第 5 章补充材料

● 支持向量机的学习

给定两个类别的训练样本集 $D=\{(\mathbf{x}_1,z_1),\cdots,(\mathbf{x}_n,z_n)\}$，$\mathbf{x}_i$ 为训练样本的特征矢量，z_i 是样本的类别标识：

$$z_i=\begin{cases}+1, & \mathbf{x}_i\in\omega_1\\ -1, & \mathbf{x}_i\in\omega_2\end{cases}$$

I．线性可分的情况

我们先来讨论样本集 D 是线性可分的情况。作为最优分类超平面来说，首先需要能够对样本集 D 正确分类，亦即需要满足：

$$z_i\left(\mathbf{w}^t\mathbf{x}_i+w_0\right)>0,\quad \forall i=1,\cdots,n \tag{1}$$

由于：

$$z_i\left(\mathbf{w}^t\mathbf{x}_i+w_0\right)\geq\min_{1\leq j\leq n}\left[z_j\left(\mathbf{w}^t\mathbf{x}_j+w_0\right)\right]=b_{\min}>0$$

$b_{\min}$ 是训练样本集中距离分类超平面最近样本的函数间隔。当权值矢量 $\mathbf{w}$ 和偏置 w_0 同时乘上一个正数 $1/b_{\min}$ 时，对应超平面的位置是不会发生变化的，因此（1）式的条件可以重写为：

$$z_i\left(\mathbf{w}^t\mathbf{x}_i+w_0\right)\geq 1,\quad \forall i=1,\cdots,n \tag{2}$$

这样做的好处是通过适当调整权值矢量和偏置，使得最优超平面到样本集的函数间隔 b 变为了 1，亦即支持面与分类界面之间的函数间隔为 1。

最优分类超平面的第 2 个条件是要使得与样本集之间的几何间隔 γ 最大，在函数间隔为 1 的条件下有：

$$\gamma=\frac{1}{\|\mathbf{w}\|}$$

这样我们就得到了训练样本集线性可分条件下学习最优分类超平面的优化准则和约束条件：

原始优化问题

$$\min_{\mathbf{w},w_0}J_{SVM}\left(\mathbf{w},w_0\right)=\frac{1}{2}\|\mathbf{w}\|^2 \tag{3}$$

约束：

$$z_i\left(\mathbf{w}^t\mathbf{x}_i+w_0\right)\geq 1,\quad i=1,\cdots,n$$

这是一个典型的线性不等式约束条件下的二次优化问题。在支持向量机的学习算法中，一般并不是直接求解这个原始问题，而是转而求解与其等价的对偶问题。

在说明对偶问题之前，先来看定义在矢量 $\mathbf{u}$ 和 $\mathbf{v}$ 上的函数 $f(\mathbf{u},\mathbf{v})$ 的 min-max 和 max-min 问题。

min-max 问题：$\begin{cases} f^*(\mathbf{u}) = \max\limits_{\mathbf{v}} f(\mathbf{u},\mathbf{v}) \\ \min\limits_{\mathbf{u}} f^*(\mathbf{u}) = \min\limits_{\mathbf{u}} \max\limits_{\mathbf{v}} f(\mathbf{u},\mathbf{v}) \end{cases}$

max-min 问题：$\begin{cases} f^*(\mathbf{v}) = \min\limits_{\mathbf{u}} f(\mathbf{u},\mathbf{v}) \\ \max\limits_{\mathbf{v}} f^*(\mathbf{v}) = \max\limits_{\mathbf{v}} \min\limits_{\mathbf{u}} f(\mathbf{u},\mathbf{v}) \end{cases}$

冯.诺依曼证明上述两个问题如果有解存在的话，必在同一点取得最优解，亦即：

$$\min_{\mathbf{u}} \max_{\mathbf{v}} f(\mathbf{u},\mathbf{v}) = \max_{\mathbf{v}} \min_{\mathbf{u}} f(\mathbf{u},\mathbf{v}) = f(\mathbf{u}^*,\mathbf{v}^*) \tag{4}$$

也就是说函数 $f(\mathbf{u},\mathbf{v})$ 先对 $\mathbf{u}$ 取最小值再对 $\mathbf{v}$ 取最大值，还是先对 $\mathbf{v}$ 取最大值再对 $\mathbf{u}$ 取最小值的结果是一样的，两个优化问题的求解顺序可以颠倒。

