MACHINE LEARNING

机器学习

Linear Models

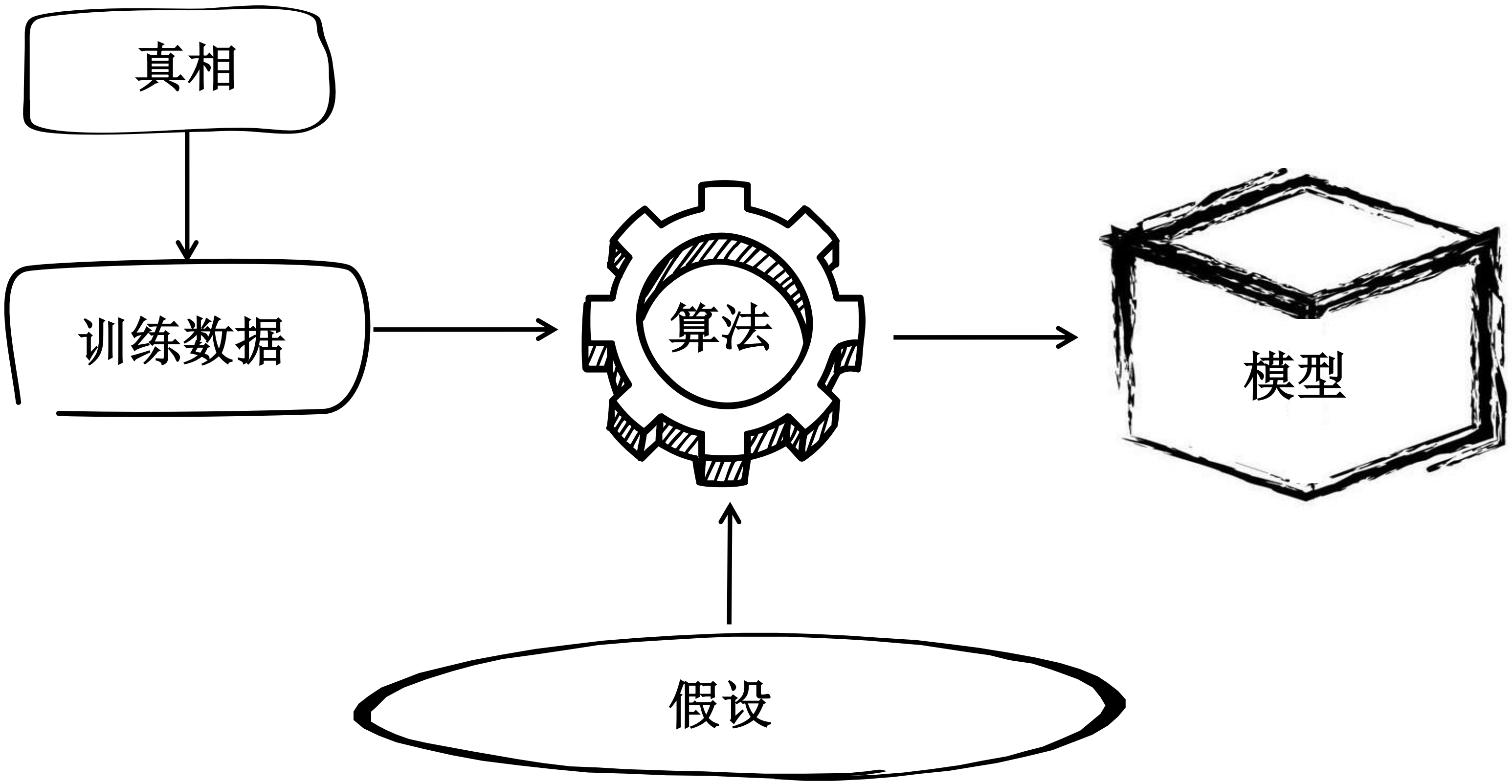
线性模型

参考：
《机器学习》

Machine Learning Course

定义

- 线性假设空间



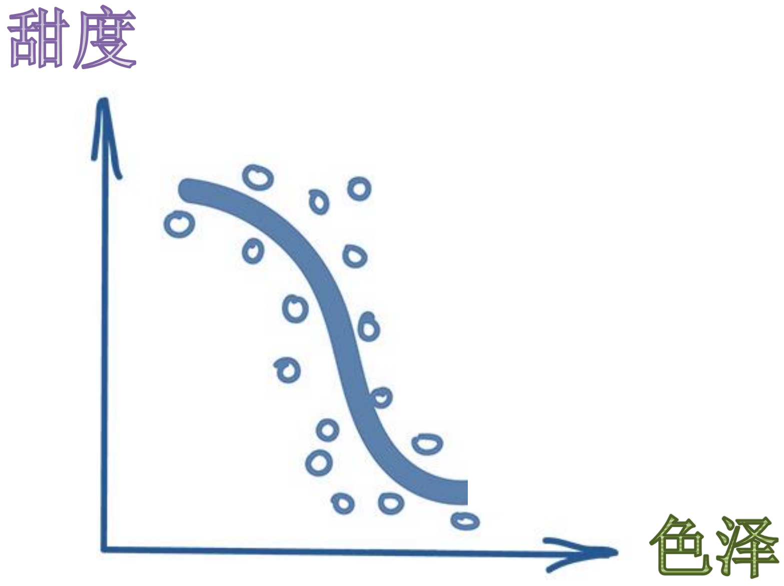
$$f(\boldsymbol{x}) = w_1x_1 + w_2x_2 + \cdots + w_dx_d + b$$

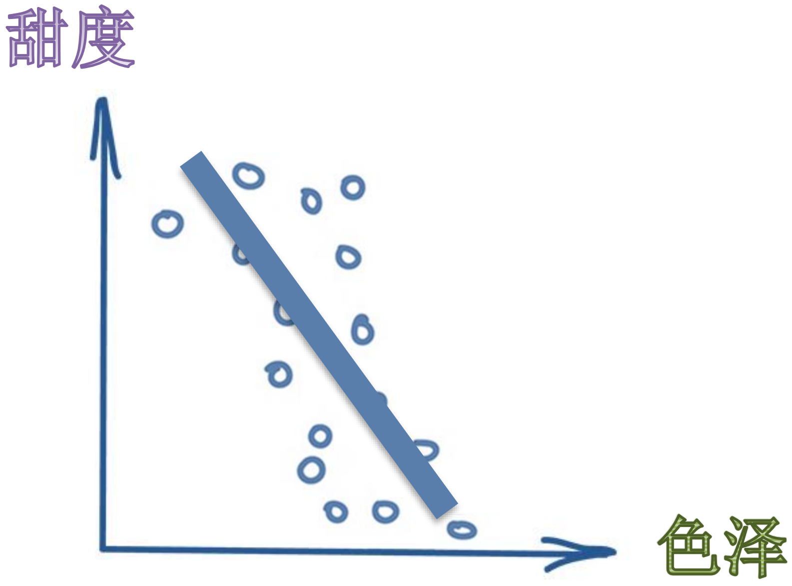
$$f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$$

$$f_{\text{好瓜}}(\boldsymbol{x}) = 0.2 \cdot x_{\text{色泽}} + 0.5 \cdot x_{\text{根蒂}} + 0.3 \cdot x_{\text{敲声}} + 1$$

