先来定义特征到结果的输出函数为：

$$u = \boldsymbol{\omega}^T \boldsymbol{x} + b$$

接着，我们回忆一下原始优化问题，如下：

$$\min \frac{1}{2} ||\omega||^2$$
$$\text{s.t.} \quad y_i\left(\boldsymbol{\omega}^T x_i + b\right) \geq 1, i = 1,2,\ldots,n$$

求导得：

$$\boldsymbol{\omega} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x_i}$$

将上述公式带入输出函数中：

$$u = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x_i}^T \boldsymbol{x} + b$$

与此同时，拉格朗日对偶后得到最终的目标化函数：

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j$$
$$s.t. \ \alpha_i \geq 0, i = 1, 2, \cdots, n$$
$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

将目标函数变形，在前面增加一个符号，将最大值问题转换成最小值问题：

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x_i}^T \boldsymbol{x_j} - \sum_{i=1}^{n} \alpha_i$$
$$s.t. \quad \alpha_i \geq 0, \ \ i = 1,2,\cdots,n$$
$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

实际上，对于上述目标函数，是存在一个假设的，即数据100%线性可分。但是，目前为止，我们知道几乎所有数据都不那么"干净"。这时我们就可以
的**松弛变量**(slack variable)，来允许有些数据点可以处于超平面的错误的一侧。这样我们的优化目标就能保持仍然不变，但是此时我们的约束条件有所改

$$s.t. \quad C \geq \alpha_i \geq 0, \ \ i = 1,2,\cdots,n$$
$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

根据KKT条件可以得出其中αi取值的意义为：

$$\alpha_i = 0 \;\Leftrightarrow\; y_i u_i \geq 1$$
$$0 < \alpha_i < C \;\Leftrightarrow\; y_i u_i = 1$$
$$\alpha_i = C \;\Leftrightarrow\; y_i u_i \leq 1$$

- 对于第1种情况，表明αi是正常分类，在边界内部；
- 对于第2种情况，表明αi是支持向量，在边界上；
- 对于第3种情况，表明αi是在两条边界之间。

而最优解需要满足KKT条件，即上述3个条件都得满足，以下几种情况出现将会不满足：

$$y_i u_i \leq 1 \quad \alpha_i < C$$
$$y_i u_i \geq 1 \quad \alpha_i > 0$$
$$y_i u_i = 1 \quad \alpha_i = 0 \text{ 或者 } \alpha_i = C$$

也就是说，如果存在不能满足KKT条件的αi，那么需要更新这些αi，这是第一个约束条件。此外，更新的同时还要受到第二个约束条件的限制，即：

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

因为这个条件，我们同时更新两个α值，因为只有成对更新，才能保证更新之后的值仍然满足和为0的约束，假设我们选择的两个乘子为α1和α2：

$$\alpha_1^{new} y_1 + \alpha_2^{new} y_2 = \alpha_1^{old} y_1 + \alpha_2^{old} y_2 = \zeta$$

其中， ksi为常数。因为两个因子不好同时求解，所以可以先求第二个乘子α2的解（α2 new），得到α2的解（α2 new）之后，再用α2的解（α2 ne
解（α1 new ）。为了求解α2 new ，得先确定α2 new的取值范围。假设它的上下边界分别为H和L，那么有：

$$L \leq \alpha_2^{new} \leq H$$

接下来，综合下面两个条件：

$$C \geq \alpha_i \geq 0, \;\; i = 1,2,\cdots,n$$
$$\alpha_1^{new} y_1 + \alpha_2^{new} y_2 = \alpha_1^{old} y_1 + \alpha_2^{old} y_2 = \zeta$$

