支持向量机

support vector machine
algorithm verify

我真的不懂忧郁

Delft University of Technology

TU Delft

支持向量机

support vector machine algorithm verify

by

我真的不懂忧郁

Student Name	Student Number
First Surname	1234567

Instructor:	I. Surname
Teaching Assistant:	I. Surname
Project Duration:	Month, Year - Month, Year
Faculty:	Faculty of Aerospace Engineering, Delft

TU Delft

Preface

A preface...

我真的不懂忧郁

Delft, June 2024

Summary

A summary...

目录

Nomenclature

If a nomenclature is required, a simple template can be found below for convenience. Feel free to use, adapt or completely remove.

Abbreviations

Abbreviation	Definition
ISA	International Standard Atmosphere
...	

Symbols

Symbol	Definition	Unit
V	Velocity	[m/s]
...		
ρ	Density	[kg/m^3]
...		

Chapter 1

拉格朗日对偶性

Keyword. 等式约束优化条件（拉格朗日乘子法)、不等式约束优化条件

1.1. 拉格朗日乘子法

考虑 n 元函数 $f(x_1, x_2, \cdots, x_n)$ 的极值。一般情况下，函数极值的必要条件是

$$f \text{ 在极值点 } r \text{ 的导数为 } 0$$

我们考虑约束下的函数极值问题

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} f(x) \\ &s.t.\ \ g(x) = c \end{aligned} \tag{1.1}$$

拉格朗日乘数法的算法是

1. 首先构造拉格朗日函数

$$L(x, \lambda) = f(x) - \lambda g(x), \quad x \in \mathbb{R}^n, \lambda \in \mathbb{R} \tag{1.2}$$

2. 目的是求出使得拉格朗日函数的梯度为 0 的点

$$\nabla L = \nabla f(x) - \lambda \nabla g = 0 \tag{1.3}$$

解出上述方程即是极值问题的解。

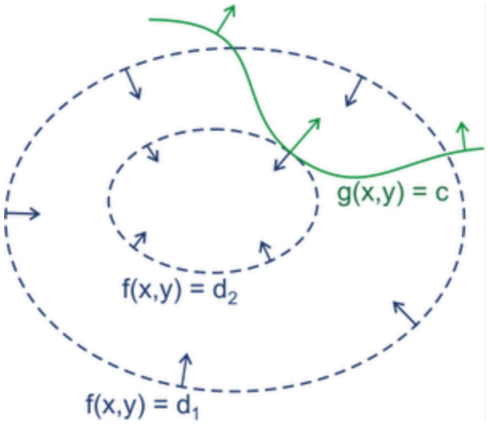

图 1.1: 拉格朗日乘子法的等高线图

根据等高线图，无约束下的极值点在 $f(x,y)=d_2$ 内部，由于 $g(x,y)=c$ 的约束使得只能取满足 $g(x,y)=c$ 和 $f(x,y)$ 的交点，而交点中能使得 $f(x,y)$ 取得最小值，正好是 $f(x,y)=d_2$ 和 $g(x,y)=c$ 的切点。而该点正好满足梯度方向相反，即存在 λ 使得

$$\nabla f = \lambda \nabla g \tag{1.4}$$

因此原问题就变成了求拉格朗日函数的无约束极值问题。

1.2. 极小极大值问题（原始问题）

假设 $f(x), c_i(x), h_j(x)$ 是定义在 $\mathbb{R}^n$ 上的连续可微函数。考虑优化问题

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} \ f(x) \\ &s.t. \quad c_i(x) \leqslant 0, \quad i=1,2,\cdots,k; \\ &\qquad h_j(x)=0, \quad j=1,2,\cdots,l; \end{aligned} \tag{1.5}$$

引入拉格朗日函数

$$L(x,\alpha,\beta) = f(x) + \sum_{i=1}^{k} \alpha_i c_i(x) + \sum_{j=1}^{l} \beta_j h_j(x) \tag{1.6}$$

其中 $x=(x^{(1)},x^{(2)},\cdots,x^{(n)})^T \in \mathbb{R}^n$，$\alpha_i, \beta_j$ 是拉格朗日乘子，$\alpha_i \geqslant 0$。考虑

$$\theta_P(x) = \max_{\alpha,\beta,\alpha_i\geqslant 0} L(x,\alpha,\beta) \tag{1.7}$$

其中下标 P 表示原始问题。

假设给定某个 x，如果 x 违反原始问题的约束条件，即存在 i 使得 $c_i(x)>0$ 或存在 j 使得 $h_j(x)\neq 0$，则

$$\theta_P(x) = \max_{\alpha,\beta,\alpha_i\geqslant 0} [f(x) + \sum_{i=1}^{k} \alpha_i c_i + \sum_{i=1}^{l} \beta_j h_j(x)] = +\infty \tag{1.8}$$

因为若 $c_i(x)>0$，那么令 $\alpha_i \to +\infty$，若 $h_j(x)\neq 0$，则令 β_j 使得 $\beta_j h_j(x) \to +\infty$，其余将 α_i, β_i 取为 0 即可。

因此

$$\theta_P(x) = \begin{cases} f(x), & \text{x 满足原始问题约束} \\ +\infty, & else \end{cases} \tag{1.9}$$

所以如果考虑极小化问题

$$\theta_P(x) = \min_x \max_{\alpha,\beta,\alpha_i \geqslant 0} L(x,\alpha,\beta) \tag{1.10}$$

它与原始优化问题是等价的，即解相同。问题 $\min\limits_x \max\limits_{\alpha,\beta,\alpha_i \geqslant 0} L(x,\alpha,\beta)$ 也被称为*广义拉格朗日函数的极小极大问题*。为了方便，定义原始问题为最优值

$$p^* = \min_x \theta_P(x) \tag{1.11}$$

为原始问题的值。

1.3. 极大极小值问题（对偶问题）

定义

$$\theta_D(\alpha,\beta) = \min_x L(x,\alpha,\beta) \tag{1.12}$$

在考虑极大化 $\theta_D(\alpha,\beta)$

$$\max_{\alpha,\beta,\alpha_i \geqslant 0} \theta_D(\alpha,\beta) = \max_{\alpha,\beta,\alpha_i \geqslant 0} \min_x L(x,\alpha,\beta) \tag{1.13}$$

这样的问题称为*广义拉格朗日函数的极大极小问题*

可以将广义拉格朗日函数的极大极小问题表示为约束最优化问题:

$$\begin{aligned} &\max_{\alpha,\beta,\alpha_i \geqslant 0} \theta_D(\alpha,\beta) = \max_{\alpha,\beta,\alpha_i \geqslant 0} \min_x L(x,\alpha,\beta) \\ &s.t. \quad \alpha_i \geqslant 0, \quad i = 1,2,\cdots,k \end{aligned} \tag{1.14}$$

称为原始问题的对偶问题，定义对偶问题的最优值

$$d^* = \max_{\alpha,\beta,\alpha_i \geqslant 0} \theta_D(\alpha,\beta) \tag{1.15}$$

称为对偶问题的值。

1.4. 原始问题和对偶问题的关系

theorem 1.4.1: *若原始问题和对偶问题都有最优值，则*

$$d^* = \max_{\alpha,\beta,\alpha_i \geqslant 0} \theta_D(\alpha,\beta) \leqslant \min_{\alpha,\beta,\alpha_i \geqslant 0} \theta_P(\alpha,\beta) = p^* \tag{1.16}$$

proof. 对任意的 α,β 和 x，有

$$\theta_D(\alpha,\beta) = \min_x L(x,\alpha,\beta) \leqslant L(x,\alpha,\beta) \leqslant \min_x \max_{\alpha,\beta,\alpha_i \geqslant 0} L(x,\alpha,\beta) = p^* \tag{1.17}$$

即

$$\theta_D(\alpha,\beta) \leqslant \theta_P(x) \tag{1.18}$$

由于原始问题和对偶问题都有最优值

$$\max_{\alpha,\beta,\alpha_i \geqslant 0} \theta_D(\alpha,\beta) \leqslant \min_x \theta_P(x) \tag{1.19}$$

