



北京航空航天大学  
COLLEGE OF SOFTWARE 软件学院  
BEIHANG UNIVERSITY

# 人工智能

## 第4讲：机器学习-有监督学习

### 线性判别分析与支持向量机

张晶

2025春季

- 参考资料： 吴飞, 《人工智能导论：模型与算法》，高等教育出版社
- 在线课程： <https://www.icourse163.org/course/ZJU-1003377027?from=searchPage>

请思考，线性模型 $f(x_i) = wx_i + b$ 中，如果省去bias项，即 $b = 0$ ，线性模型变为 $f(x_i) = wx_i$ ，会出现什么问题？

[作答](#)

多元线性回归中，学习模型为

$$f(\mathbf{x}_i) = \sum_{j=1}^D w_j x_{i,j} + w_0 = \mathbf{w}^T \mathbf{x}_i + w_0$$

请回顾，为什么为每一个数据  $\mathbf{x}_i$  扩展一个维度，其值为1，对应参数  $w_0$ 。

$$X = \begin{bmatrix} \mathbf{x}_1, \dots, \mathbf{x}_n \\ 1, \dots, 1 \end{bmatrix}^T$$

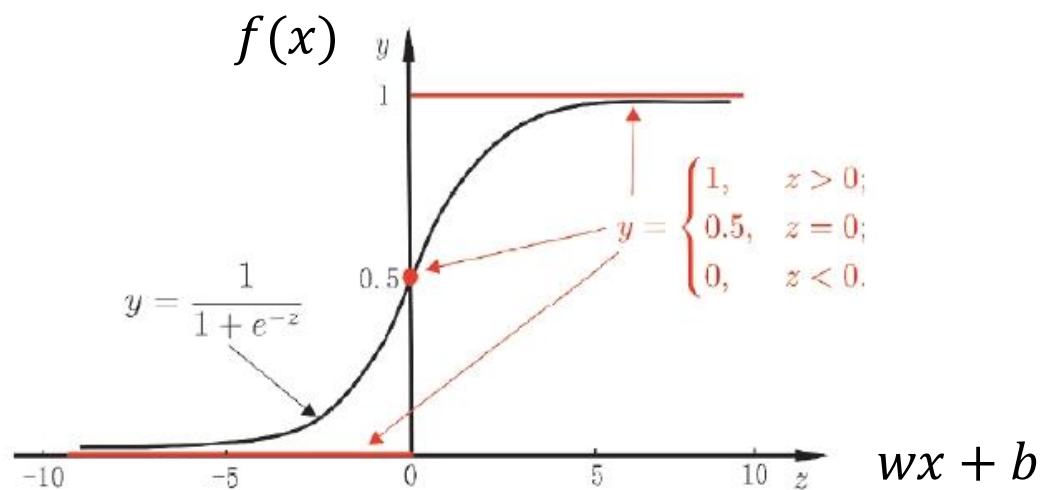
作答



## 回顾-线性分类模型：对数几率回归 (Logistic Regression)

- **训练数据**:  $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ , 其中  $y_i \in \{0, 1\}$

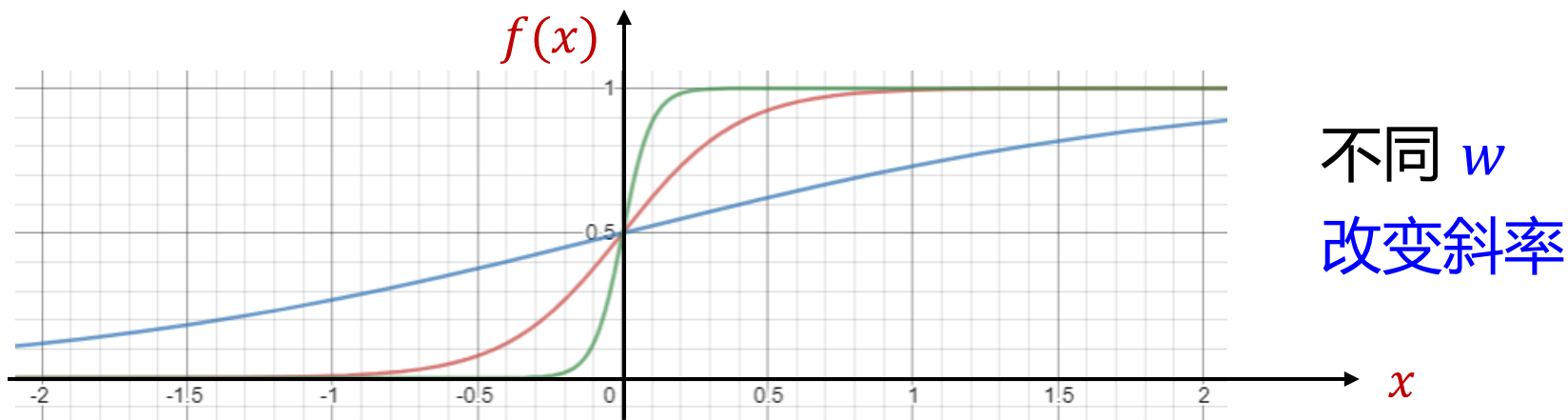
- **学习模型**:  $f(x_i) = \frac{1}{1+e^{-(w^T x_i + b)}} \quad (1 \leq i \leq n)$





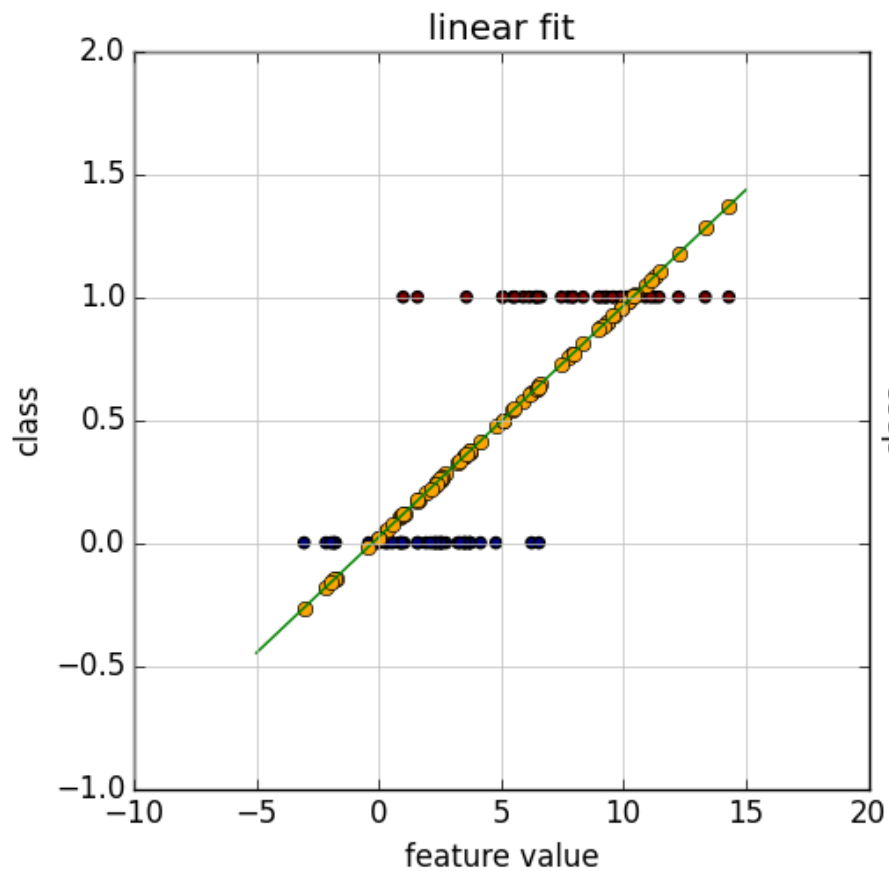
## 回顾-线性分类模型：对数几率回归 (Logistic Regression)

- 学习模型：
$$f(x_i) = \frac{1}{1+e^{-(w^T x_i + b)}} \quad (1 \leq i \leq n)$$

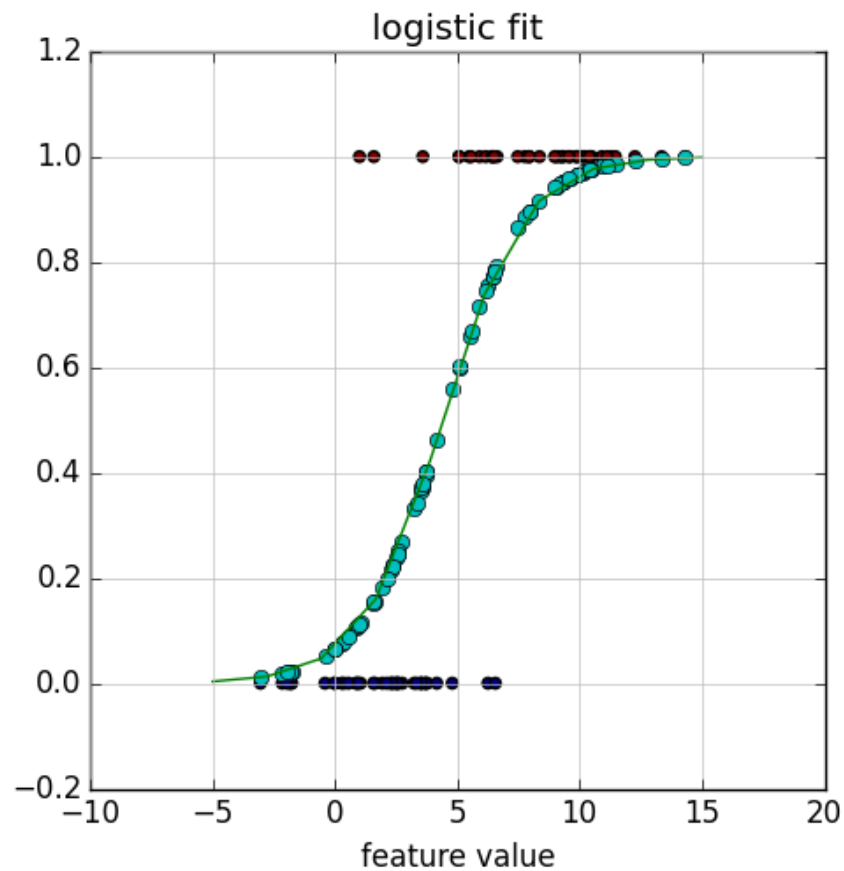




# 线性回归 v.s. 对数几率回归



线性回归：回归任务



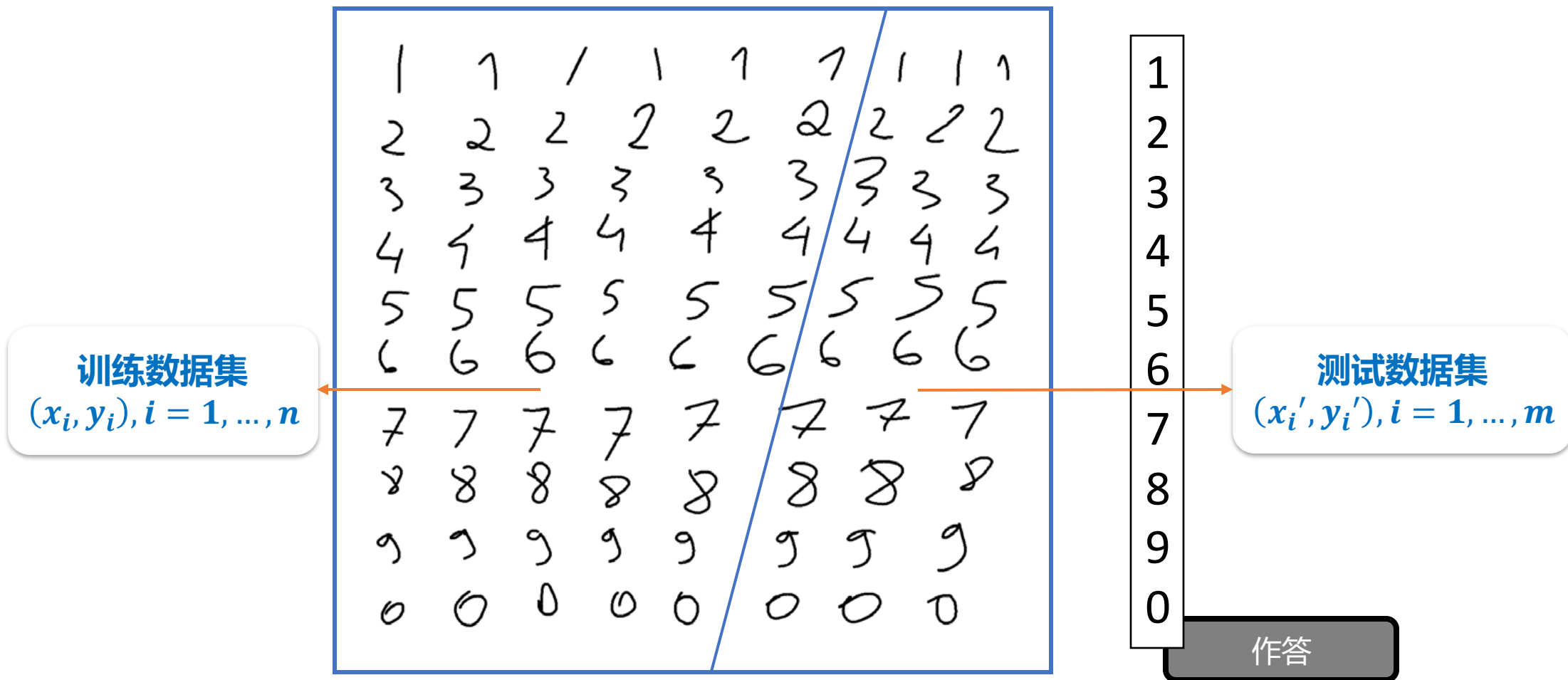
对数几率回归：分类任务



# 线性分类：对数几率回归 (Logistic Regression)

- 训练数据：  $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ ，其中  $y_i \in \{0, 1\}$
- 学习模型：  $f(x_i) = \frac{1}{1 + e^{-(w^T x_i + b)}} \quad (1 \leq i \leq n)$
- 损失函数：  $L(\theta) = -(\sum_{i=1}^n y_i \log(f_\theta(x_i)) + (1 - y_i) \log(1 - f_\theta(x_i)))$
- 优化方法：梯度下降
  - 对  $L(\theta)$  参数  $\theta$  分别求偏导，  $\frac{\partial L(\theta)}{\partial \theta_j}$ 。
  - 更新  $\theta_j = \theta_j - \eta \frac{\partial L(\theta)}{\partial \theta_j}$ ，其中  $\eta$  为学习率

请思考，为什么需要划分训练集和测试集？







北京航空航天大学  
COLLEGE OF SOFTWARE  
BEIHANG UNIVERSITY 软件学院

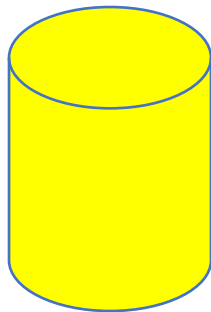
# 监督学习：经验风险与期望风险

从训练数据集  
学习映射函数  $f$

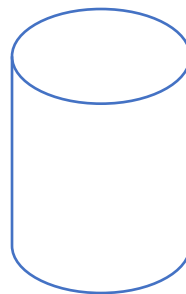
在测试数据集  
测试映射函数  $f$

经验风险(empirical risk)

- 训练集中数据产生的损失。  
经验风险越小说明学习模型  
对训练数据拟合程度越好。



训练数据集  
 $(x_i, y_i), i = 1, \dots, n$



测试数据集  
 $(x'_i, y'_i), i = 1, \dots, m$

期望风险(expected risk):

- 当测试集中存在无穷多数据  
时产生的损失。期望风险越  
小，学习所得模型越好。



# 监督学习：经验风险与期望风险

- 映射函数训练目标：**经验风险最小化**  
(empirical risk minimization, ERM)

$$\min_{f \in \Phi} \frac{1}{n} \sum_{i=1}^n \text{Loss}(y_i, f(x_i))$$



训练数据集  
 $(x_i, y_i), i = 1, \dots, n$

选取一个使得训练集所有数据损失平均值最小的映射函数。这样的考虑是否够？

- 映射函数训练目标：**期望风险最小化**  
(expected risk minimization)

$$\min_{f \in \Phi} \int_{x \times y} \text{Loss}(y, f(x)) P(x, y) dx dy$$



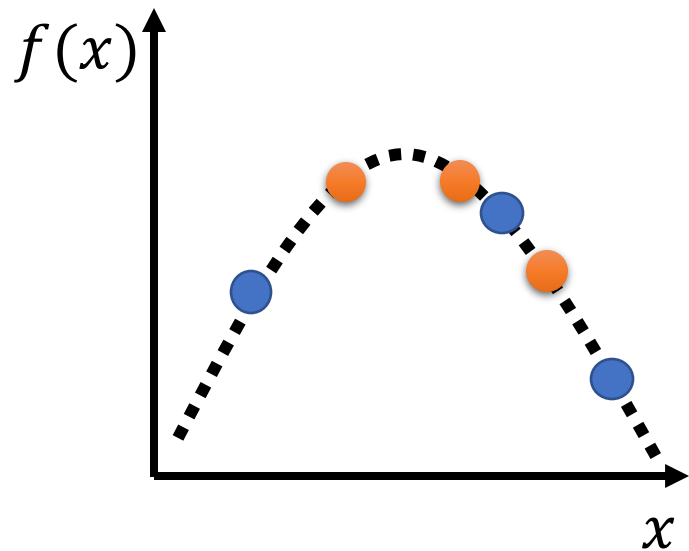
测试数据集数据无穷多  
 $(x_i', y_i'), i = 1, \dots, \infty$

- 期望风险是模型关于联合分布**期望损失**，经验风险是模型关于训练样本集**平均损失**。
- 根据大数定律，当样本容量趋于无穷时，经验风险趋于期望风险。所以在实践中很自然用经验风险来估计期望风险。
- 由于现实中训练样本数目有限，用经验风险估计期望风险并不理想，要对经验风险进行一定的约束。



