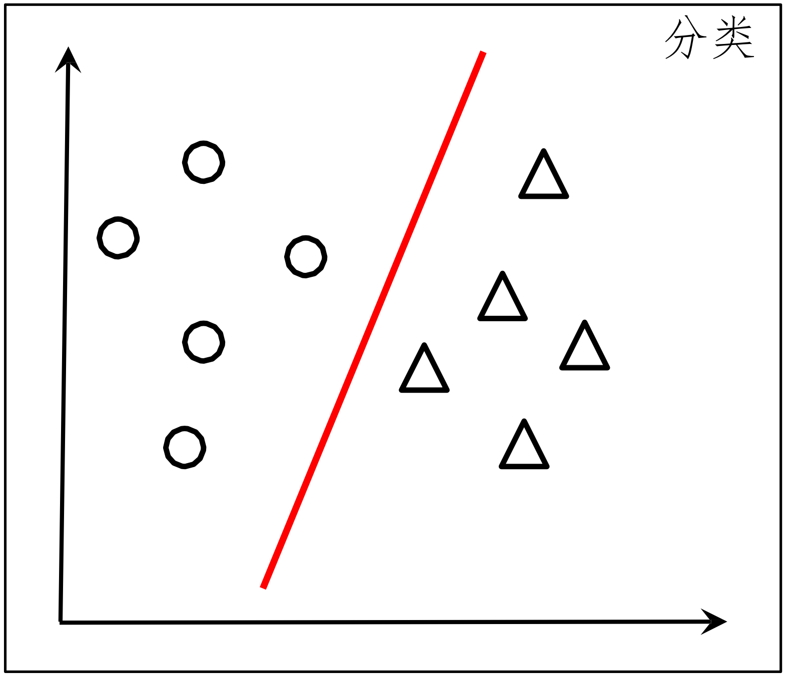
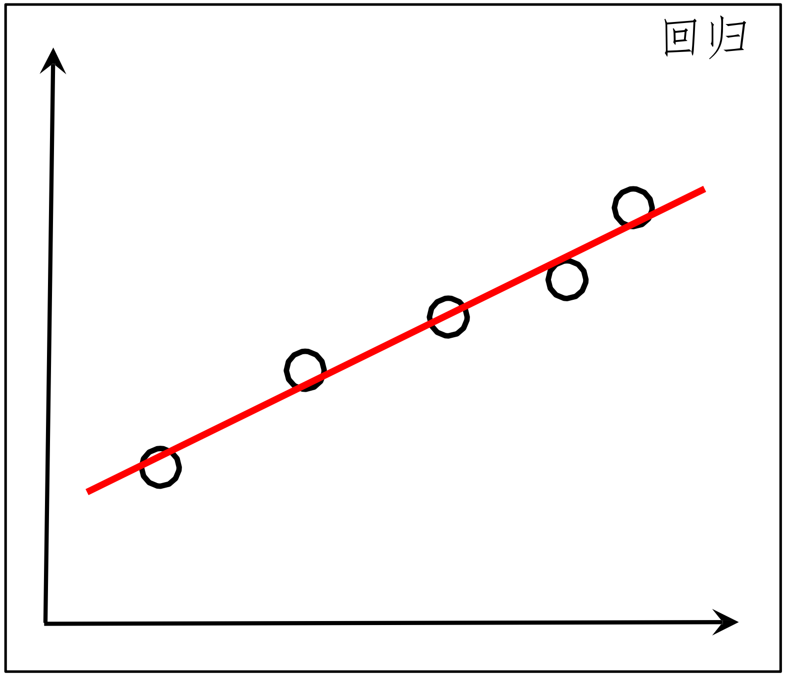
# 第三章：线性模型

# 目录

- 线性回归
  - 最小二乘法
  - 梯度下降
- 二分类任务
  - 对数几率回归 – Logistic Regression
  - 线性判别分析 – Linear Discriminate Analysis
- 多分类任务
  - 一对一
  - 一对其余
  - 多对多
- 类别不平衡问题

# 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

# 线性模型优点

- 形式简单、易于建模
- 可解释性
- 非线性模型的基础
  - 引入层级结构或高维映射

- 一个例子
  - 综合考虑色泽、根蒂和敲声来判断西瓜好不好
  - 其中根蒂的系数最大，表明根蒂最要紧；而敲声的系数比色泽大，说明敲声比色泽更重要

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

# 线性回归 (linear regression)

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

学得一个线性模型以尽可能准确地预测实值输出标记

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

Cost function

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

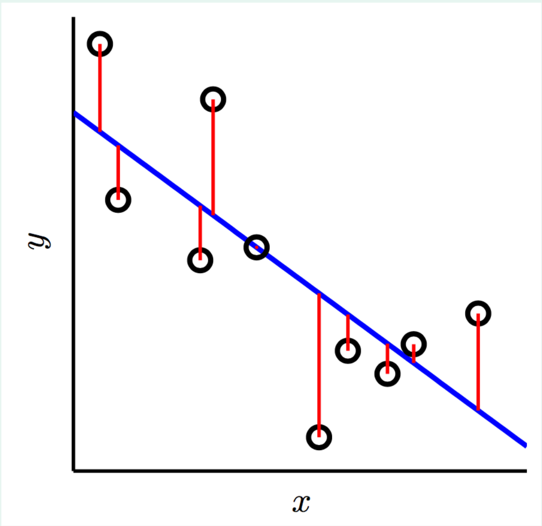
对 $E_{(w,b)} = \sum_{i=1}^{m} \left(y_i - wx_i - b\right)^2$ 进行最小二乘参数估计
（least square method）

