机器学习（西瓜书）

注　解

（第 6 章　支持向量机）

https://blog.csdn.net/jbb0523

前言

经常听人说南大周老师所著的《机器学习》（以下统称为西瓜书）是一本入门教材，是一本科普性质的教科书。在该书第十次印刷之际，周老师在“如何使用本书”中也提到“这是一本入门级教科书”。然而，本人读起来却感觉该书远不止“科普”“入门”那么简单，书中的很多公式需要思考良久方能推导，很多概念需要反复咀嚼才能消化。边读边想着是不是应该将自己学习时遇到的一些知识难点的解析分享出来，以帮助更多的人入门。自己的确也随手做过一些笔记，但由于怀疑这仅是自己的个别现象，毕竟读书期间，思考更多的是如何使用单片机、DSP、ARM、FGPA 等，而这些基本是不需要推导任何公式的，因此作罢。偶然间在周老师的新浪微博看到如下对话：

此时方知，可能“读不懂”并不是个别现象。因此决定写一本“西瓜书注解”或者称为“西瓜书读书笔记”，对自己研读西瓜书时遇到的“台阶”进行解释和推导，以帮助更多的人能够更快地进入到这个领域。另外，近期越来越强地意识到，扎扎实实地推导一些基础算法的公式，无论是对于理解算法本身机理还是进行学术研究，都是非常有必要的。

自己会根据个人学习进度和研究需要按章发布，不知道能不能坚持写完，加油！

毕竟自己也是一名初学者，所以可能一些概念解释并不完整、一些公式推导并不优美，甚至会存在错误，这是不可避免的，不接受漫骂，但欢迎将问题反馈给我，共同学习进步！

（网盘链接：https://pan.baidu.com/s/1QtEiNnk8jMzmbs0KPBN-_w）

第 6 章目录

第 6 章 支持向量机

6.1 间隔与支持向量

1、式(6.2)的推导

首先说两点预备知识：

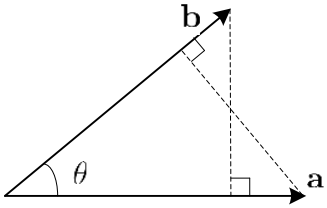
【预备 1】向量投影

两个向量的内积（点乘）：$\boldsymbol{a}\cdot\boldsymbol{b}=\boldsymbol{a}^{\top}\boldsymbol{b}=\|\boldsymbol{a}\|\,\|\boldsymbol{b}\|\cos(\theta)$.

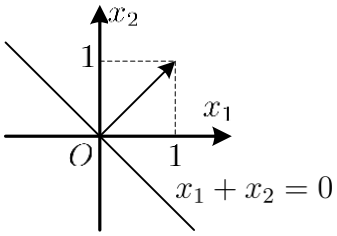
$\boldsymbol{a}$在$\boldsymbol{b}$上的投影长度为$\|\boldsymbol{a}\|\cos(\theta)=\frac{\boldsymbol{a}\cdot\boldsymbol{b}}{\|\boldsymbol{b}\|}$

$\boldsymbol{b}$在$\boldsymbol{a}$上的投影长度为$\|\boldsymbol{b}\|\cos(\theta)=\frac{\boldsymbol{a}\cdot\boldsymbol{b}}{\|\boldsymbol{a}\|}$

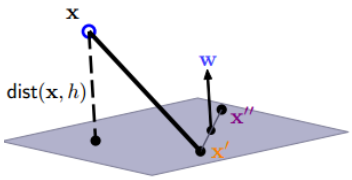
注意：以上计算投影长度的公式仅适用于当$\theta<\frac{\pi}{2}$时，若$\frac{\pi}{2}<\theta<\pi$，则$\cos(\theta)<0$，计算投影的公式应取绝对值 。



向量投影



超平面法向量



点到平面的距离

【预备 2】超平面法向量

超平面$\boldsymbol{w}^{\top}\boldsymbol{x}+b=0$的法向量为$\boldsymbol{w}$（当$\boldsymbol{w}$为二维向量时则是一条直线），例如：

当$\boldsymbol{w}=(1,1),b=0$时，超平面即为直线$x_1+x_2=0$。

注意：$x_1+x_2=0$和我们常见的$y=-x$表达意思相同；个人理解$x_1+x_2=0$更侧重于表达直线本身，而$y=-x$表达y随x线性变化。

有关超平面法向量，可以用更通用的证明：

任意取超平面上两点$\boldsymbol{x}',\boldsymbol{x}''$（即满足超平面方程$\boldsymbol{w}^{\top}\boldsymbol{x}'+b=0,\boldsymbol{w}^{\top}\boldsymbol{x}''+b=0$）

则两点的连线$\boldsymbol{x}'-\boldsymbol{x}''$一定在超平面上（这是常识吧）

又因为$\boldsymbol{w}^{\top}(\boldsymbol{x}'-\boldsymbol{x}'')=\boldsymbol{w}^{\top}\boldsymbol{x}'+b-(\boldsymbol{w}^{\top}\boldsymbol{x}''+b)=0$

即$\boldsymbol{w}\perp(\boldsymbol{x}'-\boldsymbol{x}'')$（向量内积等于零，说明二者垂直）

即$\boldsymbol{w}$垂直于超平面上任意向量$\boldsymbol{x}'-\boldsymbol{x}''$，也就是说$\boldsymbol{w}$是超平面法向量，得证！

以下正式开始推导式(6.2)：

已知超平面$\boldsymbol{w}^{\top}\boldsymbol{x}+b=0$，求空间中任意点$\boldsymbol{x}$到该超平面的距离。

【解】（参考林轩田老师《机器学习技法》课程的第 1 讲 PPT）

任意取超平面上一个点$\boldsymbol{x}'$，则点$\boldsymbol{x}$到超平面的距离等于向量$(\boldsymbol{x}-\boldsymbol{x}')$在法向量$\boldsymbol{w}$（参考预备 2）的投影长度（参考预备 1）：

$$\text{dist}=\frac{\left|\boldsymbol{w}^{\top}(\boldsymbol{x}-\boldsymbol{x}')\right|}{\|\boldsymbol{w}\|}=\frac{\left|\boldsymbol{w}^{\top}\boldsymbol{x}-\boldsymbol{w}^{\top}\boldsymbol{x}'\right|}{\|\boldsymbol{w}\|}=\frac{\left|\boldsymbol{w}^{\top}\boldsymbol{x}+b\right|}{\|\boldsymbol{w}\|}$$

注意：上式推导过程中，分子之所有取绝对值是由于向量内积可能小于零；另外，由于$\boldsymbol{x}'$是超平上面的点，因此$\boldsymbol{w}^\top \boldsymbol{x}' + b = 0$，即$b = -\boldsymbol{w}^\top \boldsymbol{x}'$。

2、式(6.3)的推导

由上一个公式的推导可知，对于超平面外任意点$\boldsymbol{x}$来说，$\boldsymbol{w}^\top \boldsymbol{x} + b$表示超平面法向量$\boldsymbol{w}$与$(\boldsymbol{x} - \boldsymbol{x}')$的内积，其中$\boldsymbol{x}'$为超平面上任一个点。

对于超平面两侧不同的点$\boldsymbol{x}$来说，内积$\boldsymbol{w}^\top \boldsymbol{x} + b$的正负号肯定是不同的，原因在于$\boldsymbol{w}$与$(\boldsymbol{x} - \boldsymbol{x}')$的夹角一定是其中一侧是$0 \leq \theta < \frac{\pi}{2}$，另一侧是$\frac{\pi}{2} < \theta \leq \pi$，到底哪一侧是$0 \leq \theta < \frac{\pi}{2}$并不重要，但只要确定了$\boldsymbol{w}$，夹角的范围也就随之确定了。

假设存在超平面能将训练样本正确分类，也就是说对于所有样本来说，两种不同类别的样本各自位于超平面的其中一侧。假设这个超平面是$(\boldsymbol{w}')^\top \boldsymbol{x} + b' = 0$，对于$(\boldsymbol{x}_i, y_i) \in D$

$$\begin{cases} (\boldsymbol{w}')^\top \boldsymbol{x}_i + b' > 0, & y_i = +1 \\ (\boldsymbol{w}')^\top \boldsymbol{x}_i + b' < 0, & y_i = -1 \end{cases}$$

注意：若正类和反类对应的符号与此假设相反，则只需对$\boldsymbol{w}', b'$添加负号即可。

将以上关系修正为

$$\begin{cases} (\boldsymbol{w}')^\top \boldsymbol{x}_i + b' \geq +\zeta, & y_i = +1 \\ (\boldsymbol{w}')^\top \boldsymbol{x}_i + b' \leq -\zeta, & y_i = -1 \end{cases}$$

其中ζ为某个大于零的常数，两边同除以ζ，再次修正以上关系为

$$\begin{cases} \left(\frac{1}{\zeta}\boldsymbol{w}'\right)^\top \boldsymbol{x}_i + \frac{b'}{\zeta} \geq +1, & y_i = +1 \\ \left(\frac{1}{\zeta}\boldsymbol{w}'\right)^\top \boldsymbol{x}_i + \frac{b'}{\zeta} \leq -1, & y_i = -1 \end{cases}$$

此时，再令$\boldsymbol{w} = \frac{1}{\zeta}\boldsymbol{w}', b = \frac{b'}{\zeta}$，则以上关系可写为

$$\begin{cases} \boldsymbol{w}^\top \boldsymbol{x}_i + b \geq +1, & y_i = +1 \\ \boldsymbol{w}^\top \boldsymbol{x}_i + b \leq -1, & y_i = -1 \end{cases}$$

此即式(6.3)，推导完毕。

3、式(6.4)的推导

注意到，距离超平面最近的训练样本$(\boldsymbol{x}_i, y_i)$可以使式(6.3)的等号成立，即

$$\begin{cases} \boldsymbol{w}^\top \boldsymbol{x}_i + b = +1, & y_i = +1 \\ \boldsymbol{w}^\top \boldsymbol{x}_i + b = -1, & y_i = -1 \end{cases}$$

根据式(6.2)可知，这些点（即候选支持向量）到超平面的距离为

$$\text{dist} = \frac{\left|\boldsymbol{w}^\top \boldsymbol{x} + b\right|}{\|\boldsymbol{w}\|} = \frac{1}{\|\boldsymbol{w}\|}$$

那么很容易知道，两个异类支持向量到超平面的距离之和是$\frac{2}{\|\boldsymbol{w}\|}$。

4、式(6.6)的解释

式(6.6)的约束条件意思是训练样本线性可分，也就是说不存在被分类错误的样本，因此也就不存在欠拟合问题；已知优化式(6.6)目标函数是在寻找“最大间隔”的划分超平面，而“最大间隔”划分超平面所产生的分类结果是最鲁棒的，对未见示例的泛化能力最强，因此可将式(6.6)优化目标进一步解释为**“最小化$\frac{1}{2}\|\boldsymbol{w}\|^2$则相当于寻找最不可能过拟合的分类超**