北京航空航天大学  
COLLEGE OF SOFTWARE  
BEIHANG UNIVERSITY 软件学院

# 监督学习：“过学习(over-fitting)”与“欠学习(under-fitting)”

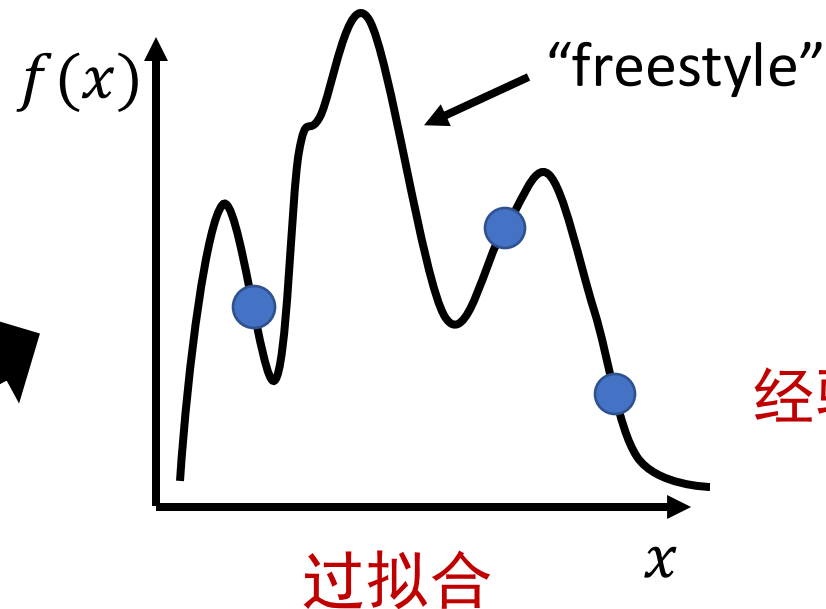
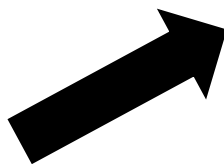


..... 真实数据分布  
(不可观测)

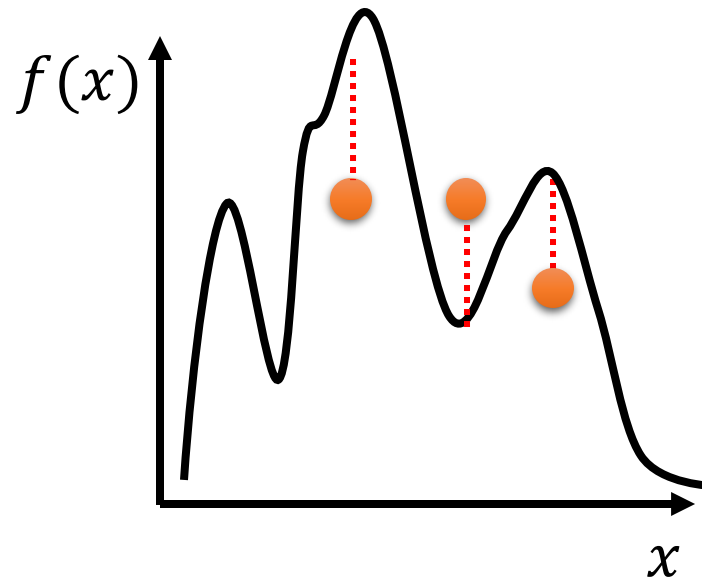
● 训练数据

● 测试数据

复杂模型

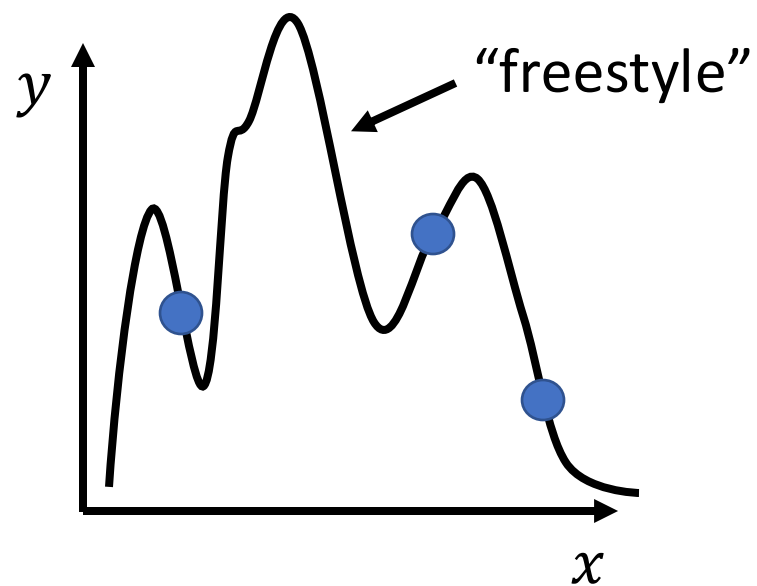


经验风险小



期望风险大

你能想到的缓解过拟合问题的方法或思路？

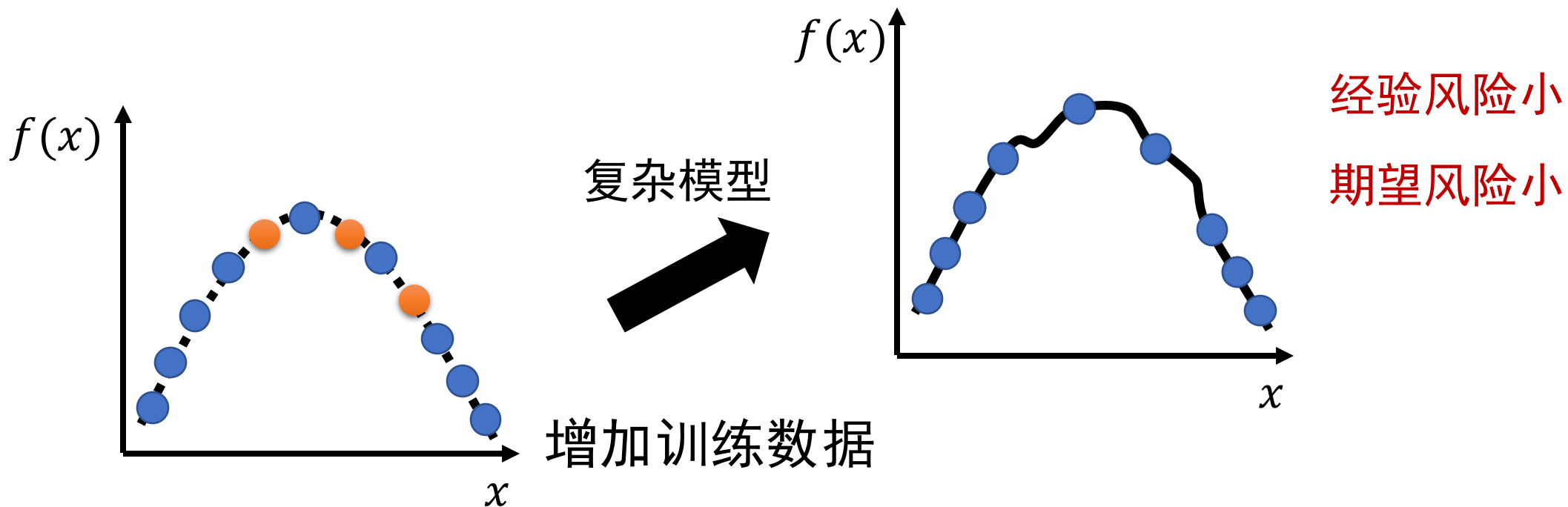


作答



北京航空航天大学  
COLLEGE OF SOFTWARE  
BEIHANG UNIVERSITY 软件学院

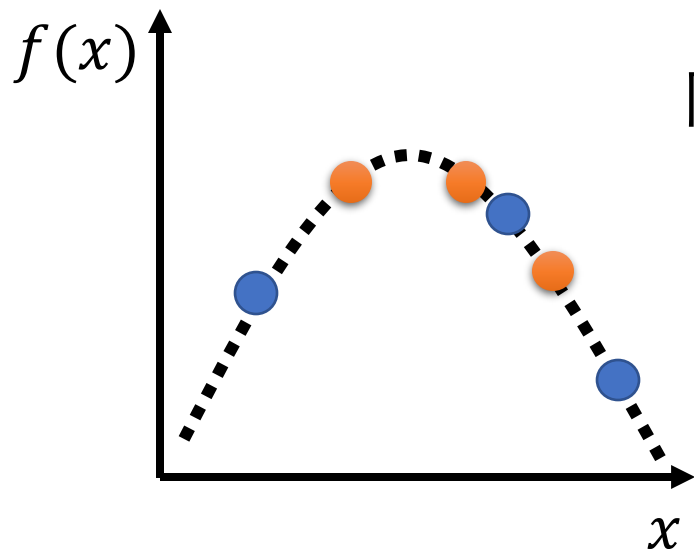
# 监督学习：“过学习(over-fitting)”与“欠学习(under-fitting)”



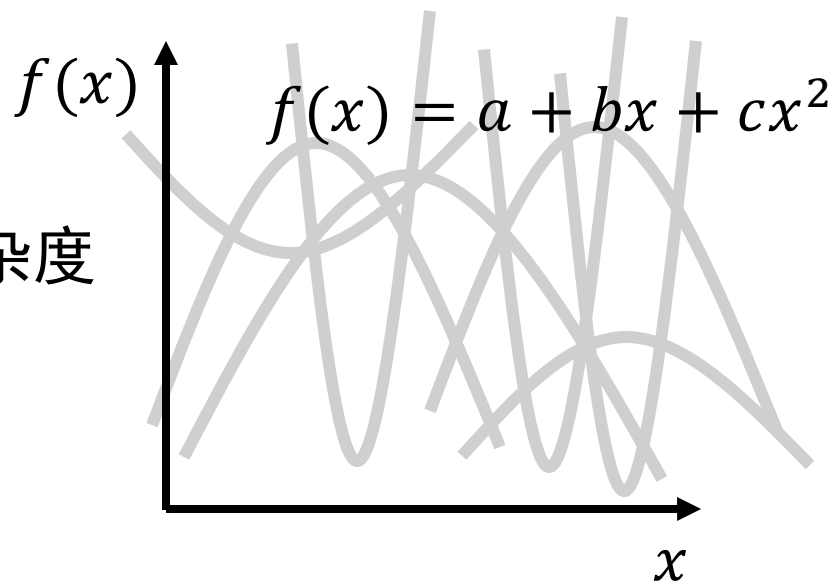
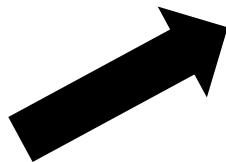


北京航空航天大学  
COLLEGE OF SOFTWARE  
BEIHANG UNIVERSITY 软件学院

# 监督学习: “过学习(over-fitting)” 与 “欠学习(under-fitting)”



限制模型复杂度



..... 真实数据分布  
(不可观测)

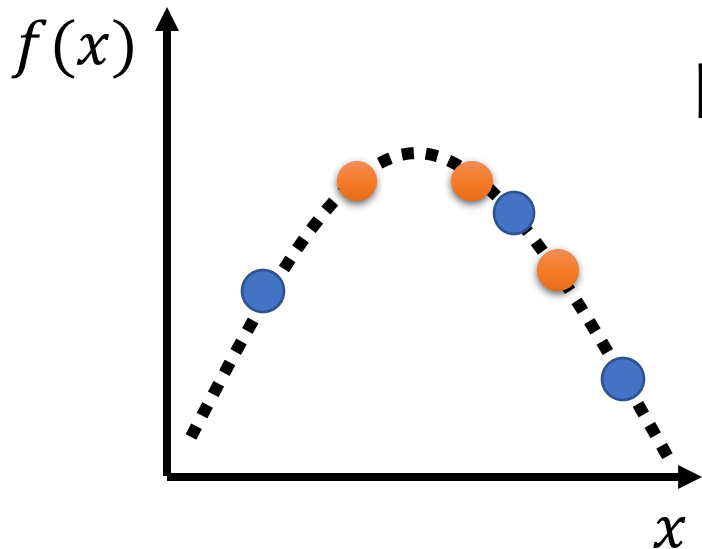
● 训练数据

● 测试数据

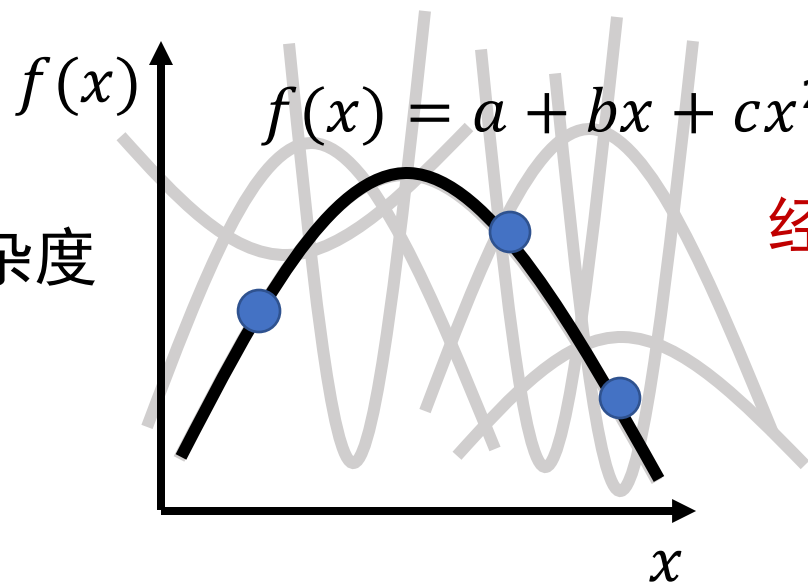
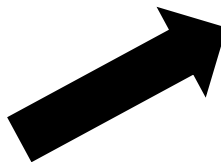


北京航空航天大学  
COLLEGE OF SOFTWARE  
BEIHANG UNIVERSITY 软件学院

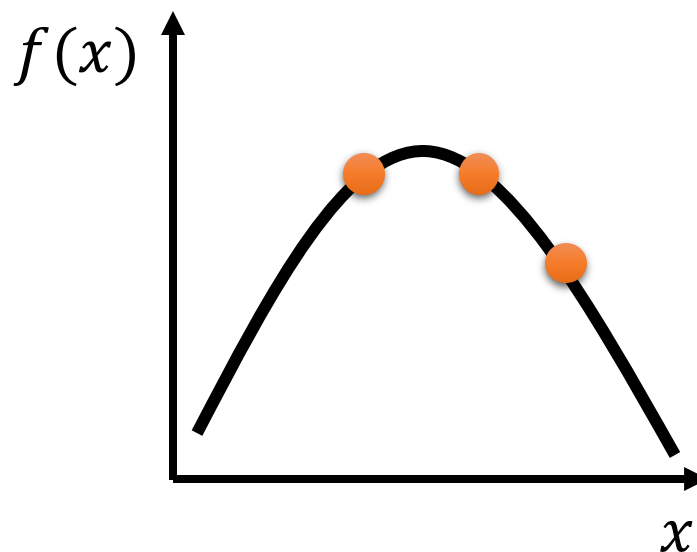
# 监督学习：“过学习(over-fitting)”与“欠学习(under-fitting)”



限制模型复杂度



经验风险小



期望风险小

..... 真实数据分布  
(不可观测)

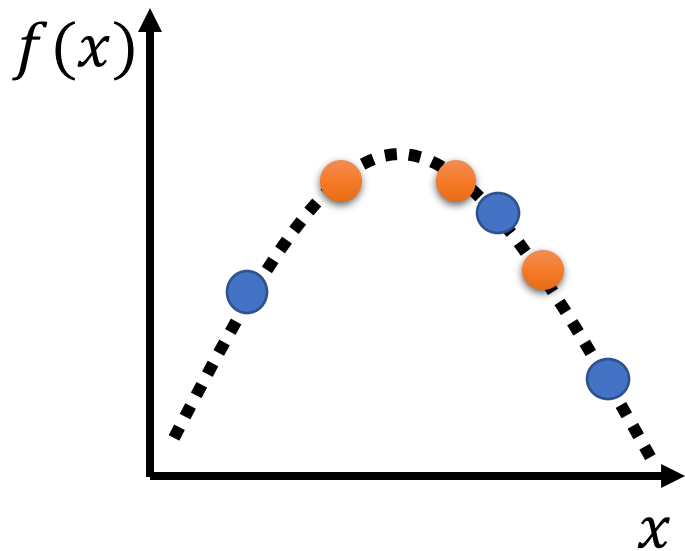
● 训练数据

● 测试数据

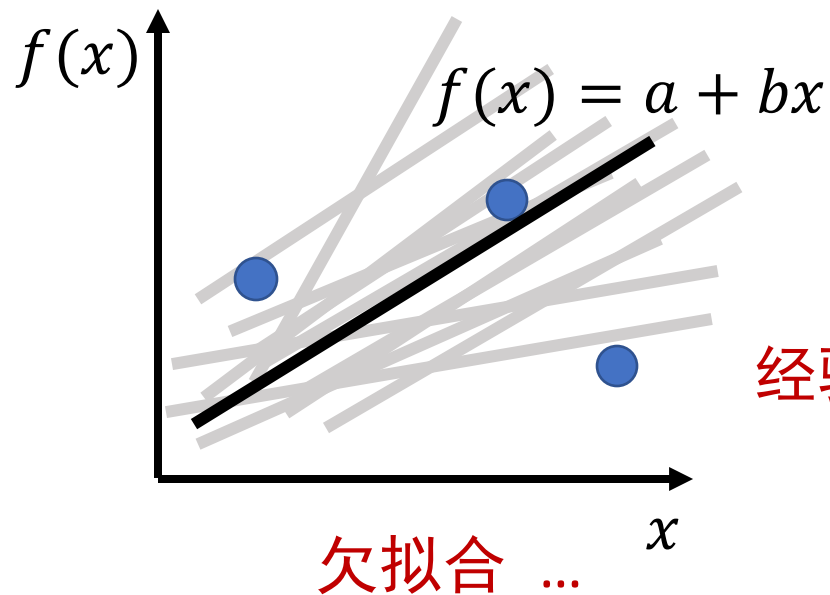
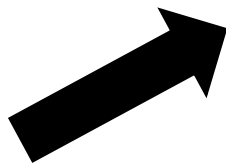