# 线性回归 - 最小二乘法

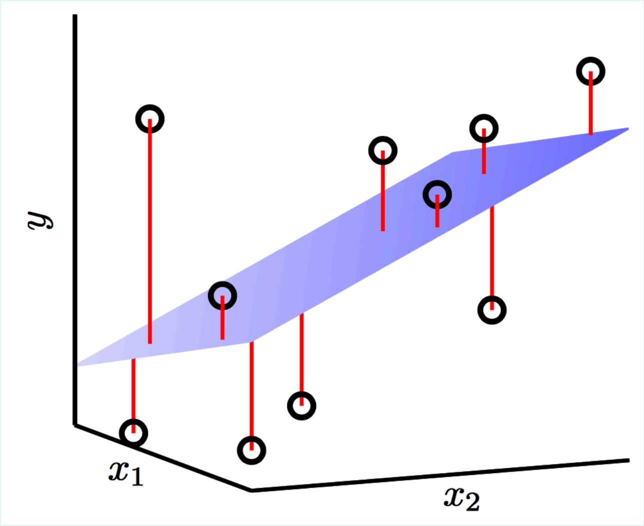
线性回归中，最小二乘法就是试图找到一条直线，使所有样本到直线上的欧式距离（vertical，y方向）之和最小：

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

$\mathbf{x} = (x) \in \mathbb{R}$




$\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$



# 线性回归 - 最小二乘法

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

分别对 $w$ 和 $b$ 求导（凸函数）：

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

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

令导数为 0, 得到闭式(closed-form)解：

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

# 多元(multi-variate)线性回归

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

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

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，目标变为：

$$\hat{\boldsymbol{w}}^* = \underset{\hat{\boldsymbol{w}}}{\arg\min} \left(\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}}\right)^{\mathrm{T}} \left(\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}}\right)$$

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

$$\frac{\partial E_{\hat{\boldsymbol{w}}}}{\partial \hat{\boldsymbol{w}}} = 2\mathbf{X}^{\mathrm{T}} \left(\mathbf{X}\hat{\boldsymbol{w}} - \boldsymbol{y}\right)$$

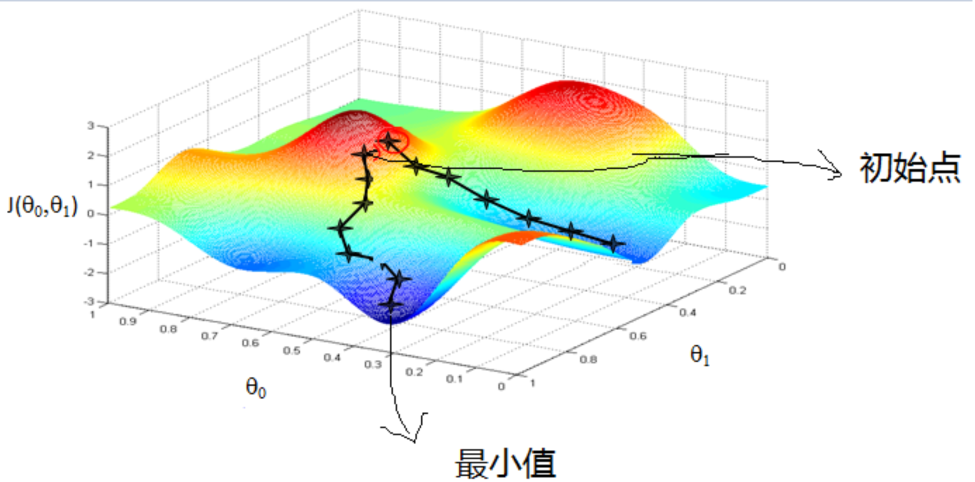
令其为零可得 $\hat{\boldsymbol{w}}$

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) $\longrightarrow$ 第6、11章

# 线性回归 - 梯度下降




对于线性回归，假设函数表示为：

$$h_\theta(x_1, x_2, \ldots x_n) = \theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n$$

其中$\theta_i$ (i = 0,1,2... n)为模型参数，$x_i$ (i = 0,1,2... n)为每个样本的n个特征值。这个表示可以简化，我们增加一个特征 $x_0 = 1$，有：

$$h_\theta(x_0, x_1, \ldots x_n) = \sum_{i=0}^{n} \theta_i x_i$$

损失函数为：

$$J(\theta_0, \theta_1 \ldots, \theta_n) = \sum_{i=0}^{m} (h_\theta(x_0, x_1, \ldots x_n) - y_i)^2$$

# 线性回归 - 梯度下降

$$J(\theta_0,\theta_1\ldots,\theta_n)=\sum_{i=0}^{m}(h_\theta(x_0,x_1,\ldots x_n)-y_i)^2 \quad h_\theta(x_0,x_1,\ldots x_n)=\sum_{i=0}^{n}\theta_i x_i$$

**算法过程：**

1. 确定当前位置的损失函数的梯度，对于$\theta_i$， 其梯度表达式如下：

$$\frac{\partial}{\partial\theta_i}J(\theta_0,\theta_1\ldots,\theta_n)=\frac{1}{m}\sum_{j=0}^{m}(h_\theta(x_0^j,x_1^j,\ldots x_n^j)-y_j)x_i^j$$

2.用步长乘以损失函数的梯度，得到当前位置下降的距离， 即

$\alpha\frac{\partial}{\partial\theta_i}J(\theta_0,\theta_1\ldots,\theta_n)$ 对应于前面图中例子中的某一步。

3.确定是否所有的$\theta_i$, 梯度下降的距离都小于 ε， 如果小于 ε 则算法终止，当前所有的$\theta_i$ (i=0,1,...n)即为最终结果。否则进入步骤4.