□

corollary 1.4.2: 设 x^* 和 α^*, β^* 分别是原始问题和对偶问题的可行解，并且 $d^* = p^*$，则 x^* 和 α^*, β^* 分别是原始问题和对偶问题的最优解。

theorem 1.4.3: 考虑原始问题和对偶问题。假设函数 $f(x)$ 和 $c_i(x)$ 是凸函数，$h_j(x)$ 是仿射函数；并且假设不等式约束 $c_i(x)$ 是严格可行的，即存在 x_i 对所有的 i 有 $c_i(x) < 0$，则存在 x^*, α^*, β^*，使得 x^* 是原始问题的解，α^*, β^* 是对偶问题的解，并且

$$p^* = d^* = L(x^*, \alpha^*, \beta^*) \tag{1.20}$$

1.5. KKT 条件

theorem 1.5.1: 对原始问题和对偶问题。假设函数 $f(x)$ 和 $c_i(x)$ 是凸函数，$h_j(x)$ 是仿射函数；并且不等式约束 $c_i(x)$ 是严格可行的，则 x^* 和 α^*, β^* 分别是原始问题和对偶问题的解的充分必要条件是：α^*, β^*, x^* 满足下面的 ***Karush-Kuhn-Tucker* 条件** *(KKT)*

$$\bigtriangledown_x L(x^*, \alpha^*, \beta^*) = 0 \tag{1.21}$$

$$\alpha_i^* c_i(x^*) = 0, \quad i = 1, 2, \cdots, k \tag{1.22}$$

$$c_i(x^*) \leqslant 0, \quad i = 1, 2, \cdots, k \tag{1.23}$$

$$\alpha_i^* \geqslant 0, \quad i = 1, 2, \cdots, k \tag{1.24}$$

$$h_j(x^*) = 0, \quad i = 1, 2, \cdots, l \tag{1.25}$$

其中 (1.22) 是 KKT 的**对偶互补条件**。由此条件，若 $\alpha_i^* > 0$，则 $c_i(x^*) = 0$

Chapter 2

硬间隔最大化与线性可分向量机

Keyword: 硬间隔最大化、软间隔最大化、核技巧

2.1. 线性可分支持向量机

definition 2.1.1: *(线性可分支持向量机)* 给定线性可分训练数据集，通过间隔最大化成等价地求解相应的凸二次规划问题学习得到的分离超平面为

$$\omega^* \cdot x + b^* = 0 \tag{2.1}$$

以及相应的分类决策函数

$$f(x) = sign(\omega^* \cdot x + b^*) \tag{2.2}$$

称为**线性可分支持向量机**。

线性支持向量机假设输入空间和特征空间的元素一一对应，并将输入空间中的输入映射为特征空间中的特征向量。非线性支持向量机利用一个从输入空间到特征孔甲你中毒非线性映射将输入映射为特征向量。所以支持向量机的学习都是在特征空间中进行的。

一般地，当训练数据集线性可分时，存在无穷个分离超平面可以将两类数据正确分开。感知机利用误分类最小策略，求得分离超平面，不过这个时候的解有无数个。线性可分支持向量机利用间隔最大化求最有分离超平面，这个时候的解是唯一的。

2.2. 函数间隔和几何间隔

函数间隔

definition 2.2.1: *(函数间隔)* 对于给定的训练数据集 T 和超平面 (ω, b)，定义超平面 (ω, b) 关于样本点 (x_i, y_i) 的函数间隔为

$$\hat{\gamma}_i = y_i(\omega \cdot x_i + b) \tag{2.3}$$

定义超平面 (ω, b) 关于训练数据集 T 的函数间隔为超平面 (ω, b) 关于 T 中所有样本点的函数间隔的最小值

$$\hat{\gamma}_i = \min_{i=1,\cdots,N} \hat{\gamma}_j \tag{2.4}$$

其中点到平面的距离公式由下面的形式给出

$$d = \frac{\overrightarrow{OP} \cdot \vec{n}}{\|\vec{n}\|} \tag{2.5}$$

函数间隔可以表示分类预测的正确性和确信程度。但是选择分离超平面时，我们发现只要等比例 ω 和 b，分离超平面是不变的如：

$$\omega \cdot x + b = 0 \quad \Leftrightarrow \quad 2\omega \cdot x + 2b = 0 \tag{2.6}$$

但是此时函数间隔却变成了 2 倍

$$\hat{\gamma}_i = y_i(2\omega \cdot x_i + 2b) = 2y_i(\omega \cdot x_i + b) \tag{2.7}$$

几何间隔

因此，需要对分离超平面的法向量加某些约束，如规范化 $\|\omega\| = 1$，使得间隔是确定的，这时函数间隔成为几何间隔。

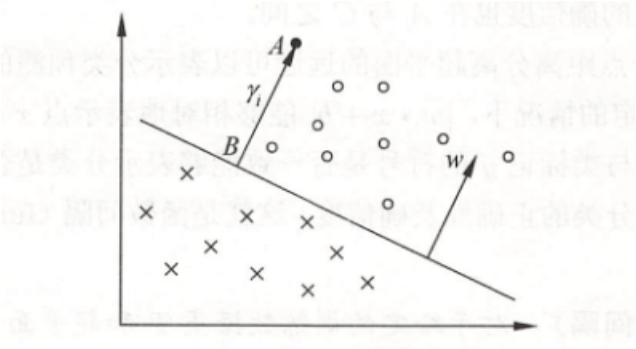

图 2.1: 几何间隔

给定超平面 (ω, b) 以及法向量 ω。点 A 表示某一实例 x_i，分类标记为 $y_i = \pm 1$。A 与超平面 (ω, b) 的距离由线段 AB 给出，记为 γ_i。

$$\gamma_i = \pm(\frac{\omega}{\|\omega\|} \cdot x_i + \frac{b}{\|\omega\|}) \tag{2.8}$$

一般地，当样本点 (x_i, y_i) 被正确分类时，点 x_i 与超平面 (ω, b) 的距离为

$$\gamma_i = y_i(\frac{\omega}{\|\omega\|} \cdot x_i + \frac{b}{\|\omega\|}) \tag{2.9}$$

definition 2.2.2: *(几何间隔)* 对于给定的训练数据集 T 和超平面 (ω, b)，定义超平面 (ω, b) 关于样本点 (x_i, y_i) 的几何间隔为

$$\gamma_i = y_i(\frac{\omega}{\|\omega\|} \cdot x_i + \frac{b}{\|\omega\|}) \tag{2.10}$$

定义超平面 (ω, b) 关于训练数据集 T 的几何间隔为超平面 (ω, b) 关于 T 中所有样本点的几何间隔的最小值

$$\gamma_i = \min_{i=1,\cdots,N} \hat{\gamma}_j \tag{2.11}$$

函数间隔和几何间隔的关系

从函数间隔和几何间隔的定义，二者有如下关系

$$\gamma = \frac{\hat{\gamma}}{\|\omega\|} \tag{2.12}$$

可见如果 $\|\omega\| = 1$，则二者相等。如果超平面参数 ω 和 b 成比例改变，函数间隔也成比例改变而几何间隔不变。

2.3. 最大间隔分离超平面

如何求得几何间隔最大的分离超平面？这个问题表述为下列优化问题

$$\begin{aligned} &\max_{\omega,b} \quad \gamma \\ &s.t. \quad y_i(\frac{\omega}{\|\omega\|} \cdot x_i + \frac{b}{\|\omega\|}) \geqslant \gamma, \quad i = 1, 2, \cdots, N \end{aligned} \tag{2.13}$$

考虑几何间隔和函数间隔的关系，优化问题等价为

$$\begin{aligned} &\max_{\omega,b} \quad \frac{\hat{\gamma}}{\|\omega\|} \\ &s.t. \quad y_i(\frac{\omega}{\|\omega\|} \cdot x_i + \frac{b}{\|\omega\|}) \geqslant \hat{\gamma}, \quad i = 1, 2, \cdots, N \end{aligned} \tag{2.14}$$

函数间隔 $\hat{\gamma}$ 的取值并不影响优化问题的解。事实上，假设 ω 和 b 成比例改变为 $\lambda\omega$ 和 λb，这时函数间隔成为 $\lambda\hat{r}$。函数间隔的这一改变对上面最优化问题的不等式约束没有影响，对目标函数的优化也没有影响，即它产生一个等价的最优化问题，这样就可以取 $\hat{\gamma} = 1$ 代入优化问题。

等价的优化问题

注意到最大化 $1/\|\omega\|$ 和最小化 $\frac{1}{2}\|\omega\|^2$ 是等价的[1]。于是就得到下面的线性可分支持向量机学习的最优化问题：

$$\begin{aligned} &\max_{\omega,b} \quad \frac{1}{2}\|\omega\|^2 \\ &s.t. \quad y_i(\frac{\omega}{\|\omega\|} \cdot x_i + \frac{b}{\|\omega\|}) \geqslant 1, \quad i = 1, 2, \cdots, N \end{aligned} \tag{2.15}$$

这是一个**凸二次规划问题**。通俗来描述这个问题是，寻找到一个超平面，使得样本数据集到超平面的几何间距最大，但同时几何间距又收到本身定义的制约，即样本数据集的几何间距是所有样本点几何间距的最小值。

2.4. 最大间隔算法

输入：线性可分训练数据集 $T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$，其中 $x_i \in \mathbb{R}^n, y_i \in \{-1, +1\}$。
输出：最大间隔分离超平面和分类决策函数。

[1] 在趋近于 0 时，$\frac{1}{2}\|\omega\|^2$ 趋于最小，而 $\frac{1}{\|\omega\|}$ 趋于最大

1. 构造并求解约束最优化问题

$$\min_{\omega,b} \frac{1}{2}\|\omega\|^2$$
$$s.t.\ y_i(\omega \cdot x_i + b) - 1 \geqslant 0,\ \ i = 1, 2, \cdots, N \tag{2.16}$$

求得最优解 ω^*, b^*。

2. 由此得到分离超平面

$$\omega^* \cdot x + b^* = 0 \tag{2.17}$$

分类决策函数可以写成

$$f(x) = sign(\omega^* \cdot x + b^*) \tag{2.18}$$

2.5. 最大间隔超平面的存在性和唯一性

theorem 2.5.1: 若训练数据集 T 是线性可分的，则可将数据集中样本点完全正确分开的最大间隔分离超平面存在且唯一

proof.

