2020.2.18 - 2020.6.2 9:50-12:15@6C102雨课堂+腾讯会议
《模式识别与机器学习》

# 第二章
# 线性学习机器
# （2）

2020.2.18

Xuegong Zhang 1

# 2.4 最小平方误差分类器
## Minimum Squared Error

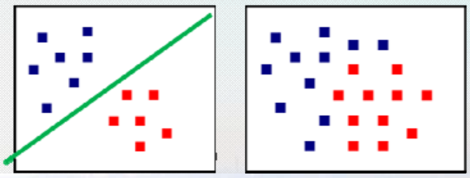
Xuegong Zhang 2

# 回顾

**讨论**

- **感知器有什么问题？**
  - **样本线性不可分呢？**
  - **线性可分时多解？**
  - **多类呢？**
- **有问题怎么办？**
  - **容忍错误，使错误尽量小**
    - **比如强制收敛、MSE**
  - **寻求非线性方法**
    - **比如神经网络、SVM**
  - **寻求“最优分类器”**
    - **比如支持向量机SVM**
  - **多类分类方法**
  - **用两类分类器完成多类分类**

$$a(k+1) = a(k) + \rho_k \sum_{y_j \in Y^k} y_j$$

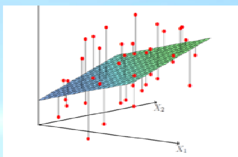




# MSE方法的思想

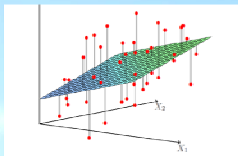
- 对线性分类器，怎样最小化其错误？
- 线性回归：求$\boldsymbol{w}$使$y_i = \boldsymbol{w}^T \boldsymbol{x}_i, i = 1, \cdots, N$
  - 不可能全部样本正好都满足，于是最小化平方误差，方法是：

$$\min E = \frac{1}{N} \sum_{j=1}^{N} \left( f(\boldsymbol{x}_j) - y_j \right)^2$$



## MSE方法的思想

- 对线性分类器，怎样最小化其错误？



- 线性回归：求$\boldsymbol{w}$使$y_i = \boldsymbol{w}^T \boldsymbol{x}_i, i = 1, \cdots, N$
  - 不可能全部样本正好都满足，于是最小化平方误差，方法是：

$$\min E = \frac{1}{N} \sum_{j=1}^{N} \left(f(\boldsymbol{x}_j) - y_j\right)^2$$

- 线性分类器：求$\boldsymbol{\alpha}$使$\boldsymbol{\alpha}^T \boldsymbol{y}_i > 0, i = 1, \cdots, N$
  - 当样本集线性不可分时，使尽可能多的样本满足不等式

注意：这里切换回感知器时的规范化增广向量形式

- 考虑：为每个样本引入$b_i$, 并令

$$\boldsymbol{\alpha}^T \boldsymbol{y}_i = b_i > 0, \quad j = 1, \cdots, N$$

不等式组 ➔ 等式组，可用最小二乘求解



## MSE分类器准则

$$\boldsymbol{\alpha}^T \mathbf{y}_i > 0, \ i = 1, \cdots, N \quad \Longleftrightarrow \quad \boldsymbol{\alpha}^T \mathbf{y}_i = b_i > 0, \ i = 1, \cdots, N$$

- 不等式转化成等式

$$\mathbf{Y}\boldsymbol{\alpha} = \boldsymbol{b}, \ \boldsymbol{b} = [b_1, b_2, \cdots, b_N]^T$$

- MSE准则 $\boldsymbol{\alpha}^*$: $\min\limits_{\boldsymbol{\alpha}} J_S(\boldsymbol{\alpha})$

$$J_S(\boldsymbol{\alpha}) = \|\mathbf{Y}\boldsymbol{\alpha} - \mathbf{b}\|^2 = \sum_{i=1}^{N} (\boldsymbol{\alpha}^T \mathbf{y}_i - b_i)^2$$

## MSE分类器准则

$$\boldsymbol{\alpha}^T\mathbf{y}_i > 0,\ i = 1,\cdots,N \quad \Longleftrightarrow \quad \boldsymbol{\alpha}^T\mathbf{y}_i = b_i > 0,\ i = 1,\cdots,N$$

- 不等式转化成等式 $\quad \mathbf{Y}\boldsymbol{\alpha} = \boldsymbol{b},\ \boldsymbol{b} = [b_1, b_2, \cdots, b_N]^T$
- MSE准则 $\quad \boldsymbol{\alpha}^*:\ \min_{\boldsymbol{\alpha}} J_S(\boldsymbol{\alpha})$

$$J_S(\boldsymbol{\alpha}) = \|\mathbf{Y}\boldsymbol{\alpha} - \mathbf{b}\|^2 = \sum_{i=1}^{N}(\boldsymbol{\alpha}^T\mathbf{y}_i - b_i)^2$$

- 解法：
  - 最小二乘伪逆解 $\quad \boldsymbol{\alpha}^* = (\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{b} = \mathbf{Y}^+\mathbf{b},\quad \mathbf{Y}^+ = (\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T$
  - 梯度下降学习

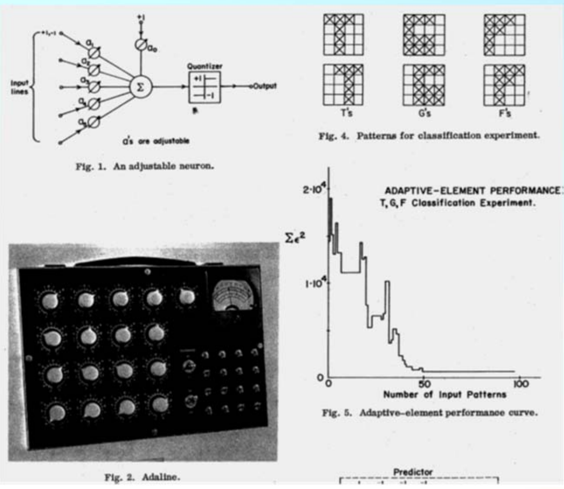
$$\nabla J_S(\boldsymbol{\alpha}) = 2\mathbf{Y}^T(\mathbf{Y}\boldsymbol{\alpha} - \mathbf{b})$$
$$\boldsymbol{\alpha}(k+1) = \boldsymbol{\alpha}(k) + \rho_k(b_k - \boldsymbol{\alpha}(k)^T\mathbf{y}^k)\mathbf{y}^k$$

--- Widrow-Hoff算法，ADALINE

Xuegong Zhang 7

## ADALINE

Widrow & Hoff, Adaptive switching circuits, *1960 IRE Western Electric Show and Convention Record, Part 4*, pp.96-104, Aug, 1960




Widrow & Hoff, Adaptive switching circuits, *1960 IRE Western Electric Show and Convention Record, Part 4*, pp.96-104, Aug, 1960

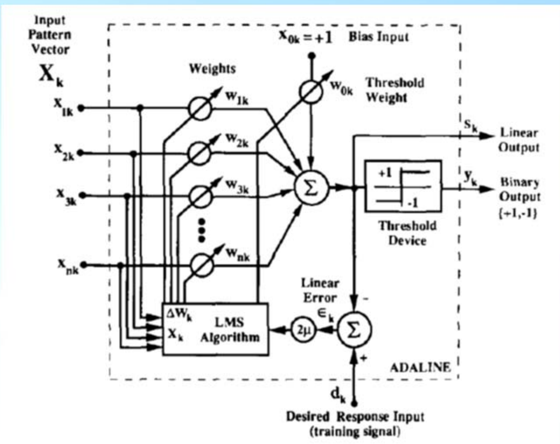



Fig. 2. Adaptive linear element (Adaline).

