北京郵電大學
Beijing University of Posts and Telecommunications

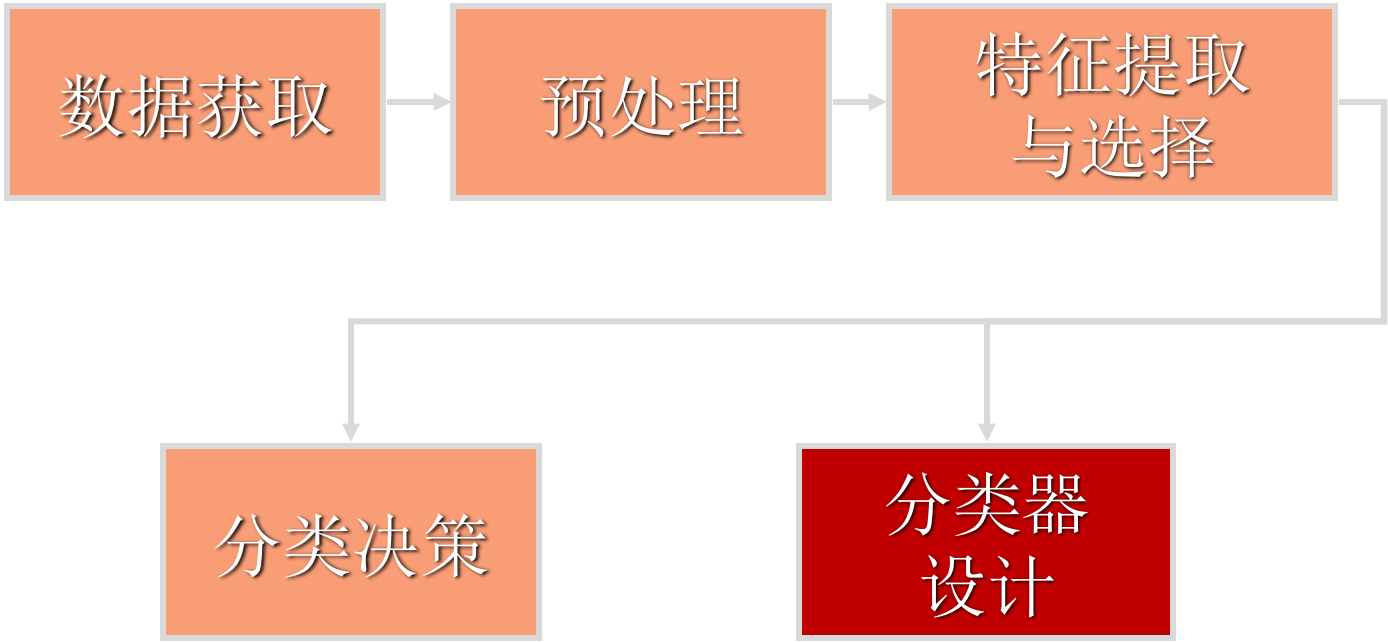
线性模型

戚 琦

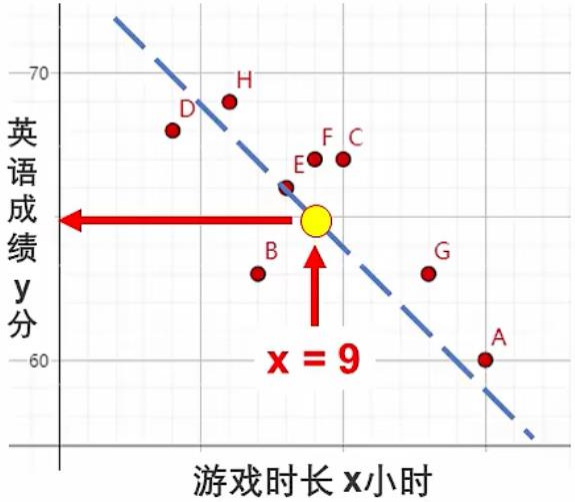
网络与交换技术国家重点实验室　网络智能研究中心　科研楼511

qiqi8266@bupt.edu.cn　　13466759972

回顾：监督学习



一个简单例子



同学编号	1	2	3	4	5	6	7	8	小明
英语成绩 y 分	60	63	67	68	66	67	63	69	?
游戏时长 x 小时	15	7	10	4	8	9	13	6	9

小明 把试卷 藏到床底下

已知 小明的 x=9

预测 小明的 y

假如已经求出这条直线方程，把x=9代入到方程中去，可预测那张试卷的分数。

求出线性相关的这条直线，可以精确的描述线性规律，进而对未知的数据进行预测。

大纲

- 基本概念
- 线性回归
- 对数几率回归
- 线性判别分析
- 多分类学习

周志华，《机器学习》

线性：变量之间具有简单的形式

$$y = Wx$$

W 是模型参数

有些问题x与y不存在直接线性关系，但可以通过某些变换$\phi(\cdot)$建立这种关系

$$y = W\phi(x)$$

线性模型

- 多种成分的组合
- 预测变量与属性变量之间有相关关系；
- 模型简单、易于建模，蕴含机器学习中一些重要的基本思想
- 功能强大的非线性模型，通过在线性模型基础上引入层级结构或高维映射而得
- 线性回归是基本形式，由此可以获得许多其他有用的学习算法

基本形式

- 机器学习问题：设假设空间H是全体函数集合，已知数据集合：
$\mathcal{D}=\{(x_1, T_1), (x_2, T_2), \ldots, (x_m, T_m)\}$，求$f \in H$，使得 $T_i \approx f(x_i)$，$i=1,2,\ldots,m$

- 给定由d个属性描述的对象 $x=(x_1, x_2, \ldots, x_d)$，预测值y，假设y与x相关，求y与x的关系：y=f(x)

线性模型：

$$f(x)=w_1x_1+w_2x_2+w_3x_3+w_4x_4+b$$
$$=W^Tx+b$$

$$where: W=\begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{pmatrix}; x=\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

例：基站负载预测，已知：
时间 x_1=
位置 x_2=
用户量x_3=
输出基站的负载 T?