4.更新所有的θ，对于$\theta_i$，其更新表达式如下。更新完毕后继续转入步骤1.

$$\theta_i=\theta_i-\alpha\frac{\partial}{\partial\theta_i}J(\theta_0,\theta_1\ldots,\theta_n)=\theta_i-\alpha\frac{1}{m}\sum_{j=0}^{m}(h_\theta(x_0^j,x_1^j,\ldots x_n^j)-y_j)x_i^j$$

# 线性回归-梯度下降与最小二乘

## 相同

1.本质相同：两种方法都是在给定已知数据（independent & dependent variables）的前提下对dependent variables算出一个一般性的估值函数。然后对给定新数据的dependent variables进行估算。

2.目标相同：都是在已知数据的框架内，使得估算值与实际值的总平方差尽量更小。

## 不同

1. 实现方法和结果不同：最小二乘法是直接求导找出**全局最小**，是非迭代法。而梯度下降法是一种迭代法，先给定一个，然后向下降最快的方向调整，在若干次迭代之后找到**局部最小**。

2. 梯度下降法的缺点是到最小点的时候收敛速度变慢，并且对初始点的选择极为敏感，其改进大多是在这两方面下功夫。

## 场景：

1. 如果样本量不算很大，且存在解析解，最小二乘法比起梯度下降法要有优势，计算速度很快。

2. 但是如果样本量很大，用最小二乘法由于需要求一个超级大的逆矩阵，这时就很难或者很慢才能求解解析解了，使用迭代的梯度下降法比较有优势。

# 线性模型的变化

对于样例 $(\boldsymbol{x}, y)$, $y \in \mathbb{R}$, 则得到线性回归模型

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

若希望线性模型的预测值逼近真实标记，

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

$$\Updownarrow$$

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

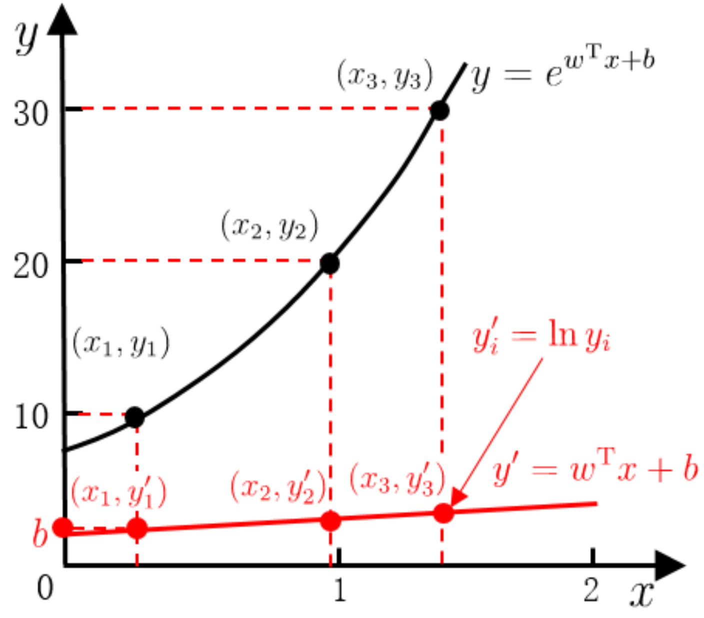
令预测值逼近 y 的衍生物?

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




**输出标记的对数为线性模型逼近的目标**

# 广义(generalized)线性模型

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

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

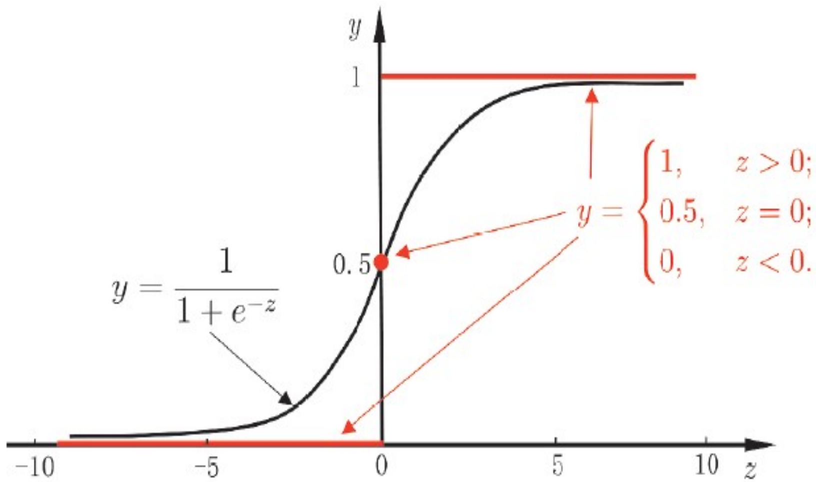
线性回归模型产生的实值输出 $z = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$

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)

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




常用 单调可微、任意阶可导

性质不好，
需找“替代函数”
(surrogate function)

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

对数几率函数
(logistic function)
简称“对率函数”

# 对率回归

以对率函数为联系函数:

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

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)