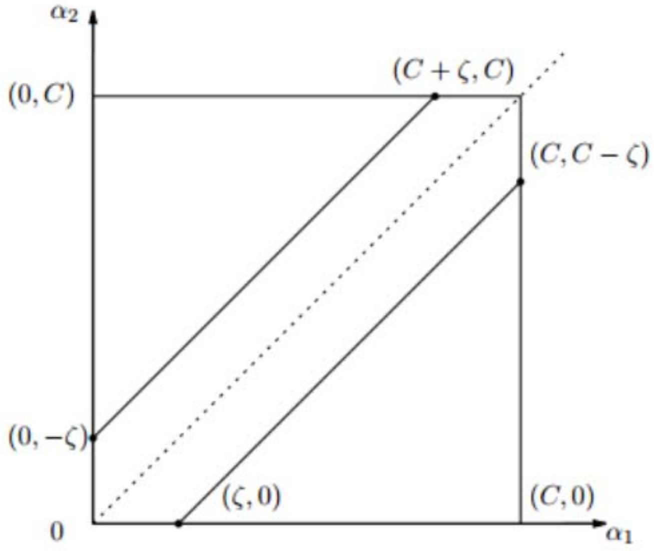
当y1不等于y2时，即一个为正1，一个为负1的时候，可以得到：

$$\alpha_1^{old} - \alpha_2^{old} = \zeta$$

所以有：

$$L = \max(0, -\zeta), \;\; H = \min(C, C - \zeta)$$

此时，取值范围如下图所示：



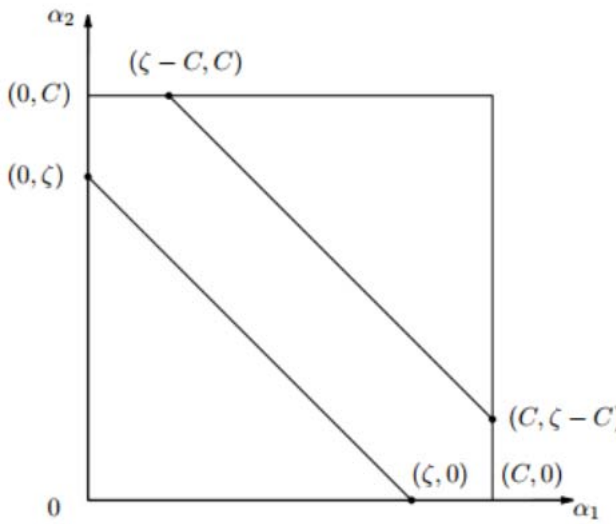
当y1等于y2时，即两个都为正1或者都为负1，可以得到：

$$\alpha_1^{old} + \alpha_2^{old} = \zeta$$

所以有：

$$L = \max(0, \zeta - C), \ \ H = \min(C, \zeta)$$

此时，取值范围如下图所示：



如此，根据y1和y2异号或同号，可以得出α2 new的上下界分别为：

$$\begin{cases} L = \max(0, \alpha_2^{old} - \alpha_1^{old}), H = \min(C, C + \alpha_2^{old} - \alpha_1^{old}) & \text{if } y_1 \neq y_2 \\ L = \max(0, \alpha_2^{old} + \alpha_1^{old} - C), H = \min(C, \alpha_2^{old} + \alpha_1^{old}) & \text{if } y_1 = y_2 \end{cases}$$