$T=f(x_1, x_2, x_3)$

只要求得**参数*w*、*b***就可以由数据(经验)获得预测模型

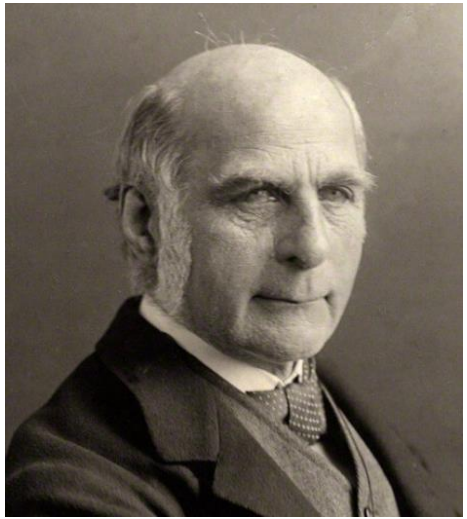
*w*直观表达了个属性在预测中的重要性

大纲

- 基本概念
- 线性回归
- 对数几率回归
- 线性判别分析
- 多分类学习

线性回归

- 线性回归是统计学中的一种回归分析方法
- 英国遗传学家、统计学家弗朗斯·西高尔顿1877年发表关于种子的研究成果，指出了一种回归到平均值（Regression Towards the Mean）的现象
- 统计学继承了这个词汇



弗朗斯·西高尔顿

单变量线性回归

已知-数据集合(D)：

$$D=\{(x_1,y_1),(x_2,y_2),\mathrm{L}\ ,(x_m,y_m)\}$$

$$where: x_i=\begin{pmatrix} x_{i1} \\ x_{i2} \\ \mathrm{M} \\ x_{id} \end{pmatrix}\in R^d; y_i\in R$$

假设空间(H)：

$$H=\left\{f\,\middle|\,f(x)=W^Tx+b, W, x\in R^d, b\in R\right\}$$

求：*W*和 *b*

样本的属性值只有一个

$$f(x_i)=Wx_i+b \Rightarrow \boxed{f(x_i)\approx y_i}$$

如何衡量差别？

性能评价-均方误差 几何意义：欧氏距离

$$(W^*,b^*)=Arg\min_{(w,b)}\sum_{i=1}^{m}(f(x_i)-y_i)^2$$

$$=Arg\min_{(w,b)}\sum_{i=1}^{m}(W\ x_i+b-y_i)^2$$

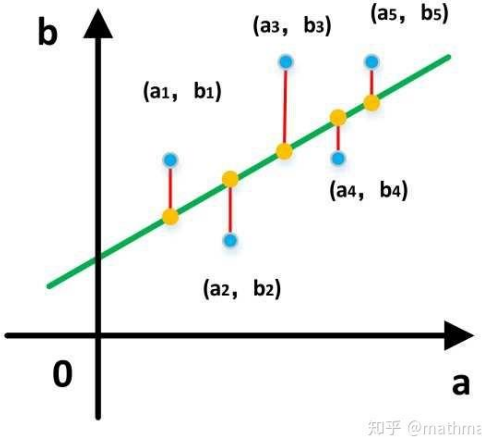
每个样本

$$E(W,b)=\sum_{i=1}^{m}(f(x_i)-y_i)^2$$

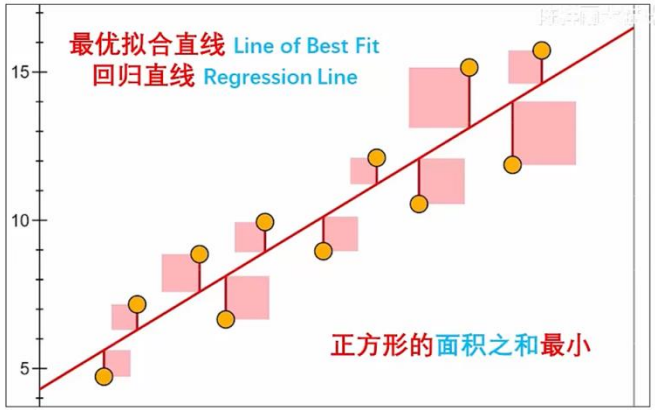
m个样本到这条直线上的距离**欧氏距离之和最小**

*最小二乘法

- 二乘是平方的意思（Squares）
- 通过最小化误差的平方和寻找数据的最佳函数匹配。



竖直距离：直接相减，方便计算



以每个"竖直”的线段，做出一个正方形。求出每个正方形的面积，然后相加，得到一个面积之和。使所有这些正方形的面积之和最小的那一条直线，就叫做"最优拟合"直线，"Line of Best Fit",或者叫做“回归"直线，“Regression Line”。

*最小二乘法

- 利用最小二乘法可以简便地求得未知的数据，并使得这些求得的数据与实际数据之间误差的平方和为最小
- 可解决曲线拟合问题

斜率　截距

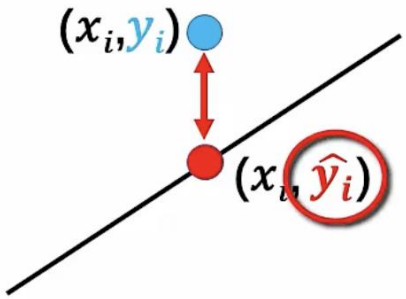
$$y = ax + b$$

⇩

权重　偏置

$$y = wx + b$$

样本：$(x_1, y_1), (x_2, y_2), \dots, (x_i, y_i), \dots, (x_n, y_n)$



竖直距离

y_i − $\hat{y}_i$

观测值 Observed value　预测值 Predicted value

$= y_i - (wx_i + b)$

= 余数

Residual

那么，这两个点之间的竖直距离就是y_i -$\hat{y}_i$，也就是$y_i - (wx_i + b)$。这里引入概念，直线上的点的纵坐标$\hat{y}_i$，叫做预测值，英语叫做Predicted Value；直线外的样本中的数据点的纵坐标y_i，叫做实际观察到的值，或观测值，Observed Value。$\hat{y}_i$和y_i的差值，也就是小正方形的边长，叫做“余数”，英语叫做Residual。竖直距离就是余数，等于观测值减去预测值。

