i\left(\frac{w}{\|w\|} \cdot x_i + \frac{b}{\|w\|}\right)$$

间隔最大化

我们希望的得到一个最大间隔分离超平面，使每个训练样本点距离这个超平面的距离都至少是 γ 用**几何间隔**来表示就是：

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

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

这里我们取函数间隔为 1 ，代入上述公式，即为 `1/||w||` 最大化问题，其和 $1/2||w||^2$ 最小化等价。

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

$$\text{s.t.} \quad y_i(w \cdot x_i + b) - 1 \geqslant 0, \quad i = 1, 2, \cdots, N$$

以上述问题进行求最优解 w^*，和 b^*,即是**线性可分支持向量机**的最优化问题(**凸二次优化问题**)。 这时使用拉格朗日对偶性，先对 w 和 b 求极小，然后对引入参数求极大，最后将引入参数代入 w 和 b 得到最终结果 w^*，和 b^* 。

在线性可分的条件下，支持向量是指训练集样本点与分离超平面距离最近的样本点的实例，二分类中支持向量在正例方的超平面 H_1 与负例方超平面 H_2 的间隔为：`2/||w||`

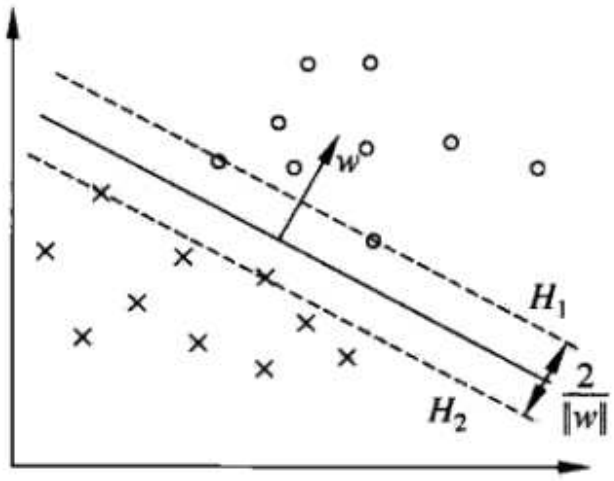

图 7.3　支持向量

由上图可见，**支持向量是由较少的重要的训练样本决定的，移动支持向量将改变解，移动支持向量以外的数据点，甚至删除它们，对 SVM 最后的结果都没有影响。**

7.2 线性支持向量机

这里倘若数据不是线性可分的，上述方法将不再适用，为了解决这一问题，引入松弛变量 ε_i 的概念，将对间隔的约束条件改为：

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

对每一个松弛变量 ε_i 都支付一个代价，最后的目标函数为：

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

这里的 C 是惩罚系数，一般视应用情况而定。 对上面的问题进行间隔最大化求解，就是**软间隔最大化。** 最后求解出来的 **w 是唯一解， b 则不是唯一解，b 的解存在于一个区间。**