Widrow & Lehr, 30 years of adaptive neural networks: Perceptron, Madaline, and Backpropagation, *Proceedings of the IEEE*, 78(9): 1415-1442, 1990

Xuegong Zhang 8

# 有没有问题？

请在雨课堂中弹幕提问



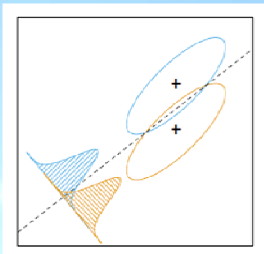
# 有没有问题？

如何给定 $\boldsymbol{b} = [b_1, b_2, \cdots, b_N]^T$ ？

- 可以证明，如果$\boldsymbol{b}$选为

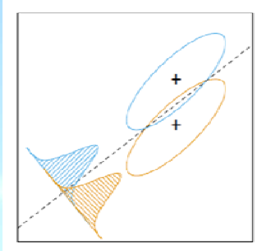
$$b_i = \begin{cases} N/N_1, & if\ y_i \epsilon \omega_1 \\ N/N_2, & if\ y_i \epsilon \omega_2 \end{cases},$$

则MSE解等价于$w_0 = -\hat{m}$的FLD解。



# 有没有问题？

如何给定 $\boldsymbol{b} = [b_1, b_2, \cdots, b_N]^T$ ？



- 可以证明，如果$\boldsymbol{b}$选为

$$b_i = \begin{cases} N/N_1, & if\ y_i \epsilon \omega_1 \\ N/N_2, & if\ y_i \epsilon \omega_2 \end{cases},$$

则MSE解等价于$w_0 = -\hat{m}$的FLD解。

- 如果$\boldsymbol{b}$选为：

$$b_i = 1,\ i = 1, \cdots, N\ ,$$

则当$N \to \infty$时，MSE解以最小均方误差最优逼近贝叶斯判别函数

$$g_0(\boldsymbol{x}) = P(\omega_1|\boldsymbol{x}) - P(\omega_2|\boldsymbol{x})$$

即$\boldsymbol{\alpha}_{\mathrm{MSE}}$使$e^2 = \int [\boldsymbol{\alpha}^T \boldsymbol{y} - g_0(\boldsymbol{x})]^2 p(\boldsymbol{x}) d(\boldsymbol{x})$最小。

Xuegong Zhang 11

# 休息1分钟

Xuegong Zhang 12

# 2.5 罗杰斯特回归
## Logistic Regression

注：单词logistic与logic没有关系，不应译作“逻辑回归”，
应音译作“罗杰斯特回归”或“罗杰斯蒂回归”



# 简单线性回归

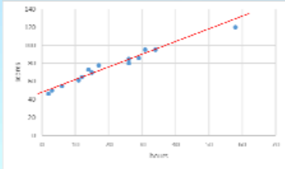
| Age | SBP | Age | SBP | Age | SBP |
|---|---|---|---|---|---|
| 22 | 131 | 41 | 139 | 52 | 128 |
| 23 | 128 | 41 | 171 | 54 | 105 |
| 24 | 116 | 46 | 137 | 56 | 145 |
| 27 | 106 | 47 | 111 | 57 | 141 |
| 28 | 114 | 48 | 115 | 58 | 153 |
| 29 | 123 | 49 | 133 | 59 | 157 |
| 30 | 117 | 49 | 128 | 63 | 155 |
| 32 | 122 | 50 | 183 | 67 | 176 |
| 33 | 99 | 51 | 130 | 71 | 172 |
| 35 | 121 | 51 | 133 | 77 | 178 |
| 40 | 147 | 51 | 144 | 81 | 217 |

Table 1 Age and systolic blood pressure (SBP) among 33 adult women

$$\mathrm{SBP} = 81.54 + 1.222 \cdot \mathrm{Age}$$

# 回归用来预测二值结果？

| Student id | Final score | Study Hours per Week |
|---|---|---|
| 1 | 0 | 3 |
| 2 | 1 | 34 |
| 3 | 0 | 17 |
| 4 | 0 | 6 |
| 5 | 0 | 12 |
| 6 | 1 | 15 |
| 7 | 1 | 26 |
| 8 | 1 | 29 |
| 9 | 0 | 14 |
| 10 | 1 | 58 |
| 11 | 0 | 2 |
| 12 | 1 | 31 |
| 13 | 1 | 26 |
| 14 | 0 | 11 |

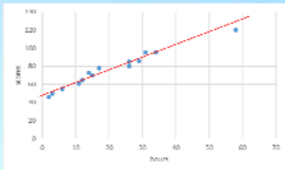


Simple Learn Regression
$y = w_0 + w_1 x$

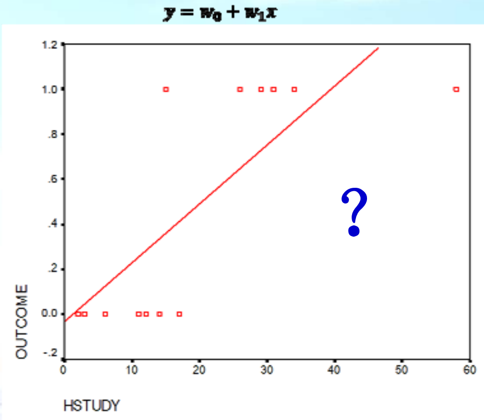


# 回归用来预测二值结果？

| Student id | Final score | Study Hours per Week |
|---|---|---|
| 1 | 0 | 3 |
| 2 | 1 | 34 |
| 3 | 0 | 17 |
| 4 | 0 | 6 |
| 5 | 0 | 12 |
| 6 | 1 | 15 |
| 7 | 1 | 26 |
| 8 | 1 | 29 |
| 9 | 0 | 14 |
| 10 | 1 | 58 |
| 11 | 0 | 2 |
| 12 | 1 | 31 |
| 13 | 1 | 26 |
| 14 | 0 | 11 |



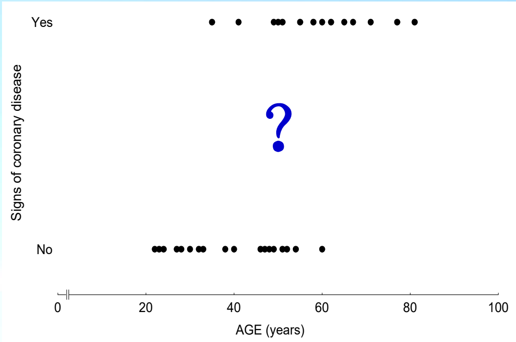
Simple Learn Regression
$y = w_0 + w_1 x$



