最大分隔面算法中的对偶问题推导

DeepSeek, and A Human Curator.

2026-03-16

目录

1 原始问题 (Primal Problem)

支持向量机（SVM）的核心目标是寻找一个超平面 $w^T x + b = 0$，在保证正确分类训练样本的同时，使分类间隔（Margin）最大化。这一目标可以转化为如下的凸二次规划问题：

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|^2 \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1, \quad i = 1, \ldots, N \end{aligned} \tag{1}$$

其中，w 是法向量，b 是偏置，$y_i \in \{+1, -1\}$ 是样本标签。该问题的目标是最小化 $\|w\|^2/2$，等价于最大化几何间隔 $2/\|w\|$。

2 通过拉格朗日函数，把原问题转变为无约束优化问题

为了将有约束优化问题转化为无约束优化问题，我们引入拉格朗日乘子 $\alpha_i \geq 0$。

2.1 函数定义

定义拉格朗日函数 $L(w, b, \alpha)$ 如下：

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i \left[y_i(w^T x_i + b) - 1 \right] \tag{2}$$

该函数由原始目标函数与约束项的加权组合构成。原始问题可以表示为如下的 **min-max** 形式：

$$\min_{w,b} \max_{\alpha_i \geq 0} L(w, b, \alpha) \tag{3}$$

2.2 符号逻辑解释：为何是减号?

在构造拉格朗日函数时，约束项前的负号并非随意设定的，它背后的逻辑在于通过“惩罚”机制确保约束条件的履行：

1. **约束条件的标准化**：SVM 的原始约束为 $y_i(w^T x_i + b) - 1 \geq 0$。我们将这一项记作 $g_i(w, b)$。

2. **违反约束的代价**：如果某个样本 i 违反了约束（即 $g_i(w, b) < 0$），那么在内部的极大化操作 $\max_{\alpha_i \geq 0} L$ 中，由于 g_i 是负数，前面带有减号的项 $-\alpha_i g_i$ 就变成了正数。此时，只要让 α_i 趋向于 $+\infty$，整个函数值 L 就会趋向于 $+\infty$。

3. **强制执行**：外部的最小化操作 $\min_{w,b}$ 绝不会选择一个使结果为无穷大的解。因此，这种结构强制要求 w 和 b 必须满足所有 $g_i(w, b) \geq 0$ 的条件。此时 $\max_{\alpha_i \geq 0}[-\alpha_i g_i]$ 的最大值只能在 $\alpha_i g_i = 0$ 时取得，即 0。

简而言之，**减号配合非负乘子** α_i 构成了一个强大的“惩罚项”，确保了优化过程在合法的可行域内进行。

2.3 习题训练

通过以下两个习题，读者可以深刻体会 $\min_w \max_\alpha$ 架构是如何通过“惩罚机制”自动执行约束条件的。

题目 1：数值推导

设优化问题为：

$$\begin{aligned} \min_{x} \quad & f(x) = x^2 \\ \text{s.t.} \quad & x - 1 \leq 0 \end{aligned} \tag{4}$$

请通过 $\min_x \max_{\alpha \geq 0} L(x, \alpha)$ 的框架进行分析。

【答案在文末】

题目 2：抽象符号

设通用优化问题为：

$$\begin{aligned} \min_{w} \quad & f(w) \\ \text{s.t.} \quad & g(w) \leq 0 \end{aligned} \tag{5}$$

请描述其拉格朗日函数 $L(w, \alpha) = f(w) + \alpha g(w)$ 在 $\min_w \max_{\alpha \geq 0}$ 框架下的运行逻辑。

【答案在文末】

3 原问题的“对偶问题”求解，及其对应的数学背景

利用拉格朗日对偶性，通过交换最小化和最大化的顺序，我们得到对偶问题的 **max-min** 形式：

$$\max_{\alpha_i \geq 0} \min_{w,b} L(w, b, \alpha) = \max_{\alpha_i \geq 0} \min_{w,b} \left(\frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(w^T x_i + b) - 1] \right) \tag{6}$$

3.1 求解步骤与梯度计算

1. **内部极小化**：首先固定 α，对 w 和 b 求极小值。根据多元函数求极值的必要条件，令 L 对 w 和 b 的偏导数（梯度）为 0：

   - **对 w 求梯度**：

$$\nabla_w L(w, b, \alpha) = w - \sum_{i=1}^{N} \alpha_i y_i x_i = 0 \implies w = \sum_{i=1}^{N} \alpha_i y_i x_i \tag{7}$$

   该式揭示了最优权重向量 w 本质上是样本点 x_i 的线性组合。

- **对 b 求偏导**：

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{N} \alpha_i y_i = 0 \implies \sum_{i=1}^{N} \alpha_i y_i = 0 \tag{8}$$