定义

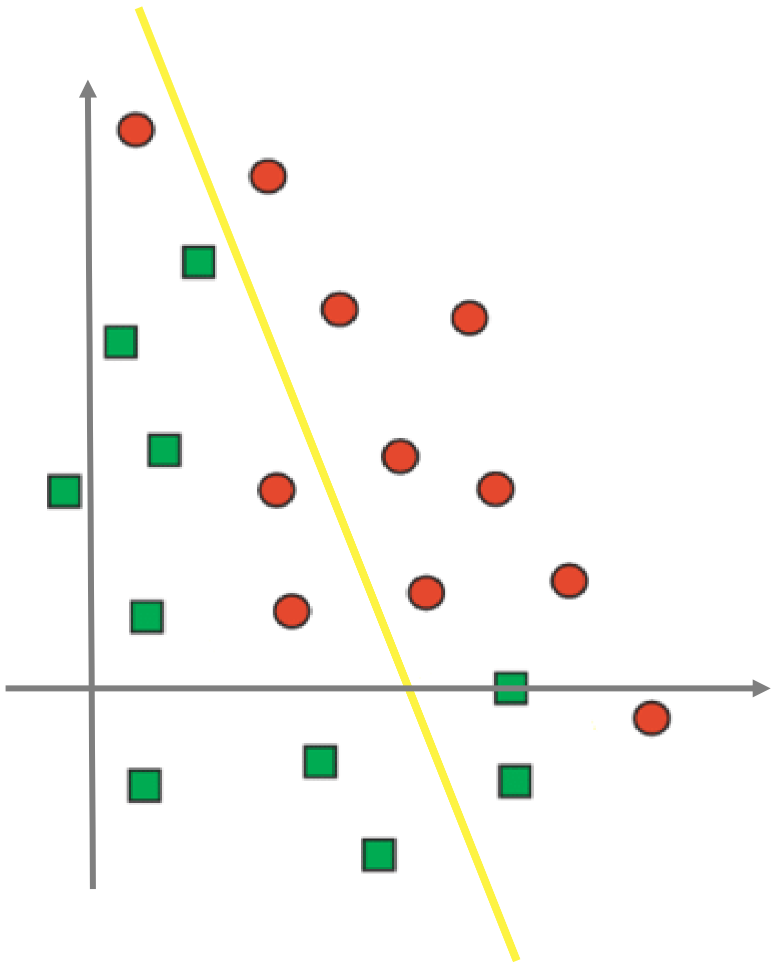
- 线性拟合函数

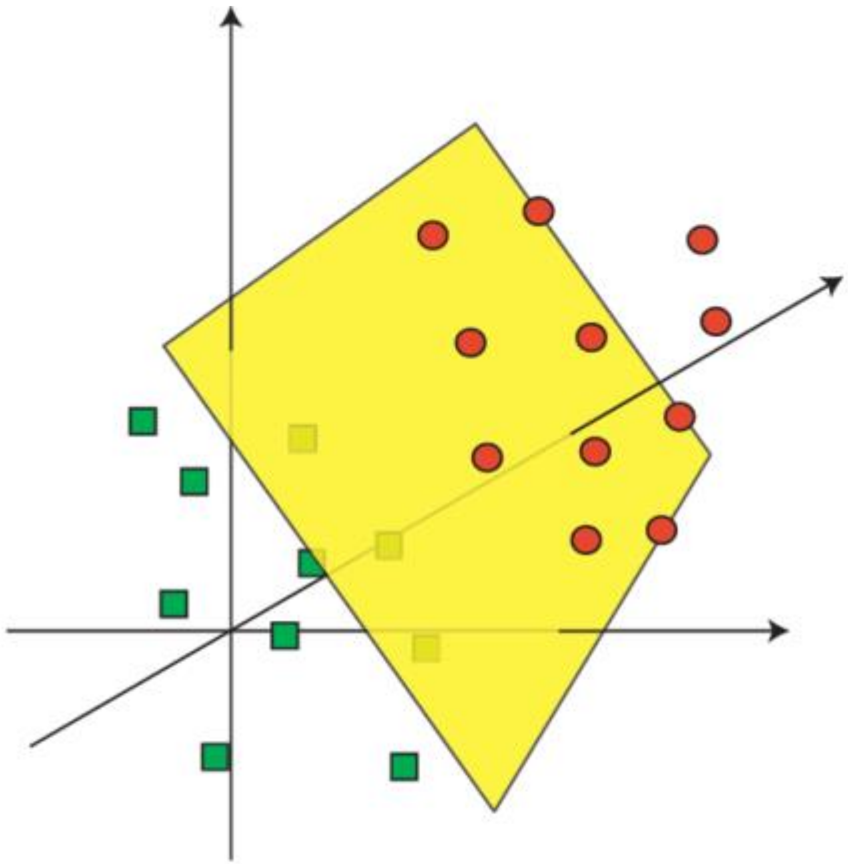




定义

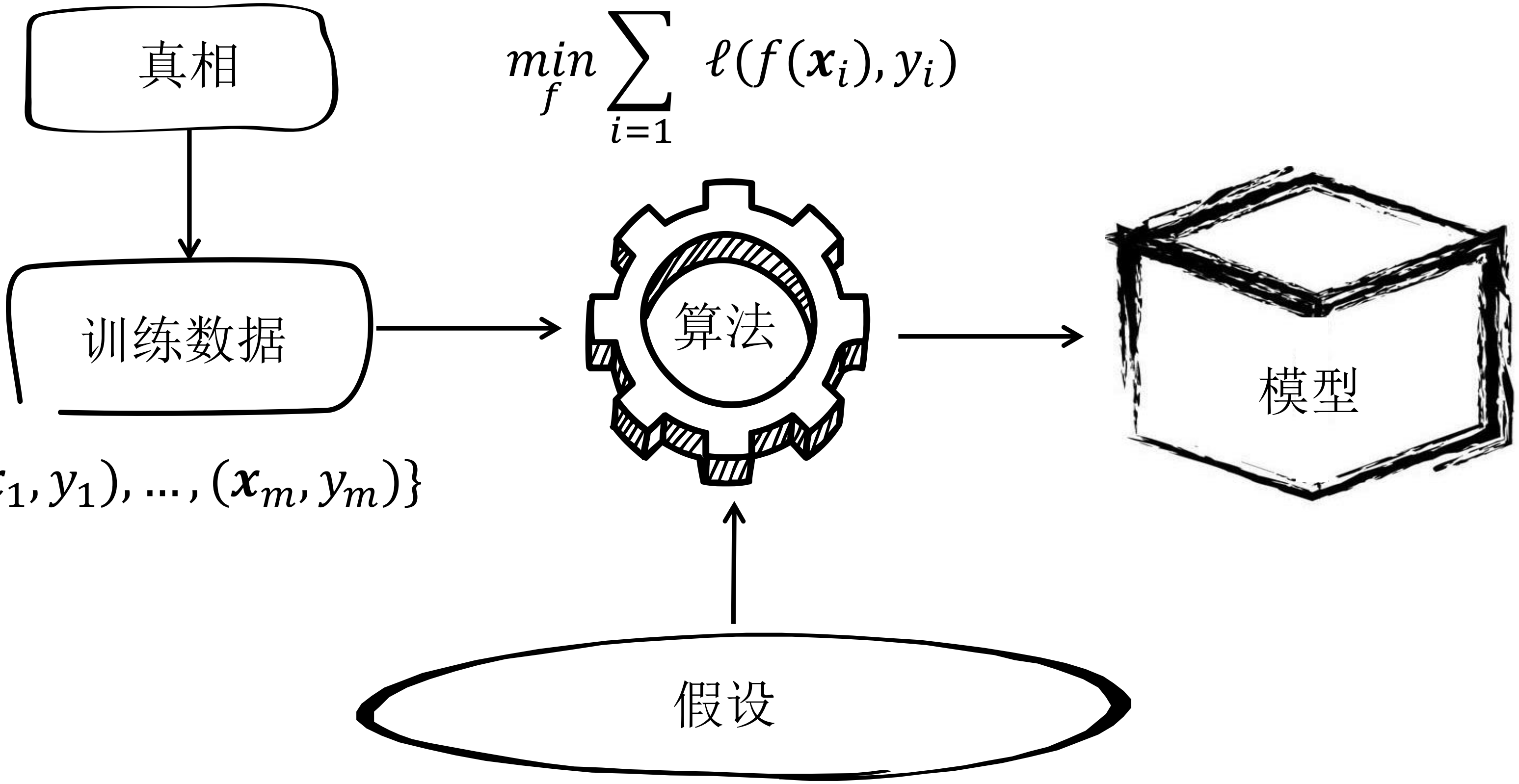
- 线性分类超平面





线性回归

- 最小经验损失



$$\min_f \sum_{i=1}^{m} \ell(f(\boldsymbol{x}_i), y_i)$$

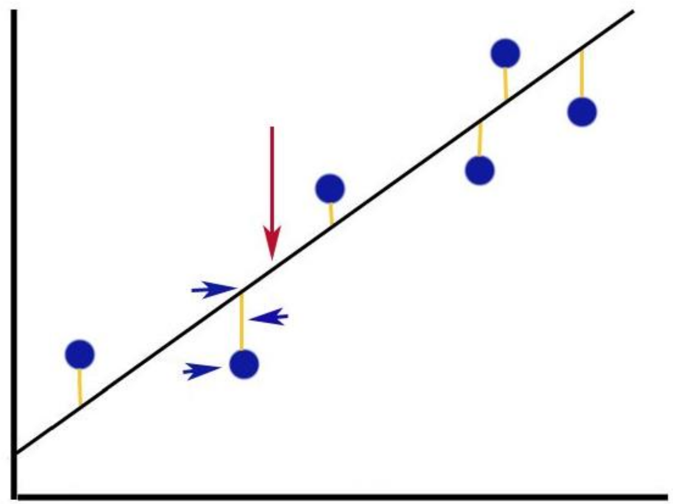
$$\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_m, y_m)\}$$

$$f(\boldsymbol{x}) = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + b$$
$$f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x} + b$$

线性回归

- 损失函数

$$\ell(f(\boldsymbol{x}_i), y_i) = (f(x_i) - y_i)^2$$



- 经验误差

$$\min_f \sum_{i=1}^{m} \ell(f(\boldsymbol{x}_i), y_i) \quad = \arg\min_{(w,b)} \sum_{i=1}^{m} (f(x_i) - y_i)^2$$

$$= \arg\min_{(w,b)} \sum_{i=1}^{m} (y_i - wx_i - b)^2$$

基于均方误差最小化来进行模型求解的方法称为**"最小二乘法"(least square method).**
在线性回归中，最小二乘法就是试图找到一条直线，使所有样本到直线上的欧氏距离之和最小.

线性回归

- 解析解

$$\mathbf{X}=\begin{pmatrix} x_{11} & x_{12} & \dots & x_{1d} & 1 \\ x_{21} & x_{22} & \dots & x_{2d} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{md} & 1 \end{pmatrix}=\begin{pmatrix} \boldsymbol{x}_1^{\mathrm{T}} & 1 \\ \boldsymbol{x}_2^{\mathrm{T}} & 1 \\ \vdots & \vdots \\ \boldsymbol{x}_m^{\mathrm{T}} & 1 \end{pmatrix}$$

$$\boldsymbol{y}=(y_1;y_2;\dots;y_m)$$

$$\widehat{\boldsymbol{w}}^*=\arg\min_{\hat{w}}\,(\boldsymbol{y}-\mathbf{X}\widehat{\boldsymbol{w}})^{\mathrm{T}}(\boldsymbol{y}-\mathbf{X}\widehat{\boldsymbol{w}})$$