利用已知的样本结果，反推最有可能（最大概率）导致这样结果的参数值。

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

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

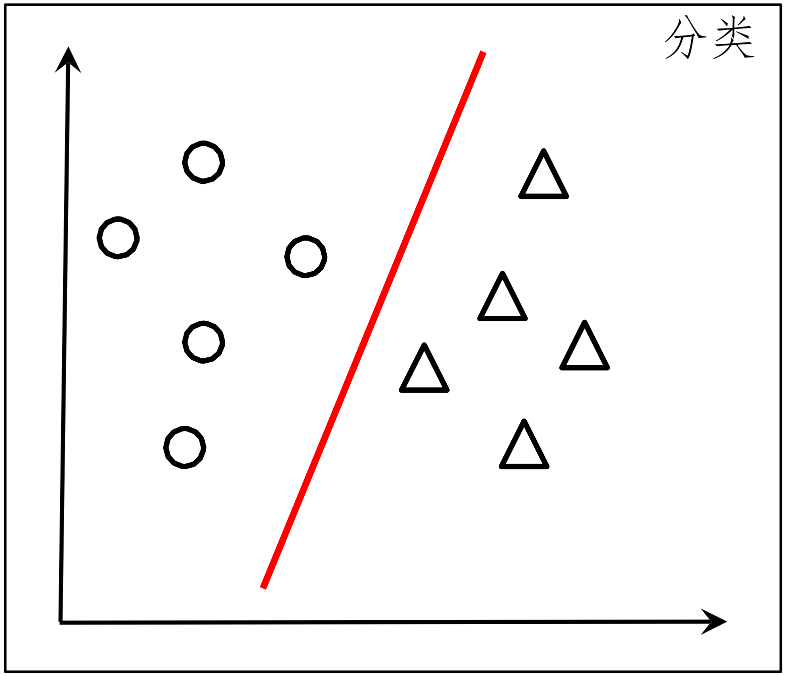
则似然项可重写为 $p(y_i \mid \boldsymbol{x}_i; \boldsymbol{w}_i, b) = y_i p_1(\hat{\boldsymbol{x}}_i; \boldsymbol{\beta}) + (1 - y_i) p_0(\hat{\boldsymbol{x}}_i; \boldsymbol{\beta})$

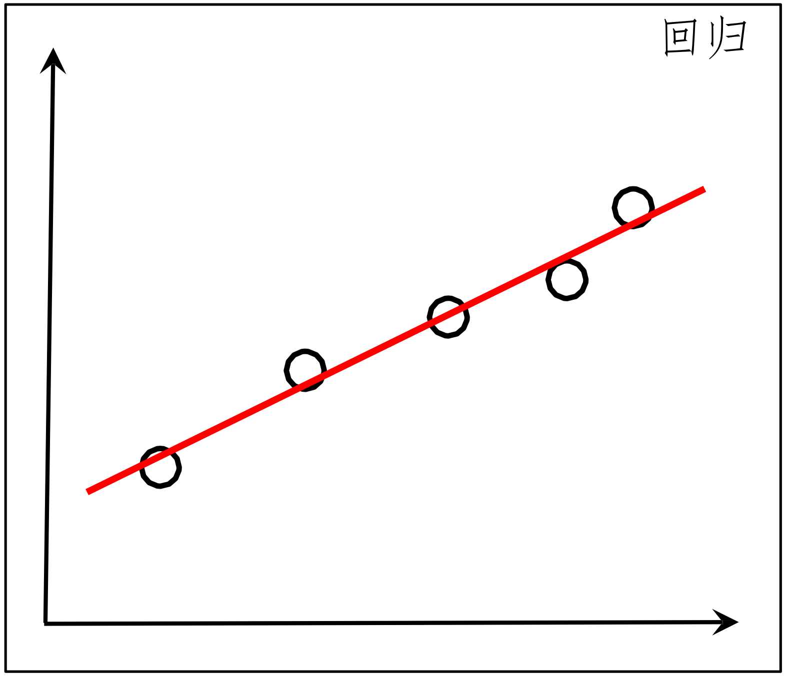
于是，最大化似然函数 $\ell(\boldsymbol{w}, b) = \sum_{i=1}^{m} \ln p(y_i \mid \boldsymbol{x}_i; \boldsymbol{w}, b)$

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

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]

