第5章　支持向量机

引　　言

分类作为数据挖掘领域中一项非常重要的任务，其目的是学会一个分类函数或分类模型(或分类器)，而支持向量机本身就是一种监督式学习的方法，它广泛应用于统计分类及回归分析中。支持向量机(support vector machine，SVM)是 Vapnik 等于 1995 年首先提出的，它在解决小样本、非线性及高维模式识别中表现出许多特有的优势，并推广到人脸识别、行人检测、文本自动分类等其他机器学习问题中。支持向量机方法是建立在统计学习理论的 VC 维理论和结构风险最小原理基础上的，根据有限的样本信息在模型的复杂性和学习能力之间寻求最佳折中，以求获得最好的推广能力。

5.1　支持向量机概述

支持向量机是 20 世纪 90 年代中期发展起来的基于统计学习理论的一种机器学习方法，通过寻求结构化风险最小化提高学习机泛化能力，实现经验风险和置信范围的最小化，从而达到在统计样本量较少的情况下也能获得良好统计规律的目的。其基本模型定义为特征空间上的间隔最大的线性分类器，即支持向量机的学习策略便是间隔最大化，最终可转化为一个凸二次规划问题的求解。

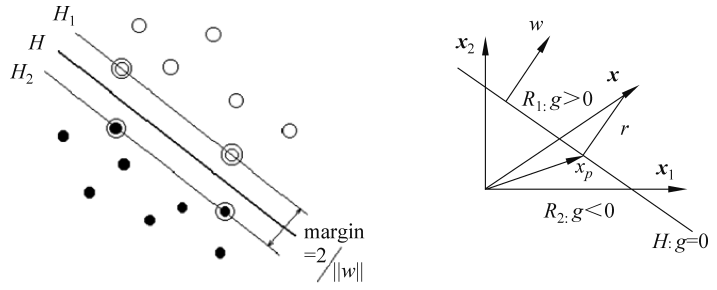
5.1.1　margin 最大化

首先考虑两类线性可分的情况，如图 5-1 所示。两类训练样本分别为实心点与空心点，SVM 的最优分类面要求分类线不但能将两类正确分开，即训练错误率为 0，且使得分类间隔(margin)最大。在图 5-1(a)中，H 为把两类训练样本正确分开的分类线，H_1、H_2 为这两类训练样本中距离 H 最近，且平行于 H 的直线，则 margin 即为 H_1、H_2 之间的垂直距离。

设训练数据集为$(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)$，$\boldsymbol{x}=(x_1,x_2,\cdots,x_n)\in\mathbb{R}^n$，$y\in\{+1,-1\}$。线性判别函数设为

$$g(\boldsymbol{x})=(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x})+b \tag{5-1}$$

其中，$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$ 为 $\boldsymbol{w}$ 与 $\boldsymbol{x}$ 的内积。分类面方程为$(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x})+b=0$。将判别函数进行归一化，使两类所有的样本都满足$|g(\boldsymbol{x})|\geqslant 1$，使 $y=-1$ 时，$g(\boldsymbol{x})\leqslant -1$；$y=1$ 时，$g(\boldsymbol{x})\geqslant 1$。其中，离分类



(a)　　(b)

图 5-1　最优分类面示意图

面最近的样本的$|g(\boldsymbol{x})|=1$。通常，还可以定义$g(\boldsymbol{x})>0$时，$\boldsymbol{x}$被分为ω_1类；$g(\boldsymbol{x})<0$时，$\boldsymbol{x}$被分为ω_2类；$g(\boldsymbol{x})=0$时为决策面。设$\boldsymbol{x}_1$，$\boldsymbol{x}_2$是决策面上的两点，于是就有

$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_1+b=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_2+b,\quad \text{即 } \boldsymbol{w}^{\mathrm{T}}(\boldsymbol{x}_1-\boldsymbol{x}_2)=0 \tag{5-2}$$