# 用年龄回归冠心病

Age and signs of coronary heart disease (CD)

| Age | CD | Age | CD | Age | CD |
|---|---|---|---|---|---|
| 22 | 0 | 40 | 0 | 54 | 0 |
| 23 | 0 | 41 | 1 | 55 | 1 |
| 24 | 0 | 46 | 0 | 58 | 1 |
| 27 | 0 | 47 | 0 | 60 | 1 |
| 28 | 0 | 48 | 0 | 60 | 0 |
| 30 | 0 | 49 | 1 | 62 | 1 |
| 30 | 0 | 49 | 0 | 65 | 1 |
| 32 | 0 | 50 | 1 | 67 | 1 |
| 33 | 0 | 51 | 0 | 71 | 1 |
| 35 | 1 | 51 | 1 | 77 | 1 |
| 38 | 0 | 52 | 0 | 81 | 1 |

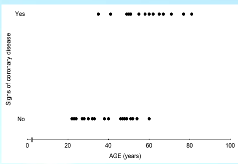





# 让我们来计数

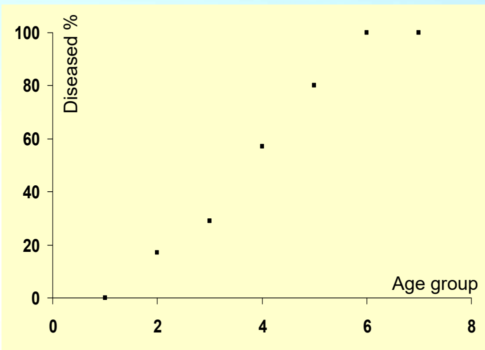
Age and signs of coronary heart disease (CD)

| Age | CD | Age | CD | Age | CD |
|---|---|---|---|---|---|
| 22 | 0 | 40 | 0 | 54 | 0 |
| 23 | 0 | 41 | 1 | 55 | 1 |
| 24 | 0 | 46 | 0 | 58 | 1 |
| 27 | 0 | 47 | 0 | 60 | 1 |
| 28 | 0 | 48 | 0 | 60 | 0 |
| 30 | 0 | 49 | 1 | 62 | 1 |
| 30 | 0 | 49 | 0 | 65 | 1 |
| 32 | 0 | 50 | 1 | 67 | 1 |
| 33 | 0 | 51 | 0 | 71 | 1 |
| 35 | 1 | 51 | 1 | 77 | 1 |
| 38 | 0 | 52 | 0 | 81 | 1 |



Prevalence (%) of signs of CD
according to age group

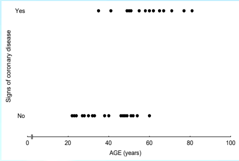
| | | Diseased | |
|---|---|---|---|
| Age group | # in group | # | % |
| 20 - 29 | 5 | 0 | 0 |
| 30 - 39 | 6 | 1 | 17 |
| 40 - 49 | 7 | 2 | 29 |
| 50 - 59 | 7 | 4 | 57 |
| 60 - 69 | 5 | 4 | 80 |
| 70 - 79 | 2 | 2 | 100 |
| 80 - 89 | 1 | 1 | 100 |



# 让我们来计数

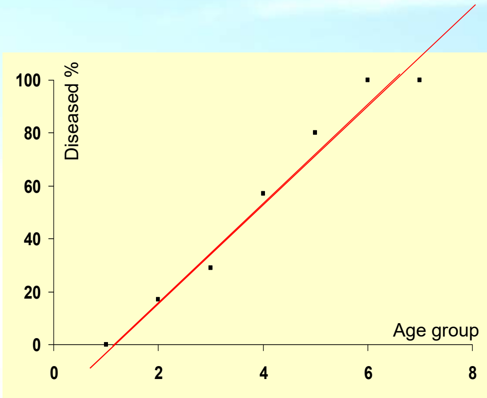
Age and signs of coronary heart disease (CD)

| Age | CD | Age | CD | Age | CD |
|---|---|---|---|---|---|
| 22 | 0 | 40 | 0 | 54 | 0 |
| 23 | 0 | 41 | 1 | 55 | 1 |
| 24 | 0 | 46 | 0 | 58 | 1 |
| 27 | 0 | 47 | 0 | 60 | 1 |
| 28 | 0 | 48 | 0 | 60 | 0 |
| 30 | 0 | 49 | 1 | 62 | 1 |
| 30 | 0 | 49 | 0 | 65 | 1 |
| 32 | 0 | 50 | 1 | 67 | 1 |
| 33 | 0 | 51 | 0 | 71 | 1 |
| 35 | 1 | 51 | 1 | 77 | 1 |
| 38 | 0 | 52 | 0 | 81 | 1 |




Prevalence (%) of signs of CD according to age group

| | | Diseased | |
|---|---|---|---|
| Age group | # in group | # | % |
| 20 - 29 | 5 | 0 | 0 |
| 30 - 39 | 6 | 1 | 17 |
| 40 - 49 | 7 | 2 | 29 |
| 50 - 59 | 7 | 4 | 57 |
| 60 - 69 | 5 | 4 | 80 |
| 70 - 79 | 2 | 2 | 100 |
| 80 - 89 | 1 | 1 | 100 |

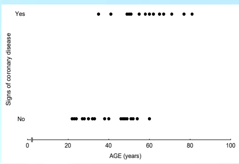





# 让我们来计数

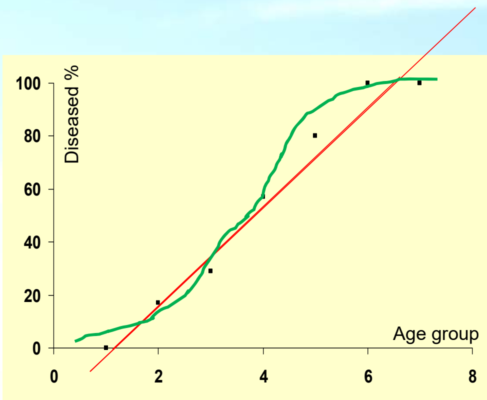
Age and signs of coronary heart disease (CD)

| Age | CD | Age | CD | Age | CD |
|---|---|---|---|---|---|
| 22 | 0 | 40 | 0 | 54 | 0 |
| 23 | 0 | 41 | 1 | 55 | 1 |
| 24 | 0 | 46 | 0 | 58 | 1 |
| 27 | 0 | 47 | 0 | 60 | 1 |
| 28 | 0 | 48 | 0 | 60 | 0 |
| 30 | 0 | 49 | 1 | 62 | 1 |
| 30 | 0 | 49 | 0 | 65 | 1 |
| 32 | 0 | 50 | 1 | 67 | 1 |
| 33 | 0 | 51 | 0 | 71 | 1 |
| 35 | 1 | 51 | 1 | 77 | 1 |
| 38 | 0 | 52 | 0 | 81 | 1 |