最小二乘法

参数估计-最小二乘法

$$\sum_{i=1}^{m}\left(W^T x_i + b - y_i\right)^2$$

由： $\dfrac{\partial E(W,b)}{\partial W} = 0; \dfrac{\partial E(W,b)}{\partial b} = 0$

得：

$$\frac{\partial E(W,b)}{\partial W} = 2\left(W\sum_{i=1}^{m} x_i^2 - \sum_{i=1}^{m}(y_i - b)x_i\right) = 0;$$

$$\frac{\partial E(W,b)}{\partial b} = 2\left(mb - \sum_{i=1}^{m}\left(y_i - Wx_i\right)\right) = 0$$

先求出b，带入求W

$$W = \frac{\sum_{i=1}^{m} y_i(x_i - \bar{x})}{\sum_{i=1}^{m} x_i^2 - \frac{1}{m}\left(\sum_{i=1}^{m} x_i\right)^2};$$

$$b = \frac{1}{m}\sum_{i=1}^{m}\left(y_i - Wx_i\right)$$

$$y = f(x) = Wx + b$$

其中： $\bar{x} = \dfrac{1}{m}\sum_{i=1}^{m} x_i$

多元线性回归

m个样本每个样本有d个属性 $x \in R^d; W = \begin{pmatrix} w_1 & w_2 & \mathrm{L} & w_d \end{pmatrix}^T; b \in R$

为了方便引入符号：

$$X = \begin{pmatrix} x_{11} & x_{12} & \mathrm{L} & x_{1d} & 1 \\ x_{21} & x_{22} & \mathrm{L} & x_{2d} & 1 \\ \mathrm{M} & \mathrm{M} & \mathrm{O} & \mathrm{M} & \mathrm{M} \\ x_{m1} & x_{m2} & \mathrm{L} & x_{md} & 1 \end{pmatrix} = \begin{pmatrix} x_1^T & 1 \\ x_2^T & 1 \\ \mathrm{M} & \mathrm{M} \\ x_m^T & 1 \end{pmatrix} \quad \hat{W} = \begin{pmatrix} w_1 \\ \mathrm{M} \\ w_d \\ b \end{pmatrix} = \begin{pmatrix} W \\ b \end{pmatrix} \quad y = \begin{pmatrix} y_1 \\ y_2 \\ \mathrm{M} \\ y_m \end{pmatrix}$$

$m(d+1)$维矩阵

相乘

均方误差（MSE）： $E(\hat{W}) = \left(y - X\hat{W}\right)^T \left(y - X\hat{W}\right)$ y向量和$X\hat{W}$向量的均方误差

问题描述： $\hat{W}^* = \arg\min_{\hat{W}} E(\hat{W}) = \arg\min_{\hat{W}} \left(y - X\hat{W}\right)^T \left(y - X\hat{W}\right)$

多元线性回归

$$\frac{\partial E(\vec{w})}{\partial \vec{w}} = \frac{\partial}{\partial \vec{w}}(y - X\vec{w})^T(y - X\vec{w})$$

$$= \frac{\partial}{\partial \vec{w}}(y^T y - y^T X\vec{w} - \vec{w}^T X^T y + \vec{w}^T X^T X\vec{w})$$

矩阵求偏导

$$= -2X^T y + 2X^T X\vec{w} = 2X^T(X\vec{w} - y) = 0$$

$$\Longrightarrow \quad \hat{w}^* = \boxed{(X^T X)^{-1}} X^T y$$

讨论 $X^T X$ 的情况：

$(X^T X)^{-1}$ 是 $X^T X$ 的逆矩阵

$(X^T X)^{-1}$ **存在**

$$f(\hat{x}_i) = \hat{x}_i^T (X^T X)^{-1} X^T y$$

$(X^T X)^{-1}$ **不存在**

样本的属性很多，多于样本个数

有多个解$\hat{w}$，使得均方误差最小

广义线性模型

线性回归模型 $y = w^T x + b$

假如样本输出不是一条线，而是变成指数形式，就可以将输出标记的对数作为线性模型逼近目标，称为对数线性回归。

输出标记在指数形式上变化

$$\ln y = w^T x + b$$

$$\Rightarrow y = e^{w^T x + b}$$

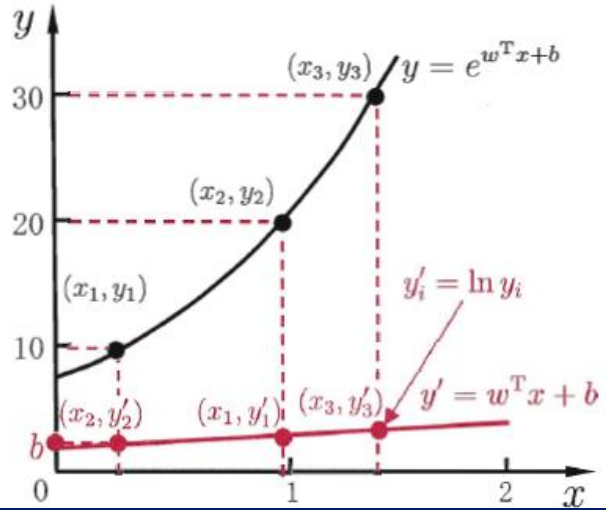
联系函数 $g: Y \rightarrow Y'$ 单调可微

$$\downarrow \quad \downarrow$$

$$y \quad y'$$

$$x_i \rightarrow h(x)$$

$$y = g^{-1}(w^T x + b)$$



对数函数起到的是将线性回归模型的预测值与真实标记联系起来的作用

大纲

- 基本概念
- 线性回归
- 对数几率回归
- 线性判别分析
- 多分类学习

周志华，《机器学习》

对数几率回归

线性回归完成的任务是预测，能否使用回归的方法完成分类的任务?