可以看出，$\boldsymbol{w}$与$\boldsymbol{x}_1-\boldsymbol{x}_2$正交，$\boldsymbol{x}_1-\boldsymbol{x}_2$即决策面的方向，所以$\boldsymbol{w}$就是决策面的法向量。

我们的目标是求分类间隔最大的决策面，首先表示出分类间隔 margin。空间任意$\boldsymbol{x}$（见图 5-1）可表示为

$$\boldsymbol{x}=\boldsymbol{x}_p+r\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} \tag{5-3}$$

在式(5-3)中，$\boldsymbol{x}_p$是$\boldsymbol{x}$在H上的投影向量（见图 5-1(b)），r是x到H的垂直距离。$\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$表示$\boldsymbol{w}$方向上的单位向量，将式(5-3)代入式(5-1)中，可得

$$g(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\left(\boldsymbol{x}_p+r\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}\right)+\boldsymbol{b}_0=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_p+\boldsymbol{b}_0+r\frac{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}}{\|\boldsymbol{w}\|}=r\|\boldsymbol{w}\| \tag{5-4}$$

$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_p+b=0$，则$r$表示为

$$r=\frac{|g(\boldsymbol{x})|}{\|\boldsymbol{w}\|} \tag{5-5}$$

由以上分析可知，距离分类面最近的样本满足$|g(\boldsymbol{x})|=1$，这样分类间隔就为

$$\text{margin}=2r=\frac{2}{\|\boldsymbol{w}\|} \tag{5-6}$$

因此，若要求 margin 的值最大，即求$\|\boldsymbol{w}\|$或$\|\boldsymbol{w}\|_2$的最小值。

5.1.2　支持向量机优化

因为要求所有训练样本正确分类，即需要满足如下的条件：

$$y_i[(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x})+b]-1\geqslant 0,\quad i=1,2,\cdots,n \tag{5-7}$$

在以上条件下，求使$\|\boldsymbol{w}\|^2$最小的分类面。而H_1、H_2上的训练样本就是式(5-7)中等号成立的那些样本，叫作支持向量(support vector)，在图 5-1(a)中用圆圈标记。所以，最优分类面问题可以表示为如下的约束优化问题：

$$\text{Min}\Phi(\boldsymbol{w})=\frac{1}{2}\|\boldsymbol{w}\|^2=\frac{1}{2}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}) \tag{5-8}$$

约束条件为

$$y_i[(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x})+b]-1\geqslant 0,\quad i=1,2,\cdots,n$$

构造 Lagrange 函数，即

$$L(\boldsymbol{w},b,a)=\underset{\boldsymbol{w}}{\mathrm{Min}}\underset{a}{\mathrm{Max}}\frac{1}{2}\|\boldsymbol{w}\|^2-\sum_{i=1}^{n}a_i(y_i((\boldsymbol{x}_i,\boldsymbol{w})+b)-1) \tag{5-9}$$

其中，a_i 为 Lagrange 系数，这里就对 $\boldsymbol{w}$ 和 b 求 Lagrange 函数的极小值。将 L 对 $\boldsymbol{w}$ 求偏导数，并令其等于 $\mathbf{0}$，可得

$$\nabla_w L(\boldsymbol{w},b,a)=\boldsymbol{w}-\sum a_i y_i \boldsymbol{x}_i=\mathbf{0}$$

得出

$$\boldsymbol{w}^*=\sum a_i y_i \boldsymbol{x}_i \tag{5-10}$$

将式(5-10)代入 L 的方程，就得到了 L 关于 $\boldsymbol{w}$ 的最优解，即

$$L(\boldsymbol{w}^*,b,a)=-\frac{1}{2}\sum_i\sum_j a_i a_j y_i y_j(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j)-\sum_i a_i y_i b+\sum_i a_i \tag{5-11}$$