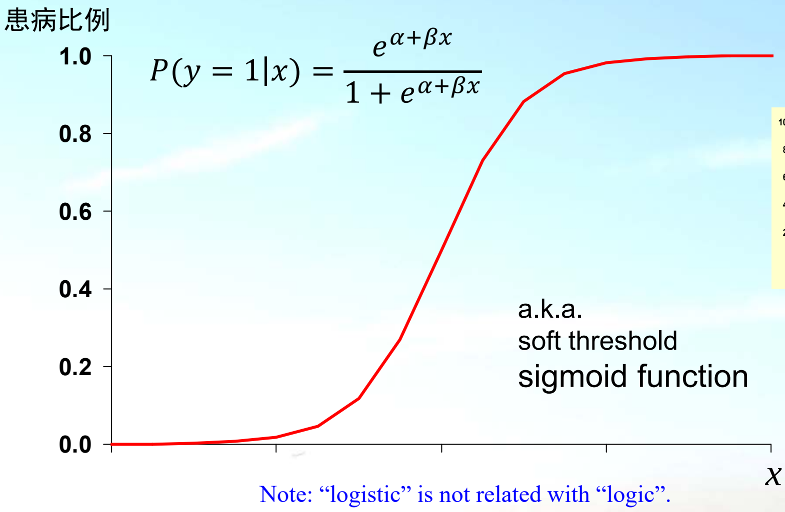



Prevalence (%) of signs of CD according to age group

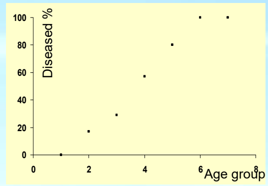
| | | Diseased | |
|---|---|---|---|
| Age group | # in group | # | % |
| 20 - 29 | 5 | 0 | 0 |
| 30 - 39 | 6 | 1 | 17 |
| 40 - 49 | 7 | 2 | 29 |
| 50 - 59 | 7 | 4 | 57 |
| 60 - 69 | 5 | 4 | 80 |
| 70 - 79 | 2 | 2 | 100 |
| 80 - 89 | 1 | 1 | 100 |



# 罗杰斯特函数Logistic function




$$P(y=1|x)=\frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}}$$

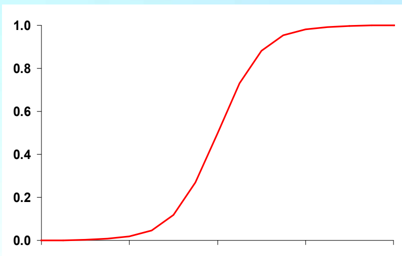



a.k.a.
soft threshold
sigmoid function

Note: "logistic" is not related with "logic".
So I prefer the translation 罗杰斯特 or 罗杰斯蒂 .

Xuegong Zhang 21

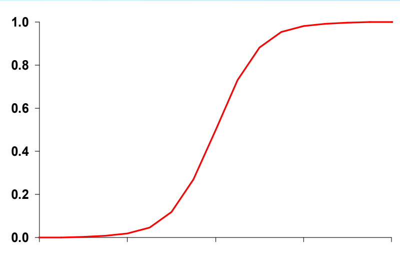
# 罗杰斯特回归

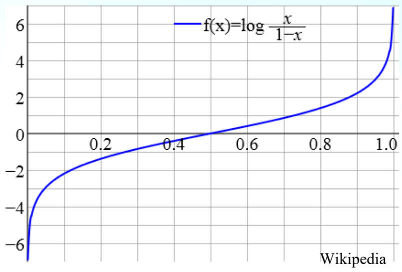



$$P(y|x)=\frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}}$$

Xuegong Zhang 22

## 罗杰斯特回归与几率




$$P(y|x) = \frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}}$$

反函数

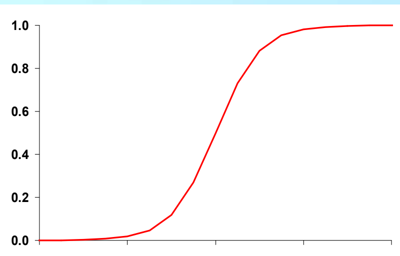
$$\ln\left[\frac{P(y|x)}{1-P(y|x)}\right] = \alpha + \beta x$$

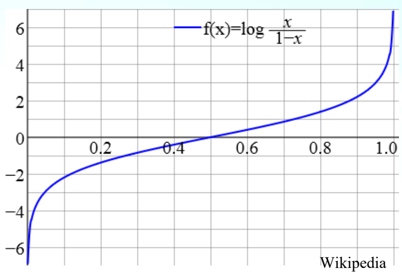
$P(y|x)$的罗杰特（logit）函数

---- 对数几率（log odds）

Xuegong Zhang 23

## 罗杰斯特回归与几率




$$P(y|x) = \frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}}$$

反函数

$$\ln\left[\frac{P(y|x)}{1-P(y|x)}\right] = \alpha + \beta x$$

$P(y|x)$的罗杰特（logit）函数

---- 对数几率（log odds）

表示发生或不发生的概率

注意中文中的混淆词：几率、机率、概率、可能性、…

Xuegong Zhang 24

# 多元罗杰斯特回归

$$P(y|\boldsymbol{x}) = \frac{e^{\alpha+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_p x_p}}{1+e^{\alpha+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_p x_p}}$$

$$\ln\left(\frac{P}{1-P}\right) = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots \beta_i x_i$$

$$odds = \frac{P}{1-P} = e^{\alpha+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_p x_p}$$



# 多元罗杰斯特回归

$$P(y|\boldsymbol{x}) = \frac{e^{\alpha+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_p x_p}}{1+e^{\alpha+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_p x_p}}$$

$$\ln\left(\frac{P}{1-P}\right) = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots \beta_i x_i$$

$$odds = \frac{P}{1-P} = e^{\alpha+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_p x_p}$$

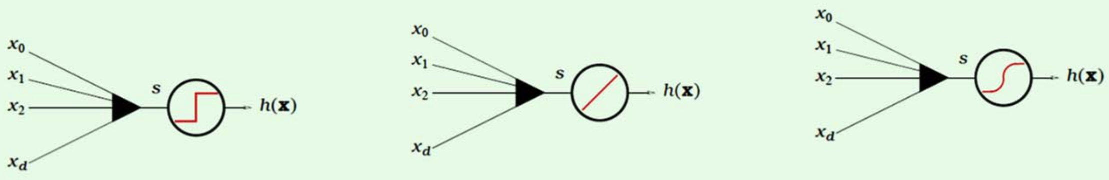
- $\beta_i$的解释
  - 在其他因素不变的情况下，因素$x_i$增加一个单位带来的对数几率的增加
  - 可以用来从流行病学数据中研究各种因素与患病的关系

用线性加权和激活函数来看
三种线性机器

# 三种线性机器

$$s = \sum_{i=0}^{d} w_i x_i$$

| linear classification | linear regression | logistic regression |
|---|---|---|
| $h(\mathbf{x}) = \mathrm{sign}(s)$ | $h(\mathbf{x}) = s$ | $h(\mathbf{x}) = \theta(s)$ |



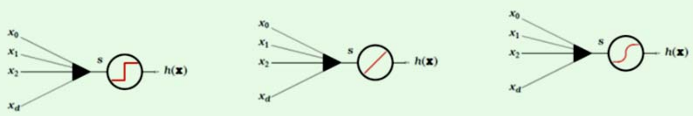

Abu-Mostafa, Magdon-Ismail, Lin, *Learning from Data, Lecture 9*

Xuegong Zhang 27

# 罗杰斯特回归的概率意义

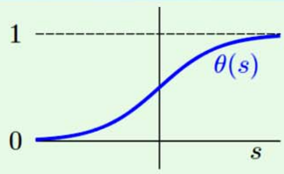
$$s = \sum_{i=0}^{d} w_i x_i$$

| linear classification | linear regression | logistic regression |
|---|---|---|
| $h(\mathbf{x}) = \mathrm{sign}(s)$ | $h(\mathbf{x}) = s$ | $h(\mathbf{x}) = \theta(s)$ |