平面”。如果看过 Coursera 吴恩达的机器学习课程，想想这是谁在做的事情？对的，为了防止过拟合引入了**正则化**(Regularization)，而正则化就是在最小化的目标函数中加入分类器的所有参数的模值的平方（不含位移项b），与此不谋而合！

6.2 对偶问题

由于本节的推导核心是拉格朗日乘子法，因此先简单介绍拉格朗日乘子法，这里仅回答“怎么做”的问题，不回答“为什么这么做”的问题，书中附录 B.1 后半部分介绍了拉格朗日乘子法（从式(B.4)开始）。此处不考虑极值是否存在的问题，假设优化问题一定有解。

设约束优化问题为：（即原始问题 Primal Problem）

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}),\ \text{s.t. } \boldsymbol{h}(\boldsymbol{x})=0,\ \boldsymbol{g}(\boldsymbol{x})\leq 0$$

其中约束条件$\boldsymbol{h}(\boldsymbol{x})=0,\ \boldsymbol{g}(\boldsymbol{x})\leq 0$分别是由$m$个等式方程和$n$个不等式方程组成的方程组。

引入拉格朗日乘子（Lagrangian Multiplier）：

$$\boldsymbol{\lambda}=(\lambda_1,\lambda_2,\ldots,\lambda_m)\text{和}\boldsymbol{\mu}=(\mu_1,\mu_2,\ldots,\mu_n)$$

即可得优化问题的拉格朗日函数：

$$L(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu})=f(\boldsymbol{x})+\boldsymbol{\lambda}\boldsymbol{h}(\boldsymbol{x})+\boldsymbol{\mu}\boldsymbol{g}(\boldsymbol{x})$$

这里要求$\boldsymbol{\mu}\succeq 0$（所有分量非负），这与约束条件$\boldsymbol{g}(\boldsymbol{x})\leq 0$是对应的，即要求$\boldsymbol{\mu}\boldsymbol{g}(\boldsymbol{x})\leq 0$，也就是$L(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu})\leq f(\boldsymbol{x})$恒成立。若实际优化中不等式约束为$\boldsymbol{g}(\boldsymbol{x})\geq 0$，可将其转化为$-\boldsymbol{g}(\boldsymbol{x})\leq 0$；同理若优化目标为$\max_{\boldsymbol{x}} f(\boldsymbol{x})$，可将其转化为$\min_{\boldsymbol{x}}(-f(\boldsymbol{x}))$。

拉格朗日对偶问题（Lagrangian Dual Problem）为

$$\max_{\boldsymbol{\lambda},\boldsymbol{\mu}}\Gamma(\boldsymbol{\lambda},\boldsymbol{\mu}),\ \text{s.t. } \boldsymbol{\mu}\succeq 0$$

其中$\Gamma(\boldsymbol{\lambda},\boldsymbol{\mu})$为$L(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu})$的下确界(infimum)，即

$$\Gamma(\boldsymbol{\lambda},\boldsymbol{\mu})=\inf_{\boldsymbol{x}} L(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu})$$

下确界inf与最小值min有区别，暂时不去区分，认为$\Gamma(\boldsymbol{\lambda},\boldsymbol{\mu})=\min_{\boldsymbol{x}} L(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu})$即可。与下确界相对的概念是上确界(supremum)，有兴趣自行查询，不再赘述。

如何求$\Gamma(\boldsymbol{\lambda},\boldsymbol{\mu})$的表达式呢？令$\frac{\partial L(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu})}{\partial \boldsymbol{x}}=0$，将$\boldsymbol{x}$用$\boldsymbol{\lambda},\boldsymbol{\mu}$表示代入$L(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu})$即可。

总结一下以上介绍的几个步骤：

待求目标：$\min_{\boldsymbol{x}} f(\boldsymbol{x}),\ \text{s.t. } \boldsymbol{h}(\boldsymbol{x})=0,\ \boldsymbol{g}(\boldsymbol{x})\leq 0$

① 拉格朗日函数$L(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu})=f(\boldsymbol{x})+\boldsymbol{\lambda}\boldsymbol{h}(\boldsymbol{x})+\boldsymbol{\mu}\boldsymbol{g}(\boldsymbol{x})$

② 令$\frac{\partial L(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu})}{\partial \boldsymbol{x}}=0$，解得用$\boldsymbol{\lambda},\boldsymbol{\mu}$表示的$\boldsymbol{x}$

③ 将$\boldsymbol{x}$代入$L(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu})$得：$\Gamma(\boldsymbol{\lambda},\boldsymbol{\mu})=\min_{\boldsymbol{x}} L(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu})$

④ 对偶问题：$\max_{\boldsymbol{\lambda},\boldsymbol{\mu}}\Gamma(\boldsymbol{\lambda},\boldsymbol{\mu}),\ \text{s.t. } \boldsymbol{\mu}\succeq 0$

更多内容参见以下两个 PPT：

http://www.csc.kth.se/utbildning/kth/kurser/DD3364/Lectures/Duality.pdf

http://www.csc.kth.se/utbildning/kth/kurser/DD3364/Lectures/KKT.pdf

1、式(6.8)的推导

这就是应用了拉格朗日乘子法，只是注意一点，式(6.6)中的约束条件是大于等于 1，在此处应用拉格朗日乘子法时要将其转化为小于等于零的约束，即$1-y_i\left(\boldsymbol{w}^{\top}\boldsymbol{x}_i+b\right)\leq 0$，为了推导后面的两个求偏导的公式，此处先将式(6.8)整理为

$$
\begin{aligned}
L(\boldsymbol{w}, b, \boldsymbol{\alpha}) &= \frac{1}{2}\|\boldsymbol{w}\|^2 + \sum_{i=1}^{m} \alpha_i \left(1 - y_i\left(\boldsymbol{w}^\top \boldsymbol{x}_i + b\right)\right) \\
&= \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + \sum_{i=1}^{m} \alpha_i - \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{w}^\top \boldsymbol{x}_i - \sum_{i=1}^{m} \alpha_i y_i b
\end{aligned}
$$

2、式(6.9)的推导

式(6.8)对$\boldsymbol{w}$求偏导可得（与$\boldsymbol{w}$无关的项求偏导后均等于零）：

$$
\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\alpha})}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i
$$

令偏导等于零即可得式(6.9)：

$$
\boldsymbol{w} - \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i = 0 \Rightarrow \boldsymbol{w} = \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i
$$

3、式(6.10)的推导

式(6.8)对b求偏导可得（与b无关的项求偏导后均等于零）：

$$
\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\alpha})}{\partial b} = -\sum_{i=1}^{m} \alpha_i y_i
$$

令偏导等于零即可得式(6.10)：

$$
-\sum_{i=1}^{m} \alpha_i y_i = 0 \Rightarrow \sum_{i=1}^{m} \alpha_i y_i = 0
$$

4、式(6.11)的推导

式(6.11)即为式(6.6)的对偶问题，这里主要推导式(6.11)的目标函数，即将根据式(6.9)(6.10)将式(6.8)转化为式(6.11)目标函数的过程。将式(6.8)继续整理为

$$
\begin{aligned}
L(\boldsymbol{w}, b, \boldsymbol{\alpha}) &= \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + \sum_{i=1}^{m} \alpha_i - \sum_{i=1}^{m} \alpha_i y_i \boldsymbol{w}^\top \boldsymbol{x}_i - \sum_{i=1}^{m} \alpha_i y_i b \\
&= \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + \sum_{i=1}^{m} \alpha_i - \boldsymbol{w}^\top \left(\sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i\right) - b\left(\sum_{i=1}^{m} \alpha_i y_i\right)
\end{aligned}
$$

其中后两项的变形是因为与求和变量无关的项（即相对于求和变量来说是一个常量）可以拿到求和号外面（即提公因式），再结合式(6.9)(6.10)：

$$
\begin{aligned}
L(\boldsymbol{w}, b, \boldsymbol{\alpha}) &= \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} + \sum_{i=1}^{m} \alpha_i - \boldsymbol{w}^\top \boldsymbol{w} - 0 \\
&= \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w}
\end{aligned}
$$

将式(6.9)代入即可得式(6.11)：

$$
\begin{aligned}
L(\boldsymbol{w}, b, \boldsymbol{\alpha}) &= \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} \\
&= \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\left(\sum_{i=1}^{m} \alpha_i y_i \boldsymbol{x}_i\right)^\top \left(\sum_{j=1}^{m} \alpha_j y_j \boldsymbol{x}_j\right) \\
&= \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j
\end{aligned}
$$

注意，两个求和项相乘即将每一项均分别对应相乘一遍即可；因为有两个不同的求和，所以使用了i, j两个求和变量；式(6.11)有两个约束条件，第 1 个即为式(6.10)，第 2 个是拉格朗日乘子法要求不等式约束的拉格朗日乘子非负，以保证$L(\boldsymbol{w}, b, \boldsymbol{\alpha})$不大于原始问题目标函数。

梳理一下式(6.8)到式(6.11)四个公式：针对原始问题式(6.6)，式(6.8)是其对应的拉格朗日函数；通过对自变量$\boldsymbol{w}, b$求导并令导数等于零，得到式(6.9)和式(6.10)；将式(6.9)和式(6.10)代入式(6.8)得到以$\boldsymbol{\alpha} = (\alpha_1; \alpha_2; \ldots; \alpha_m)$为自变量的$\min_{\boldsymbol{w},b} L(\boldsymbol{w}, b, \boldsymbol{\alpha})$表达式，亦即对偶问题式(6.11)的目标函数。注意，式(6.6)中自变量为$\boldsymbol{w}, b$，而$y_i, \boldsymbol{x}_i$均为常量。

这里解释一下为什么要引入对偶的概念去求解目标函数。对比式(6.6)和式(6.11) 可以知道，二者的约束条件个数均为m（即训练集样本个数），但目标函数的变量个数不一样，式(6.6)目标函数的变量个数与$\boldsymbol{w}$的维度相同，而式(6.11)则仍为m，当$\boldsymbol{w}$的维度远远大于m时，优化式(6.11)将比优化式(6.6)简单的多；另外，通过对偶问题，可以引入核函数。

5、式(6.12)的推导

该式就是将式(6.9)直接代入即可。

注意此处的$\boldsymbol{x}$为待预测的示例（未见示例），$\boldsymbol{x}_i$为训练集中的示例；拉格朗日乘子$\boldsymbol{\alpha}$的求解见后文 SMO 算法；已知$\boldsymbol{\alpha}$后根据式(6.9)可得$\boldsymbol{w}$，偏移项b的求解参见式(6.17)和式(6.18)。

6、式(6.13)的推导

该式就是 KKT 条件，具体参见附录 B.1 或前面给出的两个链接内容。

第 1 个是要求不等式约束的拉格朗日乘子不小于 0；

第 2 个就是式(6.6)的约束条件，只是用$f(\boldsymbol{x}_i)$替换了$\boldsymbol{w}^\top \boldsymbol{x}_i + b$表达形式并进行了移项；