再对 b 求偏导，即

$$\nabla_b L(\boldsymbol{w},b,a)=\sum\nolimits_i a_i y_i=0 \tag{5-12}$$

将式(5-12)代入 L 关于 $\boldsymbol{w}$ 的最优解，就可以得到 L 关于 $\boldsymbol{w}$ 和 b 的最优解，即

$$L(\boldsymbol{w}^*,b^*,a)=-\frac{1}{2}\sum_i\sum_j a_i a_j y_i y_j(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j)+\sum_i a_i \tag{5-13}$$

下面寻找原始问题的对偶问题并求解，原始问题的对偶问题为

$$\begin{aligned}&\mathrm{Max}\ Q(a)=-\frac{1}{2}\sum_i\sum_j a_i a_j y_i y_j(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j)+\sum_i a_i\\&\mathrm{s.t.}\sum_{i=1}^{n}a_i y_i=0,\quad a_i\geqslant 0,i=1,2,\cdots,n\end{aligned} \tag{5-14}$$

若 a_i^* 为最优解，则可求得

$$\boldsymbol{w}^*=\sum_{i=1}^{n}a_i^* y_i \boldsymbol{x}_i \tag{5-15}$$

可以看出，这是不等式约束的二次函数极值问题，满足 KKT (karush-kuhn-tucker)条件。这样，使得式(5-15)最大化的 $\boldsymbol{w}^*$ 和 b^* 需要满足

$$\sum_{i=1}^{n}a_i(y_i[(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x})+b]-1)=0 \tag{5-16}$$

而对于多数样本，它们不在离分类面最近的直线上，即 $y_i[(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i)+b]-1>0$，从而一定有对应的 $a_i=0$，也就是说，只有在边界上的数据点(支持向量)才满足

$$\begin{aligned}&y_i[(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x})+b]-1=0\\&a_i\neq 0,i=1,2,\cdots,n\end{aligned} \tag{5-17}$$

它们只是全体样本中很少的一部分，相对于原始问题大幅减少了计算的复杂度。最终求得上述问题的最优分类函数，即

$$f(\boldsymbol{x})=\mathrm{sgn}\{(\boldsymbol{w}^{*\mathrm{T}}\boldsymbol{x})+b^*\}=\mathrm{sgn}\Big\{\sum a_i{}^* y_i(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x})+b^*\Big\} \tag{5-18}$$

其中，sgn()为符号函数。由于非支持向量对应的 a_i 都为 0，因此式(5-18)中的求和实际上只对支持向量进行。b^* 是分类的阈值，可以由任意一个支持向量用式(5-16)求得。这样就求得了在两类线性可分情况下的 SVM 分类器。然而，并不是所有的两类分类问题都是线性可分的。对于非线性问题，SVM 设法将它通过非线性变换转化为另一空间中的线性问题，在这个变换空间求解最优的线性分类面。而这种非线性变换可以通过定义适当的内积函数，即核函数实现。目前得到的常用核函数主要有多项式核、径向基核以及 Sigmoid 核，其参数的选择对最终的分类效果也有较大影响。也就是说，以前新来的要分类的样本首先根据 $\boldsymbol{w}$ 和 b 做一次线性运算，然后看求的结果是大于 0 还是小于 0，由此判断是正例还是负例。现在有了 a_i，不需要求出 $\boldsymbol{w}$，只将新来的样本和训练数据中的所有样本做内积和即可。从 KKT 条件中得到，只有支持向量的 $a_i>0$，其他情况 $a_i=0$。因此，只求新来的样本和支持向量的内积，然后运算即可。

核函数概念的提出使 *SVM* 完成了向非线性分类的转变。观察图 5-2，把横轴上端点 a 和 b 之间红色部分里的所有点定为正类，两边的黑色部分里的点定为负类。试问，能找到一个线性函数用来把两类正确分开么？答案是不能，因为二维空间里的线性函数就是直线，显然找不到符合条件的直线。但可以找到一条曲线，如图 5-3 中的这条曲线。