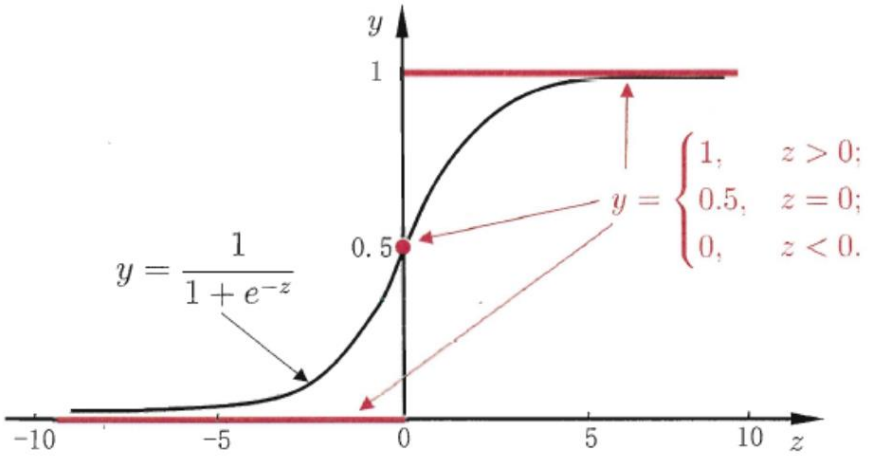
问题：已知 $D=\{(x_1,y_1),(x_2,y_2),\ldots,(x_m,y_m)\}$，
其中 $x_i \in R^d; y \in \{0 \quad 1\}$
求$y=f(x)$；其中 f 的值域为{0 1}判别函数模型

$$y = f(x) = \begin{cases} 1 & x \in D_1 \\ 0 & x \in D_0 \end{cases}$$

找到一个单调可微函数将y与线性回归的预测值联系起来

$$y = \frac{1}{1+e^{-z}}$$ 对数几率函数

一种sigmod函数



红色：单位阶跃函数，不连续
黑色：对数几率函数，0附近陡

对数几率函数

将对数几率函数作为 $g^{-}(\cdot)$ $y=\frac{1}{1+e^{-z}}$ ⇨ $y=\frac{1}{1+e^{-(w^{T}x+b)}}$

$$\ln\frac{y}{1-y}=w^{T}x+b$$ 正例的**可能性**与反例的**可能性**的比值，称为“几率”

用线性模型的预测结果逼近真实标记的几率（odds），
所以称为“对数几率回归”（logistic或logit回归）

优点：直接对分类的可能性建模，无需假设数据分布；
对数几率回归求解的目标函数是任意阶可导的凸函数

逻辑回归，logistic中文翻译为逻辑并不恰当，但是也常见到很多材料中用

模型参数估计

条件分布P(Y|X)

设 $p = P(y=1|x)$

$\Rightarrow 1-p = P(y=0|x)$

事件的几率: 取 $1-p = \frac{1}{1+e^{-(w^T x+b)}}$

$$\ln\frac{p}{1-p} = \ln\frac{1-\frac{1}{1+e^{-(w^T x+b)}}}{\frac{1}{1+e^{-(w^T x+b)}}} = w^T x + b$$

类后验概率估计

$$P(y=1|x;w,b) = \frac{e^{w^T x+b}}{1+e^{w^T x+b}}$$

$$P(y=0|x;w,b) = \frac{1}{1+e^{w^T x+b}}$$

用极大似然法估计参数w，b

目标：每个样本属于其真实标记的概率越大越好

$L(w,b) =$ 似然函数

$$\prod_{i=1}^{m} P(y_i|x_i;w,b) = \prod_{i=1}^{m}[P(y_i=1|x_i;w,b)]^{y_i}[P(y_i=0|x_i;w,b)]^{1-y_i}$$

对数似然函数

- 对数似然

$$l(w,b)=\sum_{i=1}^{m}\ln P(y_i|\boldsymbol{x}_i;\boldsymbol{w},b)$$

问题转换为**以对数似然函数为目标的最优化问题**

取 $\hat{W},\hat{x}$ 如多元回归：$P(y=1|\hat{x};\hat{W})=\frac{e^{\hat{W}^T\hat{x}}}{1+e^{\hat{W}^T\hat{x}}}$；$P(y=0|\hat{x};\hat{W})=\frac{1}{1+e^{\hat{W}^T\hat{x}}}$

矩阵

对数似然函数为：$l(\hat{W})=\sum_{i=1}^{m}\ln P(y_i|\hat{x}_i;\hat{W})=\sum_{i=1}^{m}\left[y_i\hat{W}^T\hat{x}_i-\ln\left(1+e^{\hat{W}^T\hat{x}_i}\right)\right]$

求解优化问题，得到解：$\hat{W}^*=\arg\max_{\hat{W}} l(\hat{W})$ 梯度下降法、牛顿法等数值算法求解

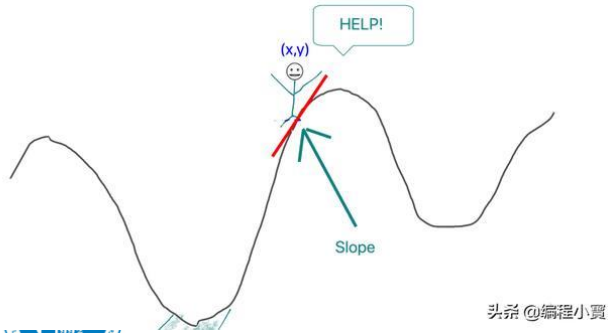
最优化问题求解

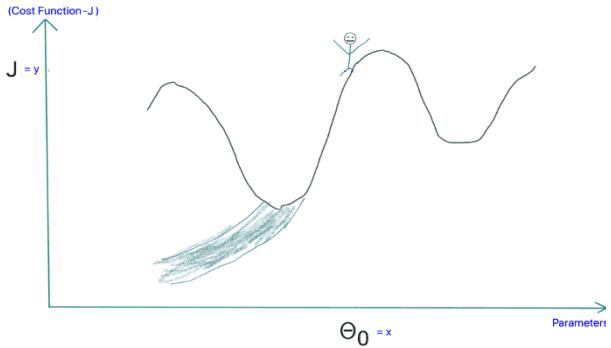
梯度下降法(gradient descent)：是一种求解无约束优化问题的常用方法，其基本思想是对于最小化问题，沿**目标函数下降最快**的方向，逐步搜索直到最小值点。

