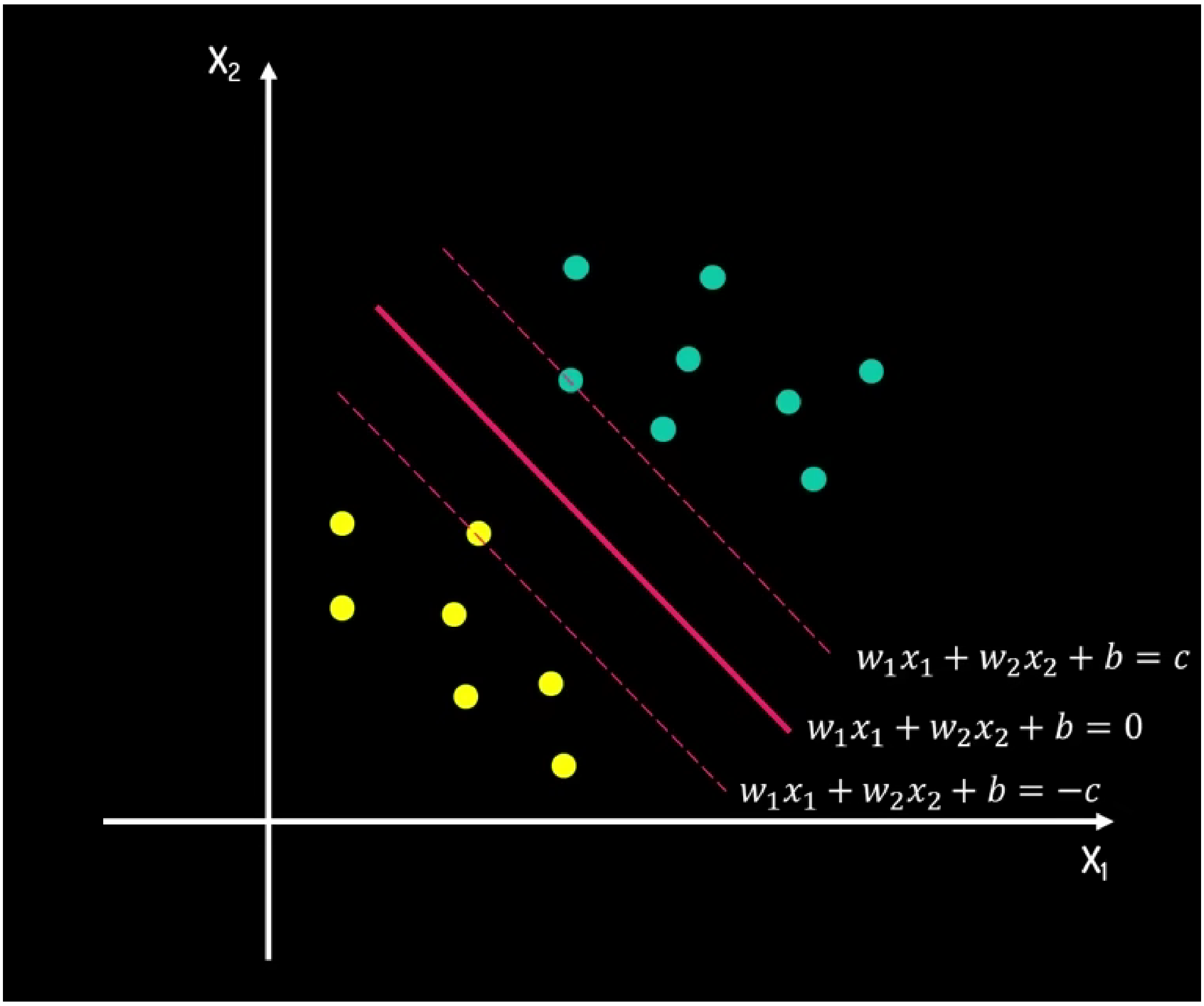
# 第三周 | Third Week：SVM（支持向量机）

## SVM公式推导

### 问题简化

已知n个样本，样本维度为m，如何求得一个**超平面（hyperplane）**$\sum_{i=1}^{m} W_i X_i + B = 0$将两类样本分隔开来。最佳决策边界，即求解两类数据的最大间隔问题。而间隔的正中，就是我们的决策边界。**支持向量（Support Vector）**就是距离分隔超平面最近的那些点。




写出决策边界和间隔上下边界方程，两边同除C：

$$
\begin{aligned}
w_1x_1 + w_2x_2 + b = c \quad & \frac{w_1}{c}x_1 + \frac{w_2}{c}x_2 + \frac{b}{c} = 1 \\
w_1x_1 + w_2x_2 + b = 0 \quad & \frac{w_1}{c}x_1 + \frac{w_2}{c}x_2 + \frac{b}{c} = 0 \\
w_1x_1 + w_2x_2 + b = -c \quad & \frac{w_1}{c}x_1 + \frac{w_2}{c}x_2 + \frac{b}{c} = -1
\end{aligned}
$$