北京航空航天大学  
COLLEGE OF SOFTWARE  
BEIHANG UNIVERSITY 软件学院

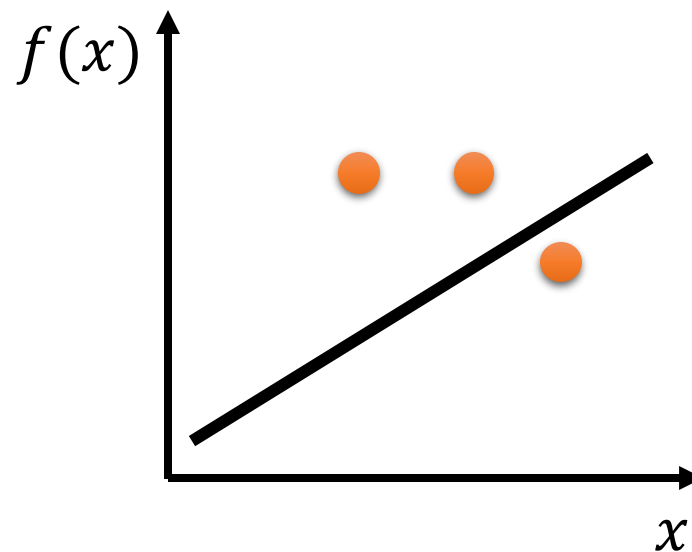
# 监督学习：“过学习(over-fitting)”与“欠学习(under-fitting)”



过度约束



经验风险大



期望风险大

..... 真实数据分布  
(不可观测)

● 训练数据

● 测试数据



# 监督学习: “过学习(over-fitting)” 与 “欠学习(under-fitting)”

- 经验风险最小化

$$\min_{f \in \Phi} \frac{1}{n} \sum_{i=1}^n Loss(y_i, f(x_i))$$

- 期望风险最小化

$$\min_{f \in \Phi} \int_{x \times y} Loss(y, f(x)) P(x, y) dx dy$$

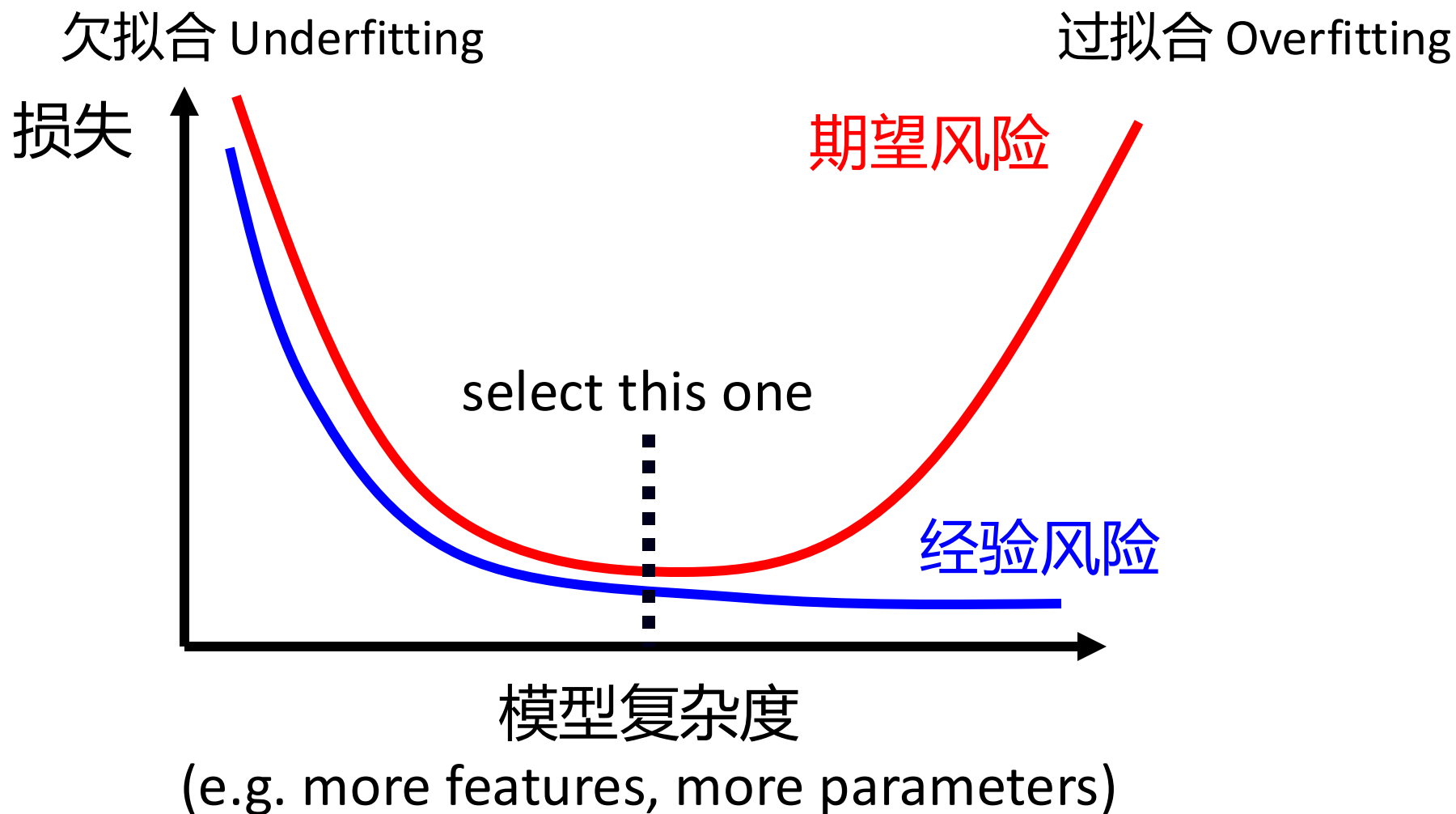
经验风险小 (训练集上表现好)	期望风险小 (测试集上表现好)	泛化能力强
经验风险小 (训练集上表现好)	期望风险大 (测试集上表现不好)	过学习 (模型过于复杂)
经验风险大 (训练集上表现不好)	期望风险大 (测试集上表现不好)	欠学习
经验风险大 (训练集上表现不好)	期望风险小 (测试集上表现好)	“神仙算法” 或 “黄粱美梦”

表4.3 模型泛化能力与经验风险、期望风险的关系



北京航空航天大学  
COLLEGE OF SOFTWARE  
BEIHANG UNIVERSITY 软件学院

# 监督学习: 偏见-复杂权衡 Bias-Complexity Trade-off





# 监督学习: 结构风险最小

- 经验风险最小化: 仅反映了局部数据

$$\min_{f \in \Phi} \frac{1}{n} \sum_{i=1}^n \text{Loss}(y_i, f(x_i))$$

- 期望风险最小化: 无法得到全量数据

$$\min_{f \in \Phi} \int_{x \times y} \text{Loss}(y, f(x)) P(x, y) dx dy$$

- 结构风险最小化(structural risk minimization):

- 为了防止过拟合, 在经验风险上加上表示模型复杂度的正则化项(regulatizer)或惩罚项(penalty term) :

$$\min_{f \in \Phi} \frac{1}{n} \sum_{i=1}^n \text{Loss}(y_i, f(x_i)) + \lambda R(f)$$

经验风险                      模型复杂度

在最小化经验风险与降低模型复杂度之间寻找平衡



# 监督学习: 结构风险最小

- 常用的正则化项:

- **L1正则化**: 通过在损失函数中加入权重的L1范数（即权重的绝对值之和）来惩罚模型的复杂度, 即 $R(f) = \|w\|_1$ 。其损失函数形式为:

$$L(w)_{L1} = \frac{1}{n} \sum_{i=1}^n \text{Loss}(y_i, f(x_i)) + \lambda \|w\|_1$$

- **L2正则化**: 通过在损失函数中加入权重的L2范数（即权重的平方和）来惩罚模型的复杂度, 即 $R(f) = \|w\|_2^2$ 。其损失函数形式为:





$$L(w)_{L2} = \frac{1}{n} \sum_{i=1}^n \text{Loss}(y_i, f(x_i)) + \lambda \|w\|_2^2$$

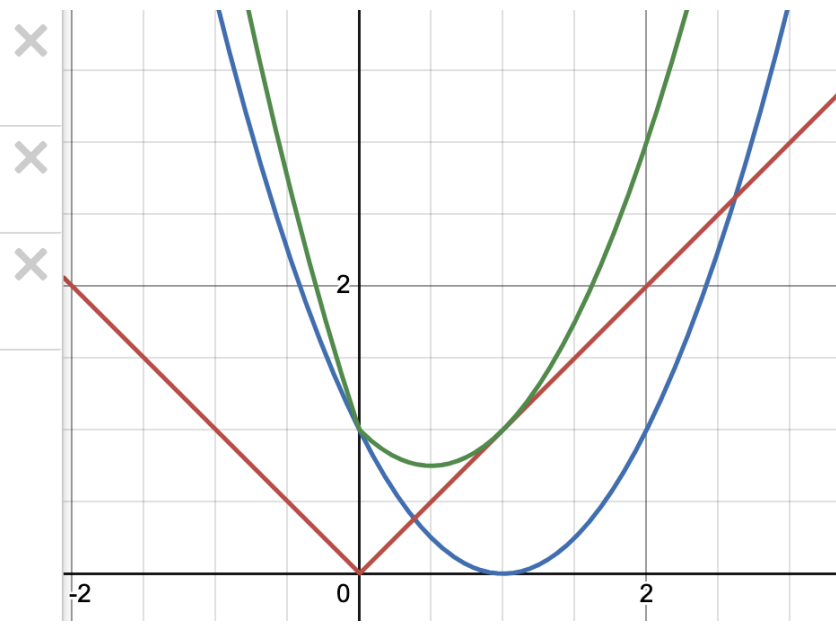


# 监督学习：结构风险最小

- **L1正则化**：通过在损失函数中加入权重的L1范数（即权重的绝对值之和）来惩罚模型的复杂度，即 $R(f) = \|w\|_1$ 。其损失函数形式为：

$$L(w)_{L1} = \frac{1}{n} \sum_{i=1}^n \text{Loss}(y_i, f(x_i)) + \lambda \|w\|_1$$

$\text{Loss}(y_i, f(x_i)) =$	1 	$= (w - 1)^2$
$R(f) =$	2 	$=  w $
$L(w)_{L1} =$	3 	$= (w - 1)^2 +  w $
	4 	






# 监督学习: 结构风险最小

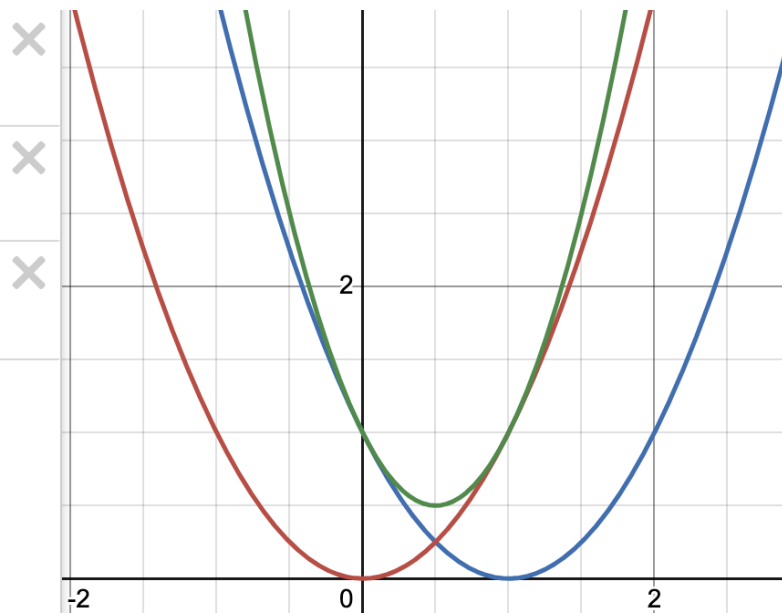
- **L2正则化**: 通过在损失函数中加入权重的L2范数（即权重的平方和）来惩罚模型的复杂度，即 $R(f) = \|w\|_2^2$ 。其损失函数形式为：

$$L(w)_{L2} = \frac{1}{n} \sum_{i=1}^n \text{Loss}(y_i, f(x_i)) + \lambda \|w\|_2^2$$

$\text{Loss}(y_i, f(x_i)) =$    $= (w - 1)^2$

$R(f) =$    $= w^2$

$L(w)_{L2} =$    $= (w - 1)^2 + w^2$





北京航空航天大学  
COLLEGE OF SOFTWARE  
BEIHANG UNIVERSITY 软件学院

# 提纲

**一、机器学习基本概念**

**二、线性回归与线性分类**

**三、线性判别分析**

**四、支持向量机**

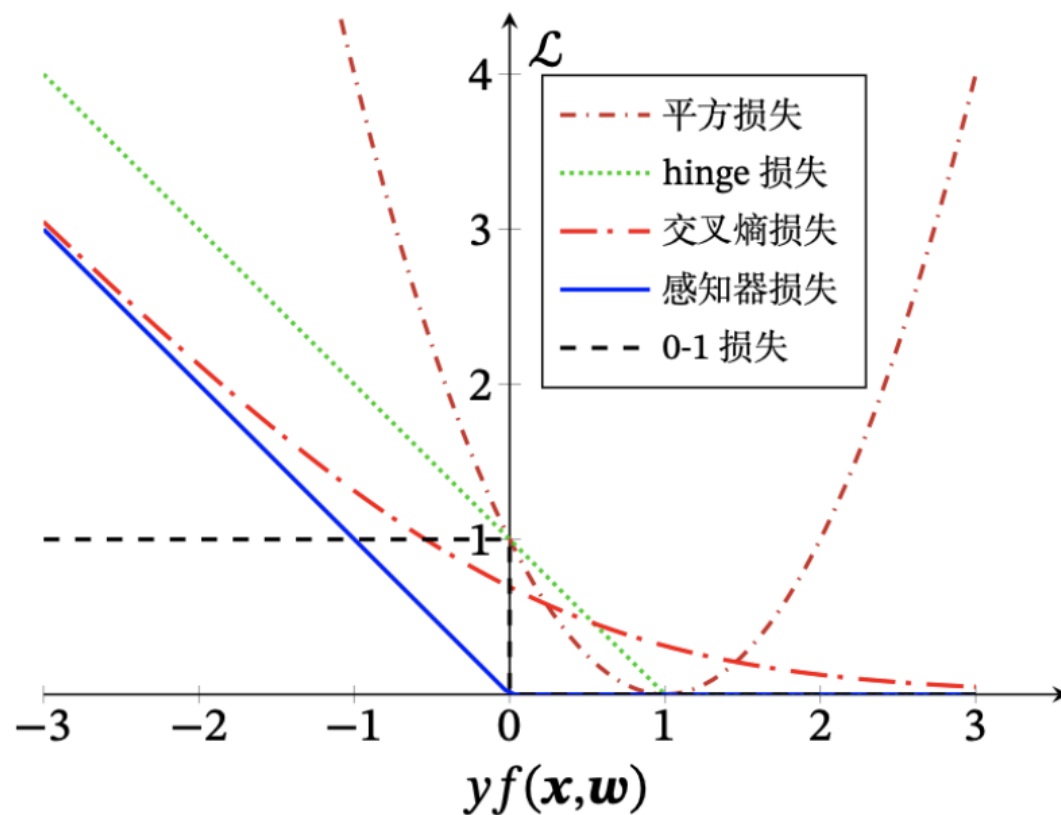
**五、决策树**

**六、集成学习**



# 线性分类模型

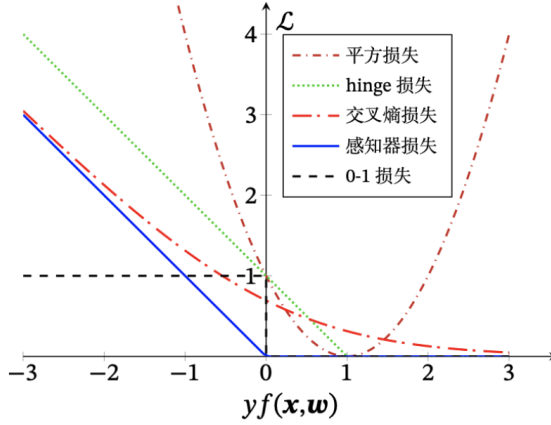
- 线性回归:  $\frac{1}{N} \sum (y_i - f(x_i))^2$
- 对数几率回归:  $-\log P(y_i | x_i)$
- 线性判别分析
- 感知器
- 支持向量机
- ...



