二、模型评估与选择

主讲教师：周志华

F1

F1 度量：

$$F1 = \frac{2 \times P \times R}{P + R}$$

$$= \frac{2 \times TP}{样例总数 + TP - TN}$$

$$\frac{1}{F1} = \frac{1}{2} \cdot \left(\frac{1}{P} + \frac{1}{R}\right)$$

若对查准率/查全率有不同偏好：

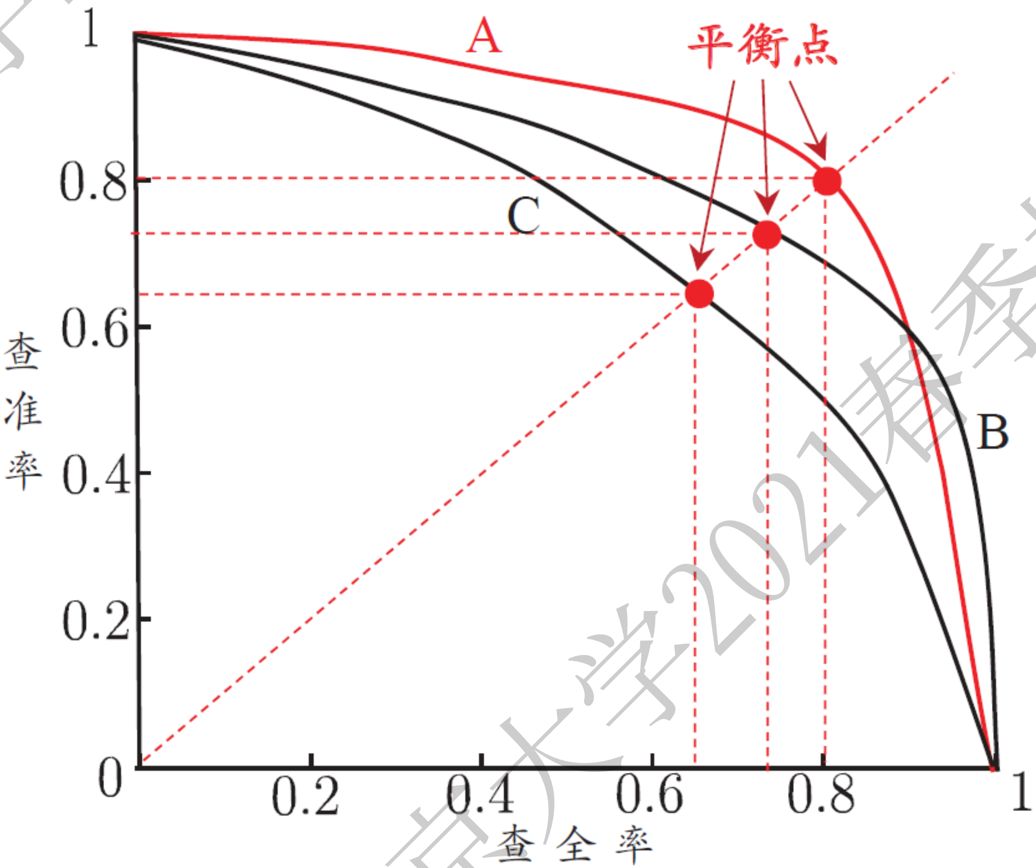
$$F_\beta = \frac{(1 + \beta^2) \times P \times R}{(\beta^2 \times P) + R}$$

$$\frac{1}{F_\beta} = \frac{1}{1 + \beta^2} \cdot \left(\frac{1}{P} + \frac{\beta^2}{R}\right)$$

$\beta > 1$ 时查全率有更大影响；$\beta < 1$ 时查准率有更大影响

PR图，BEP

根据学习器的预测结果按正例可能性大小对样例进行排序，并逐个把样本作为正例进行预测



PR图：

- 学习器 A 优于 学习器 C
- 学习器 B 优于 学习器 C
- 学习器 A ?? 学习器 B

BEP：

- 学习器 A 优于 学习器 B
- 学习器 A 优于 学习器 C
- 学习器 B 优于 学习器 C

宏xx vs. 微xx

若能得到多个混淆矩阵:

(例如多次训练/测试的结果，多分类的两两混淆矩阵)