2.6. 支持向量和间隔边界

训练数据集的样本点中与分离超平面距离最近的样本点的实例称为**支持向量** *(support vector)*。支持向量是使得约束条件 $y_i(\frac{\omega}{\|\omega\|} \cdot x_i + \frac{b}{\|\omega\|}) \geqslant \gamma_i,\ \ i = 1, 2, \cdots, N$ 成立的点，即

$$y_i(\omega \cdot x_i + b) - 1 = 0 \tag{2.19}$$

对于 $y_1 = +1$ 的正例点，支持向量在超平面

$$H_1 : \omega \cdot x + b = 1 \tag{2.20}$$

对于 $y_1 = -1$ 的服例点，支持向量在超平面

$$H_2 : \omega \cdot x + b = -1 \tag{2.21}$$

如下图所示，H_1 和 H_2 上的点就是支持向量。

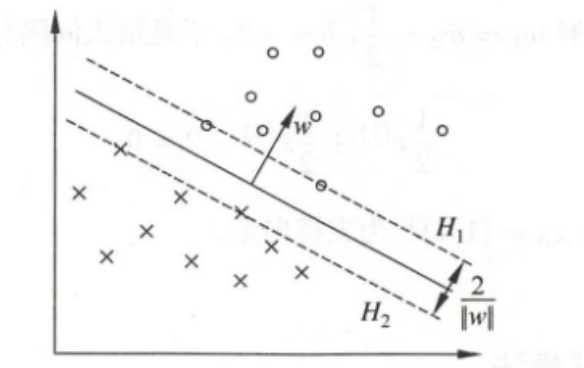

图 2.2: 支持向量

H_1 和 H_2 之间的距离称为**间隔** *(margin)*。间隔依赖于分离超平面的法向量 ω，等于 $2/\|\omega\|$。H_1 和 H_2 称为**间隔边界** *(margin boundary)*。

在决定分离超平面时，只有支持向量起作用，而其他实例点不起作用。所以支持向量机由很少的支持向量作为训练样本决定。

2.7. 支持向量机的对偶算法

应用**拉格朗日对偶性**，通过求解**对偶问题** *(dual problem)* 得到原始问题的最优解。这就是线性可分支持向量机的对偶算法。

首先构建**拉格朗日函数** *(Lagrange function)*。引入**拉格朗日乘子** *(Lagrange multiplier)*：$\alpha_1 \geqslant 0$。

$$L(\omega, b, \alpha_i) = \frac{1}{2}\|\omega\|^2 - \sum_{i=1}^{N} \alpha_i y_i(\omega \cdot x_i + b) + \sum_{i=1}^{N} \alpha_i \tag{2.22}$$

其中 $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_N)^T$ 为拉格朗日乘子向量。根据拉格朗日乘子法，原始问题的对偶问题是极大极小问题

$$\max_{\alpha} \min_{\omega, b} L(\omega, b, \alpha) \tag{2.23}$$

theorem 2.7.1: *设 $\alpha^* = (\alpha_1^*, \alpha_2^*, \cdots, \alpha_l^*)^T$ 是最优化对偶问题 (2.23) 的解，则存在下标 j，使得 $\alpha_j^* > 0$，并可按照下式求得原始最优化问题的解 ω^*, b^*。*

$$\omega^* = \sum_{i=1}^{N} \alpha_i^* y_i x_i \tag{2.24}$$

$$b^* = y_i - \sum_{i=1}^{N} \alpha_i^* y_i (x_i \cdot x_j) \tag{2.25}$$

由此定理，分离超平面可以写成

$$\sum_{i=1}^{N} \alpha_i^* y_i (x \cdot x_i) + b^* = 0 \tag{2.26}$$

分类决策函数可以写成

$$f(x) = sign(\sum_{i=1}^{N} \alpha_i^* y_i (x \cdot x_i) + b^*) \tag{2.27}$$

这样分类决策函数只依赖于输入 x 和训练样本输入的内积。$f(x) = sign(\sum_{i=1}^{N} \alpha_i^* y_i (x \cdot x_i) + b^*)$ **线性可分支持向量机的对偶形式**。

线性可分支持向量机学习算法

输入：线性可分训练集 $T = (x_1, , y_1), (x_2, y_2), \cdots, (x_N, y_N)$，其中 $x_i \in \mathbb{R}^n$，$y_i \in \{-1, +1\}$。输出：分离超平面和分类决策函数

1. 构造并求解约束最优化问题

$$\begin{aligned} &\min_{\alpha} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) = \sum_{i=1}^{N}\alpha_i \\ &.s.t \sum_{i=1}^{N}\alpha_i y_i = 0 \\ &\alpha_i \geqslant 0, \quad i = 1, 2, \cdots, N \end{aligned} \tag{2.28}$$

求得最优解 $\alpha^* = (\alpha_1^*, \alpha_2^*, \cdots, \alpha_N^*)^T$

2. 计算

$$\omega^* = \sum_{i=0}^{N}\alpha_i^* y_i x_i \tag{2.29}$$

并选择 α^* 的一个正分量 $\alpha_j^* > 0$ 计算

$$b^* = y_i \sum_{i=1}^{N}\alpha_i^* y_i (x_i \cdot x_j) \tag{2.30}$$

3. 求得分离超平面和决策函数

$$\omega^* \cdot x + b^* = 0 \tag{2.31}$$

$$f(x) = sign(\omega^* \cdot x + b^*) \tag{2.32}$$

线性可分支持向量机中，ω^* 和 b^* 只依赖于训练数据中对应 $\alpha_i^* > 0$ 的样本点 (x_i, y_i)，我们将训练数据中对应于 $\alpha_i^* > 0$ 的实例点 $x_I \in \mathbb{R}^n$ 称为支持向量。

definition 2.7.1 (支持向量)**:** *考虑原始最优化问题以及对偶最优化问题，将训练数据集集中对应于 $\alpha_i^* > 0$ 的样本点 (x_i, y_i) 的实例 $x_i \in \mathbb{R}^n$ 称为* ***支持向量***。

Chapter 3

软间隔最大化和线性支持向量机

3.1. 线性支持向量机

假定给定一个特征空间的训练数据集

$$T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\} \tag{3.1}$$

其中 $x_i \in \mathbb{R}^n$，$y_i = -1, +1, i = 1, 2, \cdots, N$，其中 x_i 为第 i 个特征向量。通常情况下，训练数据中有一些**特异点** *(outlier)*，将这些特异点除去后，剩下大部分的样本点的集合是线性可分的。

线性不可分意味着某些样本点 (x_i, y_i) 不能满足函数间隔大于等于 1 的约束条件。为了解决这个问题，可以对每个样本点 (x_i, y_i) 引进一个松弛变量 $\xi_i \geqslant 0$，使得函数间隔加上松弛变量大于等于 1，这样约束条件变为

$$y_i(\omega \cdot x_i + b) \geqslant 1 - \xi_i \tag{3.2}$$

同时对每个松弛变量 ξ_i，支付一个代价 ξ_i，目标函数由由原来的 $\frac{1}{2}\|\omega\|^2$ 变成

$$\frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{N}\xi_i \tag{3.3}$$

其中 $C > 0$ 称为**惩罚参数**。C 值大时对误分类的惩罚增大，反之惩罚减小。

线性支持向量机的定义

definition 3.1.1: *对于给定的线性不可分的训练数据集，通过求解以下凸二次规划问题 (原始问题)*

$$\begin{aligned} &\min_{\omega,b,\xi} \quad \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{N}\xi_i \\ &s.t. \quad y_i(\omega \cdot x_i + b) \geqslant 1 - \xi_i, \quad i = 1, 2, \cdots, N \\ &\qquad \xi_i \geqslant 0, \quad i = 1, 2, \cdots, N \end{aligned} \tag{3.4}$$

