支持向量机学习笔记

（欢迎关注微信公众号：Genlovy562）

一些废话，不想看请悄悄略过：支持向量机作为机器学习领域的经典方法，之前一直在用，但是也仅仅会用而已，并没有深入理解这个模型的原理和过程。但是近来觉得仅仅会用是不够的，所以决定咬咬牙把理论部分给啃下来。由于数学基础不好，学习这个模型花了不少时间。我这个人以前没有认真写笔记的习惯，但学完这个支持向量机之后觉得内容挺多的，有必要做个笔记将一些**细节**记录以备将来查阅。想到如果把它分享出来，一则可以督促我把这笔记写得像样些；二则如果能给到正在学习这个模型又恰巧看到这个笔记的人一丢丢帮助，那也不是坏事，毕竟我们在学习过程中也在网上查过许多相当好的共享资料，大家共同分享，共同进步嘛。好了，带着一颗炽热但不躁动的心，我们一起来学习吧！

支持向量与最大化间隔

首先考虑**线性可分**的二分类问题。假设有如图 1 所示的数据，图中“+”为正例样本，“-”为负例样本。直观地，我们可以通过一条直线（超平面，在高维空间中通常是这样叫的）将正样本数据和负样本数据分开。图 1 中可以达到我们目的的直线（超平面）有五条，实际上，有无数条直线（超平面）都可以做到将正例样本和负例样本分开，那么，哪一条“最好”呢。直观上看，两类样本“正中间”那条红色的最好，因为它对不论正例还是负例样本局部扰动的“容忍性”最好，我们说它“鲁棒性”最好。实际上，看图 1 就可以知道，当我们放一个新的样本进来时，红色那条直线（超平面）将其正确分类的可能性最大，因此，这条直线（超平面）的泛化能力最强。

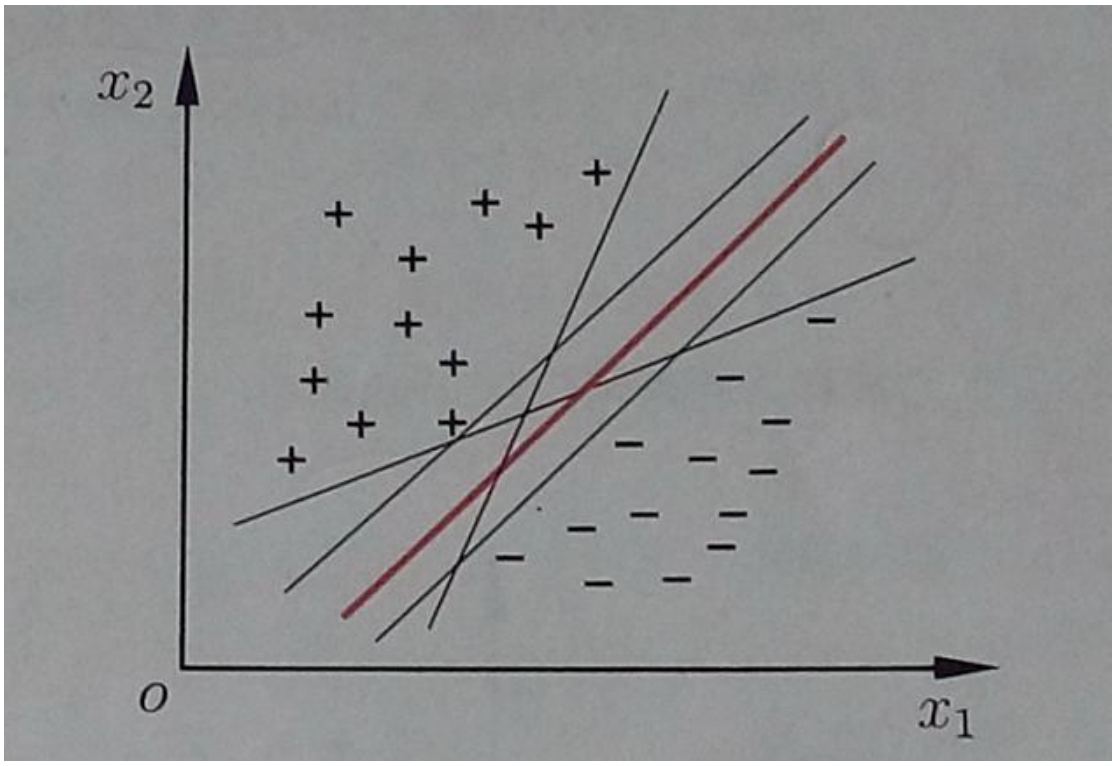

图 1：划分超平面（摘自周志华著《机器学习》P121，图 6.1）

为了方便，在不引起歧义的情况下下文将前面提到的划分直线统称为超平面。现在将我们的样本数据记为 $D=\{(\boldsymbol{X}_1,y_1),(\boldsymbol{X}_2,y_2),\cdots\cdots,(\boldsymbol{X}_m,y_m)\}$，$\boldsymbol{X}_i\in\mathbf{R}^n$为 n 维向量（n 是自变量的维数），m 为样本个数，$y_i\in\{-1,+1\}$。假设那个鲁棒性最好的超平面的方程为$\boldsymbol{\omega}^T\boldsymbol{X}+b=0$，其中$\boldsymbol{\omega}=(\omega_1;\omega_2;\cdots\cdots;\omega_n)$为法向量（可以理解成线性方程的系数），$b$ 为位移项（线性方程的常数）。现在若这个超平面能将所有样本全部正确分类，也就是说正例和负例样本分别位于超平面的两侧，写成表达式即：若$y_i=+1$，则有$\boldsymbol{\omega}^T\boldsymbol{X}_i+b>0$；若$y_i=-1$，则有$\boldsymbol{\omega}^T\boldsymbol{X}_i+b<0$。据此，现在我们可以令：

$$\begin{cases}\boldsymbol{\omega}^T\boldsymbol{X}_i+b\geq+\gamma，\text{若 } y_i=+1\\ \boldsymbol{\omega}^T\boldsymbol{X}_i+b\leq-\gamma，\text{若 } y_i=-1\end{cases}，\gamma>0 \text{ 为常数} \qquad (1)$$

那么，如图 2 所示（图 2 为取$\gamma=1$ 的情况），距离超平面最近的样本点可能使（1）式中的等号成立，这些样本点就被称为“支持向量”。使（1）式中等号成立的那些点，它们离划分超平面的距离刚好为$\frac{\gamma}{\|\boldsymbol{\omega}\|}$（参考点到直线距离公式，不

记得的话去查查），使等号成立的两个异类样本距划分超平面的距离之和为$\frac{2\gamma}{\|\boldsymbol{\omega}\|}$，这个距离被称为“间隔”。为了使模型更加可靠，或者说使分类结果更加准确，我们就要最大化这个“间隔”。这里要重点理解下为何最大化这个“间隔”可以使模型更鲁棒。我们从图 2 来看，如果一个样本离超平面的距离更远，那么它就越容易被正确分类，或者也可以理解为离超平面越远的样本点，它被正确分类的可信程度越高。既然“支持向量”是离超平面最近的点，那么如果最大化的它们离超平面的距离（就是“间隔”），整个模型的鲁棒性自然就更好了。

现在来重新理解下（1）式的意义：我们要找到这样的$\boldsymbol{\omega}$和 b，它能使超平面将所有样本全部分类正确，即：若$y_i = +1$，则必有$\boldsymbol{\omega}^T\boldsymbol{X}_i + b > 0$；若$y_i = -1$，则必有$\boldsymbol{\omega}^T\boldsymbol{X}_i + b < 0$。由于一定存在正常数 p 使得$\boldsymbol{\omega}^T\boldsymbol{X}_i + b > 0$ 与$\boldsymbol{\omega}^T\boldsymbol{X}_i + b \geq p$ 等价（若一个数大于 0，则它一定大于等于某个正数）；同理，一定存在正常数 q 使得$\boldsymbol{\omega}^T\boldsymbol{X}_i + b < 0$ 与$\boldsymbol{\omega}^T\boldsymbol{X}_i + b \leq -q$ 等价。在（1）式中，我们取 $p = q = \gamma$，是为了使超平面能够处于两个异类样本的“正中间”，这样才能达到最好的泛化能力（参考图 1）。