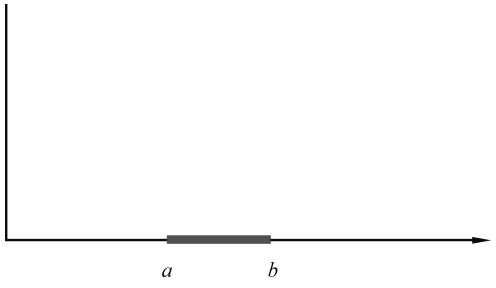


图 5-2　二维空间线性不可分的例子(见彩图)

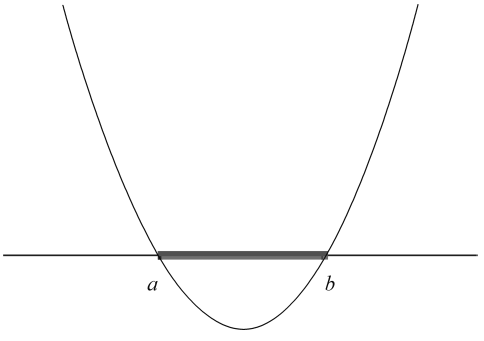


图 5-3　二维空间核函数举例

显然，通过点在这条曲线的上方还是下方就可以判断点所属的类别。这条曲线就是大家熟知的二次曲线，它的函数表达式可以写为

$$g(\boldsymbol{x})=c_0+c_1\boldsymbol{x}+c_2\boldsymbol{x}^2$$

那么，首先需要将特征 $\boldsymbol{x}$ 扩展到三维$(1,x,x^2)$，然后寻找特征和结果之间的模型。通常将这种特征变换称作特征映射(feature mapping)。映射函数称作 $\Phi()$，在这个例子中，$\Phi(\boldsymbol{x})=\begin{bmatrix}1\\x\\x^2\end{bmatrix}$，我们希望将得到的特征映射后的特征应用于 SVM 分类，而不是最初的特征。这样，需要将前面 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b$ 公式中的内积从$<x^{(i)},\boldsymbol{x}>$映射到$<\Phi(x^{(i)}),\Phi(\boldsymbol{x})>$。由式(5-15)可知，线性分类用的是原始特征的内积$<x^{(i)},\boldsymbol{x}>$，在非线性分类时只选用映射后的内积即可，至于选择何种映射，需要根据样本特点和分类效果选择。

然而，为了进行非线性分类，特征映射后会使维度大幅增加，对运算速度是一个极大的

挑战，而核函数很好地解决了这个问题。将核函数形式化定义，如果原始特征内积是$<\boldsymbol{x},\boldsymbol{z}>$，映射后为$<\Phi(\boldsymbol{x}),\Phi(\boldsymbol{z})>$，那么定义核函数(kernel)为$k(\boldsymbol{x},\boldsymbol{z})=\Phi(\boldsymbol{x})^{\mathrm{T}}\Phi(\boldsymbol{z})$。下面举例说明这个定义的意义。令$K(\boldsymbol{x},\boldsymbol{z})=(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{z})^2$，展开后得到

$$K(\boldsymbol{x},\boldsymbol{z})=(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{z})^2=\left(\sum_{i=1}^{m}x_iz_i\right)\left(\sum_{j=1}^{m}x_jz_j\right)=\sum_{i=1}^{m}\sum_{j=1}^{m}x_iy_jz_iz_j$$

$$=\sum_{i=1}^{m}\sum_{j=1}^{m}(x_ix_j)(z_iz_j)=\Phi(\boldsymbol{x})^{\mathrm{T}}\Phi(\boldsymbol{z}) \tag{5-19}$$

这里的$\Phi()$指的是如下的映射(维数$n=3$时)：

$$\Phi(\boldsymbol{x})=\begin{bmatrix}x_1x_1\\x_1x_2\\x_1x_3\\x_2x_1\\x_2x_2\\x_2x_3\\x_3x_1\\x_3x_2\\x_3x_3\end{bmatrix}$$