这是一个关于拉格朗日乘子的等式约束。

2. **代入消元过程**：将上述两个结论代回拉格朗日函数 L 以消去 w 和 b：

$$\begin{aligned} L(w, b, \alpha) &= \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i y_i w^T x_i - \sum_{i=1}^{N} \alpha_i y_i b + \sum_{i=1}^{N} \alpha_i \\ &= \frac{1}{2} w^T \left(\sum_{j=1}^{N} \alpha_j y_j x_j \right) - w^T \left(\sum_{i=1}^{N} \alpha_i y_i x_i \right) - b \underbrace{\sum_{i=1}^{N} \alpha_i y_i}_{=0} + \sum_{i=1}^{N} \alpha_i \\ &= \sum_{i=1}^{N} \alpha_i - \frac{1}{2} w^T w \\ &= \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \left(\sum_{i=1}^{N} \alpha_i y_i x_i \right)^T \left(\sum_{j=1}^{N} \alpha_j y_j x_j \right) \end{aligned} \tag{9}$$

3. **外部极大化**：整理后得到仅包含对偶变量 α 的最终对偶目标函数：

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \\ \text{s.t.} \quad & \sum_{i=1}^{N} \alpha_i y_i = 0 \\ & \alpha_i \geq 0, \quad i = 1, \ldots, N \end{aligned} \tag{10}$$

3.2 算法实施：SMO 算法 (Sequential Minimal Optimization)

在 SVM 的对偶问题中，目标函数是一个带有线性等式约束的二次规划问题。由于样本量 N 可能很大，直接调用通用的二次规划求解器计算开销巨大。因此，John Platt 提出了 SMO（Sequential Minimal Optimization，序列最小优化）算法。

求解上述对偶问题，本质上是处理一个带有变量 α 且包含约束 $\sum \alpha_i y_i = 0$ 和 $\alpha_i \geq 0$ 的二次规划问题。SMO 算法的核心思想是：**化整为零，逐对优化**。

1. **基本思路**：在每一步优化中，SMO 仅选择两个变量 α_1 和 α_2，并固定其他所有变量 $\alpha_i (i > 2)$。由于等式约束 $\sum_{i=1}^{N} \alpha_i y_i = 0$ 的存在，一旦其他变量被固定，α_1 与 α_2 之间就存在如下关系：

$$\alpha_1 y_1 + \alpha_2 y_2 = -\sum_{i=3}^{N} \alpha_i y_i = \text{Constant} \tag{11}$$

这意味着，只要确定了 α_2 的值，α_1 也会随之确定。因此，原有的 N 变量优化问题被简化为了一个单变量优化问题。

2. **子问题的解析解**：由于目标函数对 α_2 是二次的，我们可以直接求出其解析解（闭式解）。计算过程通常包括：

   - 根据当前的预测误差计算未剪切的更新值 $\alpha_2^{new,unc}$。
   - 考虑约束 $0 \leq \alpha_i$（以及带软间隔时的 $\alpha_i \leq C$），将更新值剪切到合法的矩形区域边界内。
   - 根据更新后的 α_2 回代计算 α_1。

3. **变量的选择启发式 (Heuristics)**：为了加快收敛速度，SMO 并不随机选择变量，而是采用启发式策略：

   - **第一变量选择**：优先选择违反 KKT 条件最严重的样本点所对应的 α_i。
   - **第二变量选择**：选择能使目标函数有足够大变化的 α_j，通常通过最大化 $|E_i - E_j|$（预测误差之差）来选取。

结论：SMO 算法将复杂的全局优化转化为了一系列极简单的两变量优化子问题，极大地提升了 SVM 处理大规模数据集的能力。

3.3 SMO 算法联动更新算例

为了直观理解变量间的联动关系，我们考虑一个包含 3 个样本的简易系统。设标签分别为 $y_1 = +1, y_2 = -1, y_3 = +1$，当前的拉格朗日乘子初值为 $\alpha_1 = 0.5, \alpha_2 = 0.2, \alpha_3 = 0.3$。

1. **固定变量与建立约束**
   在 SMO 的单步迭代中，我们固定 $\alpha_3 = 0.3$，仅对 α_1, α_2 进行优化。为了满足全局等式约束 $\sum_{i=1}^{N} \alpha_i y_i = 0$，必须满足：

$$\alpha_1 y_1 + \alpha_2 y_2 = -\alpha_3 y_3 \tag{12}$$

   代入已知数值：

$$\alpha_1(1) + \alpha_2(-1) = -(0.3)(1) \implies \alpha_1 - \alpha_2 = -0.3 \tag{13}$$