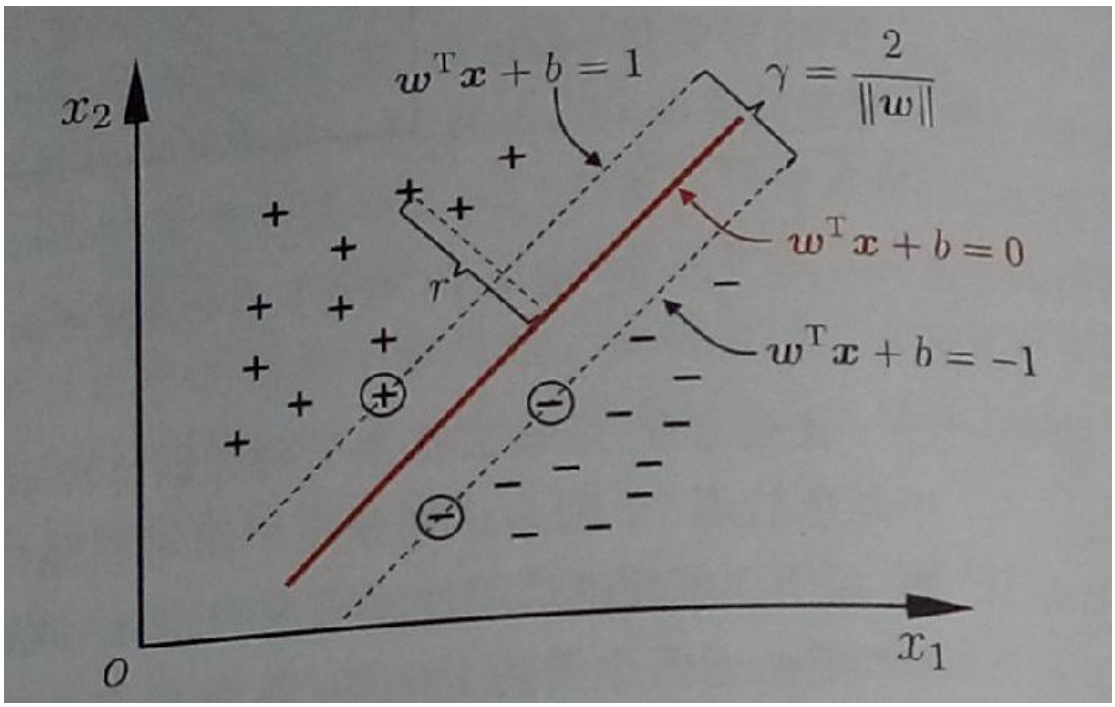

图 2：支持向量与间隔（摘自周志华著《机器学习》P122，图 6.2）

总之，现在我们的目标就是找到一组$\boldsymbol{\omega}$和 b，能在保证样本全部被正确分类

的情况下，使得前述“间隔”最大化，写成表达式即：

$$\begin{cases} \max\limits_{\boldsymbol{\omega},b} \dfrac{2\gamma}{\|\boldsymbol{\omega}\|} \\ \text{s.t. } y_i(\boldsymbol{\omega}^{\boldsymbol{T}}\boldsymbol{X}_i + b) \geq \gamma \end{cases} \qquad (2)$$

（2）式中的约束条件由（1）式而来，是为了保证样本全部被正确分类。（2）式中的γ是一个常数，其取值对优化问题没有任何影响。实际上，在**保证超平面实质上没有变化**时（因为理论上最优超平面只有一个，变化了就不是最优的了），可以同比例放缩$\boldsymbol{\omega}$和 b，即将超平面方程变为$(\lambda\boldsymbol{\omega})^{\boldsymbol{T}}\boldsymbol{X} + \lambda b = 0$。此时为了保证（1）式仍然成立，其中的$\gamma$要相应调整为$\lambda\gamma$，这样一来，我们的“间隔”变为$\frac{2\lambda\gamma}{\lambda\|\boldsymbol{\omega}\|}$，不会有变化，约束条件变为$y_i((\lambda\boldsymbol{\omega})^{\boldsymbol{T}}\boldsymbol{X}_i + \lambda b) \geq \lambda\gamma$，也没有变化。因此，$\gamma$的取值不会对优化问题产生影响。为了方便，就将其取为 1。实际上，这里的γ就是所谓的“函数间隔”，不过所以本文不打算采用这个概念。

至此，我们应该理解力支持向量机基本的思想和接下来要解决的优化问题了。想要深入了解关于“间隔”、“函数间隔”、“几何间隔”以及最优超平面的唯一性证明等问题，可参考李航著《统计学习方法》第 7 章相关内容。

最大间隔优化问题的对偶解法

前文（2）式给出了在**线性可分**假设及要求**分类错误率为 0** 时需要解决的优化问题。由于最大化$\frac{2}{\|\boldsymbol{\omega}\|}$与最小化$\frac{1}{2}\|\boldsymbol{\omega}\|^2$是等价的，故为了数学上处理方便，将（2）式的优化问题转化为：

$$\begin{cases} \min\limits_{\boldsymbol{\omega},b} \dfrac{1}{2}\|\boldsymbol{\omega}\|^2 \\ \text{s.t. } y_i(\boldsymbol{\omega}^{\boldsymbol{T}}\boldsymbol{X}_i + b) \geq 1 \end{cases} \qquad (3)$$

（3）式是一个凸二次规划问题（想要了解二次规划问题 == 就去学习最优化理论相关知识吧），可以用现成的二次规划方法求解。但为了追求求解效率，通常采用拉格朗日乘数法和对偶理论（要详细了解这些知识 = = 去补数学分析和运筹学吧）来求解。对（3）式引入拉格朗日乘数$\alpha_i \geq 0$，得到拉格朗日函数：

$$L(\boldsymbol{\omega},b,\boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{\omega}\|^2 + \sum_{i=1}^{m} \alpha_i (1 - y_i(\boldsymbol{\omega}^{\boldsymbol{T}}\boldsymbol{X}_i + b)) \qquad (4)$$

我们可以这样理解（4）式，等号右边的前半部分就是我们的优化目标，而后半

部分是当样本点不满足约束条件（即分类不正确）时的“惩罚项”。很明显，若有样本点$(\boldsymbol{X}_i,y_i)$不满足约束条件，即$1-y_i(\boldsymbol{\omega}^T\boldsymbol{X}_i+b)>0$，那么只要$\alpha_i$的值取很大，“惩罚项”的值可以达到$+\infty$，此时由于约束条件不再满足，这个最优化问题便没有最优解了，同时$\min L(\boldsymbol{\omega},b,\boldsymbol{\alpha})$自然也没有最优解了。若所有样本点都满足约束条件，则“惩罚项”必小于等于0，若令所有α_i均为0，此时“惩罚项”可取到最大值0，并且此时最小化$L(\boldsymbol{\omega},b,\boldsymbol{\alpha})$与最小化$\frac{1}{2}\|\boldsymbol{\omega}\|^2$是等价的。再整理一下，也就是说只有“惩罚项”的最大值为0（所有样本都满足约束条件）的时候我们才能最小化$L(\boldsymbol{\omega},b,\boldsymbol{\alpha})$，并且此时最小化$L(\boldsymbol{\omega},b,\boldsymbol{\alpha})$与最小化$\frac{1}{2}\|\boldsymbol{\omega}\|^2$是等价的，若“惩罚项”的最大值不为0（存在样本点不满足约束条件），那么优化问题没有最优解。因此，我们必须“迫使”“惩罚项”的最大值为0，然后再求解$L(\boldsymbol{\omega},b,\boldsymbol{\alpha})$的最小值，也即$\frac{1}{2}\|\boldsymbol{\omega}\|^2$的最小值。因此原来的优化问题（3）式就通过（4）式转化为：

$$\min_{\boldsymbol{\omega},b}\max_{\boldsymbol{\alpha}}\ \frac{1}{2}\|\boldsymbol{\omega}\|^2+\sum_{i=1}^{m}\alpha_i\left(1-y_i(\boldsymbol{\omega}^T\boldsymbol{X}_i+b)\right)\qquad(5)$$

