

# Support Vector Machine

Fish

2019 年 6 月 27 日

# Content

<b>第一章</b>	<b>perception</b>	<b>1</b>
1.1	concept . . . . .	1
1.2	linearly separable dataset . . . . .	1
1.3	loss function . . . . .	1
1.4	original form of perception algorithm . . . . .	2
1.5	dual form of perception algorithm . . . . .	3
<b>第二章</b>	<b>Convex function</b>	<b>5</b>
2.1	Convex collection and Convex function . . . . .	5
2.1.1	Convex function . . . . .	5
2.1.2	Convex collection . . . . .	5
2.2	affine transformation . . . . .	6
2.2.1	concept . . . . .	6
2.2.2	theory . . . . .	6
2.3	perspective collineation . . . . .	7
2.3.1	concept . . . . .	7
2.3.2	direct significance . . . . .	7
2.3.3	summary . . . . .	7
2.4	segmentation plane . . . . .	7
2.4.1	concept . . . . .	7
2.4.2	segmentation plane . . . . .	8
2.4.3	the structure of segmentation plane . . . . .	8
2.4.4	support hyperplane . . . . .	9
2.4.5	question . . . . .	9
2.5	property of convex function . . . . .	10
2.5.1	concept . . . . .	10

2.5.2	first-order derivative of convex function . . . . .	10
2.5.3	second-order derivative of convex function . . . . .	10
2.6	examples of convex function . . . . .	11
<b>第三章</b>	<b>Convex Optimization</b>	<b>12</b>
3.1	optimization problem . . . . .	12
3.2	convex optimization . . . . .	13
3.2.1	basic form . . . . .	13
3.2.2	Lagrange 乘子法 . . . . .	13
3.2.3	Lagrange 函数的极小极大最优值 $\min_x \max_{\lambda, \nu: \lambda_i \geq 0} L(x, \lambda, \nu)$ 和 原问题的最优值 $p^*$ . . . . .	14
3.2.4	Lagrange 对偶函数 (dual function) . . . . .	15
<b>第四章</b>	<b>SVM</b>	<b>19</b>
4.1	linear separable support vector machine and hard margin maxi- mization . . . . .	19
4.1.1	question? . . . . .	19
4.1.2	input data . . . . .	19
4.1.3	linear separable support vector machine . . . . .	20
4.1.4	list symbol . . . . .	20
4.1.5	derivation target function . . . . .	20
4.1.6	Linear divisible support Vector Machine Learning algorithm	25
4.2	linear support vector machine and soft margin maximization . .	27
4.2.1	concept . . . . .	27
4.2.2	derivation target function . . . . .	28
4.2.3	dual property . . . . .	28
4.2.4	Linear support vector machine learning algorithm . . .	29

# 第一章 perception

## 1.1 concept

1. 感知机是二类分类的线性分类模型, 其输入空间为实例的特征向量, 输出为实例的类别, 取  $+1$  和  $-1$  二值.
2. 感知机对应于输入控件 ((特征空间) 中将实例划分为正负两类的分离超平面, 是一种线性分类模型, 属于判别模型.
3. 感知机学习旨在求出将训练数据进行线性划分的分离超平面.
4. 导入基于误分类的损失函数, 利用梯度下降法对损失函数进行极小化, 求得感知机模型.
5.  $f(x) = \text{sign}(w \cdot x + b)$

## 1.2 linearly separable dataset

给定数据集  $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ , 其中,  $x_i \in \chi = \mathbb{R}^n$ ,  $y_i \in \gamma = \{+1, -1\}$ ,  $i = 1, 2, \dots, N$ .

如果存在某个超平面  $S: w \cdot x + b = 0$  将数据集的正负实例点完全正确划分到超平面的两侧:

对所有  $y_i = +1$  的实例  $i$ , 有  $w \cdot x_i + b > 0$

对所有  $y_i = -1$  的实例  $i$ , 有  $w \cdot x_i + b < 0$

则数据集  $T$  为线性数据可分数据集, 否则称数据集  $T$  线性不可分

## 1.3 loss function

假设训练数据集是线性可分的, 为了找出可将训练集正负实例点完全正确分开的分离超平面, 即确定参数  $w, b$ , 需要确定一个学习策略, 即定义 (经验) 损失函数并将损失函数极小化

两种选择:

1. 选择误分类点的总数, 但这样的损失函数不是参数  $w, b$  的连续可导函数, 不易优化
2. 选择误分类点到超平面  $S$  的总距离

定义输入空间  $\mathbb{R}^n$  中任一点  $x_0$  到超平面  $S$  的距离:

$$\frac{1}{\|w\|} |w \cdot x_0 + b| \quad (1.1)$$

这里  $\|w\|$  是  $w$  的  $L_2$  范数.

对于误分类的数据  $(x_i, y_i)$  来说:

$$-y_i(w \cdot x_i + b) > 0 \quad (1.2)$$

成立.

因此, 误分类点  $x_i$  到超平面  $S$  的距离是

$$-\frac{1}{\|w\|} y_i(w \cdot x_i + b) \quad (1.3)$$

这样, 假设超平面  $S$  的误分类点集合为  $M$ , 则所有误分类点到超平面  $S$  的总距离为

$$-\frac{1}{\|w\|} \sum_{x_i \in M} y_i(w \cdot x_i + b) \quad (1.4)$$

不考虑  $\frac{1}{\|w\|}$ , 得到感知机  $\text{sign}(w \cdot x + b)$  在训练集  $T$  的损失函数定义为

$$L(w, b) = - \sum_{x_i \in M} y_i(w \cdot x_i + b) \quad (1.5)$$

#### 1.4 original form of perception algorithm

关键: 感知机算法是误分类驱动的, 当误分类点位于分离超平面的错误一侧时, 调整  $w, b$  的值, 使分离超平面向该误分类点的一侧移动, 以减少该误分类点与超平面间的距离, 直至超平面越过该误分类点使其被正确分类.

输入: 训练数据集  $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ , 其中,  $x_i \in \chi = \mathbb{R}^n$ ,  $y_i \in \gamma = \{+1, -1\}$ ,  $i = 1, 2, \dots, N$ ; 学习率  $\eta$  ( $0 < \eta \leq 1$ ).

输出:  $w, b$ ; 感知机模型  $f(x) = \text{sign}(w \cdot x + b)$ .

1. 选取初值  $w_0, b_0$

2. 在训练集中选取数据  $(x_i, y_i)$

3. 如果  $y_i(w_i + b) \leq 0$

(a) 损失函极小化  $\min_{w,b} L(w, b) = - \sum_{x_i \in M} y_i(w \cdot x_i + b)$

(b) 梯度

$$\nabla_w L(w, b) = - \sum_{x_i \in M} y_i x_i \quad (1.6)$$

$$\nabla_b L(w, b) = - \sum_{x_i \in M} y_i \quad (1.7)$$

(c) 随机选取一个误分类点  $(x_i, y_i)$ , 更新参数  $w, b$

$$w \leftarrow w + \eta y_i x_i \quad (1.8)$$

$$b \leftarrow b + \eta y_i \quad (1.9)$$

4. 转至 (2), 直至训练集中没有误分类点。

### 1.5 dual form of perception algorithm

基本思路: 将  $w$  和  $b$  表示为实例  $x_i$  和标记  $y_i$  的线性组合的形式, 通过求解其系数而求得  $w$  和  $b$ .