即软间隔最大化问题，得到的分离超平面为

$$\omega^* \cdot x + b^* = 0 \tag{3.5}$$

以及相应的分类决策函数

$$f(x) = sign(\omega^* \cdot x + b^*) \tag{3.6}$$

称为**线性支持向量机**。

3.2. 学习的对偶问题

原始问题的拉格朗日函数为

$$L(\omega, b, \xi, \alpha, \mu) = \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i(y_i(\omega \cdot x_i + b) - 1 + \xi_i) - \sum_{i=1}^{N}\mu_i\xi_i \tag{3.7}$$

其中 $\alpha_i \geqslant 0, \mu_i \geqslant 0$。

对偶问题时拉格朗日函数的极大极小值问题，首先对 $L(\omega, b, \xi, \alpha, \mu)$ 求极小

$$\nabla_\omega L(\omega, b, \xi, \alpha, \mu) = \omega - \sum_{i=1}^{N}\alpha_i y_i x_i = 0 \tag{3.8}$$

$$\nabla_b L(\omega, b, \xi, \alpha, \mu) = -\sum_{i=1}^{N}\alpha_i y_i = 0 \tag{3.9}$$

$$\nabla_{\xi_i} L(\omega, b, \xi, \alpha, \mu) = C - \alpha_i - \mu_i = 0 \tag{3.10}$$

求解得到

$$\omega = \sum_{i=1}^{N}\alpha_i y_i x_i \tag{3.11}$$

$$\sum_{i=1}^{N}\alpha_i y_i = 0 \tag{3.12}$$

$$C - \alpha_i - \mu_i = 0 \tag{3.13}$$

代入拉格朗日函数得到

$$\min_{\omega,b,\xi} L(\omega, b, \xi, \alpha, \mu) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) + \sum_{i=1}^{N}\alpha_i \tag{3.14}$$

再对极小问题求极大，即得到对偶最优化问题

$$\max_{\alpha} \quad -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j(x_i \cdot x_j) + \sum_{i=1}^{N}\alpha_i \tag{3.15}$$

$$s.t. \quad \sum_{i=1}^{N}\alpha_i y_i = 0 \tag{3.16}$$

$$C - \alpha_i - \mu_i = 0 \tag{3.17}$$

$$\alpha_i \geqslant 0 \tag{3.18}$$

$$\mu_i \geqslant 0, \ \ i = 1, 2, \cdots, N \tag{3.19}$$

将对偶最优化问题进行变换，利用等式约束 (3.17) 消去 μ_i，从而只留下变量 α_i，并将对目标函数求极小转换为求极大。即得到原始问题的对偶问题是

$$\min_{\alpha} \quad \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{N}\alpha_i \tag{3.20}$$

$$s.t. \quad \sum_{i=1}^{N}\alpha_i y_i = 0 \tag{3.21}$$

$$0 \leqslant \alpha_i \leqslant C, \ \ i = 1, 2, \cdots, N \tag{3.22}$$

可以通过求解对偶问题而得到原始问题的解，进而确定分离超平面和决策函数。

原始问题最优解和对偶问题最优解的关系

theorem 3.2.1: 设 $\alpha^* = (\alpha_1^*, \alpha_2^*, \cdots, \alpha_N^*)^T$ 是对偶问题的一个解，若存在 α^* 的一个分量 $a_j^*(0 < \alpha^* < C)$，则原始问题的解 ω^*, b^* 可按照下式求得

$$\omega = \sum_{i=1}^{N}\alpha_i^* y_i x_i \tag{3.23}$$

$$b^* = y_i - \sum_{i=1}^{N} y_i \alpha^* (x_i \cdot x_j) \tag{3.24}$$

proof. 原始问题是凸二次规划问题，解满足 KKT 条件，

$$\nabla_{\omega} L(\omega^*, b^*, \xi^*, \alpha^*, \mu^*) = \omega^* - \sum_{i=1}^{N}\alpha^* y_i x_i = 0 \tag{3.25}$$

$$\nabla_{b} L(\omega^*, b^*, \xi^*, \alpha^*, \mu^*) = -\sum_{i=1}^{N}\alpha^* y_i = 0 \tag{3.26}$$

$$\nabla_{\xi} L(\omega^*, b^*, \xi^*, \alpha^*, \mu^*) = C - \alpha^* - \mu^* = 0 \tag{3.27}$$

$$\alpha_i^*(y_i(\omega^* \cdot x_i + b^*) - 1 + \xi_i^*) = 0 \tag{3.28}$$

$$\mu_i^* \xi_i^* = 0 \tag{3.29}$$

$$y_i(\omega^* \cdot x_i + b^*) - 1 + \xi_i^* \geqslant 0 \tag{3.30}$$

$$\xi_i^* \geqslant 0 \tag{3.31}$$

$$\alpha_i^* \geqslant 0 \tag{3.32}$$

$$\mu_i^* \geqslant 0, \ \ i = 1, 2, \cdots, N \tag{3.33}$$

由此定理可知，分离超平面可以写成

$$\sum_{i=1}^{N}\alpha_i^* y_i (x \cdot x_i) + b^* = 0 \tag{3.34}$$

分类决策函数可以写成

$$f(x) = sign(\sum_{i=1}^{N}\alpha_i^* y_i (x \cdot x_i) + b^*) \tag{3.35}$$

上式子称为**线性支持向量机的对偶形式**。

3.3. 线性支持向量机学习算法

输入：训练数据集 $T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$，其中 $x_i\in\mathbb{R}^n$，$y_i=\{-1,+1\}$。
输出：分离超平面和决策函数。

1. 选择惩罚参数 $C>0$，构造并求解凸二次规划问题

$$\min_{\alpha}\quad \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j(x_i\cdot x_j)-\sum_{i=1}^{N}\alpha_i \tag{3.36}$$

$$s.t.\quad \sum_{i=1}^{N}\alpha_i y_i=0 \tag{3.37}$$

$$0\leqslant\alpha_i\leqslant C,\ \ i=1,2,\cdots,N \tag{3.38}$$

求得最优解 $\alpha^*=(\alpha_1^*,\alpha_2^*,\cdots,\alpha_N^*)^T$。

2. 计算 $\omega^*=\sum_{i=1}^{N}\alpha_i^* y_i x_i$。选择 α^* 的一个分量 α_j^* 适合条件 $0<\alpha_j^*<C$，计算

$$b^*=y_i-\sum_{i=1}^{N}y_i\alpha_i^*(x_i\cdot x_i) \tag{3.39}$$

求得分离超平面和决策函数

$$\omega^*\cdot x+b^*=0 \tag{3.40}$$

$$f(x)=sign(\omega^*\cdot x+b^*) \tag{3.41}$$

从理论上，原始问题对 b 的解可能不唯一，然而实际应用中，往往只会出现算法叙述的情况。

3.4. 支持向量

在线性不可分的情况下，将对偶问题的解 $\alpha^*=(\alpha_1^*,\alpha_2^*,\cdots,\alpha_N^*)$ 中对应 $\alpha_i^*>0$ 的样本点 (x_i,y_i) 的实例 x_i 称为**软间隔的支持向量**。

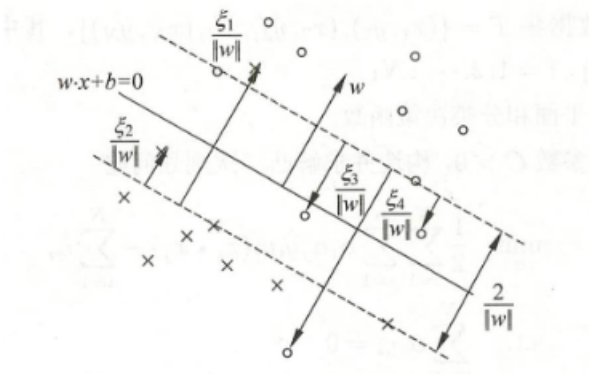

图 3.1: 软间隔的支持向量

如上图所示，分离超平面由实线表示，间隔边界由虚线表示，正例点由”○“表示，负例点由 × 表示。图中还标出了实例 x_i 到间隔边界的距离 $\frac{\xi_i}{\|\omega\|}$。

软间隔的支持向量 x_i 或者在间隔边界上，或者在间隔边界与分离超平面之间，或者在分离超平面误分一侧。

1. 若 $\alpha_i^* < C$，则 $\xi_i = 0$，支持向量 x_i 恰好活在间隔边界上；
2. 若 $\alpha_i^* = C$，$0 < \xi_i < 1$，则分类正确，x_i 在间隔边界与分离超平面之间；
3. 若 $\alpha_i^* = C$，$\xi_i = 1$，则 x_i 在分离超平面上；
4. 若 $\alpha_i^* = C$，$\xi_i > 1$，则 x_i 位于分离超平面误分一侧。