$$\frac{\partial E_{\widehat{\boldsymbol{w}}}}{\partial \widehat{\boldsymbol{w}}}=2\mathbf{X}^{\mathrm{T}}(\mathbf{X}\widehat{\boldsymbol{w}}-\boldsymbol{y}) \rightarrow \widehat{\boldsymbol{w}}^*=(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{y}$$

$$f\left(\widehat{\boldsymbol{x}}_i\right)=\widehat{\boldsymbol{x}}_i^{\mathrm{T}}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{y},\quad \widehat{\boldsymbol{x}}_i=(\boldsymbol{x}_i,1)$$

线性回归

- 解析解

$\hat{\boldsymbol{w}}^* = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{y}$，此解析解假设了$(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}$存在，这需要$\mathbf{X}^{\mathrm{T}}\mathbf{X}$是一个满秩矩阵 (full rank matrix)

若$m<d+1$, 则$(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}$便不存在，此时无法用解析解求解最小二乘法

- 梯度下降法

考虑无约束优化问题$\min\limits_{\boldsymbol{x}} f(\boldsymbol{x})$

若能构造一 个序列$\boldsymbol{x}^0, \boldsymbol{x}^1, \boldsymbol{x}^2, \ldots$

满足$f(\boldsymbol{x}^{t+1}) < f(\boldsymbol{x}^t), t=0,1,2, \ldots$

则不断执行该过程即可收剑到局部极小点

根据泰勒展式有$f(\boldsymbol{x}+\Delta\boldsymbol{x}) \simeq f(\boldsymbol{x}) + \Delta\boldsymbol{x}^{\mathrm{T}}\nabla f(\boldsymbol{x})$

于是, 欲满足$f(\boldsymbol{x}+\Delta\boldsymbol{x}) < f(\boldsymbol{x})$

可选择$\Delta\boldsymbol{x} = -\eta\nabla f(\boldsymbol{x})$

其中步长η是一个小常数. 这就是梯度下降法

- 随机梯度下降法求解最小二乘法

$$w^{(t+1)} = w^{(t)} - \eta\nabla\mathcal{L}_i$$
$$w^{(t+1)} = w^{(t)} + \eta(y_i - w^{(t)T}x_i)x_i$$

线性回归

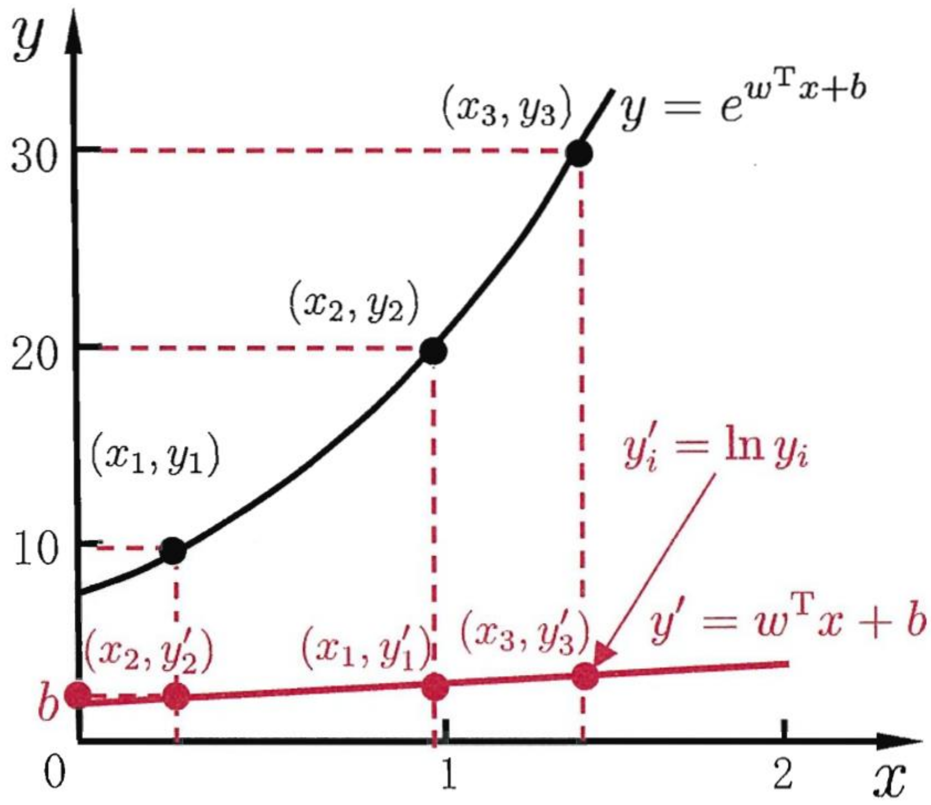
- 对数线性回归

$$\ln y = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$$

- 广义线性模型

$$y = \underset{\uparrow}{g}^{-1}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b)$$