   由此可得 α_1 是关于 α_2 的函数：$\alpha_1 = \alpha_2 - 0.3$。

2. **联动更新过程**
   假设通过解析求解子问题，发现目标函数在 $\alpha_2^{new} = 0.6$ 时取得极值。为了不破坏等式约束，α_1 必须“被动”调整：

$$\alpha_1^{new} = \alpha_2^{new} - 0.3 = 0.6 - 0.3 = 0.3 \tag{14}$$

   此时，更新后的变量对为 $(\alpha_1, \alpha_2) = (0.3, 0.6)$。

3. **边界剪切 (Clipping)**
   由于存在 $0 \le \alpha_i \le C$ 的约束，若计算出的 α_1^{new} 或 α_2^{new} 超出此范围，则需要将其映射回合法区间。这种联动确保了在每一步局部优化中，系统始终在可行域的“等式长廊”上滑动。

逻辑总结：在本例中，y_1 与 y_2 异号，因此 α_1 与 α_2 呈现同步增减的关系；若两者同号，则会呈现此消彼长的反向变动关系。

3.4 强对偶性的数学定义与条件

通常情况下，对偶问题的解只是原始问题解的一个下界（弱对偶）。但在 SVM 中，由于满足以下两个关键性质，两者的解完全相等，即满足 **强对偶性 (Strong Duality)**：

- **凸性 (Convexity)**：目标函数和约束空间都是凸的，这意味着问题不存在局部最优陷阱，形状如同一个规则的碗。
- **Slater 条件**：只要训练数据在数学上是线性可分的（即存在至少一个超平面能完美分开两类），则强对偶性成立。

得益于强对偶性，我们通过求解计算上更简便的对偶变量 α，即可获得原始问题的最优分类器。

通常情况下，对偶问题的最优值总是小于或等于原始问题的最优值（即弱对偶性）。但在 SVM 中，由于问题具备良好的数学性质，两者可以取到相同的值，即满足 **强对偶性 (Strong Duality)**。

为了严谨地说明这一点，我们需要给出以下两个核心条件的数学定义：

3.4.1 凸性 (Convexity)

在最优化理论中，一个问题被称为凸优化问题，需要满足目标函数是凸函数，且约束集合是凸集。

1. **凸函数定义**：函数 $f(w)$ 被称为凸函数，当且仅当对于定义域内的任意两点 w_1, w_2 及 $\lambda \in [0,1]$，均满足：

$$f(\lambda w_1 + (1-\lambda)w_2) \le \lambda f(w_1) + (1-\lambda) f(w_2) \tag{15}$$

   在 SVM 中，$f(w) = \frac{1}{2}\|w\|^2$ 是一个开口向上的二次函数，其二阶导数恒大于 0，因此是严格凸函数。

2. **凸集定义**：集合 C 被称为凸集，意味着集合内任意两点的连线仍在该集合内。在 SVM 中，线性约束 $y_i(w^T x_i + b) \ge 1$ 定义的是一系列半空间的交集，这在数学上保证了可行域是一个凸集。

3.4.2 Slater 条件 (Slater's Condition)

Slater 条件是确保强对偶性成立的一个充分条件。对于一个凸优化问题，如果存在一个点 (w, b) 使得所有的不等式约束都严格成立，则强对偶性成立。

严格数学表述：对于原始问题的约束 $g_i(w, b) = 1 - y_i(w^T x_i + b) \leq 0$，如果存在至少一组参数 (w, b)，使得：

$$g_i(w, b) < 0, \quad \forall i = 1, \ldots, N \tag{16}$$

则称该问题满足 Slater 条件。

直观物理意义：这意味着训练数据集必须是 **线性可分**的。即存在一个超平面，不仅能把两类样本分开，还能让所有样本点都不落在最大分隔面的边缘上（即所有点距离超平面的函数间隔都严格大于 1）。

3.4.3 结论

由于 SVM 的目标函数 $\frac{1}{2}\|w\|^2$ 是凸的，且在数据线性可分时满足 Slater 条件，因此根据拉格朗日对偶理论，该问题满足：

$$d^* = \max_{\alpha \geq 0} \min_{w,b} L(w, b, \alpha) = \min_{w,b} \max_{\alpha \geq 0} L(w, b, \alpha) = p^* \tag{17}$$

其中 d^* 为对偶问题的最优值，p^* 为原始问题的最优值。这使得我们通过求解 α 获得的解与直接求解 w, b 获得的结果完全一致。

4 从对偶问题中拉格朗日函数的正则项谈开去

在深入推导了对偶过程后，我们回过头审视拉格朗日函数 $L(w, b, \alpha)$。你会发现，这种“目标函数 + 约束项”的结构，与机器学习中常见的“损失函数 + 正则项”有着异曲同工之妙。