输入: 训练数据集  $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ , 其中,  $x_i \in \chi = \mathbb{R}^n$ ,  $y_i \in \gamma = \{+1, -1\}$ ,  $i = 1, 2, \dots, N$ ; 学习率  $\eta$  ( $0 < \eta \leq 1$ ).

输出:  $\alpha, b$ ; 感知机模型  $f(x) = \text{sign} \left( \sum_{j=1}^N \alpha_j y_j x_j \cdot x + b \right)$

其中,  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_N)^T$ , 注意这里是一个向量, 所以后面更新时,  $\alpha$  直接定义步长更新

1.  $\alpha \leftarrow 0, b \leftarrow 0$

2. 在训练集中选取数据  $(x_i, y_i)$

3. 如果  $y_i \left( \sum_{j=1}^N \alpha_j y_j x_j \cdot x_i + b \right)$

(a) 损失函极小化  $\min_{w,b} L(w, b) = \min_{\alpha,b} L(\alpha, b) = - \sum_{x_i \in M} y_i (\alpha_j y_j x_j \cdot x_i + b)$

(b) 梯度

$$\nabla_b L(w, b) = - \sum_{x_i \in M} y_i \quad (1.10)$$

(c) 随机选取一个误分类点  $(x_i, y_i)$ , 更新参数  $w, b$

$$\alpha \leftarrow \alpha + \eta \quad (1.11)$$

$$b \leftarrow b + \eta y_i \quad (1.12)$$

4. 转至 (2) 直到没有误分类数据.

注意: 对偶形式中训练实例仅以内积的形式出现.

为了方便, 预先将训练集中实例间的内积计算出来并以矩阵的形式存储, 即 Gram 矩阵

$$G = [x_i \cdot x_j] \quad (1.13)$$

## 第二章 Convex function

### 2.1 Convex collection and Convex function

#### 2.1.1 Convex function

$y = x^2$  是凸函数, 函数图像上位于  $y = x^2$  上方的区域构成凸集.

1. 凸函数图像的上方区域, 一定是凸集
2. 一个函数图像的上方区域为凸集, 则该函数是凸函数

#### 2.1.2 Convex collection

1. Affine set 仿射集

(a) define: 通过集合  $C$  中任意两个不同点的直线仍然在集合  $C$  内, 则称集合  $C$  为仿射集.

$$\forall x_1, x_2 \in C, \forall \theta \in R, \text{ 则 } x = \theta \cdot x_1 + (1 - \theta) \cdot x_2 \in C \quad (2.1)$$

(b) 仿射集的例子: 直线、平面、超平面

$n$  维空间的  $n - 1$  维仿射集为  $n - 1$  维超平面

2. Convex 凸集

两种表述 (可以思考其内涵一样吗):

(a) 集合  $C$  内任意两点间的线段均在集合  $C$  内, 则称集合  $C$  为凸集.

$$\forall x_1, x_2 \in C, \forall \theta \in [0, 1], \text{ 则 } x = \theta \cdot x_1 + (1 - \theta) \cdot x_2 \in C \quad (2.2)$$

(b)  $k$  个点的版本:

$$\forall x_1, x_2, \dots, x_k \in C, \theta_i \in [0, 1] \text{ 且 } \sum_{i=1}^k \theta_i x_i \in C \quad (2.3)$$



3. 因为仿射集的条件比凸集的条件强, 所以, 仿射集必然是凸集

4. judgement of convex collection

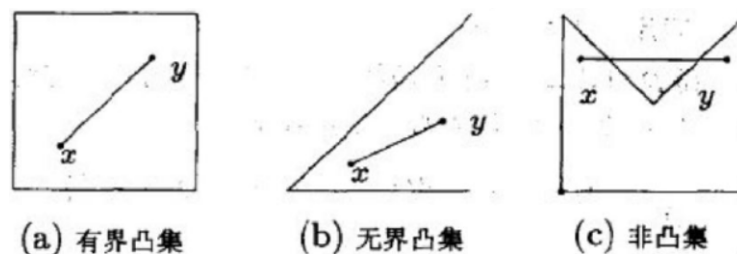


Fig 2.1: convex collection

## 2.2 affine transformation

### 2.2.1 concept

函数  $f(x) = Ax + b$  的形式, 称函数是仿射的: 即线性函数加常数的形式

### 2.2.2 theory

1. 仿射变换  $f(x) = Ax + b$ ,  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$

(a) 伸缩、平移、投影

2. 若  $f$  是仿射变换,  $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$   $f(S) = \{f(x)|x \in S\}$

(a) 若  $S$  为凸集, 则  $f(S)$  为凸集;

(b) 若  $f(S)$  为凸集, 则  $S$  为凸集.

3. 两个凸集的和为凸集

$$S_1 + S_2 = \{x + y | x \in S_1, y \in S_2\} \quad (2.4)$$

4. 两个凸集的笛卡尔积 (直积) 为凸集

$$S_1 \times S_2 = \{(x_1, x_2) | x_1 \in S_1, x_2 \in S_2\} \quad (2.5)$$

5. 两个集合的部分和为凸集 (分配率)

$$S = \{(x, y_1 + y_2) | (x, y_1) \in S_1, (x, y_2) \in S_2\} \quad (2.6)$$

## 2.3 perspective collineation

### 2.3.1 concept

透视函数对向量进行伸缩 (规范化), 使得最后一维的分量为 1 并舍弃之.

$$P: \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n, P(z, t) = z/t \quad (2.7)$$

### 2.3.2 direct significance

小孔成像

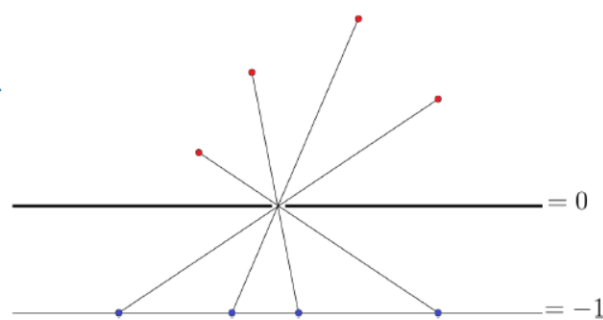


Fig 2.2: significance of perspective collineation

### 2.3.3 summary

凸集的透视变换仍然是凸集

## 2.4 segmentation plane

### 2.4.1 concept

设  $C$  和  $D$  为两不相交的凸集, 则存在超平面  $P$ ,  $P$  可以将  $C$  和  $D$  分离.

$$\forall x \in C, a^T x \leq b \text{ 且 } \forall x \in D, a^T x \geq b \quad (2.8)$$