3.5. 合页损失函数

线性支持向量机学习还有另一种解释，就是最小化以下目标函数

$$\sum_{i=1}^{N}[1 - y_i(\omega \cdot x_i + b)]_+ + \lambda\|\omega\|^2 \tag{3.42}$$

目标函数的第 1 项是经验损失或经验风险，函数

$$L(\omega \cdot x + b) = [1 - y_i(\omega \cdot x_i + b)]_+ \tag{3.43}$$

称为**合页损失函数** *(hinge loss function)*。下标"+" 表示一下正值函数

$$|z|_+ = \begin{cases} z, & z > 0 \\ 0, & z \leqslant 0 \end{cases} \tag{3.44}$$

即当样本点被正确分类且函数间隔 (确信度)$y_i(\omega \cdot x_i + b)$ 大于 1 时，损失为 0。否则损失是 $1 - y_i(\omega \cdot x_i + b)$。注意到图 (3.1) 中的实例点 x_4 被正确分类，但损失部位 0，目标函数的第 2 项是系数为 λ 的 ω 的 L_2 范数，是正则化项。

线性支持向量机原始问题的等价最优化问题

theorem 3.5.1: 线性支持向量机的原始最优化问题等价于最优化问题

$$\min_{\omega,b} \sum_{i=1}^{N}[1 - y_i(\omega \cdot x_i + b)]_+ + \lambda\|\omega\|^2 \tag{3.45}$$

proof.

合页损失函数的几何意义

合页损失函数的图形如下图所示

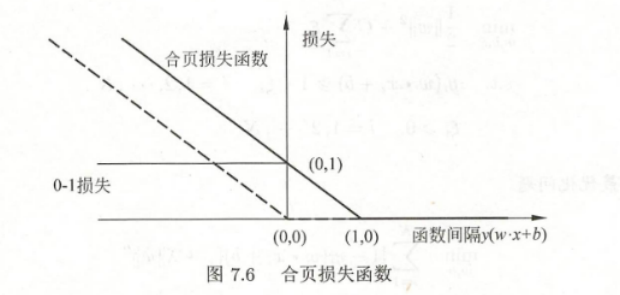

图 7.6 合页损失函数

图 3.2: 合页损失函数

横轴是函数间隔 $y(\omega \cdot x + b)$，纵轴是损失[1]。

图中还画出 ***0-1 损失函数***。可以认为他是二分类问题的真正的损失函数，而合页损失函数是 0-1 损失函数的上界。由于 0-1 损失函数不是连续可导的，直接优化由于其构成的目标函数比较困难，可以认为线性支持向量机是优化由 0-1 损失函数的上界构成的目标函数。这时的上界损失函数又称为**代理损失函数** ***(surrogate loss function)***。

图中虚线显示的是感知机的损失函数

$$[-y_i(\omega \cdot x_i + b)]_+ \tag{3.46}$$

这时，当样本点被正确分类时，损失为 0，否则损失是 $-y_i(\omega \cdot x_i + b)$。相比之下，合页损失函数不仅要求分类正确，且确信度足够高时损失才是 0，即合页损失函数对学习有更高要求。

[1]由于函数形状像一个合页，故称合页损失函数

Chapter 4

核函数和非线性支持向量机

4.1. 核技巧

非线性分类问题

非线性分类问题无法通过线性模型分类，如下图所示，正例点和负例点混杂在一起。

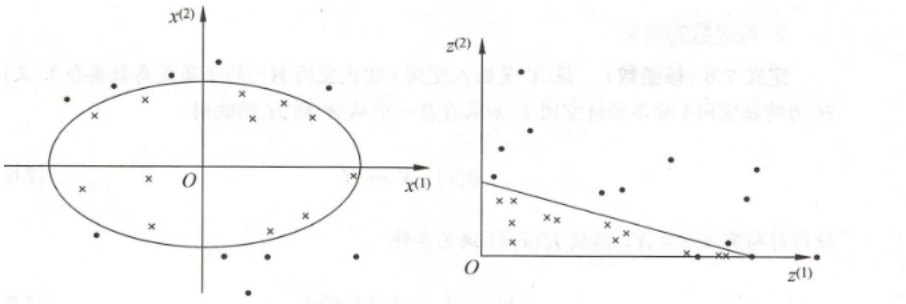

图 4.1: 非线性分类问题与核技巧

一般来说，对给定一个训练数据集 $T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$，其中，实例 x_i 属于输入空间，$x_i \in \mathbb{R}^n$，对应的标记有两类 $y_i = \{-1, +1\}$，如果能用 $\mathbb{R}^n$ 中的一个**超曲面**将正负例正确分开，则称这个问题为**非线性可分问题**。

通过线性技巧解决非线性问题

但是非线性问题往往不好求解，所以希望能用线性分类问题的方法去解决这类问题，所采取的方法是进行一个 **非线性变换**，将非线性问题变成线性问题，通过解变换后的线性问题的方法求解原来的非线性问题。

4.2. 核函数的定义

definition 4.2.1: 设 $\mathcal{X}$ 是输入空间，$\mathcal{H}$ 为特征空间 *(希尔伯特空间)*，如果存在一个映射

$$\phi(\cdot) : \mathcal{X} \to \mathcal{H} \tag{4.1}$$

使得对于所有的 $x, z \in \mathcal{X}$，函数 $K(x, z)$ 满足条件

$$K(x, z) = \phi(x) \cdot \phi(z) \tag{4.2}$$

则称 $K(x, z)$ 为**核函数**，$\phi(\cdot)$ 为映射函数。

核技巧的想法是，在学习与预测中只定义核函数 $K(x, z)$，而不是显式地定义映射函数 ϕ。

example 4.2.1: 假设假设输入空间是 $\mathbb{R}^2$，核函数是 $K(x, z) = (x \cdot z)^2$，找出相关特征空间 $\mathcal{H}$ 和映射函数 $\phi(x)$。

可以去特征空间为 $\mathbb{R}^3$，记 $x = (x^{(1)}, x^{(2)})^T$，$z = (z^{(1)}, z^{(2)})^T$

$$(x \cdot z)^2 = (x^{(1)}z^{(1)} + x^{(2)}z^{(2)})^2 = (x^{(1)}z^{(1)})^2 + 2x^{(1)}z^{(1)}x^{(2)}z^{(2)} + (x^{(2)}z^{(2)})^2 \tag{4.3}$$

所以可以取映射

$$\phi(x) = ((x^{(1)})^2, \sqrt{2}x^{(1)}x^{(2)}, (x^{(2)})^2)^T \tag{4.4}$$

验证一下

$$\phi(x) \cdot \phi(z) = (x \cdot z)^2 = K(x, z) \square \tag{4.5}$$

仍然取 $\mathcal{H} = \mathbb{R}^3$

$$\phi(x) = \frac{1}{\sqrt{2}}((x^{(1)})^2 - (x^{(2)})^2, 2x^{(1)}x^{(2)}, ((x^{(1)})^2 + (x^{(2)})^2)^T \tag{4.6}$$

验证内积仍然得到核函数。

还可以取 $\mathcal{H} = \mathbb{R}^4$

$$\phi(x) = ((x^{(1)})^2, (x^{(1)})(x^{(2)}), (x^{(1)})(x^{(2)}), (x^{(2)})^2)^2 \tag{4.7}$$

4.3. 核技巧在支持向量机中的应用

在线性支持向量机的对偶问题中，无论是目标函数还是决策函数，都只涉及输入实例和实例直接的内积，在线性支持向量机的对偶问题的目标函数中的内积 $x_i \cdot x_j$ 可以用核函数 $K(x_1, x_2) = \phi(x_i) \cdot \phi(x_j)$ 来替代。此时对偶问题的目标函数

$$W(\alpha) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{N}\alpha_i \tag{4.8}$$

同样分类决策函数中的内积也可以用核函数来替代。

$$f(x) = sign(\sum_{i=1}^{N_s}\alpha_i^* y_i K(x_i, x) + b^*) \tag{4.9}$$

这等价于经过映射函数将原来的输入空间变换到一个新的特征空间。在实际应用中，往往依赖领域知识直接选择核函数，因此核函数的需要通过实验去验证。

4.4. 正定核

不用构造映射 $\phi(x)$ 能否直接判断一个给定的函数 $K(x,z)$ 是不是核函数?