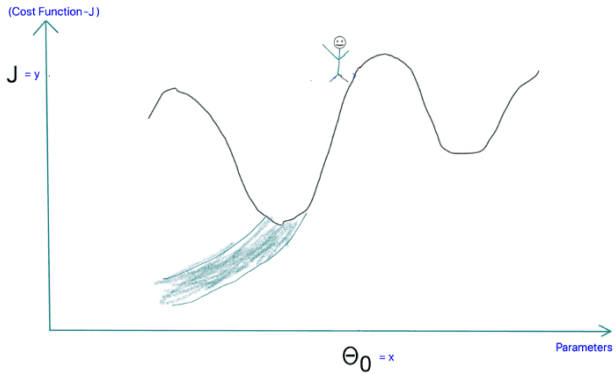
问题：f 是R^n上具有一阶连续偏导数的函数。

$$x^* = \arg\min_{x \in R^n} f(x)$$

当目标函数是凸函数时，梯度下降法的解是全局最优解，一般情况不能保证全局最优。







梯度下降法

输入：目标函数$f(x)$，梯度函数$g(x)$，精度 ε；

输出：$f(x)$ 的极小值点x^*；

梯度：单变量求导
多变量求偏导，组成向量

(1) 取初始值$x^{(0)}$，置k=0；

(2) 计算 $f(x^{(k)})$；

(3) 计算梯度$g_k=g(x^{(k)})$，当$|g_k|<\varepsilon$时，停止迭代$x^*=x^{(k)}$；否则令$p_k=-g(x^{(k)})$求 λ_k使：

负梯度

$$f(x^{(k)}+\lambda_k p_k)=\min_{\lambda\geq 0} f(x^{(k)}+\lambda p_k)$$

(4) 置 $x^{(k+1)}=x^{(k)}+\lambda_k p_k$，计算 $f(x^{(k+1)})$，

当$|f(x^{(k+1)})-f(x^{(k)})|<\varepsilon$ 或$|x^{(k+1)}-x^{(k)}|<\varepsilon$，停止迭代，令 $x^*=x^{(k+1)}$；

(5) 否则置$k=k+1$，转 (3)

多项对数几率回归

多项Logistic回归：将二分类问题推广到用于多分类问题的多项对数几率回归：

$$P(y=k \mid x;W) = \frac{e^{W_k x}}{1+\sum_{k=1}^{K-1} e^{W_k x}}, k = 1,2,\mathrm{L}\ ,K-1$$

K个类别

$$P(y=K \mid x;W) = \frac{1}{1+\sum_{k=1}^{K-1} e^{W_k x}}$$

大纲

- 基本概念
- 线性回归
- 对数几率回归
- 线性判别分析
- 多分类学习

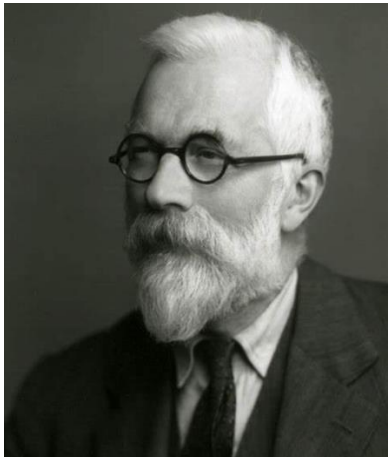
周志华，《机器学习》

Fisher判别分析

线性判别分析（LDA）是一种经典的线性学习方法，1936年**Fisher（英国统计学家和生物学家）**提出，也称为Fisher判别分析

使用极大似然估计法（1912-1922）

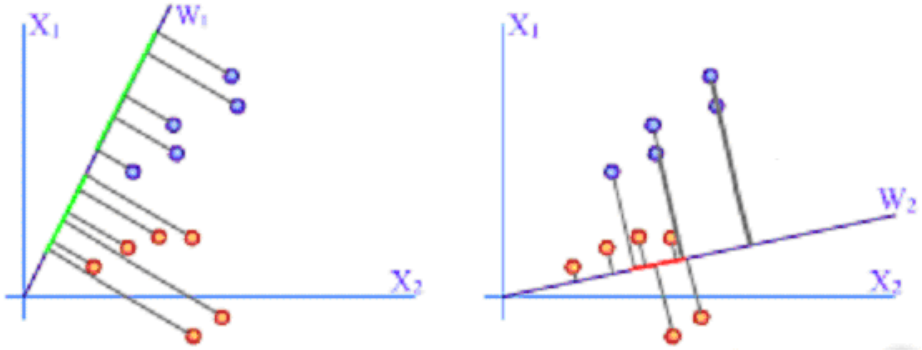
鸢尾花数据集Iris：常用的分类实验数据集



罗纳德·费希尔 Ronald Fisher（1890～1962），现代统计科学的奠基人之一。

Fisher判别分析

- 把X 空间各点投影到X 空间的一直线上(Z)，维数降为一维。
- 若适当选择w的方向，可以使二类分开。从数学上寻找最好的投影方向，即寻找最好的变换向量w的问题。