这个界限就是编程的时候需要用到的。已经确定了边界，接下来，就是推导迭代式，用于更新 α值。

我们已经知道，更新α的边界，接下来就是讨论如何更新α值。我们依然假设选择的两个乘子为α1和α2。固定这两个乘子，进行推导。于是目标函数变

$$
\begin{aligned}
W(\alpha_2) &= \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \boldsymbol{x_i}^T \boldsymbol{x_j}\alpha_i\alpha_j \\
&= \alpha_1 + \alpha_2 + \sum_{i=3}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\left(\sum_{j=1}^{2} y_i y_j \alpha_i\alpha_j \boldsymbol{x_i}^T\boldsymbol{x_j} + \sum_{j=3}^{n} y_i y_j \alpha_i\alpha_j \boldsymbol{x_i}^T\boldsymbol{x_j}\right) \\
&= \alpha_1 + \alpha_2 + \sum_{i=3}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{2}\left(\sum_{j=1}^{2} y_i y_j \alpha_i\alpha_j \boldsymbol{x_i}^T\boldsymbol{x_j} + \sum_{j=3}^{n} y_i y_j \alpha_i\alpha_j \boldsymbol{x_i}^T\boldsymbol{x_j}\right) \\
&\qquad - \frac{1}{2}\sum_{i=3}^{n}\left(\sum_{j=1}^{2} y_i y_j \alpha_i\alpha_j \boldsymbol{x_i}^T\boldsymbol{x_j} + \sum_{j=3}^{n} y_i y_j \alpha_i\alpha_j \boldsymbol{x_i}^T\boldsymbol{x_j}\right) \\
&= \alpha_1 + \alpha_2 + \sum_{i=3}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{2}\sum_{j=1}^{2} y_i y_j \alpha_i\alpha_j \boldsymbol{x_i}^T\boldsymbol{x_j} - \sum_{i=1}^{2}\sum_{j=3}^{n} y_i y_j \alpha_i\alpha_j \boldsymbol{x_i}^T\boldsymbol{x_j} \\
&\qquad - \frac{1}{2}\sum_{i=3}^{n}\sum_{j=3}^{n} y_i y_j \alpha_i\alpha_j \boldsymbol{x_i}^T\boldsymbol{x_j} \\
&= \alpha_1 + \alpha_2 - \frac{1}{2}\alpha_1^2\boldsymbol{x_1}^T\boldsymbol{x_1} - \frac{1}{2}\alpha_2^2\boldsymbol{x_2}^T\boldsymbol{x_2} - y_1 y_2 \alpha_1\alpha_2\boldsymbol{x_1}^T\boldsymbol{x_2} - y_1\alpha_1\sum_{j=3}^{n}\alpha_j y_j \boldsymbol{x_1}^T\boldsymbol{x_j} \\
&\qquad - y_2\alpha_2\sum_{j=3}^{n}\alpha_j y_j \boldsymbol{x_2}^T\boldsymbol{x_j} + \sum_{i=3}^{n}\alpha_i - \frac{1}{2}\sum_{i=3}^{n}\sum_{j=3}^{n} y_i y_j \alpha_i\alpha_j \boldsymbol{x_i}^T\boldsymbol{x_j}
\end{aligned}
$$

为了描述方便，我们定义如下符号：

$$
\begin{aligned}
f(\boldsymbol{x_i}) &= \sum_{j=1}^{n}\alpha_j y_j \boldsymbol{x_i}^T\boldsymbol{x_j} + b \\
v_i &= \sum_{j=3}^{n}\alpha_j y_j \boldsymbol{x_i}^T\boldsymbol{x_j} = f(\boldsymbol{x_i}) - \sum_{j=1}^{2}\alpha_j y_j \boldsymbol{x_i}^T\boldsymbol{x_j} - b
\end{aligned}
$$

最终目标函数变为：

$$
\begin{aligned}
W(\alpha_2) = \alpha_1 + \alpha_2 - \frac{1}{2}\alpha_1^2\boldsymbol{x_1}^T\boldsymbol{x_1} - \frac{1}{2}\alpha_2^2\boldsymbol{x_2}^T\boldsymbol{x_2} - y_1 y_2\alpha_1\alpha_2\boldsymbol{x_1}^T\boldsymbol{x_2} - y_1\alpha_1 v_1 - y_2\alpha_2 v_2 \\
+ constant
\end{aligned}
$$

我们不关心constant的部分，因为对于α1和α2来说，它们都是常数项，在求导的时候，直接变为0。对于这个目标函数，如果对其求导，还有个未知推导出α1和α2的关系，然后用α2代替α1，这样目标函数就剩一个未知数了，我们就可以求导了，推导出迭代公式。所以现在继续推导α1和α2的关系。注条件：

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

我们在求α1和α2的时候，可以将α3,α4,...,αn和y3,y4,...,yn看作常数项。因此有：

$$
\alpha_1 y_1 + \alpha_2 y_2 = B
$$

我们不必关心常数B的大小，现在将上述等式两边同时乘以y1，得到(y1y1=1)：

$$
\alpha_1 = \gamma - s\alpha_2
$$

其中γ为常数By1，我们不关心这个值，s=y1y2。接下来，我们将得到的α1带入W(α2)公式得：

$$
\begin{aligned}
W(\alpha_2) = \gamma - s\alpha_2 + \alpha_2 - \frac{1}{2}(\gamma - s\alpha_2)^2\boldsymbol{x_1}^T\boldsymbol{x_1} - \frac{1}{2}\alpha_2^2\boldsymbol{x_2}^T\boldsymbol{x_2} - s(\gamma - s\alpha_2)\alpha_2\boldsymbol{x_1}^T\boldsymbol{x_2} \\
- y_1(\gamma - s\alpha_2)v_1 - y_2\alpha_2 v_2 + constant
\end{aligned}
$$