# 线性模型做“分类”



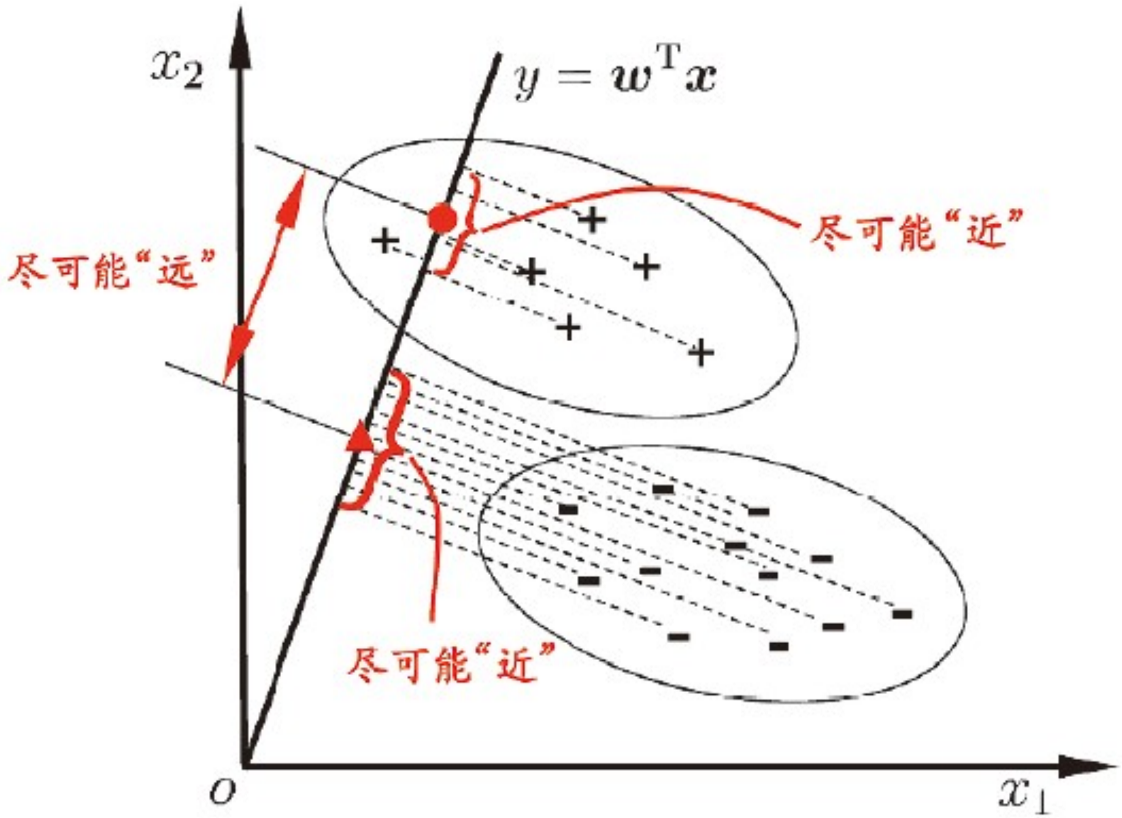




广义线性模型；
通过“联系函数”

如何“直接”做分类？

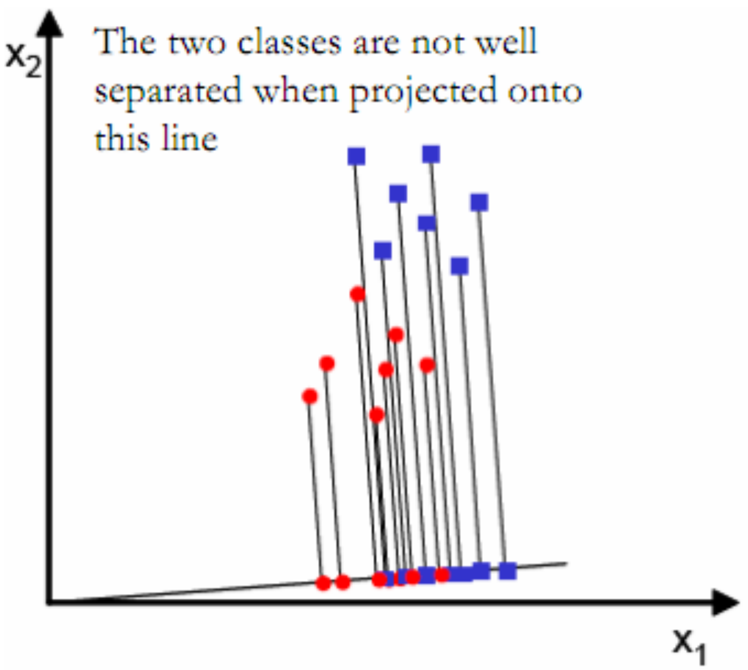
例如，对率回归

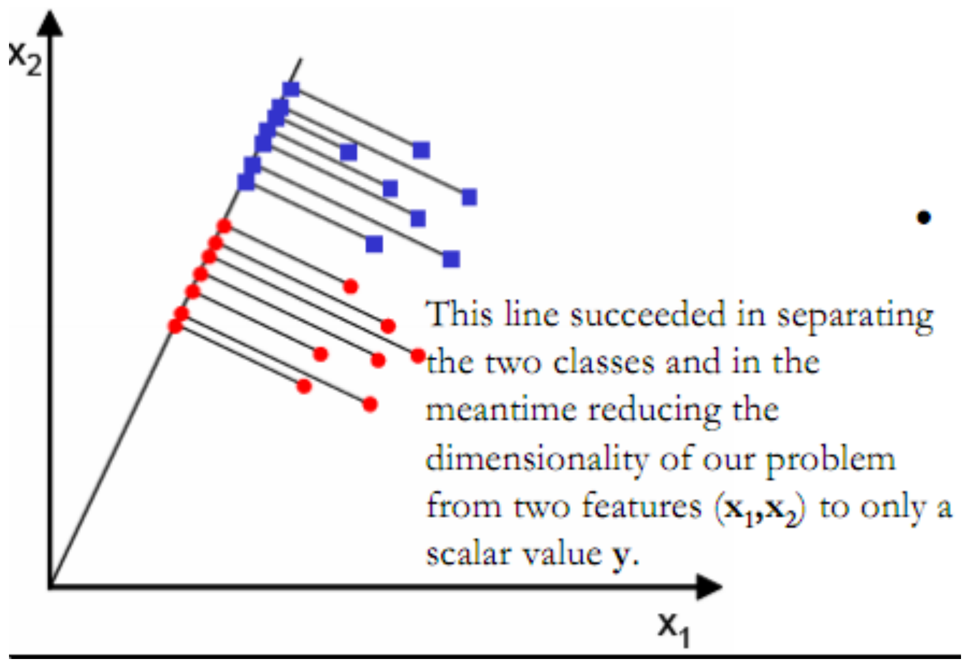
# 线性判别分析 (Linear Discriminant Analysis)