第 3 个表示前两个式子必有其一等于零，当$\alpha_i = 0$时表示其对应的不等式约束不起作用，也就是说极值点位于不等式约束区域内部，当$\alpha_i > 0$时表示极值点位于不等式约束区域边缘，即$y_i f(\boldsymbol{x}_i) - 1 = 0$。

为什么要求不等式约束的拉格朗日乘子不小于 0 呢？其一是为了保证式(6.8)拉格朗日函数$L(\boldsymbol{w}, b, \boldsymbol{\alpha})$不大于原始问题目标函数$\frac{1}{2}\|\boldsymbol{w}\|^2$；其二，当$\alpha_i > 0$时表示极值点位于不等式约束区域边缘，此时$\alpha_i > 0$保证了原始问题**目标函数的梯度方向**与**约束函数的梯度方向**相反（参见附录 B.1 的式(B.1)，当$\lambda > 0$时两个梯度正负号一定相反），梯度方向是使函数值增加最快的方向（梯度反方向则是使函数值下降最快的方向），当在约束区域边缘某点取得极值时，在该点**目标函数的梯度方向一定是朝向约束区域内部的**（梯度反方向使函数值下降，此时梯度反方向指向约束区域外部，自变量不能在外部取值，因此函数值不能再下降了，即取得极小值），而**约束函数的梯度方向一定是朝向约束区域外部的**（因为约束函数为小于等于零类型，约束区域外部即为使约束函数大于零的自变量区域，梯度方向肯定指向能使约束函数的函数值增加的方向，即外部）。

6.3 核函数

1、式(6.19)的解释

唯一注意的是，此时的$\boldsymbol{w}$的维度与$\phi(\boldsymbol{x})$维度相同，而不再与$\boldsymbol{x}$维度相同。

2、式(6.22)的解释

此即核函数的定义，即核函数可以分解成两个向量的内积。要想了解某个核函数是如何将原始特征空间映射到更高维的特征空间的，只需要$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$分解为两个表达形式完全一样的向量$\phi(\boldsymbol{x}_i)$和$\phi(\boldsymbol{x}_j)$即可（有时很难分解）。以下是 LIBSVM 中的几个核函数：

- 多项式核（Polynomial Kernel）：$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\gamma \boldsymbol{x}_i^\top \boldsymbol{x}_j + \zeta\right)^{\mathrm{d}}$
- 高斯核（Gaussian Kernel or Radial Basis Function Kernel）：$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\gamma\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}$

  高斯核可以将原始特征空间映射到无穷维，推导如下：（$\boldsymbol{x}$为列向量）

$$
\begin{aligned}
\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) &= e^{-\gamma\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2} \\
&= e^{-\gamma\|\boldsymbol{x}_i\|^2 - \gamma\|\boldsymbol{x}_j\|^2 + 2\gamma \boldsymbol{x}_i^\top \boldsymbol{x}_j} \\
&= e^{-\gamma\|\boldsymbol{x}_i\|^2} e^{-\gamma\|\boldsymbol{x}_j\|^2} e^{2\gamma \boldsymbol{x}_i^\top \boldsymbol{x}_j} \\
&= e^{-\gamma\|\boldsymbol{x}_i\|^2} e^{-\gamma\|\boldsymbol{x}_j\|^2} \left(\sum_{n=0}^{\infty} \frac{\left(2\gamma \boldsymbol{x}_i^\top \boldsymbol{x}_j\right)^n}{n!}\right) \\
&= \sum_{n=0}^{\infty} \left(e^{-\gamma\|\boldsymbol{x}_i\|^2} e^{-\gamma\|\boldsymbol{x}_j\|^2} \frac{\left(2\gamma \boldsymbol{x}_i^\top \boldsymbol{x}_j\right)^n}{n!}\right) \\
&= \sum_{n=0}^{\infty} \left(e^{-\gamma\|\boldsymbol{x}_i\|^2} e^{-\gamma\|\boldsymbol{x}_j\|^2} \sqrt{\frac{(2\gamma)^n}{n!}} \left(\boldsymbol{x}_i^\top\right)^n \sqrt{\frac{(2\gamma)^n}{n!}} \left(\boldsymbol{x}_j\right)^n\right) \\
&= \phi\left(\boldsymbol{x}_i\right)^\top \phi\left(\boldsymbol{x}_j\right)
\end{aligned}
$$

其中

$$
\phi\left(\boldsymbol{x}\right) = e^{-\gamma\|\boldsymbol{x}\|^2} \left(1, \sqrt{\frac{(2\gamma)^1}{1!}} \boldsymbol{x}^\top, \sqrt{\frac{(2\gamma)^2}{2!}} \left(\boldsymbol{x}^\top\right)^2, \sqrt{\frac{(2\gamma)^3}{3!}} \left(\boldsymbol{x}^\top\right)^3, \ldots\right)^\top
$$

推导过程中$(\boldsymbol{x})^n$表示将$\boldsymbol{x}$的每个元素均取n次方，即其结果仍为与$\boldsymbol{x}$同维度的向量；从第 3 行到第 4 行应用了泰勒展开公式$f(x) = e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots$。

这是网上流传较多的推导，包括林轩田在 Coursera 的机器学习技法课程第 3 讲中也是如此推导的，但细想这个推导结果实际是有问题的，因为我们认为经过推导得到了一个无穷维的向量，但$\phi\left(\boldsymbol{x}\right)$并不是一个向量，因为$\phi\left(\boldsymbol{x}\right)$的无穷维向量表达式中除 1 之外其余均是与$\boldsymbol{x}$同维度的向量！

是么？众人皆醉我独醒？当然不是！设$\boldsymbol{x} = (x_1, x_2, \ldots, x_d)^\top \in \mathbb{R}^{\mathrm{d}}$，经过函数$\phi\left(\cdot\right)$映射后$\phi(\boldsymbol{x}) = (x'_1, x'_2, \ldots, x'_{d'})^\top \in \mathbb{R}^{\mathrm{d'}}$，对应到推导结果：

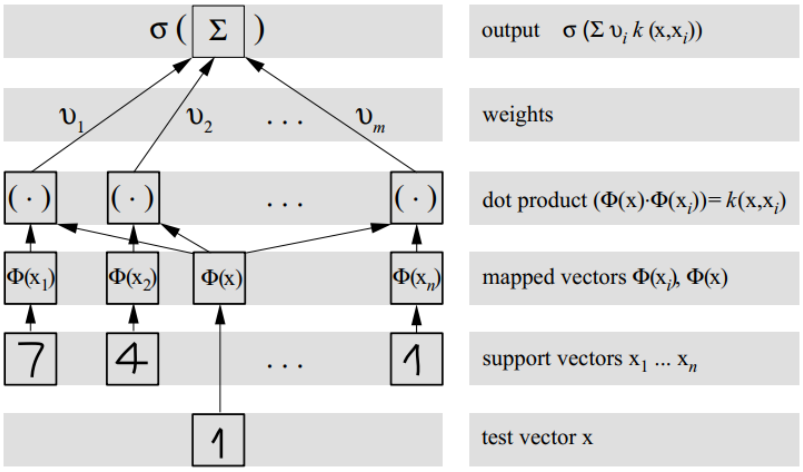
$$\phi(\boldsymbol{x}) = e^{-\gamma\|\boldsymbol{x}\|^2}\left(1, \sqrt{\frac{(2\gamma)^1}{1!}}x_1, \sqrt{\frac{(2\gamma)^1}{1!}}x_2, \ldots, \sqrt{\frac{(2\gamma)^1}{1!}}x_d, \right.$$
$$\sqrt{\frac{(2\gamma)^2}{2!}}x_1^2, \sqrt{\frac{(2\gamma)^2}{2!}}x_2^2, \ldots, \sqrt{\frac{(2\gamma)^2}{2!}}x_d^2,$$
$$\left.\sqrt{\frac{(2\gamma)^3}{3!}}x_1^3, \sqrt{\frac{(2\gamma)^3}{3!}}x_2^3, \ldots, \sqrt{\frac{(2\gamma)^3}{3!}}x_d^3, \ldots\ldots\right)^\top$$

即函数$\phi(\cdot)$将原始特征映射到了无穷维。

- Sigmoid 核：$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh\left(\gamma \boldsymbol{x}_i^\top \boldsymbol{x}_j + \zeta\right)$

3、式(6.24)的解释

由式(6.24)的最终结果可以看出，对于未见示例$\boldsymbol{x}$，支持向量机可以看作将原始特征$\boldsymbol{x}$先进行特征转换$\hat{\boldsymbol{x}} = [\kappa(\boldsymbol{x}, \boldsymbol{x}_1), \kappa(\boldsymbol{x}, \boldsymbol{x}_2), \ldots, \kappa(\boldsymbol{x}, \boldsymbol{x}_m)]^\top \in \mathbb{R}^m$，然后进行线性分类的结果，分类超平面为$f(\widetilde{\boldsymbol{x}}) = \widetilde{\boldsymbol{w}}^\top \widetilde{\boldsymbol{x}} + b$，其中$\widetilde{\boldsymbol{w}} = [\alpha_1 y_1, \alpha_2 y_2, \ldots, \alpha_m y_m]^\top$。通过 6.2 节的推导知道，对于非支持向量，其$\alpha_i = 0$，因此特征转换实际并不需要$m$维，而是等于支持向量的个数。下图摘自《Advances in Large Margin Classifiers》(Figure 1.4)：



值得注意的是，上图的 mapper vectors 和 dot product 两层实际由核函数同时完成。这种结构是不是有种似曾相似的感觉？对的，与第 108 页 5.5.1 节介绍的 RBF 网络形式基本一致，不同的是这里进行特征转换时使用的是支持向量，而 RBF 网络则须以某种方式确定神经元中心；另外，优化目标也有差异。

6.4 软间隔与正则化

本节推导基本与 6.2 节类似，相似之处不再重复推导。

1、式(6.35)的推导

其实我不明白的是为什么式(6.35)与式(6.34)等价？

(1)有一点很容易推导，那就是式(6.35)是式(6.34)的上限，最小化式(6.35)的同时也会最化小式(6.34)，这是因为：

由式(6.35)的约束条件$y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)\geqslant 1-\xi_i$可得$\xi_i\geqslant 1-y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)$，再加上约束条件$\xi_i\geqslant 0$，即$\xi_i\geqslant \max(0,1-y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b))$，因此式(6.35)是式(6.34)的上限。

(2)式(6.34)的上限有很多种，比如将式(6.34)中的第 2 部分换为式(6.32)或式(6.33)？为什么式(6.35)对应的是 hinge 损失呢？即为什么式(6.35)与式(6.34)等价？

式(6.35)的目标函数是最小化，即要求ξ_i越小越好：当$y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)\geqslant 1$成立时，$\xi_i=0$（这已经是$\xi_i$的最小值了）即可保证$y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)\geqslant 1-\xi_i$成立；而当$y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)\geqslant 1$不成立时，$\xi_i=1-y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)$即可保证$y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)\geqslant 1-\xi_i$成立；当然，对于这两种情况也有其它$\xi_i$值可以使$y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)\geqslant 1-\xi_i$成立，例如第一种情况中选$\xi_i>0$的任意值也可以使不等式成立，第二种情况中选$\xi_i>1-y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)$的任意值也可以使不等式成立，但回到最开始所说“式(6.35)的目标函数是最小化，即要求ξ_i越小越好”，所以ξ_i的取值为$\xi_i=\begin{cases}1-y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b), & y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)<1\\ 0, & y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)\geqslant 1\end{cases}=\max(0,1-y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b))$，得证。