### 2.4.2 segmentation plane

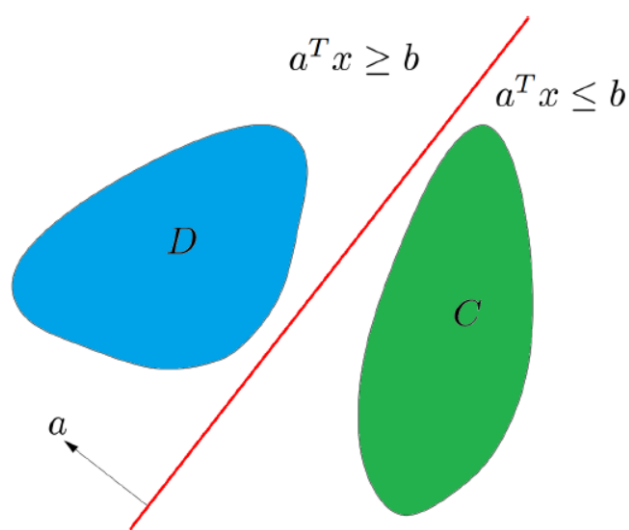


Fig 2.3: segmentation plane

### 2.4.3 the structure of segmentation plane

1. 两个集合的距离, 定义为两个集合间元素的最短距离
2. 做集合  $C$  和集合  $D$  最短线段的垂直平分线
3. picture

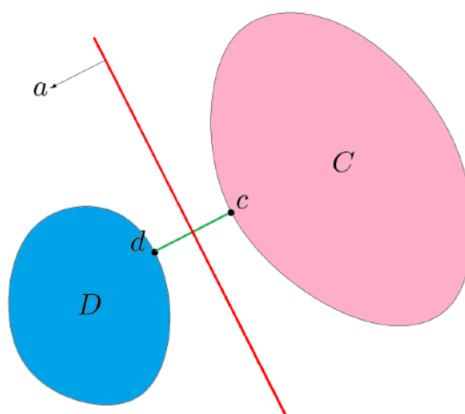


Fig 2.4: the distance of segmentation plane

### 2.4.4 support hyperplane

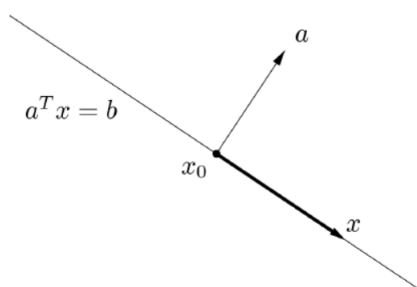


Fig 2.5: hyperplane

1. 设集合  $C$ ,  $x_0$  为  $C$  边界上的点. 若存在  $a \neq 0$ , 满足对任意  $x \in C$ , 都有  $a^T x \leq a^T x_0$  成立, 则称超平面  $\{x | a^T x = a^T x_0\}$  为集合  $C$  在点  $x_0$  处的支撑超平面.
2. 凸集边界上任意一点, 均存在支撑超平面
3. 反之, 若在一个闭的非中空 (内部点不为空) 集合, 在边界上的任意一点存在支撑超平面, 则该集合为凸集

### 2.4.5 question

1. 如何定义两个集合的“最优”分割超平面?
  - (a) 找到集合“边界”上的若干点, 以这些点为“基础”计算超平面的方向; 以两个集合边界上的这些点的平均作为超平面的“截距”
  - (b) 支持向量: support vector
  - (c) optimal segmentation plane

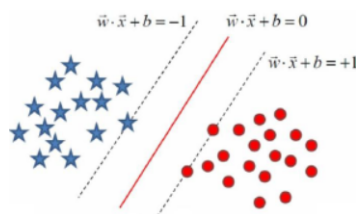


Fig 2.6: optimal segmentation plane

2. 若两个集合有部分相交, 如何定义超平面, 使得两个集合“尽量”分开?
  - (a) 注: 上述“集合”不一定是凸集, 可能是由若干离散点组成. 若一组集合为  $(x, 1)$ , 另一组集合为  $(x, 2)$ , 则为机器学习中的分类问题.

## 2.5 property of convex function

### 2.5.1 concept

若函数  $f(x)$  的定义域  $\text{dom} f$  为凸集, 且满足

$$\forall x, y \in \text{dom } f, 0 \leq \theta \leq 1, \text{ 有} \quad (2.9)$$

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y) \quad (2.10)$$



Fig 2.7: convex function

### 2.5.2 first-order derivative of convex function

若  $f(x)$  一阶可微, 则函数  $f$  为凸函数当前仅当  $f$  的定义域  $\text{dom} f$  为凸集, 且

$$\forall x, y \in \text{dom } f, f(y) \geq f(x) + \nabla f(x)^T(y - x) \quad (2.11)$$

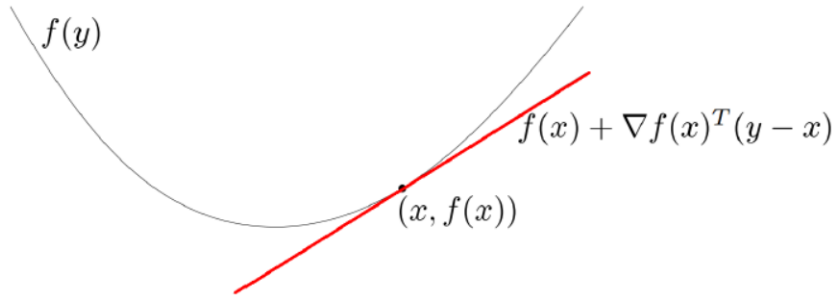


Fig 2.8: first-order derivative of convex function

对于凸函数, 其一阶 Taylor 近似本质上是该函数的全局下估计

### 2.5.3 second-order derivative of convex function

1. 若函数  $f$  二阶可微, 则函数  $f$  为凸函数当前仅当  $\text{dom}$  为凸集, 且

$$\nabla^2 f(x) \succeq 0 \quad (2.12)$$

2. 若  $f$  是一元函数, 上式表示二阶导大于等于 0
3. 若  $f$  是多元函数, 上式表示二阶导 Hessian 矩阵半正定

## 2.6 examples of convex function

- 指数函数  $f(x) = e^{ax}$
- 幂函数  $f(x) = x^a, x \in R^+, a \geq 1$  或  $a \leq 0$
- 负对数函数  $f(x) = -\ln x$
- 负熵函数  $f(x) = x \ln x$
- 范数函数  $f(\vec{x}) = \|x\|$
- 最大值函数  $f(\vec{x}) = \max(x_1, x_2, \dots, x_n)$
- 指数线性函数  $f(\vec{x}) = \log(e^{x_1} + e^{x_2} + \dots + e^{x_n})$

## 第三章 Convex Optimization

### 3.1 optimization problem

#### 1. common format

$$\begin{aligned}
 & \text{minimize } f_0(x), x \in \mathbf{R}^n \\
 & \text{subject to } g_i(x) \leq 0, \quad i = 1, \dots, m \\
 & \quad \quad \quad h_j(x) = 0, \quad j = 1, \dots, p \\
 & \text{优化变量 } x \in \mathbf{R}^n \\
 & \text{不等式约束 } g_i(x) \leq 0 \\
 & \text{等式约束 } h_j(x) = 0 \\
 & \text{无约束优化 } m = p = 0
 \end{aligned} \tag{3.1}$$

#### 2. domain of optimal problem

$$D = \bigcap_{i=1}^m \text{dom} g_i \cap \bigcup_{j=1}^p \text{dom} h_j \tag{3.2}$$

#### 3. feasible dot (solution)

- $x \in D$ , 满足式 3.1 中约束条件

#### 4. feasible domain (feasible collection)

- 所有可行点的集合

#### 5. optimization value

$$p^* = \inf\{f_0(x) | g_i(x) \leq 0, i = 1, \dots, m, h_j(x) = 0, j = 1, \dots, p\} \tag{3.3}$$

#### 6. optimization solution

$$p^* = f_0(x^*) \tag{3.4}$$

## 3.2 convex optimization

### 3.2.1 basic form

$$\begin{aligned} & \text{minimize} && f_0(x), x \in \mathbb{R}^n \\ & \text{subject to} && g_i(x) \leq 0, i = 1, \dots, m \\ & && h_j(x) = 0, j = 1, \dots, p \end{aligned} \quad (3.5)$$

- among,  $g_i(x)$  为凸函数,  $h_j(x)$  为仿射函数
- important property of convex optimization problem

- 凸优化问题的可行域为凸集
- 凸优化问题的局部最优解即为 **全局最优解**

### 3.2.2 Lagrange 乘子法

在支持向量机模型 (SVM) 的推导中一步很关键的就是利用 Lagrange 对偶性将原问题转化为对偶问题.