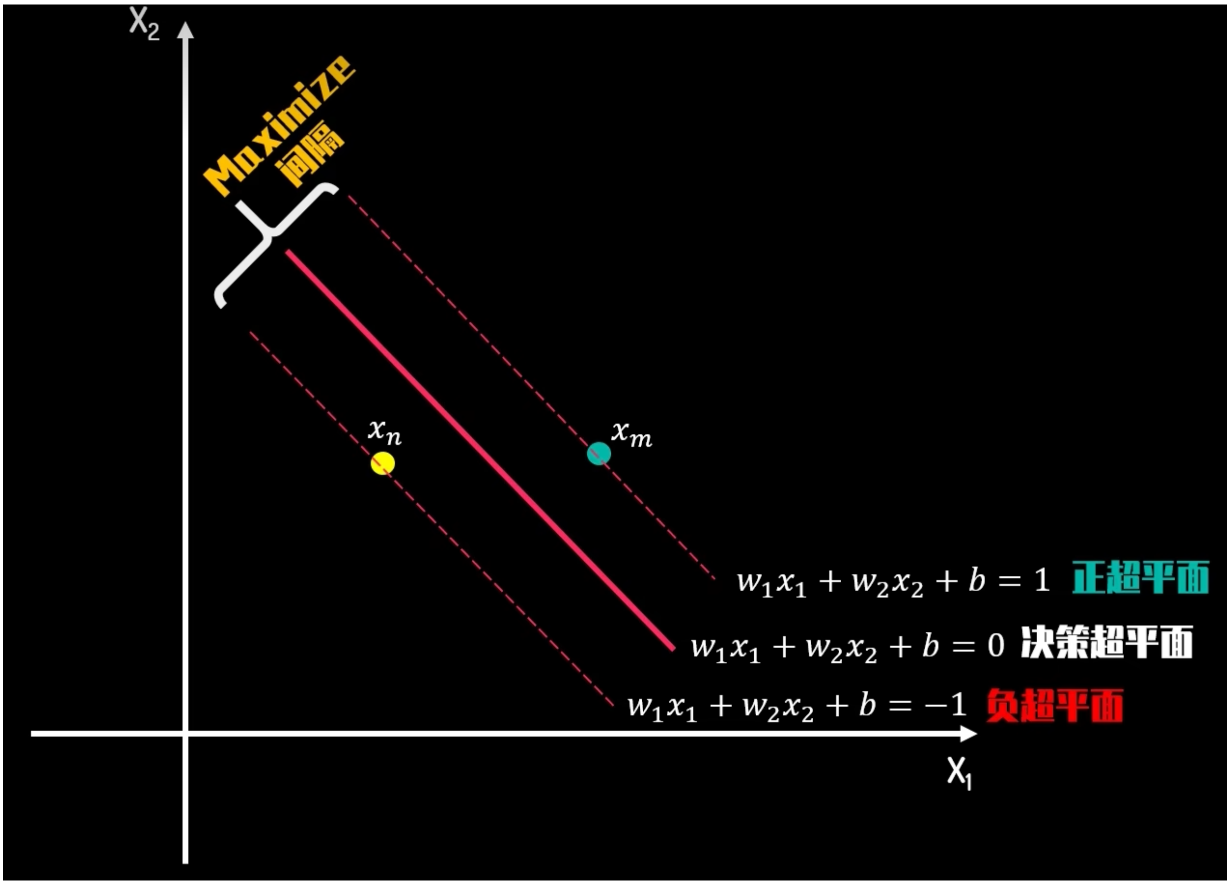
作变量代换

$$
w_1' = \frac{w_1}{c} \quad w_2' = \frac{w_2}{c} \quad b' = \frac{b}{c}
$$

w,b只是代号，可以不用写成$w'$的形式，可得：

$$
\begin{aligned}
w_1x_1 + w_2x_2 + b = 1 \\
w_1x_1 + w_2x_2 + b = 0 \\
w_1x_1 + w_2x_2 + b = -1
\end{aligned}
$$

依次为：**正超平面，决策超平面，负超平面**。现在我们要最大化两者距离。




$x_m, x_n$位于正负超平面上，满足：

$$(1) w_1 x_{1\,\mathrm{m}} + w_2 x_{2\mathrm{m}} + b = 1$$
$$(2) w_1 x_{1\mathrm{n}} + w_2 x_{2\mathrm{n}} + b = -1$$