2、式(6.40)的推导

其实按部就班地按 6.2 节推导也能得到式(6.40)的优化目标，这里提一种直观方法。

将式(6.36)分成两部分：与ξ_i有关的项和与ξ_i无关的项，即

$$\begin{aligned}L(\boldsymbol{w},b,\boldsymbol{\alpha},\boldsymbol{\xi},\boldsymbol{\mu})&=\frac{1}{2}\|\boldsymbol{w}\|^2+C\sum_{i=1}^{m}\xi_i+\sum_{i=1}^{m}\alpha_i\left(1-\xi_i-y_i\left(\boldsymbol{w}^\top\boldsymbol{x}_i+b\right)\right)-\sum_{i=1}^{m}\mu_i\xi_i\\&=\left(\frac{1}{2}\|\boldsymbol{w}\|^2+\sum_{i=1}^{m}\alpha_i\left(1-y_i\left(\boldsymbol{w}^\top\boldsymbol{x}_i+b\right)\right)\right)\\&+\left(C\sum_{i=1}^{m}\xi_i-\sum_{i=1}^{m}\alpha_i\xi_i-\sum_{i=1}^{m}\mu_i\xi_i\right)\\&=\left(\frac{1}{2}\|\boldsymbol{w}\|^2+\sum_{i=1}^{m}\alpha_i\left(1-y_i\left(\boldsymbol{w}^\top\boldsymbol{x}_i+b\right)\right)\right)+\sum_{i=1}^{m}(C-\alpha_i-\mu_i)\xi_i\end{aligned}$$

根据式(6.39)可知$C-\alpha_i-\mu_i=0$，式(6.36)仅剩

$$L(\boldsymbol{w},b,\boldsymbol{\alpha},\boldsymbol{\xi},\boldsymbol{\mu})=\frac{1}{2}\|\boldsymbol{w}\|^2+\sum_{i=1}^{m}\alpha_i\left(1-y_i\left(\boldsymbol{w}^\top\boldsymbol{x}_i+b\right)\right)$$

即与式(6.8)相同，而式(6.37)(6.38)分别与式(6.9)(6.10)相同，因此代入后所得式(6.40)目标函数与式(6.11)目标函数亦相同。至于约束条件，由$C=\alpha_i+\mu_i$和$\alpha_i>0,\mu_i>0$，很容易得到$0\leqslant\alpha_i\leqslant C$。

3、对率回归与支持向量机的关系

在第 132 页最后一段，讨论了对率回归与支持向量机的关系，提到“如果使用对率损失函数ℓ_{log}来替代式(6.29)中的$0/1$损失函数，则几乎就得到了对率回归模型(3.27)”，但式(6.29)与文中所提到的第三章对率回归模型式(3.27)形式上相差甚远。为了更清晰的说明对率回归与软间隔支持向量机的关系，以下先对式(3.27)的形式进行变化。

将$\boldsymbol{\beta}=(\boldsymbol{w};b)$和$\hat{\boldsymbol{x}}=(\boldsymbol{x};1)$代入式(3.27)：

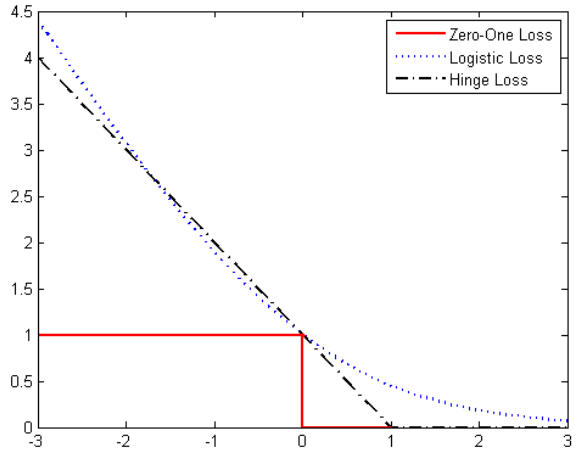
$$
\begin{aligned}
\ell(\boldsymbol{w}, b) &= \sum_{i=1}^{m} \left(-y_i \left(\boldsymbol{w}^\top \boldsymbol{x}_i + b \right) + \ln \left(1 + e^{\boldsymbol{w}^\top \boldsymbol{x}_i + b} \right) \right) \\
&= \sum_{i=1}^{m} \left(\ln \frac{1}{e^{y_i \left(\boldsymbol{w}^\top \boldsymbol{x}_i + b \right)}} + \ln \left(1 + e^{\boldsymbol{w}^\top \boldsymbol{x}_i + b} \right) \right) \\
&= \sum_{i=1}^{m} \ln \frac{1 + e^{\boldsymbol{w}^\top \boldsymbol{x}_i + b}}{e^{y_i \left(\boldsymbol{w}^\top \boldsymbol{x}_i + b \right)}} \\
&= \begin{cases} \sum_{i=1}^{m} \ln \left(1 + e^{-\left(\boldsymbol{w}^\top \boldsymbol{x}_i + b \right)} \right) & , y_i = 1 \\ \sum_{i=1}^{m} \ln \left(1 + e^{\boldsymbol{w}^\top \boldsymbol{x}_i + b} \right) & , y_i = 0 \end{cases}
\end{aligned}
$$

上式中，正例和反例分别用$y = 1$和$y = 0$分别表示，这是对率回归中的惯例；而在支持向量机中正例和反例习惯用$y = +1$和$y = -1$表示。实际上，若用$y = +1$和$y = -1$分别表示正例和反例，上式可以表示如下：

$$
\begin{aligned}
\ell(\boldsymbol{w}, b) &= \sum_{i=1}^{m} \ln \left(1 + e^{-y_i \left(\boldsymbol{w}^\top \boldsymbol{x}_i + b \right)} \right) \\
&= \begin{cases} \sum_{i=1}^{m} \ln \left(1 + e^{-\left(\boldsymbol{w}^\top \boldsymbol{x}_i + b \right)} \right) & , y_i = +1 \\ \sum_{i=1}^{m} \ln \left(1 + e^{\boldsymbol{w}^\top \boldsymbol{x}_i + b} \right) & , y_i = -1 \end{cases}
\end{aligned}
$$

即以上两个公式等价，而上式的求和项正是式(6.33)表述的对率损失。

书中图 6.5 中的$0/1$损失、hinge 损失、对率损失如下图所示：



为了保正对率损失函数通过固定点$(0, 1)$，实际画的是以 2 为底的对数，根据换底公式，这与以自然常数 e 为底的对数只相差一个固定的倍数，而这在优化目标中并无实质影响：

$$
\ell_{log_2}(z) = \log_2(1 + \exp(-z)) = \tfrac{1}{\ln 2} \ln(1 + \exp(-z))
$$

现在使用对率损失函数ℓ_{log}来替代式(6.29)中的$0/1$损失函数

$$
\min_{\boldsymbol{w}, b} \frac{1}{2} \|\boldsymbol{w}\|^2 + C \sum_{i=1}^{m} \log \left(1 + e^{-y_i \left(\boldsymbol{w}^\top \boldsymbol{x}_i + b \right)} \right)
$$

上式实际上就是带有正则化(regularization)的对率回归模型，按当前符号换为 Andrew Ng 在 Coursera 上的机器学习课程中的对率回归优化目标表达方式

$$\min_{\boldsymbol{w},b} \frac{1}{m}\sum_{i=1}^{m}\log\left(1+e^{-y_i\left(\boldsymbol{w}^{\top}\boldsymbol{x}_i+b\right)}\right)+\frac{\lambda}{2m}\|\boldsymbol{w}\|^2$$

通过以上推导可以得出结论：对率回归与软间隔支持向量机本质上相同，区别仅在于代价函数不同。对率回归从最大似然概率（或最小化分类代价）角度入手，为了防止过拟合在目标函数中引入正则化项$\|\boldsymbol{w}\|^2$；而支持向量机则从最大间隔角度入手，为了允许部分样本分类出错在目标函数中引入损失函数，**若将损失函数取为对率损失，则软间隔支持向量机即为对率回归**，但软间隔支持向量机却巧妙地使用了 Hinge 损失，进而保留了硬间隔支持向量机中支持向量的概念。在式(6.6)的解释中已经提到最大化间隔相当于寻找最不可能过拟合的分类超平面，这就是正则化的目标。

在课本第 133 页 6.4 节的最后一段专门进行了讨论，最大化间隔或正则化所对应的即为式(6.42)中的结构风险$\Omega(f)$，表述了我们希望获得具有何种性质的模型，削减了假设空间，降低了一味地最小化训练误差（代价函数）的过拟合风险。

西瓜书在“对目标函数引入正则化项降低过拟合风险”这一点的处理上可谓高屋建瓴，并未在第三章有任何提及，而是在软间隔支持向量机结束时进行讲述，这与 Coursera 上无论是 Andrew Ng 的机器学习课程还是林轩田的课程都有较大区别，因为二者在讲课时都是将正则化作为一项抑制过拟合的技术单独讲述的。

另外，在第 132 页最后一段还特别提到“支持向量机与对率回归的优化目标相近，通常情形下它们的性能也相当”，由于最早学习了 Coursera 上 Andrew Ng 的机器学习课程，但对率回归是课程的第一个分类模型，支持向量机课程的是最后一个分类模型，所以第一印象是对率回归是一个简单的、初级的模型，而支持向量机是一个复杂的、高级的模型，这一误解到此应该纠正过来。

学过 Andrew Ng 在 Coursera 上的 Machine Learning 课程的人现在是不是可以理解课程对 SVM 的讲解方式，实际上是将对率回归的损失改为 hinge 损失引入 SVM 的，而且 Andrew Ng 对 large margin 的概念进行了较为清晰的解释；另外，Andrew Ng 对核函数的讲解与一般书目也不同，是从相似函数的角度去讲的，其实就是式(6.24)，难道式(6.24)不是对核函数的回归么？唯一的遗憾是没有对支持向量的概念讲解清楚，但这个概念可能并不太好讲明白。当对 SVM 有一定的理解后再去看 Andrew Ng 的讲解会发现角度很独特。

有关支持向量机的输出如何转化为概率输出参见书中给出的参考文献。

对率回归应用于多分类任务即著名的 softmax 回归（对率回归是 softmax 回归的特殊形式），而支持向量机处理多分类任务时有专门的推广，但据说算法效率还不如采用直接采用 one vs one 方式进行多次二分类，因此无论是 LIBSVM 还是 WEKA 均使用基本的二分类支持向量机结合 one vs one 方式处理多分类任务。