图中w_1方向之所以比w_2方向优越，可以归纳出这样一个准则，即向量w的方向选择应能使两类样本投影的均值之差尽可能大些，而使类内样本的离散程度尽可能小----Fisher准则函数的基本思路。

Fisher准则基本原理：找到一个最合适的投影轴，使两类样本在该轴上投影的交迭部分最少，从而使分类效果为最佳。

二分类问题

已知-数据集合(D)：$D=\{(x_1,y_1),(x_2,y_2),\mathrm{L}\ ,(x_m,y_m)\}$

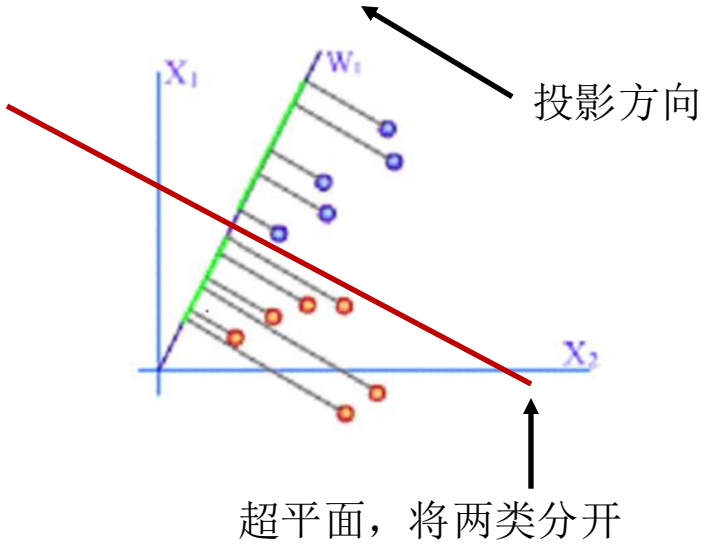
$$where: x_i=\begin{pmatrix} x_{i1} \\ x_{i2} \\ \mathrm{M} \\ x_{id} \end{pmatrix} \in R^d; y_i \in \{0,1\}$$

假设空间(H)：$H=\{f(x)=w^Tx+b\}$

求判别函数：$f(x)=w^Tx+b$

第一步：确定投影方向 $Z=w^Tx$

第二步：确定判别函数 $Z=w^Tx+b>0$ 或<0



确定投影方向---投影点评估

同类样本的投影点尽可能接近----协方差尽可能小

异类样本的投影点尽可能远离----两类中心距离尽可能大

定义样本在d维特征空间的一些描述量

(1) 样本均值向量μ_i $\mu_i = \frac{1}{N}\sum_{x \in D_i} x, \quad i = 0,1$

(2) 样本协方差矩阵$\boldsymbol{\Sigma}_i$ $\Sigma_i = \sum_{x \in D_i} (x - \mu_i)(x - \mu_i)^T$

样本中心在直线的投影：$w^T \mu_i$

两类样本直线投影上的协方差：$w^T \Sigma_i w$

*协方差

- 协方差是一个衡量两个随机变量**线性相关程度**以及它们的变化方向的度量。
- 协方差矩阵有助于了解数据集中**不同变量之间的关系**和它们的总体分布。
- 协方差计算公式如下： $Cov(x_i, x_j) = E[(x_i - \mu_i)(x_j - \mu_j)]$

为n维随机变量 X 的协方差矩阵（covariance matrix），也记为 $D(X)$，其中

$c_{ij} = Cov(X_i, X_j), i, j = 1, 2, \ldots, n$ 为 X 的分量 X_i 和 X_j 的协方差（设它们都存在）。

例如，二维随机变量 (X_1, X_2) 的协方差矩阵为 $C = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$

其中 $c_{11} = E[X_1 - E(X_1)]^2, c_{12} = E[X_1 - E(X_1)][X_2 - E(X_2)]$

$c_{21} = E[X_2 - E(X_2)][X_1 - E(X_1)], c_{22} = E[X_2 - E(X_2)]^2$

由于 $c_{ij} = c_{ji}(i, j = 1, 2, \ldots, n)$，所以协方差矩阵为对称非负定矩阵。 [2]

优化目标

同时考虑类内协方差最小与类中心之间的距离最大

$$J = \frac{||w^T\mu_0 - w^T\mu_1||_2^2}{w^T\Sigma_0 w + w^T\Sigma_1 w} = \frac{w^T(\mu_0 - \mu_1)(\mu_0 - \mu_1)^T w}{w^T(\Sigma_0 + \Sigma_1)w} \begin{matrix} \longrightarrow \text{大} \\ \longrightarrow \text{小} \end{matrix}$$

定义“样本类内离散度矩阵”

$$\begin{aligned} S_w &= \Sigma_0 + \Sigma_1 \\ &= \sum_{x \in D_0}(x - \mu_0)(x - \mu_0)^T + \sum_{x \in D_1}(x - \mu_1)(x - \mu_1)^T \end{aligned}$$

定义“样本间离散度矩阵”

$$S_b = (\mu_0 - \mu_1)(\mu_0 - \mu_1)^T$$

求解

确定w：评价投影方向w的函数为：

$$J = \frac{w^T S_b w}{w^T S_w w} \begin{matrix} \longrightarrow 大 \\ \longrightarrow 小 \end{matrix}$$

求解使分子尽可能大，分母尽可能小的w作为投影向量

最佳w值的确定实际上就是对Fisher准则函数求取其达极大值时的w^*。

$$\min_w -w^T S_b w$$

$$s.t. \quad w^T S_w w = 1$$

采用拉格朗日乘子法 令： $w^T S_w w = c \neq 0$ **，定义Lagrange函数：**

$$L(w, \lambda) = w^T S_b w - \lambda(w^T S_w w - c)$$

$$L(w,\lambda) = w^T S_b w - \lambda(w^T S_w w - c)$$

求解

对拉格朗日函数分别对w求偏导并置为0来求w的解

令：$\dfrac{\partial L(w,\lambda)}{\partial w} = 2(S_b w - \lambda S_w w) = 0$ → $S_b w^* = \lambda S_w w^*$ → $S_w^{-1} S_b w^* = \lambda w^*$