现在我们的目标是求解（5）式即可。根据对偶理论，（5）式的对偶问题为：

$$\max_{\boldsymbol{\alpha}}\min_{\boldsymbol{\omega},b}\ \frac{1}{2}\|\boldsymbol{\omega}\|^2+\sum_{i=1}^{m}\alpha_i\left(1-y_i(\boldsymbol{\omega}^T\boldsymbol{X}_i+b)\right)\qquad(6)$$

请注意，这里（6）式和（5）式并不是等价的，实际上，若令：

$$\min_{\boldsymbol{\omega},b}\max_{\boldsymbol{\alpha}}\ \frac{1}{2}\|\boldsymbol{\omega}\|^2+\sum_{i=1}^{m}\alpha_i\left(1-y_i(\boldsymbol{\omega}^T\boldsymbol{X}_i+b)\right)=p*$$

$$\max_{\boldsymbol{\alpha}}\min_{\boldsymbol{\omega},b}\ \frac{1}{2}\|\boldsymbol{\omega}\|^2+\sum_{i=1}^{m}\alpha_i\left(1-y_i(\boldsymbol{\omega}^T\boldsymbol{X}_i+b)\right)=q*$$

则$q*\leq p*$，这可以简单的这样理解：最大值中最小的一个总会大于或等于最小值中最大一个。也就是说（6）式的最优值$q*$是（5）式的最优值$p*$的下界。在有最优解并且满足KKT条件的情况下，这两个值是相等的（想知道为啥 == 啥是KKT条件 = = 继续啃优化理论吧）。这样一来，我们就可以通过求解（6）式间接求解（5）式，也即我们原来的优化问题（3）式到现在为止已经转化成了（6）式。

从表达式就可以看到，求解（6）要分两步走：首先求解 $L(\boldsymbol{\omega},b,\boldsymbol{\alpha})$对$(\boldsymbol{\omega},b)$的最小化，再求解其对$\boldsymbol{\alpha}$的最大化。

首先来求$\min\limits_{\boldsymbol{\omega},b} L(\boldsymbol{\omega},b,\boldsymbol{\alpha})$。$L(\boldsymbol{\omega},b,\boldsymbol{\alpha})$对$\boldsymbol{\omega}$和 b 分别求偏导数并令其等于 0，容易得到：

$$\frac{\partial(L(\boldsymbol{\omega},b,\boldsymbol{\alpha}))}{\partial\boldsymbol{\omega}}=0\Rightarrow\boldsymbol{\omega}=\sum_{i=1}^{m}\alpha_i y_i\boldsymbol{X}_i \quad (7)$$

$$\frac{\partial(L(\boldsymbol{\omega},b,\boldsymbol{\alpha}))}{\partial b}=0\Rightarrow\sum_{i=1}^{m}\alpha_i y_i=0 \quad (8)$$

将（7），（8）式带入 $L(\boldsymbol{\omega},b,\boldsymbol{\alpha})$得到：

$$\min_{\boldsymbol{\omega},b} L(\boldsymbol{\omega},b,\boldsymbol{\alpha})=\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j\boldsymbol{X}_i^{\boldsymbol{T}}\boldsymbol{X}_j \quad (9)$$

得到（9）式的过程如下：

$$\begin{aligned}\frac{1}{2}\|\boldsymbol{\omega}\|^2+\sum_{i=1}^{m}\alpha_i\left(1-y_i(\boldsymbol{\omega}^{\boldsymbol{T}}\boldsymbol{X}_i+b)\right)&=\frac{1}{2}\boldsymbol{\omega}^{\boldsymbol{T}}\boldsymbol{\omega}+\sum_{i=1}^{m}\alpha_i-\sum_{i=1}^{m}\alpha_i y_i(\boldsymbol{\omega}^{\boldsymbol{T}}\boldsymbol{X}_i+b)\\&=\sum_{i=1}^{m}\alpha_i+\frac{1}{2}\left(\sum_{i=1}^{m}\alpha_i y_i\boldsymbol{X}_i^{\boldsymbol{T}}\right)\left(\sum_{j=1}^{m}\alpha_j y_j\boldsymbol{X}_j\right)\\&\quad-\left(\sum_{i=1}^{m}\alpha_i y_i\right)\left(\sum_{j=1}^{m}\alpha_j y_j\boldsymbol{X}_j^{\boldsymbol{T}}\boldsymbol{X}_i\right)-b\sum_{i=1}^{m}\alpha_i y_i\\&=\sum_{i=1}^{m}\alpha_i+\frac{1}{2}\left(\sum_{i=1}^{m}\alpha_i y_i\boldsymbol{X}_i^{\boldsymbol{T}}\right)\left(\sum_{j=1}^{m}\alpha_j y_j\boldsymbol{X}_j\right)\\&\quad-\left(\sum_{i=1}^{m}\alpha_i y_i\boldsymbol{X}_i^{\boldsymbol{T}}\right)\left(\sum_{j=1}^{m}\alpha_j y_j\boldsymbol{X}_j\right)-0=\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j\boldsymbol{X}_i^{\boldsymbol{T}}\boldsymbol{X}_j\end{aligned}$$

要注意的是，（8）式虽然是通过 $L(\boldsymbol{\omega},b,\boldsymbol{\alpha})$对 b 求偏导得到的，但是其并没有给出 b 的表达式，而是通过偏导数为 0 导出了一个约束条件：$\sum_{i=1}^{m}\alpha_i y_i=0$。这个约束条件在下一步求解过程中要用到。

现在求得了$\min\limits_{\boldsymbol{\omega},b} L(\boldsymbol{\omega},b,\boldsymbol{\alpha})$，根据（6）式，接下来就要求$\max\limits_{\boldsymbol{\alpha}}\min\limits_{\boldsymbol{\omega},b} L(\boldsymbol{\omega},b,\boldsymbol{\alpha})$。结合约束条件（8），即求解：

$$\begin{cases} \max\limits_{\boldsymbol{\alpha}} \sum\limits_{i=1}^{m} \alpha_i - \frac{1}{2}\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{m} \alpha_i\alpha_j y_i y_j \boldsymbol{X_i^T X_j} \\ \text{s.t.} \sum\limits_{i=1}^{m} \alpha_i y_i = 0 \\ \alpha_i \geq 0,\ i = 1,2,\cdots,m \end{cases} \tag{10}$$

从优化问题（10）解出来的是$\boldsymbol{\alpha}$，然后可求得$\boldsymbol{\omega}$和b（通过（7）式求$\boldsymbol{\omega}$，b的求解稍后会讲），这样就可以得到我们想要的决策模型：

$$f(\boldsymbol{X}) = \boldsymbol{\omega^T X} + b = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{X_i^T X} + b \tag{11}$$

即我们可以放一个新样本进来，然后计算$f(\boldsymbol{X})$，如果其值大于0，则其类别可判断为“+”，如果其值小于0，则其类别可判断为“−”（如果忘了模型最初的想法= = 看看前文（1）式前面一段）。