二分类问题中不同损失函数的对比  
(横轴表示  $yf(x, w)$ , 纵轴表示损失)



# 线性分类模型



线性模型	激活函数	损失/目标函数	损失/目标函数定义	优化方法
线性回归	-	平方损失	$(y_i - \mathbf{w}^T \mathbf{x}_i)^2$	最小二乘、梯度下降
对数几率回归	$\text{sigmoid}(\mathbf{w}^T \mathbf{x})$	二值交叉熵损失	$-y_i \log \sigma(\mathbf{w}^T \mathbf{x}_i) + (1 - y_i) \log(1 - \sigma(\mathbf{w}^T \mathbf{x}_i))$	梯度下降
Softmax分类	$\text{softmax}(\mathbf{W}^T \mathbf{x})$	交叉熵损失	$-y_i \log \text{softmax}(\mathbf{w}^T \mathbf{x}_i)$	梯度下降
线性判别分析	-	Fisher准则	$\frac{\ m_2 - m_1\ _2^2}{s_1^2 + s_2^2}$	广义特征值分解
感知器	$\text{sign}(\mathbf{w}^T \mathbf{x})$	0-1损失	$\begin{cases} 0, y \cdot f(\mathbf{x}_i) \geq 0 \\ 1, y \cdot f(\mathbf{x}_i) < 0 \end{cases}$	迭代优化
支持向量机	$\text{sign}(\mathbf{w}^T \mathbf{x})$	Hinge损失	$\max(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i)$	二次规划、SMO等



# 线性分类-Fisher线性判别分析

- 线性判别分析(linear discriminant analysis, LDA) 是一种基于**监督学习**的**分类/降维**方法, 也称为Fisher线性判别分析 (fisher's discriminant analysis, FDA) [Fisher 1936]。
- 对于一组具有标签信息的高维数据样本, LDA 利用其类别信息, 将其线性投影到一个低维空间上, 在低维空间中同一类别样本尽可能靠近, 不同类别样本尽可能彼此远离。

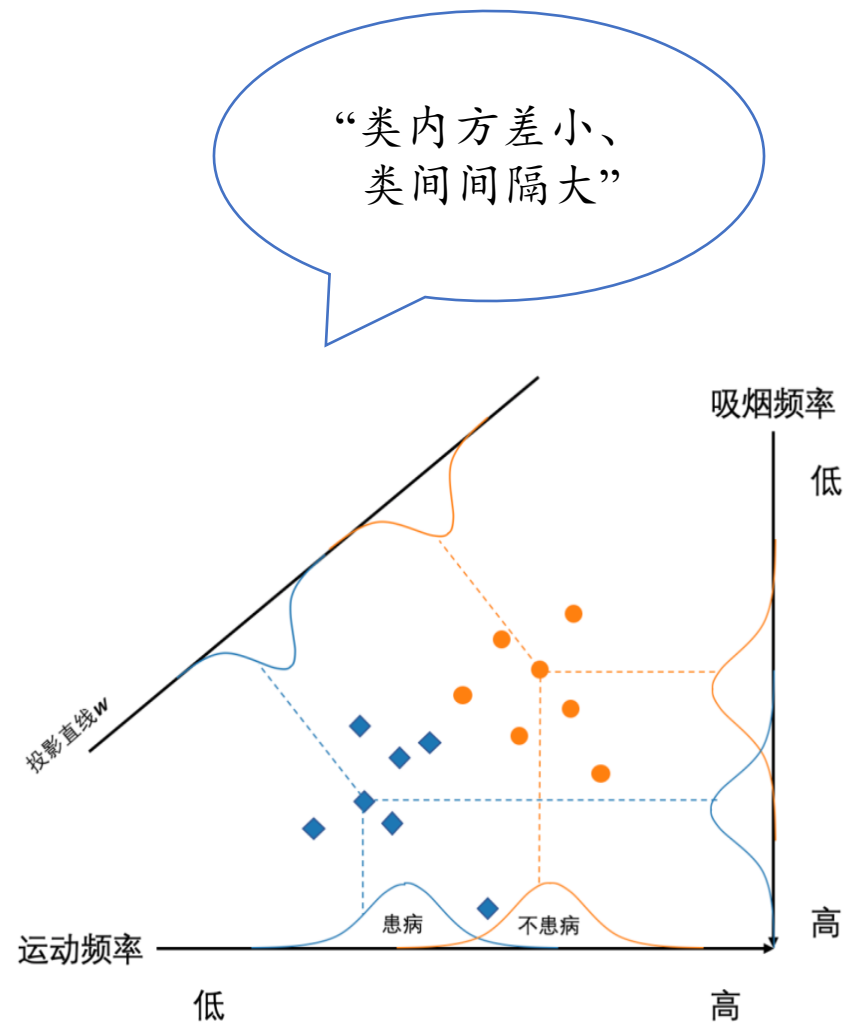


图4.8 两个类别数据所对应的不同投影方式  
君子而不同、小人同而不和



# Fisher线性判别分析：符号定义

- 假设样本集为  $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$ ，样本  $\mathbf{x}_i \in \mathbb{R}^d$  的类别标签为  $y_i$ 。其中， $y_i$  的取值范围是  $\{C_1, C_2, \dots, C_K\}$ ，即一共有  $K$  类样本。
  - 定义  $X$  为所有样本构成集合
  - $N_i$  为第  $i$  个类别所包含样本个数
  - $X_i$  为第  $i$  类样本的集合
  - $\mathbf{m}$  为所有样本的均值向量
  - $\mathbf{m}_i$  为第  $i$  类样本的均值向量
  - $\Sigma_i$  为第  $i$  类样本的散度矩阵，其定义为： $\Sigma_i = \sum_{\mathbf{x} \in X_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^T$



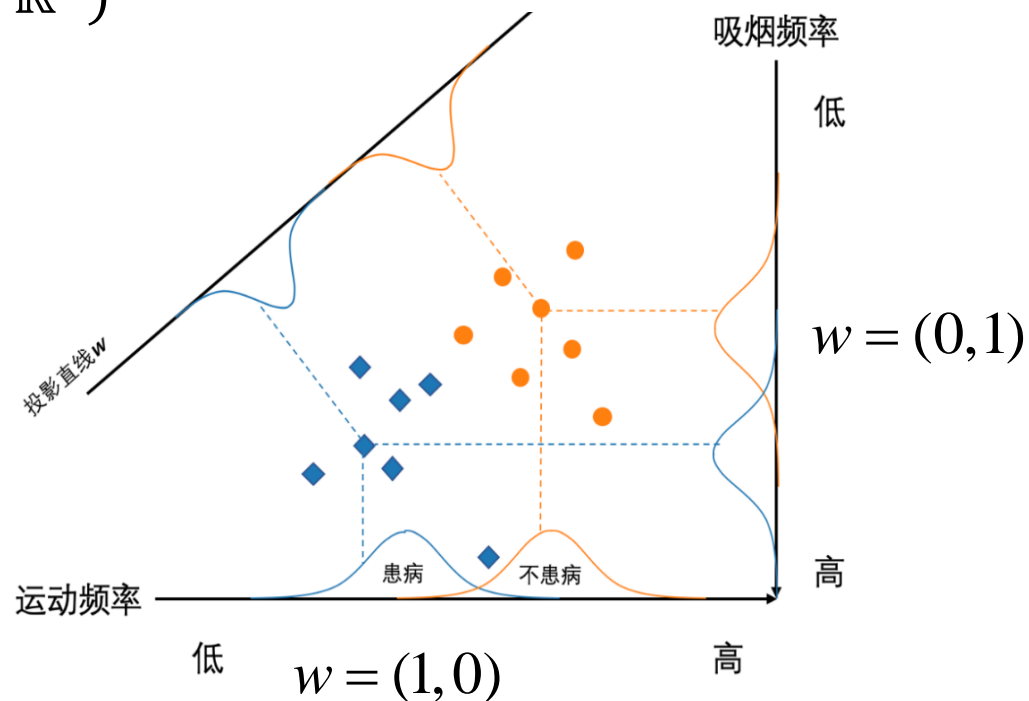
# Fisher线性判别分析：二分类问题

- 先来看 $K = 2$ 的情况，即二分类问题。在二分类问题中，训练样本归属于 $\mathcal{C}_1$ 或 $\mathcal{C}_2$ 两个类别，并通过如下的线性函数投影到一维空间上：

$$\hat{y}(\mathbf{x}) = \mathbf{w}^T \mathbf{x} \quad (\mathbf{w} \in \mathbb{R}^d)$$

数据点  $(1,1), (2,2), (3,3), (4,4) \dots$

都会投影到同一个点。





# Fisher线性判别分析：二分类问题

- 希望寻找一个投影方向 $\mathbf{w}$ ，使得两个类别的数据在投影以后 $\hat{y}(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$ 容易被分开

- 两个类各自的均值为

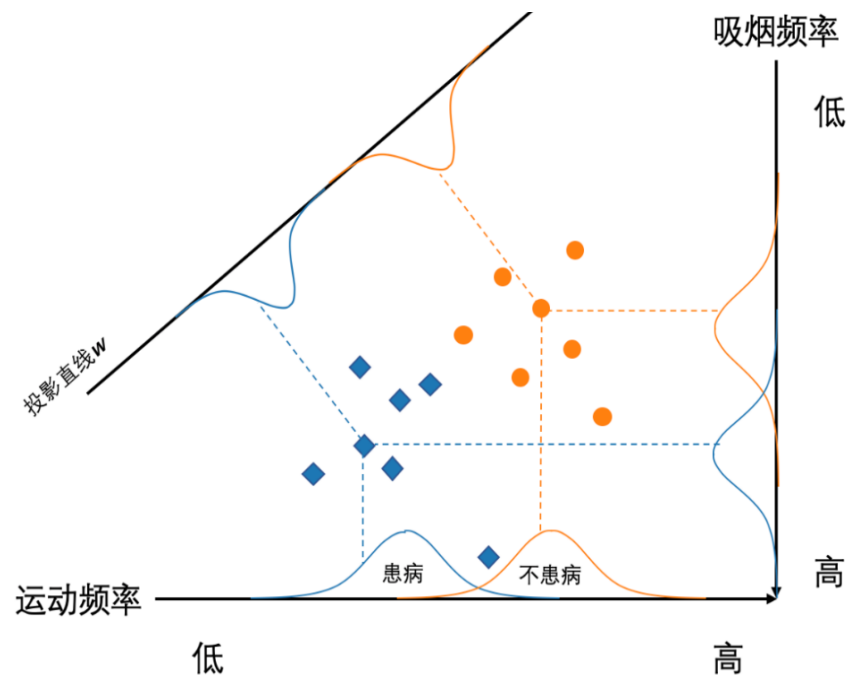
- $\mathbf{m}_1 = \frac{1}{N_1} \sum_{\mathbf{x} \in \mathcal{C}_1} \mathbf{x},$

- $\mathbf{m}_2 = \frac{1}{N_2} \sum_{\mathbf{x} \in \mathcal{C}_2} \mathbf{x}$

- 投影以后的均值为

- $m_1 = \mathbf{w}^T \mathbf{m}_1,$

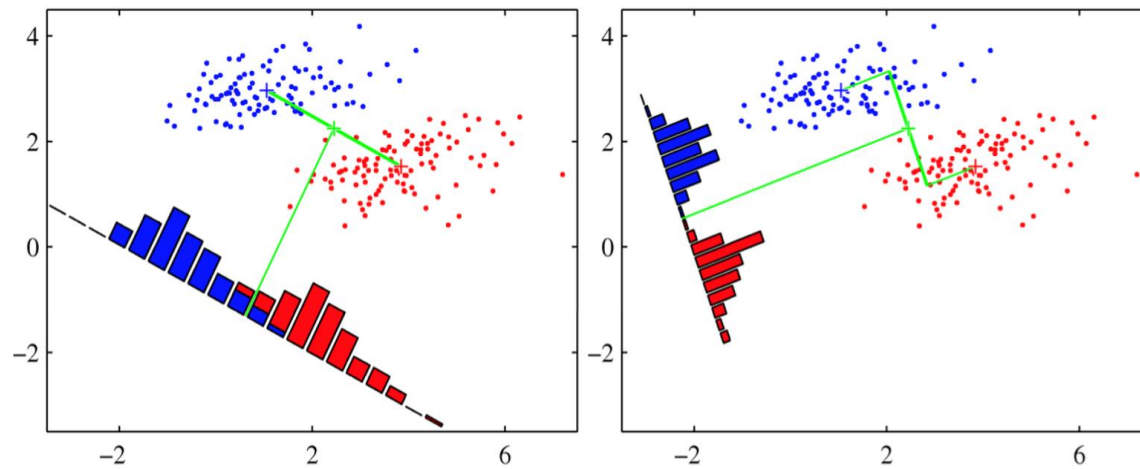
- $m_2 = \mathbf{w}^T \mathbf{m}_2$





# Fisher线性判别分析：二分类问题

- 怎样描述“分开”的程度？
- 最大化  $\|m_2 - m_1\|_2^2$  ?



下图哪个映射方向使得红色和蓝色两个类别更容易分开？

A

图 1

B

图 2

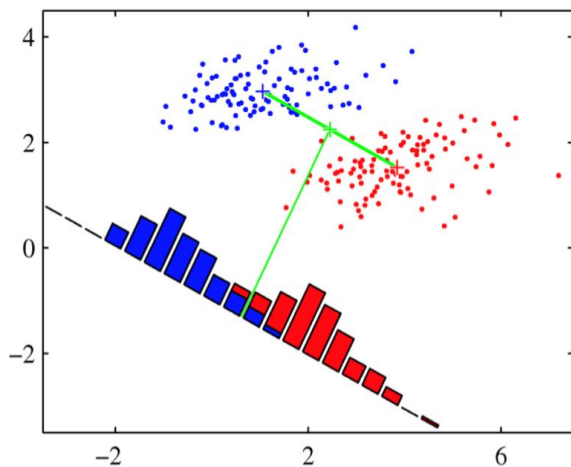


图 1

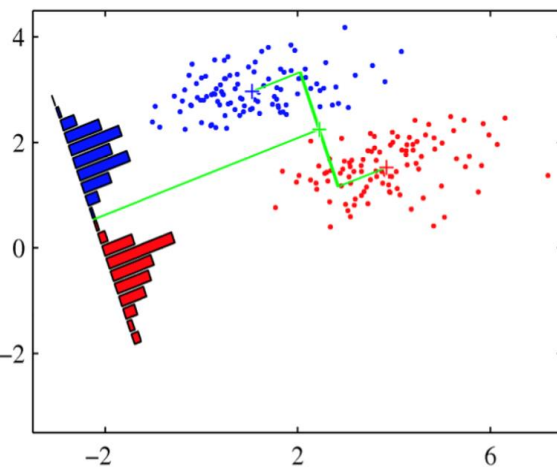


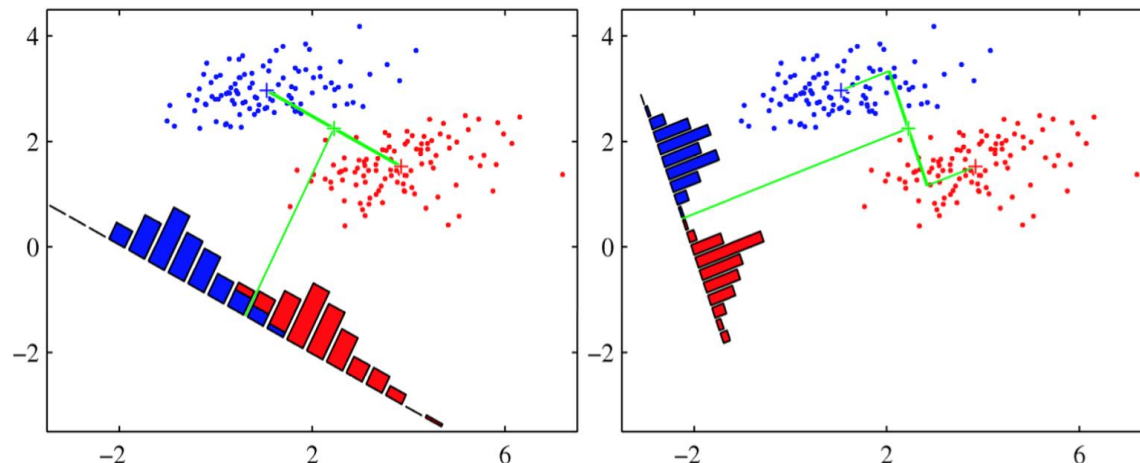
图 2

提交



# Fisher线性判别分析：二分类问题

- 怎样描述“分开”的程度？
- 最大化  $\|m_2 - m_1\|_2^2$ ？问题？
  - 这个值可以无限大。
  - 如图所示，这个值不是越大越好。





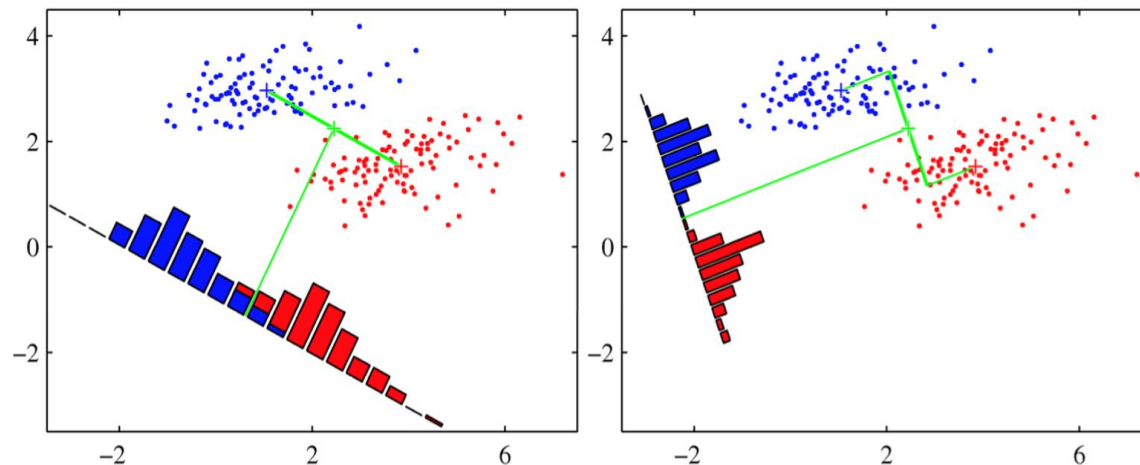
单纯最大化 $\|m_2 - m_1\|_2^2$ 并非最优，直觉上，需要加入什么样的额外约束？

作答



# Fisher线性判别分析：二分类问题

- 怎样描述“分开”的程度？
- 最大化 $\|m_2 - m_1\|_2^2$ ？问题？
  - 这个值可以无限大。
  - 如图所示，这个值不是越大越好。



## • Fisher准则

- 在要求 $\|m_2 - m_1\|_2^2$ 尽量大的同时，要求两类在投影以后尽量集中，或者不分散。怎么度量分散程度？

$$J(\mathbf{w}) = \frac{\|m_2 - m_1\|_2^2}{s_1^2 + s_2^2}$$



# Fisher线性判别分析：二分类问题

- **Fisher准则**：在要求 $\|m_2 - m_1\|_2^2$ 尽量大的同时，要求两类在投影以后尽量集中，或者不分散。怎么度量分散程度？

$$J(\mathbf{w}) = \frac{\|m_2 - m_1\|_2^2}{s_1^2 + s_2^2}$$

投影之后类别 $\mathcal{C}_1$ 的散度矩阵 $s_1$ 为：

$$s_1 = \sum_{x \in \mathcal{C}_1} (\mathbf{w}^T \mathbf{x} - \mathbf{w}^T \mathbf{m}_1)^2 = \mathbf{w}^T \sum_{x \in \mathcal{C}_1} [(\mathbf{x} - \mathbf{m}_1)(\mathbf{x} - \mathbf{m}_1)^T] \mathbf{w}$$