Abu-Mostafa, Magdon-Ismail, Lin, *Learning from Data, Lecture 9*

$$\theta(s) = \frac{e^s}{1 + e^s}$$




- $s = \boldsymbol{w}^T\boldsymbol{x}$是事件的总信息，是各种因素（特征）的加权求和
- $h(\boldsymbol{x}) = \theta(s)$是对事件$y = 1$ 概率的估计

Xuegong Zhang 28

单选题 1分 设置

休息4分钟，回到座位后请答题

A 已回座位

B 还没有

提交



## 机器学习的基本要素：感知器版

- 怎样造一个学习机器？
  - 它需要老师
    → 我们设计它（特征和模型） $y = \mathrm{sgn}(\sum_{i=1}^{d} w_i x_i + w_0)$
  - 它需要训练/学习材料
    → 训练数据 $\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_N, y_N)\}$, $\boldsymbol{x}_j \in R^{d+1}, y_j \in \{-1,1\}$
  - 我们需要为它树立学习的目标
    → 目标函数、学习准则 $\min J_P(\boldsymbol{\alpha}) = \sum_{\mathbf{y}_j \in \mathcal{Y}^k} (-\boldsymbol{\alpha}^T \mathbf{y}_j)$
  - 我们需要告诉它怎样学
    → 学习/训练算法 $\boldsymbol{\alpha}(k+1) = \boldsymbol{\alpha}(k) - \rho_k \nabla J = \boldsymbol{\alpha}(k) + \rho_k \sum_{\mathbf{y}_j \in Y^k} \mathbf{y}_j$

# 机器学习的基本要素：线性回归版

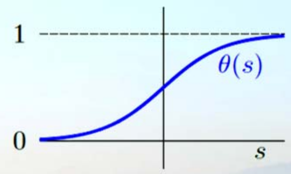
- 怎样造一个学习机器？
  - 它需要老师
    - → 我们设计它（特征和模型） $f(\boldsymbol{x}) = \sum_{i=0}^{d} w_i x_i = \boldsymbol{w}^T \boldsymbol{x}$
  - 它需要训练/学习材料
    - → 训练数据 $\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_N, y_N)\}$, $\boldsymbol{x}_j \in R^{d+1}, y_j \in R$
  - 我们需要为它树立学习的目标
    - → 目标函数、学习准则 $\min E = \frac{1}{N}\sum_{j=1}^{N}\left(f(\boldsymbol{x}_j) - y_j\right)^2$
  - 我们需要告诉它怎样学
    - → 学习/训练算法 $\boldsymbol{w}(k+1) = \boldsymbol{w}(k) - \rho_k \nabla E$



# 机器学习的基本要素：罗杰斯特回归版？

- 怎样造一个学习机器？
  - 它需要老师
    - → 我们设计它（特征和模型） $h(\boldsymbol{x}) = \theta(\boldsymbol{w}^T \boldsymbol{x})$
  - 它需要训练/学习材料
    - → 训练数据 $\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_N, y_N)\}$, $\boldsymbol{x}_j \in R^{d+1}, y_j \in \{-1,1\}$
  - 我们需要为它树立学习的目标
    - → 目标函数、学习准则 ？
  - 我们需要告诉它怎样学
    - → 学习/训练算法？

$$\theta(s) = \frac{e^s}{1+e^s}$$



# 似然函数

- 设独立同分布(i.i.d.)样本集 $\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_N, y_N)\}$, $\boldsymbol{x}_j \in R^{d+1}, y_j \in \{-1,1\}$ 依以下概率产生：

$$P(y|\boldsymbol{x}) = \begin{cases} f(\boldsymbol{x}) & for\ y = +1 \\ 1 - f(\boldsymbol{x}) & for\ y = -1 \end{cases}$$

生成模型
Generative model

- 罗杰斯特回归用$h(\boldsymbol{x}) = \theta(\boldsymbol{w}^T\boldsymbol{x})$估计$f(\boldsymbol{x})$

生成模型的概念



# 似然函数

- 设独立同分布(i.i.d.)样本集 $\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_N, y_N)\}$, $\boldsymbol{x}_j \in R^{d+1}, y_j \in \{-1,1\}$ 依以下概率产生：

$$P(y|\boldsymbol{x}) = \begin{cases} f(\boldsymbol{x}) & for\ y = +1 \\ 1 - f(\boldsymbol{x}) & for\ y = -1 \end{cases}$$

生成模型
Generative model

- 罗杰斯特回归用$h(\boldsymbol{x}) = \theta(\boldsymbol{w}^T\boldsymbol{x})$估计$f(\boldsymbol{x})$
- 似然函数（Likelihood）：
  - 对数据中的一个实例$(\boldsymbol{x}_j, y_j)$，如果$h = f$，我们有多大可能对$\boldsymbol{x}_j$得到$y_j$？

$$P(y_j|\boldsymbol{x}_j) = \begin{cases} h(\boldsymbol{x}_j) & for\ y_j = +1 \\ 1 - h(\boldsymbol{x}_j) & for\ y_j = -1 \end{cases}$$

  - 换言之，已经有这个数据实例， $h$有多大可能是产生数据的模型？

## 似然函数

- 设独立同分布(i.i.d.)样本集 $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\}$，$\boldsymbol{x}_j \in R^{d+1}, y_j \in \{-1,1\}$ 依以下概率产生：

$$P(y|\boldsymbol{x}) = \begin{cases} f(\boldsymbol{x}) & for\ y = +1 \\ 1 - f(\boldsymbol{x}) & for\ y = -1 \end{cases}$$

生成模型
Generative model

- 罗杰斯特回归用$h(\boldsymbol{x}) = \theta(\boldsymbol{w}^T\boldsymbol{x})$估计$f(\boldsymbol{x})$
- 似然函数（Likelihood）：
  - 对数据中的一个实例$(\boldsymbol{x}_j, y_j)$，如果$h = f$，我们有多大可能对$\boldsymbol{x}_j$得到$y_j$？

$$P(y_j|\boldsymbol{x}_j) = \begin{cases} h(\boldsymbol{x}_j) & for\ y_j = +1 \\ 1 - h(\boldsymbol{x}_j) & for\ y_j = -1 \end{cases}$$

  - 换言之，已经有这个数据实例，$h$有多大可能是产生数据的模型？
- 注意到$\theta(-s) = 1 - \theta(s)$，$\boldsymbol{x}_j$上的似然函数可写为：

H

$$P(y_j|\boldsymbol{x}_j) = \theta(y_j \boldsymbol{w}^T \boldsymbol{x}_j)$$



## 似然函数

- 设独立同分布(i.i.d.)样本集 $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\}$，$\boldsymbol{x}_j \in R^{d+1}, y_j \in \{-1,1\}$ 依以下概率产生：

$$P(y|\boldsymbol{x}) = \begin{cases} f(\boldsymbol{x}) & for\ y = +1 \\ 1 - f(\boldsymbol{x}) & for\ y = -1 \end{cases}$$

生成模型
Generative model

- 罗杰斯特回归用$h(\boldsymbol{x}) = \theta(\boldsymbol{w}^T\boldsymbol{x})$估计$f(\boldsymbol{x})$