- theory:

一般的求极值问题, 求导等于 0. 但是如果不但要求极值, 还要求一个满足一定约束条件的极值, 那么此时就可以构造 Lagrange 函数, 其实就是 **把约束项添加到原函数上, 然后对构造的新函数求导**

- example picture

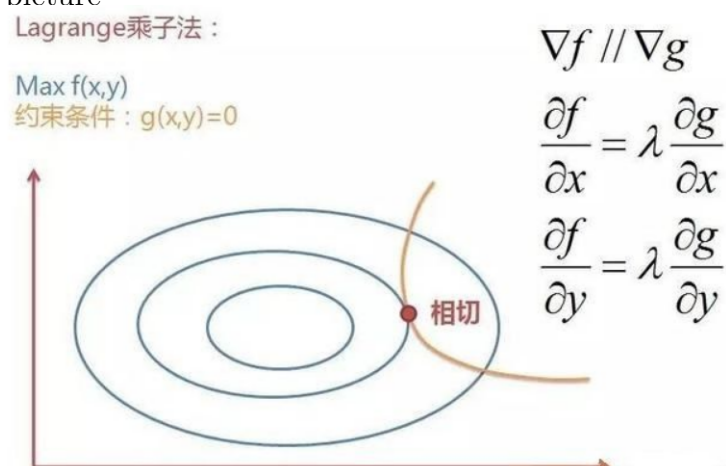


Fig 3.1: Lagrange multiplier method



## 3. analysis:

对于一个要求极值的函数  $f(x, y)$ , 图上的蓝圈就是这个函数的等高图, 就是说  $f(x, y) = c_1, c_2, \dots, c_n$  分别代表不同的数值 (每个值代表一圈, 等高图), 我要找到一组  $(x, y)$ , 使它的  $c_i$  值越大越好, 但是这点必须满足约束条件  $g(x, y)$  (在黄线上)

## 4. conclusion:

就是说  $f(x, y)$  和  $g(x, y)$  相切, 或者说它们的梯度  $\nabla f$  和  $\nabla g$  平行, 因此它们的梯度 (偏导) 成倍关系; 那我们就假设为  $\lambda$  倍, 然后把约束条件加到原函数后再对它求导, 其实就等于满足了图上的式子了.

3.2.3 Lagrange 函数的极小极大最优值  $\min_x \max_{\lambda, \nu: \lambda_i \geq 0} L(x, \lambda, \nu)$  和原问题的最优值  $p^*$ 

## 1. 一般优化问题的 Lagrange 乘子法

$$\begin{aligned} & \text{minimize} && f_0(x), x \in \mathbb{R}^n \\ & \text{subject to} && g_i(x) \leq 0, i = 1, \dots, m \\ & && h_j(x) = 0, j = 1, \dots, p \end{aligned} \quad (3.6)$$

## 2. Lagrange 函数

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^m \lambda_i g_i(x) + \sum_{j=1}^p \nu_j h_j(x) \quad (3.7)$$

对固定的  $x$ , Lagrange 函数  $L(x, \lambda, \nu)$  为关于  $\lambda$  和  $\nu$  的仿射函数, 这里,  $x = (x^1, x^2, \dots, x^n)^T \in \mathbb{R}^n$ ,  $\lambda, \nu$  是 Lagrange 乘子,  $\lambda \geq 0$

3. Lagrange 函数的极小极大最优值  $\min_x \max_{\lambda, \nu: \lambda_i \geq 0} L(x, \lambda, \nu)$  和原问题的最优值  $p^*$  之间的关系

考虑  $x$  的函数:

$$f(x) = \max_{\lambda, \nu: \lambda \geq 0} L(x, \lambda, \nu) \quad (3.8)$$

即原问题  $f(x)$  与 Lagrange 函数在乘子变量最大化时的值是等价的

4. 证明上式: 假设给定某个  $x$ , 如果  $x$  违反原始问题的约束条件, 即存在某个  $i$  使得  $g_i(x) > 0$  或者存在某个  $j$  使得  $h_j(x) \neq 0$ , 那么就有

$$\phi_P(x) = \max_{\lambda, \nu: \lambda \geq 0} \left[ f(x) + \sum_{i=1}^m \lambda_i g_i(x) + \sum_{i=1}^p \nu_i h_i(x) \right] = +\infty \quad (3.9)$$

这里, 下标 P 表示原始问题

5. 因为若某个  $i$  使约束  $g_i(x) > 0$ , 则可令  $\lambda_i \rightarrow +\infty$ , 若某个  $j$  使  $h_j(x) \neq 0$ , 则可令  $\nu_j$  使  $\nu_j h_j(x) \rightarrow +\infty$ , 而将其余各  $\lambda_i, \nu_j$  均取为 0

如果  $x$  满足约束条件, 则由式 3.7 和式 3.8 可知,  $\phi_P(x) = f(x)$

6. summary:

$$\phi_P(x) = \begin{cases} f(x), & x \text{ 满足原始问题约束} \\ +\infty, & \text{其他} \end{cases} \quad (3.10)$$

所以如果考虑极小化问题

$$\min_x \phi_P(x) = \min_x \max_{\lambda, \nu: \lambda \geq 0} L(x, \lambda, \nu) \quad (3.11)$$

它是与原始最优化问题 3.1 等价的, 即它们有相同的解

此时将原始最优化问题表示为广义 Lagrange 函数的极小极大问题.

为了方便, 定义原始问题的最优值

$$p^* = \min_x \phi_P(x) = \min f(x) \quad (3.12)$$

称为原始问题的值

### 3.2.4 Lagrange 对偶函数 (dual function)

1. Lagrange 对偶函数, 注意此处 下式中  $x$  需不需要改成  $\tilde{x}$

$$\Gamma(\lambda, \nu) = \inf_{x \in D} L(x, \lambda, \nu) = \inf_{x \in D} (f_0(x) + \sum_{i=1}^m \lambda_i g_i(x) + \sum_{i=1}^p \nu_i h_i(x)) \quad (3.13)$$

2. 若  $\tilde{x} \in \mathbb{D}$  为主问题 3.1 可行域中的点, 则对任意  $\nu$  和  $\lambda \succeq 0$  都有

$$\sum_{i=1}^m \lambda_i g_i(x) + \sum_{i=1}^n \nu_j h_j(x) \leq 0 \quad (3.14)$$

进而有

$$\Gamma(\lambda, \nu) = \inf_{x \in D} L(x, \lambda, \nu) \leq \inf L(\tilde{x}, \lambda, \nu) \leq f(\tilde{x}) \quad (3.15)$$

3. 若没有下确界, 定义 主问题下界:

$$\Gamma(\lambda, \nu) = -\infty \quad (3.16)$$

4. 根据定义, 显然有: 对  $\forall \lambda > 0, \forall \nu$ , 若原优化问题有最优值  $p^*$ , 则

$$\Gamma(\lambda, \nu) \leq p^* \quad (3.17)$$

即对偶函数给出了主问题的最优值的下界, 显然, 这个下界取决于  $\lambda$  和  $\nu$  的值. 于是, 一个很自然的问题是: 基于对偶函数能获得的最好下界是什么?