也就是说，核函数$K(\boldsymbol{x},\boldsymbol{z})=(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{z})^2$只能在选择类似这样的$\Phi()$作为映射才等价于映射后特征的内积。此处用三维向量的核函数代表了九维向量的内积，大幅减小了运算量。核函数的形式有很多，判断核函数有效性的是Mercer定理，常用的核函数有以下几种。

多项式核函数：$K(\boldsymbol{x},\boldsymbol{z})=[\boldsymbol{x}^{\mathrm{T}}\boldsymbol{z}+1]^q$，

径向基函数：$K(\boldsymbol{x},\boldsymbol{z})=\exp\left(-\dfrac{|\boldsymbol{x}-\boldsymbol{z}|^2}{\sigma^2}\right)$，

S形函数：$K(\boldsymbol{x},\boldsymbol{z})=\tanh(v(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{z})+c)$。

对核函数进行概括，即不同的核函数用原始特征不同的非线性组合拟合分类曲面。SVM还有一个问题，回到线性分类器，在训练线性最小间隔分类器时，如果样本线性可分，则可以得到正确的训练结果；但如果样本线性不可分，那么目标函数无解，会出现训练失败。而在实际应用中，这种现象是很常见的，所以SVM引入了松弛变量，即

$$\Phi(\boldsymbol{w},\boldsymbol{\xi})=\frac{1}{2}\|\boldsymbol{w}\|^2+c\sum_{i=1}^{l}\xi$$

$$\boldsymbol{w}^T\boldsymbol{x}_i+b\geqslant+1-\xi_i \qquad y_i=+1$$

$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b\leqslant-1+\xi_i \qquad y_i=-1$$

$$\xi_i\geqslant 0 \quad \forall i$$

c值有明确的含义：选取大的c值，意味着更强调最小化训练错误。非线性分类有相同的做法，这里不再赘述。

5.2　支持向量机的实例

目前，关于支持向量机的研究，除理论研究外，主要集中在对它和一些已有方法进行实验对比研究。比如，贝尔实验室利用美国邮政标准手写数字库进行的对比实验，每个样本数字都是 16×16 的点阵(即 256 维)，训练集共有 7300 个样本，测试集有 2000 个样本。表 5-1 是用人工和几种传统方法得到的分类器的测试结果，其中两层神经网络的结果是取多个两层神经网络中的较好者，而 LeNet 1 是一个专门针对这个手写数字识别问题设计的五层神经网络。

表 5-1　传统方法对美国邮政手写数字库的识别结果

分　类　器	测试错误率/%	分　类　器	测试错误率/%
人工分类	2.5	两层神经网络	5.9
决策树方法	16.2	LeNet 1	5.1

3 种支持向量机的实验结果见表 5-2。

表 5-2　3 种支持向量机的实验结果

支持向量机类型	内积函数中的参数	支持向量个数	测试错误率/%
多项式内积	$q=3$	274	4.0
径向基函数内积	$\sigma^2=0.3$	291	4.1
Sigmoid 内积	$b=2, c=1$	254	4.2

这个实验一方面初步说明了 SVM 方法较传统方法有明显的优势，也说明了不同的 SVM 方法可以得到性能相近的结果(不像神经网络那样十分依赖对模型的选择)。另外，实验中还得到 3 种不同的支持向量机，最终得出的支持向量只是总训练样本中很少的一部分，而且 3 组支持向量中有 80%以上是重合的，也说明支持向量本身对不同的方法具有一定的不敏感性。遗憾的是，这些结论目前都仅仅是有限实验中观察到的现象，如果能够证明它们确实是正确的，将会使支持向量机的理论和应用有巨大的突破。此外，支持向量机有一些免费软件，如 LibSVM、SVM^{light}、bSVM、mySVM、MATLAB SVM toolbox 等。其中，LibSVM 是台湾大学林智仁(Lin Chih-Jen)教授等开发设计的一个简单、易于使用和快速有效的 SVM 模式识别与回归的软件包，它不仅提供了编译好的可在 Windows 系统上执行的文件，还提供了源代码。