宏(macro-)查准率、查全率、F1

$$\text{macro-}P = \frac{1}{n}\sum_{i=1}^{n} P_i \ ,$$

$$\text{macro-}R = \frac{1}{n}\sum_{i=1}^{n} R_i \ ,$$

$$\text{macro-}F1 = \frac{2 \times \text{macro-}P \times \text{macro-}R}{\text{macro-}P + \text{macro-}R} \ .$$

微(micro-)查准率、查全率、F1

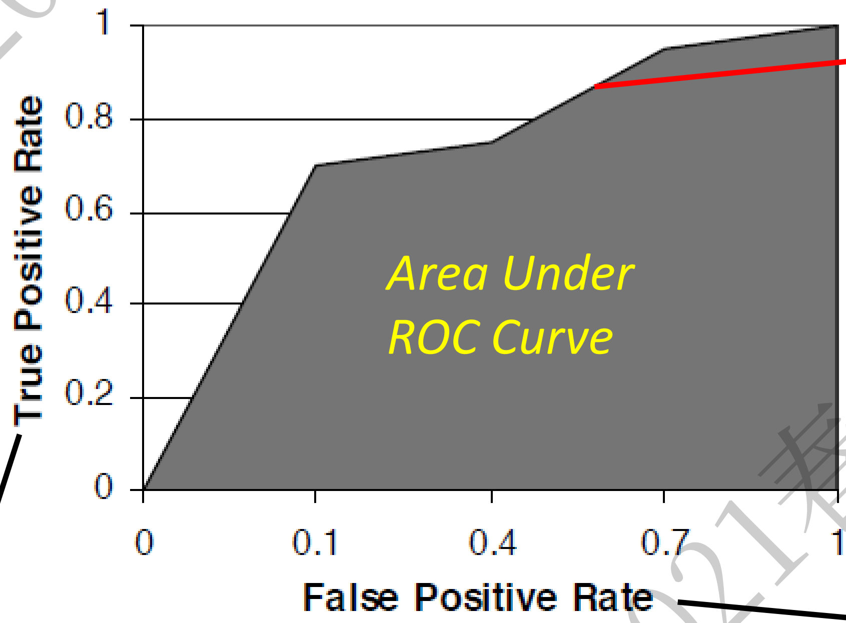
$$\text{micro-}P = \frac{\overline{TP}}{\overline{TP} + \overline{FP}} \ ,$$

$$\text{micro-}R = \frac{\overline{TP}}{\overline{TP} + \overline{FN}} \ ,$$

$$\text{micro-}F1 = \frac{2 \times \text{micro-}P \times \text{micro-}R}{\text{micro-}P + \text{micro-}R} \ .$$

ROC, AUC

AUC: **A**rea **U**nder the ROC **C**urve



ROC (Receiver Operating Characteristic) Curve [Green & Swets, Book 66; Spackman, IWML'89]

The bigger, the better

$$\boxed{tpr = \frac{TP}{TP+FN} = \frac{TP}{m_+}}$$

$$\boxed{fpr = \frac{FP}{FP+TN} = \frac{FP}{m_-}}$$

$$\mathrm{AUC} = 1 - \frac{1}{m^+ m^-} \sum_{x^+ \in D^+} \sum_{x^- \in D^-} \left(\mathbb{I}\left(f(x^+) < f(x^-)\right) + \frac{1}{2}\mathbb{I}\left(f(x^+) = f(x^-)\right) \right)$$

非均等代价

犯不同的错误往往会造成不同的损失

此时需考虑 “非均等代价”
(unequal cost)

表 2.2 二分类代价矩阵

真实类别	预测类别	
	第 0 类	第 1 类
第 0 类	0	$cost_{01}$
第 1 类	$cost_{10}$	0

- 代价敏感(cost-sensitive)错误率：

$$E(f;D;cost)=\frac{1}{m}\left(\sum_{\boldsymbol{x}_i\in D^+}\mathbb{I}\left(f\left(\boldsymbol{x}_i\right)\neq y_i\right)\times cost_{01}\right.$$
$$\left.+\sum_{\boldsymbol{x}_i\in D^-}\mathbb{I}\left(f\left(\boldsymbol{x}_i\right)\neq y_i\right)\times cost_{10}\right)$$

模型选择 (model selection)

三个关键问题：

- 如何获得测试结果？ ⇨ 评估方法
- 如何评估性能优劣？ ⇨ 性能度量
- 如何判断实质差别？ ⇨ 比较检验

比较检验

在某种度量下取得评估结果后，是否可以直接比较以评判优劣？

NO！ 因为：
- 测试性能不等于泛化性能
- 测试性能随着测试集的变化而变化
- 很多机器学习算法本身有一定的随机性

机器学习 …… “概率近似正确”