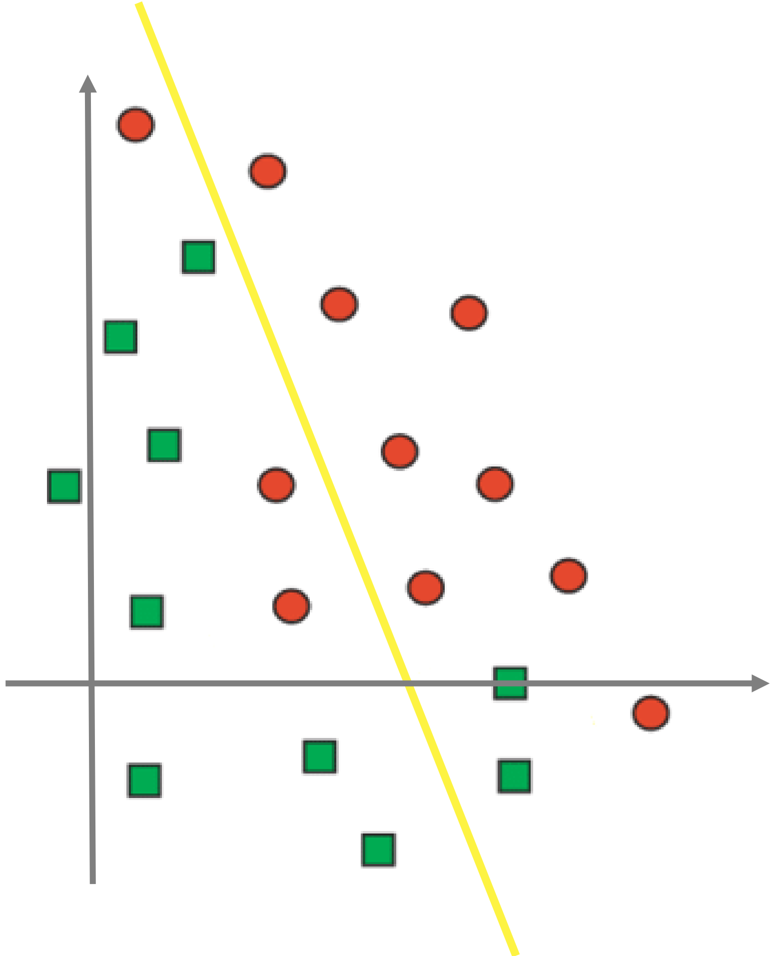
“联系函数”(link function)

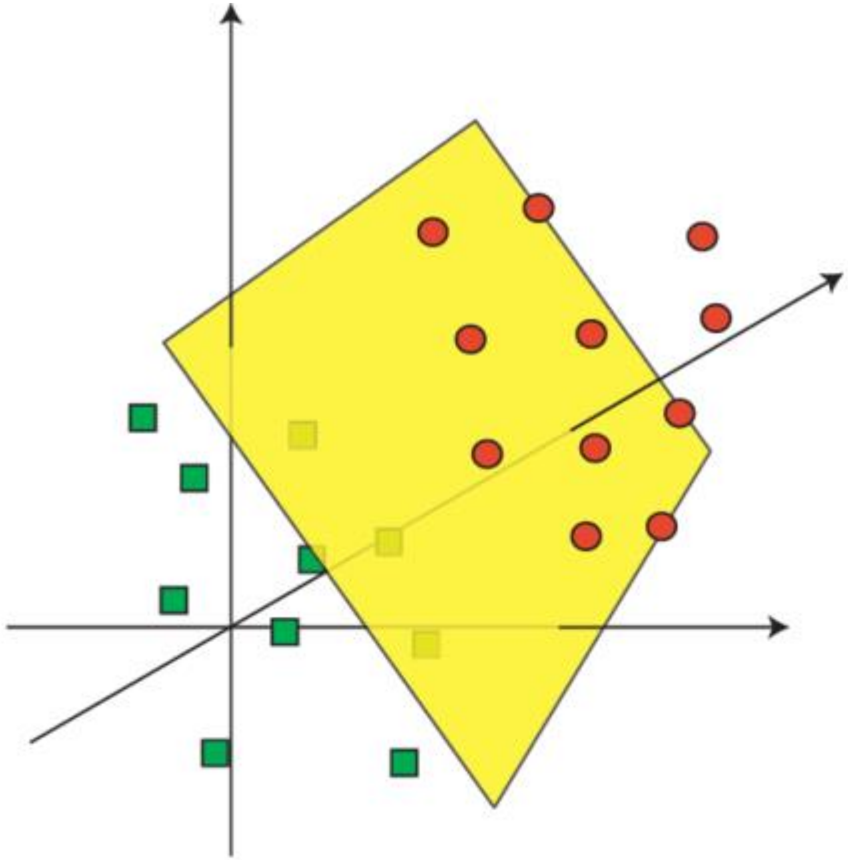


线性分类

- 困难

$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b \xrightarrow{?} y \in \{0,1\}$$