5.3　支持向量机的实现算法

下面用台湾大学林智仁教授所做的 SVM 工具箱做一个简单的分类，这个工具箱能够给出分类的精度和每类的支持向量，但是用 MATLAB 工具箱不能画出分类面，用训练样本点

作为输入来测试模型的性能试验程序和结果，如图 5-4 所示。

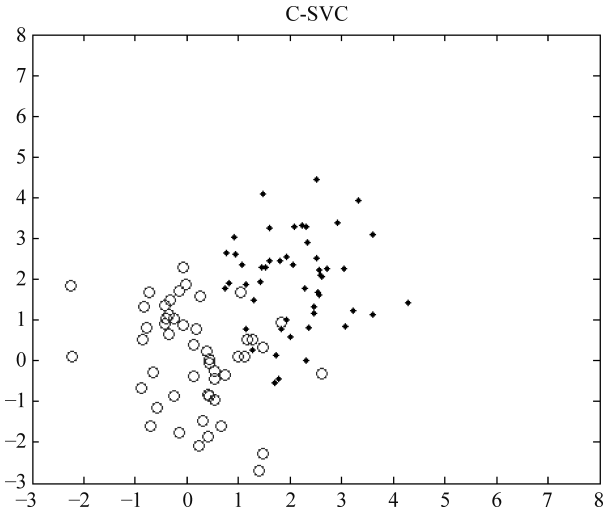


图 5-4　训练样本

测试样本如图 5-5 所示。

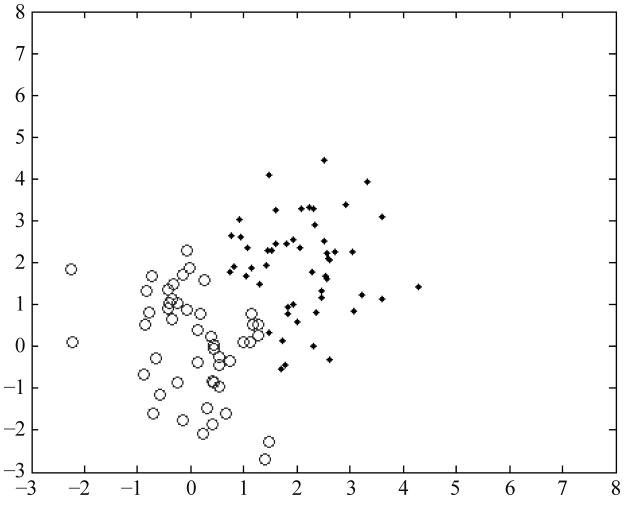


图 5-5　测试样本

分类器分类代码如下。

```
N=50;
n=2*N;
x1=randn(2,N);
y1=ones(1,N);
x2=2+randn(2,N);
y2=-ones(1,N);
figure;
plot(x1(1,:),x1(2,:),'o',x2(1,:),x2(2,:),'k.');
```

```
axis([-3 8 -3 8]);
title('C-SVC')
hold on;
X1=[x1,x2];
Y1=[y1,y2];
X=X1';
Y=Y1';

model=svmtrain(Y,X)
Y_later=svmpredict(Y,X,model);
%C1num=sum(Y_later>0);
%C2num=2*N-C1num;
%
%x3=zeros(2,C1num)
%x4=zeros(2,C2num)

figure;
for i=1:2*N
    if Y_later(i) > 0
        plot(X1(1,i),X1(2,i),'o');
        axis([-3 8 -3 8]);
        hold on
    else
        plot(X1(1,i),x1(2,i),'k.');
        hold on
    end
end
```

