

# 支持向量机 (SVM) 完整讲义

## 第一部分：拉格朗日函数与极大极小问题

### 四、构造拉格朗日函数

我们将约束写成标准形式  $g_i(\mathbf{w}, b) \leq 0$ 。由于  $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$  等价于  $1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b) \leq 0$ ，我们令：

$$g_i(\mathbf{w}, b) = 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b) \leq 0.$$

为每个约束引入一个拉格朗日乘子  $\alpha_i \geq 0$ ，构造拉格朗日函数：

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^n \alpha_i [1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)].$$

这里  $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_n)^\top$  是乘子向量。

#### 为什么要这样构造？

对于固定的  $(\mathbf{w}, b)$ ，如果某个约束被违反（即  $1 - y_i(\dots) > 0$ ），那么由于  $\alpha_i \geq 0$ ，该项  $\alpha_i(1 - y_i(\dots))$  为正，且可以任意大（如果我们让  $\alpha_i$  很大）。

当我们试图最小化  $L$  时，这样的点不可能成为最小值点，因为我们可以通过增大  $\alpha_i$  使  $L$  趋于无穷大。因此，最小化  $L$  会迫使那些  $\alpha_i$  只对应满足约束的点。这就是拉格朗日函数“吸收”约束的机理。

原始优化问题等价于求解：

$$\min_{\mathbf{w}, b} \max_{\boldsymbol{\alpha} \geq 0} L(\mathbf{w}, b, \boldsymbol{\alpha}).$$

#### 核心原理解析

为什么要先对  $\boldsymbol{\alpha}$  取最大？对于不满足约束的点，最大值是无穷大；对于满足约束的点，最大值就是原目标函数  $f$ （因为  $\alpha_i$  可以取 0 使后项为 0）。

外面再对  $(\mathbf{w}, b)$  取最小，就得到了原问题的最优解。

## 逻辑链路总结

1. **约束转化**: 将 SVM 几何间隔约束转化为标准形式  $g_i(\mathbf{w}, b) \leq 0$ 。
2. **引入乘子**: 构造拉格朗日函数  $L$ ，将约束融入目标函数。
3. **筛选机制**: 利用  $\max_{\alpha \geq 0}$  操作，自动将不满足约束的解赋值为  $+\infty$  从而被排除。
4. **最优求解**: 外层  $\min_{\mathbf{w}, b}$  在可行域内寻找使原目标函数  $\frac{1}{2}\|\mathbf{w}\|^2$  最小的最优解。

## 核心结论

SVM 原始凸优化问题

$$\min_{\mathbf{w}, b} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{s.t.} \quad g_i(\mathbf{w}, b) = 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b) \leq 0$$

等价于拉格朗日函数的极大极小问题:

$$\min_{\mathbf{w}, b} \max_{\alpha \geq 0} L(\mathbf{w}, b, \alpha)$$

其中拉格朗日函数定义为:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^n \alpha_i g_i(\mathbf{w}, b)$$

## 完整推导过程

### 第一步: 定义内层最大值函数

固定参数  $\mathbf{w}, b$ ，将拉格朗日函数关于拉格朗日乘子  $\alpha$  求最大值，定义内层函数:

$$F(\mathbf{w}, b) = \max_{\alpha \geq 0} \left[ \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^n \alpha_i g_i(\mathbf{w}, b) \right]$$

### 第二步: 分情况讨论 $F(\mathbf{w}, b)$ 的取值

**情况1: 满足所有约束  $g_i(\mathbf{w}, b) \leq 0$**

对所有样本  $i$ ，约束条件成立，此时  $\alpha_i \geq 0$ ，因此  $\alpha_i g_i(\mathbf{w}, b) \leq 0$ 。要使拉格朗日函数取得最大值，必须令所有  $\alpha_i = 0$ ，代入得:

$$F(\mathbf{w}, b) = \frac{1}{2}\|\mathbf{w}\|^2$$

情况2: 存在约束被违反  $g_k(\mathbf{w}, b) > 0$

至少存在一个样本  $k$  不满足约束, 此时令  $\alpha_k \rightarrow +\infty$ , 其余  $\alpha_i = 0$ , 则:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 + \alpha_k g_k(\mathbf{w}, b) \rightarrow +\infty$$

因此:

$$F(\mathbf{w}, b) = +\infty$$

第三步: 合并两种情况

综合可得内层最大值函数的分段表达式:

$$F(\mathbf{w}, b) = \begin{cases} \frac{1}{2}\|\mathbf{w}\|^2, & \forall i, g_i(\mathbf{w}, b) \leq 0 \\ +\infty, & \text{存在 } i, g_i(\mathbf{w}, b) > 0 \end{cases}$$

第四步: 求解外层最小值

对  $F(\mathbf{w}, b)$  关于  $\mathbf{w}, b$  求最小值:

$$\min_{\mathbf{w}, b} F(\mathbf{w}, b)$$

取值为  $+\infty$  的点不可能是最小值, 会被自动排除; 仅保留满足所有约束的点, 此时  $F(\mathbf{w}, b) = \frac{1}{2}\|\mathbf{w}\|^2$ 。