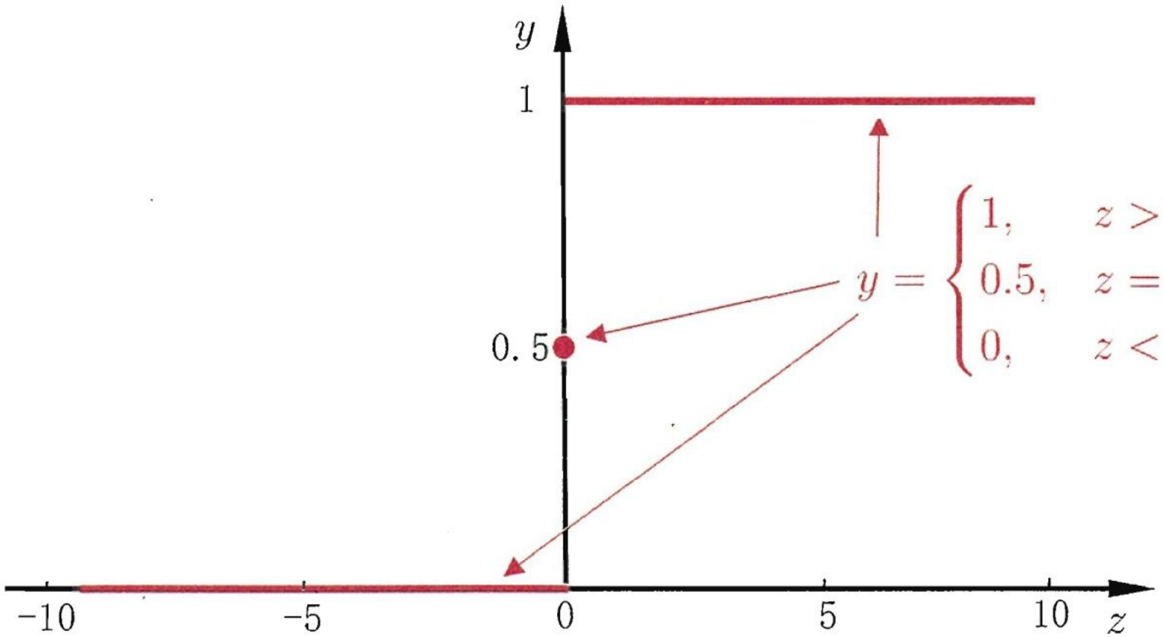


线性分类

- 阶跃函数来联系

$$z = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$$

$$y = \begin{cases} 0, & z < 0 \\ 0.5, & z = 0 \\ 1, & z > 0 \end{cases}$$

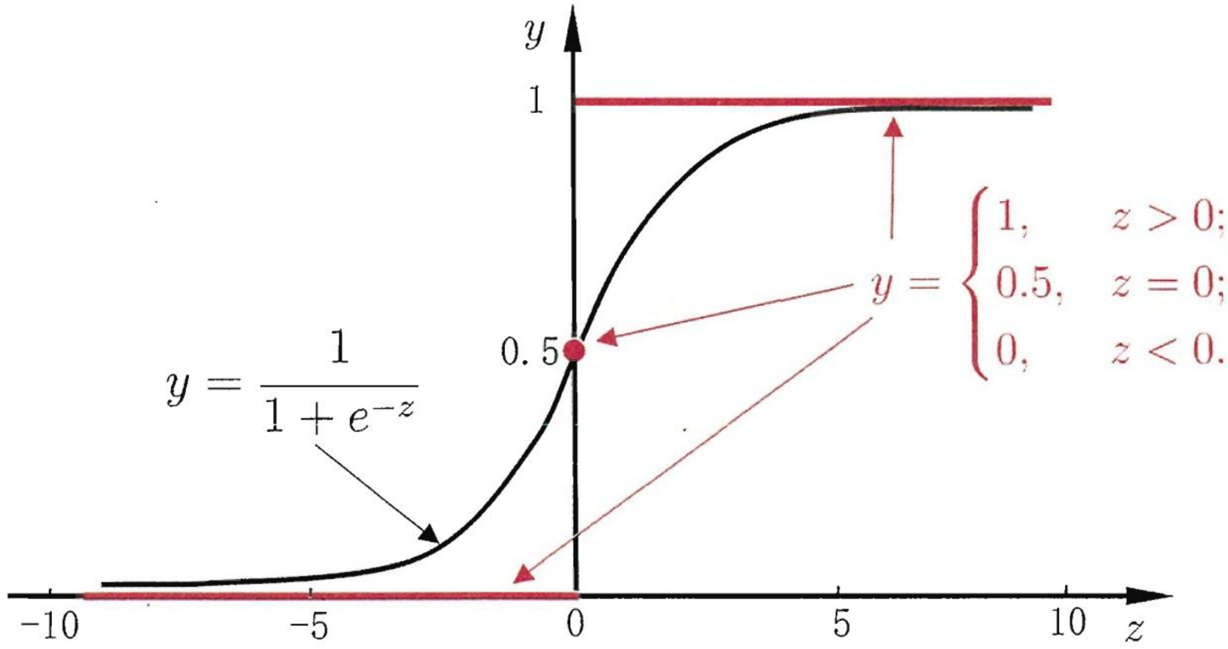


对数几率回归

- Sigmoid函数来联系

$z = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$

$$y = \frac{1}{1 + e^{-z}}$$



对数几率回归

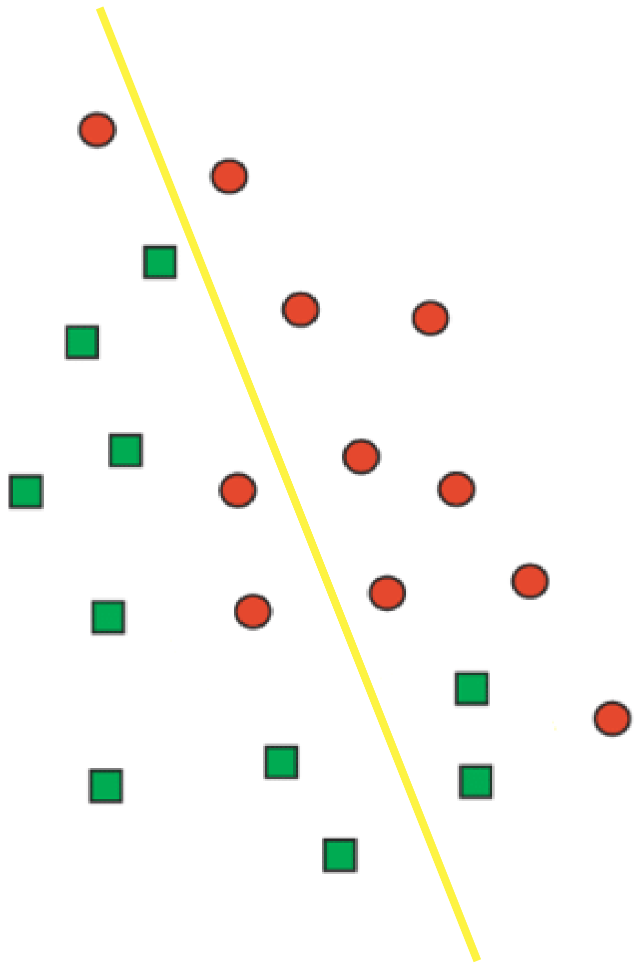
- 回归？分类？

$$y = \frac{1}{1+e^{-(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b)}} = p(y=1 \mid \boldsymbol{x})$$

$$\ln \frac{y}{1-y} = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b$$

$$\underbrace{\ln \underbrace{\frac{p(y=1 \mid \boldsymbol{x})}{p(y=0 \mid \boldsymbol{x})}}_{\text{几率}}}_{\text{对数几率}}$$

回归



对数几率回归

- 最大似然法求解

$$\ell(\boldsymbol{w}, b) = \sum_{i=1}^{m} \ln p(y_i \mid \boldsymbol{x}_i; \boldsymbol{w}, b)$$

$$p(y_i \mid \boldsymbol{x}_i; \boldsymbol{w}, b) = y_i p_1(\hat{\boldsymbol{x}}_i; \boldsymbol{\beta}) + (1 - y_i) p_0(\hat{\boldsymbol{x}}_i; \boldsymbol{\beta})$$

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{m} -\ln p(y_i \mid \boldsymbol{x}_i; \boldsymbol{w}, b) = \sum_{i=1}^{m} -\ln \left(y_i p_1\left(\hat{\boldsymbol{x}}_i; \boldsymbol{\beta}\right) + (1 - y_i) p_0\left(\hat{\boldsymbol{x}}_i; \boldsymbol{\beta}\right)\right)$$