同理可得到投影之后类别 $\mathcal{C}_2$ 的散度矩阵 $s_2$ 。



# Fisher线性判别分析：二分类问题

$$J(\mathbf{w}) = \frac{\|\mathbf{m}_2 - \mathbf{m}_1\|_2^2}{s_1^2 + s_2^2} = \frac{\|\mathbf{w}^T(\mathbf{m}_2 - \mathbf{m}_1)\|_2^2}{\mathbf{w}^T \Sigma_1 \mathbf{w} + \mathbf{w}^T \Sigma_2 \mathbf{w}} = \frac{\mathbf{w}^T (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T \mathbf{w}}{\mathbf{w}^T (\Sigma_1 + \Sigma_2) \mathbf{w}} = \frac{\mathbf{w}^T \mathbf{S}_b \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w \mathbf{w}}$$

- 其中， $\mathbf{S}_b$ 称为**类间散度矩阵**(between-class scatter matrix)，即衡量两个类别均值点之间的“分离”程度，可定义如下：

$$\mathbf{S}_b = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T$$

- $\mathbf{S}_w$ 则称为**类内散度矩阵**(within-class scatter matrix)，即衡量每个类别中数据点的“分离”程度，可定义如下：

$$\mathbf{S}_w = \Sigma_1 + \Sigma_2$$

- 由于 $J(\mathbf{w})$ 的分子和分母都是关于 $\mathbf{w}$ 的二项式，因此最后的解只与 $\mathbf{w}$ 的方向有关，与 $\mathbf{w}$ 的长度无关，因此可令分母 $\mathbf{w}^T \mathbf{S}_w \mathbf{w} = 1$ ，然后用拉格朗日乘子法来求解这个问题。



# Fisher线性判别分析：二分类问题

对应拉格朗日函数为：

$$L(\mathbf{w}) = \mathbf{w}^T \mathbf{S}_b \mathbf{w} - \lambda(\mathbf{w}^T \mathbf{S}_w \mathbf{w} - 1)$$

- 参考解法：对 $\mathbf{w}$ 求偏导并使其求导结果为零，可得

$$\mathbf{S}_b \mathbf{w} = \lambda \mathbf{S}_w \mathbf{w}, \text{ 或 } \mathbf{S}_w^{-1} \mathbf{S}_b \mathbf{w} = \lambda \mathbf{w}$$

由此可见， $\lambda$ 和 $\mathbf{w}$ 分别是 $\mathbf{S}_w^{-1} \mathbf{S}_b$ 的特征根和特征向量

对矩阵和向量求导规则参见：

[https://www.sfu.ca/~haiyunc/notes/matrix\\_calculus.pdf](https://www.sfu.ca/~haiyunc/notes/matrix_calculus.pdf)

<https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf>



# Fisher线性判别分析：以二分类为例

- 训练数据：  $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ ，其中  $y_i \in \{0, 1\}$
- 学习模型：  $f(x_i) = \mathbf{w}^T \mathbf{x}_i + w_0$
- 损失函数（可写为）：  $L(\mathbf{w}) = \mathbf{w}^T \mathbf{S}_b \mathbf{w} - \lambda(\mathbf{w}^T \mathbf{S}_w \mathbf{w} - 1)$
- 优化方法：广义特征值分解



# Fisher线性判别分析：降维/分类器学习步骤

- 对Fisher线性判别分析的降维/分类器学习步骤总结如下：
  1. 计算数据样本集中每个类别样本的均值
  2. 计算类内散度矩阵 $S_w$ 和类间散度矩阵 $S_b$
  3. 根据 $S_w^{-1}S_bW = \lambda W$ 来求解 $S_w^{-1}S_b$ 所对应前 $r$ 个最大特征值所对应特征向量 $(w_1, w_2, \dots, w_r)$ ，构成矩阵 $W$
  4. 通过矩阵 $W$ 将每个样本映射到低维空间，实现特征降维或分类器学习。



北京航空航天大学  
COLLEGE OF SOFTWARE  
BEIHANG UNIVERSITY 软件学院

# 提纲

**一、机器学习基本概念**

**二、线性回归与线性分类**

**三、线性判别分析**

**四、支持向量机**

**五、决策树**

**六、集成学习**





北京航空航天大学  
COLLEGE OF SOFTWARE BEIHANG UNIVERSITY  
软件学院

# 线性分类-支持向量机：从问题入手

- 已知在二维空间中两类数据，表示为“+”和“-”
- 问题：请画一条直线，实现以下要求：
  - 这条直线能将两类数据**区分开**
  - 当有新数据加入时，能通过这条直线**判别新数据属于哪一类**

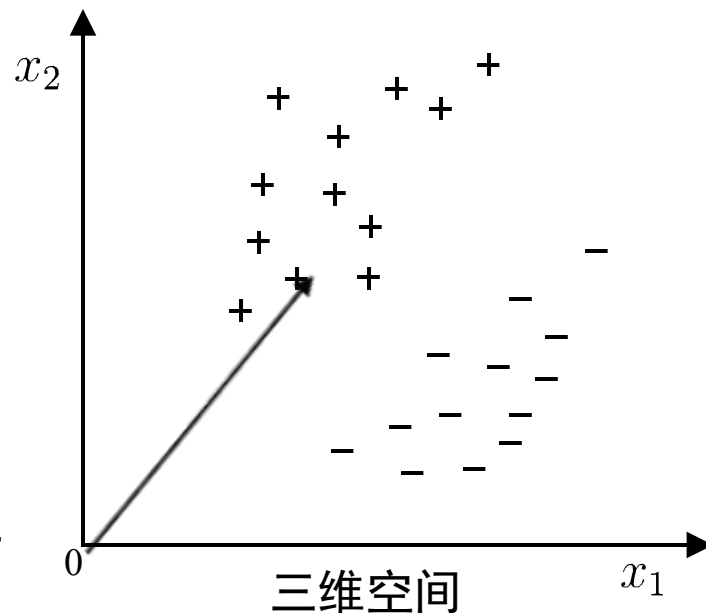
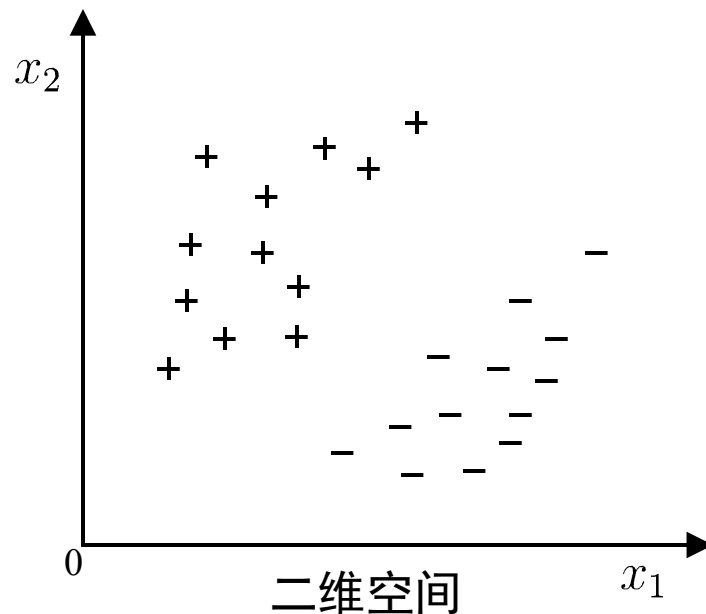


延伸到三维空间：如何让一个二维平面区分两类数据

四维、五维.....

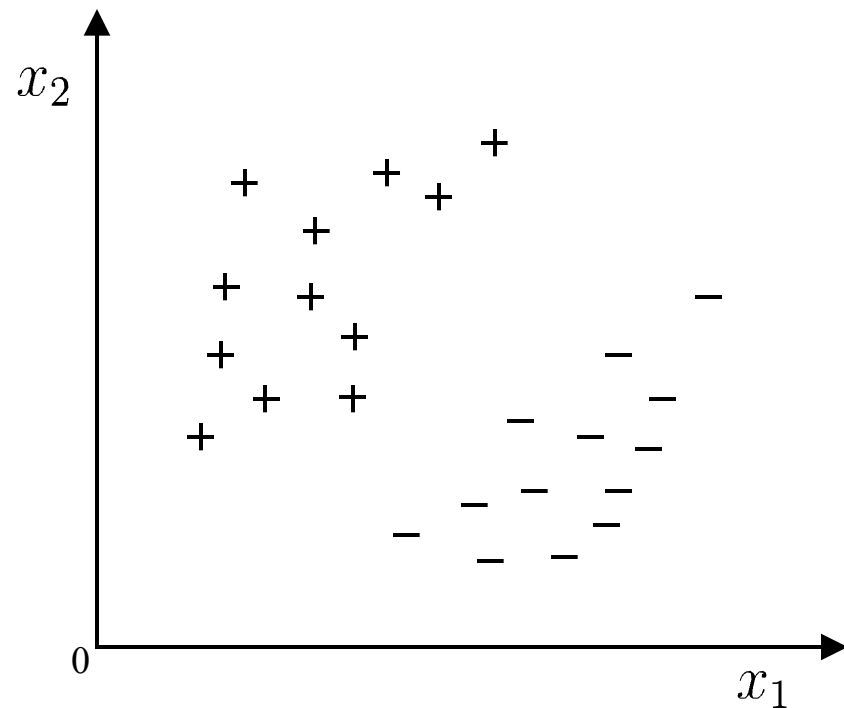


在M维空间中，如何设计一个**M-1维的超平面**区分两类数据



右图所示数据描述的二分类问题存在多少个可以完美分类的分类界线？

- ☐ A 0
- ☐ B 1
- ☐ C 2
- ☒ D 无限多



提交

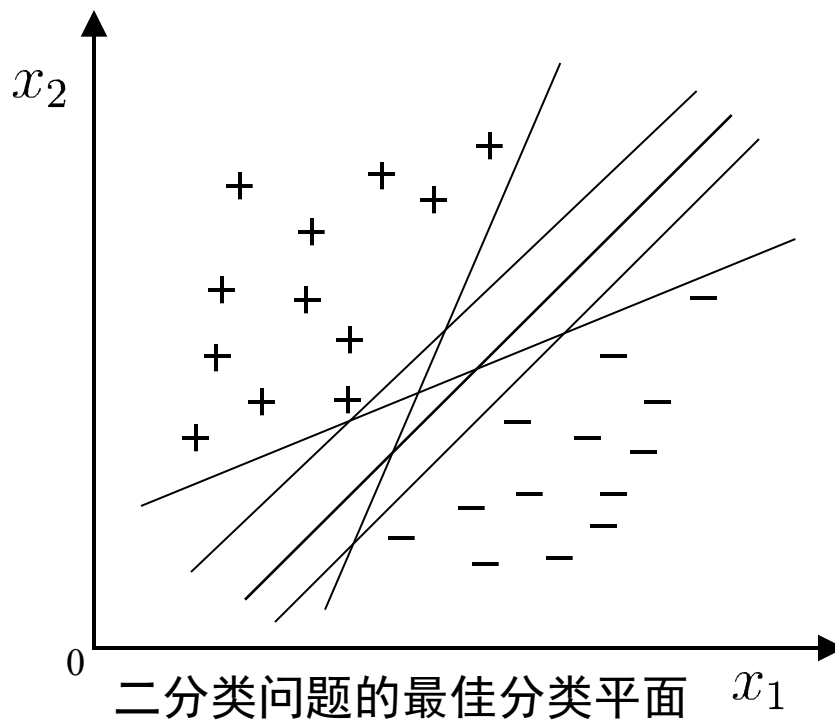


# 线性分类-从感知器模型到支持向量机

- 感知器模型

$$\hat{y}(\mathbf{x}) = g(\mathbf{w}^T \mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x}), \text{ 其中 } g(z) = \begin{cases} +1, & z \geq 0 \\ -1, & z < 0 \end{cases}$$

- $g(\mathbf{w}^T \mathbf{x}) > 0 \Rightarrow \mathcal{C}_1$
- $g(\mathbf{w}^T \mathbf{x}) < 0 \Rightarrow \mathcal{C}_2$





# 感知器模型

- 训练数据:  $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ , 其中  $y_i \in \{-1, +1\}$
- 学习模型:  $f(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i + w_0$
- 损失函数:  $L(\mathbf{w}) = \begin{cases} 0, & y \cdot f(\mathbf{x}_i) \geq 0 \\ 1, & y \cdot f(\mathbf{x}_i) < 0 \end{cases}$  (直接计算分类错误的次数)
- 优化方法: 迭代优化

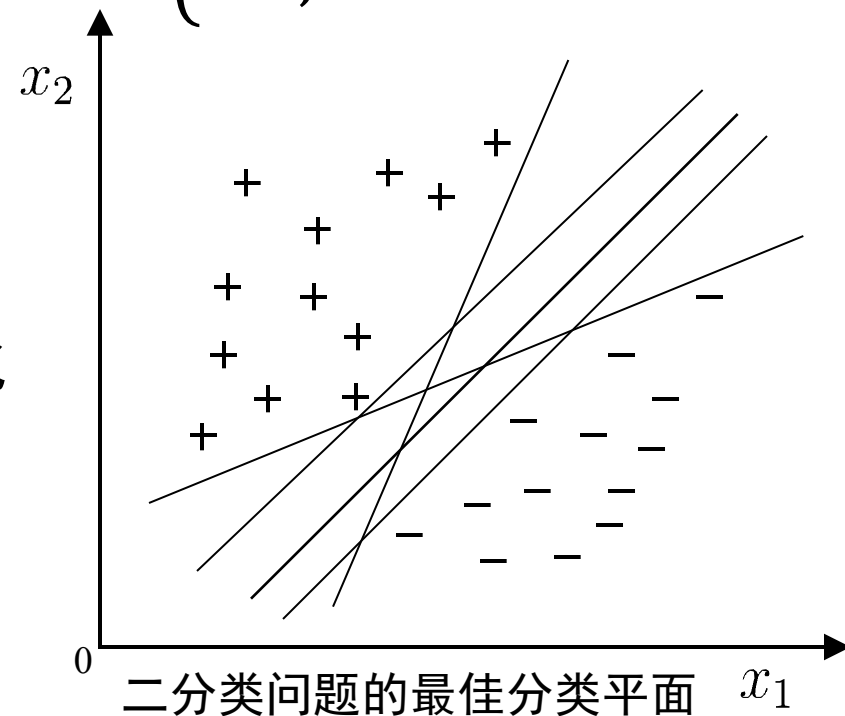


# 线性分类-从感知器模型到支持向量机

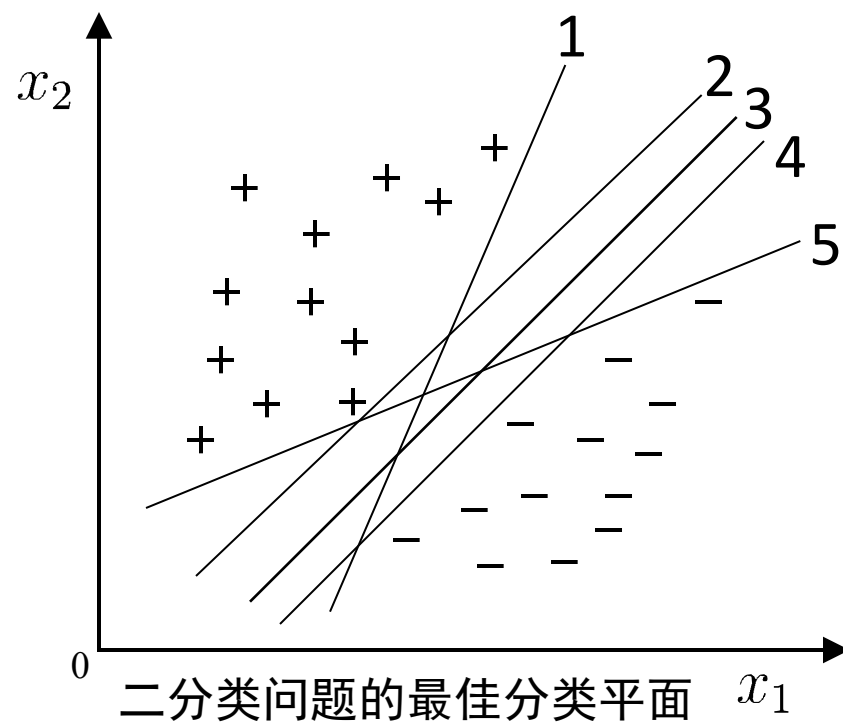
- 感知器模型

$$\hat{y}(\mathbf{x}) = g(\mathbf{w}^T \mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x}), \text{ 其中 } g(z) = \begin{cases} +1, & z \geq 0 \\ -1, & z < 0 \end{cases}$$

- $g(\mathbf{w}^T \mathbf{x}) > 0 \Rightarrow \mathcal{C}_1$
- $g(\mathbf{w}^T \mathbf{x}) < 0 \Rightarrow \mathcal{C}_2$
- 问题1: 阶跃函数（分段常数函数）难以优化
- 问题2: 无限多完全正确分类解, 哪个最佳?