因此:

$$\min_{\mathbf{w}, b} F(\mathbf{w}, b) = \min_{\mathbf{w}, b} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{s.t.} \quad g_i(\mathbf{w}, b) \leq 0$$

最终等价性结论

$$\underbrace{\min_{\mathbf{w}, b} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{s.t.} \quad g_i \leq 0}_{\text{SVM 原始优化问题}} \iff \underbrace{\min_{\mathbf{w}, b} \max_{\alpha \geq 0} L(\mathbf{w}, b, \alpha)}_{\text{拉格朗日极大极小问题}}$$

总结

1. 核心逻辑: 通过拉格朗日乘子将带约束优化转化为无约束优化, 约束不满足时目标函数趋于无穷大, 自动过滤无效解; 2. 等价性本质: 满足约束时, 极大极小问题退化为原始目标函数, 完美等价。

## SVM 拉格朗日极大极小问题的两层作用解析

第一点：内层  $\max_{\alpha \geq 0}$  的作用（筛选机制）

情况 1：不满足约束  $(1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b) > 0)$ ，分类间隔不足）

由于  $\alpha_i \geq 0$ ，此时  $\alpha_i \cdot (1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b))$  为正数。若将  $\alpha_i$  调至  $+\infty$ ，则拉格朗日函数  $L(\mathbf{w}, b, \alpha)$  会趋向  $+\infty$ 。后果：外层要对  $(\mathbf{w}, b)$  求最小值，因此这类不满足约束的  $(\mathbf{w}, b)$  会被自动排除。

情况 2：满足约束  $(1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b) \leq 0)$

为使  $L(\mathbf{w}, b, \alpha)$  取最大值，需令所有  $\alpha_i = 0$ （否则会使  $L$  变小）。后果：此时  $L$  的最大值退化为原始目标函数  $\frac{1}{2}\|\mathbf{w}\|^2$ 。

结论：先对  $\alpha$  求最大值，相当于在过程中自动过滤掉所有不满足约束的点，仅保留满足约束的  $(\mathbf{w}, b)$ 。

第二点：外层  $\min_{\mathbf{w}, b}$  的作用（求解最优）

经过内层筛选后，只剩下满足约束的  $(\mathbf{w}, b)$ ，此时  $L$  的最大值为  $\frac{1}{2}\|\mathbf{w}\|^2$ ，因此外层优化问题变为：

$$\min_{\mathbf{w}, b}(\text{满足约束时的 } L) = \min_{\mathbf{w}, b} \frac{1}{2}\|\mathbf{w}\|^2$$

这与 SVM 最初的原始优化目标完全一致。

### 整体逻辑总结

1. 内层 max：充当“过滤器”，将不满足约束的解标记为  $+\infty$ ，自动排除。2. 外层 min：在过滤后的可行解中，求解  $\frac{1}{2}\|\mathbf{w}\|^2$  的最小值，等价于原始带约束优化问题。

## 第二部分：对偶问题与 SMO 算法

### 5.3 对偶问题与 SMO 算法详解

#### 一、SVM 对偶问题形式

对偶问题是关于拉格朗日乘子  $\alpha$  的凸二次规划问题：

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^{\top} \mathbf{x}_j$$

约束条件：

$$\text{subject to } \sum_{i=1}^n \alpha_i y_i = 0, \quad \alpha_i \geq 0, \quad i = 1, \dots, n.$$

说明：最大化一个凹函数等价于最小化其负函数，因此该问题仍属于凸优化范畴，变量个数等于训练样本数  $n$ 。

#### 二、SMO 算法核心思想

SMO (Sequential Minimal Optimization, 序列最小优化) 是专门为求解上述对偶问题设计的高效迭代算法。

**核心思想：**将“一次性求解所有  $\alpha_i$ ”的大规模优化问题，拆解为每次只优化 2 个  $\alpha$  的极小子问题，通过迭代逐步收敛到全局最优解。

#### 三、为什么必须同时优化 2 个变量？

关键约束  $\sum_{i=1}^n \alpha_i y_i = 0$  是一个线性等式约束，这决定了变量更新的最小单元：如果只修改 1 个  $\alpha_i$ ，那么  $\sum \alpha_i y_i$  的值必然改变，约束会被直接破坏。必须同时修改 2 个变量  $\alpha_i, \alpha_j$ ，让它们的变化相互抵消，才能保持  $\sum \alpha_i y_i = 0$  不变。