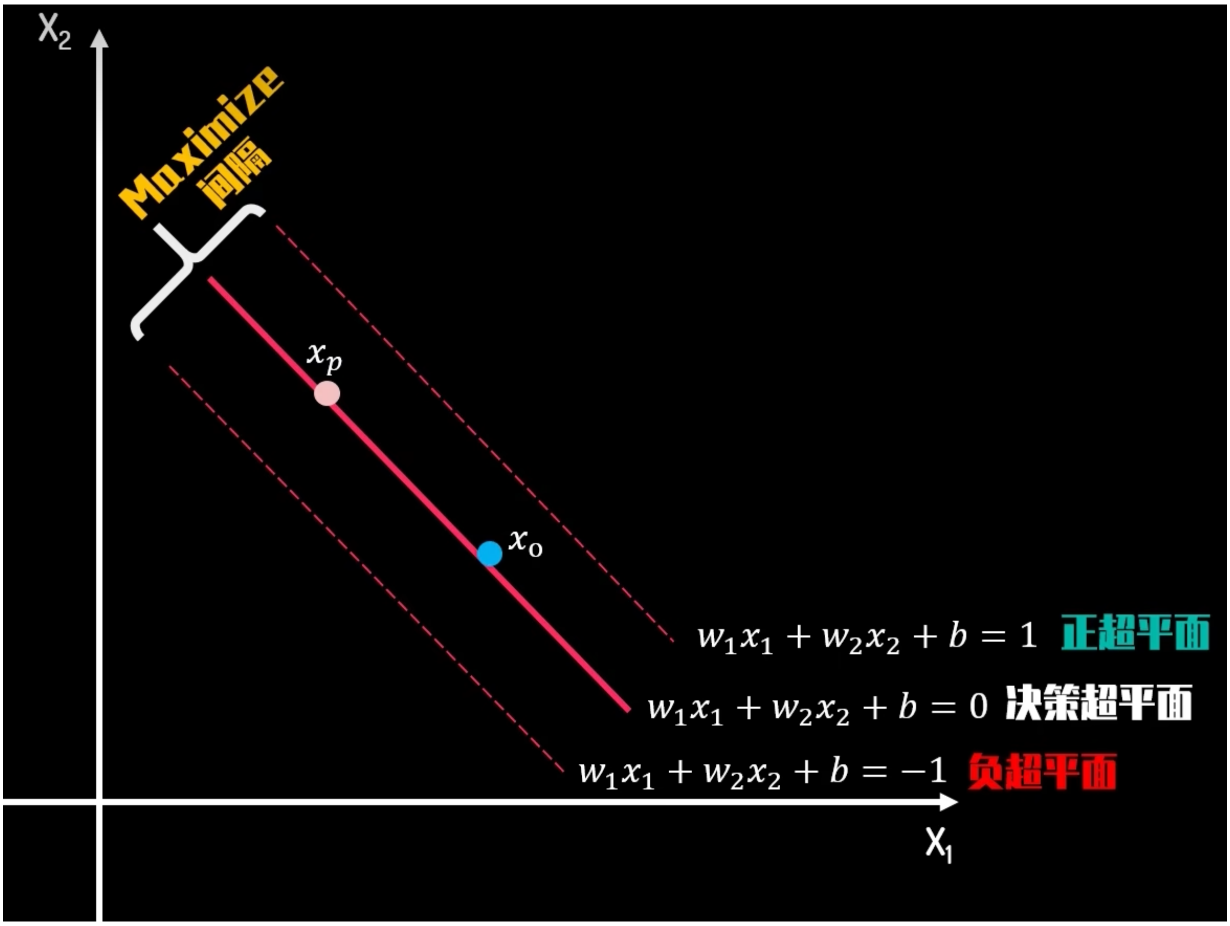
(1) - (2) 得：

$$(3) w_1 \left(x_{1\,\mathrm{m}} - x_{1n}\right) + w_2 \left(x_{2\,\mathrm{m}} - x_{2n}\right) = 2$$

即：

$$(4) \vec{w} \cdot \left(\vec{x}_m - \vec{x}_n\right) = 2$$

同理我们选取位于决策超平面上的点O，P：




同理可得：

$$
\begin{gathered}
(5) w_1 x_{1o} + w_2 x_{2o} + b = 0 \\
(6) w_1 x_{1p} + w_2 x_{2p} + b = 0 \\
(5)-(6): w_1\left(x_{1o} - x_{1p}\right) + w_2\left(x_{2o} - x_{2p}\right) = 0 \\
(7) \vec{w} \cdot \left(\vec{x}_o - \vec{x}_p\right) = 0
\end{gathered}
$$

即$\vec{w}$是决策超平面的法向量。因此（4）式等价于：

$$
\left\|\vec{x}_m - \vec{x}_n\right\| * \cos\theta * \|\vec{w}\| = 2
$$

而我们要求的间隔为：

$$
\left\|\vec{x}_m - \vec{x}_n\right\| * \cos\theta = L
$$

因此：

$$
\begin{gathered}
L * \|\vec{w}\| = 2 \\
L = \frac{2}{\|\vec{w}\|}
\end{gathered}
$$