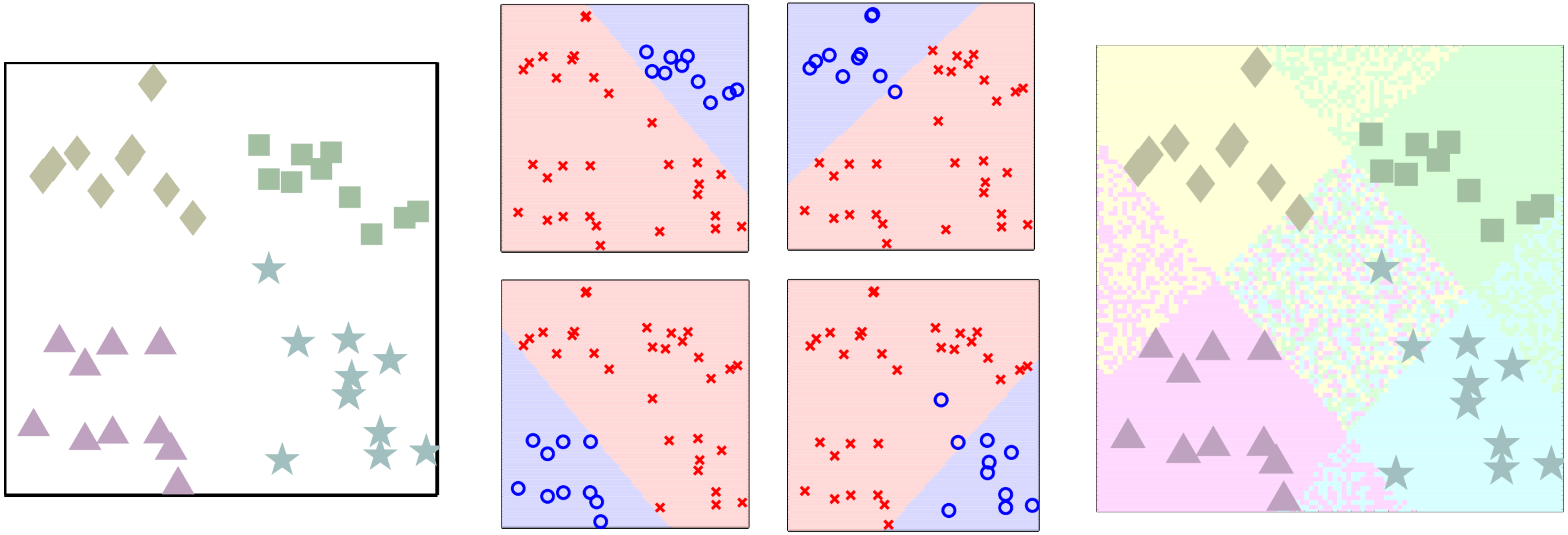
$$= \sum_{i=1}^{m} \left(-y_i \boldsymbol{\beta}^{\mathrm{T}} \hat{\boldsymbol{x}}_i + \ln\left(1 + e^{\boldsymbol{\beta}^{\mathrm{T}} \hat{\boldsymbol{x}}_i}\right)\right)$$

$$\boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\beta}} \ell(\boldsymbol{\beta})$$

$$\frac{\partial \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = -\sum_{i=1}^{m} \hat{\boldsymbol{x}}_i \left(y_i - p_1(\hat{\boldsymbol{x}}_i; \boldsymbol{\beta})\right) \qquad \frac{\partial^2 \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^{\mathrm{T}}} = \sum_{i=1}^{m} \hat{\boldsymbol{x}}_i \hat{\boldsymbol{x}}_i^{\mathrm{T}} p_1(\hat{\boldsymbol{x}}_i; \boldsymbol{\beta})\left(1 - p_1(\hat{\boldsymbol{x}}_i; \boldsymbol{\beta})\right)$$