#### 举例说明

约束可拆分为：

$$\alpha_i y_i + \alpha_j y_j = \text{常数}$$

可以将  $\alpha_i$  增加一点，同时将  $\alpha_j$  减少一点，使两者的加权和保持不变，从而满足等式约束。

#### 四、SMO 算法的整体逻辑

1. **问题分解**: 将高维二次规划拆分为最小的 2 变量子问题。2. **变量选择**: 每次选择 2 个乘子  $\alpha_i, \alpha_j$  进行优化, 其余变量固定。3. **子问题求解**: 在约束  $\alpha_i y_i + \alpha_j y_j = \text{常数}$  及  $\alpha_i, \alpha_j \geq 0$  下, 解析求解使  $W(\alpha)$  最大的新值。4. **迭代收敛**: 重复选择变量并更新, 直到所有  $\alpha$  满足 KKT 条件, 算法收敛。

#### 总结

**对偶问题**: 将原问题转化为关于  $\alpha$  的凸二次规划, 便于引入核函数处理非线性分类。**SMO 算法**: 通过“每次只解 2 个变量”的策略, 在保证约束的前提下大幅降低计算复杂度, 是 SVM 实际应用中最常用的求解方法。

#### SMO 算法: 为何必须优化两个变量详解

##### 一、核心约束拆分

SVM 对偶问题存在关键等式约束:

$$\sum_{k=1}^n \alpha_k y_k = 0$$

在 SMO 算法中, 我们仅优化两个变量  $\alpha_i, \alpha_j$ , 其余所有  $\alpha_k$  ( $k \neq i, j$ ) 固定不变。

##### 第一步: 拆分“可变”与“不变”部分

将求和拆分为两部分:

$$\underbrace{\alpha_i y_i + \alpha_j y_j}_{\text{要优化的两个变量}} + \underbrace{\sum_{k \neq i, j} \alpha_k y_k}_{\text{固定不变的部分}} = 0$$

由于其余  $\alpha_k$  固定,  $\sum_{k \neq i, j} \alpha_k y_k$  是一个常数, 记为  $-\gamma$ :

$$\sum_{k \neq i, j} \alpha_k y_k = -\gamma$$

代回原式并整理:

$$\alpha_i y_i + \alpha_j y_j = \gamma$$

这里  $\gamma$  是常数，在本次迭代中保持不变。

## 二、消元为一元问题

我们的目标是消去一个变量，将二元优化转为一元问题。从约束方程解出  $\alpha_j$ ：

$$\alpha_j y_j = \gamma - \alpha_i y_i$$

两边同时除以  $y_j$  ( $y_j \in \{+1, -1\}$ , 故  $y_j \neq 0$ , 可安全除)：

$$\alpha_j = \frac{\gamma - \alpha_i y_i}{y_j}$$

## 三、直观含义：两个变量“绑定”

你不能单独修改  $\alpha_i$ ，因为一旦改变  $\alpha_i$ ， $\alpha_j$  必须跟着变化，才能保证  $\alpha_i y_i + \alpha_j y_j = \gamma$  不变。这就像天平的两端：左边放  $\alpha_i y_i$  右边放  $\alpha_j y_j$  它们的总和必须是固定的“重量”  $\gamma$  因此  $\alpha_i$  和  $\alpha_j$  是相互决定的：修改一个，另一个会被自动确定。

## 四、具体数值例子

假设条件

$y_i = +1, y_j = -1$  当前  $\alpha_i = 2, \alpha_j = 3$  其他  $\alpha_k$  固定，因此  $\gamma = \alpha_i y_i + \alpha_j y_j = 2 \times 1 + 3 \times (-1) = -1$

更新演示

现在更新  $\alpha_i$ ，令  $\alpha_i^{\text{new}} = 3$ ，则：

$$\alpha_j^{\text{new}} = \frac{\gamma - \alpha_i^{\text{new}} y_i}{y_j} = \frac{-1 - 3 \times 1}{-1} = \frac{-4}{-1} = 4$$

验证约束

$$\alpha_i^{\text{new}} y_i + \alpha_j^{\text{new}} y_j = 3 \times 1 + 4 \times (-1) = -1 = \gamma$$

约束依然满足！

## 总结

1. **约束本质**:  $\sum_{k=1}^n \alpha_k y_k = 0$  是线性等式约束, 仅修改 1 个变量会破坏约束。2. **变量绑定**: 必须同时修改 2 个变量  $\alpha_i, \alpha_j$ , 通过  $\alpha_i y_i + \alpha_j y_j = \gamma$  相互抵消变化, 保持约束不变。3. **算法简化**: 将高维优化问题拆解为最小的 2 变量子问题, 通过解析解高效迭代收敛。

## SMO 算法执行步骤详解

### 步骤 1: 初始化

给所有拉格朗日乘子  $\alpha_i$  赋初值 (通常初始化为全 0)。计算初始偏置项  $b$ 。

### 步骤 2: 启发式选择两个变量 $\alpha_i, \alpha_j$

为了让每次更新都能最大程度改善目标函数、加速收敛, 采用启发式策略选择要优化的变量: 1. **第一个变量**: 优先选择违反 KKT 条件最严重的样本, 例如满足  $\alpha_i > 0$  但  $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \neq 1$  的样本。2. **第二个变量**: 选择与第一个样本“差异最大”的样本, 例如使预测误差差  $|E_i - E_j|$  最大的样本 ( $E_i$  为样本  $i$  的预测误差)。