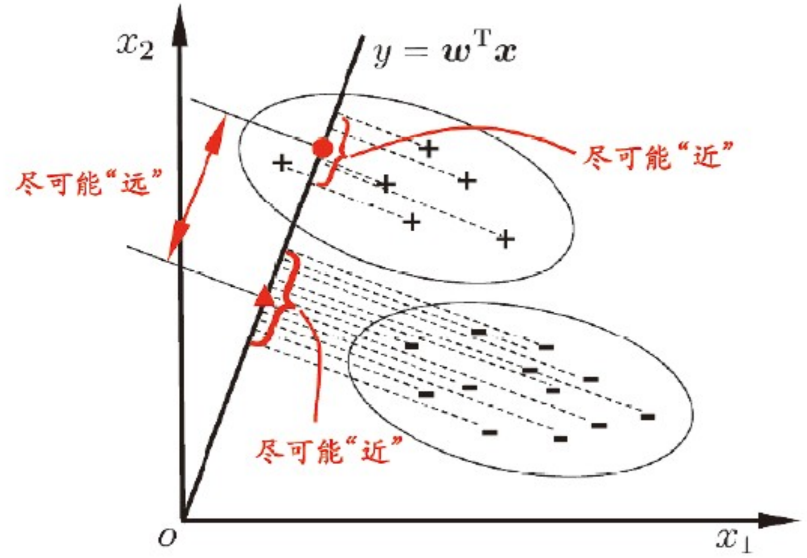

由于将样例投影到一条直线（低维空间），因此也被视为一种“监督降维”技术 降维 ⟶ 第10章

# 线性判别分析 (Linear Discriminant Analysis)



# LDA的目标




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

第 $i$ 类示例的集合 $X_i$

第 $i$ 类示例的均值向量 $\boldsymbol{\mu}_i$

第 $i$ 类示例的协方差矩阵 $\boldsymbol{\Sigma}_i$

两类样本的中心在直线上的投影：$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_0$ 和 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_1$

两类样本的协方差：$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_0\boldsymbol{w}$ 和 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_1\boldsymbol{w}$

同类样例的投影点尽可能接近 → $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_0\boldsymbol{w} + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_1\boldsymbol{w}$ 尽可能小

异类样例的投影点尽可能远离 → $\|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_0 - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_1\|_2^2$ 尽可能大

于是，最大化

$$J = \frac{\|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_0 - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_1\|_2^2}{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_0\boldsymbol{w} + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_1\boldsymbol{w}} = \frac{\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{\Sigma}_0 + \boldsymbol{\Sigma}_1)\boldsymbol{w}}$$

# LDA的目标

$$J=\frac{\left\|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_0-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_1\right\|_2^2}{\boldsymbol{w}^{\mathrm{T}}\Sigma_0\boldsymbol{w}+\boldsymbol{w}^{\mathrm{T}}\Sigma_1\boldsymbol{w}}=\frac{\boldsymbol{w}^{\mathrm{T}}\boxed{\left(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1\right)\left(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1\right)^{\mathrm{T}}}\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}\boxed{\left(\Sigma_0+\Sigma_1\right)}\boldsymbol{w}}$$

类内散度矩阵 (within-class scatter matrix)

$$\begin{aligned}\mathbf{S}_w&=\Sigma_0+\Sigma_1\\&=\sum_{\boldsymbol{x}\in X_0}\left(\boldsymbol{x}-\boldsymbol{\mu}_0\right)\left(\boldsymbol{x}-\boldsymbol{\mu}_0\right)^{\mathrm{T}}+\sum_{\boldsymbol{x}\in X_1}\left(\boldsymbol{x}-\boldsymbol{\mu}_1\right)\left(\boldsymbol{x}-\boldsymbol{\mu}_1\right)^{\mathrm{T}}\end{aligned}$$

类间散度矩阵 (between-class scatter matrix)

$$\mathbf{S}_b=\left(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1\right)\left(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1\right)^{\mathrm{T}}$$

LDA的目标：最大化广义瑞利商 (generalized Rayleigh quotient)

$$J=\frac{\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_b\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w}}$$

$\boldsymbol{w}$ 成倍缩放不影响 $J$ 值

仅考虑方向

## 求解思路

$$J = \frac{\boldsymbol{w}^{\mathrm{T}}\boxed{\mathbf{S}_b}\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}\boxed{\mathbf{S}_w}\boldsymbol{w}}$$

**最大化广义瑞利商**

令 $\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w} = 1$（**归一化**），最大化广义瑞利商等价形式为

将有d个变量与k个约束条件的最优化问题转换为d+k个变量的无约束优化问题

$$\min_{\boldsymbol{w}} \; -\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_b\boldsymbol{w}$$

$$\text{s.t. } \boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w} = 1$$

运用拉格朗日乘子法，有 $\mathbf{S}_b\boldsymbol{w} = \lambda\mathbf{S}_w\boldsymbol{w}$ 附录B

$\mathbf{S}_b\boldsymbol{w}$ 的方向恒为 $\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1$，不妨令 $\mathbf{S}_b\boldsymbol{w} = \lambda(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)$

于是 $\boldsymbol{w} = \mathbf{S}_w^{-1}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)$ 只需要求出原始样本的均值和方差就可以求出最佳的方向 $\boldsymbol{w}$

实践中通常是进行奇异值分解 $\mathbf{S}_w = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$ 附录 A

然后 $\mathbf{S}_w^{-1} = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^{\mathrm{T}}$