前文提到对偶问题（5）式和（6）式的最优解相等的前提是满足KKT条件，这里，这个KKT条件就是（如果不是特别想学习KKT条件相关内容 = = 可以暂时不必深究这是怎么来的）对所有$i = 1,2,\cdots,m$，需要满足：

$$\begin{cases} \alpha_i \geq 0 \\ y_i f(\boldsymbol{X_i}) - 1 \geq 0 \\ \alpha_i (y_i f(\boldsymbol{X_i}) - 1) = 0 \end{cases}$$

从这个KKT条件可以知道，总有$\alpha_i = 0$或$y_i f(\boldsymbol{X_i}) = 1$，若$\alpha_i = 0$，其对应的样本对（11）式中的求和没有任何影响，也就是说该样本不会影响我们的决策模型。若$\alpha_i > 0$，则$y_i f(\boldsymbol{X_i}) = 1$，其所对应的样本刚好做最大“间隔”的边界上，也即是一个“支持向量”，它对（11）式的求和将产生影响，也就是说能影响我们的决策模型。这就显示了支持向量机的一个很好的性质：训练完成后，决策结果仅与“支持向量”有关，而与大多数训练样本无关，所以这些样本可以不必保留。这直观上也很好理解，看看图2，我们做决策时只需计算样本离划分超平面的距离，而“支持向量”是在超平面边界上的点，所以这些点保留了超平面的信息，因此做决策时仅需要这些“支持向量”即可。

现在来看看怎么通过求解出来的$\boldsymbol{\alpha}$来求解b。从KKT条件的分析可以知道，要么有$\alpha_i = 0$，要么有$\alpha_i > 0$。那可有没有可能所有的α_i都等于0呢？这当然是不可能的，因为如果所以的α_i都为0，根据（7）式，有$\boldsymbol{\omega} = \boldsymbol{0}$，这显然不是问题的最优解。因此，在有最优解的情况下，必存在$\alpha_j > 0$。选取这样一个大于0的

α_j（易知其所对应的样本是一个“支持向量”），依据$y_j f(\boldsymbol{X}_j) = y_j(\sum_{i=1}^{m} \alpha_i y_i \boldsymbol{X}_i^{\boldsymbol{T}} \boldsymbol{X}_j + b) = 1$ 以及$y_j{}^2 = 1$，容易求得 b 的值为：

$$b = y_j - \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{X}_i^{\boldsymbol{T}} \boldsymbol{X}_j \qquad (12)$$

显然，用这种方式，理论上任何一个“支持向量”均可计算 b。在实际使用中，常用的做法是采用所有“支持向量”求解得到的 b 的平均值作为最终的 b 值在决策模型（11）中使用。

到目前为止，我们已经把求解支持向量机要优化的问题转化为（10）式了。我们还讨论了需要满足的 KKT 条件，以及（10）式求解完成之后如何将其结果运用到决策模型中。现在我们先不考虑如何求解（10）式，先来考虑下所谓的“软间隔”问题。

软间隔（正则化）

在前面的讨论中，我们有一个假设是**线性可分**并且**所有样本都能被正确分类**，此时的间隔叫做“硬间隔”。实际中，这个假设条件过于严格，可能导致无法得到最优解，或者就算得到了最优解，过拟合的风险也很大。如图 3 所示，为了把模型应用到并不是严格线性可分的数据集上，一个很好的办法就是允许一些样本出现在“间隔”内部（更准确的说，前文所讲的**所有样本都被正确分类，其实是要所有的样本都不能出现在“间隔之内”**），这样可以降低过拟合风险，提高模型的鲁棒性。“软间隔”就是允许一些样本可以不满足（3）式中的约束条件：$y_i(\boldsymbol{\omega}^{\boldsymbol{T}} \boldsymbol{X}_i + b) \geq 1$。引入松弛变量变量$\xi_i \geq 0$ 来表征样本不满足（3）式中约束条件的程度，可以将新的约束条件写成：$y_i(\boldsymbol{\omega}^{\boldsymbol{T}} \boldsymbol{X}_i + b) \geq 1 - \xi_i$。

当然，放宽约束条件并不是免费的，因为这同样可能导致欠拟合风险，所以需要对此进行惩罚。惩罚办法就是在优化目标上加一个“惩罚项”（这种加惩罚项的方式也可以叫做所谓的“正则化”）：$\sum_{i=1}^{m} \xi_i$。这样，“软间隔”情况下的优化问题就可以写成：

$$\begin{cases} \min\limits_{\boldsymbol{\omega},b,\xi} \dfrac{1}{2}\|\boldsymbol{\omega}\|^2 + C\sum\limits_{i=1}^{m}\xi_i \\ \text{s.t. } y_i(\boldsymbol{\omega}^{\boldsymbol{T}}\boldsymbol{X}_i + b) \geq 1 - \xi_i \\ \xi_i \geq 0, i = 1,2,\cdots,m \end{cases} \tag{13}$$

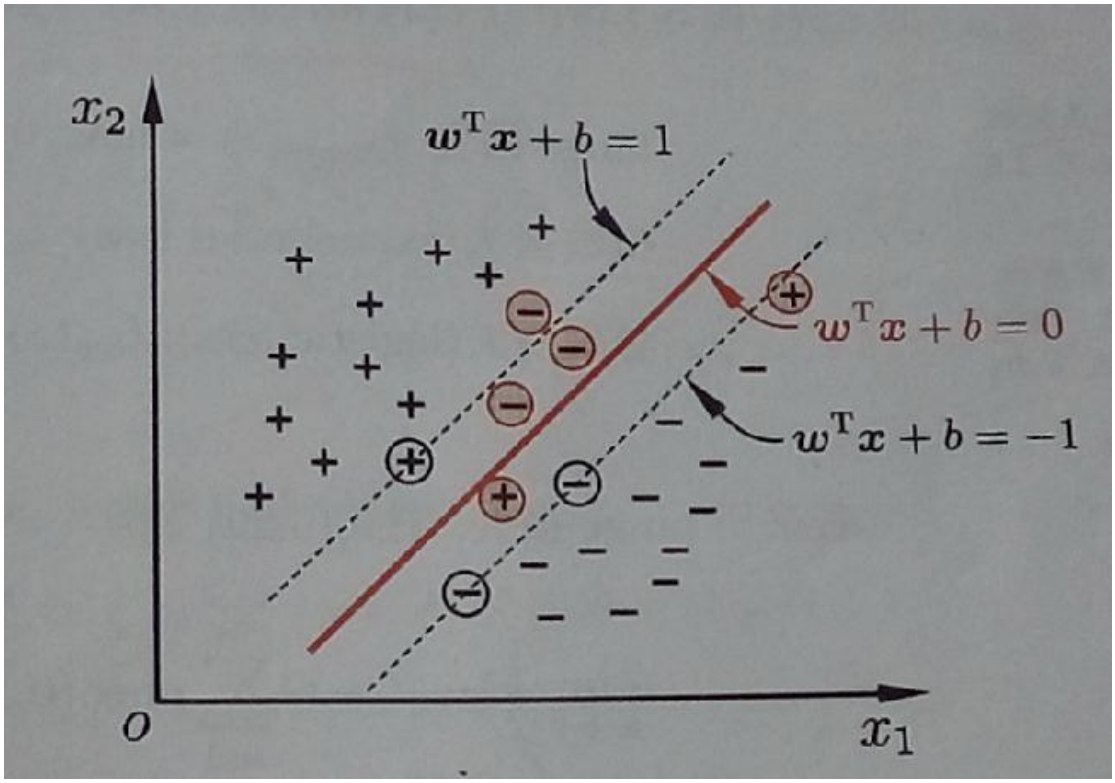

图 3：软间隔（红圈表示不满足约束的样本）（摘自周志华著《机器学习》P129，图 6.4）

（13）式中，常数 $C > 0$ 是惩罚常数，其作用是权衡惩罚项与最大化间隔的重要性。实际上，（13）式中需要最小化的目标函数包含个部分：前部分是为了使“间隔”尽量打，后一部分是为了让尽量少的样本出现在“间隔”内部。很明显这两个目的是存在矛盾的，因为间隔越大，出现在间隔内部的样本数就越容易增多，为此引入常数 C 来调和这两个方面。C 的取值应视具体问题而定，当 C 很大时，相当于迫使所有样本都不落在“间隔”内部，即满足（3）式中的约束条件，当 C 取值不大时，相当于允许某些样本不满足（3）式中的约束条件。