### 步骤 3: 在约束下更新 $\alpha_i, \alpha_j$ (算法数学核心)

#### 1. 用约束绑定两个变量

由 SVM 对偶问题的等式约束  $\sum_{k=1}^n \alpha_k y_k = 0$ , 固定其他  $\alpha_k$  后可得:

$$\alpha_i y_i + \alpha_j y_j = \gamma \quad (\gamma \text{ 为常数})$$

将  $\alpha_j$  用  $\alpha_i$  表示:

$$\alpha_j = \frac{\gamma - \alpha_i y_i}{y_j}$$

#### 2. 代入目标函数, 转化为一元二次问题

将上式代入对偶目标函数  $W(\boldsymbol{\alpha})$ , 得到只含  $\alpha_i$  的二次函数:

$$W(\alpha_i) = A\alpha_i^2 + B\alpha_i + C$$

对其求导并令导数为 0, 得到未裁剪的最优解  $\alpha_i^{\text{new,unc}}$ 。

### 步骤 3 逻辑总结

1. **变量绑定**: 利用线性等式约束, 将两个变量  $\alpha_i, \alpha_j$  关联, 保证更新时约束不被破坏。2. **问题降维**: 将二元优化问题转化为只含单个变量的一元二次问题, 可通过解析法快速求解。3. **得到候选解**: 求导得到未加约束的最优值, 后续会对其进行范围裁剪, 以满足  $\alpha \geq 0$  的约束。

## SMO 算法步骤三详解: 拆分目标函数

### 前置条件

上一步我们得到两个变量的绑定关系:

$$\alpha_j = \frac{\gamma - \alpha_i y_i}{y_j}$$

SVM 对偶目标函数为:

$$W(\boldsymbol{\alpha}) = \sum_{k=1}^n \alpha_k - \frac{1}{2} \sum_{k=1}^n \sum_{l=1}^n \alpha_k \alpha_l y_k y_l \mathbf{x}_k^\top \mathbf{x}_l$$

我们的目标是将  $\alpha_j$  全部替换为  $\alpha_i$  的表达式, 使  $W$  仅含  $\alpha_i$  这一个变量。

### 第一步: 把 $W(\boldsymbol{\alpha})$ 拆成“含 $\alpha_i, \alpha_j$ ”和“不含”的部分

#### 1. 核心项的通俗理解

我们重点关注第二个双重求和项:

$$\text{核心项} = \sum_{k=1}^n \sum_{l=1}^n \alpha_k \alpha_l y_k y_l \mathbf{x}_k^\top \mathbf{x}_l$$

可以把它想象成一个  $n \times n$  的表格:  $k$  是行号 (遍历所有样本),  $l$  是列号 (遍历所有样本), 每一格  $(k, l)$  里放的数是  $\alpha_k \alpha_l y_k y_l \mathbf{x}_k^\top \mathbf{x}_l$ , 双重求和  $\sum_k \sum_l$  就是把这个  $n \times n$  表格里所有  $n^2$  个格子的数全部加起来。

#### 2. 拆分表格 (按是否含 $i, j$ )

我们把这个大表格按照  $k, l$  是否等于  $i, j$ , 切成 9 个区域, 只保留和  $\alpha_i, \alpha_j$  相关的部分: 只含  $i$  的区域  $k = i, l = i$ , 含  $i$  和  $j$  的区域  $k = i, l = j$ , 只含  $j$  的区域  $k = j, l = j$ , 含  $j$  和  $i$  的区域  $k = j, l = i$ , 完全不含  $i, j$  的区域  $k, l \neq i, j$ , 以及其他杂项, 也就是与  $\alpha_i, \alpha_j$  无关的交叉项。

### 3. 展开后的完整形式

将  $W(\alpha)$  拆开，只关注和  $\alpha_i, \alpha_j$  有关的项，其他项都合并为常数  $C$ ：

$$W(\alpha) = \underbrace{\alpha_i + \alpha_j}_{\text{线性项}} - \frac{1}{2} \left( \underbrace{\alpha_i^2 y_i^2 \mathbf{x}_i^\top \mathbf{x}_i + \alpha_j^2 y_j^2 \mathbf{x}_j^\top \mathbf{x}_j + 2\alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j}_{\text{二次项}} + 2\alpha_i \sum_{k \neq i, j} \alpha_k y_k y_i \mathbf{x}_i^\top \mathbf{x}_k + 2\alpha_j \sum_{k \neq i, j} \alpha_k y_k y_j \mathbf{x}_j^\top \mathbf{x}_k \right)$$

### 4. 关键简化

所有  $k \neq i, j$  的  $\alpha_k$  都固定，所以和  $\alpha_i, \alpha_j$  无关的项，最后都会合并成常数  $C$ 。注意  $y_i^2 = 1, y_j^2 = 1$ ，这会极大简化二次项的计算。