直觉上，你认为这5个分类界线中哪个最优？为什么？

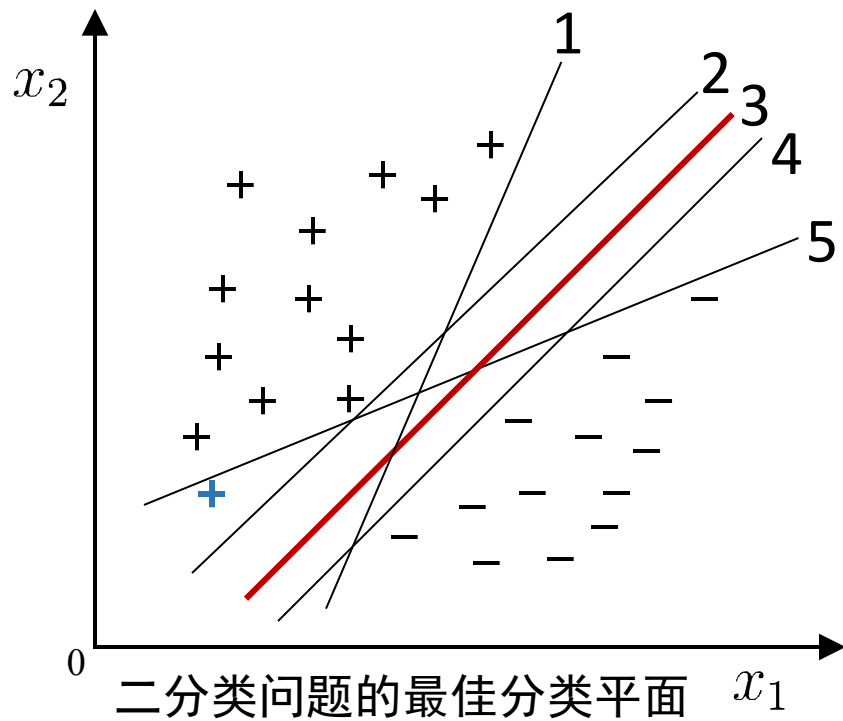


作答



# 线性分类-线性可分支持向量机

直觉上，你认为这5个分类界线中哪个最优？为什么？



- 界线1、界线5:

- 两类数据都有相应的数据点离分类界限**非常接近**，当有一个新的数据同样接近该直线时，**分类错误的概率非常大**（如图中新增的蓝色“+”数据）。

- 界线3:

- 两类数据都与分类界线**保持了一定距离**（称为“间隔”），起到缓冲区的作用。**间隔越大，两类数据差异越大**，区分起来越容易。



无限多完全正确分类解，怎么找出最佳？  
**最大化分类间隔**



# 支持向量机：线性可分支持向量机

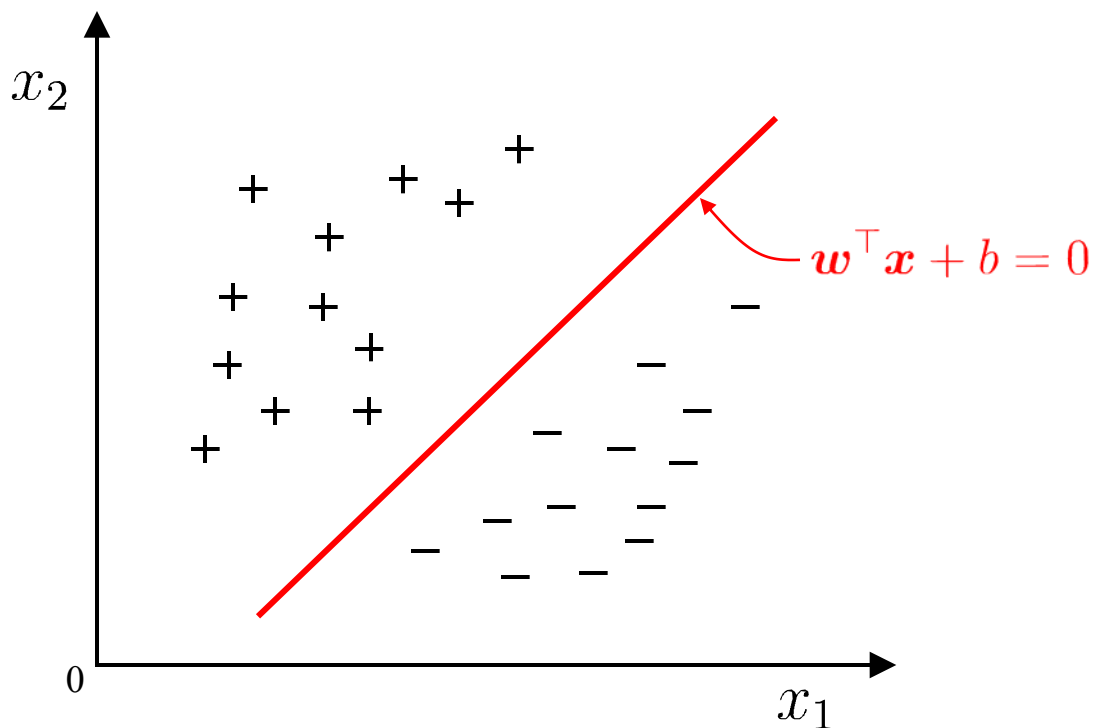
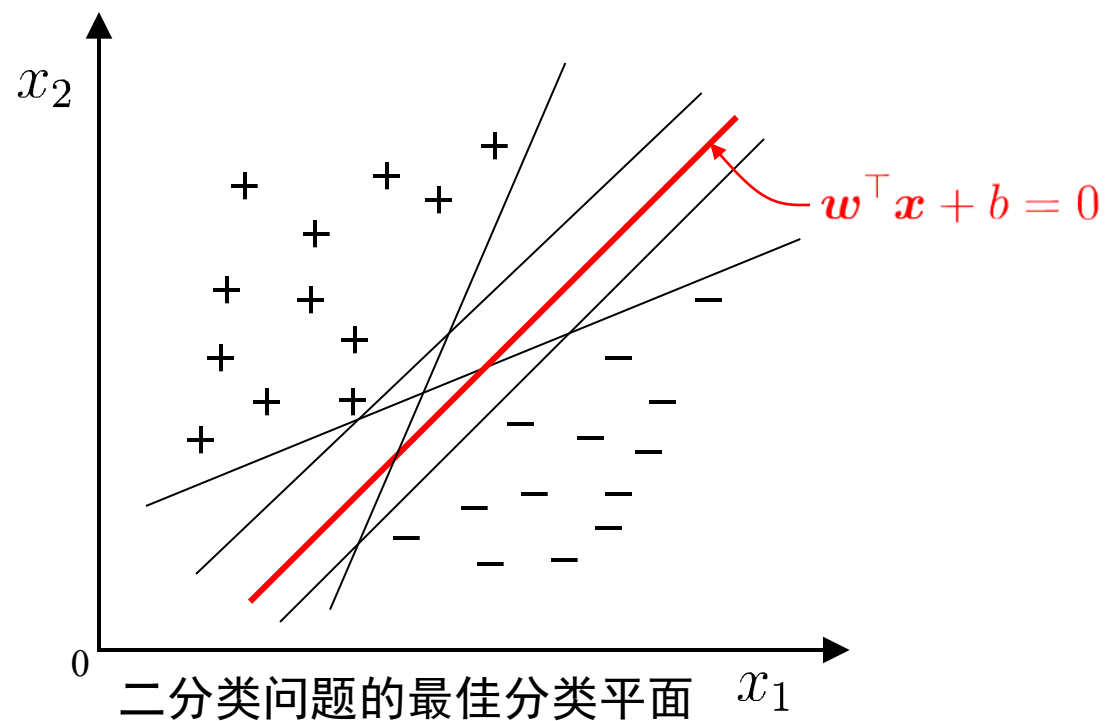


图4.12 线性分类中分类平面

- 寻找一个最优的超平面，其方程为
$$\mathbf{w}^T \mathbf{x} + b = 0$$
- 这里  $\mathbf{w} = (w_1, w_2, \dots, w_d)$  为超平面的法向量，与超平面的方向有关；
- $b$  为偏置项，是一个标量，其决定了超平面与原点之间的距离。
- 使得分类间隔最大！



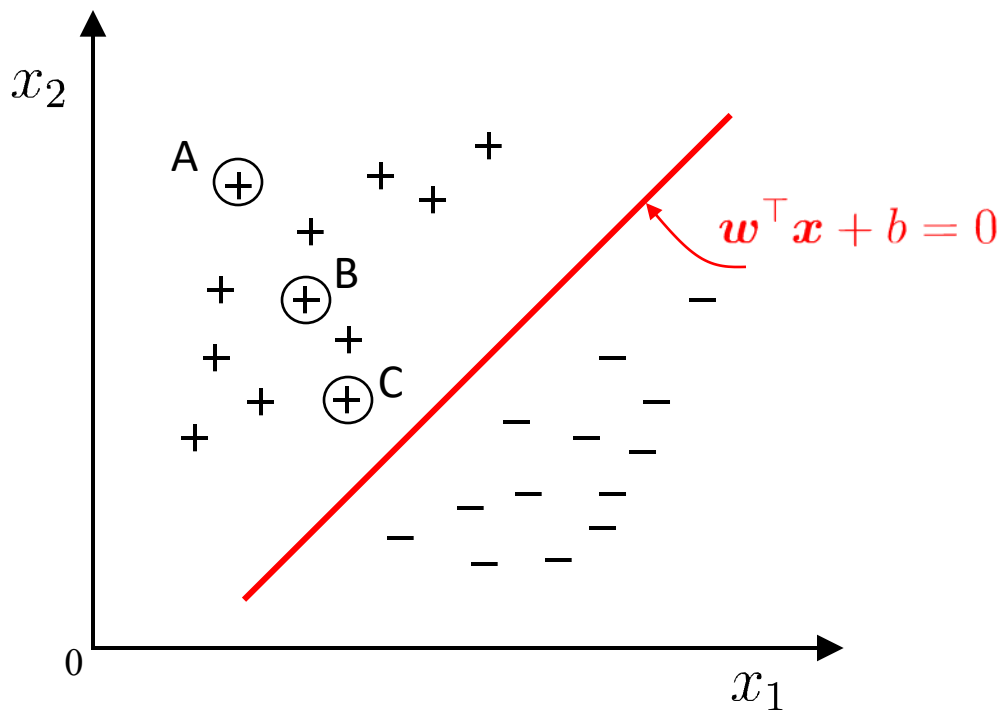
直觉上，如何找到这条能够使得分类间隔最大的分类界面？



作答



# 支持向量机： 线性可分支持向量机



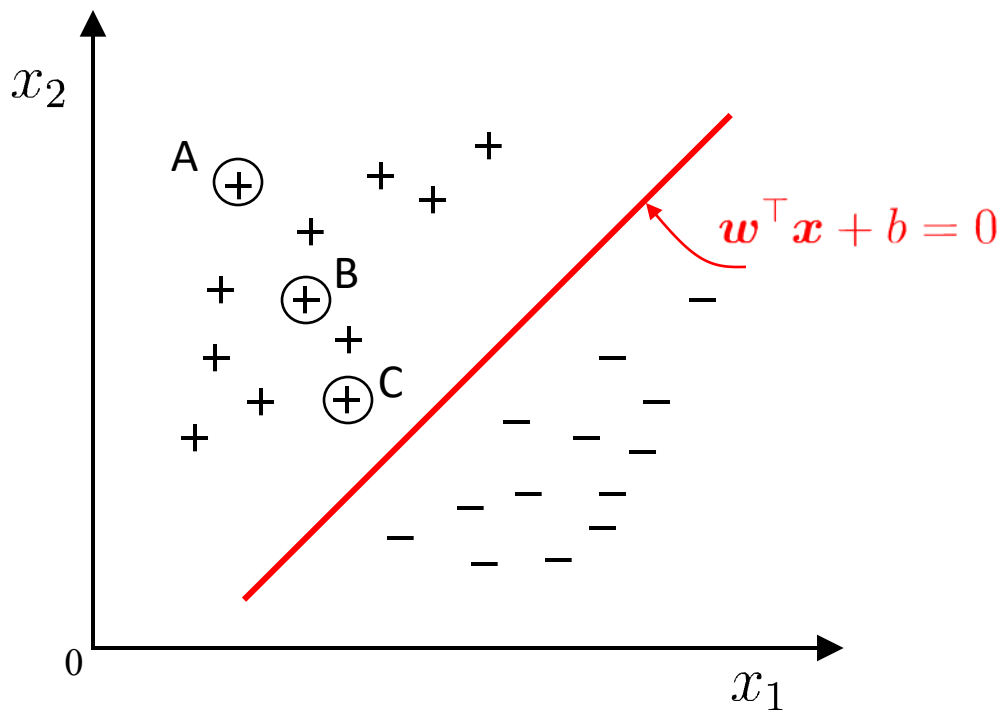
- **函数间隔**：对于数据点  $(\mathbf{x}_i, y_i)$ ，其中  $y_i \in \{-1, 1\}$  是标签，超平面由  $\mathbf{w}^T \mathbf{x} + b = 0$  定义，函数间隔定义为：

$$\hat{y}_i = y_i (\mathbf{w}^T \mathbf{x}_i + b)$$

- 如果分类**正确**， $y_i (\mathbf{w}^T \mathbf{x}_i + b) > 0$ ，函数间隔为**正**。
- 如果分类**错误**， $y_i (\mathbf{w}^T \mathbf{x}_i + b) < 0$ ，函数间隔为**负**。
- 函数间隔的绝对值越大，表示分类的置信度越高。
- 所以  $y(\mathbf{w}^T \mathbf{x}_i + b)$  可以用来表示分类的**正确性及置信度**。



# 支持向量机： 线性可分支持向量机



- **几何间隔：**对于数据点  $(x_i, y_i)$ ，其中  $y_i \in \{-1, 1\}$  是标签，几何间隔定义为：

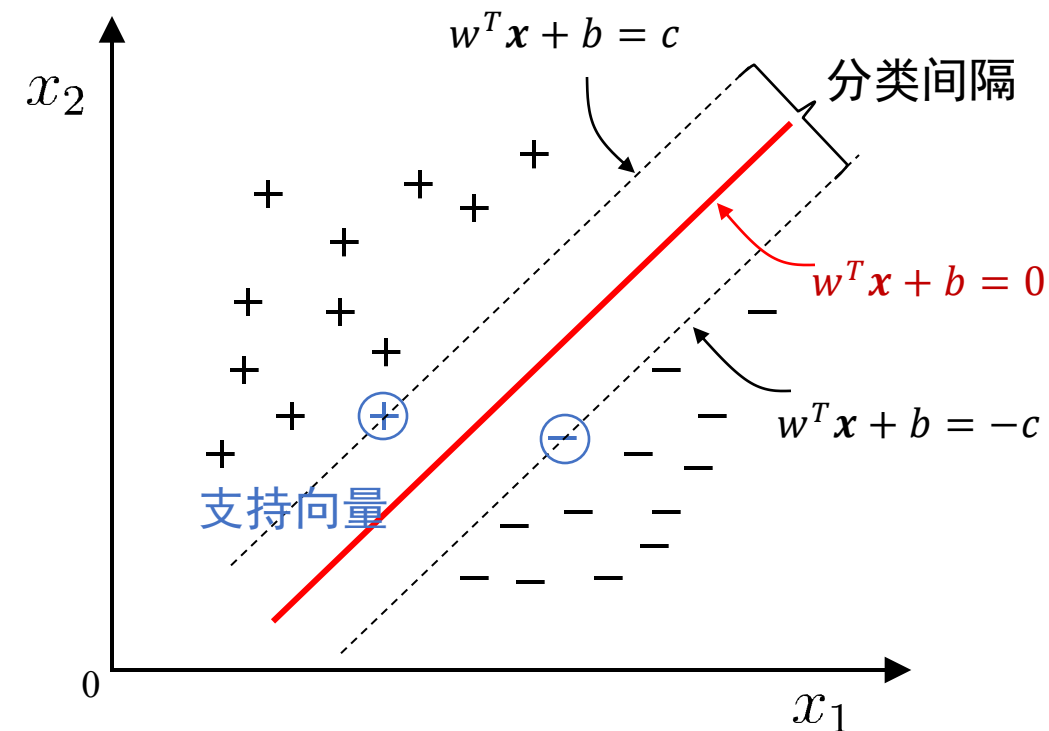
$$\gamma_i = \frac{y_i(\mathbf{w}^T \mathbf{x}_i + b)}{\|\mathbf{w}\|_2}$$

- 其中， $\|\mathbf{w}\|_2 = \sqrt{\mathbf{w}^T \mathbf{w}}$
- 几何间隔是数据点到超平面的垂直距离，是一个几何意义上的距离。
- **对  $\mathbf{w}$  和  $b$  同时缩放，几何间隔不变。**

$$\frac{cy_i(\mathbf{w}^T \mathbf{x}_i + b)}{\|c\mathbf{w}\|_2} = \frac{cy_i(\mathbf{w}^T \mathbf{x}_i + b)}{\sqrt{c\mathbf{w}^T c\mathbf{w}}} = \frac{y_i(\mathbf{w}^T \mathbf{x}_i + b)}{\sqrt{\mathbf{w}^T \mathbf{w}}}$$