下面根据（3）的原始优化问题构造 Lagrange 函数：

$$L(\mathbf{w},w_0,\boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n}\alpha_i\left[z_i\left(\mathbf{w}^t\mathbf{x}_i + w_0\right) - 1\right] \tag{5}$$

其中 $\boldsymbol{\alpha} = (\alpha_1,\cdots,\alpha_n)^t$，$\alpha_i \geq 0$ 是针对（3）优化问题中每个约束不等式引入的 Lagrange 系数。

考虑 Lagrange 函数关于矢量 $\boldsymbol{\alpha}$ 的最大化问题：$\max\limits_{\boldsymbol{\alpha}\geq\mathbf{0}} L(\mathbf{w},w_0,\boldsymbol{\alpha})$，当 $z_i\left(\mathbf{w}^t\mathbf{x}_i + w_0\right) > 1$ 时，Lagrange 函数在 $\alpha_i = 0$ 处取得最大值；而当 $z_i\left(\mathbf{w}^t\mathbf{x}_i + w_0\right) = 1$ 时，α_i 可以大于 0。总之当 Lagrange 函数取得最大值时，（5）求和式中的两个乘积项 α_i 和 $z_i\left(\mathbf{w}^t\mathbf{x}_i + w_0\right) - 1$ 必有一项为 0，因此：

$$\max_{\boldsymbol{\alpha}} L(\mathbf{w},w_0,\boldsymbol{\alpha}) = J_{SVM}(\mathbf{w},w_0) = \frac{1}{2}\|\mathbf{w}\|^2$$

这样（3）的原始优化问题就等价于一个 min-max 问题。考虑到（4）式，这个问题也等价于一个 max-min 问题：

$$\min_{\mathbf{w},w_0} J_{SVM}(\mathbf{w},w_0) = \min_{\mathbf{w},w_0}\max_{\boldsymbol{\alpha}} L(\mathbf{w},w_0,\boldsymbol{\alpha}) = \max_{\boldsymbol{\alpha}}\min_{\mathbf{w},w_0} L(\mathbf{w},w_0,\boldsymbol{\alpha})$$

首先计算 Lagrange 函数针对 $\mathbf{w}$ 和 w_0 的最小化问题：

$$\frac{\partial L(\mathbf{w},w_0,\boldsymbol{\alpha})}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n}\alpha_i z_i \mathbf{x}_i = 0 \quad \rightarrow \quad \mathbf{w} = \sum_{i=1}^{n}\alpha_i z_i \mathbf{x}_i \tag{6}$$

$$\frac{\partial L(\mathbf{w},w_0,\boldsymbol{\alpha})}{\partial w_0} = -\sum_{i=1}^{n}\alpha_i z_i = 0 \quad \rightarrow \quad \sum_{i=1}^{n}\alpha_i z_i = 0 \tag{7}$$

将(6)和(7)重新代入 Lagrange 函数：

$$\begin{aligned} L(\mathbf{w},w_0,\boldsymbol{\alpha}) &= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n}\alpha_i\left[z_i\left(\mathbf{w}^t\mathbf{x}_i + w_0\right) - 1\right] \\ &= \frac{1}{2}\left(\sum_{i=1}^{n}\alpha_i z_i \mathbf{x}_i\right)^t\left(\sum_{i=1}^{n}\alpha_i z_i \mathbf{x}_i\right) - \sum_{i=1}^{n}\left\{\alpha_i z_i\left(\sum_{j=1}^{n}\alpha_j z_j \mathbf{x}_j\right)^t \mathbf{x}_i + \alpha_i z_i w_0 - \alpha_i\right\} \\ &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j z_i z_j \mathbf{x}_i^t\mathbf{x}_j - \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j z_i z_j \mathbf{x}_i^t\mathbf{x}_j - w_0\sum_{i=1}^{n}\alpha_i z_i + \sum_{i=1}^{n}\alpha_i \\ &= \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j z_i z_j \mathbf{x}_i^t\mathbf{x}_j \end{aligned}$$

此时，Lagrange 函数只与优化矢量 $\boldsymbol{\alpha}$ 有关，而与 $\mathbf{w},w_0$ 无关。因此，可以由 Lagrange 函

数针对 $\boldsymbol{\alpha}$ 的最大化，同时考虑（7）式的约束，得到原始问题的对偶优化问题：

对偶优化问题

$$\max_{\boldsymbol{\alpha}} L(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j z_i z_j \mathbf{x}_i^t \mathbf{x}_j \qquad (8)$$