#### 总结

这一步的核心是**降维**：通过拆分双重求和项，把与  $\alpha_i, \alpha_j$  无关的部分全部归为常数，只保留变量相关的线性项和二次项，为下一步代入  $\alpha_j$  并转化为一元二次函数做准备。

## SMO 算法步骤三详解：逐项展开与拆分

### 前置条件

上一步我们得到两个变量的绑定关系：

$$\alpha_j = \frac{\gamma - \alpha_i y_i}{y_j}$$

SVM 对偶目标函数为：

$$W(\alpha) = \sum_{k=1}^n \alpha_k - \frac{1}{2} \sum_{k=1}^n \sum_{l=1}^n \alpha_k \alpha_l y_k y_l \mathbf{x}_k^\top \mathbf{x}_l$$

我们的目标是将  $\alpha_j$  全部替换为  $\alpha_i$  的表达式，使  $W$  仅含  $\alpha_i$  这一个变量。

### 第一步：拆分双重求和项

将整个双重求和拆成 5 块的总和：

$$\sum_{k,l} \dots = \underbrace{(k=i, l=i)}_{\text{第1块}} + \underbrace{(k=i, l=j)}_{\text{第2块}} + \underbrace{(k=j, l=i)}_{\text{第3块}} + \underbrace{(k=j, l=j)}_{\text{第4块}} + \underbrace{\text{其他 } (k, l \neq i, j)}_{\text{第5块}}$$

**第1块:**  $k = i, l = i$

项变为:  $\alpha_i \alpha_i y_i y_i \mathbf{x}_i^\top \mathbf{x}_i$ , 即  $\alpha_i^2 y_i^2 \mathbf{x}_i^\top \mathbf{x}_i$ , 对应公式里的  $\alpha_i^2 y_i^2 \mathbf{x}_i^\top \mathbf{x}_i$ 。

**第2块 + 第3块:**  $k = i, l = j$  和  $k = j, l = i$

第2块为  $\alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$ , 第3块为  $\alpha_j \alpha_i y_j y_i \mathbf{x}_j^\top \mathbf{x}_i$ , 两者相加为  $2 \cdot \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$ , 因为  $\mathbf{x}_i^\top \mathbf{x}_j = \mathbf{x}_j^\top \mathbf{x}_i$ , 对应公式里的  $2\alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j$ 。

**第4块:**  $k = j, l = j$

项变为:  $\alpha_j \alpha_j y_j y_j \mathbf{x}_j^\top \mathbf{x}_j$ , 即  $\alpha_j^2 y_j^2 \mathbf{x}_j^\top \mathbf{x}_j$ , 对应公式里的  $\alpha_j^2 y_j^2 \mathbf{x}_j^\top \mathbf{x}_j$ 。

**第5块:**  $k, l \neq i, j$

这是剩下的所有格子, 范围是  $k, l$  都不等于  $i$  或  $j$ 。因为  $\alpha_k$  和  $\alpha_l$  都固定不变, 所以这一大坨可以看作一个常数, 对应公式里的  $\sum_{k, l \neq i, j} \alpha_k \alpha_l y_k y_l \mathbf{x}_k^\top \mathbf{x}_l$ 。

## 第二步: 合并为 $W(\boldsymbol{\alpha})$ 的结构化形式

将上述拆分代入  $W(\boldsymbol{\alpha})$ , 整理为 4 个部分:

$$W(\boldsymbol{\alpha}) = \underbrace{\alpha_i + \alpha_j}_{\text{块1: 线性变量项}} - \frac{1}{2} \left( \underbrace{\alpha_i^2 \cdots + \alpha_j^2 \cdots + 2\alpha_i \alpha_j \cdots}_{\text{块2: 二次变量项}} + \underbrace{2\alpha_i \sum_{k \neq i, j} \cdots + 2\alpha_j \sum_{k \neq i, j} \cdots}_{\text{块3: 含 } \alpha_i, \alpha_j \text{ 的线性项}} \right) + \underbrace{\left( \sum_{k \neq i, j} \alpha_k - \frac{1}{2} \sum_{k, l \neq i, j} \right)}_{\text{块4: 纯常数项}}$$

### 说明

块1是只含  $\alpha_i, \alpha_j$  的一次项, 块2是只含  $\alpha_i, \alpha_j$  的二次项, 块3是含  $\alpha_i, \alpha_j$  与其他固定  $\alpha_k$  的交叉线性项, 块4是完全不含  $\alpha_i, \alpha_j$  的纯常数项。

### 总结

这一步的核心是把复杂的双重求和拆解成变量相关项和常数项, 为后续代入  $\alpha_j = \frac{\gamma - \alpha_i y_i}{y_j}$ 、将  $W(\boldsymbol{\alpha})$  化简为只含  $\alpha_i$  的一元二次函数做准备。