位于正超平面上侧的点满足：

$$
\begin{cases}
y_i = 1 \\
\vec{w} \cdot \vec{x}_i + b \geq 1
\end{cases}
$$

式中$y_i$为分类值，同理负超平面下侧的点满足：

$$
\begin{cases}
y_i = -1 \\
\vec{w} \cdot \vec{x}_i + b \leq 1
\end{cases}
$$

故约束条件可以进一步简化为：

$$
y_i * \left(\vec{w} \cdot \vec{x}_i + b\right) \geq 1
$$

可得优化问题：

$$
\operatorname{minimize} \|\vec{w}\|
$$

其中：

$$\|\vec{w}\| = \sqrt{w_1^2 + w_2^2}$$

## KKT条件

为方便计算，令$f(w) = \frac{\|\vec{w}\|^2}{2}$,可得

$$\text{minimize} \quad f(w) = \frac{\|\vec{w}\|^2}{2}$$

约束条件为：

$$g_i(w,b) = y_i * (\vec{w} \cdot \vec{x}_i + b) - 1 \geq 0 \quad i = 1,2,\cdots,s, \text{s为样本数}$$

为使用Lagrange数乘法，令

$$g_i(w,b) = p_i^2$$

得到如下得Lagrange方程式：

$$L\left(w,b,\lambda_i,p_i\right) = \frac{\|\vec{w}\|^2}{2} - \sum_{i=1}^{s} \lambda_i * \left(y_i * \left(\vec{w} \cdot \vec{x}_i + b\right) - 1 - p_i^2\right) \text{求偏导得：}$$

对$w,b,\lambda_i,p_i$求偏导得：

$$(1)\vec{w} - \sum_{i=1}^{S} \lambda_i y_i \vec{x}_i = 0$$

$$(2) - \sum_{i=1}^{s} \lambda_i y_i = 0$$

$$(3) y_i * \left(\vec{w} \cdot \vec{x}_i + b\right) - 1 - p_i^2 = 0$$

$$(4) 2\lambda_i p_i = 0 \Longrightarrow \lambda_i p_i^2 = 0$$

对（4）变形得：

$$(4) 2\lambda_i p_i = 0 \Longrightarrow \lambda_i p_i^2 = 0$$

将（3）带入（4）得：

$$\lambda_i\left(y_i * \left(\vec{w} \cdot \vec{x}_i + b\right) - 1\right) = 0$$

注意到：

$$y_i * \left(\vec{w} \cdot \vec{x}_i + b\right) \geq 1$$

故有：

$$\begin{cases} y_i * \left(\vec{w} \cdot \vec{x}_i + b\right) - 1 > 0, \lambda_i = 0 \\ y_i * \left(\vec{w} \cdot \vec{x}_i + b\right) - 1 = 0, \lambda_i \neq 0 \end{cases}$$

再看

$$L\left(w,b,\lambda_i,p_i\right) = \frac{\|\vec{w}\|^2}{2} - \sum_{i=1}^{s} \lambda_i * \left(y_i * \left(\vec{w} \cdot \vec{x}_i + b\right) - 1 - p_i^2\right)$$

将$\lambda_i$看成违背约束条件得惩罚系数，不满足约束条件时,$(\vec{w} \cdot \vec{x}_i + b) - 1 < 0$,如果$\lambda_i < 0$,相当于Lagrange函数最后会变得更小，鼓励违反约束而获得更小的解，这不符合常理，可以推断出：

$$\lambda_i \geq 0$$

最后我们得到以下五个条件，即**KKT条件**

$$\begin{cases}\vec{w}-\sum_{i=1}^{s}\lambda_i y_i\vec{x}_i=0\\ -\sum_{i=1}^{s}\lambda_i y_i=0\\ y_i*(\vec{w}\cdot\vec{x}_i+b)-1\geq 0\\ \lambda_i\left(y_i*(\vec{w}\cdot\vec{x}_i+b)-1\right)=0\\ \lambda_i\geq 0\end{cases}$$

## SVM对偶性

原问题：

$$\begin{aligned}&\text{minimize} \quad f(w)=\frac{\|\vec{w}\|^2}{2}\\ &\text{subject to} \quad g_i(w,b)=y_i*(\vec{w}\cdot\vec{x}_i+b)-1\geq 0 \quad i=1,2,\cdots,s,\text{s为样本数}\end{aligned}$$

问题有最优解$w^*,b^*$

构造：

$$q(\lambda_i)=\text{ minimize }\left(L(w,b,\lambda_i)\right)=\text{ minimize }\left(f(w)-\sum_{i=1}^{s}\lambda_i*g_i(w,b)\right),\quad i=1,2,3,4\ldots s$$