4.1 正则项的定义

在最优化问题中，**正则项 (Regularization Term)** 通常是指加在原始目标函数上的额外项，用于对模型的复杂度或参数范数进行限制。其通用形式为：

$$\min_{\theta} \text{Loss}(\theta) + \lambda \Omega(\theta) \tag{18}$$

其中 $\Omega(\theta)$ 即为正则项，λ 为控制限制强度的超参数。

4.2 约束条件即正则项

在本讲义的最大分隔面算法中，我们可以观察到：

- 原始目标 $\frac{1}{2}\|w\|^2$ 实际上是对模型平滑度（间隔最大化）的正则约束。
- 而拉格朗日函数中的 $\sum \alpha_i[y_i(w^T x_i + b) - 1]$ 这一部分，在逻辑上充当了“正则项”的角色。

不同于普通的 L_1 或 L_2 正则，这里的“正则项”是由约束条件演化而来的。乘子 α_i 扮演了动态调节权重的系数：对于那些容易分类错误的点，系统会自动调大 α_i，增加该约束在总目标中的权重。

4.3 正则项的通常作用

正则项在数据挖掘与机器学习中主要有以下三大作用：

1. **防止过拟合**：通过惩罚过大的权重参数，限制模型的复杂度。
2. **数值稳定性**：使原本可能不适定（Ill-posed）的问题变得有唯一解。
3. **引入先验知识**：例如 L_1 正则项（Lasso）可以诱导稀疏性，这与 SVM 中只有少数点（支持向量）起作用的稀疏特性在哲学上是一致的。

4.4 拉格朗日解法的通用性与注意细节

拉格朗日乘子法是一种处理正则问题的通用框架，但在实际使用时，需要注意以下细节：

- **互补松弛性 (Complementary Slackness)**：这是解法的灵魂。在最优解处，必须满足 $\alpha_i[y_i(w^T x_i + b) - 1] = 0$。这意味着要么约束刚好取到边界（支持向量），要么该约束的权重 α_i 为 0。
- **参数敏感性**：在带软间隔的 SVM 中，正则系数 C（对应 α 的上界）直接决定了模型对噪声的容忍度，设置不当会导致欠拟合或过拟合。
- **计算开销**：虽然对偶转化很优雅，但如果样本量 N 极大，计算 $\sum\sum \alpha_i \alpha_j$ 的核矩阵将面临内存挑战。

5 一份给《数据挖掘》课堂的特别提醒

亲爱的同学们，虽然刚才这段推导看起来逻辑丝滑、公式如画，但这主要是因为 LLM 特别擅长模仿数学专家的“优雅姿态”。请务必记住：LLM 可以是你的超级助教，但它偶尔也会在微小的符号或逻辑推导上“一本正经地胡说八道”。

在《数据挖掘》的学习旅程中，LLM 生成的代码和文档只是起点而非终点。请像检查代码 Bug 一样去复核它的每一个 $\sum$ 和 α。最硬核的智力成果依然凝结在人类编写的教材中——那才是不会因为断网或提示词干扰而动摇的真理。总之，**要做 LLM 的驾驭者，不要做它的复读机**。毕竟，期末考试时，坐在考场里的可不是那个会跳 Latex 代码的对话框！

5.1 习题答案

【习题 1 答案解析】:

1. 构造拉格朗日函数：$L(x, \alpha) = x^2 + \alpha(x - 1)$，其中 $\alpha \geq 0$。

2. 分析内部极大化 $\max_{\alpha \geq 0} L(x, \alpha)$：

- 若 $x > 1$（违反约束）：则 $(x - 1) > 0$，令 $\alpha \to +\infty$，则 $L \to +\infty$。
- 若 $x \leq 1$（满足约束）：则 $(x - 1) \leq 0$，为了取得极大值，最优的 $\alpha = 0$，此时 $L = x^2$。

3. 外部极小化：$\min_x \left(\max_{\alpha \geq 0} L \right)$ 实际上是在 $\{x^2 \mid x \leq 1\}$ 与 $\{+\infty \mid x > 1\}$ 之间寻优。显然，最优解为 $x = 0$，此时目标函数值为 0。

【习题 2 答案解析】:

1. **内部算子：** 定义 $\theta_P(w) = \max_{\alpha \geq 0} [f(w) + \alpha g(w)]$。

- 如果 w 落在可行域外（$g(w) > 0$），α 会作为“惩罚因子”无限放大该违规行为，使得 $\theta_P(w) = +\infty$。
- 如果 w 满足约束（$g(w) \leq 0$），由于 $\alpha \geq 0$，项 $\alpha g(w) \leq 0$。为了最大化该式，只能取 $\alpha g(w) = 0$，此时 $\theta_P(w) = f(w)$。

2. **外部算子：** $\min_w \theta_P(w)$。由于不可行域的函数值已被内部算子通过 α 强行拉升至 $+\infty$，外部的极小化算子会被迫在满足 $g(w) \leq 0$ 的“平原”区域内寻找 $f(w)$ 的最低点。