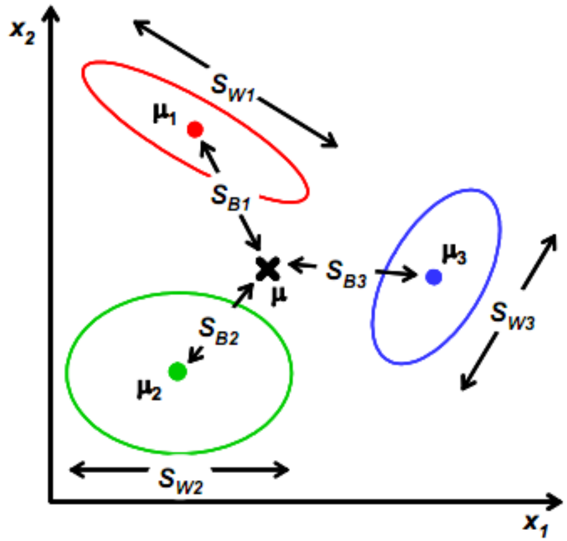
# 推广到多类

假定有 $N$ 个类

- 全局散度矩阵 $\mathbf{S}_t = \mathbf{S}_b + \mathbf{S}_w = \sum_{i=1}^{m} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T$

- 类内散度矩阵（每个类别的散度矩阵之和） $\mathbf{S}_w = \sum_{i=1}^{N} \mathbf{S}_{w_i}$ $\quad \mathbf{S}_{w_i} = \sum_{\boldsymbol{x} \in X_i} (\boldsymbol{x} - \boldsymbol{\mu}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)^T$

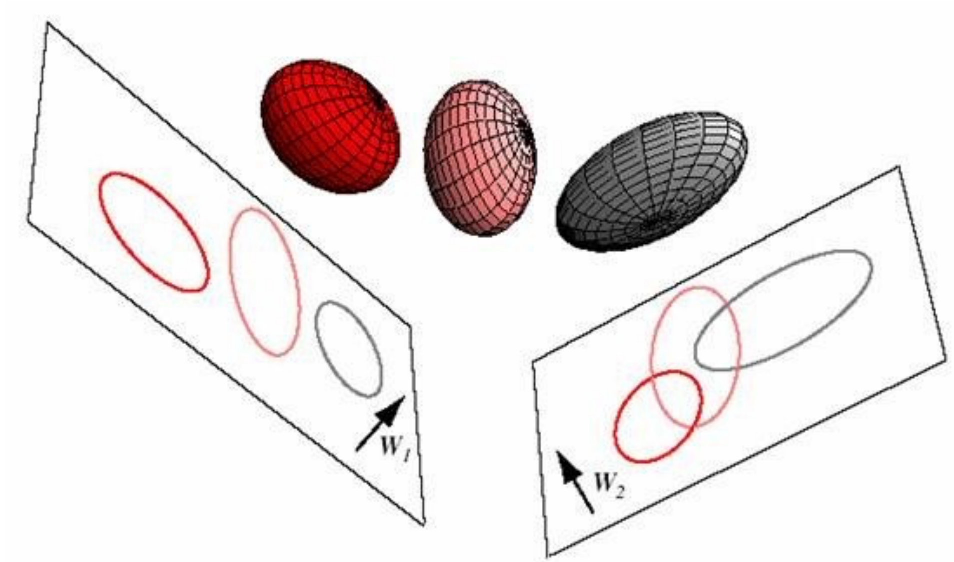
- 类间散度矩阵 $\mathbf{S}_b = \mathbf{S}_t - \mathbf{S}_w = \sum_{i=1}^{N} m_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T$