4、有关支持向量意义的疑问

上面的解释中提到，若将损失函数取为对率损失，则软间隔支持向量机即为对率回归，但软间隔支持向量机却巧妙地使用了 Hinge 损失，进而保留了硬间隔支持向量机中支持向量的概念。那么引出支持向量的概念究竟有什么好处呢？

课本第 133 页第 2 行提到“对率回归的解依赖于更多训练样本，其预测开销更大”，对此我一直有疑问：对率回归和支持向量机均是求解分类超平面$\boldsymbol{w}^{\top}\boldsymbol{x}+b=0$，训练的过程即从训练集中找出$\boldsymbol{w},b$，然后对于未见示例$\boldsymbol{x}$只需代入$f(\boldsymbol{x})=\boldsymbol{w}^{\top}\boldsymbol{x}+b$，若$f(\boldsymbol{x})>0$则判为正例，反之判为反例，这个过程与支持向量并没有任何关系。另外，之所以 SVM 有支持向量的念，是因为它人为地将预测正确的部分样本代价置零（即 hinge 损失的零区域），

有人说去掉非支持向量样本并不影响支持向量机的分类超平面，而实际上非支持向量样本即使对应到对率损失其代价也是很小的，去掉它们其实对对率回归的影响也很小（即这些样本即使在对率损失情景下对代价函数影响也很小，除非有蝴蝶效应），而且最关键的是支持向量是在训练过程结束后才确定的，而预测过程只需$\boldsymbol{w}, b$，支持向量的意义何在？

目前个人只能从核函数的角度去解释，如式(6.24)使用核函数后的最终预测模型，这时如果核函数为高斯核，$\boldsymbol{w}$根本不能被显式地计算出来，因为此时$\boldsymbol{w}$为无穷维（详见式(6.22)的解释），只能通过(6.24)这种形式进行计算，而此时实际仅有支持向量对应的α_i非零，因此计算$\boldsymbol{w}$时不需要遍历整个数据集。对率回归也有其核化版（详见 6.6 节的第 3 个解释），但由于没有支持向量的概念，因此进行预测时，计算$\boldsymbol{w}$需要遍历整个数据集。

还有，支持向量影响最终 SVM 模型的 VC 维，进而影响泛化误差上界……

更多支持向量的意义仍有待进一步理解……

6.5 支持向量回归

【SVR 与 SVM 的区别】

首先，SVR 是用来做回归任务的，SVM 是用来做分类任务的；回归任务是希望有一条直线（或超平面）使更多的点落在这条直线上，而分类任务是希望有一条直线（或超平面）将不同类别的点一分为二。

比较式(6.43)与式(6.34)可以知道，二者的主要区别就在于代价函数(Cost Function)，这也是回归任务和分类任务的本质决定的，SVR 的代价函数是希望更多的点落在ϵ间隔带，SVM 的代价函数是希望更多的点满足$y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)>1$，因为此时代价函数为零，这也是 SVR 区别于线性回归、SVM 区别于对率回归的最本质之处，正是因为代价函数有零值区域才导致了“支持向量”的概念（支持向量就是使代价函数非零的点），而线性回归和对率回归的代价函数均为连续的非零函数，故不能导出支持向量的概念（或者说所有训练样本都是支持向量）。

1、式(6.45)的解释

类似于式(6.35)的解释，对于任意$\boldsymbol{x}_i$，$f(\boldsymbol{x}_i)-y_i \leqslant \epsilon+\xi_i$和$y_i-f(\boldsymbol{x}_i) \leqslant \epsilon+\hat{\xi}_i$两个约束条件一定有其一成立，因为$f(\boldsymbol{x}_i)-y_i$和$y_i-f(\boldsymbol{x}_i)$一定有其一小于等于零，因此两个松弛变量$\xi_i$和$\hat{\xi}_i$之中定有一个等于零。

2、式(6.46)的解释

由于式(6.45)是最小化目标函数，因此引入拉格朗日乘子时要将所有不等式约束转化为小于等于零的形式，所以针对ξ_i和$\hat{\xi}_i$的两项前面的符号是负号。

3、式(6.47)的推导

$$\frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\alpha}, \hat{\boldsymbol{\alpha}}, \boldsymbol{\xi}, \hat{\boldsymbol{\xi}}, \boldsymbol{\mu}, \hat{\boldsymbol{\mu}})}{\partial \boldsymbol{w}}=\boldsymbol{w}+\sum_{i=1}^{m} \alpha_i \boldsymbol{x}_i-\sum_{i=1}^{m} \hat{\alpha}_i \boldsymbol{x}_i$$

令偏导等于零即可得式(6.47)。

同理可得式(6.48)(6.49)(6.50)，注意求偏导时，所有不含求导变量的变量统一舍去。

4、式(6.51)的推导

首先将式(6.46)所有与ξ_i和$\hat{\xi}_i$有关项进行整合（第 2~4 项及第 5~6 项中的一部分）：

$$C\sum_{i=1}^{m}(\xi_i+\hat{\xi}_i)-\sum_{i=1}^{m}\mu_i\xi_i-\sum_{i=1}^{m}\hat{\mu}_i\hat{\xi}_i-\sum_{i=1}^{m}\alpha_i\xi_i-\sum_{i=1}^{m}\hat{\alpha}_i\hat{\xi}_i$$

$$=\mathrm{C}\sum_{i=1}^{m}(\xi_i+\hat{\xi}_i)-\sum_{i=1}^{m}(\mu_i+\alpha_i)\xi_i-\sum_{i=1}^{m}(\hat{\mu}_i+\hat{\alpha}_i)\hat{\xi}_i$$

将式(6.49)和式(6.50)代入得：

$$=C\sum_{i=1}^{m}(\xi_i+\hat{\xi}_i)-\sum_{i=1}^{m}C\xi_i-\sum_{i=1}^{m}C\hat{\xi}_i$$

$$=C\sum_{i=1}^{m}(\xi_i+\hat{\xi}_i)-C\sum_{i=1}^{m}\xi_i-C\sum_{i=1}^{m}\hat{\xi}_i=0$$

然后将式(6.46)所有与ϵ有关项进行整合（第 5~6 项中的一部分）：

$$-\sum_{i=1}^{m}\alpha_i\epsilon-\sum_{i=1}^{m}\hat{\alpha}_i\epsilon=-\sum_{i=1}^{m}\epsilon(\hat{\alpha}_i+\alpha_i)$$

接下来将式(6.46)所有与y_i有关项进行整合（第 5~6 项中的一部分）：

$$-\sum_{i=1}^{m}\alpha_i y_i+\sum_{i=1}^{m}\hat{\alpha}_i y_i=\sum_{i=1}^{m}y_i(\hat{\alpha}_i-\alpha_i)$$

到此为止，式(6.46)未处理的三项分别是$\frac{1}{2}\|\boldsymbol{w}\|^2$、$\sum_{i=1}^{m}\alpha_i f(\boldsymbol{x}_i)$和$-\sum_{i=1}^{m}\hat{\alpha}_i f(\boldsymbol{x}_i)$，由式(6.7)的$f(\boldsymbol{x}_i)=\boldsymbol{w}^T\boldsymbol{x}_i+b$代入：

$$\sum_{i=1}^{m}\alpha_i f(\boldsymbol{x}_i)-\sum_{i=1}^{m}\hat{\alpha}_i f(\boldsymbol{x}_i)=\sum_{i=1}^{m}(\alpha_i-\hat{\alpha}_i)(\boldsymbol{w}^T\boldsymbol{x}_i+b)$$

$$=\sum_{i=1}^{m}(\alpha_i-\hat{\alpha}_i)(\boldsymbol{w}^T\boldsymbol{x}_i)+\sum_{i=1}^{m}(\alpha_i-\hat{\alpha}_i)b$$

$$=-\boldsymbol{w}^T\sum_{i=1}^{m}(\hat{\alpha}_i-\alpha_i)\boldsymbol{x}_i-b\sum_{i=1}^{m}(\hat{\alpha}_i-\alpha_i)$$

根据式(6.48)上式后一项等于 0。将式(6.47)代入上式第一项：

$$=-\boldsymbol{w}^T\boldsymbol{w}$$

综合以上结果：

$$L(\boldsymbol{w},b,\boldsymbol{\alpha},\hat{\boldsymbol{\alpha}},\boldsymbol{\xi},\hat{\boldsymbol{\xi}},\boldsymbol{\mu},\hat{\boldsymbol{\mu}})=\frac{1}{2}\|\boldsymbol{w}\|^2-\boldsymbol{w}^T\boldsymbol{w}+\sum_{i=1}^{m}y_i(\hat{\alpha}_i-\alpha_i)-\sum_{i=1}^{m}\epsilon(\hat{\alpha}_i+\alpha_i)$$

$$=\sum_{i=1}^{m}(y_i(\hat{\alpha}_i-\alpha_i)-\epsilon(\hat{\alpha}_i+\alpha_i))-\frac{1}{2}\|\boldsymbol{w}\|^2$$

将式(6.47)代入$\frac{1}{2}\|\boldsymbol{w}\|^2$

$$\frac{1}{2}\|\boldsymbol{w}\|^2=\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}=\frac{1}{2}(\sum_{i=1}^{m}(\hat{\alpha}_i-\alpha_i)\boldsymbol{x}_i)^T(\sum_{i=1}^{m}(\hat{\alpha}_i-\alpha_i)\boldsymbol{x}_i)$$

$$=\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\hat{\alpha}_i-\alpha_i)(\hat{\alpha}_j-\alpha_j)\boldsymbol{x}_i^T\boldsymbol{x}_j$$

代入前面$L(\boldsymbol{w},b,\boldsymbol{\alpha},\hat{\boldsymbol{\alpha}},\boldsymbol{\xi},\hat{\boldsymbol{\xi}},\boldsymbol{\mu},\hat{\boldsymbol{\mu}})$推导结果即得式(6.51)。

由于$\alpha_i\geqslant 0,\hat{\alpha}_i\geqslant 0,\mu_i\geqslant 0,\hat{\mu}_i\geqslant 0$，结合式(6.49)(6.50)即可得约束$0\leqslant\alpha_i,\hat{\alpha}_i\leqslant C$ 。

5、式(6.52)的解释

完整的 KKT 条件是（每三个一组，对应某一类不等式约束）：

$$f(\boldsymbol{x}_i)-y_i-\epsilon-\xi_i\leqslant 0$$

$$\alpha_i\geqslant 0$$

$$\alpha_i(f(\boldsymbol{x}_i)-y_i-\epsilon-\xi_i)=0$$（**式(6.52)第 1 行式子**）

$y_i - f(\boldsymbol{x}_i) - \epsilon - \hat{\xi}_i \leqslant 0$

$\hat{\alpha}_i \geqslant 0$

$\hat{\alpha}_i(y_i - f(\boldsymbol{x}_i) - \epsilon - \hat{\xi}_i) = 0$（**式(6.52)第 2 行式子**）

$\xi_i \geqslant 0$
$\mu_i \geqslant 0$
$\mu_i \xi_i = 0$（由式(6.49)替换μ_i，**式(6.52)第 4 行左侧式子**）