约束：

$$\alpha_i \geq 0, \quad i = 1, \cdots, n$$

$$\sum_{i=1}^{n} \alpha_i z_i = 0$$

原始优化问题和对偶优化问题都是典型的线性不等式约束条件下的二次优化问题，求解两者中的任何一个都是等价的。但 SVM 算法一般求解的是对偶问题，因为它有如下两个特点：

1、 对偶问题不直接优化权值矢量 $\mathbf{w}$，因此与样本的特征维数 d 无关，只与样本的数量 n 有关。当样本的特征维数很高时，对偶问题更容易求解；

2、 对偶优化问题中，训练样本只以任意两个矢量内积的形式出现，因此只要能够计算矢量之间的内积，而不需要知道样本的每一维特征就可以进行优化求解。

以上两个特点使得我们可以很容易地将“核函数”引入到算法中，实现非线性的 SVM 分类。

II．线性不可分的情况

下面来看一下样本集 D 是线性不可分的情况。重新考察（3）式的优化问题，当样本集线性不可分时，不存在任何一个权值矢量 $\mathbf{w}$ 和偏置 w_0 能够使得作为约束的 n 个不等式都得到满足。通过在每个不等式上引入一个非负的“松弛变量” ξ，使得不等式变为：

$$z_i \left(\mathbf{w}^t \mathbf{x}_i + w_0 \right) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

只要选择一系列适合的松弛变量 ξ，不等式约束条件总是可以得到满足的。然而，即使训练样本集是线性不可分的，我们也希望学习得到的分类器能够正确识别尽量多的训练样本，换句话说就是希望尽量多的松弛变量 $\xi_i = 0$。因此目标函数就需要同时考虑两方面因素的优化：与分类界面和样本集之间的几何间隔相关的 $\|\mathbf{w}\|^2$，以及不为 0 的松弛变量的数量。

直接优化松弛变量的数量存在一定的难度，一般是转而优化一个相关的目标：$\sum_{i=1}^{n} \xi_i$。在一个优化问题中无法同时优化两个目标，需要引入一个大于 0 的常数 C 来协调对两个优化目标的关注程度。C 值越大表示我们希望更少的训练样本被错误识别，C 值越小表示我们希望分类界面与训练样本集的间隔更大。这样，我们就得到了在样本集线性不可分情况下的原始优化问题：

原始优化问题

$$\min_{\mathbf{w}, w_0} J_{SVM}(\mathbf{w}, w_0) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{n} \xi_i \qquad (8)$$

约束：

$$z_i \left(\mathbf{w}^t \mathbf{x}_i + w_0 \right) \geq 1 - \xi_i, \quad i = 1, \cdots, n$$

$$\xi_i \geq 0, \quad i = 1, \cdots, n$$

类似于线性可分情况，针对两组不等式约束分别引入 Lagrange 系数 $\boldsymbol{\alpha}$ 和 $\boldsymbol{\beta}$，建立

Lagrange 函数：

$$L(\mathbf{w},w_0,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta})=\frac{1}{2}\|\mathbf{w}\|^2+C\sum_{i=1}^{n}\xi_i-\sum_{i=1}^{n}\alpha_i\left[z_i\left(\mathbf{w}^t\mathbf{x}_i+w_0\right)-1+\xi_i\right]-\sum_{i=1}^{n}\beta_i\xi_i$$

同样道理，原始问题的优化等价于 Lagrange 函数首先对 $\mathbf{w},w_0$ 和 $\boldsymbol{\xi}$ 进行最小值优化，然后对 $\boldsymbol{\alpha},\boldsymbol{\beta}$ 在非负的约束下进行最大值优化：

$$\frac{\partial L(\mathbf{w},w_0,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \mathbf{w}}=\mathbf{w}-\sum_{i=1}^{n}\alpha_i z_i\mathbf{x}_i=0 \quad \rightarrow \quad \mathbf{w}=\sum_{i=1}^{n}\alpha_i z_i\mathbf{x}_i \tag{9a}$$

$$\frac{\partial L(\mathbf{w},w_0,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial w_0}=-\sum_{i=1}^{n}\alpha_i z_i=0 \quad \rightarrow \quad \sum_{i=1}^{n}\alpha_i z_i=0 \tag{9b}$$

$$\frac{\partial L(\mathbf{w},w_0,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \xi_i}=C-\alpha_i-\beta_i=0 \tag{9c}$$