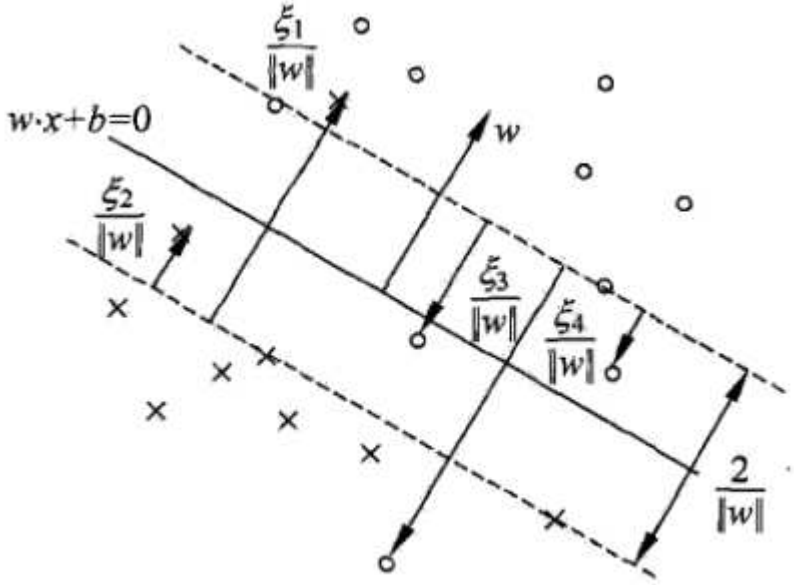

图 7.5　软间隔的支持向量

可以看到，软间隔的支持向量或者在间隔边界上，或者在间隔边界与分离超平面之间，或者在分离超平面误分一侧。

线性支持向量机学习的另一种解释就是用合页损失函数（hinge loss function）来解释,即最小化一下目标函数：

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

其中 $[z]_+$ 即为合页损失函数。

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

这里合页损失函数是 0-1 损失函数的上界，因为 0-1 损失函数不是连续可导的，所以这里用其上界，合页损失函数构建目标函数，便于优化。

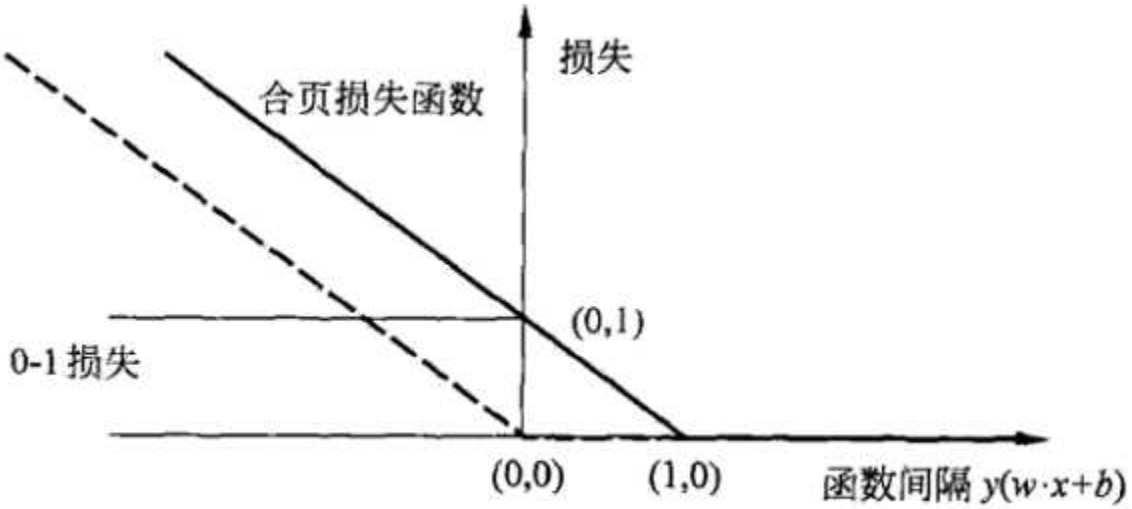

图 7.6　合页损失函数

7.3 非线性支持向量机

核函数

核函数表示将输入空间（欧式空间 R^n 的子集或离散集合）映射到特征空间（希尔伯特空间）得到的向量之间的内积。

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

K(x,z) 是核函数，Φ(x) 为映射函数。 在线性支持向量机中，目标函数还是决策函数，只涉及到输入实例与实例之间的内积，在这里我们用核函数的内积代替就可以在支持向量机中使用核函数了。

函数需要是正定函数才能称为核函数，正定是充要条件。

常用的核函数

- 多项式核函数

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

- 高斯核函数

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

- 字符串核函数

这是用在离散数据集合上的，如本文分类，信息检索，生物信息学等方向。

非线性支持向量机

利用上述核技巧，利用所给非线性分类训练集，结合核函数与软间隔最大化来学习分类决策函数

7.4 序列最小最优化算法

正常对上述向量机求解过程进行学习，当样本里很大时是很慢的，快速的学习算法称为关键，这里序列最小最优算法（SMO）就是这类算法，先固定参数 α_1，α_2，固定 α_3 —— α_N,利用 KKT 来确定 α_2，α_1 也随之确定，依次类推，知道整体所有变量满足 KKT 条件为止。

```
from sklearn.svm import SVC
clf = SVC()
clf.fit(X_train, y_train)
print(clf.score(X_test, y_test))
```

> (C=1.0, kernel='rbf', degree=3, gamma='auto', coef0=0.0, shrinking=True, probability=False,tol=0.001, cache*size=200, class*weight=None, verbose=False, max*iter=-1, decisionfunction*shape=None,random*state=None)
>
> 参数：
>
> C：C-SVC的惩罚参数C?默认值是1.0 C越大，相当于惩罚松弛变量，希望松弛变量接近0，即对误分类的惩罚增大，趋向于对训练集全分对的情况，这样对训练集测试时准确率很高，但泛化能力弱。C值小，对误分类的惩罚减小，允许容错，将他们当成噪声点，泛化能力较强。
>
> kernel ： 核函数，默认是rbf，可以是‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’
>
> – 线性：u'v
>
> – 多项式：(gammau'v + coef0)^degree
>
> – RBF函数：exp(-gamma|u-v|^2)
>
> – sigmoid：tanh(gammau'v + coef0)
>
> degree ： 多项式poly函数的维度，默认是3，选择其他核函数时会被忽略。
> gamma ： ‘rbf’,‘poly’ 和 "sigmoid" 的核函数参数。默认是’auto’，则会选择1/n*features*
> *coef0 ： 核函数的常数项。对于‘poly’和‘sigmoid’有用*
> *probability ： 是否采用概率估计？.默认为False*
> *shrinking ： 是否采用shrinking heuristic方法，默认为true*
> *tol ： 停止训练的误差值大小，默认为1e-3*
> *cache_size ： 核函数cache缓存大小，默认为200*
> *class_weight ： 类别的权重，字典形式传递。设置第几类的参数C为weight*C(C-SVC中的C)*
> *verbose ： 允许冗余输出？*
> *max_iter ： 最大迭代次数。-1为无限制。*
> *decision_function*shape ： ‘ovo’, ‘ovr’ or None, default=None3
> random_state ： 数据洗牌时的种子值，int值
> 主要调节的参数有：C、kernel、degree、gamma、coef0。