另外需要注意的是，“惩罚项”（正则化）实际上就是所谓的“损失函数”，损失函数有许多，本文不打算罗列出来，要详细了解损失函数以及结构化风险等内容，可参阅相关资料（周志华著《机器学习》，李航著《统计学习方法》）。（13）式中使用正则化项使用的是 hinge 损失函数（合页损失函数）：$\ell_{hinge}(z) =$

$\max(0,1-z)$，此函数中 $1-z$ 其实就代表了损失程度。在分类问题中，我们可将损失记为 $1-y_i(\boldsymbol{\omega}^T\boldsymbol{X}_i+b)$，因此直观上可将（13）式中的$\xi_i$看成是损失项$\max(0,1-zy_i(\boldsymbol{\omega}^T\boldsymbol{X}_i+b))$的替代。

总之，在“软间隔”思想之下，我们现在有了新的优化问题（13）式。对（13）式引入拉格朗日乘数，得到拉格朗日函数：

$$L(\boldsymbol{\omega},b,\boldsymbol{\alpha},\xi,\boldsymbol{\mu})=\frac{1}{2}\|\boldsymbol{\omega}\|^2+C\sum_{i=1}^{m}\xi_i$$
$$+\sum_{i=1}^{m}\alpha_i\left(1-\xi_i-y_i(\boldsymbol{\omega}^T\boldsymbol{X}_i+b)\right)-\sum_{i=1}^{m}\mu_i\xi_i \quad (14)$$

与“硬间隔”下的情况类似，求解（13）式相当于求解如下问题：

$$\min_{\boldsymbol{\omega},b,\xi}\ \max_{\boldsymbol{\alpha}}\ L(\boldsymbol{\omega},b,\boldsymbol{\alpha},\xi,\boldsymbol{\mu}) \quad (15)$$

（15）式的对偶问题为：

$$\max_{\boldsymbol{\alpha}}\ \min_{\boldsymbol{\omega},b,\xi}\ L(\boldsymbol{\omega},b,\boldsymbol{\alpha},\xi,\boldsymbol{\mu}) \quad (16)$$

求解（16）式同样分两步走。首先求$\min_{\boldsymbol{\omega},b,\xi} L(\boldsymbol{\omega},b,\boldsymbol{\alpha},\xi,\boldsymbol{\mu})$。$L(\boldsymbol{\omega},b,\boldsymbol{\alpha},\xi,\boldsymbol{\mu})$对$\boldsymbol{\omega},b,\xi_i$求偏导数并令其为0，得到：

$$\frac{\partial(L(\boldsymbol{\omega},b,\boldsymbol{\alpha},\xi,\boldsymbol{\mu}))}{\partial\boldsymbol{\omega}}=0\Rightarrow\boldsymbol{\omega}=\sum_{i=1}^{m}\alpha_i y_i\boldsymbol{X}_i \quad (17)$$

$$\frac{\partial(L(\boldsymbol{\omega},b,\boldsymbol{\alpha},\xi,\boldsymbol{\mu}))}{\partial b}=0\Rightarrow\sum_{i=1}^{m}\alpha_i y_i=0 \quad (18)$$

$$\frac{\partial(L(\boldsymbol{\omega},b,\boldsymbol{\alpha},\xi,\boldsymbol{\mu}))}{\partial\xi_i}=0\Rightarrow C=\alpha_i+\mu_i \quad (19)$$

将（17），（18），（19）式带入（14）式，得到：

$$\min_{\boldsymbol{\omega},b,\xi}L(\boldsymbol{\omega},b,\boldsymbol{\alpha},\xi,\boldsymbol{\mu})=\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j\boldsymbol{X}_i^T\boldsymbol{X}_j \quad (20)$$

同样，（18）式和（19）式导出了两个关于α_i的约束条件（由于拉格朗日乘数$\mu_i\geq 0$的值我们并不关心，因此从（19）式可以导出约束条件$0\leq\alpha_i\leq C$）。

现在求得了$\min_{\boldsymbol{\omega},b,\xi} L(\boldsymbol{\omega},b,\boldsymbol{\alpha},\xi,\boldsymbol{\mu})$，根据（16）式，接下来的任务就是要求解$\max_{\boldsymbol{\alpha}}\min_{\boldsymbol{\omega},b,\xi} L(\boldsymbol{\omega},b,\boldsymbol{\alpha},\xi,\boldsymbol{\mu})$。结合约束条件（18）式和（19）式，即求解：

$$\begin{cases} \max\limits_{\boldsymbol{\alpha}} \sum\limits_{i=1}^{m} \alpha_i - \frac{1}{2}\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{m} \alpha_i\alpha_j y_i y_j \boldsymbol{X}_i^{\boldsymbol{T}} \boldsymbol{X}_j \\ \text{s.t.} \sum\limits_{i=1}^{m} \alpha_i y_i = 0 \\ 0 \le \alpha_i \le C,\ i = 1,2,\cdots,m \end{cases} \tag{21}$$

可以看到，虽然（21）式是在“软间隔”情况下导出的优化问题，但其与（10）式的差别仅在最后一个约束条件，即α_i的上界被约束为C。

从（21）式求解出$\boldsymbol{\alpha}$之后，求解$\boldsymbol{\omega}$和b的方法跟“硬间隔”情况是一样的，这样我们就可以根据（11）式构建决策函数了。

当然，（21）式的解就是（13）式的解的条件仍然是要满足KKT条件，即对所有$i = 1,2,\cdots,m$，需要满足::

$$\begin{cases} \alpha_i \ge 0,\ \mu_i \ge 0 \\ y_i f(\boldsymbol{X}_i) - 1 + \xi_i \ge 0 \\ \alpha_i(y_i f(\boldsymbol{X}_i) - 1 + \xi_i) = 0 \\ \xi_i \ge 0,\ \mu_i \xi_i = 0 \end{cases} \tag{22}$$

从(22)式可知，对任意训练样本$(\boldsymbol{X}_i, y_i)$，总有$\alpha_i = 0$或$y_i f(\boldsymbol{X}_i) = 1 - \xi_i$，若$\alpha_i = 0$，其对应的样本对决策函数没有任何影响。若$\alpha_i > 0$，则$y_i f(\boldsymbol{X}_i) = 1 - \xi_i$，此时该样本是一个“支持向量”，它将影响我们的决策模型。由（19）式可知，若$\alpha_i < C$，则$\mu_i > 0$，进而有$\xi_i = 0$，即该样本恰好做“间隔”边界上；若$\alpha_i = C$，则有$\mu_i = 0$，此时若$\xi_i < 1$，则该样本被正确分类，但会落在间隔内部；若$\xi_i = 1$，则样本刚好落在划分超平面上；若$\xi_i > 1$，则样本将被错误分类。由此可以看到，“软间隔”支持向量机最终的决策也仅与支持向量有关，即通过合页损失函数仍然保持了稀疏性。

到现在为止，只要解出了（21）式，我们就可以得到“软间隔”支持向量机模型并可以用其进行决策，但现在我们还是先不考虑如何继续求解（21）式，先来了解线性不可分问题以及核方法之后再求解。

线性不可分与核方法

一开始，我们用“**硬间隔**”假设**所有样本都不能出现在间隔之内**，然后“**软间隔**”思想**允许部分样本出现在间隔之内，甚至允许出现分类错误**。在这两种情