假设 $K(x,z)\in\mathcal{X}\times\mathcal{X}$ 是对称函数,并且对任意的 $x_1,x_2,\cdots,x_m\in\mathcal{X}$,$K(x,z)$ 关于 $x_1,x_2,\cdots,x_m$ 的 Gram 矩阵是半正定的。可以依据函数 $K(x,z)$，构成一个希尔伯特空间，其步骤是，首先定义映射 ϕ 并构成向量空间 $\mathcal{S}$，然后在 $\mathcal{S}$ 上定义内积空间，最后讲 $\mathcal{S}$ 完备化成希尔伯特空间。

定义映射，构成向量空间 $\mathcal{S}$

先定义映射

$$\phi: x\to K(\cdot,x) \tag{4.10}$$

根据这一映射，对任意 $x_i\in\mathcal{X}$，$\alpha_i\in\mathbb{R}$，$i=1,2,\cdots,m$，定义线性组合

$$f(\cdot)=\sum_{i=1}^{m}\alpha_i K(\cdot,x_i) \tag{4.11}$$

$\mathcal{S}$ 构成一个向量空间。

在 $\mathcal{S}$ 上定义内积，使其成为内积空间

在 $\mathcal{S}$ 定义一个运算 $*$，使得对于任意的 $f,g\in\mathcal{S}$，

$$f*g=\sum_{i=1}^{m}\sum_{j=1}^{l}\alpha_i\beta_j K(x_i,z_j) \tag{4.12}$$

其中

$$f(\cdot)=\sum_{i=1}^{m}\alpha_i K(\cdot,x_i) \tag{4.13}$$

$$g(\cdot)=\sum_{j=1}^{l}\beta_i K(\cdot,z_i) \tag{4.14}$$

只需要验证 $*$ 是 $\mathcal{S}$ 上的内积

proposition 4.4.1: 运算 $*$ 是 $\mathcal{S}$ 上的内积。[1]

proof.

4.5. 将内积空间 $\mathcal{S}$ 完备化成希尔伯特空间

定义内积的范数

$$\|f\|=\sqrt{f\cdot f} \tag{4.15}$$

则 $\mathcal{S}$ 变成一个赋范线性空间，对于不完备的赋范线性空间 $\mathcal{S}$，一定可以使之完备化，得到完备的赋范线性空间 $\mathcal{H}$。一个内积空间，当作为一个赋范空间是完备的时候，就是希尔伯特空间。

[1] 内积的性质：(1) 线性性；(2) 共轭对称；(3) 正定性

再生核希尔伯特空间

这一希尔伯特空间 $\mathcal{H}$ 称为**再生核希尔伯特空间** *(reproducing kernel Hilbert sapce,RKHS)*，这是由于核 K 的再生性。

definition 4.5.1: （再生核）满足

$$K(\cdot, x) \cdot f = f(x) \tag{4.16}$$

以及

$$K(\cdot, x) \cdot K(\cdot, z) = K(x, z) \tag{4.17}$$

称为**再生核**

正定核的充要条件

theorem 4.5.1: 设 $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ 是对称函数，则 $K(x, z)$ 为正定核的充要条件是：对任意的 $x_i \in \mathcal{X}, i = 1, 2, \cdots, m$，$K(x, z)$ 对应的 *Gram* 矩阵

$$K = [K(x_i, x_j)]_{m \times m} \tag{4.18}$$

是半正定矩阵。

正定核的等价定义

definition 4.5.2: 设 $\mathcal{X} \in \mathbb{R}^n$, $K(x, z)$ 是定义在 $\mathcal{X} \times \mathcal{X}$ 上的对称函数，如果对任意的 $x_i \in \mathcal{X}, i = 1, 2, \cdots, m,$，$K(x, z)$ 对应的 *Gram* 矩阵

$$K = [K(x_i, x_j)]_{m \times m} \tag{4.19}$$

是半正定矩阵，则 $K(x, z)$ 是正定核。

这一定义在构造核函数的时候很有用，但是对于一个具体函数，检验他是核函数其实并不容易，在实际问题中往往应用已有的核函数。*Mercer* 定理可以得到 *Mercer* 核，正定核比 Mercer 核更具有一般性。

4.6. 常用的核函数

多项式核函数

$$K(x, z) = (x \cdot z + 1)^p \tag{4.20}$$

对应的支持向量机是一个 p 次多项式分类器。在此情形下，分类决策函数称为

$$f(x) = sign(\sum_{i=1}^{N_s} \alpha_i^* y_i (x_i \cdot x + 1)^p + b^*) \tag{4.21}$$

高斯核函数

$$K(x,z)=exp(-\frac{\|x-z\|^2}{2\sigma^2}) \tag{4.22}$$

对应的支持向量机是一个**高斯径向基函数** *(radial basis function)*。在此情形下，分类决策函数称为

$$f(x)=sign(\sum_{i=1}^{N_s}\alpha_i^* y_i exp(-\frac{\|x-z\|^2}{2\sigma^2})+b^*) \tag{4.23}$$

字符串核函数

字符串核是定义在字符串集合上的核函数，字符串核函数在文本分类，信息检索，生物信息学方面有所应用。

4.7. 非线性支持向量分类机算法

definition 4.7.1: 从非线性分类训练集，通过核函数与软间隔最大化，或者凸二次规划学习到分类决策函数

$$f(x)=sign(\sum_{i=1}^{N}\alpha_i^* y_i K(x,x_i)+b^*) \tag{4.24}$$

称为非线性支持向量机，$K(x,z)$ 是正定核函数。

下面叙述非线性支持向量机学习算法

非线性支持向量机学习算法

输入：训练数据集 $T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$；输出：分类决策函数

1. 选取适当核函数 $K(x,z)$ 和适当的参数 C，构造并求解最优化问题

$$\min_{\alpha} \quad \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i\cdot x_j)K(x_i,x_j)-\sum_{i=1}^{N}\alpha_i \tag{4.25}$$

$$s.t. \quad \sum_{i=1}^{N}\alpha_i y_i=0 \tag{4.26}$$

$$0\leqslant\alpha_i\leqslant C,\ \ i=1,2,\cdots,N \tag{4.27}$$

求得最优解 α^*。

2. 选择 α^* 的一个正分量 $0<\alpha_j^*<C$，计算

$$b^*=y_j-\sum_{i=1}^{N}\alpha_i^* y_i K(x_i,x_j) \tag{4.28}$$

3. 构造决策函数

$$f(x)=sign(\sum_{i=1}^{N}\alpha_i^* y_i K(x,x_i)+b^*) \tag{4.29}$$

当 $K(x,z)$ 是正定核函数时，最优化问题时凸二次规划问题，解是存在的。

Chapter 5

序列最小最优化算法 (SMO)

本节讨论支持向量机的实现问题，支持向量机的学习问题可以形式华为求解凸二次规划问题，这样的凸二次规划问题具有全局最优解。

5.1. SMO 算法

SMO 算法要解如下凸二次规划的对偶问题

$$\min_{\alpha} \quad \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j)K(x_i, x_j) - \sum_{i=1}^{N}\alpha_i \tag{5.1}$$

$$s.t. \quad \sum_{i=1}^{N}\alpha_i y_i = 0 \tag{5.2}$$

$$0 \leqslant \alpha_i \leqslant C, \ \ i = 1, 2, \cdots, N \tag{5.3}$$

在这个问题中，变量是拉格朗日乘子，一个变量 α_i 对应一个样本点 (x_i, y_i)，变量的总数等于训练样本的容量 N。

SMO 算法是一种启发式算法，基本思路是：如果所有变量的解都满足次最优化问题的 KKT 条件，那么这个最优化问题的解就得到了，因为 KKT 条件是该最优化问题的充分必要条件，否则，选择两个变量，固定其他变量，针对这两个变量构建一个二次规划问题。这个二次规划问题关于这两个变量的解应该更接近原始二次规划问题的解。因为这会使得原始二次规范问题的目标函数值变小，重要的是，这时子问题可以通过解析方法求解，这样就可以大大提高整个算法的计算速度。

子问题有两个变量，一个违反 KKT 条件最严重的那一个，另一个由约束条件自动确定。如此 SMO 算法将原问题不断分解为子问题并对子问题求解，进而达到求解原问题的目的。

5.2. 两个变量的二次规划的求解方法

不是一般性，假设选择的两个变量是 $alpha_1$,$alpha_2$，其他变量 $alpha_i(i=3,4,\cdots,N)$ 是固定的，于是 SMO 的最优化问题可以写成

$$\min_{\alpha_1,\alpha_2} \quad W(\alpha_1,\alpha_2)=\frac{1}{2}K_{11}\alpha_1^2+\frac{1}{2}K_{22}\alpha_2^2- (\alpha_1+\alpha_2)+y_1\alpha_1\sum_{i=3}^{N}y_i\alpha_iK_{i1}+y_2\alpha_2\sum_{i=3}^{N}y_i\alpha_iK_{i2} \tag{5.4}$$

$$s.t. \quad \alpha_1y_1+\alpha_2y_2=-\sum_{i=3}^{N}y_i\alpha_i=\varsigma \tag{5.5}$$

$$0\leqslant\alpha_i\leqslant C, \quad i=1,2 \tag{5.6}$$

其中 $K_{ij}=K(x_i,x_j),i,j=1,2,\cdots,N$，$\varsigma$ 是常数，目标函数式 (5.4) 中省略了不含 α_1,α_2 的常数项。