常用方法

统计假设检验 (hypothesis test) 为学习器性能比较提供了重要依据

统计显著性

- 两学习器比较
  - 交叉验证 t 检验 (基于成对 t 检验)
    - k 折交叉验证； 5x2交叉验证
  - McNemar 检验 (基于列联表，卡方检验)
- 多学习器比较
  - Friedman + Nemenyi
    - Friedman检验 (基于序值，F检验; 判断”是否都相同”)
    - Nemenyi 后续检验 (基于序值，进一步判断两两差别)

Friedman 检验图

横轴为平均序值，每个算法圆点为其平均序值，线段为临界阈值的大小

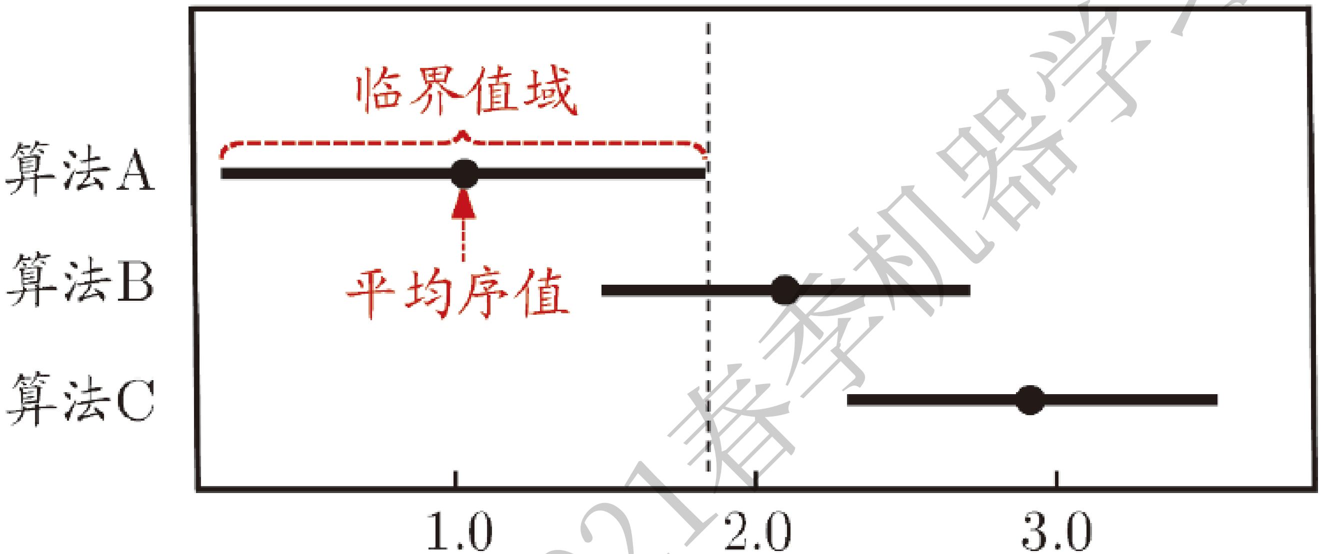


图 2.8 Friedman 检验图

若两个算法有交叠 (A 和 B)，则说明没有显著差别；
否则有显著差别 (A 和 C)，算法 A 显著优于算法 C

"误差"包含了哪些因素？

换言之，从机器学习的角度看，

"误差"从何而来?

偏差-方差分解 (bias-variance decomposition)

对回归任务，泛化误差可通过“偏差-方差分解”拆解为：

$$E(f;D) = bias^2(\boldsymbol{x}) + var(\boldsymbol{x}) + \varepsilon^2$$

期望输出与真实输出的差别

$$bias^2(\boldsymbol{x}) = \left(\bar{f}(\boldsymbol{x}) - y\right)^2$$

同样大小的训练集的变动，所导致的性能变化

$$var(\boldsymbol{x}) = \mathbb{E}_D\left[\left(f(\boldsymbol{x};D) - \bar{f}(\boldsymbol{x})\right)^2\right]$$

训练样本的标记与真实标记有区别

表达了当前任务上任何学习算法所能达到的期望泛化误差下界

$$\varepsilon^2 = \mathbb{E}_D\left[(y_D - y)^2\right]$$

泛化性能是由学习算法的能力、数据的充分性以及学习任务本身的难度共同决定

偏差-方差窘境 (bias-variance dillema)