矩阵求导

这是一个求矩阵$S_w^{-1} S_b$ 的特征值问题

$S_b = (\mu_0 - \mu_1)(\mu_0 - \mu_1)^T$

$$\lambda w^* = S_w^{-1} S_b\, w^* = S_w^{-1}(\mu_0 - \mu_1)\underline{(\mu_0 - \mu_1)^T w^*}$$

标量，数值R

$$w^* = \frac{R}{\lambda} S_w^{-1}(\mu_0 - \mu_1) \simeq S_w^{-1}(\mu_0 - \mu_1)$$

实际上只关心向量w^*的方向，其数值大小对分类器没有影响。因此在忽略了数值因子R/λ后，可得：$w^* = S_w^{-1}(\mu_0 - \mu_1)$

使用Fisher准则求最佳法线向量的解

对最佳投影的理解

向量$\boldsymbol{w}^*$就是使Fisher准则函数$J_F(w)$达极大值的解，也就是按**Fisher准则将d维X空间投影到一维Y空间的最佳投影方向，该向量$\boldsymbol{w}^*$的各分量值是对原d维特征向量求加权和的权值**。

最佳投影方向的理解 $w^* = S_w^{-1}(\mu_0 - \mu_1)$

- $(\mu_0 - \mu_1)$是一维向量，显然从两类均值在变换后距离最远这一点看，**对与$(\mu_0 - \mu_1)$平行的向量投影可使两均值点的距离最远。**
- 但是如从使类间分得较开，同时又使类内密集程度较高这样一个**综合指标来看**，则需**根据两类样本的分布离散程度对投影方向作相应的调整**，这就体现在对向量按$S_w^{-1}(\mu_0 - \mu_1)$作一线性变换，从而使Fisher准则函数达到极值点。

贝叶斯公式的理解

假设各个类别的样本数据符合高斯分布：

训练：利用LDA进行投影后，利用**极大似然估计计算各个类别投影数据的均值和方差**，得到**该类别高斯分布的概率密度函数**。

测试：新的样本到来后，将它投影，**将投影后的样本特征分别带入各个类别的高斯分布概率密度函数，计算它属于该类别的概率**，最大的概率对应的类别即为预测类别。

$$p(x_i \mid c) = \frac{1}{\sqrt{2\pi}\sigma_{c,i}} \exp\left(-\frac{(x_i - \mu_{c,i})^2}{2\sigma_{c,i}^2}\right).$$

$$h^*(x) = arg\max_{c \in y} P(c|x)$$

确定判别函数

分类器设计-确定，判别函数 $f(x) = w^T x + b$

(1) 当维数d与样本数m都很大时，可采用**贝叶斯决策规则（每个类别的投影结果，代入贝叶斯公式）**，获得一种在一维空间的“最优”分类器。

(2) 当上述条件不满足时，一般可采用以下几种方法确定分界阈值点b:

$$b = -\frac{\widehat{\mu_0} + \widehat{\mu_1}}{2} \qquad b = -\frac{\widehat{\mu_0} m_0 + \widehat{\mu_1} m_1}{m_0 + m_1} = \hat{\mu}$$

(1) 中只考虑采用投影后均值连线中点作为阈值点，相当于贝叶斯决策中先验概率相等的情况；(2) 考虑类别概率不等的影响，以减小先验概率不等时的错误率。

决策规则

当b确定之后，则可按以下规则分类：

$$\begin{cases} if \quad w^{*T}x > b \rightarrow y = 1 \\ if \quad w^{*T}x < b \rightarrow y = 0 \end{cases}$$

使用Fisher准则方法确定最佳线性分界面的方法是一个著名的方法，尽管提出该方法的时间比较早，仍见有人使用，如人脸识别中用于特征提取。

例：设两类样本的类内离散矩阵分别如下，试用Fisher准则求其决策面方程。

$$S_1 = \begin{pmatrix} 1 & 1/2 \\ 1/2 & 1 \end{pmatrix}; S_2 = \begin{pmatrix} 1 & -1/2 \\ -1/2 & 1 \end{pmatrix};$$

$$S_W = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}; W^* = S_W^{-1}(\mu_1 - \mu_2) = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$$

$$\mu_1 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}; \mu_2 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$$

直接计算

由于两类样本分布形状是相同的（只是方向不同），因此b应为两类均值的中点 $b = w^{*T}\dfrac{\mu_1 + \mu_2}{2} = -1$

测试： 样本x_2的类别 $w^{*T}x - b = -x_2 + 1 = 0$

LDA降维算法流程

LDA可用于降维

- 1) 计算类内散度矩阵S_w
- 2) 计算类间散度矩阵S_b
- 3) 计算矩阵$S_w^{-1}S_b$
- 4）计算$S_w^{-1}S_b$的最大的d个特征值和对应的d个特征向量$(w_1, w_2, \ldots, w_d)$，得到投影矩阵w
- 5) 计算样本集中每个样本转换后的新样本： $z_i = w^T x$
- 6) 得到输出样本集$D'=\{(z_i, y_i)\}$

*LDA的核方法

如果F空间的维数非常高甚至是无穷维数，那么单纯的只是将原数据投影到F空间就是一个很大的计算量。

但是，可以并不显式的进行数据的投影，而只是计算原数据的点乘：$(\Phi(x)\cdot\Phi(y))$。如果我们可以快速高效的计算出点乘来，那么可以无须将原数据投影到F空间就解决问题。