5. 上述引出了 **对偶优化问题**

$$\max \Gamma(\lambda, \nu) \quad \text{s.t. } \lambda \succeq 0 \quad (3.18)$$

式 3.18 就是主问题 3.1 的对偶问题, 其中  $\lambda$  和  $\nu$  称为“对偶变量” (dual variable)

6. 进一步: 无论主问题的 3.1 的凸性如何, 对偶问题 Lagrange 对偶函数 3.18 始终是凹函数

7. 假设式 3.18 的最优值为  $d^* = \max_{\lambda, \nu} \Gamma(\lambda, \nu)$ , 也即主问题 3.1 中最优值  $p^*$  的下界

8. 显然有  $d^* = \max_{\lambda, \nu} \Gamma(\lambda, \nu) \leq p^*$ , 这称为“弱对偶性” (weak duality) 成立;

若  $d^* = \max_{\lambda, \nu} \Gamma(\lambda, \nu) = p^*$ , 则称为“强对偶性” (strong duality) 成立, 此时由对偶问题能获得主问题的最优下界.

9. 对于一般优化问题, 强对偶性通常不成立. 但是, 若主问题为凸优化问题, 如式 3.1 中  $f(x)$  和  $g_j(x)$  均为凸函数,  $h_i(x)$  为仿射函数, 且其可行域中至少有一点使不等式约束严格成立, 此时强对偶性成立. 注意: 在强对偶性成立时, 将 Lagrange 函数对原变量求导, 并令导数等于 0, 即可得到原变量与对偶变量的数值关系.

10. 上述分析, 若满足强对偶性的 KKT 条件, 此时:

$$\begin{aligned} d^* &= \max_{\lambda, \nu: \lambda_i \geq 0} \min_x L(x, \lambda, \nu) = \min_x \max_{\lambda, \nu: \lambda_i \geq 0} L(x, \lambda, \nu) \\ &= p^* = \min f(x) \quad \{x \text{ subject to 约束条件}\} \end{aligned} \quad (3.19)$$

11. prove:

若原始问题和对偶问题都有最优值, 则  $d^* = \max_{\lambda, \nu: \lambda_i \geq 0} \min_x L(x, \lambda, \nu) \leq \min_x \max_{\lambda, \nu: \lambda_i \geq 0} L(x, \lambda, \nu) = p^* = \min f(x) \quad \{x \text{ subject to 约束条件}\}$

process:

对任意的  $\lambda, \nu$  和  $x$ , 有

$$L(x, \lambda, \nu) \leq \max_{\lambda, \nu: \lambda \geq 0} L(x, \lambda, \nu) \quad (3.20)$$

$$\therefore \Gamma_D(\lambda, \nu) = \min_x L(x, \lambda, \nu) \leq L(x, \lambda, \nu) \leq \max_{\lambda, \nu: \lambda \geq 0} L(x, \lambda, \nu) = \phi_P(x) \quad (3.21)$$

‘即:

$$\Gamma_D(\lambda, \nu) \leq \phi_P(x) \quad (3.22)$$

由于原始问题和对偶问题均有最优值

$$\therefore \max_{\lambda, \nu: \lambda \geq 0} \Gamma_D(\lambda, \nu) \leq \min_x \phi_P(x) \quad (3.23)$$

即:

$$\begin{aligned} d^* &= \max_{\lambda, \nu: \lambda_i \geq 0} \min_x L(x, \lambda, \nu) \leq \min_x \max_{\lambda, \nu: \lambda_i \geq 0} L(x, \lambda, \nu) \\ &= p^* = \min f(x) \quad \{x \text{ subject to 约束条件} \} \end{aligned} \quad (3.24)$$

12. KKT 条件:

$$g_i(x^*) \leq 0, \quad i = 1, 2, \dots, k \quad (3.25)$$

$$\lambda_i^* \geq 0, \quad i = 1, 2, \dots, k \quad (3.26)$$

$$\lambda_i^* g_i(x^*) = 0, \quad i = 1, 2, \dots, k \quad (3.27)$$

$$h_j(x^*) = 0, \quad j = 1, 2, \dots, l \quad (3.28)$$

$$\nabla L(x^*, \lambda^*, \nu^*) = \nabla f(x^*) + \sum_{i=1}^k \lambda_i^* \nabla g_i(x^*) + \sum_{j=1}^l \nu_j^* \nabla h_j(x^*) = 0 \quad (3.29)$$

上式中 3.29 等价于

$$\begin{cases} \nabla_x L(x^*, \lambda^*, \nu^*) = 0 \\ \nabla_\lambda L(x^*, \lambda^*, \nu^*) = 0 \\ \nabla_\nu L(x^*, \lambda^*, \nu^*) = 0 \end{cases} \quad (3.30)$$

特别指出, 式 3.27 称为 KKT 的对偶互补条件. 由此条件可知: 若  $\lambda_i^* > 0$ , 则  $g_i(x^*) = 0$

上式中,  $x^*$  是原始变量的解,  $\lambda^*, \nu^*$  是对偶变量的解, 并且  $p^* = d^* = L(x^*, \lambda^*, \nu^*)$