这样目标函数中就只剩下α2了，我们对其求偏导（注意：s=y1y2，所以s的平方为1，y1的平方和y2的平方均为1）：

$$
\begin{aligned}
\frac{\partial W(\alpha_2)}{\partial \alpha_2} &= -s + 1 + \gamma s\boldsymbol{x_1}^T\boldsymbol{x_1} - \alpha_2\boldsymbol{x_1}^T\boldsymbol{x_1} - \alpha_2\boldsymbol{x_2}^T\boldsymbol{x_2} \\
&\quad - \gamma s\boldsymbol{x_1}^T\boldsymbol{x_2} + 2\alpha_2\boldsymbol{x_1}^T\boldsymbol{x_2} + y_2 v_1 - y_2 v_2 = 0
\end{aligned}
$$

继续化简，将s=y1y2带入方程。

$$
\alpha_2^{new} = \frac{y_2\left(y_2 - y_1 + y_1\gamma(\boldsymbol{x_1}^T\boldsymbol{x_1} - \boldsymbol{x_1}^T\boldsymbol{x_2}) + v_1 - v_2\right)}{\boldsymbol{x_1}^T\boldsymbol{x_1} + \boldsymbol{x_2}^T\boldsymbol{x_2} - 2\boldsymbol{x_1}^T\boldsymbol{x_2}}
$$

我们令：

$$E_i = f(x_i) - y_i$$
$$\eta = \boldsymbol{x_1}^T\boldsymbol{x_1} + \boldsymbol{x_2}^T\boldsymbol{x_2} - 2\boldsymbol{x_1}^T\boldsymbol{x_2}$$

Ei为误差项，η为学习速率。

再根据我们已知的公式：

$$\gamma = \alpha_1{}^{\mathrm{old}} + s\alpha_2{}^{old}$$
$$v_j = \sum_{i=3}^{n} \alpha_i y_i \boldsymbol{x_j}^T \boldsymbol{x_i} = f(\boldsymbol{x_j}) - \sum_{i=1}^{2} \alpha_i y_i \boldsymbol{x_j}^T \boldsymbol{x_i} - b$$

将α2 new继续化简得：

$$\alpha_2^{new} = \alpha_2^{old} + \frac{y_2(E_1 - E_2)}{\eta}$$

这样，我们就得到了最终需要的迭代公式。这个是没有经过剪辑是的解，需要考虑约束：

$$0 < \alpha_i < C$$

根据之前推导的α取值范围，我们得到最终的解析解为：

$$\alpha_2^{new,clipped} = \begin{cases} H & \alpha_2^{new} > H \\ \alpha_2^{new} & L \leq \alpha_2^{new} \leq H \\ L & \alpha_2^{new} < L \end{cases}$$

又因为：

$$\alpha_1^{old} = \gamma - s\alpha_2^{old}$$
$$\alpha_1^{new} = \gamma - s\alpha_2^{new,clipped}$$

消去γ得：

$$\alpha_1^{new} = \alpha_1^{old} + y_1 y_2 (\alpha_2^{old} - \alpha_2^{new,clipped})$$

这样，我们就知道了怎样计算α1和α2了，也就是如何对选择的α进行更新。

当我们更新了α1和α2之后，需要重新计算阈值b，因为b关系到了我们f(x)的计算，也就关系到了误差Ei的计算。

我们要根据α的取值范围，去更正b的值，使间隔最大化。当α1 new在0和C之间的时候，根据KKT条件可知，这个点是支持向量上的点。因此，满足

$$y_1(\boldsymbol{\omega}^T \boldsymbol{x_1} + b) = 1$$

公式两边同时乘以y1得(y1y1=1)：

$$\sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x_i} \boldsymbol{x_1} + b = y_1$$

因为我们是根据α1和α2的值去更新b，所以单独提出i=1和i=2的时候，整理可得：

$$b_1^{new} = y_1 - \sum_{i=3}^{n} \alpha_i y_i \boldsymbol{x_i}^T \boldsymbol{x_1} - \alpha_1{}^{new} y_1 \boldsymbol{x_1}^T \boldsymbol{x_1} - \alpha_2{}^{new} y_2 \boldsymbol{x_2}^T \boldsymbol{x_1}$$

其中前两项为：

$$y_1 - \sum_{i=3}^{n} \alpha_i y_i \boldsymbol{x_i}^T \boldsymbol{x_1} = -E_1 + \alpha_1{}^{old} y_1 \boldsymbol{x_1}^T \boldsymbol{x_1} + \alpha_2{}^{old} y_2 \boldsymbol{x_2}^T \boldsymbol{x_1} + b^{old}$$