将上述 3 式重新代入 Lagrange 函数：

$$\begin{aligned}
L(\mathbf{w},w_0,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta})&=\frac{1}{2}\|\mathbf{w}\|^2+C\sum_{i=1}^{n}\xi_i-\sum_{i=1}^{n}\alpha_i\left[z_i\left(\mathbf{w}^t\mathbf{x}_i+w_0\right)-1+\xi_i\right]-\sum_{i=1}^{n}\beta_i\xi_i\\
&=\frac{1}{2}\left(\sum_{i=1}^{n}\alpha_i z_i\mathbf{x}_i\right)^t\left(\sum_{i=1}^{n}\alpha_i z_i\mathbf{x}_i\right)+C\sum_{i=1}^{n}\xi_i-\sum_{i=1}^{n}\left\{\alpha_i z_i\left(\sum_{j=1}^{n}\alpha_j z_j\mathbf{x}_j\right)^t\mathbf{x}_i+\alpha_i z_i w_0-\alpha_i+\alpha_i\xi_i\right\}-\sum_{i=1}^{n}\beta_i\xi_i\\
&=\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j z_i z_j\mathbf{x}_i^t\mathbf{x}_j-\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j z_i z_j\mathbf{x}_i^t\mathbf{x}_j-w_0\sum_{i=1}^{n}\alpha_i z_i+\sum_{i=1}^{n}\alpha_i+\sum_{i=1}^{n}(C-\alpha_i-\beta_i)\xi_i\\
&=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j z_i z_j\mathbf{x}_i^t\mathbf{x}_j
\end{aligned}$$

可以看出，重写的 Lagrange 函数与线性可分情况是完全相同的，与 $\mathbf{w},w_0$，松弛矢量 $\boldsymbol{\xi}$ 无关，也与引入的第 2 组 Lagrange 系数 $\boldsymbol{\beta}$ 无关。对偶优化问题与线性可分情况的唯一不同点是由（9c）式引入的：$\alpha_i=C-\beta_i$，考虑到 $\beta_i\geq 0$，因此需要增加约束 $\alpha_i\leq C$。

对偶优化问题

$$\max_{\boldsymbol{\alpha}} L(\boldsymbol{\alpha})=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j z_i z_j\mathbf{x}_i^t\mathbf{x}_j \tag{10}$$

约束：

$$C\geq\alpha_i\geq 0,\quad i=1,\cdots,n$$

$$\sum_{i=1}^{n}\alpha_i z_i=0$$

在线性不可分的情况下，对偶问题相比于原始优化问题要简单。线性 SVM 分类器的学习，就是采用二次规划算法对（10）优化问题的求解。最优化方法的研究已经证明此类问题属于凸规划问题，存在唯一的最优解，并且可以由相关算法计算求解。常用的二次规划算法包括：内点法、有效集法、椭球算法等等，而且经过研究已经找到了专门针对 SVM 学习的有效算法--序列最小化算法（SMO, Sequential Minimal Optimization）。

通过对偶问题的优化，可以得到与每个训练样本相关的一组最优 Lagrange 系数 $\boldsymbol{\alpha}$。构造线性判别函数需要的是权值矢量 $\mathbf{w}$ 和偏置 w_0，因此下面需要考虑如何由系数 $\boldsymbol{\alpha}$ 计算 $\mathbf{w}$ 和 w_0。在此之前我们首先来看一下 $\boldsymbol{\alpha}$ 中元素的含义。

从前面针对 $\boldsymbol{\alpha}$ 优化的分析中可以看到，α_i 是与（3）优化问题的第 i 个约束 $z_i\left(\mathbf{w}^t\mathbf{x}_i+w_0\right)\geq 1$ 相关的 Lagrange 系数。当约束不等式以大于 1 的方式得到满足时，相应的

Lagrange 系数 $\alpha_i = 0$；而当约束以等于 1 的方式得到满足时，系数 α_i 可以大于 0。同样道理，线性不可分情况下优化问题（8）中，由于有（9c）式中 $\alpha_i = C - \beta_i$ 关系存在，因此当 $\xi_i > 0$ 时，Lagrange 系数 $\beta_i = 0$，而 $\alpha_i = C$；当 $\xi_i = 0$ 时，β_i 可以大于 0，相应的 α_i 可以小于 C。