必然有：

$$q(\lambda_i)=\text{ minimize }\left(L(w,b,\lambda_i)\right)=\text{ minimize }\left(f(w)-\sum_{i=1}^{s}\lambda_i*g_i(w,b)\right)\leq \boldsymbol{f}\left(\overrightarrow{\boldsymbol{w}^*}\right)-\sum_{i=1}^{s}\lambda_i*\boldsymbol{g}_i\left(\overrightarrow{\boldsymbol{w}^*},\boldsymbol{b}^*\right)$$

根据KKT条件：

$$\begin{cases}\lambda_i\geq 0\\ g_i\left(\overrightarrow{w^*},b^*\right)\geq 0\end{cases}$$

$\boldsymbol{f}\left(\overrightarrow{\boldsymbol{w}^*}\right)$所减项为正，故有

$$q(\lambda_i)\leq f\left(\overrightarrow{w^*}\right)\leq f(w)$$

寻找最优下界：让$q(\lambda_i^*)$与$f(\vec{w^*})$尽可能的接近，有：

$$q(\lambda_i)\leq q(\lambda_i^*)\leq f\left(\overrightarrow{w^*}\right)\leq f(w)$$

故原问题可以等价于如下的对偶问题：

$$\begin{aligned}&\text{minimize} \quad q(\lambda_i)=\text{maximize}\left(\text{minimize}\left(L(w,b,\lambda_i)\right)\right)\\ &\text{subject to} \quad \lambda_i\geq 0,\quad i=1,2,3,4,\ldots,s\end{aligned}$$

当$q(\lambda_i^*)<f\left(\overrightarrow{w^*}\right)$时，称为弱对偶；当$q(\lambda_i^*)=f\left(\overrightarrow{w^*}\right)$时，称为强对偶，可以证明当强对偶条件成立时原问题和对偶问题同时达到最优解。即：

$$\begin{aligned}&f(w)\geq q(\lambda_i^*)=f\left(\overrightarrow{w^*}\right) && f(w)\geq f\left(\overrightarrow{w^*}\right)\\ &f(w)\geq q(\lambda_i^*)=f\left(\overrightarrow{w^*}\right)\geq q(\lambda_i) && q(\lambda_i^*)\geq q(\lambda_i)\end{aligned}$$

由此，对偶问题可化为：

$$\begin{gathered}\text{minimize} \quad q(\lambda)=\text{maximize}\left(\text{minimize}\left(\frac{\|\vec{w}\|^2}{2}-\sum_{i=1}^{s}\lambda_i*\left(y_i*(\vec{w}\cdot\vec{x}_i+b)-1\right)\right)\right)\\ \text{subject to} \quad \lambda_i\geq 0,\quad i=1,2,3,4\ldots s\end{gathered}$$

带入KKT条件（1），（2）得：

$$\text{maximize} \quad q\left(\lambda_i\right)=\text{maximize}\left(\frac{1}{2}\left(\sum_{i=1}^{s}\lambda_i y_i \vec{x}_i\right)\cdot\left(\sum_{j=1}^{s}\lambda_j y_j \vec{x}_j\right)-\sum_{i=1}^{s}\lambda_i * \left(y_i * \left(\left(\left(\sum_{j=1}^{s}\lambda_j y_j \vec{x}_j\right)\cdot\vec{x}_i+b\right)-1\right)\right)\right)$$

即:

$$\text{maximize}\, q\left(\lambda_i\right)=\text{maximize}\left(\sum_{i=1}^{s}\lambda_i-\frac{1}{2}\sum_{i=1}^{s}\sum_{j=1}^{s}\lambda_i\lambda_j y_i y_j \overrightarrow{\boldsymbol{x}_{\boldsymbol{i}}}\cdot\overrightarrow{\boldsymbol{x}_{\boldsymbol{j}}}\right)$$

我们可以通过支持向量求解$\lambda_i^*$，再由KKT（1）求得w，再求得b，便达成我们的目的。

## SVM核技巧 | Kernel Trick

求解$\lambda_i^*$，取决于$y_i y_j \overrightarrow{\boldsymbol{x}_{\boldsymbol{i}}}\cdot\overrightarrow{\boldsymbol{x}_{\boldsymbol{j}}}$,若方程在原维度无解(非线性)，可以通过构造维度转换函数T（x）,得到新的维度数据向量T（x），可以得到如下的优化问题:

$$\text{maximize}\, q\left(\lambda_i\right)=\text{maximize}\left(\sum_{i=1}^{S}\lambda_i-\frac{1}{2}\sum_{i=1}^{s}\sum_{j=1}^{s}\lambda_i\lambda_j y_i y_j T\left(\vec{x}_i\right)\cdot T\left(\vec{x}_j\right)\right)$$
$$\text{subject to} \quad \lambda_i \geq 0, \quad i=1,2,3,4\ldots S$$