$\hat{\xi}_i \geqslant 0$

$\hat{\mu}_i \geqslant 0$

$\hat{\mu}_i \hat{\xi}_i = 0$（由式(6.50)替换$\hat{\mu}_i$，**式(6.52)第 4 行右侧式子**）

如式(6.45)的解释，两个松弛变量ξ_i和$\hat{\xi}_i$定有其一等于零，即**式(6.52)第 3 行右侧式子**；

如式(6.52)下面一段话的叙述，仅当样本$(\boldsymbol{x}_i, y_i)$不落入ϵ间隔带中，相应的α_i和$\hat{\alpha}_i$才能取非零值，而$f(\boldsymbol{x}_i) - y_i - \epsilon - \xi_i = 0$和$y_i - f(\boldsymbol{x}_i) - \epsilon - \hat{\xi}_i = 0$分别代表样本$(\boldsymbol{x}_i, y_i)$落在间隔带外的某一侧，但这不可能同时发生（如图 6.6 所示，一个样本不可能同时位于间隔带外的左上方和右下方），因此α_i和$\hat{\alpha}_i$中至少有一个为零（如果样本就在间隔带内部则二者同时为零），即**式(6.52)第 3 行左侧式子。**

6、式(6.53)的解释

如式(6.52)的解释所述，α_i和$\hat{\alpha}_i$中至少有一个为零，若二者都为零，即样本在间隔带内部，则不出现在式(6.53)中；若二者有一个不为零（且最多有一个不为零），即样本在间隔带外部，则出现在式(6.53)中，即为 SVR 的支持向量。即“显然，SVR 的支持向量仅是训练样本的一部分，即其解仍具有稀疏性”，而之所以其解具有稀疏性就是因为 SVR 的代价函数（如图 6.7 红线）有一段零值区域，而不像线性回归一样代价函数是平方损失（如图 6.7 黑线）。

7、式(6.54)的解释

若$0 < \alpha_i < C$，则$\mu_i = C - \alpha_i \neq 0$，由 KKT 条件$\mu_i \xi_i = 0$则必有$\xi_i = 0$；再由 KKT 条件$\alpha_i(f(\boldsymbol{x}_i) - y_i - \epsilon - \xi_i) = 0$，由于$\alpha_i \neq 0$则必有$f(\boldsymbol{x}_i) - y_i - \epsilon - \xi_i = 0$；考虑刚才得到的$\xi_i = 0$再加之$f(\boldsymbol{x}_i) = \boldsymbol{w}^T \boldsymbol{x}_i + b$，即$\boldsymbol{w}^T \boldsymbol{x}_i + b - y_i - \epsilon = 0$，故$b = y_i + \epsilon + \boldsymbol{w}^T \boldsymbol{x}_i$，将式(6.47)代入替换$\boldsymbol{w}$，即得式(6.54)。

6.6 核方法

1、定理 6.2(表示定理)的说明

式(6.24)是式(6.20)的解；式(6.56)是式(6.43)的解。对应到表示定理式(6.57)当中，式(6.20)和式(6.43)均为$\Omega\left(\|h\|_{\mathbb{H}}\right) = \frac{1}{2}\|\boldsymbol{w}\|^2$；式(6.20)的$\ell\left(h(\boldsymbol{x}_1), h(\boldsymbol{x}_2), \ldots, h(\boldsymbol{x}_m)\right) = 0$，而式(6.43)的$\ell\left(h(\boldsymbol{x}_1), h(\boldsymbol{x}_2), \ldots, h(\boldsymbol{x}_m)\right) = C\sum_{i=1}^{m} \ell_\epsilon\left(f(\boldsymbol{x}_i) - y_i\right)$；即均满足式(6.57)的要求，式(6.20)和式(6.43)的解$f(\boldsymbol{x})$均为$\kappa(\boldsymbol{x}, \boldsymbol{x}_i)$的线性组合，即式(6.58)。

2、式(6.70)的推导

式(6.70)实际就是用式(6.65)的$\boldsymbol{w}$替换式(6.60)，化简而得。

(1)先化简分子部分，将式(6.61)代入式(6.62)，再将式(6.62)和式(6.65)代入分子：

$$\begin{aligned}\boldsymbol{w}^\top \boldsymbol{S}_b^\phi \boldsymbol{w} = &\left(\sum_{i=1}^{m}\alpha_i\phi(\boldsymbol{x}_i)\right)^\top \cdot \left(\frac{1}{m_1}\sum_{\boldsymbol{x}\in X_1}\phi(\boldsymbol{x}) - \frac{1}{m_0}\sum_{\boldsymbol{x}\in X_0}\phi(\boldsymbol{x})\right) \\ &\cdot \left(\frac{1}{m_1}\sum_{\boldsymbol{x}\in X_1}\phi(\boldsymbol{x}) - \frac{1}{m_0}\sum_{\boldsymbol{x}\in X_0}\phi(\boldsymbol{x})\right)^\top \cdot \left(\sum_{j=1}^{m}\alpha_j\phi(\boldsymbol{x}_j)\right)\end{aligned}$$

涉及到矩阵的复杂公式推导，多留心各项的维度。例如，设$\phi(\boldsymbol{x}) \in \mathbb{R}^{\widetilde{d}\times 1}$，则

$$\left(\sum_{i=1}^{m}\alpha_i\phi(\boldsymbol{x}_i)\right)^\top \in \mathbb{R}^{1\times\widetilde{d}}$$

$$\left(\frac{1}{m_1}\sum_{\boldsymbol{x}\in X_1}\phi(\boldsymbol{x}) - \frac{1}{m_0}\sum_{\boldsymbol{x}\in X_0}\phi(\boldsymbol{x})\right) \in \mathbb{R}^{\widetilde{d}\times 1}$$

$$\left(\frac{1}{m_1}\sum_{\boldsymbol{x}\in X_1}\phi(\boldsymbol{x}) - \frac{1}{m_0}\sum_{\boldsymbol{x}\in X_0}\phi(\boldsymbol{x})\right)^\top \in \mathbb{R}^{1\times\widetilde{d}}$$

$$\left(\sum_{j=1}^{m}\alpha_j\phi(\boldsymbol{x}_j)\right) \in \mathbb{R}^{\widetilde{d}\times 1}$$

先计算前两项矩阵乘积

$$\begin{aligned}&\left(\sum_{i=1}^{m}\alpha_i\phi(\boldsymbol{x}_i)\right)^\top \cdot \left(\frac{1}{m_1}\sum_{\boldsymbol{x}\in X_1}\phi(\boldsymbol{x}) - \frac{1}{m_0}\sum_{\boldsymbol{x}\in X_0}\phi(\boldsymbol{x})\right) \\ =&\sum_{i=1}^{m}\left(\alpha_i\phi(\boldsymbol{x}_i)^\top \cdot \left(\frac{1}{m_1}\sum_{\boldsymbol{x}\in X_1}\phi(\boldsymbol{x}) - \frac{1}{m_0}\sum_{\boldsymbol{x}\in X_0}\phi(\boldsymbol{x})\right)\right) \\ =&\sum_{i=1}^{m}\alpha_i\left(\frac{1}{m_1}\sum_{\boldsymbol{x}\in X_1}\phi(\boldsymbol{x}_i)^\top\phi(\boldsymbol{x}) - \frac{1}{m_0}\sum_{\boldsymbol{x}\in X_0}\phi(\boldsymbol{x}_i)^\top\phi(\boldsymbol{x})\right) \\ =&\sum_{i=1}^{m}\alpha_i\left(\frac{1}{m_1}\sum_{\boldsymbol{x}\in X_1}\kappa(\boldsymbol{x}_i,\boldsymbol{x}) - \frac{1}{m_0}\sum_{\boldsymbol{x}\in X_0}\kappa(\boldsymbol{x}_i,\boldsymbol{x})\right)\end{aligned}$$

令$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)^\top$，结合式(6.66)和式(6.67)，并且$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \kappa(\boldsymbol{x}_j, \boldsymbol{x}_i)$（即核矩阵$\mathbf{K}$为对称阵），则（详见推导结束后的补充说明）

$$\sum_{i=1}^{m}\alpha_i\left(\frac{1}{m_1}\sum_{\boldsymbol{x}\in X_1}\kappa(\boldsymbol{x}_i,\boldsymbol{x})\right) = \boldsymbol{\alpha}^\top\hat{\boldsymbol{\mu}}_1$$

$$\sum_{i=1}^{m}\alpha_i\left(\frac{1}{m_0}\sum_{\boldsymbol{x}\in X_0}\kappa(\boldsymbol{x}_i,\boldsymbol{x})\right) = \boldsymbol{\alpha}^\top\hat{\boldsymbol{\mu}}_0$$

同理，计算后两项矩阵乘积

$$
\begin{aligned}
&\left(\frac{1}{m_1}\sum_{\boldsymbol{x}\in X_1}\phi(\boldsymbol{x})-\frac{1}{m_0}\sum_{\boldsymbol{x}\in X_0}\phi(\boldsymbol{x})\right)^{\top}\cdot\left(\sum_{j=1}^{m}\alpha_j\phi(\boldsymbol{x}_j)\right)\\
&=\sum_{j=1}^{m}\left(\frac{1}{m_1}\sum_{\boldsymbol{x}\in X_1}\kappa(\boldsymbol{x},\boldsymbol{x}_j)-\frac{1}{m_0}\sum_{\boldsymbol{x}\in X_0}\kappa(\boldsymbol{x},\boldsymbol{x}_j)\right)\alpha_j\\
&=(\hat{\boldsymbol{\mu}}_1-\hat{\boldsymbol{\mu}}_0)^{\top}\boldsymbol{\alpha}
\end{aligned}
$$

综上并结合式(6.68)，分子项$\boldsymbol{w}^{\top}\boldsymbol{S}_b^{\phi}\boldsymbol{w}=\boldsymbol{\alpha}^{\top}(\hat{\boldsymbol{\mu}}_1-\hat{\boldsymbol{\mu}}_0)(\hat{\boldsymbol{\mu}}_1-\hat{\boldsymbol{\mu}}_0)^{\top}\boldsymbol{\alpha}=\boldsymbol{\alpha}^{\top}\mathbf{M}\boldsymbol{\alpha}$ 。