## 步骤三详解：代入化简与解析求解（数学核心）

### 前置准备

已知两个变量的绑定关系（由等式约束  $\sum \alpha_k y_k = 0$  推导）：

$$\alpha_j = \frac{\gamma - \alpha_i y_i}{y_j}$$

我们的目标是将  $W(\alpha)$  转化为只含  $\alpha_i$  的一元二次函数，并求其极值。

### 第二步：把 $\alpha_j$ 用 $\alpha_i$ 代入

#### (1) 先处理线性项 $\alpha_i + \alpha_j$

将绑定关系代入线性项：

$$\alpha_i + \alpha_j = \alpha_i + \frac{\gamma - \alpha_i y_i}{y_j}$$

这是一个关于  $\alpha_i$  的一次项。

#### (2) 再处理二次项

二次项包含多种成分，代入  $\alpha_j$  后全部转化为  $\alpha_i$  的多项式： $\alpha_i^2$  项自身保留， $\alpha_j^2$  项代入后变为  $\left(\frac{\gamma - \alpha_i y_i}{y_j}\right)^2$ ，即  $\alpha_i^2$  项， $\alpha_i \alpha_j$  项代入后变为  $\alpha_i \cdot \frac{\gamma - \alpha_i y_i}{y_j}$ ，即  $\alpha_i^2$  项，交叉项为  $\alpha_i$  与其他固定  $\alpha_k$  的交叉项，化简为  $\alpha_i$  的一次项。

#### (3) 合并系数

将所有含  $\alpha_i^2$  的项合并得到系数  $A$ ；将所有含  $\alpha_i$  的一次项合并得到系数  $B$ ；剩余不含  $\alpha_i$  的常数项合并为  $C$ 。最终，目标函数转化为标准的一元二次函数：

$$W(\alpha_i) = A\alpha_i^2 + B\alpha_i + C$$

## 第三步：一元二次函数求极值（数学核心）

### 函数特性分析

我们要最大化  $W(\alpha_i)$ 。由于 SVM 对偶问题是凸优化问题（目标函数为凹），因此二次项系数  $A < 0$ ，函数开口向下。结论为开口向下的抛物线，

顶点即为最大值点。

### 解析求解

对  $W(\alpha_i)$  求导并令导数为 0:

$$\frac{dW}{d\alpha_i} = 2A\alpha_i + B$$

令  $\frac{dW}{d\alpha_i} = 0$ , 解出  $\alpha_i$ :

$$2A\alpha_i + B = 0 \implies \alpha_i = -\frac{B}{2A}$$

### 结果定义

这个解被称为**未裁剪的最优解**  $\alpha_i^{\text{new,unc}}$ 。它是“还未考虑  $\alpha_i \geq 0, \alpha_j \geq 0$  箱型约束”时的纯数学极值点。后续操作是在得到  $\alpha_i^{\text{new,unc}}$  后, 根据  $\alpha$  的取值范围 ( $0 \leq \alpha \leq C$ ) 进行**裁剪 (Clipping)**, 才能得到最终的可行解。

### 总结

这一步是 SMO 算法的**数学核心**: 一是降维, 利用约束将二元问题转化为一元二次函数; 二是求极值, 利用微积分知识直接求解最大值点 (解析解), 避免了复杂的数值迭代, 极大提升了计算效率。

## SMO 算法步骤 3.3 & 步骤 4 详解

### 步骤 3.3: 裁剪到合法区间 & 计算 $\alpha_j^{\text{new}}$

裁剪到合法区间  $[L, H]$

由于  $\alpha_i \geq 0, \alpha_j \geq 0$  (软间隔还需满足  $\alpha_i \leq C, \alpha_j \leq C$ ), 未裁剪的最优解  $\alpha_i^{\text{new,unc}}$  必须被限制在区间  $[L, H]$  内。

若  $y_i \neq y_j$ :

$$L = \max(0, \alpha_j^{\text{old}} - \alpha_i^{\text{old}}), \quad H = \min(C, C + \alpha_j^{\text{old}} - \alpha_i^{\text{old}})$$

硬间隔时  $H = +\infty$ 。

若  $y_i = y_j$ :

$$L = \max(0, \alpha_i^{\text{old}} + \alpha_j^{\text{old}} - C), \quad H = \min(C, \alpha_i^{\text{old}} + \alpha_j^{\text{old}})$$

最终裁剪为:

$$\alpha_i^{\text{new}} = \text{clip}(\alpha_i^{\text{new,unc}}, L, H)$$

即:

$$\alpha_i^{\text{new}} = \begin{cases} L, & \alpha_i^{\text{new,unc}} < L \\ \alpha_i^{\text{new,unc}}, & L \leq \alpha_i^{\text{new,unc}} \leq H \\ H, & \alpha_i^{\text{new,unc}} > H \end{cases}$$

由  $\alpha_i^{\text{new}}$  计算  $\alpha_j^{\text{new}}$

根据变量绑定关系  $\alpha_i y_i + \alpha_j y_j = \gamma$  (常数), 可由  $\alpha_i^{\text{new}}$  直接计算  $\alpha_j^{\text{new}}$ :