# 支持向量机：线性可分支持向量机：定义



- 假设分类边界的超平面方程为：

$$w_1 x_1 + w_2 x_2 + b = 0$$

记为： $w^T x + b = 0$

- 分别将分类边界上下移动 $c$ ，直至分类间隔的边界，得到对应的上、下间隔边界：

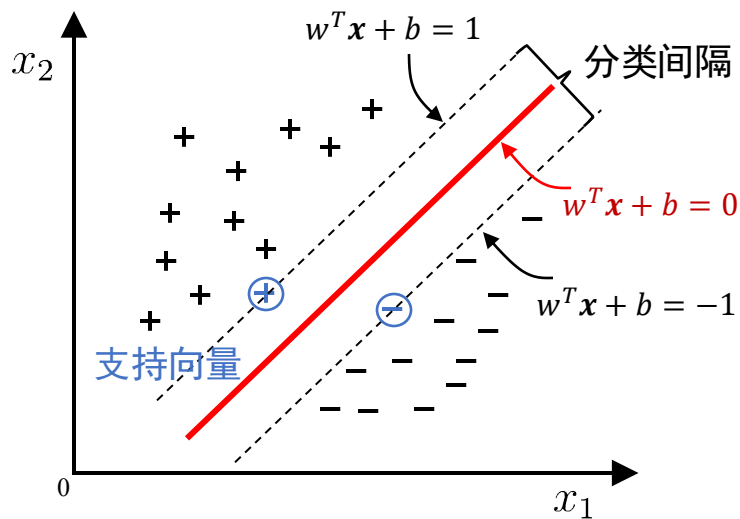
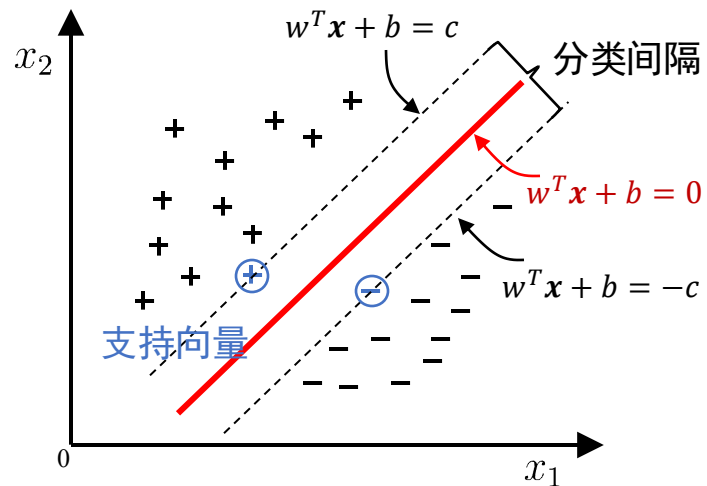
$$w^T x + b = c$$

$$w^T x + b = -c$$

- 上、下间隔边界会经过一些样本数据点，这些点距离分类边界最近，被称为“**支持向量**”（图中蓝色数据）



# 支持向量机：线性可分支持向量机：定义



- 分类边界和上、下间隔边界的方程为：

$$\mathbf{w}^T \mathbf{x} + b = 0$$

$$\mathbf{w}^T \mathbf{x} + b = c$$

$$\mathbf{w}^T \mathbf{x} + b = -c$$

- 对  $\mathbf{w}$  和  $b$  同时缩放，几何间隔不变，因此可以同时除  $c$ ：

$$\frac{\mathbf{w}^T}{c} \mathbf{x} + \frac{b}{c} = 0$$

$$\frac{\mathbf{w}^T}{c} \mathbf{x} + \frac{b}{c} = 1$$

$$\frac{\mathbf{w}^T}{c} \mathbf{x} + \frac{b}{c} = -1$$

- 记：  $\mathbf{w}^T = \frac{\mathbf{w}^T}{c}$ ，  $b = \frac{b}{c}$ ，得到新的分类边界和上、下间隔边界方程：

$$\mathbf{w}^T \mathbf{x} + b = 0$$

$$\mathbf{w}^T \mathbf{x} + b = 1$$

$$\mathbf{w}^T \mathbf{x} + b = -1$$



# 支持向量机：线性可分支持向量机

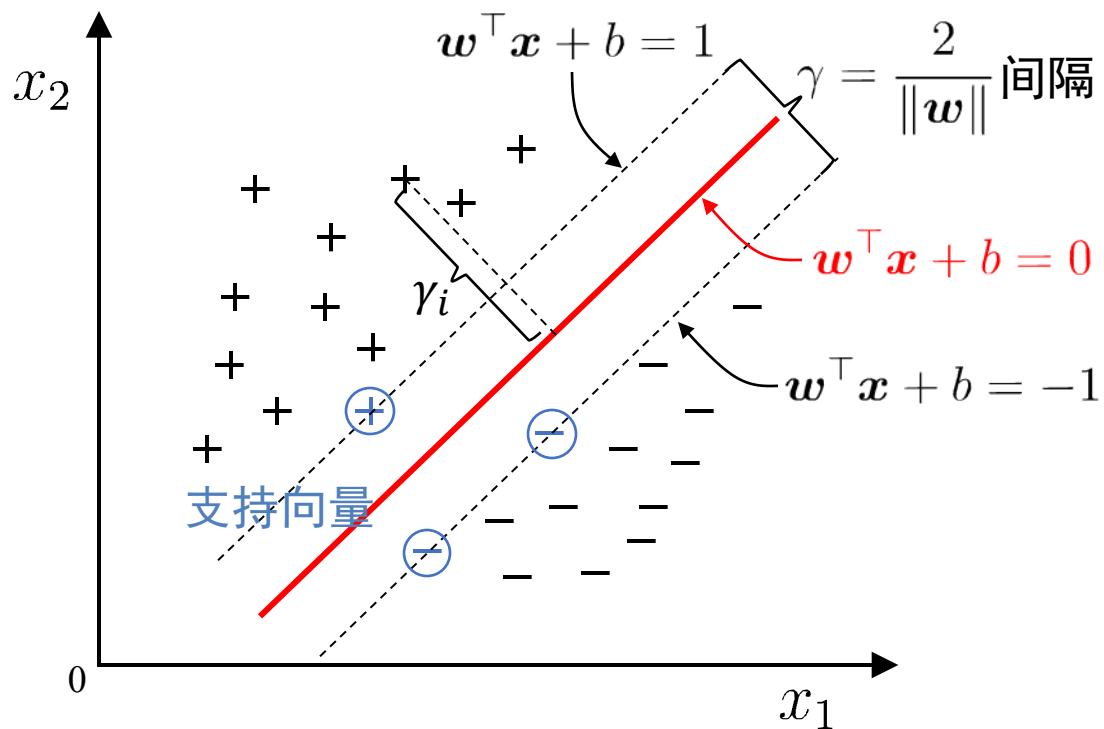


图4.12 线性分类中分类平面及其支持向量

- 根据上、下间隔边界方程：

$$\mathbf{w}^T \mathbf{x} = 1 - b$$

$$\mathbf{w}^T \mathbf{x} = -1 - b$$

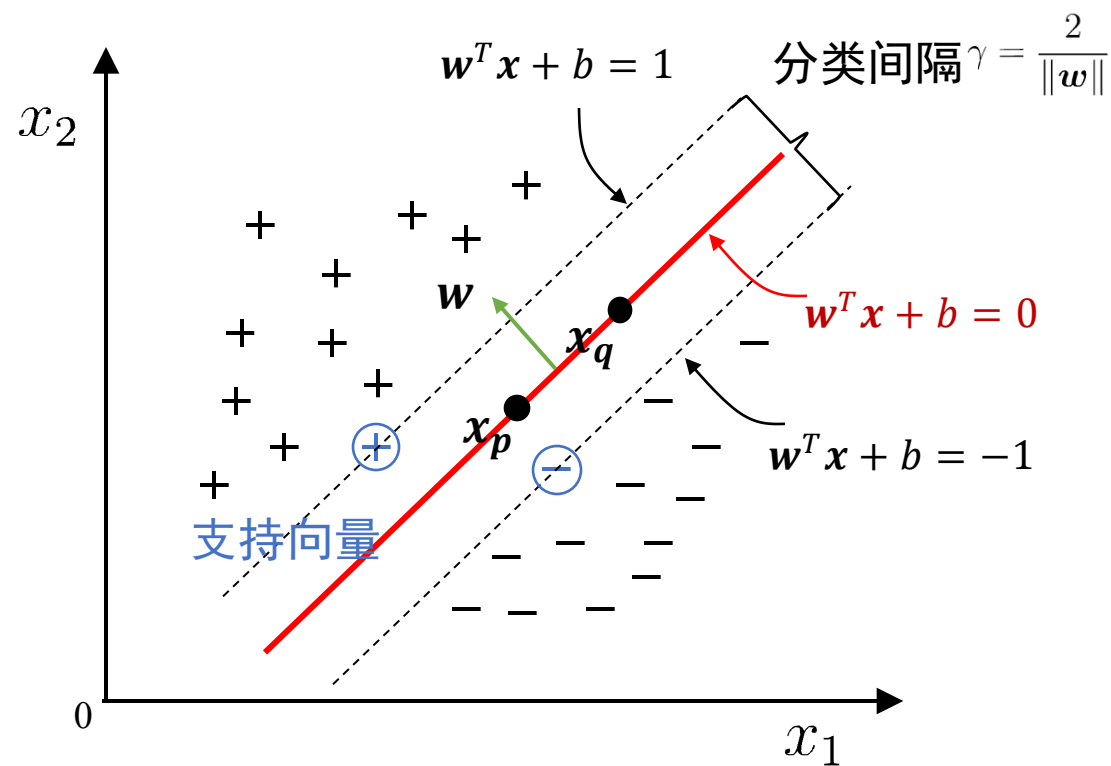
- 计算上、下间隔边界几何距离，也就是两个平行超平面几

何距离：
$$\gamma = \frac{|1-b - (-1-b)|}{\|\mathbf{w}\|_2} = \frac{2}{\|\mathbf{w}\|_2}$$

- $\gamma = \frac{2}{\|\mathbf{w}\|_2}$ 即为“分类间隔” (margin)



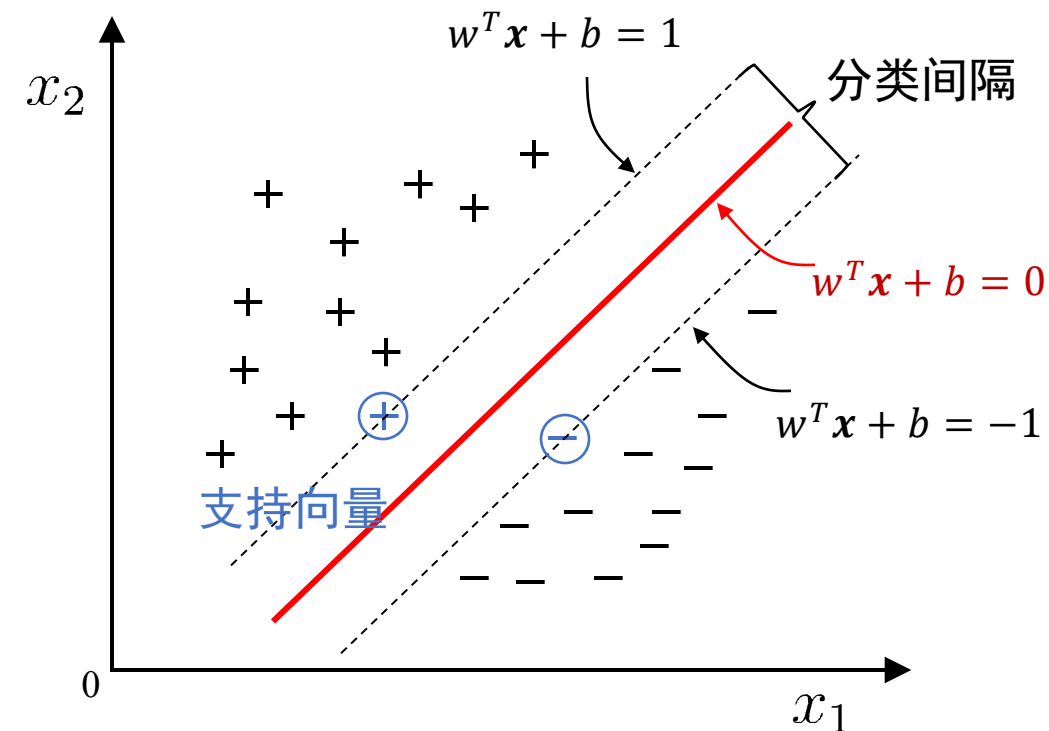
# 支持向量机：线性可分支持向量机：分类间隔的表达式



- 在分类边界上任选两个点，记为  $x_p$  和  $x_q$
- 满足：①  $w^T x_p + b = 0$  和 ②  $w^T x_q + b = 0$
- ①-②得：③  $w^T (x_p - x_q) = 0$
- ③说明向量  $w^T$  和向量  $(x_p - x_q)$  垂直
- 即  $w^T$  方向与分类界面垂直



# 支持向量机：线性可分支持向量机：约束条件

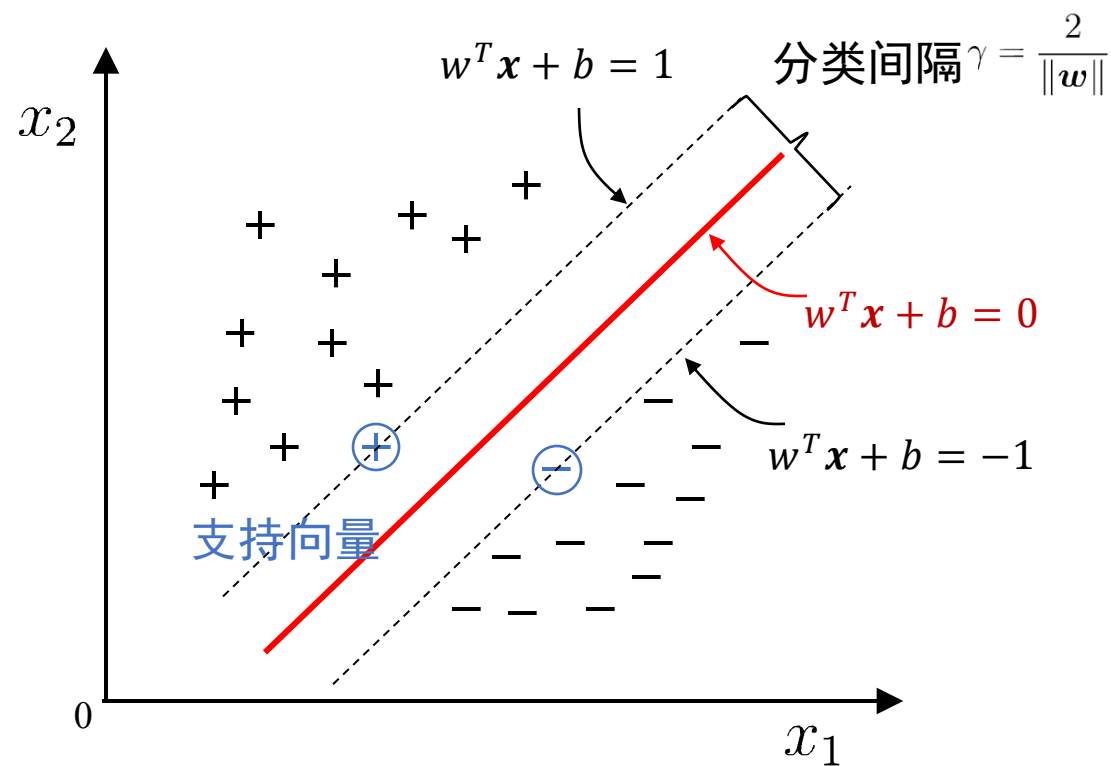


- 由于“+”样本都在 $w^T x + b = 1$ 上方，即 $(w^T x_i + b) \geq 1$   
所以， $y_i(w^T x_i + b) \geq 1$ ，其中 $x_i$ 为“+”样本， $y_i$ 为标签，符号为正，定义为1
  - 由于“-”样本都在 $w^T x + b = -1$ 下方，即 $(w^T x_i + b) \leq -1$   
所以， $y_i(w^T x_i + b) \leq -1$ ，其中 $x_i$ 为“-”样本， $y_i$ 为标签，符号为负，定义为-1
- ↓
- 于是对超平面的约束变为：  
$$y_i(w^T x_i + b) \geq 1$$
  - 其中，满足等号成立的样本被称为支持向量（support vector）





# 支持向量机：线性可分支持向量机：优化目标



- 支持向量机的基本形式就是最大化分类间隔
- 即在满足约束的条件下找到参数 $\mathbf{w}$ 和 $b$ 使得 $\gamma$ 最大,
- 最大化 $\gamma = \frac{2}{\|\mathbf{w}\|_2}$ 与最小化 $\frac{\|\mathbf{w}\|_2^2}{2}$ 等价, 即:

$$\min_{\mathbf{w}, b} \frac{\|\mathbf{w}\|_2^2}{2} = \min_{\mathbf{w}, b} \frac{1}{2} \mathbf{w}^T \mathbf{w}$$

$$\text{s.t. } y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, i = 1, 2, \dots, n$$

- 可以使用拉格朗日乘子法求解



北京航空航天大学  
COLLEGE OF SOFTWARE BEIHANG UNIVERSITY  
软件学院

# 支持向量机： 线性可分支持向量机： 拉格朗日乘子法求解

bilibili 首发

@FunInCode

优化问题  
minimize  $\|\vec{w}\|$

subject to  $y_i * (\vec{w} \cdot \vec{x}_i + b) \geq 1, i = 1, 2, 3, 4 \dots s, s$ 为全部样本数



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

subject to  $g_i(w, b) = y_i * (\vec{w} \cdot \vec{x}_i + b) - 1 \geq 0, i = 1, 2, 3, 4 \dots s, s$ 为全部样本数

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



# 支持向量机： 线性可分支持向量机

- 还可以通过拉格朗日对偶性（Lagrange Duality） 变换到对偶变量 (dual variable) 的优化问题

$$\min_{\mathbf{w}, b} \frac{\|\mathbf{w}\|^2}{2} = \min_{\mathbf{w}, b} \frac{1}{2} \mathbf{w}^T \mathbf{w}$$

$$\text{s.t. } y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, i = 1, 2, \dots, n$$