(2)再化简分母部分

为了方便推导，先拆解式(6.63)的$\boldsymbol{S}_w^{\phi}$：

$$
\begin{aligned}
\boldsymbol{S}_w^{\phi}&=\sum_{i=0}^{1}\sum_{\boldsymbol{x}\in X_i}\left(\phi(\boldsymbol{x})\phi(\boldsymbol{x})^{\top}-\phi(\boldsymbol{x})\left(\boldsymbol{\mu}_i^{\phi}\right)^{\top}-\boldsymbol{\mu}_i^{\phi}\phi(\boldsymbol{x})^{\top}+\boldsymbol{\mu}_i^{\phi}\left(\boldsymbol{\mu}_i^{\phi}\right)^{\top}\right)\\
&=\sum_{i=0}^{1}\sum_{\boldsymbol{x}\in X_i}\phi(\boldsymbol{x})\phi(\boldsymbol{x})^{\top}-\sum_{i=0}^{1}\sum_{\boldsymbol{x}\in X_i}\phi(\boldsymbol{x})\left(\boldsymbol{\mu}_i^{\phi}\right)^{\top}\\
&\quad-\sum_{i=0}^{1}\sum_{\boldsymbol{x}\in X_i}\boldsymbol{\mu}_i^{\phi}\phi(\boldsymbol{x})^{\top}+\sum_{i=0}^{1}\sum_{\boldsymbol{x}\in X_i}\boldsymbol{\mu}_i^{\phi}\left(\boldsymbol{\mu}_i^{\phi}\right)^{\top}
\end{aligned}
$$

第一项：

$$
\begin{aligned}
\sum_{i=0}^{1}\sum_{\boldsymbol{x}\in X_i}\phi(\boldsymbol{x})\phi(\boldsymbol{x})^{\top}&=\sum_{\boldsymbol{x}\in X_0}\phi(\boldsymbol{x})\phi(\boldsymbol{x})^{\top}+\sum_{\boldsymbol{x}\in X_1}\phi(\boldsymbol{x})\phi(\boldsymbol{x})^{\top}\\
&=\sum_{\boldsymbol{x}\in X}\phi(\boldsymbol{x})\phi(\boldsymbol{x})^{\top}
\end{aligned}
$$

第二项：

$$
\begin{aligned}
\sum_{i=0}^{1}\sum_{\boldsymbol{x}\in X_i}\phi(\boldsymbol{x})\left(\boldsymbol{\mu}_i^{\phi}\right)^{\top}&=\sum_{\boldsymbol{x}\in X_0}\phi(\boldsymbol{x})\left(\boldsymbol{\mu}_0^{\phi}\right)^{\top}+\sum_{\boldsymbol{x}\in X_1}\phi(\boldsymbol{x})\left(\boldsymbol{\mu}_1^{\phi}\right)^{\top}\\
&=m_0\boldsymbol{\mu}_0^{\phi}\left(\boldsymbol{\mu}_0^{\phi}\right)^{\top}+m_1\boldsymbol{\mu}_1^{\phi}\left(\boldsymbol{\mu}_1^{\phi}\right)^{\top}
\end{aligned}
$$

第三项：

$$
\begin{aligned}
\sum_{i=0}^{1}\sum_{\boldsymbol{x}\in X_i}\boldsymbol{\mu}_i^{\phi}\phi(\boldsymbol{x})^{\top}&=\sum_{i=0}^{1}\boldsymbol{\mu}_i^{\phi}\sum_{\boldsymbol{x}\in X_i}\phi(\boldsymbol{x})^{\top}\\
&=\boldsymbol{\mu}_0^{\phi}\sum_{\boldsymbol{x}\in X_0}\phi(\boldsymbol{x})^{\top}+\boldsymbol{\mu}_1^{\phi}\sum_{\boldsymbol{x}\in X_1}\phi(\boldsymbol{x})^{\top}\\
&=m_0\boldsymbol{\mu}_0^{\phi}\left(\boldsymbol{\mu}_0^{\phi}\right)^{\top}+m_1\boldsymbol{\mu}_1^{\phi}\left(\boldsymbol{\mu}_1^{\phi}\right)^{\top}
\end{aligned}
$$

第四项：

$$\begin{aligned}\sum_{i=0}^{1}\sum_{\boldsymbol{x}\in X_i}\boldsymbol{\mu}_i^{\phi}\left(\boldsymbol{\mu}_i^{\phi}\right)^{\top} &= \sum_{\boldsymbol{x}\in X_0}\boldsymbol{\mu}_0^{\phi}\left(\boldsymbol{\mu}_0^{\phi}\right)^{\top}+\sum_{\boldsymbol{x}\in X_1}\boldsymbol{\mu}_1^{\phi}\left(\boldsymbol{\mu}_1^{\phi}\right)^{\top}\\ &= m_0\boldsymbol{\mu}_0^{\phi}\left(\boldsymbol{\mu}_0^{\phi}\right)^{\top}+m_1\boldsymbol{\mu}_1^{\phi}\left(\boldsymbol{\mu}_1^{\phi}\right)^{\top}\end{aligned}$$

将第一项至第四项各自的拆解结果代回到式(6.63)得最终$\boldsymbol{S}_w^{\phi}$拆解结果：

$$\boldsymbol{S}_w^{\phi}=\sum_{\boldsymbol{x}\in X}\phi(\boldsymbol{x})\phi(\boldsymbol{x})^{\top}-m_0\boldsymbol{\mu}_0^{\phi}\left(\boldsymbol{\mu}_0^{\phi}\right)^{\top}-m_1\boldsymbol{\mu}_1^{\phi}\left(\boldsymbol{\mu}_1^{\phi}\right)^{\top}$$

将式(6.65)代入式(6.60)的分母，$\boldsymbol{S}_w^{\phi}$按最终拆解结果分成三部分：

$$\boldsymbol{w}^{\top}\boldsymbol{S}_w^{\phi}\boldsymbol{w}=\left(\sum_{j=1}^{m}\alpha_j\phi(\boldsymbol{x}_j)\right)^{\top}\left(\sum_{\boldsymbol{x}\in X}\phi(\boldsymbol{x})\phi(\boldsymbol{x})^{\top}-m_0\boldsymbol{\mu}_0^{\phi}\left(\boldsymbol{\mu}_0^{\phi}\right)^{\top}-m_1\boldsymbol{\mu}_1^{\phi}\left(\boldsymbol{\mu}_1^{\phi}\right)^{\top}\right)\left(\sum_{k=1}^{m}\alpha_k\phi(\boldsymbol{x}_k)\right)$$

第一部分：

$$\begin{aligned}&\left(\sum_{j=1}^{m}\alpha_j\phi(\boldsymbol{x}_j)\right)^{\top}\left(\sum_{\boldsymbol{x}\in X}\phi(\boldsymbol{x})\phi(\boldsymbol{x})^{\top}\right)\left(\sum_{k=1}^{m}\alpha_k\phi(\boldsymbol{x}_k)\right)\\ &=\sum_{j=1}^{m}\sum_{\boldsymbol{x}\in X}\sum_{k=1}^{m}\left(\alpha_j\phi(\boldsymbol{x}_j)^{\top}\phi(\boldsymbol{x})\phi(\boldsymbol{x})^{\top}\alpha_k\phi(\boldsymbol{x}_k)\right)\\ &=\sum_{j=1}^{m}\sum_{\boldsymbol{x}\in X}\sum_{k=1}^{m}\left(\alpha_j\kappa(\boldsymbol{x}_j,\boldsymbol{x})\alpha_k\kappa(\boldsymbol{x},\boldsymbol{x}_k)\right)\\ &=\boldsymbol{\alpha}^{\top}\mathbf{K}\mathbf{K}^{\top}\boldsymbol{\alpha}\end{aligned}$$

第二部分：

$$\begin{aligned}&\left(\sum_{j=1}^{m}\alpha_j\phi(\boldsymbol{x}_j)\right)^{\top}\left(m_0\boldsymbol{\mu}_0^{\phi}\left(\boldsymbol{\mu}_0^{\phi}\right)^{\top}\right)\left(\sum_{k=1}^{m}\alpha_k\phi(\boldsymbol{x}_k)\right)\\ &=\left(\sum_{j=1}^{m}\alpha_j\phi(\boldsymbol{x}_j)\right)^{\top}\left(m_0\cdot\frac{1}{m_0}\sum_{\boldsymbol{x}\in X_0}\phi(\boldsymbol{x})\cdot\frac{1}{m_0}\sum_{\boldsymbol{x}\in X_0}\phi(\boldsymbol{x})^{\top}\right)\left(\sum_{k=1}^{m}\alpha_k\phi(\boldsymbol{x}_k)\right)\\ &=m_0\cdot\frac{1}{m_0}\sum_{j=1}^{m}\sum_{\boldsymbol{x}\in X_0}\alpha_j\phi(\boldsymbol{x}_j)^{\top}\phi(\boldsymbol{x})\cdot\frac{1}{m_0}\sum_{\boldsymbol{x}\in X_0}\sum_{k=1}^{m}\alpha_k\phi(\boldsymbol{x})^{\top}\phi(\boldsymbol{x}_k)\\ &=m_0\cdot\frac{1}{m_0}\sum_{j=1}^{m}\sum_{\boldsymbol{x}\in X_0}\alpha_j\kappa(\boldsymbol{x}_j,\boldsymbol{x})\cdot\frac{1}{m_0}\sum_{\boldsymbol{x}\in X_0}\sum_{k=1}^{m}\alpha_k\kappa(\boldsymbol{x},\boldsymbol{x}_k)\\ &=m_0\boldsymbol{\alpha}^{\top}\hat{\boldsymbol{\mu}}_0\hat{\boldsymbol{\mu}}_0^{\top}\boldsymbol{\alpha}\end{aligned}$$

第三部分：（与第二部分推导过程相同）

$$\left(\sum_{j=1}^{m}\alpha_j\phi(\boldsymbol{x}_j)\right)^{\top}\left(m_1\boldsymbol{\mu}_1^{\phi}\left(\boldsymbol{\mu}_1^{\phi}\right)^{\top}\right)\left(\sum_{k=1}^{m}\alpha_k\phi(\boldsymbol{x}_k)\right)$$
$$=m_1\boldsymbol{\alpha}^{\top}\hat{\boldsymbol{\mu}}_1\hat{\boldsymbol{\mu}}_1^{\top}\boldsymbol{\alpha}$$

将以上三部分的结果合并，并将式(6.69)代入：

$$\begin{aligned}\boldsymbol{w}^{\top}\boldsymbol{S}_w^{\phi}\boldsymbol{w}&=\boldsymbol{\alpha}^{\top}\mathbf{K}\mathbf{K}^{\top}\boldsymbol{\alpha}-m_0\boldsymbol{\alpha}^{\top}\hat{\boldsymbol{\mu}}_0\hat{\boldsymbol{\mu}}_0^{\top}\boldsymbol{\alpha}-m_1\boldsymbol{\alpha}^{\top}\hat{\boldsymbol{\mu}}_1\hat{\boldsymbol{\mu}}_1^{\top}\boldsymbol{\alpha}\\&=\boldsymbol{\alpha}^{\top}\left(\mathbf{K}\mathbf{K}^{\top}-m_0\hat{\boldsymbol{\mu}}_0\hat{\boldsymbol{\mu}}_0^{\top}-m_1\hat{\boldsymbol{\mu}}_1\hat{\boldsymbol{\mu}}_1^{\top}\right)\boldsymbol{\alpha}\\&=\boldsymbol{\alpha}^{\top}\left(\mathbf{K}\mathbf{K}^{\top}-\sum_{i=0}^{1}m_i\hat{\boldsymbol{\mu}}_i\hat{\boldsymbol{\mu}}_i^{\top}\right)\boldsymbol{\alpha}\\&=\boldsymbol{\alpha}^{\top}\mathbf{N}\boldsymbol{\alpha}\end{aligned}$$