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

#### 步骤 4: 更新偏置 $b$

用更新后的  $\alpha_i^{\text{new}}, \alpha_j^{\text{new}}$  计算新的偏置  $b$ , 保证支持向量满足  $y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1$ 。

计算候选  $b$

$$b_1 = y_i - \sum_{k=1}^n \alpha_k y_k \mathbf{x}_k^\top \mathbf{x}_i$$
$$b_2 = y_j - \sum_{k=1}^n \alpha_k y_k \mathbf{x}_k^\top \mathbf{x}_j$$

选择最终  $b$

若  $\alpha_i^{\text{new}} \in (0, C)$ , 则  $b = b_1$ ; 若  $\alpha_j^{\text{new}} \in (0, C)$ , 则  $b = b_2$ ; 若  $\alpha_i^{\text{new}}, \alpha_j^{\text{new}}$  都在边界上 (0 或  $C$ ), 则取平均值:

$$b = \frac{b_1 + b_2}{2}$$

## 总结

一是裁剪约束，通过区间  $[L, H]$  保证  $\alpha_i, \alpha_j$  满足非负性与软间隔上限，得到可行解；二是变量更新，利用绑定关系快速计算  $\alpha_j^{\text{new}}$ ，无需重复求解二次函数；三是偏置更新，根据更新后的  $\alpha$  重新计算  $b$ ，确保分类超平面与支持向量的几何关系正确。

## 为何要裁剪，怎么裁剪？

### 一、为什么要裁剪？——区间 $[L, H]$ 的由来

SMO 处理的是软间隔 SVM（带惩罚参数  $C$ ），必须满足两个核心非负约束：一是自身非负  $\alpha_i \geq 0$ ，二是同伴非负  $\alpha_j \geq 0$ 。这两个约束共同决定了  $\alpha_i$  的活动范围  $[L, H]$ ，未裁剪的极值解  $\alpha_i^{\text{new,unc}}$  必须落在这个区间内才是可行解。

#### 分情况讨论（由 $y_i, y_j$ 的符号决定）

类别标签  $y$  只有  $\pm 1$  两种取值，因此分两类讨论：

情况 A:  $y_i \neq y_j$ （异号，如  $+1$  和  $-1$ ），数学约束关系为  $\alpha_i y_i + \alpha_j y_j = \text{常数}$ ，因为  $y_i, y_j$  符号相反，等价于  $\alpha_i - \alpha_j = \text{常数}$ 。变化关系为  $\alpha_i$  和  $\alpha_j$  必须同方向变化（同涨或同跌）。下限  $L$  为不能小于  $0$ （非负约束），也不能小于  $\alpha_j - \alpha_i$ （保证  $\alpha_j \geq 0$ ），即  $L = \max(0, \alpha_j - \alpha_i)$ 。上限  $H$  为不能大于  $C$ （软间隔惩罚），也不能大于  $C + \alpha_j - \alpha_i$ （保证  $\alpha_j \leq C$ ），即  $H = \min(C, C + \alpha_j - \alpha_i)$ 。

情况 B:  $y_i = y_j$ （同号，如都为  $+1$ ），数学约束关系为  $\alpha_i y_i + \alpha_j y_j = \text{常数}$ ，因为  $y_i, y_j$  同号，等价于  $\alpha_i + \alpha_j = \text{常数}$ 。变化关系为  $\alpha_i$  和  $\alpha_j$  必须反方向变化（一个涨，另一个必须跌）。下限  $L$  为不能小于  $0$ （非负约束），也不能小于  $\alpha_i + \alpha_j - C$ （保证  $\alpha_j \leq C$ ），即  $L = \max(0, \alpha_i + \alpha_j - C)$ 。上限  $H$  为不能大于  $C$ （软间隔惩罚），也不能大于  $\alpha_i + \alpha_j$ （保证  $\alpha_j \geq 0$ ），即  $H = \min(C, \alpha_i + \alpha_j)$ 。

## 二、怎么裁剪？——clip 操作

将未裁剪的解  $\alpha_i^{\text{new,unc}}$  限制在区间  $[L, H]$  内：

$$\alpha_i^{\text{new}} = \text{clip}(\alpha_i^{\text{new,unc}}, L, H)$$

具体规则为：若  $\alpha_i^{\text{new,unc}} > H$  则取  $H$ ，若  $\alpha_i^{\text{new,unc}} < L$  则取  $L$ ，若  $L \leq \alpha_i^{\text{new,unc}} \leq H$  则保持原值。

## 三、由 $\alpha_i^{\text{new}}$ 计算 $\alpha_j^{\text{new}}$

由约束  $\alpha_i y_i + \alpha_j y_j = \text{常数}$ ，对变化量分析可得：

$$y_i (\alpha_i^{\text{new}} - \alpha_i^{\text{old}}) + y_j (\alpha_j^{\text{new}} - \alpha_j^{\text{old}}) = 0$$