13. 左侧是原函数, 右侧为对偶函数

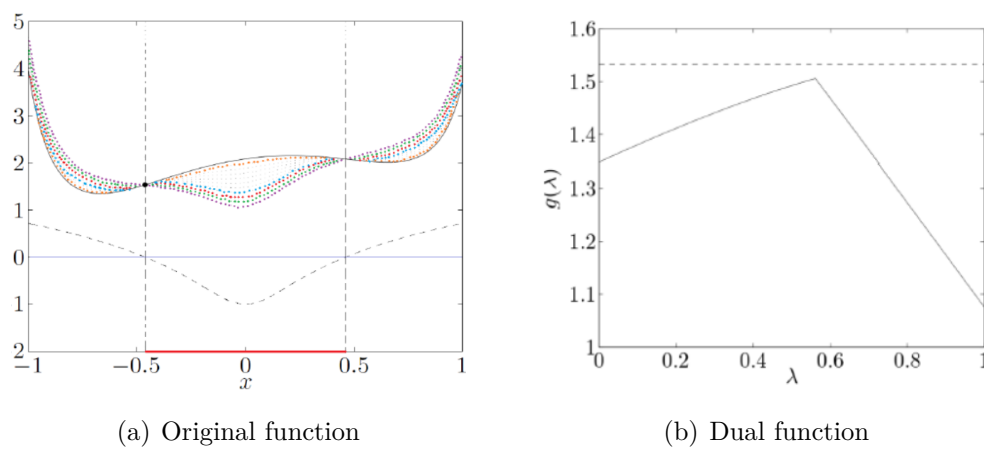


Fig 3.2: dual problem

SVM 里的  $w, b$  看做是 Lagrange 函数里的变量  $x$

## 第四章 SVM

### 4.1 linear separable support vector machine and hard margin maximization

#### 4.1.1 question?

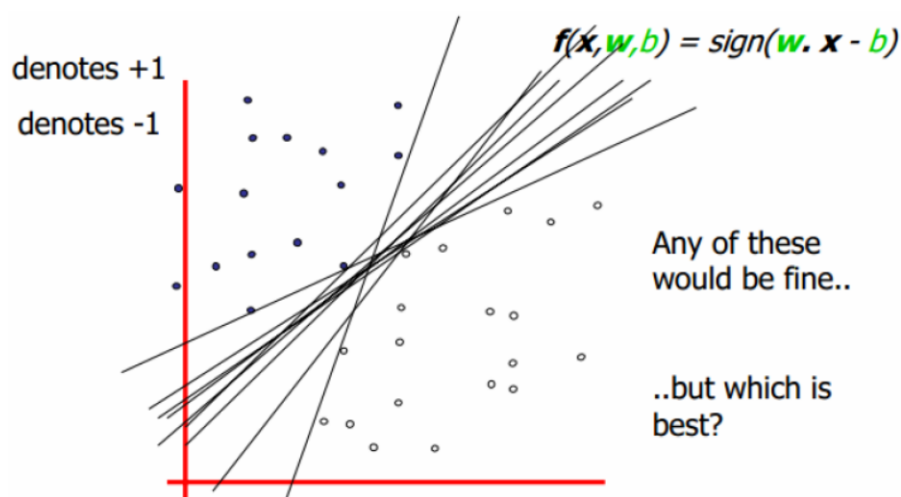


Fig 4.1: linear classifier

#### 4.1.2 input data

1. 假设给定有一个特征空间上的训练数据集  $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$  其中,  $x_i \in \mathbb{R}^n, y_i \in \{+1, -1\}, i = 1, 2, \dots, N$
2.  $x_i$  为第  $i$  个实例 (若  $n > 1$ ,  $x_i$  为向量)
3.  $y_i$  为  $x_i$  的类标记
  - (a) 当  $y = +1$  时, 称  $x_i$  为正例
  - (b) 当  $y = -1$  时, 称  $x_i$  为负例
4.  $(x_i, y_i)$  称为样本点

### 4.1.3 linear separable support vector machine

1. 给定线性可分训练数据集, 通过 间隔最大化 得到的分离超平面为

$$y(x) = w^T \phi(x) + b \quad (4.1)$$

相应的分类决策函数

$$f(x) = \text{sign}(w^T \phi(x) + b) \quad (4.2)$$

该决策函数称为线性可分支持向量机

2.  $\phi(x)$  是某个确定的特征空间转换函数, 它的作用是将  $x$  映射到 (更高的) 维度

(a) 最简单直接的:  $\phi(x) = x$

3. 求解分离超平面问题可以等价求解相应的 凸二次规划问题

### 4.1.4 list symbol

1. 分割平面:  $y(x) = w^T \phi(x) + b$
2. 训练集:  $x_1, x_2, \dots, x_n$
3. 目标值:  $y_1, y_2, \dots, y_n, \quad y_i \in \{-1, +1\}$
4. 新数据的分类:  $\text{sign}(y(x))$

### 4.1.5 derivation target function

1. 根据题设  $y(x) = w^T \phi(x) + b$
2. 有:

$$\begin{cases} y(x_i) > 0 \Leftrightarrow y_i = +1 \\ y(x_i) < 0 \Leftrightarrow y_i = -1 \end{cases} \Rightarrow y_i \cdot y(x_i) > 0 \quad (4.3)$$

3. functional margin and geometric margin

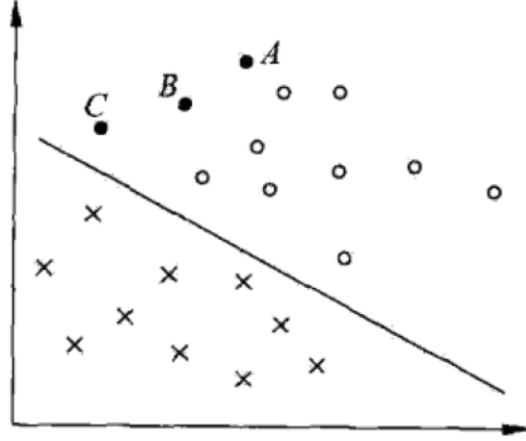


Fig 4.2: functional margin and geometric margin

在图 3 中, 有 A,B,C 三个点, 表示 3 个实例, 均在分离超平面的正类侧, 预测它们的类, 点 A 距分离超平面较远, 若预测该点为正类, 就比较确信预测是正确的, 点 C 距分离超平面较近, 若预测该点为正类就不那么确信, 点 B 介于点 A 与 C 之间, 预测其为正类的确信度也在 A 与 C 之间

- **函数间隔**: 一般来说, 一个点距离分离超平面的远近可以表示分类预测的确信程度. 在超平面  $w^T \phi(x) + b = 0$  确定的情况下,  $|w^T \phi(x) + b|$  能够相对地表示点  $x$  距离超平面的远近, 而  $w^T \phi(x) + b$  的符号与类标记  $y$  的符号是否一致能够表示分类是否正确. 所以可用量  $y_i \cdot (w^T \cdot \phi(x_i) + b)$  来表示分类的正确性及确信度.

定义超平面  $(w, b)$  关于样本点  $(x_i, y_i)$  的函数间隔为

$$\hat{\gamma}_i = y_i(w \cdot x_i + b) \quad (4.4)$$

- **几何间隔**: 函数间隔可以表示分类预测的正确性及确信度. 但是选择分离超平面时, 只有函数间隔还不够, 因为只要成比例的缩放  $w$  和  $b$ , 例如将它们改为  $2w$  和  $2b$ , 超平面并没有改变, 但函数间隔却成为原来的 2 倍, 这一事实启示我们, 可以对分离超平面的法向量  $w$  加某些约束, 如规范化,  $\|w\| = 1$ , 使得间隔是确定的. 这是函数间隔称为几何间隔.

$w, b$  等比例缩放, 则  $y_i \cdot y$  的值同样缩放, 从而:

$$\gamma_i = \frac{y_i \cdot y(x_i)}{\|w\|} = \frac{y_i \cdot (w^T \cdot \phi(x_i) + b)}{\|w\|} \quad (4.5)$$

#### 4. maximum margin to select separable hyperplane



- assume minimum functional margin and geometric margin

定义函数间隔平面  $(w, b)$  关于  $T$  中所有样本点  $(x_i, y_i)$  的函数间隔之最小值, 即

$$\hat{\gamma} = \min_{i=1, \dots, N} \hat{\gamma}_i \quad (4.6)$$

定义几何间隔平面  $(w, b)$  关于  $T$  中所有样本点  $(x_i, y_i)$  的几何间隔之最小值, 即

$$\gamma = \min_{i=1, \dots, N} \gamma_i \quad (4.7)$$

- original target function

$$\arg \max_{w, b} \gamma = \arg \max_{w, b} \frac{\hat{\gamma}}{\|w\|} \Rightarrow \quad (4.8)$$

$$\arg \max_{w, b} \left\{ \frac{1}{\|w\|} \min_i [y_i \cdot (w^T \cdot \phi(x_i) + b)] \right\} \quad (4.9)$$

$$\text{s.t. } \frac{y_i \cdot (w^T \cdot \phi(x_i) + b)}{\|w\|} \geq \gamma \quad (4.10)$$