引入拉格朗日乘子  $\alpha$

$$\min_{\mathbf{w}, b} L(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i (y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1)$$

$$\text{s.t. } \alpha_i \geq 0, i = 1, 2, \dots, n$$



北京航空航天大学  
COLLEGE OF SOFTWARE  
BEIHANG UNIVERSITY 软件学院

# 支持向量机： 线性可分支支持向量机： 对偶问题

bilibili 首发

@FunInCode

## SVM对偶性

我们往往会将原问题转换为其自身的对偶问题



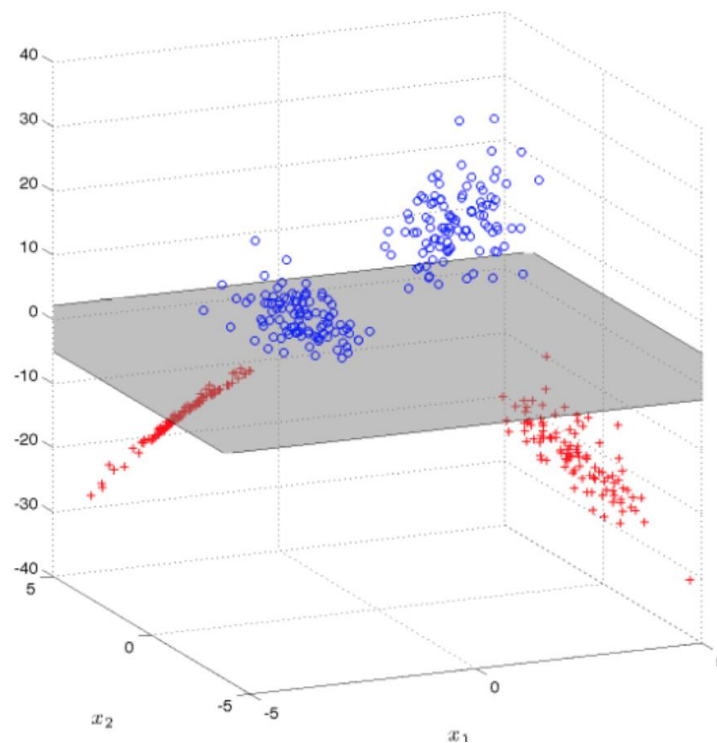
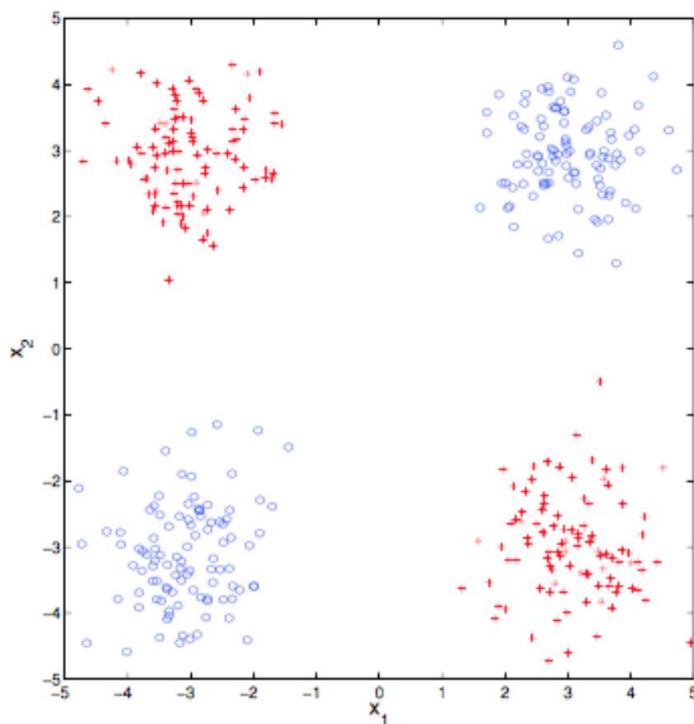
# 支持向量机：线性可分支持向量机

- 支持向量机的求解转换为对偶问题的好处是
  - 仅需求解一个变量 $\alpha$
  - 对偶问题约束方程简单
  - 优化问题可以转化为高效算法，如SOM（Sequential Minimal Optimization）
  - 模型转化为输入样本之间的内积形式，便于核函数的引入。



# 支持向量机：线性不可分-核函数

- 将线性不可分样本从原始空间映射到一个更加高维的特征空间中去，使得样本在这个特征空间中高概率线性可分。
- 如果原始空间是有限维，那么一定存在一个高维特征空间使样本可分[Shawe-Taylor, J. 2004]。





北京航空航天大学  
COLLEGE OF SOFTWARE  
BEIHANG UNIVERSITY 软件学院

# SVM核函数

bilibili 首发

@FunInCode

## SVM核技巧



核技巧 Kernel Trick



# 支持向量机：线性不可分-核函数

- 常见的核函数包括：

常见核函数

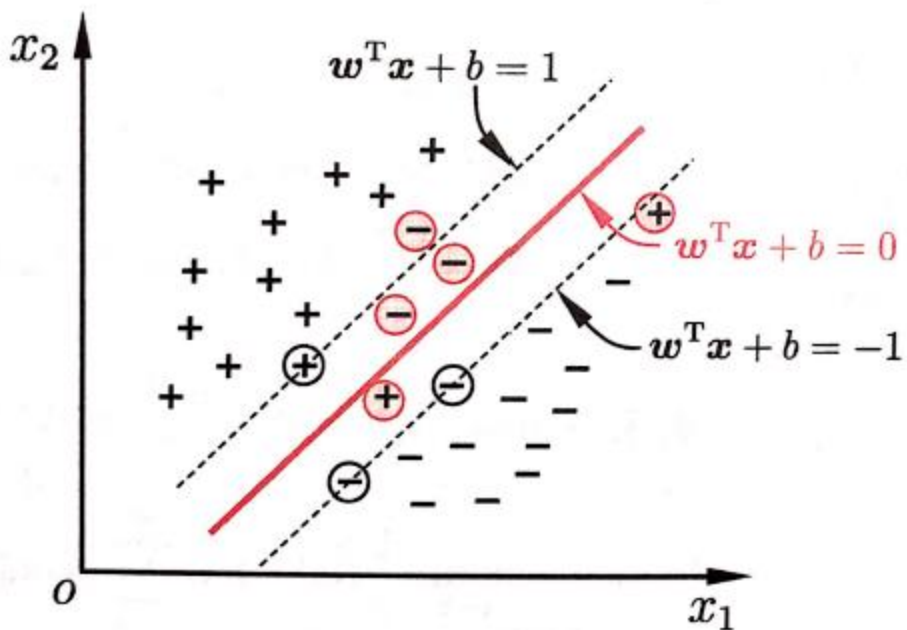
线性	$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$
多项式	$K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i^T \mathbf{x}_j + r)^n$
Radial Basis function	$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\ \mathbf{x}_i - \mathbf{x}_j\ ^2}{2\sigma^2})$
Sigmoid	$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma \mathbf{x}_i^T \mathbf{x}_j + r)$





# 支持向量机：线性不可分-松弛变量，软间隔与hinge损失函数

- 先前介绍中假设所有训练样本数据是线性可分，即存在一个线性超平面能将不同类别样本完全隔开，这种情况称为“硬间隔”（hard margin），与硬间隔相对的是“软间隔”（soft margin）。软间隔指允许部分错分给定的训练样本。



$$\min_{\mathbf{w}, b} \frac{1}{2} \mathbf{w}^T \mathbf{w} + c \times \sum_{i=1}^n \mathbb{I}[y_i \neq \text{sign}(\mathbf{w}^T \mathbf{x}_i + b)]$$

$$\text{s.t. } y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \text{ for correct } \mathbf{x}_i$$

$$y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq -\infty \text{ for incorrect } \mathbf{x}_i$$

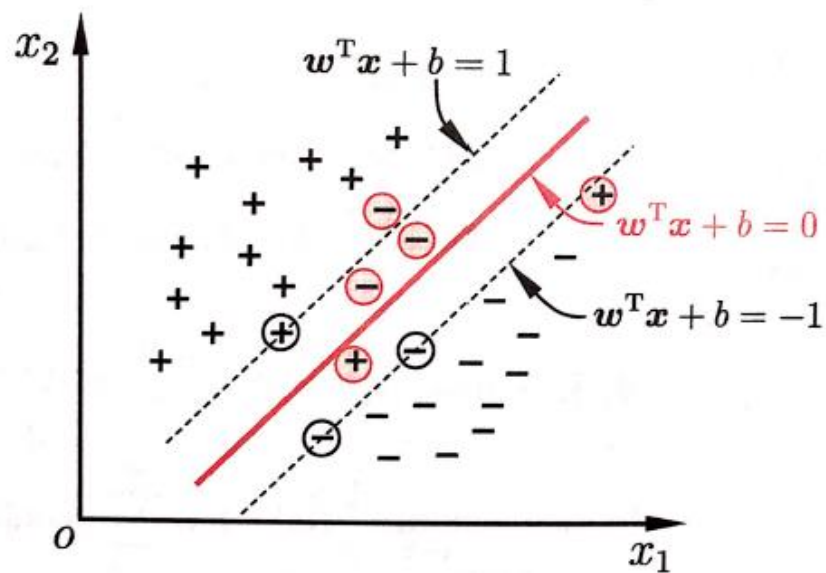
难以直接求解



# 支持向量机：线性不可分-松弛变量，软间隔与hinge损失函数

Hinge损失函数： $\min_{\mathbf{w}, b} \frac{1}{2} \mathbf{w}^T \mathbf{w} + c \sum_{i=1}^n \max(0, 1 - y_i(\mathbf{w}^T \mathbf{x}_i + b))$

正确分类数据的Hinge损失中  $\max(0, 1 - y_i(\mathbf{w}^T \mathbf{x}_i + b)) = 0$



- 记  $\xi_i = \max(0, 1 - y_i(\mathbf{w}^T \mathbf{x}_i + b))$  ( $\xi_i$ 被称为第*i*个变量的“松弛变量”，slack variables)，显然  $\xi_i \geq 0$ 。每一个样本对应一个松弛变量，用来表示该样本被分类错误所产生的损失。于是，可将上式重写为：

$$\min_{\mathbf{w}, b} \frac{1}{2} \mathbf{w}^T \mathbf{w} + c \times \sum_{i=1}^n \xi_i$$

$$\begin{aligned} \text{s.t. } & y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, i = 1, 2, \dots, n \end{aligned}$$

拉格朗日乘子法



# 线性分类模型：正则项

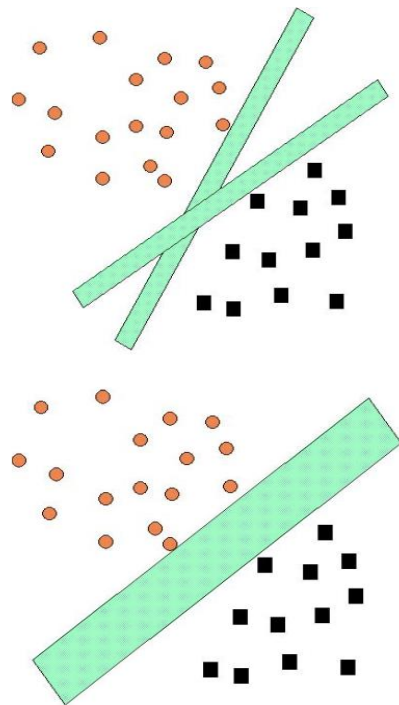
- Hinge损失函数：

$$\min_{\mathbf{w}, b} \frac{1}{2} \mathbf{w}^T \mathbf{w} + c \sum_{i=1}^n \max(0, 1 - y_i(\mathbf{w}^T \mathbf{x}_i + b))$$

- 如果替换成其他损失函数，得到更一般的分类模型形式，

$$\min_f \Omega(f) + c \sum_{i=1}^n l(f(\mathbf{x}_i), y_i)$$

- 其中， $\Omega(f)$ 称为**结构风险（structural risk）**，也被称为正则项，用于描述模型 $f$ 的复杂度等性质。常用的正则化方法包括 $L_p$ 范数，其中 $L_2$ 范数 $\|\mathbf{w}\|_2$ 倾向于 $\mathbf{w}$ 的分量取值较小， $L_1$ 范数 $\|\mathbf{w}\|_1$ 倾向于 $\mathbf{w}$ 的分量尽量稀疏。
- $\sum_{i=1}^n l(f(\mathbf{x}_i), y_i)$ 称为**经验风险（empirical risk）**，用于描述模型 $f$ 与训练数据的契合度。





# 支持向量机模型

- 训练数据:  $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ , 其中  $y_i \in \{-1, +1\}$
- 学习模型:  $f(x_i) = \mathbf{w}^T \mathbf{x}_i + b$
- 损失函数:  $L(\mathbf{w}) = \min_{\mathbf{w}, b} \frac{1}{2} \mathbf{w}^T \mathbf{w} + c \sum_{i=1}^n \max(0, 1 - y_i(\mathbf{w}^T \mathbf{x}_i + b))$
- 优化方法: 二次规划、SMO等



北京航空航天大学  
COLLEGE OF SOFTWARE  
BEIHANG UNIVERSITY 软件学院

# 支持向量机

- 更多阅读和视频资源

- 书籍：机器学习，周志华
- 视频：

[https://www.bilibili.com/video/BV1jt4y1E7BQ/?vd\\_source=5afe770cafa42a833cbfdbba5d750438](https://www.bilibili.com/video/BV1jt4y1E7BQ/?vd_source=5afe770cafa42a833cbfdbba5d750438)

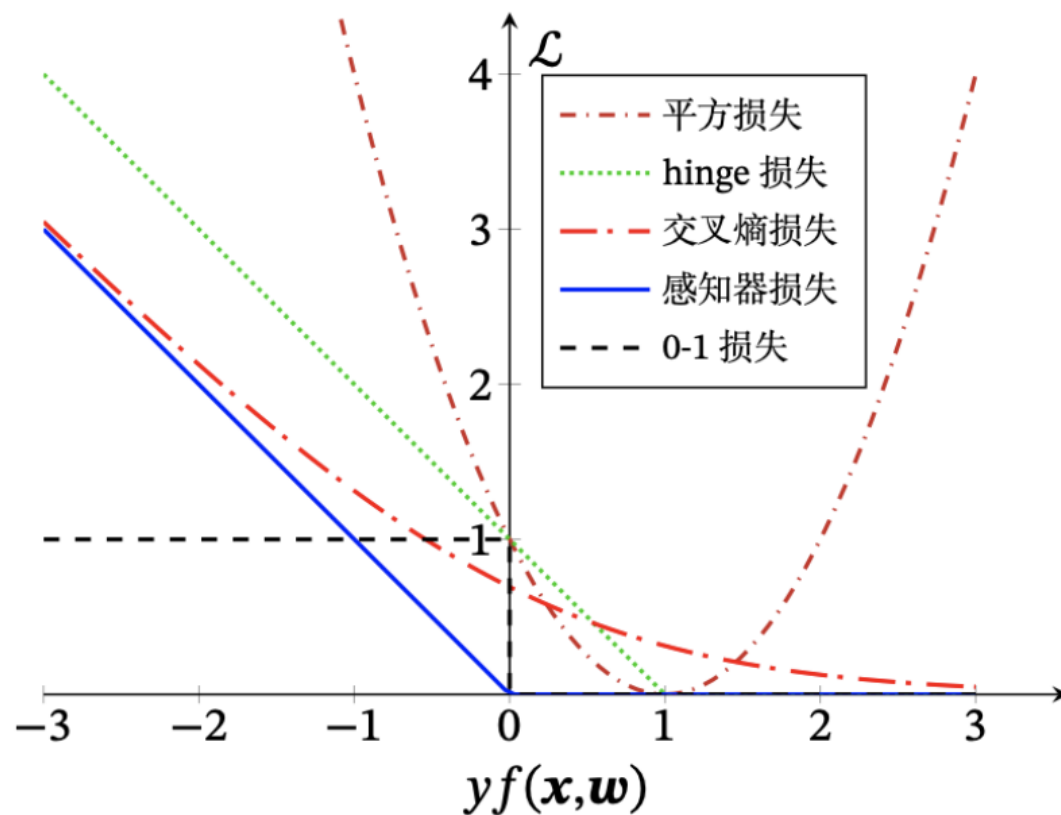
【数之道】支持向量机SVM是什么，八分钟直觉理解其本质-哔哩哔哩  
<https://b23.tv/jzUSnAW>

- 博客：[https://blog.csdn.net/v\\_JULY\\_v/article/details/7624837](https://blog.csdn.net/v_JULY_v/article/details/7624837)



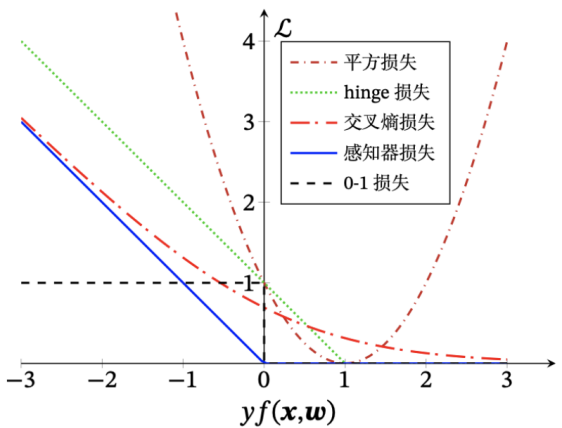
# 总结：线性分类模型

- 线性回归： $\frac{1}{N} \sum (y_i - f(x_i))^2$
- 对数几率回归： $-\log P(y_i | x_i)$
- 线性判别分析
- 感知器
- 支持向量机
- ...



二分类问题中不同损失函数的对比  
(横轴表示 $yf(x, w)$ , 纵轴表示损失)

# 总结：线性分类模型



线性模型	激活函数	损失/目标函数	损失/目标函数定义	优化方法
线性回归	-	平方损失	$(y_i - \mathbf{w}^T \mathbf{x}_i)^2$	最小二乘、梯度下降
对数几率回归	$\text{sigmoid}(\mathbf{w}^T \mathbf{x})$	二值交叉熵损失	$-y_i \log \sigma(\mathbf{w}^T \mathbf{x}_i) + (1 - y_i) \log(1 - \sigma(\mathbf{w}^T \mathbf{x}_i))$	梯度下降
Softmax分类	$\text{softmax}(\mathbf{W}^T \mathbf{x})$	交叉熵损失	$-y_i \log \text{softmax}(\mathbf{w}^T \mathbf{x}_i)$	梯度下降
线性判别分析	-	Fisher准则	$\frac{\ m_2 - m_1\ _2^2}{s_1^2 + s_2^2}$	广义特征值分解
感知器	$\text{sign}(\mathbf{w}^T \mathbf{x})$	0-1损失	$\begin{cases} 0, y \cdot f(\mathbf{x}_i) \geq 0 \\ 1, y \cdot f(\mathbf{x}_i) < 0 \end{cases}$	迭代优化
支持向量机	$\text{sign}(\mathbf{w}^T \mathbf{x})$	Hinge损失	$\max(0, 1 - y_i \mathbf{w}^T \mathbf{x}_i)$	二次规划、SMO等