况下，我们实际上都是假设数据集是“**线性可分**”的（“软间隔”虽然不要求严格线性可分，但是总体上仍然是线性可分的）。但实际上，“线性可分”条件并不一定满足。如图 4 所示，在左边的二维空间(x,y)内，我们无法找到一条直线将两类样本划分开来。如果将原始数据在三维空间(x,y,z)里考虑，那么所有样本在 z 轴上的取值均相等，我们就取其为 0。现在如果通过一个映射使红色圆圈那一类样本沿着 z 轴做一个平移，使其在 z 轴上的取值变为 ℤ，那么在 z 轴上(0,Z)内取一点，作一个垂直于 z 轴的平面，该平面就可以把样本划分为两类，如图 4 右部分所示。

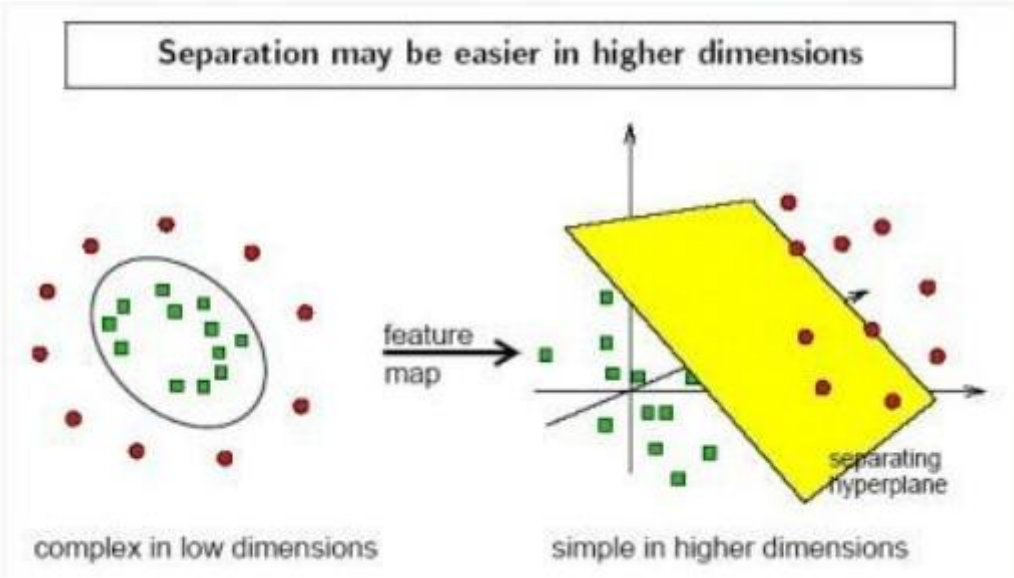

图 4：线性不可分情况（摘自博文《支持向量机通俗导论（理解 SVM 的三层境界）》）

上例中，在二维空间内线性不可分的数据被映射到三维空间之后变得线性可分了，我们把原来的二维空间叫做原始空间，映射之后的三维空间叫做“特征空间”。就像这个例子一样，很幸运的是，如果原始空间是有限维的，即属性数目是有限的，那么一定可以将原始数据映射到一个更高维的特征空间，在这个特征空间里样本能够线性可分。

现在，假设线性不可分的数据 $\boldsymbol{X}$ 可以通过函数ϕ映射到线性可分的特征空间，即在特征空间中，我们的数据变成了$\phi(\boldsymbol{X})$。既然在特征空间内能够线性可分，那么我们只需在特征空间内建模和决策就行了。因此，我们的优化问题就转化为：

$$\begin{cases} \max\limits_{\boldsymbol{\alpha}} \sum\limits_{i=1}^{m} \alpha_i - \dfrac{1}{2}\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{m} \alpha_i\alpha_j y_i y_j \boldsymbol{\phi}(\boldsymbol{X}_i)^{\boldsymbol{T}}\, \boldsymbol{\phi}(\boldsymbol{X}_j) \\ \text{s.t.} \sum\limits_{i=1}^{m} \alpha_i y_i = 0 \\ 0 \le \alpha_i \le C,\ i = 1,2,\cdots,m \end{cases} \tag{23}$$

（23）式其实就是将（21）中的 $\boldsymbol{X}$ 替换成了$\boldsymbol{\phi}(\boldsymbol{X})$。实际上，相应的 KKT 条件、决策模型以及$\boldsymbol{\omega}$,$b$ 的求解形式都不会变，只需将 $\boldsymbol{X}$ 换成$\boldsymbol{\phi}(\boldsymbol{X})$即可。

现在，一切看起来都很顺利，只需要在特征空间$\boldsymbol{\phi}(\boldsymbol{X})$中将（23）式求解出来即可。但是，这个前提是我们要先找到映射函数$\boldsymbol{\phi}$，而这通常是困难的。还有一问题是，映射完成之后，特征空间的维数比原始空间的维数高（甚至可能是无穷维的），这就给计算内积$\boldsymbol{\phi}(\boldsymbol{X}_i)^{\boldsymbol{T}}\boldsymbol{\phi}(\boldsymbol{X}_j)$造成了极大的困难（甚至在无穷维的情况下无法计算）。

由于在求解计算过程中，我们只需要计算内积$\boldsymbol{\phi}(\boldsymbol{X}_i)^{\boldsymbol{T}}\boldsymbol{\phi}(\boldsymbol{X}_j)$，而不必关心$\boldsymbol{\phi}$的具体形式，因此我们可以不必先找出映射函数$\boldsymbol{\phi}$再进行计算求解，而可以通过间接的方法来计算$\boldsymbol{\phi}(\boldsymbol{X}_i)^{\boldsymbol{T}}\boldsymbol{\phi}(\boldsymbol{X}_j)$，这个间接的方法就是所谓的核方法。

由于我们想要间接计算的是$\boldsymbol{\phi}(\boldsymbol{X}_i)^{\boldsymbol{T}}\boldsymbol{\phi}(\boldsymbol{X}_j)$，现在假设有一个函数满足我们的需要，即存在函数 K 使 $K(\boldsymbol{X}_i,\boldsymbol{X}_j) = \boldsymbol{\phi}(\boldsymbol{X}_i)^{\boldsymbol{T}}\boldsymbol{\phi}(\boldsymbol{X}_j)$成立，这个函数 K 就是所谓的“核函数”。注意，这里是将计算$\boldsymbol{\phi}(\boldsymbol{X}_i)^{\boldsymbol{T}}\boldsymbol{\phi}(\boldsymbol{X}_j)$转化成了计算 $K(\boldsymbol{X}_i,\boldsymbol{X}_j)$，而计算 $K(\boldsymbol{X}_i,\boldsymbol{X}_j)$不是在特征空间进行的，而是在原始空间进行的，这样就没有增加计算的维度成本。那么，这样的核函数 K 存在吗。很幸运，下面的这个定理告诉我们，核函数 K 一定存在。

核函数存在定理：令 $\boldsymbol{x}$ 为原始空间，$K(\cdot,\cdot)$是定义在 $\boldsymbol{x}\times\boldsymbol{x}$ 上的**对称函数**，则 K 是核函数**当且仅当**对于任意数据集$\{\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots\cdots,\boldsymbol{X}_m\}$，核矩阵$\left[K(\boldsymbol{X}_i,\boldsymbol{X}_j)\right]_{m\times m}$总是**半正定**的。

定理中的对称函数是指须满足 $K(\boldsymbol{X}_i,\boldsymbol{X}_j) = K(\boldsymbol{X}_j,\boldsymbol{X}_i)$，关于矩阵论的相关理论= = 去补代数的相关知识吧。总之，这个定理告诉我们，只要一个对称函数 K 所对应的矩阵$\left[K(\boldsymbol{X}_i,\boldsymbol{X}_j)\right]_{m\times m}$是半正定的，那么这个函数 K 就可以作为核函数使

用，即一定能找到一个映射$\boldsymbol{\phi}$使得$K(\boldsymbol{X}_i,\boldsymbol{X}_j)=\boldsymbol{\phi}(\boldsymbol{X}_i)^{\boldsymbol{T}}\boldsymbol{\phi}(\boldsymbol{X}_j)$成立。