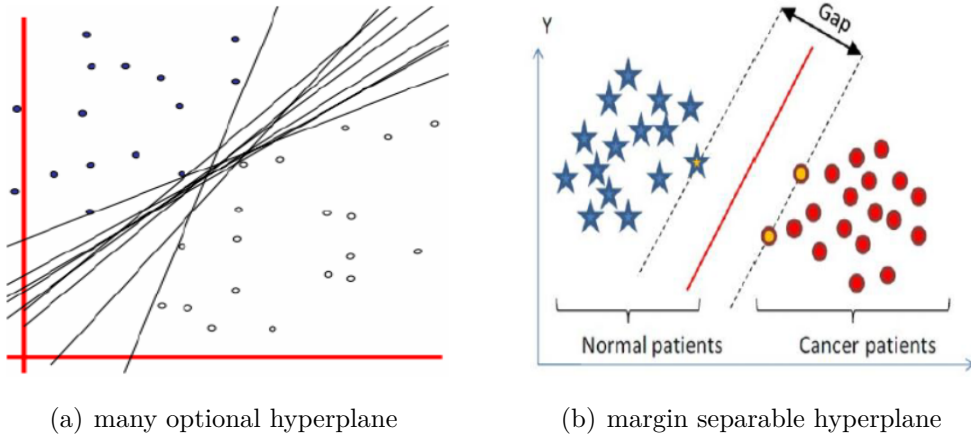


Fig 4.3: maximum margin

- optimal problem

函数间隔  $\hat{\gamma} = y(w \cdot x + b)$  不影响最优化问题的解. 事实上, 假设将  $w$  和  $b$  按比例改变为  $\lambda w$   $\lambda b$ , 这时函数间隔成为  $\lambda \hat{\gamma}$ .

函数间隔的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优化问题.

取函数间隔最小值  $\hat{\gamma} = 1$ . 将  $\hat{\gamma} = 1$  代入上面的最优化问题, 注意到最大化  $\frac{1}{\|w\|}$  和最小化  $\frac{1}{2} \|w\|^2$  是等价的. 则得到下面的 **线性可分支持向量机** 的最优化问题:

$$\min_{w,b} \quad \frac{1}{2} \|w\|^2 \quad (4.11)$$

$$\text{s.t.} \quad y_i(w \cdot x_i + b) - 1 \geq 0, \quad i = 1, 2, \dots, N \quad (4.12)$$

这是一个凸二次规划 (convex quadratic programming) 问题: 目标函数  $f(x)$  是二次函数且约束函数  $g_i(w)$  是仿射函数时, 约束最优化问题 (凸优化问题) 成为凸二次规划问题

## 5. dual algorithm

(a) Lagrange multiplier method to build Lagrange function

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i (y_i (w^T \cdot \phi(x_i) + b) - 1) \quad (4.13)$$

among,  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_N)^T$  is Lagrange multiplier vector.

(b) 原问题是极小极大问题

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

原始问题的对偶问题, 是极大极小问题

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

(c) 得到对偶问题的解

(1) 求  $\min_{w,b} L(w, b, \alpha)$

将拉格朗日函数  $L(w, b, \alpha)$  分别对  $w, b$  求偏导数并令其等于 0.

$$\nabla_w L(w, b, \alpha) = w - \sum_{i=1}^N \alpha_i y_i x_i = 0 \quad (4.14)$$

$$\nabla_b L(w, b, \alpha) = \sum_{i=1}^N \alpha_i y_i = 0 \quad (4.15)$$

$$\text{得} \quad (4.16)$$

$$w = \sum_{i=1}^N \alpha_i x_i y_i \quad (4.17)$$

$$\sum_{i=1}^N \alpha_i y_i = 0 \quad (4.18)$$

(2) 将式 4.17 代入 Lagrange 函数 4.13, 并利用式 4.18, 即得

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \alpha_i (y_i (w^T \cdot \phi(x_i) + b) - 1) \quad (4.19)$$

$$= \frac{1}{2} w^T w - w^T \sum_{i=1}^n \alpha_i y_i \phi(x_i) - b \sum_{i=1}^n \alpha_i y_i + \sum_{i=1}^n \alpha_i \quad (4.20)$$

$$= \frac{1}{2} w^T \sum_{i=1}^n \alpha_i y_i \phi(x_i) - w^T \sum_{i=1}^n \alpha_i y_i \phi(x_i) - b \cdot 0 + \sum_{i=1}^n \alpha_i \quad (4.21)$$

$$= \sum_{i=1}^n \alpha_i - \frac{1}{2} \left( \sum_{i=1}^n \alpha_i y_i \phi(x_i) \right)^T \sum_{i=1}^n \alpha_i y_i \phi(x_i) \quad (4.22)$$

$$= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \phi^T(x_i) \phi(x_j) \quad (4.23)$$

即

$$\min_{w,b} L(w, b, \alpha) = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \phi^T(x_i) \phi(x_j) + \sum_{i=1}^N \alpha_i \quad (4.24)$$

(3) 求  $\min_{w,b} L(w, b, \alpha)$  对  $\alpha$  的极大, 即是对偶问题

$$\max_{\alpha} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j (\phi(x_i) \phi(x_j)) \quad (4.25)$$

$$\text{s.t.} \quad \sum_{i=1}^n \alpha_i y_i = 0 \quad (4.26)$$

$$\alpha_i \geq 0, \quad i = 1, 2, \dots, n \quad (4.27)$$

(4) 整理目标函数: 添加负号

$$\min_{\alpha} \quad \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j (\phi(x_i) \phi(x_j)) - \sum_{i=1}^n \alpha_i \quad (4.28)$$

$$\text{s.t.} \quad \sum_{i=1}^n \alpha_i y_i = 0 \quad (4.29)$$

$$\alpha_i \geq 0, \quad i = 1, 2, \dots, n \quad (4.30)$$

#### 4.1.6 Linear divisible support Vector Machine Learning algorithm

1. 构造并求解约束最优化问题

$$\min_{\alpha} \quad \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j (\phi(x_i) \phi(x_j)) - \sum_{i=1}^n \alpha_i \quad (4.31)$$

$$\text{s.t.} \quad \sum_{i=1}^n \alpha_i y_i = 0 \quad (4.32)$$

$$\alpha_i \geq 0, \quad i = 1, 2, \dots, n \quad (4.33)$$

2. 求得最优解  $\alpha^*$ ,  $\alpha^* = (\alpha_1^*, \alpha_1^*, \dots, \alpha_1^*)^T$

3. 计算

$$w^* = \sum_{i=1}^N \alpha_i^* y_i \phi(x_i) \quad (4.34)$$

$$b^* = y_i - \sum_{i=1}^N \alpha_i^* y_i (\phi(x_i) \cdot \phi(x_j)) \quad (4.35)$$

4. 求得分离超平面

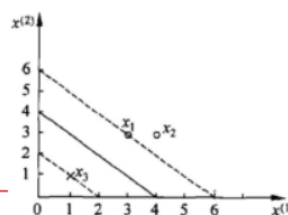
$$w^* \phi(x) + b^* = 0 \quad (4.36)$$

5. 分类决策函数

$$f(x) = \text{sign}(w^* \phi(x) + b^*) \quad (4.37)$$

6. example

## 举例



□ 给定3个数据点：正例点  $x_1 = (3, 3)^T$ ,  $x_2 = (4, 3)^T$ , 负例点  $x_3 = (1, 1)^T$ , 求线性可支持向量机。