进一步，关于最优和广义最优分类面的推广能力，有下面的结论。如果一组训练样本能够被一个最优分类面或广义最优分类面分开，则对于测试样本，分类错误率的期望的上界是训练样本中平均的支持向量占总训练样本数的比例，即

$$E(P(\text{error})) \leqslant \frac{E[\text{支持向量机}]}{\text{训练样本总数}-1} \tag{5-20}$$

因此，支持向量机的推广性与变换空间的维数也是无关的。只要能够适当地选择一种内积定义，构造一个支持向量数相对较少的最优或广义最优分类面，就可以得到较好的推广性。在这里，统计学习理论使用了与传统方法完全不同的思路，即不像传统方法那样首先试图将原输入空间降维（即特征选择和特征变换），而是设法将输入空间升维，以求在高维空间中问题变得线性可分（或接近线性可分）；因为升维后只是改变了内积运算，并没有使算法复杂性随着维数的增加而增加，而且在高维空间中的推广能力并不受维数影响，因此这种方法才是可行的。

5.4　多类支持向量机

SVM 最初是为两类问题设计的，当处理多类问题时，就需要构造合适的多类分类器。目前，构造 SVM 多类分类器的方法主要有两种：一种是直接法，通过对原始最优化问题进

行适当改变，从而同时计算出所有分类决策函数，这种方法看似简单，但其计算复杂度比较高，实现起来比较困难，只适用于小型问题；另一类是间接法，主要通过组合多个二分类器实现多分类器的构造，常见的方法有一对多法和一对一法两种。

(1) 一对多法(one-versus-rest，OVR)SVM。每次训练时，把指定类别的样本归为一类，其他剩余的样本归为另一类，这样 N 个类别的样本就构造出 N 个 SVM 分类器。对未知样本进行分类时，具有最大分类函数值的那一类作为其归属类别。

(2) 一对一法(one-versus-one，OVO)SVM。这种方法是在任意两类样本之间设计一个 SVM 分类器，因此 N 个类别的样本就需要设计 $N(N-1)/2$ 个 SVM 分类器。当对未知样本进行分类时，使用“投票法”，最后得票最多的类别即该未知样本的类别。

5.5 总　　结

SVM 以统计学习理论作为坚实的理论依据，它有很多优点：基于结构风险最小化，克服了传统方法的过学习(overfitting)和陷入局部最小的问题，具有很强的泛化能力；采用核函数方法，向高维空间映射时并不增加计算的复杂性，又有效地克服了维数灾难(curse of dimensionality)问题。同时，目前的 SVM 研究也有下面的局限性。

(1) SVM 的性能很大程度上依赖于核函数的选择，但没有很好的方法指导针对具体问题的核函数选择。

(2) 训练测试 SVM 的速度和规模是另一个问题，尤其是对实时控制问题，速度是一个对 SVM 应用限制很大的因素；针对这个问题，Platt 和 Keerthi 等分别提出 SMO(Sequential Minimal Optimization)和改进的 SMO 方法，但还值得进一步研究。

(3) 现有 SVM 理论仅讨论具有固定惩罚系数 c 的情况，而实际上正、负样本的两种误判造成的损失往往是不同的。

课 后 习 题

1. 支持向量机的基本思想和原理分别是什么？为什么要引入核函数？

2. 试分析支持向量机对缺失数据和噪声敏感的原因。

3. 比较感知机的对偶形式与线性可分支持向量机的对偶形式。

4. 已知正例点 $\boldsymbol{x}_1=(1,2)^{\mathrm{T}}$，$\boldsymbol{x}_2=(2,3)^{\mathrm{T}}$，$\boldsymbol{x}_3=(3,3)^{\mathrm{T}}$，负例点 $\boldsymbol{x}_4=(2,1)^{\mathrm{T}}$，$\boldsymbol{x}_5=(3,2)^{\mathrm{T}}$，试求最大间隔分离超平面和分类决策函数，并在图上画出分离超平面、间隔边界及支持向量。