问题：似然函数是谁的函数？

- 似然函数（Likelihood）：
  - 对数据中的一个实例$(\boldsymbol{x}_j, y_j)$，如果$h = f$，我们有多大可能对$\boldsymbol{x}_j$得到$y_j$？

$$P(y_j|\boldsymbol{x}_j) = \begin{cases} h(\boldsymbol{x}_j) & for\ y_j = +1 \\ 1 - h(\boldsymbol{x}_j) & for\ y_j = -1 \end{cases}$$

  - 换言之，已经有这个数据实例，$h$有多大可能是产生数据的模型？
- 注意到$\theta(-s) = 1 - \theta(s)$，$\boldsymbol{x}_j$上的似然函数可写为：

$$P(y_j|\boldsymbol{x}_j) = \theta(y_j \boldsymbol{w}^T \boldsymbol{x}_j)$$

单选题 10分

设置

# 似然函数是谁的函数？

$$P(y_j|\boldsymbol{x}_j) = \theta(y_j \boldsymbol{w}^T \boldsymbol{x}_j)$$

A 是$y_i$的函数

B 是$\boldsymbol{w}$的函数

C 是$\boldsymbol{x}_i$的函数

D 是$(\boldsymbol{x}_j, y_j)$的函数

提交

Xuegong Zhang 37

# 似然函数

- 设独立同分布(i.i.d.)样本集 $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\}$，$\boldsymbol{x}_j \in R^{d+1}, y_j \in \{-1,1\}$ 依以下概率产生：

$$P(y|\boldsymbol{x}) = \begin{cases} f(\boldsymbol{x}) & for\ y = +1 \\ 1 - f(\boldsymbol{x}) & for\ y = -1 \end{cases}$$

生成模型
Generative model

- 罗杰斯特回归用$h(\boldsymbol{x}) = \theta(\boldsymbol{w}^T\boldsymbol{x})$估计$f(\boldsymbol{x})$
- 似然函数（Likelihood）：
  - 对数据中的一个实例$(\boldsymbol{x}_j, y_j)$，如果$h = f$，我们有多大可能对$\boldsymbol{x}_j$得到$y_j$？

$$P(y_j|\boldsymbol{x}_j) = \begin{cases} h(\boldsymbol{x}_j) & for\ y_j = +1 \\ 1 - h(\boldsymbol{x}_j) & for\ y_j = -1 \end{cases}$$

  - 换言之，已经有这个数据实例，$h$有多大可能是产生数据的模型？
- 注意到$\theta(-s) = 1 - \theta(s)$，模型在$\boldsymbol{x}_j$上的似然函数可写为：

$$l(h|(\boldsymbol{x}_j, y_j)) = P(y_j|\boldsymbol{x}_j, h) = \theta(y_j \boldsymbol{w}^T \boldsymbol{x}_j)$$

Xuegong Zhang 38

## 罗杰斯特回归的目标：似然函数最大化

- 参数为$\boldsymbol{w}$的罗杰斯特模型在i.i.d.数据$\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_N, y_N)\}$，$\boldsymbol{x}_j \in R^{d+1}, y_j \in \{-1,1\}$上的似然函数是

$$L(\boldsymbol{w}) = \prod_{j=1}^{N} P(y_j|\boldsymbol{x}_j) = \prod_{j=1}^{N} \theta(y_j \boldsymbol{w}^T \boldsymbol{x}_j)$$

Xuegong Zhang 39

## 罗杰斯特回归的目标：似然函数最大化

- 参数为$\boldsymbol{w}$的罗杰斯特模型在i.i.d.数据$\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_N, y_N)\}$，$\boldsymbol{x}_j \in R^{d+1}, y_j \in \{-1,1\}$上的似然函数是

$$L(\boldsymbol{w}) = \prod_{j=1}^{N} P(y_j|\boldsymbol{x}_j) = \prod_{j=1}^{N} \theta(y_j \boldsymbol{w}^T \boldsymbol{x}_j)$$

- 最大化似然函数，等价于：

取对数，防止小于1的概率相乘导致取值太小。

$$\min \quad E(\boldsymbol{w}) = -\frac{1}{N} ln(L(\boldsymbol{w})) = -\frac{1}{N} ln\left(\prod_{j=1}^{N} \theta(y_j \boldsymbol{w}^T \boldsymbol{x}_j)\right)$$

$$= \frac{1}{N}\sum_{j=1}^{N} \ln\left(\frac{1}{\theta(y_j \boldsymbol{w}^T \boldsymbol{x}_j)}\right)$$

$$= \frac{1}{N}\sum_{j=1}^{N} ln\left(1 + e^{-y_j \boldsymbol{w}^T \boldsymbol{x}_j}\right)$$

$$\left[\theta(s) = \frac{1}{1 + e^{-s}}\right]$$

Xuegong Zhang 40

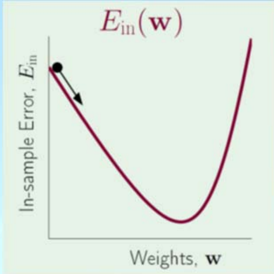
## 求解：梯度下降法

梯度下降的一般原理：

$$\boldsymbol{w}(k+1)=\boldsymbol{w}(k)+\eta\hat{\boldsymbol{v}}$$

其中，$\eta$：步长（学习率）

$$\hat{\boldsymbol{v}}=-\nabla E(\boldsymbol{w}(k))$$






Abu-Mostafa, Magdon-Ismail, Lin, *Learning from Data, Lecture 9*

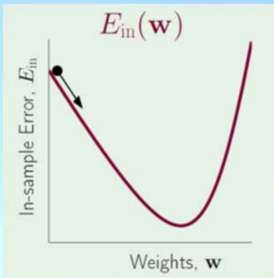
## 求解：梯度下降法

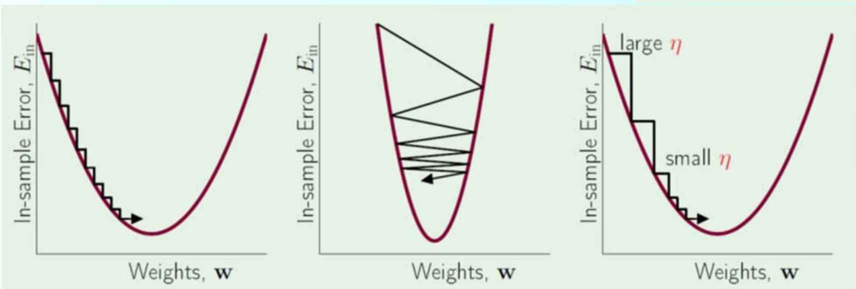
梯度下降的一般原理：

$$\boldsymbol{w}(k+1)=\boldsymbol{w}(k)+\eta\hat{\boldsymbol{v}}$$

其中，$\eta$：步长（学习率）

$$\hat{\boldsymbol{v}}=-\nabla E(\boldsymbol{w}(k))$$




$\eta$ too small　　$\eta$ too large　　variable $\eta$ – just right



Abu-Mostafa, Magdon-Ismail, Lin, *Learning from Data, Lecture 9*

## 罗杰斯特回归算法

1. Set $k = 0$, initialize $\boldsymbol{w}(0)$
2. Do
   – Compute the gradient $\nabla E = -\frac{1}{N}\sum_{j=1}^{N}\frac{y_j \boldsymbol{x}_j}{1+e^{y_j \boldsymbol{w}(k)^T \boldsymbol{x}_j}}$