如果先求出T（x）,再将其点积，当数据维度为无穷维时，无法求解，故我们引入核函数$K\left(\vec{x}_i,\vec{x}_j\right)$，使得:

$$T\left(\vec{x}_i\right)\cdot T\left(\vec{x}_j\right)=K\left(\vec{x}_i,\vec{x}_j\right)=\left(c+\vec{x}_i\cdot\vec{x}_j\right)^d$$

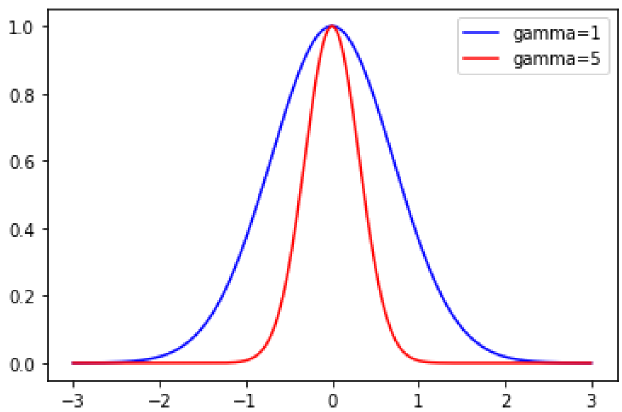
当我们需要做如下的维度转化:

$$\vec{x}=\left(x_1,x_2\right)\xrightarrow{T}T(\vec{x})=\left(a_1x_1,a_1x_2,a_1x_1x_2,a_2x_1^2,a_2x_2^2,\ldots a_nx_2^n,a_nx_2^n,\ldots\infty\right)$$

我们可以引入**高斯核函数（Radial Basis Kernel）**：

$$K\left(\vec{x}_i,\vec{x}_{\boldsymbol{j}}\right)=e^{-\gamma\left\|\vec{x}_i-\vec{x}_j\right\|^2}$$

当gamma确定后，两点的距离越大，其相似度越接近于0；两点的距离越小，其相似度越接近于1。




令$\gamma=0.5$,作如下推导:

$$\begin{aligned}
&K\left(\vec{x}_i,\vec{x}_j\right)\\
&=e^{-\frac{\left\|\vec{x}_i-\vec{x}_j\right\|^2}{2}}\\
&=e^{-\frac{1}{2}\left(\vec{x}_i-\vec{x}_j\right)\cdot\left(\vec{x}_i-\vec{x}_j\right)}\\
&=e^{-\frac{1}{2}\left(\vec{x}_i\cdot\vec{x}_i+\vec{x}_j\cdot\vec{x}_j-2\vec{x}_i\cdot\vec{x}_j\right)}\\
&=e^{-\frac{1}{2}\left(\left\|\vec{x}_i\right\|^2+\left\|\vec{x}_j\right\|^2-2\vec{x}_i\cdot\vec{x}_j\right)}\\
&=e^{-\frac{1}{2}\left(\left\|\vec{x}_i\right\|^2+\left\|\vec{x}_j\right\|^2\right)}e^{\vec{x}_i\cdot\vec{x}_j}
\end{aligned}$$

令$C=e^{-\frac{1}{2}\left(\left\|\vec{x}_i\right\|^2+\left\|\vec{x}_j\right\|^2\right)}$,Taylor展开得:

$$K\left(\vec{x}_i,\vec{x}_j\right)=Ce^{\vec{x}_i\cdot\vec{x}_j}=C\sum_{n=0}^{\infty}\frac{\vec{x}_i\cdot\vec{x}_j^n}{n!}=C\sum_{n=0}^{\infty}\frac{K_{Poly(n)}\left(\vec{x}_i,\vec{x}_j\right)}{n!}$$