一般而言，偏差与方差存在冲突：

- 训练不足时，学习器拟合能力不强，偏差主导
- 随着训练程度加深，学习器拟合能力逐渐增强，方差逐渐主导
- 训练充足后，学习器的拟合能力很强，方差主导

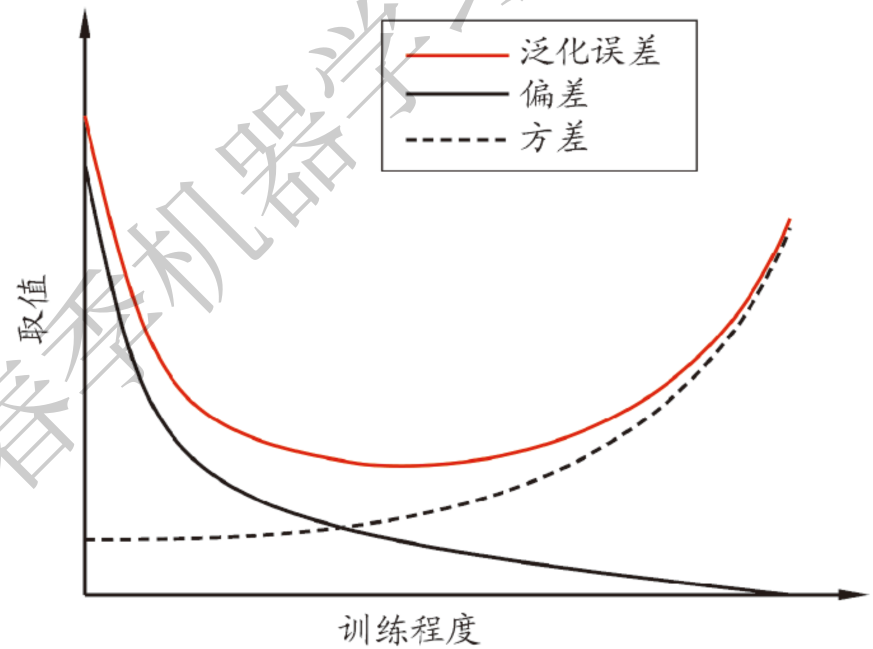


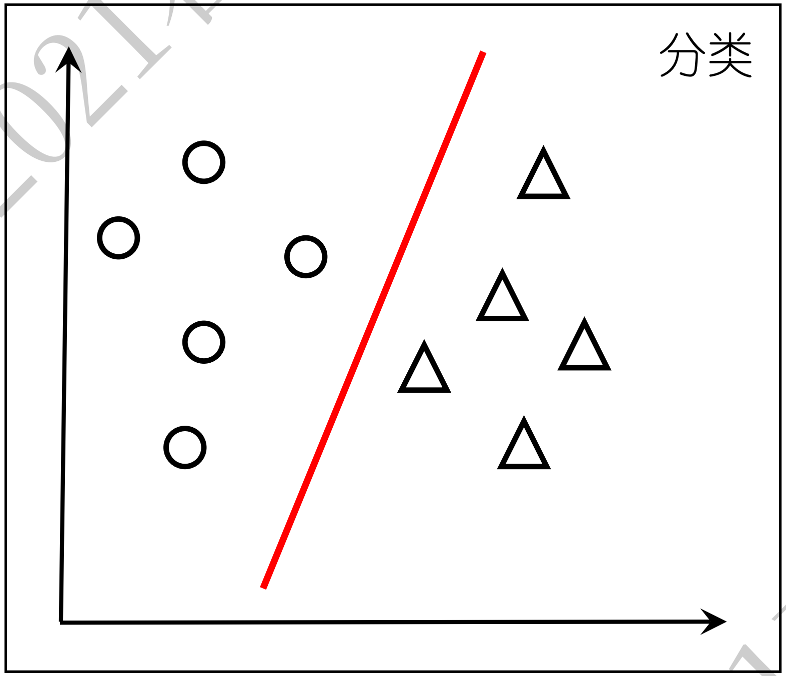
图 2.9 泛化误差与偏差、方差的关系示意图

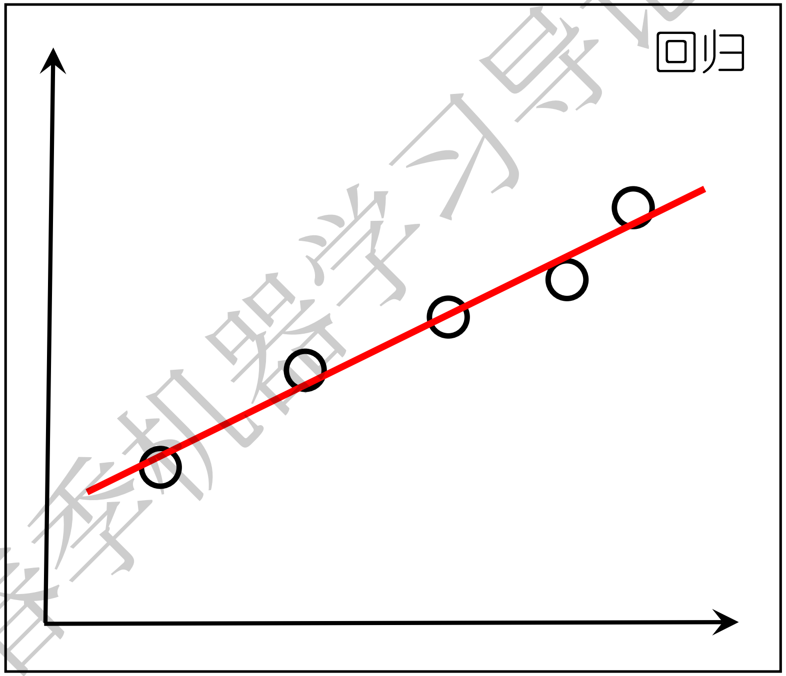
前往第三站......

三、线性模型

主讲教师：周志华

线性模型





线性模型(linear model)试图学得一个通过属性的线性组合来进行预测的函数

$$f(\boldsymbol{x}) = w_1 x_1 + w_2 x_2 + \ldots + w_d x_d + b$$

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

简单、基本、可理解性好

线性回归 (linear regression)

$$\boxed{f(x_i) = wx_i + b \text{ 使得 } f(x_i) \simeq y_i}$$

离散属性的处理：若有“序”(order)，则连续化；
否则，转化为 k 维向量

令均方误差最小化，有 $(w^*, b^*) = \underset{(w,b)}{\arg\min} \sum_{i=1}^{m} \left(f(x_i) - y_i\right)^2$

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

对 $E_{(w,b)} = \sum_{i=1}^{m} (y_i - wx_i - b)^2$ 进行最小二乘参数估计

线性回归

分别对 w 和 b 求导：

$$\frac{\partial E_{(w,b)}}{\partial w} = 2\left(w\sum_{i=1}^{m} x_i^2 - \sum_{i=1}^{m}(y_i - b)x_i\right)$$