   – Update the weights $\boldsymbol{w}(k+1) = \boldsymbol{w}(k) - \eta\nabla E$, set $k = k+1$

   Until the stopping criterion met
3. Return the final weights $\boldsymbol{w}$

Abu-Mostafa, Magdon-Ismail, Lin, *Learning from Data, Lecture 9*

Xuegong Zhang 43

## 罗杰斯特回归算法

1. Set $k = 0$, initialize $\boldsymbol{w}(0)$
2. Do
   – Compute the gradient $\nabla E = -\frac{1}{N}\sum_{j=1}^{N}\frac{y_j \boldsymbol{x}_j}{1+e^{y_j \boldsymbol{w}(k)^T \boldsymbol{x}_j}}$

   – Update the weights $\boldsymbol{w}(k+1) = \boldsymbol{w}(k) - \eta\nabla E$, set $k = k+1$

   Until the stopping criterion met
3. Return the final weights $\boldsymbol{w}$

Abu-Mostafa, Magdon-Ismail, Lin, *Learning from Data, Lecture 9*

- 初始化：可以全零，更好是小随机数，比如0均值、小方差的正态分布
- 终止条件：梯度低于一定阈值，或迭代次数达到预设上限

Xuegong Zhang 44

# 机器学习的基本要素：罗杰斯特回归版

- 怎样造一个学习机器？
  - 它需要老师
    - → 我们设计它（特征和模型） $h(\boldsymbol{x}) = \theta(\boldsymbol{w}^T\boldsymbol{x})$
  - 它需要训练/学习材料
    - → 训练数据 $\{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_N, y_N)\}$, $\boldsymbol{x}_j \in R^{d+1}, y_j \in \{-1,1\}$
  - 我们需要为它树立学习的目标
    - → 目标函数、学习准则 $\min E(\boldsymbol{w}) = \frac{1}{N}\sum_{j=1}^{N} ln\left(1 + e^{-y_j \boldsymbol{w}^T \boldsymbol{x}_j}\right)$
  - 我们需要告诉它怎样学
    - → 学习/训练算法 $\boldsymbol{w}(k+1) = \boldsymbol{w}(k) - \eta \nabla E$



# 线性学习机器小结

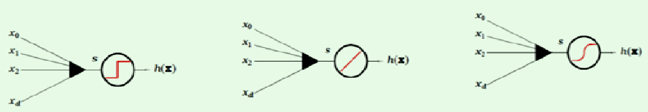
## 模型

$$s = \sum_{i=0}^{d} w_i x_i$$

linear classification $h(\mathbf{x}) = \text{sign}(s)$

linear regression $h(\mathbf{x}) = s$

logistic regression $h(\mathbf{x}) = \theta(s)$



## 学习算法

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) - \rho_k \nabla E$$

## 目标

- For perceptron

$$\min \quad J_P(\boldsymbol{\alpha}) = \sum_{y_j \in Y^k} (-\boldsymbol{\alpha}^T \boldsymbol{y}_j)$$

- For linear regression

$$\min \quad E(\boldsymbol{w}) = \frac{1}{N}\sum_{j=1}^{N} (\boldsymbol{w}^T\boldsymbol{x}_j - y_j)^2$$

- For logistic regression

$$\min \quad E(\boldsymbol{w}) = \frac{1}{N}\sum_{j=1}^{N} ln\left(1 + e^{-y_j \boldsymbol{w}^T \boldsymbol{x}_j}\right)$$

# 休息1分钟



# 2.6 多类分类器
## Multicategory Linear Classifiers

# 多类分类器

- FLD、感知器、MSE、罗杰斯特回归都是两类分类器
- 如何实现多类分类？

请用弹幕回答

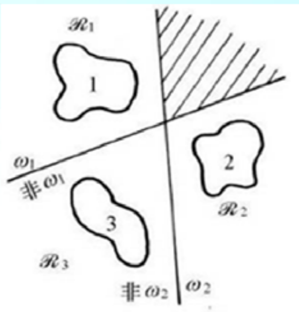
  – 用多个两类分类器实现多类分类
  – 构造多类分类器



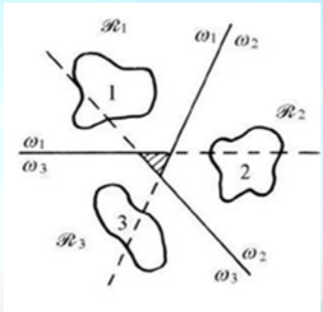
# 多类分类器

- FLD、感知器、MSE、罗杰斯特回归都是两类分类器
- 如何实现多类分类？
  – 用多个两类分类器实现多类分类
  – 构造多类分类器

- 用两类分类器实现多类分类的两种策略
  – 一对多（One-vs-rest, one-vs-all, one-over-all）
  – 一对一（Pairwise classification）



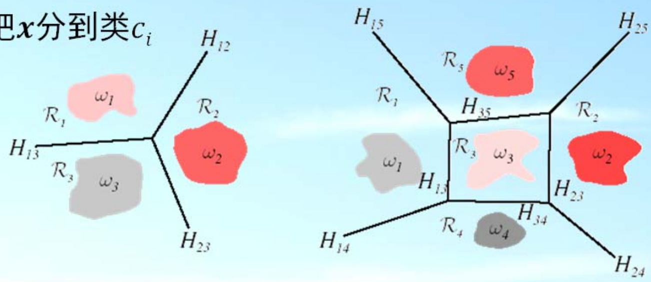

(a) $\omega_i$/非$\omega_i$




(b) $\omega_i$/$\omega_j$

# 多类线性机器（多类线性判别）

- 给定$C$类，定义$C$个判别函数$g_i(\boldsymbol{y}) = \boldsymbol{a}_i^T \boldsymbol{y}$
- 决策：若$g_i(\boldsymbol{y})> g_j(\boldsymbol{y}), \forall j \neq i$，则把$\boldsymbol{x}$分到类$c_i$



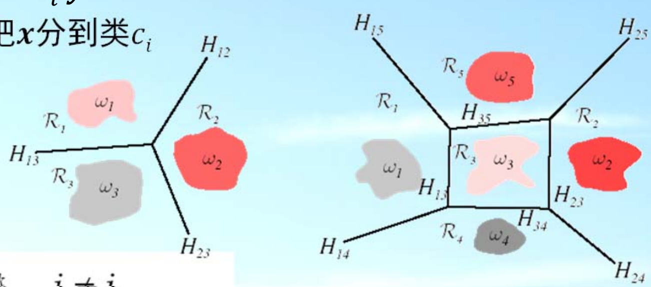

from Duda *et al.*



from Duda *et al.*

# 多类线性机器（多类线性判别）

- 给定$C$类，定义$C$个判别函数$g_i(\boldsymbol{y}) = \boldsymbol{a}_i^T \boldsymbol{y}$
- 决策：若$g_i(\boldsymbol{y})> g_j(\boldsymbol{y}), \forall j \neq i$，则把$\boldsymbol{x}$分到类$c_i$