为了求解两个变量的二次规划问题，首先分析约束条件，然后在此约束条件下求极小。

由于只有两个变量 (α_1,α_2)，约束可以用二维空间中的图形表示

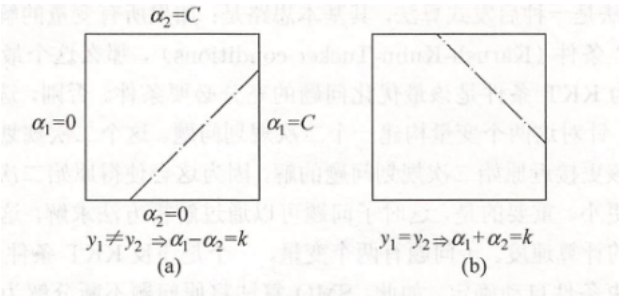

图 5.1: 二变量优化问题

不等式约束 (5.6) 使得 (α_1,α_2) 在盒子 $[0,C]\times[0,C]$ 内，等式约束 (5.6) 使得 (α_1,α_2) 在平行于盒子 $[0,C]\times[0,C]$ 的对角线的直线上，因此要求的是目标函数在一条平行于对角线的线段上的最优值。这使得两个变量的最优化问题称为实质上的单变量的最优化问题。不妨考虑为变量 α_2 的最优化问题。

假设问题的初始可行解为 $\alpha_1^{old},\alpha_2^{old}$，最优解为 $\alpha_1^{new},\alpha_2^{new}$，并且假设在沿着约束方向为经过剪辑时，$\alpha_2$ 的最优解为 $\alpha_2^{new,unc}$。

由于 α_2^{new} 需满足不等式约束 (5.6)，所以最优值 α_2^{new} 的取值范围必须满足条件

$$L\leqslant\alpha_2^{new}\leqslant H \tag{5.7}$$

其中 L 和 H 是 α_2^{new} 所在的对角线段的端点的界。如果 $y_1\neq y_2$，则

$$L=\max(0,\alpha_2^{old}-\alpha_1^{old}), \quad H=\min(C,C+\alpha_2^{new}-\alpha_1^{new}) \tag{5.8}$$

如果 $y_1=y_2$，则

$$L=\max(0,\alpha_2^{old}+\alpha_1^{old}-C), \quad H=\min(C,\alpha_2^{new}+\alpha_1^{new}) \tag{5.9}$$

下面，首先沿着约束方向未经剪辑即为考虑不等式约束 (5.6) 时，α_2 的最优解 $\alpha_2^{new,unc}$; 然后再求剪辑后 α_2 的解 α_2^{new}，我们用定律叙述这个结果，记

$$g(x) = \sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b \tag{5.10}$$

令

$$E_i = g(x_i) - y_i = (\sum_{j=1}^{N} \alpha_j y_j K(x_j, x_i) + b) - y_i, \ \ i = 1, 2 \tag{5.11}$$

当 $i = 1, 2$ 时，E_i 为函数 $g(x)$ 对输入 x_i 的预测值与真实输出 y_i 之差。

定理

theorem 5.2.1: *最优化问题 (5.4)~(5.6) 沿着约束方向未经剪辑时的解时*

$$\alpha_2^{new,unc} = \alpha_2^{old} + \frac{y_2(E_2 - E_1)}{\eta} \tag{5.12}$$

其中，

$$\eta = K_{11} + K_{22} - 2K_{12} = \|\phi(x_1) - \phi(x_2)\|^2 \tag{5.13}$$

$\phi(x)$ 是输入空间到特征空间的映射。

经过剪辑后 α_2 的解是

$$\alpha_2^{new} = \begin{cases} H, & \alpha_2^{new,unc} > H \\ \alpha_2^{new,unc}, & L \leqslant \alpha_2^{new,unc} \leqslant \alpha_2^{new,unc} \\ L, & \alpha_2^{new,unc} \leqslant L \end{cases} \tag{5.14}$$

由 α_2^{new} 求得 α_1^{new} 是

$$\alpha_1^{new} = \alpha_1^{old} + y_1 y_2 (\alpha_2^{old} - \alpha_2^{new}) \tag{5.15}$$

5.3. 变量的选择方法

SMO 算法在每个子问题中选择两个变量优化，其中至少一个变量是违反 KKT 条件的。

第一个变量的选择

SMO 选择第 1 个变量的过程称为*外层循环*。外层循环在训练样本中违反 KKT 条件最严重的样本点。并将其对应的变量作为第 1 个变量。具体地，检验训练样本点是否满足 KKT 条件，即

$$\alpha_i = 0 \Leftrightarrow y_i g(x_i) \geqslant 1 \tag{5.16}$$

$$0 < \alpha_1 < C \Leftrightarrow y_i g(x_i) = 1 \tag{5.17}$$

$$\alpha_i = C \Leftrightarrow y_i g(x_i) \leqslant 1 \tag{5.18}$$

其中，$g(x_i) = \sum_{j=1}^{N} \alpha_j y_j K(x_i, x_j) + b$。

该检验是在 ϵ 范围内进行的。在检验过程中，外层循环首先遍历所有条件 $0 \leqslant \alpha_i < C$ 的样本点，即间隔边界上的支持向量点，检验它们是否满足 KKT 条件。如果这些样本点都满足 KKT 条件，那么遍历整个训练集，检验它们是否满足 KKT 条件。

第二个变量的选择

SMO 选择第 2 个变量的过程称为*内层循环*。假设外层循环中已经找到了第 1 个变量 α_1，现在要在内层循环中要找到第 2 个变量 α_2，第 2 个变量选择的标准死后希望能使得 α_2 有足够大的变化。

计算阈值 b 和差值 E_i

5.4. SMO 算法

输入：训练数据集 $T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_1N,y_N)\}$；
输出：近似解 $\hat{\alpha}$；

(1) 取初值 $\alpha^{(0)}=0$，令 $k=0$；

(2) 选取优化变量 $\alpha_1^{(k)}$, $\alpha_2^{(k)}$, 解析求解两个变量的最优化问题 (5.4)~(5.6)。求得最优解 α_1^{k+1}, α_2^{k+1}，更新 α 为 $\alpha^{(k+1)}$。

(3) 若在精度 ϵ 范围内满足停机条件

$$\sum_{i=1}^{N}\alpha_i y_i=0,\ \ 0\leqslant\alpha_i\leqslant C,\ \ i=1,2,\cdots,N \tag{5.19}$$

$$y_i\cdot g(x_i)=\begin{cases}\geqslant 1, & \{x_i|\alpha_i=0\}\\ =1, & \{x_i|0<\alpha_i<C\}\\ \leqslant 1, & \{x_i|\alpha_i=C\}\end{cases} \tag{5.20}$$

(其中 $g(x_i)=\sum\limits_{j=1}^{N}\alpha_j y_j K(x_j,x_i)$)

则转 (4)，否则令 $k=k+1$，转 (2)。

(4) 取 $\hat{\alpha}=\alpha^{k+1}$

References

[1] I. Surname, I. Surname, and I. Surname. “The Title of the Article”. In: *The Title of the Journal* 1.2 (2000), pp. 123–456.