$$\frac{\partial E_{(w,b)}}{\partial b} = 2\left(mb - \sum_{i=1}^{m}(y_i - wx_i)\right)$$

令导数为 0，得到闭式(closed-form)解：

$$w = \frac{\sum_{i=1}^{m} y_i(x_i - \bar{x})}{\sum_{i=1}^{m} x_i^2 - \frac{1}{m}\left(\sum_{i=1}^{m} x_i\right)^2} \qquad b = \frac{1}{m}\sum_{i=1}^{m}(y_i - wx_i)$$

多元(multi-variate)线性回归

$$\boxed{f(\boldsymbol{x}_i) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b \quad \text{使得} \quad f(\boldsymbol{x}_i) \simeq y_i}$$

$$\boldsymbol{x}_i = (x_{i1}; x_{i2}; \ldots; x_{id}) \qquad y_i \in \mathbb{R}$$

把 $\boldsymbol{w}$ 和 b 吸收入向量形式 $\hat{\boldsymbol{w}} = (\boldsymbol{w}; b)$，数据集表示为

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1d} & 1 \\ x_{21} & x_{22} & \cdots & x_{2d} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{md} & 1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}_1^{\mathrm{T}} & 1 \\ \boldsymbol{x}_2^{\mathrm{T}} & 1 \\ \vdots & \vdots \\ \boldsymbol{x}_m^{\mathrm{T}} & 1 \end{pmatrix} \qquad \boldsymbol{y} = (y_1; y_2; \ldots; y_m)$$