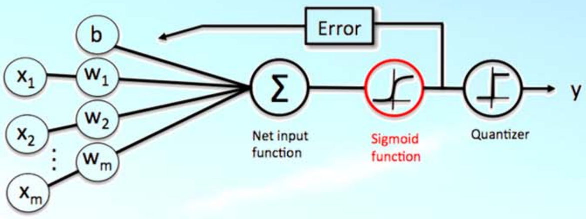

from Duda *et al.*

- 学习算法：

If $y^k \in \Upsilon_i$ but $\alpha_i(k)^T y^k \le \alpha_j(k)^T y^k$ , $j \neq i$

Then
$$\begin{cases} \alpha_i(k+1) = \alpha_i(k) + \rho_k y^k \\ \alpha_j(k+1) = \alpha_j(k) - \rho_k y^k \\ \alpha_l(k+1) = \alpha_l(k) \quad l \neq i, j \end{cases}$$

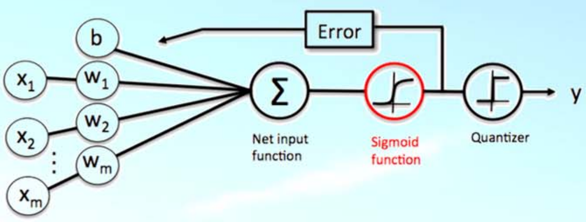
# 多类罗杰斯特回归与软最大（Softmax）

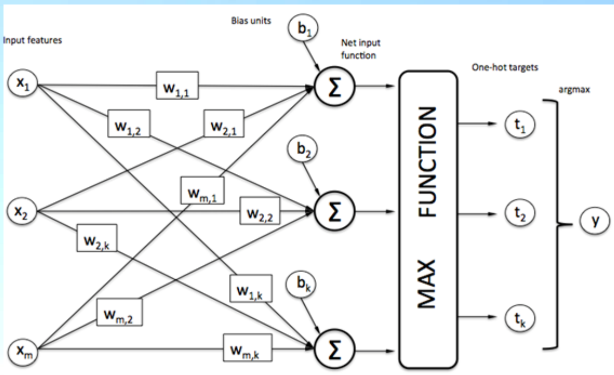



- 两类：
  - 线性加权求和 → Logistic函数 → 类别概率与阈值比较 → 分类



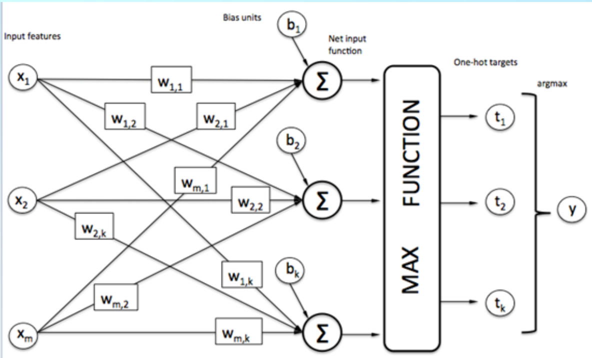

# 多类罗杰斯特回归与软最大（Softmax）


- 两类：
  - 线性加权求和 → Logistic函数 → 类别概率与阈值比较 → 分类
- 多类：
  - 线性加权求和 → 各类的Logistic函数 → 各类概率比较 → 分类

# 多类罗杰斯特回归与软最大（Softmax）



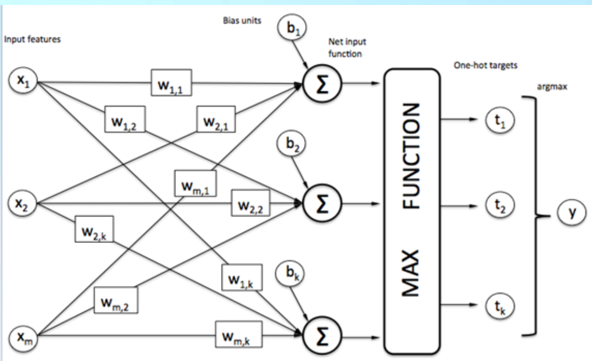

- 问题：未考虑同一样本只能属于一类

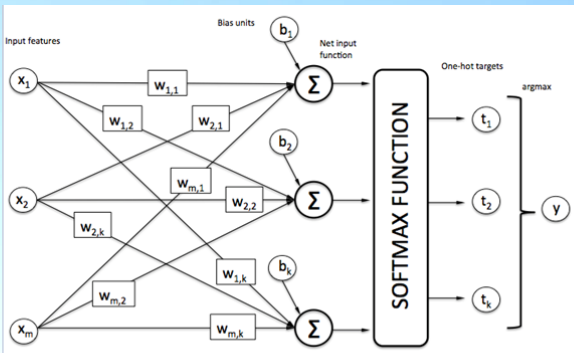
Xuegong Zhang

Modified from https://rasbt.github.io/mlxtend/user_guide/classifier/SoftmaxRegression/

55

# 多类罗杰斯特回归与软最大（Softmax）



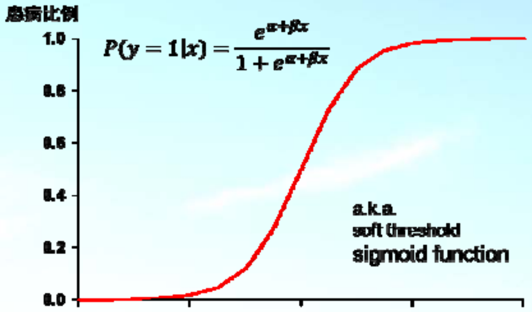

- 问题：未考虑同一样本只能属于一类
- 解决：多类logistic函数→Softmax（软最大）

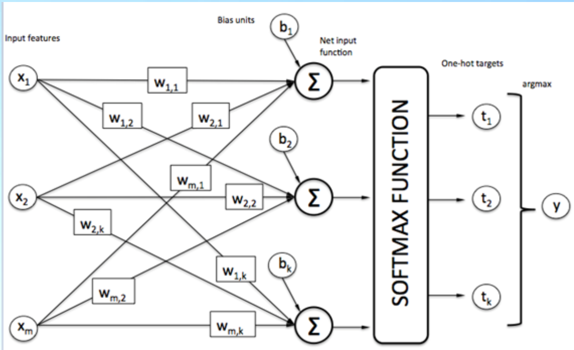
Xuegong Zhang

Modified from https://rasbt.github.io/mlxtend/user_guide/classifier/SoftmaxRegression/

56

## 软最大Softmax




$$P(y=1\mid \mathbf{x})=\frac{e^{\alpha+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_p x_p}}{1+e^{\alpha+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_p x_p}}$$

$$P(y=j|\boldsymbol{x})=\frac{e^{\boldsymbol{w}_j\cdot \boldsymbol{x}}}{\sum_{k=1}^{K} e^{\boldsymbol{w}_k\cdot \boldsymbol{x}}},$$
$$j=1,\cdots,k$$

Softmax函数（归一化指数函数）

注：机器学习领域中有很多random names ☺



## 本章知识点

- 机器学习的基本概念
- 线性学习机器的基本思想
  - 模型、目标函数、梯度下降优化
- Fisher线性判别、感知器、线性回归、MSE、罗杰斯特回归、Softmax
- 似然函数的概念

单选题 1分
设置
休息4分钟，回到座位后请答题
A 已回座位
B 还没有
Xuegong Zhang
提交
59