□ 目标函数：

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^n \alpha_i \\ = & \frac{1}{2} (18\alpha_1^2 + 25\alpha_2^2 + 2\alpha_3^2 + 42\alpha_1\alpha_2 - 12\alpha_1\alpha_3 - 14\alpha_2\alpha_3) - \alpha_1 - \alpha_2 - \alpha_3 \\ \text{s.t.} \quad & \alpha_1 + \alpha_2 - \alpha_3 = 0 \\ & \alpha_i \geq 0, \quad i = 1, 2, 3 \end{aligned}$$

## 将约束带入目标函数，化简计算

□ 将  $\alpha_1 + \alpha_2 = \alpha_3$

□ 带入目标函数，得到关于  $\alpha_1, \alpha_2$  的函数：

$$s(\alpha_1, \alpha_2) = 4\alpha_1^2 + \frac{13}{2}\alpha_2^2 + 10\alpha_1\alpha_2 - 2\alpha_1 - 2\alpha_2$$

□ 对  $\alpha_1, \alpha_2$  求偏导并令其为0，易知  $s(\alpha_1, \alpha_2)$  在点  $(1.5, -1)$  处取极值。而该点不满足条件  $\alpha_2 \geq 0$ ，所以，最小值在边界上达到。

□ 当  $\alpha_1 = 0$  时，最小值  $s(0, 2/13) = -2/13 = -0.1538$

□ 当  $\alpha_2 = 0$  时，最小值  $s(1/4, 0) = -1/4 = -0.25$

□ 于是， $s(\alpha_1, \alpha_2)$  在  $\alpha_1 = 1/4, \alpha_2 = 0$  时达到最小，此时， $\alpha_3 = \alpha_1 + \alpha_2 = 1/4$

## 分离超平面

□  $\alpha_1 = \alpha_3 = 1/4$  对应的点  $x_1, x_3$  是支持向量。

□ 带入公式:  $w^* = \sum_{i=1}^N \alpha_i^* y_i \Phi(x_i)$

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

□ 得到  $w_1 = w_2 = 0.5$ ,  $b = -2$

□ 因此, 分离超平面为  $\frac{1}{2}x_1 + \frac{1}{2}x_2 - 2 = 0$

□ 分离决策函数为  $f(x) = \text{sign}\left(\frac{1}{2}x_1 + \frac{1}{2}x_2 - 2\right)$

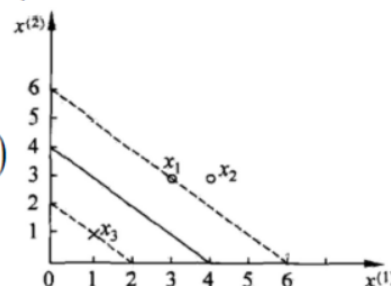


Fig 4.4: the example of solve largest margin separable hyperplane with svm method

## 4.2 linear support vector machine and soft margin maximization

### 4.2.1 concept

1. 不一定分类完全正确的超平面就是最好的
2. 样本数据本身线性不可分
3. picture

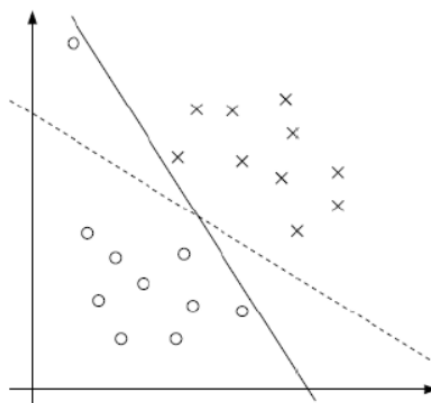


Fig 4.5: non separable dataset

#### 4.2.2 derivation target function

1. 若数据线性不可分, 则增加松弛因子  $\xi \geq 0$ , 使函数间隔加上松弛变量大于等于 1. 这样, 约束条件变成

$$y_i(w \cdot x_i + b) \geq 1 - \xi \quad (4.38)$$

2. target function with adding slack variable  $\xi$ :

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \quad (4.39)$$

3. this moment, the target function of linear svm

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \quad (4.40)$$

$$\text{s.t. } y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \dots, n \quad (4.41)$$

$$\xi_i \geq 0, \quad i = 1, 2, \dots, n \quad (4.42)$$

#### 4.2.3 dual property

1. Lagrange 函数

$$L(w, b, \xi, \alpha, \mu) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i (y_i(w \cdot x_i + b) - 1 + \xi_i) - \sum_{i=1}^n \mu_i \xi_i \quad (4.43)$$

2. 对  $w, b, \xi$  求偏导

$$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^n \alpha_i y_i \phi(x_i) \quad (4.44)$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow 0 = \sum_{i=1}^n \alpha_i y_i \quad (4.45)$$

$$\frac{\partial L}{\partial \xi} = 0 \Rightarrow C - \alpha_i - \mu_i = 0 \quad (4.46)$$

3. 将式 4.44、式 4.45、式 4.46 代入式 4.43 中, 得到

$$\min_{w,b,\xi} L(w, b, \xi, \alpha, \mu) = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) + \sum_{i=1}^n \alpha_i \quad (4.47)$$

4. 对上式 4.47 求关于  $\alpha$  的极大, 得到

$$\max_{\alpha} \quad -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) + \sum_{i=1}^n \alpha_i \quad (4.48)$$

$$\text{s.t.} \quad \sum_{i=1}^n \alpha_i y_i = 0 \quad (4.49)$$

$$C - \alpha_i - \mu_i = 0 \quad (4.50)$$

$$\alpha_i \geq 0 \quad (4.51)$$

$$\mu_i \geq 0, \quad i = 1, 2, \dots, n \quad (4.52)$$

5. 将对偶问题 4.48 ~ 4.52 进行变换: 利用等式约束 4.50 消去  $\mu_i$ , 从而只留下变量  $\alpha_i$ , 并将约束 4.50 ~ 4.52 写成

$$0 \leq \alpha_i \leq C \quad (4.53)$$

6. 整理, 得到对偶问题:

$$\min_{\alpha} \quad \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^n \alpha_i \quad (4.54)$$

$$\text{s.t.} \quad \sum_{i=1}^n \alpha_i y_i = 0 \quad (4.55)$$

$$0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, n \quad (4.56)$$

#### 4.2.4 Linear support vector machine learning algorithm

1. 构造并求解约束最优化问题

$$\min_{\alpha} \quad \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^n \alpha_i \quad (4.57)$$

$$\text{s.t.} \quad \sum_{i=1}^n \alpha_i y_i = 0 \quad (4.58)$$

$$0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, n \quad (4.59)$$

2. 求得最优解  $\alpha^*$ ,  $\alpha^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_n^*)^T$

3. 计算

$$w^* = \sum_{i=1}^n \alpha_i^* y_i x_i \quad (4.60)$$

计算  $b$  的值有两种方式:



- 选择  $\alpha^*$  的一个分量  $\alpha_j^*$  适合条件  $0 < \alpha_j^* < C$ , 计算

$$b^* = y_j - \sum_{i=1}^n y_i \alpha_i^* (x_i \cdot x_j) \quad (4.61)$$

- 实践中往往取支持向量的所有值取平均, 作为  $b^*$

$$b^* = \frac{\max_{i:y_i=-1} w^* \cdot x_i + \min_{i:y_i=1} w^* \cdot x_i}{2} \quad (4.62)$$

#### 4. 求得分离超平面

$$w^* x + b^* = 0 \quad (4.63)$$

#### 5. 分类决策函数

$$f(x) = \text{sign}(w^* x + b^*) \quad (4.64)$$

### 4.2.5 loss function

#### 1.