多元线性回归

同样采用最小二乘法求解，有

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

令 $E_{\hat{\boldsymbol{w}}} = (\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}})^{\mathrm{T}} (\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}})$，对 $\hat{\boldsymbol{w}}$ 求导：

$$\frac{\partial E_{\hat{\boldsymbol{w}}}}{\partial \hat{\boldsymbol{w}}} = 2\mathbf{X}^{\mathrm{T}} (\mathbf{X}\hat{\boldsymbol{w}} - \boldsymbol{y})$$ 令其为零可得 $\hat{\boldsymbol{w}}$

然而，麻烦来了：涉及矩阵求逆！

- 若 $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ 满秩或正定，则 $\hat{\boldsymbol{w}}^* = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1} \mathbf{X}^{\mathrm{T}}\boldsymbol{y}$
- 若 $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ 不满秩，则可解出多个 $\hat{\boldsymbol{w}}$

此时需求助于归纳偏好，或引入 正则化 (regularization) ⟶ 第6、11章

线性模型的变化

对于样例 $(\boldsymbol{x}, y)$, $y \in \mathbb{R}$, 若希望线性模型的预测值逼近真实标记,
则得到线性回归模型 $y = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$

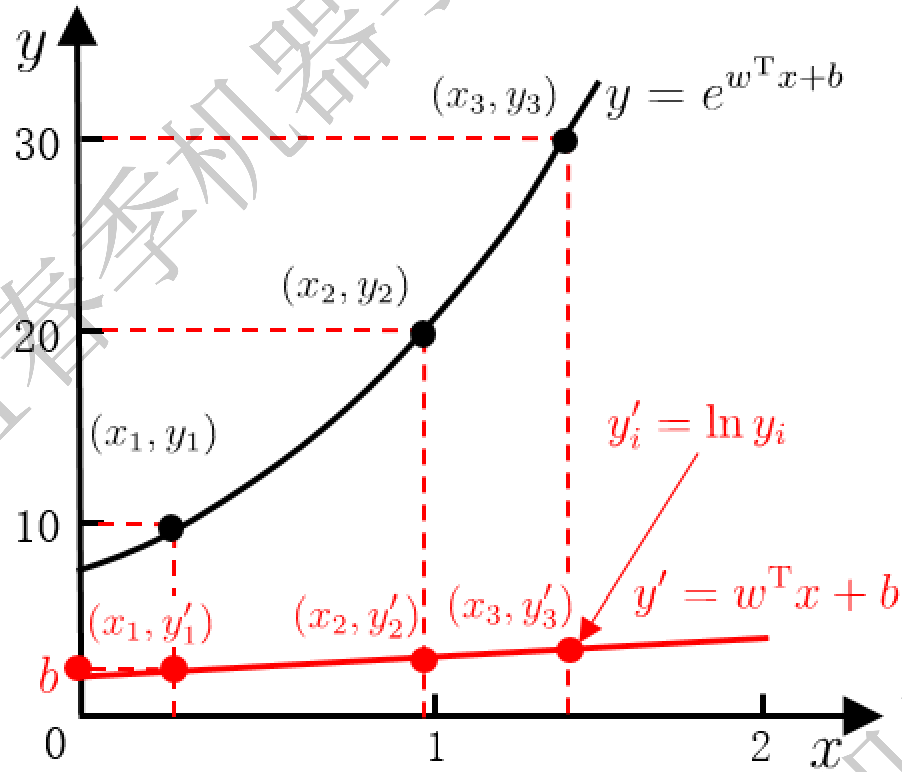
令预测值逼近 y 的衍生物？

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

则得到对数线性回归

(log-linear regression)

实际是在用 $e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}$ 逼近 y



广义(generalized)线性模型

一般形式： $y = g^{-1}\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b\right)$

单调可微的 联系函数 (link function)