## 软间隔 | SoftMargin

如果出现一个点a违法了约束条件$y_i * \left(\vec{w} \cdot \vec{x}_i + b\right) \geq 1$，即：

$$y_i * \left(\vec{w} \cdot \vec{x}_i + b\right) < 1$$

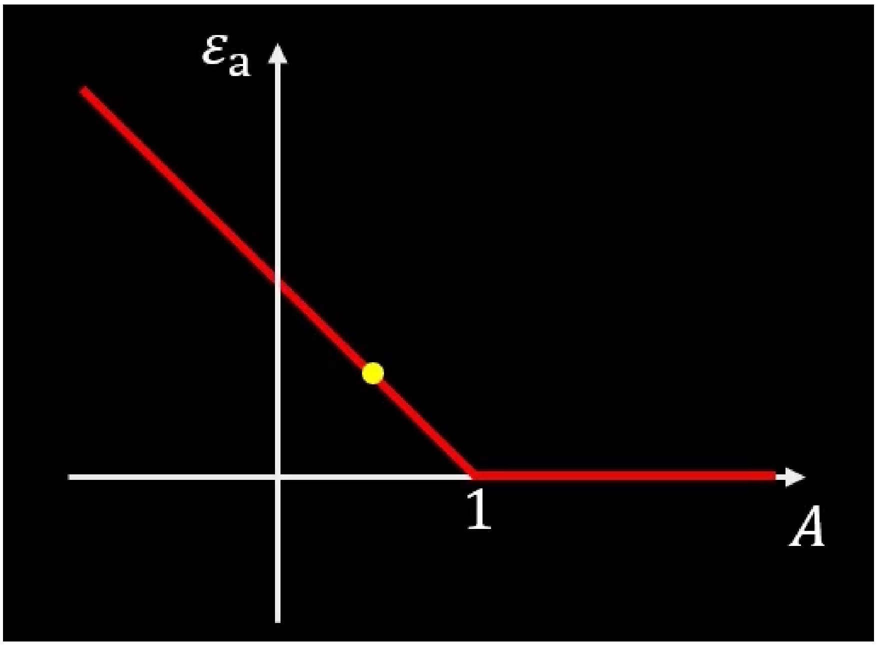
为了量化点a误差，引入：

$$\varepsilon_{\mathrm{a}} = 1 - y_{\mathrm{a}} * \left(\vec{w} \cdot \vec{x}_a + b\right)$$

令

$$A = y_{\mathrm{a}} * \left(\vec{w} \cdot \vec{x}_a + b\right)$$

很显然有：




对于任意点i，其损失值可以用如下得损失函数表示：

$$\varepsilon_i = \max\left(0, 1 - y_i * \left(\vec{w} \cdot \vec{x}_i + b\right)\right)$$

我们称之为合页损失函数（Hinge Loss Function）

对于软间隔得最优化问题，有：

$$\text{minimize } f(w) = \frac{\|\vec{w}\|^2}{2} + \sum_{i=1}^{s} \varepsilon_i \quad \text{其中 } \varepsilon_i = \max\left(0, 1 - y_i * \left(\vec{w} \cdot \vec{x}_i + b\right)\right)$$

而$\varepsilon_i$等价于：

$$\begin{aligned} & y_i * \left(\vec{w} \cdot \vec{x}_i + b\right) + \varepsilon_i \geq 1 \\ & \varepsilon_i \geq 0 \end{aligned}$$

在实际操作中，我们会对目标函数的损失值部分乘一个非负的参数C，因为我们的目标是求函数的最小值解，C可以让我们控制对损失值$\varepsilon_i$的容忍度。故最终得问题转化为：

$$\text{minimize } f(w) = \frac{\|\vec{w}\|^2}{2} + C\sum_{i=1}^{s} \varepsilon_i \quad \text{其中 } \varepsilon_i = \max\left(0, 1 - y_i * \left(\vec{w} \cdot \vec{x}_i + b\right)\right)$$

# 代码实现

## 算法原理：

优化一系列的α的值，每次选择尽量少的 α 来优化，不断迭代直到函数收敛到最优值。

**优化α:**

假设对 α1,α2 进行优化：在不考虑约束条件的情况下，经过推导可以得出：

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

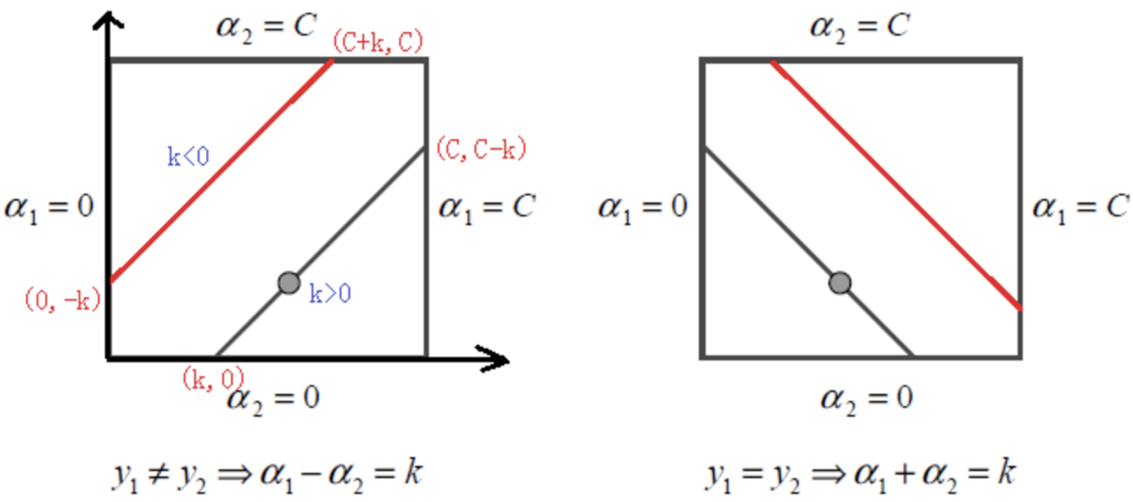
但是存在约束条件

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

故：

$$\begin{aligned} &\alpha_1 y_1 + \alpha_2 y_2 = -\sum_{i=3}^{N} \alpha_i y_i = \zeta \\ &0 \le \alpha_i \le C \end{aligned}$$