现在，由于我们不需要去找出映射$\boldsymbol{\phi}$的具体形式，因此核函数$K(\boldsymbol{X}_i,\boldsymbol{X}_j)$就相当于隐式的定义了我们需要的“特征空间”。所以核函数的选择变得十分重要，因为如果核函数选择不当，相当于样本会被映射到一个不适合的“特征空间”，这将导致模型性能不佳。常用的核函数有线性核、多项式核、高斯核、拉普拉斯核、Sigmoid 核等，另外核函数的线性组合及一些其他变换得到的函数也能作为核函数使用（想要学习更多的话 = = 去查资料吧）。

我们利用核函数$K(\boldsymbol{X}_i,\boldsymbol{X}_j)$解决了特征空间中的内积$\boldsymbol{\phi}(\boldsymbol{X}_i)^{\boldsymbol{T}}\boldsymbol{\phi}(\boldsymbol{X}_j)$的计算问题。那么现在我们的优化问题就变成了：

$$\begin{cases}\max\limits_{\boldsymbol{\alpha}} \sum\limits_{i=1}^{m}\alpha_i-\dfrac{1}{2}\sum\limits_{i=1}^{m}\sum\limits_{j=1}^{m}\alpha_i\alpha_j y_i y_j K(\boldsymbol{X}_i,\boldsymbol{X}_j)\\ \text{s.t.}\ \sum\limits_{i=1}^{m}\alpha_i y_i=0\\ 0\le\alpha_i\le C,\ i=1,2,\cdots,m\end{cases}\tag{24}$$

决策函数变成了：

$$f(\boldsymbol{X})=\boldsymbol{\omega}^{\boldsymbol{T}}\boldsymbol{\phi}(\boldsymbol{X})+b=\sum_{i=1}^{m}\alpha_i y_i\boldsymbol{\phi}(\boldsymbol{X}_i)^{\boldsymbol{T}}\boldsymbol{\phi}(\boldsymbol{X})+b=\sum_{i=1}^{m}\alpha_i y_i K(\boldsymbol{X}_i,\boldsymbol{X})+b\tag{25}$$

其中b的计算方法为：

$$b=y_j-\sum_{i=1}^{m}\alpha_i y_i K(\boldsymbol{X}_i,\boldsymbol{X}_j)\tag{26}$$

当然，b仍采用所有支持向量计算得到的b的均值。KKT 条件仍为：

$$\begin{cases}\alpha_i\ge 0,\ \mu_i\ge 0\\ y_i f(\boldsymbol{X}_i)-1+\xi_i\ge 0\\ \alpha_i(y_i f(\boldsymbol{X}_i)-1+\xi_i)=0\\ \xi_i\ge 0,\ \mu_i\xi_i=0\end{cases}\tag{27}$$

不过此处的决策函数f是（25）式中的f。

到现在为止，我们从“硬间隔”到“软间隔”，再到巧妙利用“核函数”解决线性不可分问题，最终得到了优化问题（24）式，只要将（24）式求解出来，我们就搞定了！

利用 SMO（序列最小优化）算法求解支持向量机

现在，我们的目的就是要求解（24）式。从形式上看，（24）是仍然是一个二次规划问题，但是为了提高求解效率，我们不用现有的二次规划算法来求解，而是采用一种启发式算法 SMO（序列最小优化）来求解。需要注意的是，由于启发式算法带有一定随机性，故每次求解的结果可能会有差异，但这并不会有大的影响。接下来只要学习完 SMO 算法，支持向量机最基本的理论和求解过程就被我们掌握啦！

优化问题（24）中，我们最终要求解的是 m 个α_i，一个α_i对应着一个样本。SMO 算法的基本思想是：如果所有的样本都满足（27）式中的 KKT 条件，那么也就能得到优化问题的解了，这是因为 KKT 条件是该优化问题有最优解的充要条件。SMO 算法不同时求解所有的α_i，而是依据（24）式中的第一个约束条件，每次选取一对α_i,α_j，固定其他参数，然后对选取的一对α_i,α_j进行优化。不断循环这个过程，直至收敛，就得到了最优解。

SMO 算法为什么高效呢，因为固定其他参数之后，优化问题（24）就相当于求解一个二次函数的极值问题！具体地，我们来看看（24）式，当仅考虑α_i,α_j而固定其他参数时，由于${y_i}^2 = {y_j}^2 = 1$，（24）式就转化成了：

$$\begin{cases} \max\limits_{\alpha_i,\alpha_j} \ \alpha_i + \alpha_j - \frac{1}{2}\alpha_i^2 K_{ii} - \frac{1}{2}\alpha_j^2 K_{jj} - \alpha_i\alpha_j y_i y_j K_{ij} \\ \qquad\qquad - \alpha_i y_i \sum\limits_{k\neq i,j}^{m} \alpha_k y_k K_{ik} - \alpha_j y_j \sum\limits_{k\neq i,j}^{m} \alpha_k y_k K_{jk} \\ \text{s.t. } \alpha_i y_i + \alpha_j y_j = -\sum\limits_{k\neq i,j}^{m} \alpha_k y_k = C \\ \qquad 0 \le \alpha_i,\alpha_j \le C \end{cases} \tag{28}$$

其中记号$K_{pq} = K(\boldsymbol{X}_p,\boldsymbol{X}_q) = K(\boldsymbol{X}_q,\boldsymbol{X}_p)$。看（28）式，通过等式约束，我们可以用$\alpha_j$来表示$\alpha_i$：

$$\alpha_i = y_i(C - \alpha_j y_j) \tag{29}$$

将（29）带入（28）式中的目标函数中，我们就得到了一个关于单变量α_j的二次

优化问题，其约束条件仅仅为 $0 \le \alpha_j \le C$：

$$\max_{\alpha_j} W(\alpha_j) = y_i(\mathcal{C} - \alpha_j y_j) + \alpha_j - \frac{1}{2}(\mathcal{C} - \alpha_j y_j)^2 K_{ii} - \frac{1}{2}\alpha_j^2 K_{jj} - (\mathcal{C} - \alpha_j y_j)\alpha_j y_j K_{ij} - (\mathcal{C} - \alpha_j y_j)\boldsymbol{V}_i - \alpha_j y_j \boldsymbol{V}_j \quad (30)$$

其中记号$\boldsymbol{V}_p = \sum_{k \ne i,j}^{m} \alpha_k y_k K_{pk} = f(\boldsymbol{X}_p) - \alpha_i y_i K_{pi} - \alpha_j y_j K_{pj} - b$。（30）式中当固定其它参数时就是一个关于$\alpha_j$的二次函数极值问题！$W(\alpha_j)$对$\alpha_j$求导并整理得到：

$$\frac{\partial W}{\partial \alpha_j} = -K_{ii}\alpha_j - K_{jj}\alpha_j + 2K_{ij}\alpha_j + K_{ii}\mathcal{C}y_j - K_{ij}\mathcal{C}y_j - y_i y_j + 1 + \boldsymbol{V}_i y_j - \boldsymbol{V}_j y_j \quad (31)$$

令（30）式中导数为 0，并记$\eta = K_{ii} + K_{jj} - 2K_{ij}$，得到：

$$\begin{aligned}\eta\alpha_j &= y_j(y_j - y_i + \mathcal{C}K_{ii} - \mathcal{C}K_{ij} + \boldsymbol{V}_i - \boldsymbol{V}_j) \\ &= y_j(y_j - y_i + \mathcal{C}K_{ii} - \mathcal{C}K_{ij} + (f(\boldsymbol{X}_i) - \alpha_i y_i K_{ii} - \alpha_j y_j K_{ij} - b) \\ &\quad - (f(\boldsymbol{X}_j) - \alpha_i y_i K_{ij} - \alpha_j y_j K_{jj} - b)) \quad (32)\end{aligned}$$