令 $g(\cdot) = \ln(\cdot)$ 则得到 对数线性回归

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

… …

二分类任务

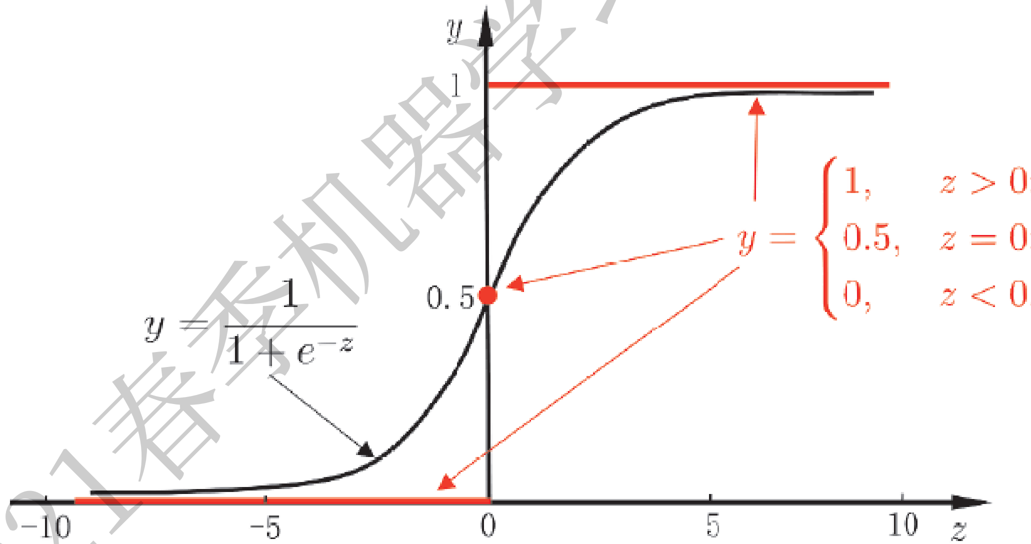
线性回归模型产生的实值输出 $z = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$

期望输出 $y \in \{0, 1\}$

找 z 和 y 的联系函数

理想的“单位阶跃函数” (unit-step function)

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



性质不好，需找“替代函数” (surrogate function)

常用
单调可微、任意阶可导

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

对数几率函数 (logistic function) 简称“对率函数”

注意：Logistic与“逻辑” 没有半毛钱关系！
1. Logistic 源自 Logit，不是Logic； 2. 实数值，并非“非0即1” 的逻辑值

对率回归

以对率函数为联系函数：

$$y = \frac{1}{1+e^{-z}} \quad \text{变为} \quad y = \frac{1}{1+e^{-(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b)}}$$

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

几率(odds)，反映了 $\boldsymbol{x}$ 作为正例的相对可能性

“对数几率”
(log odds, 亦称 logit)

“对数几率回归” (logistic regression)
简称 “对率回归”

- 无需事先假设数据分布
- 可得到 “类别” 的近似概率预测
- 可直接应用现有数值优化算法求取最优解

注意：它是
分类学习算法！

求解思路

若将 y 看作类后验概率估计 $p(y=1 \mid \boldsymbol{x})$，则

$$\ln \frac{y}{1-y} = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b \quad \text{可写为} \quad \ln \frac{p(y=1 \mid \boldsymbol{x})}{p(y=0 \mid \boldsymbol{x})} = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$$

于是，可使用“极大似然法” → 第7章

(maximum likelihood method)

给定数据集 $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$

最大化“对数似然”(log-likelihood)函数

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

求解思路

令 $\boldsymbol{\beta}=(\boldsymbol{w};b)$, $\hat{\boldsymbol{x}}=(\boldsymbol{x};1)$，则 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b$ 可简写为 $\boldsymbol{\beta}^{\mathrm{T}}\hat{\boldsymbol{x}}$

再令 $p_1(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta})=p(y=1\mid\hat{\boldsymbol{x}};\boldsymbol{\beta})=\dfrac{e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}}{1+e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}}$

$$p_0(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta})=p(y=0\mid\hat{\boldsymbol{x}};\boldsymbol{\beta})=1-p_1(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta})=\frac{1}{1+e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}}$$

则似然项可重写为 $p(y_i\mid\boldsymbol{x}_i;\boldsymbol{w}_i,b)=y_i p_1(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta})+(1-y_i)\,p_0(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta})$

于是，最大化似然函数 $\ell(\boldsymbol{w},b)=\displaystyle\sum_{i=1}^{m}\ln p(y_i\mid\boldsymbol{x}_i;\boldsymbol{w},b)$

等价为最小化 $\ell(\boldsymbol{\beta})=\displaystyle\sum_{i=1}^{m}\left(-y_i\boldsymbol{\beta}^{\mathrm{T}}\hat{\boldsymbol{x}}_i+\ln\left(1+e^{\boldsymbol{\beta}^{\mathrm{T}}\hat{\boldsymbol{x}}_i}\right)\right)$

高阶可导连续凸函数，可用经典的数值优化方法
如梯度下降法/牛顿法 [Boyd and Vandenberghe, 2004]