更严格地说，依据最优化方法中的 Kuhn-Tucker 定理可以证明有如下关系存在：

$$\begin{cases} z_i\left(\mathbf{w}^t\mathbf{x}_i + w_0\right) > 1, & \alpha_i = 0 \\ z_i\left(\mathbf{w}^t\mathbf{x}_i + w_0\right) = 1, & C > \alpha_i > 0 \\ z_i\left(\mathbf{w}^t\mathbf{x}_i + w_0\right) < 1, & \alpha_i = C \end{cases} \tag{11}$$

在建立学习优化问题的过程中，我们通过适当调整 $\mathbf{w}$ 和 w_0，使得距离最优分类界面最近的样本到分类超平面的函数间隔变为了 1，亦即两个类别的支持面与分类超平面之间的函数间隔为 1。因此，由图 1 可以看出，依据对偶优化问题的解，完全可以确定每个训练样本相对于最优分类超平面以及两个支持面之间的位置关系。$\alpha_i = 0$ 对应的训练样本处于各自类别支持面之外；$C > \alpha_i > 0$ 对应的训练样本处于支持面之上；$\alpha_i = C$ 对应的训练样本则处于各自类别支持面与分类超平面之间，甚至是分类界面的反方向区域（图 1 中黑色方框中的样本）。所有对应 $\alpha_i > 0$ 的训练样本称为支持向量。

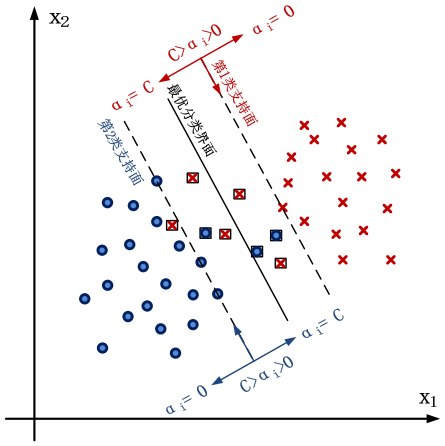

图 1 支持向量与 Lagrange 系数

借助于（9a），可以将判别函数的权值 $\mathbf{w}$ 表示为训练样本由相应 Lagrange 系数加权求和的形式：

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i z_i \mathbf{x}_i \tag{12}$$

因此，由对偶优化问题的解可以直接得到判别函数的权值矢量。这里需要注意的是实际上只有支持向量参与了求和式的计算，非支持向量的系数 α_i 为 0，对 $\mathbf{w}$ 的计算没有贡献。

任意一个处于支持面上的支持向量与分类界面之间的函数间隔为 1，因此偏置 w_0 可以利用任意一个对应于 $C > \alpha_i > 0$ 的支持向量 $\mathbf{x}_i$ 由下述方程求得：

$$z_i\left(\mathbf{w}^t\mathbf{x}_i + w_0\right) = 1 \quad \rightarrow \quad w_0 = z_i - \mathbf{w}^t\mathbf{x}_i \tag{13}$$

这样，我们就可以通过求解对偶优化问题得到一组 Lagrange 系数 $\boldsymbol{\alpha}$，进而根据（12）和（13）式计算线性判别函数的权值矢量 $\mathbf{w}$ 和偏置 w_0，得到最优的线性判别函数。

III. 非线性支持向量机

将核函数引入支持向量机，将其转化为非线性分类器。支持向量机的学习过程主要是求解优化问题（10），注意到其中只涉及到任意两个训练样本的内积计算，因此可以引入核函数 K 将其转化为（14）式进行优化，达到首先将每个训练样本由某个非线性映射 $\mathbf{\Phi}$ 映射到特征空间，然后在特征空间中求解最大间隔超平面，对应于输入空间中的非线性分类界面。

非线性 SVM 的优化问题

$$\max_{\boldsymbol{\alpha}} L(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j z_i z_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{14}$$

约束：

$$\sum_{i=1}^{n} \alpha_i z_i = 0$$

$$C \geq \alpha_i \geq 0, \quad i = 1, \cdots, n$$

通过（14）优化问题的求解，可以得到每个训练样本对应的 Lagrange 系数 $\boldsymbol{\alpha}$，而要构造判别函数需要计算权值矢量 $\mathbf{w}$ 和偏置 w_0。注意到权值矢量 $\mathbf{w}$ 是一个经过 $\mathbf{\Phi}$ 映射之后特征空间中的矢量，可以由（12）计算，只不过每个训练样本需要由映射之后的矢量 $\mathbf{\Phi}(\mathbf{x}_i)$ 来代替：