将上述两个公式，整理得：

$$b_1^{new} = b^{old} - E_1 - y_1(\alpha_1{}^{new} - \alpha_1{}^{old})\boldsymbol{x_1}^T \boldsymbol{x_1} - y_2(\alpha_2{}^{new} - \alpha_2{}^{old})\boldsymbol{x_2}^T \boldsymbol{x_1}$$

同理可得b2 new为：

$$b_2^{new} = b^{old} - E_2 - y_1(\alpha_1{}^{new} - \alpha_1{}^{old})\boldsymbol{x_1}^{\boldsymbol{T}}\boldsymbol{x_2} - y_2(\alpha_2{}^{new} - \alpha_2{}^{old})\boldsymbol{x_2}^{\boldsymbol{T}}\boldsymbol{x_2}$$

当b1和b2都有效的时候，它们是相等的，即：

$$b^{new} = b_1^{new} = b_2^{new}$$

当两个乘子都在边界上，则b阈值和KKT条件一致。当不满足的时候，SMO算法选择他们的中点作为新的阈值：

$$b = \begin{cases} b_1, & 0 < \alpha_1^{new} < C \\ b_2 & 0 < \alpha_2^{new} < C \\ (b_1 + b_2)/2 & \text{otherwise} \end{cases}$$

最后，更新所有的α和b，这样模型就出来了，从而即可求出我们的分类函数。

现在，让我们梳理下SMO算法的步骤：

- 步骤1：计算误差：

$$E_i = f(\boldsymbol{x_i}) - y_i = \sum_{j=1}^{n} \alpha_j y_j \boldsymbol{x_i}^{\boldsymbol{T}}\boldsymbol{x_j} + b - y_i$$

- 步骤2：计算上下界L和H：

$$\begin{cases} L = \max(0, \alpha_j{}^{old} - \alpha_i{}^{old}), H = \min(C, C + \alpha_j{}^{old} - \alpha_i{}^{old}) & if\ y_i \neq y_j \\ L = \max(0, \alpha_j{}^{old} + \alpha_i{}^{old} - C), H = \min(C, \alpha_j{}^{old} + \alpha_i{}^{old}) & if\ y_i \neq y_j \end{cases}$$

- 步骤3：计算η：

$$\eta = \boldsymbol{x_i}^{\boldsymbol{T}}\boldsymbol{x_i} + \boldsymbol{x_j}^{\boldsymbol{T}}\boldsymbol{x_j} - \boldsymbol{2x_i}^{\boldsymbol{T}}\boldsymbol{x_j}$$

- 步骤4：更新αj：

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

- 步骤5：根据取值范围修剪αj：

$$\alpha^{new,clipped} = \begin{cases} H & if\ \alpha_2^{new} \geq H \\ \alpha_2^{new} & if\ L \leq \alpha_2^{new} \leq H \\ L & if\ \alpha_2^{new} \leq L \end{cases}$$

- 步骤6：更新αi：

$$\alpha_i^{new} = \alpha_i^{old} + y_i y_j (\alpha_j^{old} - \alpha_j^{new,clipped})$$

- 步骤7：更新b1和b2：

$$b_1^{new} = b^{old} - E_i - y_i(\alpha_i{}^{new} - \alpha_i{}^{old})\boldsymbol{x_i}^{\boldsymbol{T}}\boldsymbol{x_i} - y_j(\alpha_j{}^{new} - \alpha_j{}^{old})\boldsymbol{x_j}^{\boldsymbol{T}}\boldsymbol{x_i}$$

$$b_2^{new} = b^{old} - E_j - y_i(\alpha_i{}^{new} - \alpha_i{}^{old})\boldsymbol{x_i}^{\boldsymbol{T}}\boldsymbol{x_j} - y_j(\alpha_j{}^{new} - \alpha_j{}^{old})\boldsymbol{x_j}^{\boldsymbol{T}}\boldsymbol{x_j}$$

- 步骤8：根据b1和b2更新b：

$$b = \begin{cases} b_1 & 0 < \alpha_1^{new} < C \\ b_2 & 0 < \alpha_2^{new} < C \\ \dfrac{b_1 + b_2}{2} & otherwise \end{cases}$$