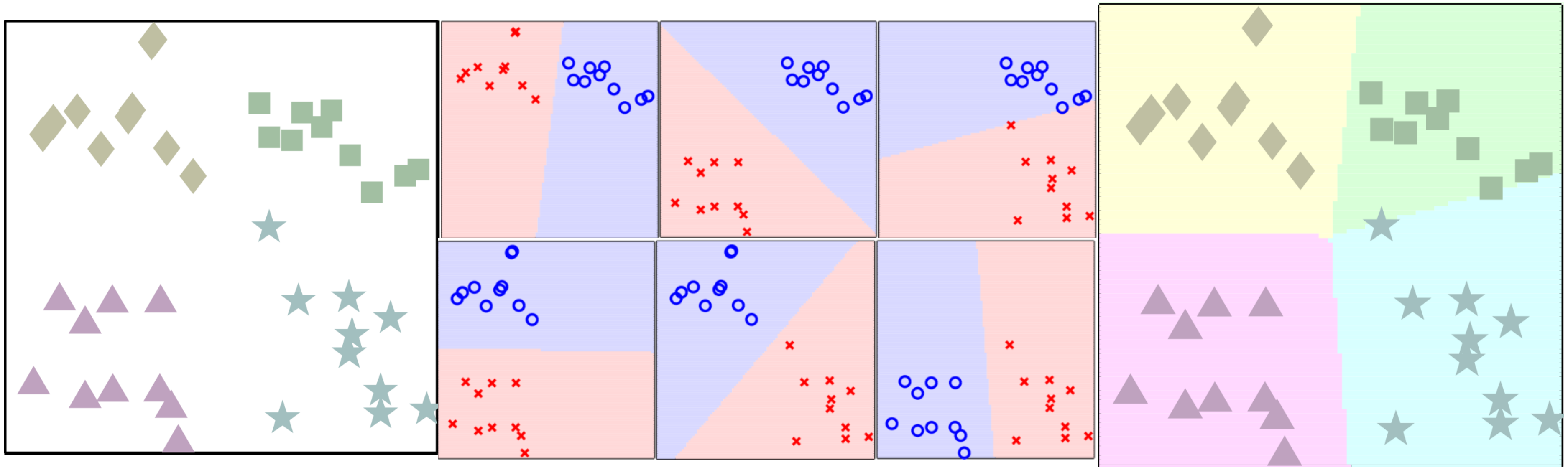
多分类

- 一对其余 One vs. Rest



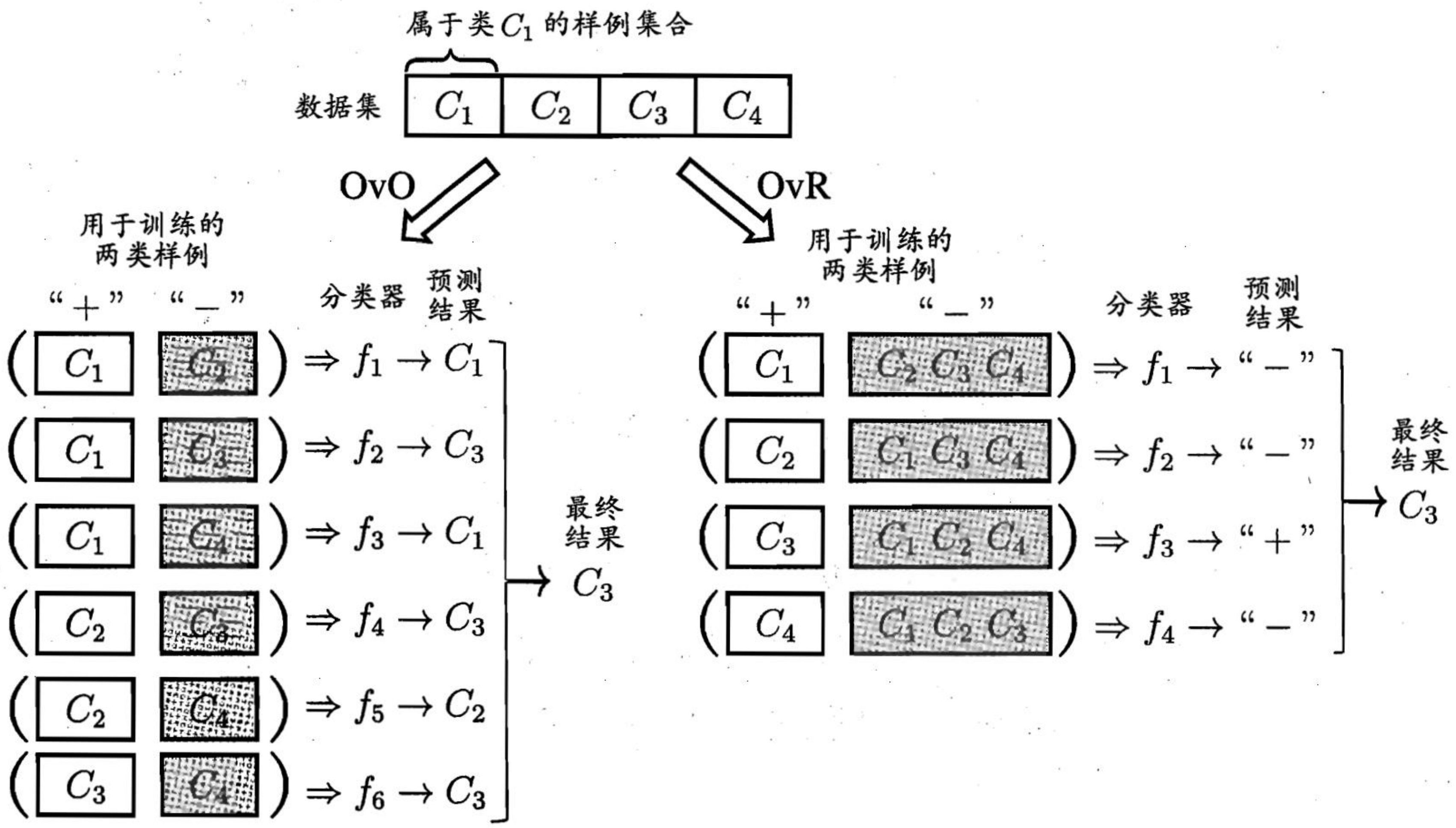
多分类

- 一对一 One vs. One



多分类

- 对比


属于类 C_1 的样例集合
数据集
C_1 C_2 C_3 C_4
OvO
OvR
用于训练的两类样例
"+" "−"
分类器
预测结果
$(C_1\ C_2) \Rightarrow f_1 \rightarrow C_1$
$(C_1\ C_3) \Rightarrow f_2 \rightarrow C_3$
$(C_1\ C_4) \Rightarrow f_3 \rightarrow C_1$
$(C_2\ C_3) \Rightarrow f_4 \rightarrow C_3$
$(C_2\ C_4) \Rightarrow f_5 \rightarrow C_2$
$(C_3\ C_4) \Rightarrow f_6 \rightarrow C_3$
最终结果
C_3
用于训练的两类样例
"+" "−"
分类器
预测结果
$(C_1\ C_2\ C_3\ C_4) \Rightarrow f_1 \rightarrow$ "−"
$(C_2\ C_1\ C_3\ C_4) \Rightarrow f_2 \rightarrow$ "−"
$(C_3\ C_1\ C_2\ C_4) \Rightarrow f_3 \rightarrow$ "+"
$(C_4\ C_1\ C_2\ C_3) \Rightarrow f_4 \rightarrow$ "−"
最终结果
C_3

类别不平衡

- 定义

类别不平衡 (class-imbalance) 就是指分类任务中不同类别的训练样例数目差别很大的情况.

- 策略

- 欠采样

- 过采样

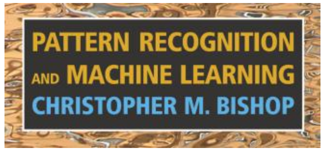
- 阈值移动

MACHINE LEARNING

机器学习

Linear Models

扩展



参考：
《PRML》

Machine Learning Course

Softmax回归

- Softmax函数

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \mathrm{softmax}\left(\begin{array}{c} W_{1,1}x_1 + W_{1,2}x_1 + W_{1,3}x_1 + b_1 \\ W_{2,1}x_2 + W_{2,2}x_2 + W_{2,3}x_2 + b_2 \\ W_{3,1}x_3 + W_{3,2}x_3 + W_{3,3}x_3 + b_3 \end{array}\right) \begin{array}{l} \longleftrightarrow \mathbf{w}_1^\top \mathbf{x} \\ \longleftrightarrow \mathbf{w}_2^\top \mathbf{x} \\ \longleftrightarrow \mathbf{w}_3^\top \mathbf{x} \end{array}$$

$$p(y = c \mid \mathbf{x}) = \mathrm{softmax}(\mathbf{w}_c^\top \mathbf{x}) = \frac{\exp(\mathbf{w}_c^\top \mathbf{x})}{\sum_{c=1}^{C} \exp(\mathbf{w}_c^\top \mathbf{x})}$$

Softmax回归

- 交叉熵损失函数

$$\mathcal{L}(\mathbf{y}, f(\mathbf{x},\theta)) = -\sum_{c=1}^{C} y_c \log f_c(\mathbf{x},\theta) = -\log f_y(x,\theta)$$

y_c是一个C维的向量，用来标注样本的多类标签。假设样本的标签为c，那么它只有c维是1，其余维都为0

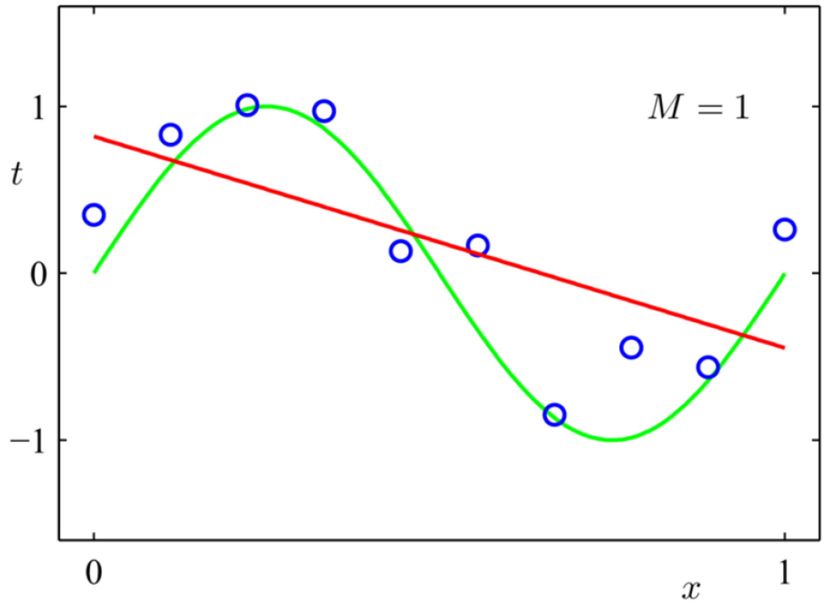
$f_c(\mathbf{x},\theta)$为模型（参数为θ）判断的样本为第c类的概率

交叉熵角度：当我们将$\mathbf{y}$看做是样本标签的真实概率分布，$f(\mathbf{x},\theta)$看作是类别标签的条件概率分布（模型估计的），那么$\mathcal{L}(\mathbf{y}, f(\mathbf{x},\theta))$就对应于信息论里面交叉熵的概念，是一种衡量两个分布差距的度量。

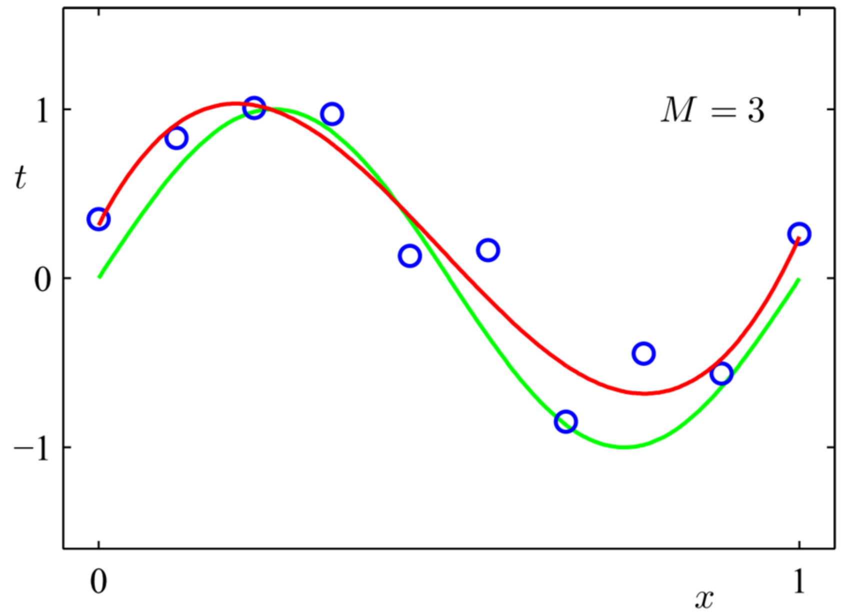
最大似然估计角度：$\log f_y(x,\theta)$实际上对应于真实类别y的对数似然函数