综合以上分子分母的最终化简结果，即得式(6.70)。

【补充说明】有一些矩阵运算可能令初学者尤其是对矩阵运算不熟悉的人头疼，比如：

$$\sum_{i=1}^{m}\alpha_i\left(\frac{1}{m_1}\sum_{\boldsymbol{x}\in X_1}\kappa(\boldsymbol{x}_i,\boldsymbol{x})\right)=\boldsymbol{\alpha}^{\top}\hat{\boldsymbol{\mu}}_1$$

上式为什么成立呢？推导过程中类似的情况还有一些，在此仅以上式为例进行说明：

先重写出 P128 定理 6.1 核矩阵$\mathbf{K}$的具体内容：

$$\mathbf{K}=\begin{bmatrix}\kappa(\boldsymbol{x}_1,\boldsymbol{x}_1) & \kappa(\boldsymbol{x}_1,\boldsymbol{x}_2) & \cdots & \kappa(\boldsymbol{x}_1,\boldsymbol{x}_j) & \cdots & \kappa(\boldsymbol{x}_1,\boldsymbol{x}_k) & \cdots & \kappa(\boldsymbol{x}_1,\boldsymbol{x}_m)\\ \kappa(\boldsymbol{x}_2,\boldsymbol{x}_1) & \kappa(\boldsymbol{x}_2,\boldsymbol{x}_2) & \cdots & \kappa(\boldsymbol{x}_2,\boldsymbol{x}_j) & \cdots & \kappa(\boldsymbol{x}_2,\boldsymbol{x}_k) & \cdots & \kappa(\boldsymbol{x}_2,\boldsymbol{x}_m)\\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots\\ \kappa(\boldsymbol{x}_i,\boldsymbol{x}_1) & \kappa(\boldsymbol{x}_i,\boldsymbol{x}_2) & \cdots & \kappa(\boldsymbol{x}_i,\boldsymbol{x}_j) & \cdots & \kappa(\boldsymbol{x}_i,\boldsymbol{x}_k) & \cdots & \kappa(\boldsymbol{x}_i,\boldsymbol{x}_m)\\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots\\ \kappa(\boldsymbol{x}_m,\boldsymbol{x}_1) & \kappa(\boldsymbol{x}_m,\boldsymbol{x}_2) & \cdots & \kappa(\boldsymbol{x}_m,\boldsymbol{x}_j) & \cdots & \kappa(\boldsymbol{x}_m,\boldsymbol{x}_k) & \cdots & \kappa(\boldsymbol{x}_m,\boldsymbol{x}_m)\end{bmatrix}$$

假设X_1集合中包含$\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_j,\boldsymbol{x}_m$（即$m_1=4$），则

$$\frac{1}{m_1}\sum_{\boldsymbol{x}\in X_1}\kappa(\boldsymbol{x}_i,\boldsymbol{x})=\frac{1}{m_1}\left(\kappa(\boldsymbol{x}_i,\boldsymbol{x}_1)+\kappa(\boldsymbol{x}_i,\boldsymbol{x}_2)+\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)+\kappa(\boldsymbol{x}_i,\boldsymbol{x}_m)\right)$$

即核矩阵$\mathbf{K}$第i行当中属于X_1集合（列号）元素的平均值。也就是说，从$i=1$到m，上式计算的是第1行到第m行当中属于X_1集合（列号）元素的平均值。再看式(6.67)：

$$\hat{\boldsymbol{\mu}}_1 = \frac{1}{m_1}\begin{bmatrix} \kappa(\boldsymbol{x}_1,\boldsymbol{x}_1) & \kappa(\boldsymbol{x}_1,\boldsymbol{x}_2) & \cdots & \kappa(\boldsymbol{x}_1,\boldsymbol{x}_j) & \cdots & \kappa(\boldsymbol{x}_1,\boldsymbol{x}_k) & \cdots & \kappa(\boldsymbol{x}_1,\boldsymbol{x}_m) \\ \kappa(\boldsymbol{x}_2,\boldsymbol{x}_1) & \kappa(\boldsymbol{x}_2,\boldsymbol{x}_2) & \cdots & \kappa(\boldsymbol{x}_2,\boldsymbol{x}_j) & \cdots & \kappa(\boldsymbol{x}_2,\boldsymbol{x}_k) & \cdots & \kappa(\boldsymbol{x}_2,\boldsymbol{x}_m) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \kappa(\boldsymbol{x}_i,\boldsymbol{x}_1) & \kappa(\boldsymbol{x}_i,\boldsymbol{x}_2) & \cdots & \kappa(\boldsymbol{x}_i,\boldsymbol{x}_j) & \cdots & \kappa(\boldsymbol{x}_i,\boldsymbol{x}_k) & \cdots & \kappa(\boldsymbol{x}_i,\boldsymbol{x}_m) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \kappa(\boldsymbol{x}_m,\boldsymbol{x}_1) & \kappa(\boldsymbol{x}_m,\boldsymbol{x}_2) & \cdots & \kappa(\boldsymbol{x}_m,\boldsymbol{x}_j) & \cdots & \kappa(\boldsymbol{x}_m,\boldsymbol{x}_k) & \cdots & \kappa(\boldsymbol{x}_m,\boldsymbol{x}_m) \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

$$= \frac{1}{m_1}\begin{bmatrix} \kappa(\boldsymbol{x}_1,\boldsymbol{x}_1)+\kappa(\boldsymbol{x}_1,\boldsymbol{x}_2)+\kappa(\boldsymbol{x}_1,\boldsymbol{x}_j)+\kappa(\boldsymbol{x}_1,\boldsymbol{x}_m) \\ \kappa(\boldsymbol{x}_2,\boldsymbol{x}_1)+\kappa(\boldsymbol{x}_2,\boldsymbol{x}_2)+\kappa(\boldsymbol{x}_2,\boldsymbol{x}_j)+\kappa(\boldsymbol{x}_2,\boldsymbol{x}_m) \\ \vdots \\ \kappa(\boldsymbol{x}_i,\boldsymbol{x}_1)+\kappa(\boldsymbol{x}_i,\boldsymbol{x}_2)+\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)+\kappa(\boldsymbol{x}_i,\boldsymbol{x}_m) \\ \vdots \\ \kappa(\boldsymbol{x}_m,\boldsymbol{x}_1)+\kappa(\boldsymbol{x}_m,\boldsymbol{x}_2)+\kappa(\boldsymbol{x}_m,\boldsymbol{x}_j)+\kappa(\boldsymbol{x}_m,\boldsymbol{x}_m) \end{bmatrix}$$

也就是说$\hat{\boldsymbol{\mu}}_1$的第i个元素即为$\frac{1}{m_1}\sum_{\boldsymbol{x}\in X_1}\kappa(\boldsymbol{x}_i,\boldsymbol{x})$。因此$\sum_{i=1}^{m}\alpha_i\left(\frac{1}{m_1}\sum_{\boldsymbol{x}\in X_1}\kappa(\boldsymbol{x}_i,\boldsymbol{x})\right)$实际是$\boldsymbol{\alpha}=(\alpha_1,\alpha_2,\ldots,\alpha_m)^\top$和$\hat{\boldsymbol{\mu}}_1$的对应元素相乘再求和的结果，这正是内积运算$\boldsymbol{\alpha}^\top\hat{\boldsymbol{\mu}}_1$ 。

由于核矩阵是对称矩阵，即$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)=\kappa(\boldsymbol{x}_j,\boldsymbol{x}_i)$，而且内积运算结果是一个标量常数，即其转置等于本身，因此

$$\sum_{j=1}^{m}\left(\frac{1}{m_1}\sum_{\boldsymbol{x}\in X_1}\kappa(\boldsymbol{x},\boldsymbol{x}_j)-\frac{1}{m_0}\sum_{\boldsymbol{x}\in X_0}\kappa(\boldsymbol{x},\boldsymbol{x}_j)\right)\alpha_j=(\hat{\boldsymbol{\mu}}_1-\hat{\boldsymbol{\mu}}_0)^\top\boldsymbol{\alpha}$$

3、核对率回归（Kernelized Logistic Regression）

现将 6.4 节第 3 个解释“对率回归与支持向量机的关系”中最后的对率回归重写如下：

$$\min_{\boldsymbol{w},b}\frac{1}{m}\sum_{i=1}^{m}\log\left(1+e^{-y_i\left(\boldsymbol{w}^\top\boldsymbol{x}_i+b\right)}\right)+\frac{\lambda}{2m}\|\boldsymbol{w}\|^2$$

假设$\boldsymbol{z}_i=\phi(\boldsymbol{x}_i)$是由原始空间经核函数映射到高维空间的特征向量，则

$$\min_{\boldsymbol{w},b}\frac{1}{m}\sum_{i=1}^{m}\log\left(1+e^{-y_i\left(\boldsymbol{w}^\top\boldsymbol{z}_i+b\right)}\right)+\frac{\lambda}{2m}\|\boldsymbol{w}\|^2$$

注意，以上两式中的$\boldsymbol{w}$维度是不同的，分别与各自的特征向量维度一致。根据表示定理，上式的解可以写为

$$\boldsymbol{w}=\sum_{j=1}^{m}\alpha_j\boldsymbol{z}_j$$

将$\boldsymbol{w}$的表达式代入优化目标

$$\min_{\boldsymbol{w},b}\frac{1}{m}\sum_{i=1}^{m}\log\left(1+e^{-y_i\left(\sum_{j=1}^{m}\alpha_j\boldsymbol{z}_j^\top\boldsymbol{z}_i+b\right)}\right)+\frac{\lambda}{2m}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j\boldsymbol{z}_i^\top\boldsymbol{z}_j$$

用核函数$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)=\boldsymbol{z}_i^\top\boldsymbol{z}_j=\phi(\boldsymbol{x}_i)^\top\phi(\boldsymbol{x}_j)$替换上式中的内积运算

$$\min_{\boldsymbol{w},b} \frac{1}{m}\sum_{i=1}^{m}\log\left(1+e^{-y_i\left(\sum_{j=1}^{m}\alpha_j\kappa(\boldsymbol{x}_j,\boldsymbol{x}_i)+b\right)}\right)+\frac{\lambda}{2m}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)$$

那么如何求解呢？随便什么优化方法喽，比如梯度下降法~

解出$\boldsymbol{\alpha}=(\alpha_1;\alpha_2;\ldots;\alpha_m)$和$b$之后，即可得$f(\boldsymbol{x})=\sum_{i=1}^{m}\alpha_j\kappa(\boldsymbol{x}_i,\boldsymbol{x})+b$ 。

注：以上推导参考了林轩田老师《机器学习技法》课程的第 5 讲 PPT；另外，本章内容对应林轩田老师《机器学习技法》课程的第 1 讲至第 6 讲，基本上对书中涉及的每一个知识点都有较为详细的推导，因此若对支持向量机基本理论感兴趣，强烈推荐林老师的课程。