注意（32）式中的α_j不是原来的（未优化的）α_j，而是优化之后的新值。为了方便，记α_i,α_j更新之后的值为α_i^{new},α_j^{new}，记$E_p = f(\boldsymbol{X}_p) - y_p$，并将 $\mathcal{C} = \alpha_i y_i + \alpha_j y_j$ 带入（32）式中，整理得到：

$$\eta\alpha_j^{new} = y_j(\eta\alpha_j y_j + y_j - y_i + f(\boldsymbol{X}_i) - f(\boldsymbol{X}_j)) = \eta\alpha_j + y_j(E_i - E_j) \quad (33)$$

由（33）式可解得优化后的α_j^{new}为：

$$\alpha_j^{new} = \alpha_j + \frac{y_j(E_i - E_j)}{\eta} \quad (34)$$

（34）式已经告诉了我们α_j的更新公式，但还没有完，为啥呢？因为我们是在固定其它参数之后才优化α_i,α_j的，因此根据（24）式中的等式约束，必须有：

$$\alpha_i^{new} y_i + \alpha_j^{new} y_j = \alpha_i y_i + \alpha_j y_j = -\sum_{k \ne i,j}^{m} \alpha_k y_k = \mathcal{C} \quad (35)$$

根据（35）式，若$y_i = y_j$，则应有$\alpha_j^{new} = \alpha_i + \alpha_j - \alpha_i^{new}$，结合（24）式中的临

界值约束，α_j^{new}的下界应为$\max(0,\alpha_i+\alpha_j-C)$，上界应为$\min(C,\alpha_i+\alpha_j)$；若$y_i \neq y_j$，则应有$\alpha_j^{new}=\alpha_j-\alpha_i+\alpha_i^{new}$，结合（24）式中的临界值约束，$\alpha_j^{new}$的下界应为$\max(0,\alpha_j-\alpha_i)$，上界应为$\min(C,\alpha_j-\alpha_i+C)$。总结一下$\alpha_j^{new}$的界限值：

$$L=\begin{cases}\max(0,\alpha_i+\alpha_j-C), & 若y_i=y_j\\ \max(0,\alpha_j-\alpha_i), & 若y_i\neq y_j\end{cases} \tag{36}$$

$$H=\begin{cases}\min(C,\alpha_i+\alpha_j), & 若y_i=y_j\\ \min(C,\alpha_j-\alpha_i+C), & 若y_i\neq y_j\end{cases} \tag{37}$$

结合以上两个临界值以及（34）式，得到α_j^{new}的最终更新值$\alpha_j^{FinalNew}$为：

$$\alpha_j^{FinalNew}=\begin{cases}L, & 若\alpha_j^{new}<L\\ \alpha_j^{new}, & 若\ L\le\alpha_j^{new}\le H\\ H, & 若\alpha_j^{new}>H\end{cases} \tag{38}$$

现在我们从（38）式得经得到了α_j的最终更新公式！然后根据$\alpha_i^{FinalNew}y_i+\alpha_j^{FinalNew}y_j=\alpha_i y_i+\alpha_j y_j$就可以得到$\alpha_i$的更新公式：

$$\alpha_i^{FinalNew}=\alpha_i-y_iy_j\left(\alpha_j-\alpha_j^{FinalNew}\right) \tag{39}$$

现在，我们已经搞定了α_i,α_j的最终更新公式！就只差搞定 b 的更新公式就能利用SMO算法实现支持向量机了！

利用（26）式，可以得到α_i对应的 b 的更新公式为：

$$b_i^{new}=y_i-\sum_{k\neq i,j}^{m}\alpha_k y_k K(\boldsymbol{X}_k,\boldsymbol{X}_i)-\alpha_i^{FinalNew}y_iK(\boldsymbol{X}_i,\boldsymbol{X}_i)-\alpha_j^{FinalNew}y_jK(\boldsymbol{X}_j,\boldsymbol{X}_i) \tag{40}$$

α_j对应的 b 的更新公式为：

$$b_j^{new}=y_j-\sum_{k\neq i,j}^{m}\alpha_k y_k K(\boldsymbol{X}_k,\boldsymbol{X}_j)-\alpha_i^{FinalNew}y_iK(\boldsymbol{X}_i,\boldsymbol{X}_j)-\alpha_j^{FinalNew}y_jK(\boldsymbol{X}_j,\boldsymbol{X}_j) \tag{41}$$

结合前面的记号应有：$E_i=f(\boldsymbol{X}_i)-y_i=\sum_{k\neq i,j}^{m}\alpha_k y_k K(\boldsymbol{X}_k,\boldsymbol{X}_i)+\alpha_i y_i K(\boldsymbol{X}_i,\boldsymbol{X}_i)+\alpha_j y_j K(\boldsymbol{X}_j,\boldsymbol{X}_i)+b-y_i$，重新整理得到：

$$b_i^{new} = b - E_i - y_i K_{ii}(\alpha_i{}^{FinalNew} - \alpha_i) - y_j K_{ij}\left(\alpha_j{}^{FinalNew} - \alpha_j\right) \quad (42)$$

同理：

$$b_j^{new} = b - E_j - y_i K_{ij}(\alpha_i{}^{FinalNew} - \alpha_i) - y_j K_{jj}\left(\alpha_j{}^{FinalNew} - \alpha_j\right) \quad (43)$$

同样，在实际使用中，取两个更新值的均值来作为 b 最终的更新值：

$$b_j^{FinalNew} = \frac{b_i^{new} + b_j^{new}}{2} \quad (44)$$

现在，我们已经了解了 SMO 算法优化参数的整个过程，可以说已经整体上能实现 SVM 模型了。最后需要补充的就是α_i和α_j的选取方式。前面我们是在选定α_i和α_j之后对其进行优化，而具体怎么选取α_i和α_j对整个优化过程的计算效率有至关重要的影响。

首先来看看怎么选取α_i。直观上来看，如果一个样本违背 KKT 条件越严重，那么它最应该被优化，所以一种办法是每次选取违背 KKT 条件程度最大的样本来优化其α_i。在实际使用中，为了避免重复计算 KKT 条件，通常采取完整遍历和非边界循环相互切换的方式来选取α_i，即先遍历所有样本，对每个α_i进行优化，然后选取取值不在边界上那些的α_i（即不等于 0 或 C 的那些α_i）进行优化，如此循环直至收敛。

再来看看怎么选取α_j。从（34）式知道，α_j的更新幅度直接依赖于$E_i - E_j$，因此，为了加快收敛速度，当α_i确定之后，每次选择使$|E_i - E_j|$最大时对应的α_j作为第二个优化对象。

总结：

完整支持向量机算法是由简到难，循序渐进的一个过程：一开始的**“硬间隔”要求所有样本均在间隔之外**；然后利用**“软间隔”允许部分样本出现在间隔之内甚至允许分类错误，这样将模型扩展到了并非严格线性可分的数据集上**；**为了解决线性不可分问题，引入“特征空间”，将原始空间利用ϕ映射到特征空间，使其在特征空间内线性可分，这样我们只需要在特征空间内求解就行了**；然后**为了解决在特征空间中计算内积的困难，又巧妙利用“核函数”使计算过程在**

原始空间内就能完成；最后**利用高效的启发式优化算法 SMO 来求解支持向量机**。

关于 SVM 的实现，现在已经有许多成熟好用的包直接调用就可以了。虽然如此，为了加深对整个过程的理解，如果有时间的话，自己写写代码也是很有帮助的。

还有就是学习过程中发现好多基础知识都不懂。= = 。= = 。

参考资料

1.周志华著《机器学习》，清华大学出版社，2016

2.李航著《统计学习方法》，清华大学出版社，2012

3.网络博文《支持向量机通俗导论（理解 SVM 的三层境界）》，作者：July、pluskid 等。网址：http://blog.csdn.net/v_july_v/article/details/7624837。