整理得：

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

### 直观理解

$y_i y_j$  取值为  $\pm 1$ ，决定了  $\alpha_j$  的调整方向，同号时调整方向相同，异号时调整方向相反。 $\alpha_i^{\text{old}} - \alpha_i^{\text{new}}$  为  $\alpha_i$  的调整量，整体逻辑为  $\alpha_j$  的新值等于旧值加上符号系数乘以  $i$  的变化量，保证了  $\sum_{k=1}^n \alpha_k y_k = 0$  这个约束在迭代中始终成立。

### 总结

一是裁剪目的为保证  $\alpha_i, \alpha_j$  满足非负性与软间隔上限  $C$ ，得到可行解；二是区间来源为由  $y_i, y_j$  的符号和  $\alpha_i, \alpha_j$  的约束共同推导得到  $[L, H]$ ；三是更新逻辑为先裁剪  $\alpha_i$ ，再通过约束关系快速计算  $\alpha_j$ ，保证等式约束不变。

## SMO 算法步骤 5 与求解结果总结

### 步骤 5：迭代直到收敛

重复执行步骤 2-4（选择变量  $\rightarrow$  更新  $\alpha_i, \alpha_j \rightarrow$  更新偏置  $b$ ），直到满足以下收敛条件之一：所有  $\alpha_i$  都满足 KKT 条件（最优性条件），或对偶目标函数  $W(\alpha)$  的变化量小于设定阈值（例如  $10^{-3}$ ）。

## 求解后得到什么？

### 1. 最优 $\alpha^*$ ：只有支持向量“活着”

大部分样本的  $\alpha_i^* = 0$ ，对模型没有任何贡献，可以直接忽略；少数样本的  $\alpha_i^* > 0$ ，这些是支持向量（Support Vectors），它们决定了分类边界。

### 2. 权重 $w^*$ ：由支持向量线性组合而成

$$\mathbf{w}^* = \sum_{i=1}^n \alpha_i^* y_i \mathbf{x}_i$$

$\mathbf{w}^*$  是分类超平面的法向量，决定了分类的“方向”；只有  $\alpha_i^* > 0$  的样本才会被加进求和式，非支持向量（ $\alpha_i^* = 0$ ）会直接从表达式中消失，因此  $\mathbf{w}^*$  本质是支持向量的加权和。

### 3. 偏置 $b^*$ ：让分类间隔“对齐”

偏置  $b^*$  由步骤 4 中更新后的  $\alpha_i, \alpha_j$  计算得到（即候选值  $b_1, b_2$ ），作用是保证支持向量满足  $y_i(\mathbf{w}^{*\top} \mathbf{x}_i + b^*) = 1$ ，即刚好落在间隔边界上。 $\mathbf{w}^*$  决定了分类超平面的方向， $b^*$  决定了超平面的位置，两者共同确定完整的分类超平面。

### 4. 决策函数

最终的分类决策函数为：

$$f(\mathbf{x}) = \text{sign} \left( \sum_{i=1}^n \alpha_i^* y_i \mathbf{x}_i^\top \mathbf{x} + b^* \right)$$

输入新样本  $\mathbf{x}$ ，通过计算符号判断其类别。

## 总结

一是收敛判断，通过 KKT 条件或目标函数变化量，确保迭代到最优解；二是解的稀疏性，只有支持向量对模型有贡献，大幅降低了预测时的计算量；三是模型表达， $\mathbf{w}^*$  和  $b^*$  共同构成分类超平面，决策函数可直接用于新样本分类。

## 八、决策函数

对于新样本  $\mathbf{x}$ ，SVM 的最终分类决策由以下函数给出：

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \mathbf{x} + b) = \text{sign}\left(\sum_{i=1}^n \alpha_i y_i \mathbf{x}_i^\top \mathbf{x} + b\right)$$

### 符号函数 $\text{sign}(z)$ 定义

若  $z > 0$ ，输出  $+1$ （归为正类）；若  $z < 0$ ，输出  $-1$ （归为负类）；若  $z = 0$ ，可任意定义（通常归为正类或负类均可）。

### 核心含义

$\mathbf{w}^\top \mathbf{x} + b$  是样本  $\mathbf{x}$  到分类超平面的有符号距离，符号决定类别，绝对值代表置信度。求和式中只有  $\alpha_i > 0$  的支持向量会参与计算，非支持向量（ $\alpha_i = 0$ ）会被自动忽略，因此预测时计算效率很高。若使用核技巧，可将  $\mathbf{x}_i^\top \mathbf{x}$  替换为核函数  $k(\mathbf{x}_i, \mathbf{x})$ ，直接处理非线性分类问题：

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^n \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b\right)$$

### 总结

决策函数是 SVM 模型的最终输出接口：输入为新样本特征  $\mathbf{x}$ ，输出为类别标签  $\pm 1$ ，本质是基于支持向量的加权核函数求和加偏置，再通过符号函数映射为类别。