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i z_i \mathbf{\Phi}(\mathbf{x}_i) \tag{15}$$

但是在核方法中我们并没有直接定义映射 $\mathbf{\Phi}$，而是通过引入核函数 K 来间接的达到非线性映射的目的，因此无法计算每个 $\mathbf{\Phi}(\mathbf{x}_i)$。但是如果将（15）式代入特征空间中的线性判别函数：

$$\mathbf{w}^t \mathbf{\Phi}(\mathbf{x}) + w_0 = \left[\sum_{i=1}^{n} \alpha_i z_i \mathbf{\Phi}(\mathbf{x}_i) \right]^t \mathbf{\Phi}(\mathbf{x}) + w_0 = \sum_{i=1}^{n} \alpha_i z_i K(\mathbf{x}, \mathbf{x}_i) + w_0$$

可以看出，输入空间中的非线性 SVM 判别函数只需要利用核函数计算测试样本 $\mathbf{x}$ 和训练样本 $\mathbf{x}_i$ 在特征空间中的内积 $K(\mathbf{x}, \mathbf{x}_i)$：

$$g(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i z_i K(\mathbf{x}, \mathbf{x}_i) + w_0 \tag{16}$$

偏置 w_0 同样可以由某个满足 $C > \alpha_j > 0$ 的支持向量 $\mathbf{x}_j$ 计算：

$$w_0 = z_j - \sum_{i=1}^{n} \alpha_i z_i K(\mathbf{x}_j, \mathbf{x}_i) \tag{17}$$

这样我们看到，通过引入核函数可以实现非线性的支持向量机分类。所付出的代价是无法像线性 SVM 一样直接计算出权值矢量 $\mathbf{w}$，而是需要在识别的时候，采用（16）式利用核函数计算测试样本与训练样本在特征空间中的内积，从而得到判别函数的输出。由于非支持向量的 Lagrange 系数 α 为 0，因此算法只需要保存和计算所有的支持向量即可。

IV. 多类别支持向量机

支持向量机可以采用“一对多方式”和“一对一方式”，将多类别问题转化为多个两类别问题来解决，但是这两种方式都存在着无法辨别的区域。在支持向量机中还可以采用一种特殊的方式进行多类别范磊。

这种特殊的多类别分类方式是在原来“一对一方式”的基础之上，增加了一个“投票”的机

制。首先，在学习过程中利用任意两个类别的样本学习出$c(c-1)/2$个区分两个类别的支持向量机分类器。在识别过程中，计算每一个支持向量机判别函数的输出，根据输出的正负向相关类别的“投票箱”中投入一票。例如，如果判别函数$g_{ij}(\mathbf{x})>0$，则在第i的投票箱中投入一票，反之则在第j类的票箱中投入一票。最后，统计c个类别的得票数，判别$\mathbf{x}$属于得票最多的类别。这个过程可以形式化的表示为：

$$v_i(\mathbf{x})=\sum_{j=1,j\neq i}^{c} I\left(g_{ij}(\mathbf{x})>0\right)$$

如果：$k=\arg\max_{1\le i\le c} v_i(\mathbf{x})$，则判别：$\mathbf{x}\in\omega_k$

其中I为示性函数：

$$I(a)=\begin{cases}1, & a \text{ is true}\\ 0, & a \text{ is false}\end{cases}$$

V. 支持向量机的最优性

结构风险最小化原则（SRM，Structural Risk Minimization）：把函数集$S=\{f(\mathbf{x},\mathbf{w}),\mathbf{w}\in\Omega\}$分解为一个函数子集序列：

$$S_1\subset S_2\subset\cdots S_k\subset\cdots\subset S$$

各个子集按照VC维的大小排序：

$$h_1\le h_2\le\cdots\le h_k\le\cdots$$

在子集序列中寻找经验风险与置信范围之和最小的子集，这个子集中使经验风险最小的函数就是所求的最优函数。

定理：在d维空间中，假设所有n个样本都在一个超球范围之内，超球的半径为R，那么γ-间隔分类超平面集合的VC维h满足如下不等式：

$$h\le\min\left(\left[\frac{R^2}{\gamma^2}\right],d\right)+1$$

而间隔$\gamma=1/\|\mathbf{w}\|$，因此根据SRM的原则，只需在保证经验风险为0的条件下（超平面能够正确分类全部训练样本），最小化权值矢量的长度$\|\mathbf{w}\|$。