线性基函数模型 *Linear Basis Function Models*

- 示例



$$y(x,\mathbf{w}) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + \cdots + w_M x^M$$



$$y(x,\mathbf{w}) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + \cdots + w_M x^M$$

线性基函数模型 *Linear Basis Function Models*

- 定义

$$y(\boldsymbol{x}, \boldsymbol{w}) = \sum_{j=0}^{M} w_j \phi_j(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}), \quad \phi_0(\boldsymbol{x}) = 1$$

$$\phi_j(x) = x^j$$

$$\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\}$$

$$\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right) \qquad \sigma_a = \frac{1}{1+\exp(-a)}$$

线性基函数模型 *Linear Basis Function Models*

- 定义

$$y(\boldsymbol{x}, \boldsymbol{w}) = \sum_{j=0}^{M} w_j \phi_j(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}), \quad \phi_0(\boldsymbol{x}) = 1$$

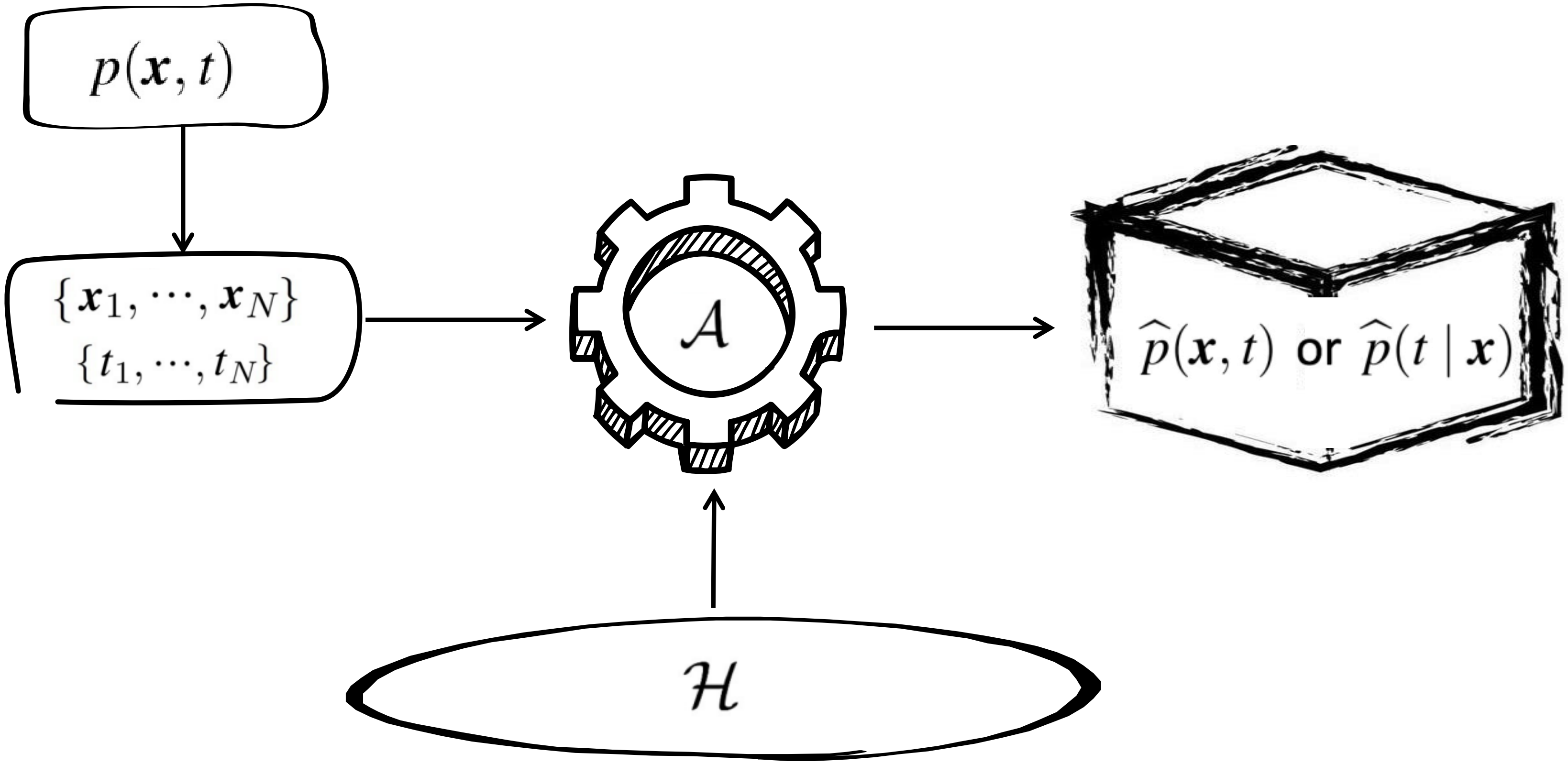
$$\phi_j(x) = x^j$$

$$\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\}$$

$$\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right) \qquad \sigma_a = \frac{1}{1+\exp(-a)}$$

最大似然法与最小二乘法

- 回归任务的概率体系框架



最大似然法与最小二乘法

- 关于回归任务的决策理论

$$L(t, y(\boldsymbol{x})) = \{y(\boldsymbol{x}) - t\}^2$$

$$\mathbb{E}[L] = \iint \{y(\boldsymbol{x}) - t\}^2 p(\boldsymbol{x}, t) \mathrm{d}\boldsymbol{x} \mathrm{d}t$$

$$\frac{\partial \mathbb{E}[L]}{\partial y(\boldsymbol{x})} = 2 \int \{y(\boldsymbol{x}) - t\} p(\boldsymbol{x}, t) \mathrm{d}t = 0$$

$$y(\boldsymbol{x}) = \frac{\int t p(\boldsymbol{x}, t) \mathrm{d}t}{p(\boldsymbol{x})} = \int t p(t \mid \boldsymbol{x}) \, \mathrm{d}t = \mathbb{E}_t[t \mid \boldsymbol{x}]$$

最大似然法与最小二乘法

- 最小二乘法与最大似然估计

$p(t \mid \boldsymbol{x})$?

$$p(t \mid \boldsymbol{x}, \boldsymbol{w}, \beta) = \mathcal{N}\left(t \mid y(\boldsymbol{x}, \boldsymbol{w}), \beta^{-1}\right) \qquad t = y(\boldsymbol{x}, \boldsymbol{w}) + \epsilon \quad \epsilon \sim \mathcal{N}\left(0, \beta^{-1}\right)$$

$$\boldsymbol{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\} \quad \mathbf{t} = \{t_1, \cdots, t_N\}$$

$$p(\mathbf{t} \mid \boldsymbol{X}, \boldsymbol{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}\left(t_n \mid \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n), \beta^{-1}\right)$$

$$\ln p(\mathbf{t} \mid \boldsymbol{w}, \beta) = \sum_{n=1}^{N} \ln \mathcal{N}\left(t_n \mid \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n), \beta^{-1}\right)$$

$$= \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) - \beta E_D(\boldsymbol{w})$$

$$E_D(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^{N} \left\{t_n - \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_n)\right\}^2$$

$$\boldsymbol{w}_{MLE}^{\star} = \arg\max_{w} \ln p(\mathbf{t} \mid \boldsymbol{w}, \beta)$$

$$\boldsymbol{w}_{MLE}^{*} = \boldsymbol{w}_{LS}^{*}$$