此约束为方形约束(Bosk constraint), 在二维平面中我们可以看到这是个限制在方形区域中的直线




(如左图) 当 $y_1 \neq y_2$ 时，线性限制条件可以写成: $\alpha_1 - \alpha_2 = k$，根据 $k$ 的正负可以得到不同 的上下界，因此统一表示成:

- 下界: $L = \max\left(0, \alpha_2^{\text{old}} - \alpha_1^{\text{old}}\right)$
- 上界: $H = \min\left(C, C + \alpha_2^{\text{old}} - \alpha_1^{\text{old}}\right)$

(如右图) 当 $y_1 = y_2$ 时，限制条件可写成: $\alpha_1 + \alpha_2 = k$，上下界表示成:

- 下界: $L = \max\left(0, \alpha_1^{\text{old}} + \alpha_2^{\text{old}} - C\right)$
- 上界: $H = \min\left(C, \alpha_2^{\text{old}} + \alpha_1^{\text{old}}\right)$

根据得到的上下界，我们可以得到修剪后的 $\alpha_2^{\text{new}}$：

$$\alpha_2^{\text{new}} = \begin{cases} H & \alpha_2^{\text{new,unclipped}} > H \\ \alpha_2^{\text{new,unclipped}} & L \le \alpha_2^{\text{new , unclipped}} \le H \\ L & \alpha_2^{\text{new,unclipped}} < L \end{cases}$$

得到了 $\alpha_2^{\text{new}}$ 我们便可以根据 $\alpha_1^{\text{old}} y_1 + \alpha_2^{\text{old}} y_2 = \alpha_1^{\text{new}} y_1 + \alpha_2^{\text{new}} y_2$ 得到 $\alpha_1^{\text{new}}$：

$$\alpha_1^{\text{new}} = \alpha_1^{\text{old}} + y_1 y_2 \left(\alpha_2^{\text{old}} - \alpha_2^{\text{new}}\right)$$

### 更新阈值b

> 当我们更新了一对 $\alpha_i, \alpha_j$ 之后都需要重新计算阈值 $b$ ，因为 $b$ 关系到我们 $f(x)$ 的计算，关系到下次优化的时候误差 $E_i$ 的计算。

为了使得被优化的样本都满足KKT条件：

当 $\alpha_1^{new}$ 不在边界，即 $0 < \alpha_1^{new} < C$ :根据KKT条件可知相应的数据点为支持向量，满足

$$y_1\left(w^T + b\right) = 1$$

两边同时乘上 $y_1$ 得到

$$\sum_{i=1}^{N} \alpha_i y_i K_{i,1} + b = y_1$$

进而得到 $b_1^{new}$ 的值:

$$b_1^{new} = y_1 - \sum_{i=3}^{N} \alpha_i y_i K_{i,1} - \alpha_1^{new} y_1 K_{1,1} - \alpha_2^{new} y_2 K_{2,1}$$

其中上式的前两项可以写成:

$$y_1 - \sum_{i=3}^{N} \alpha_i y_i K_{i,1} = -E_1 + \alpha_1^{\text{old}} y_1 K_{1,1} + \alpha_2^{\text{old}} y_2 K_{2,1} + b^{\text{old}}$$

当 $0 < \alpha_2^{\text{new}} < C$ 时，可以得到$b_2^{new}$:

$$b_2^{\text{new}} = -E_2 - y_1 K_{1,2}\left(\alpha_1^{\text{new}} - \alpha_1^{\text{old}}\right) - y_2 K_{2,2}\left(\alpha_2^{\text{new}} - \alpha_2^{\text{old}}\right) + b^{\text{old}}$$

当 $b_1$ 和 $b_2$ 都有效的时候他们是相等的， 即

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

当两个乘子 $\alpha_1, \alpha_2$ 都在边界上，且 $L \neq H$ 时， $b1, b2$ 之间的值就是和KKT条件一直的阈值。 SMO选择他们的中点作为新的阈值:

$$b^{\text{new}} = \frac{b_1^{new} + b_2^{new}}{2}$$

### 选择变量$\alpha$:

- 在**整个样本集**和**非边界样本集**间进行交替，选择第一个变量$\alpha_1$（外循环）
- 在列表中选择具有 |E1−E2| 的 α2 来近似最大化步长。
- 不断地在两个数据集中来回交替，最终所有的 α 都满足KKT条件的时候，算法中止

## 参考文献：


- 《机器学习实战》Chapter6
- 数之道：支持向量机SVM的本质及其几何解释
- SMO_simple与SMO_Platt的有关资料
- 利用搜索引擎能搜索到的我能明白的所有资料