原来度量的是两个均值点的散列情况，现在度量的是每类均值点相对于样本中心的散列情况。

# 推广到多类

多分类LDA有多种实现方法：采用 $\mathbf{S}_b, \mathbf{S}_w, \mathbf{S}_t$ 中的任何两个

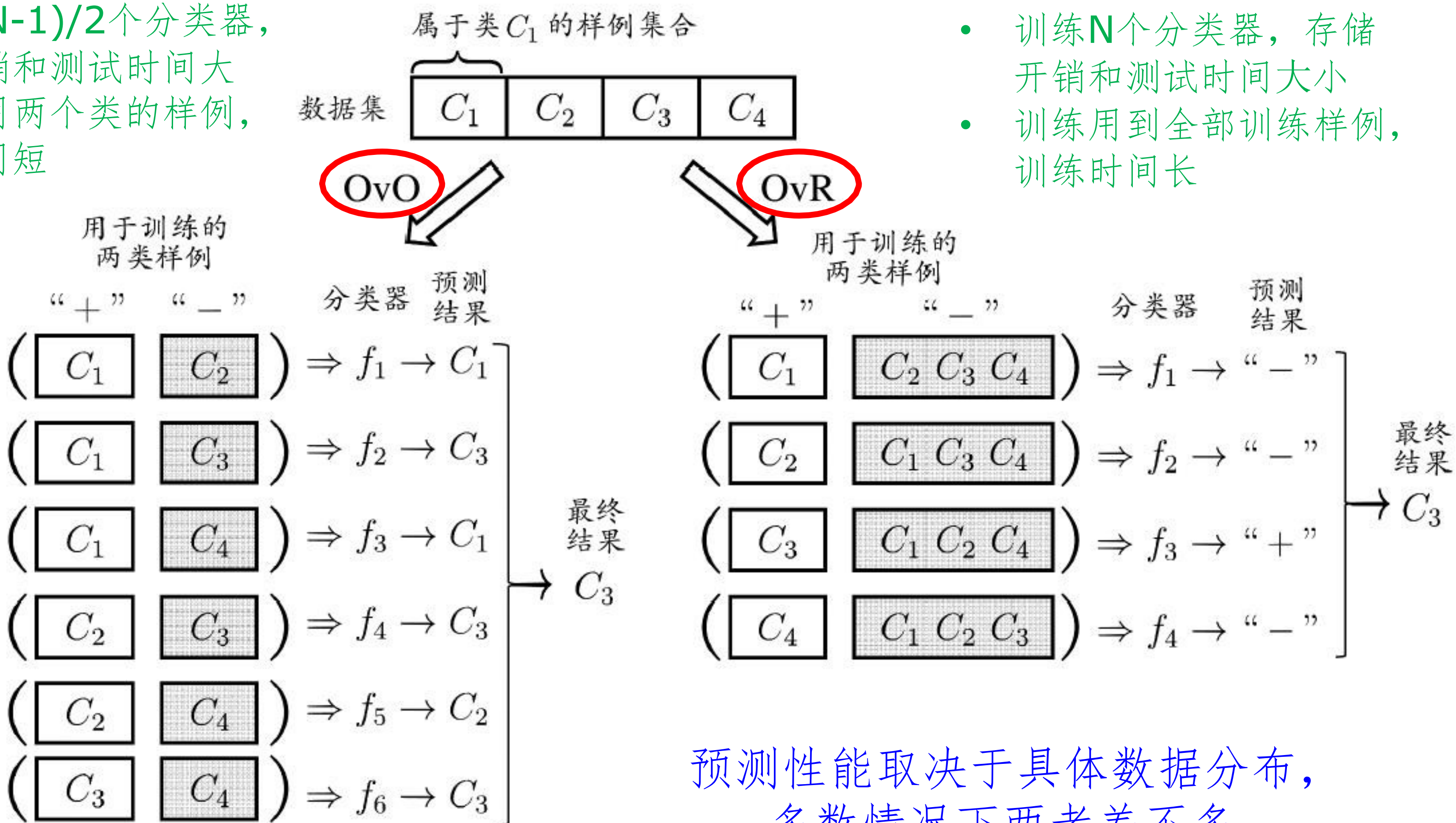


# 多分类学习

拆解法：将一个多分类任务拆分为若干个二分类任务求解

- 训练N(N-1)/2个分类器，存储开销和测试时间大
- 训练只用两个类的样例，训练时间短

- 训练N个分类器，存储开销和测试时间大小
- 训练用到全部训练样例，训练时间长




预测性能取决于具体数据分布，多数情况下两者差不多

# 纠错输出码 (ECOC)

多对多(Many vs Many, MvM)：将若干类作为正类，若干类作为反类

一种常见方法：**纠错输出码 (Error Correcting Output Code)**

**编码：** 对 N 个类别做 M 次划分，每次将一部分类别划为正类，一部分划为反类

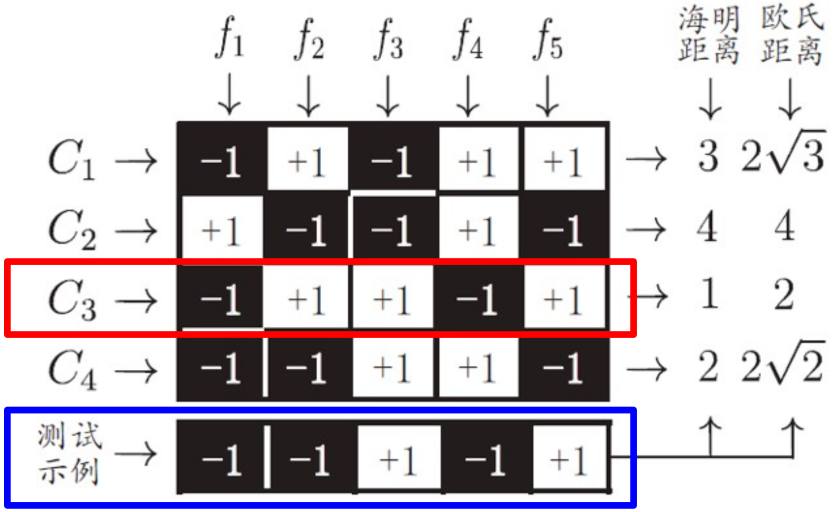
M 个二类任务；
(原)每类对应一个长为 M 的编码

距离最小的类为最终结果

**解码：** 测试样本交给 M 个分类器预测

长为 M 的预测结果编码

# 纠错输出码




(a) 二元ECOC码

[Dietterich and Bakiri,1995]

- ECOC编码对分类器错误有一定容忍和修正能力，编码越长、纠错能力越强
- 对同等长度的编码，理论上来说，任意两个类别之间的编码距离越远，则纠错能力越强

# 类别不平衡 (class-imbalance)

不同类别的样本比例相差很大；“小类”往往更重要

基本思路：（假定正类较少，反类较多）

若 $\frac{y}{1-y} > 1$ 则 预测为正例. $\Rightarrow$ 若 $\frac{y}{1-y} > \frac{m^+}{m^-}$ 则 预测为正例.

基本策略
—— “再缩放” (rescaling)：

$$\frac{y'}{1-y'} = \frac{y}{1-y} \times \frac{m^-}{m^+}$$

**然而，精确估计 $m^-/m^+$ 通常很困难！**

常见类别不平衡学习方法：

- 过采样 (oversampling)
  **增加一些正例**
- 欠采样 (undersampling)
  **去除一些反例**
- 阈值移动 (threshold-moving)

# 总结

- 线性回归
  - 最小二乘法（最小化均方误差）
- 二分类任务
  - 对数几率回归
    - 单位阶跃函数（替代函数）、对数几率函数、极大似然法
  - 线性判别分析
    - 最大化广义瑞利商，拉格朗日方法
- 多分类学习
  - 一对一
  - 一对其余
  - 多对多
    - 纠错输出码
- 类别不平衡问题
  - 基本策略：再缩放