Chapter A

Karush-Kuhn-Tucker 条件

定理的具体内容在本文的第一节已经有了叙述，这里通过一个例子来描述。

A.1. 例子

考虑下面的优化问题

$$\begin{aligned} &\min\ x_1^2 + x_2^2 \\ &s.t.\ \ x_1 + x_2 = 1 \\ &\qquad x_2 \leqslant \alpha \end{aligned} \tag{A.1}$$

考虑 *Lagrangigan* 函数

$$L(x_1, x_2, \lambda, \mu) = (x_1^2 + x_2^2) - \lambda(x_1 + x_2 - 1) - \mu(x_2 - \alpha) \tag{A.2}$$

令 $\nabla L = 0$ 即可以将上面方程转化为 KKT 条件为

$$\frac{\partial L}{\partial x_i} = 0, \quad i = 1, 2 \tag{A.3}$$

$$x_1 + x_2 = 1 \tag{A.4}$$

$$x_2 - \alpha \leqslant 0 \tag{A.5}$$

$$\mu \geqslant 0 \tag{A.6}$$

$$\mu(x_2 - \alpha) = 0 \tag{A.7}$$

我们的目标是求出上面优化问题的解。

Chapter B

凸优化问题

凸优化问题是指越是最优化问题

$$\begin{aligned} \min_{\omega} \ & f(\omega) \\ s.t. \ & g_i(\omega) \leqslant 0, \quad i = 1, 2, \cdots, k \\ & h_i(\omega) \leqslant 0, \quad i = 1, 2, \cdots, l \end{aligned} \tag{B.1}$$

其中目标函数 $f(\omega)$ 和约束函数 $g_i(\omega)$ 都是 $\mathbb{R}^n$ 上的连续可微的凸函数，约束函数 $h_i(\omega)$ 是 $\mathbb{R}^n$ 上的仿射函数。

当目标函数 $f(\omega)$ 是二次函数且约束函数 $g_i(\omega)$ 是仿射函数时，上述凸最优化问题称为凸二次规划问题。

B.1. 凸二次规划问题解的存在性

Chapter C

内积空间

Chapter D

Cauchy-Schwarz 不等式及其证明

D.1. 定理描述

一般形式

Cauchy-Schwarz 不等式的一般描述如下

theorem D.1.1: 已知 $a_1, \cdots, a_n, b_1 \cdots, b_n$ 是实数，则

$$(\sum_{i=1}^{n} a_i b_i)^2 \leqslant (\sum_{i=1}^{n} a_i^2)(\sum_{i=1}^{n} b_i^2) \tag{D.1}$$

等号成立的充分必要条件是

$$a_i = \lambda b_i, \ \ i = 1, \cdots n \tag{D.2}$$

推广到复数形式

不等式可以推广到复数。如何推广呢？不等式只有在实数时才有意义，对于复数则需要考虑角度和模长大小关系。

theorem D.1.2: 已知 $a_1, \cdots, a_n, b_1 \cdots, b_n$ 是复数，则

$$|\sum_{i=1}^{n} a_i b_i|^2 \leqslant (\sum_{i=1}^{n} |a_i|^2)(\sum_{i=1}^{n} |b_i|^2) \tag{D.3}$$

等号成立的充分必要条件是

$$a_i = \lambda b_i, \ \ i = 1, \cdots n \tag{D.4}$$

矩阵形式

根据线性代数的理论：任意正定对称矩阵都可以定义内积。因此若 $A = a_{ij}$ 为正定对称矩阵，则 $Cauchy$ 不等式存在

theorem D.1.3: 已知 $(a_{ij})_{kl}$ 是正定对称矩阵，对于 $x_1,\cdots,x_n,y_1.\cdots,y_n$ 是任意复数或者实数，则

$$|\sum_{i=1}^{n} a_{ij}x_iy_j| \leqslant \sqrt{\sum_{i,j=1}^{n} a_{ij}x_ix_j}\sqrt{\sum_{i,j=1}^{n} a_{ij}y_iy_j} \tag{D.5}$$

等号成立的充分必要条件是

$$a_i = \lambda b_i,\ \ i = 1,\cdots n \tag{D.6}$$

无穷级数形式

theorem D.1.4: 已知 $a_1,\cdots,a_n,\cdots,b_1\cdots,b_n,\cdots$ 是复数，则

$$|\sum_{i=1}^{\infty} a_ib_i| \leqslant (\sum_{i=1}^{\infty}|a_i|^2)^{\frac{1}{2}}(\sum_{i=1}^{\infty}|b_i|^2)^{\frac{1}{2}} \tag{D.7}$$

等号成立的充分必要条件是

$$a_i = \lambda b_i,\ \ i = 1,\cdots;\ \ \lambda \in \mathbb{C} \tag{D.8}$$

$$\sum_{i=1}^{\infty}|a_i|^2 < \infty \tag{D.9}$$

$$\sum_{i=1}^{\infty}|b_i|^2 < \infty \tag{D.10}$$

积分形式

theorem D.1.5: 已知 f,g 是区间 $[a,b]$ 上的连续函数，$f,g \in C[a,b]$，则

$$|\int_a^b f(x)g(x)dx|^2 \leqslant \int_a^b |f(x)|^2dx\int_a^b |g(x)|^2dx \tag{D.11}$$

Hölder 不等式

theorem D.1.6: 已知 $a_1,\cdots,a_n,b_1\cdots,b_n$ 是复数，$p,q \geqslant 1$，$\frac{1}{p}+\frac{1}{q}=1$，则

$$|\sum_{i=1}^{n} a_ib_i| \leqslant (\sum_{i=1}^{n}|a_i|^p)^{\frac{1}{p}}(\sum_{i=1}^{n}|b_i|^q)^{\frac{1}{q}} \tag{D.12}$$

广义 Cauchy-Schwarz 不等式

theorem D.1.7: 一般 n 维向量空间中的 *Cauchy-Schwarz* 不等式形式为

$$|a\cdot b| \leqslant \|a\|\|b\| \tag{D.13}$$

D.2. 余弦定律

考虑三角形 $\triangle ABC$，三条边的分量分别是

$$\vec{a} = \overrightarrow{AB},\ \vec{b} = \overrightarrow{AC},\ \vec{c} = \overrightarrow{CB} = \vec{a}-\vec{b} \tag{D.14}$$

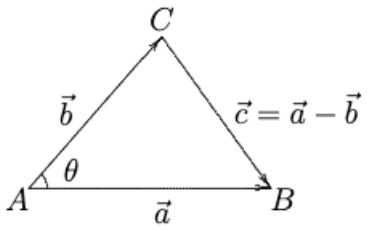

图 D.1: 余弦定律

根据余弦定律 $|\vec{a}|^2+|\vec{b}|^2-|\vec{a}-\vec{b}|^2=2|\vec{a}||\vec{b}|\cos\theta$ 以及余弦的性质 $|\cos\theta|\leqslant 1$，可得

$$|\vec{a}\cdot\vec{b}|\leqslant|\vec{a}||\vec{b}| \tag{D.15}$$

这就是 Cauchy-Schwarz 不等式。他告诉我们 Cauchy-Schwarz 不等式的几何意义：三角形两边的内积小于两个向量长度的乘积。

D.3. 证明

考虑 b 在 a 上的投影之差的最短距离，设

$$\overrightarrow{c}=\vec{b}-\lambda\vec{a},\quad \lambda\in\mathbb{R} \tag{D.16}$$

$\vec{c}$ 的长度

$$|\vec{c}|^2=\vec{c}\cdot\vec{c}=|\vec{a}|^2\lambda^2+2\vec{a}\cdot\vec{b}\lambda+|b|^2>0 \tag{D.17}$$

上式可视为 λ 的二次方程，且与 λ 轴没有交点

$$\triangle=(2\vec{a}\vec{b})^2-4|\vec{a}|^2|\vec{b}|^2\leqslant 0 \tag{D.18}$$

因此有

$$|\vec{a}\cdot\vec{b}|\leqslant|\vec{a}||\vec{b}| \tag{D.19}$$

Chapter E

Source Code Example

Adding source code to your report/thesis is supported with the package `listings`. *An example can be found below. Files can be added using* `\lstinputlisting[language=<language>]{<filename>}`.

```
"""
ISA Calculator: import the function, specify the height and it will return a
list in the following format: [Temperature,Density,Pressure,Speed of Sound].
Note that there is no check to see if the maximum altitude is reached.
"""

import math
g0 = 9.80665
R = 287.0
layer1 = [0, 288.15, 101325.0]
alt = [0,11000,20000,32000,47000,51000,71000,86000]
a = [-.0065,0,.0010,.0028,0,-.0028,-.0020]

def atmosphere(h):
    for i in range(0,len(alt)-1):
        if h >= alt[i]:
            layer0 = layer1[:]
            layer1[0] = min(h,alt[i+1])
            if a[i] != 0:
                layer1[1] = layer0[1] + a[i]*(layer1[0]-layer0[0])
                layer1[2] = layer0[2] * (layer1[1]/layer0[1])**(-g0/(a[i]*R))
            else:
                layer1[2] = layer0[2]*math.exp((-g0/(R*layer1[1]))*(layer1[0]-layer0[0]))
    return [layer1[1],layer1[2]/(R*layer1[1]),layer1[2],math.sqrt(1.4*R*layer1[1])]
```

Chapter F

Task Division Example

If a task division is required, a simple template can be found below for convenience. Feel free to use, adapt or completely remove.

表 F.1: Distribution of the workload

	Task	Student Name(s)
	Summary	
Chapter 1	Introduction	
Chapter 2		
Chapter 3		
Chapter *		
Chapter *	Conclusion	
	Editors	
	CAD and Figures	
	Document Design and Layout	