大纲

- 基本概念
- 线性回归
- 对数几率回归
- 线性判别分析
- 多分类学习

多分类问题

前面讨论的分类问题时大多是二分类问题，但多数实际问题是多分类：

特征空间：$X \subseteq R^d; x = (x_1 \quad x_2 \quad \mathrm{L} \quad x_d) \in R^d$

输出空间：$Y=\{C_1, \ C_2, \cdots, \ C_N\}$

数据集：$D=\{ \ (x_1, \ y_1), \ (x_2, \ y_2), \cdots, \ (x_m, \ y_m)\}=\{D_1, \ D_2, \ \cdots, \ D_N\}$

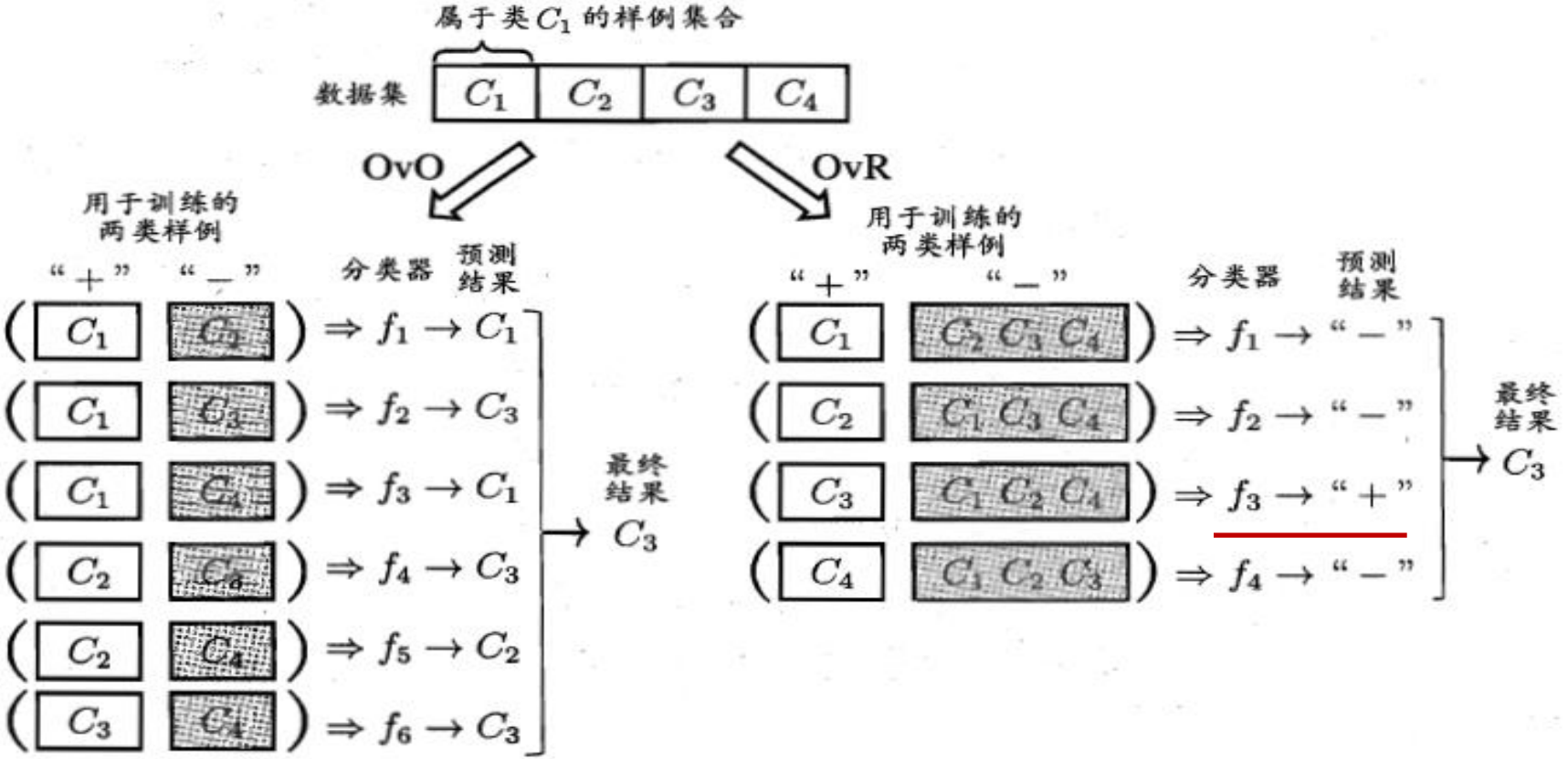
解决此问题的方法是拆分，**将多分类问题拆分为若干个二分类问题**：

（1）一对一与一对多拆分-(OvO，OvR)

OvO：将N类问题分解为N(N-1)/2个二分类问题，训练N(N-1)/2个分类器，使用时，将样本同时提交给所有分类器，然后对N(N-1)/2个分类结果投票产生最终结果；

OvR：选择一类作为正类，其余皆为负类，训练N个二分类器，使用时，将样本同时提交给所有分类器，若结果只有一个是正类，则最终结果即为此类，若有多个分类器的结果是正类，最终结果需要其他方法确定。

一对多



N(N-1)个二分类器

多对多

纠错输出编码(Error-Correcting Output Codes，ECOC) （1995年）为一种多类分解框架，一般将多类分类问题分解为编码、训练、解码三个阶段：

1、编码：对N个类做M次划分，每次划分将一部分类划为正类，另一部分划为负类，从而形成一个二分类训练集。这样一共有M个训练集，可以训练出M个分类。一般采用二元码或三元码的方式编码。

二元ECOC码

C_1	-1	1	-1	1	1
C_2	1	-1	-1	1	-1
C_3	-1	1	1	-1	1
C_4	-1	-1	1	1	-1

三元ECOC码

C_1	-1	1	-1	-1	-1	0	1
C_2	1	-1	0	0	-1	1	-1
C_3	-1	1	0	1	1	-1	1
C_4	-1	-1	1	0	1	1	-1

编码策略：事前编码(predefined coding) 、基于样本数据编码(data depended coding)和基于基分类器编码 (based dichotomizes coding)

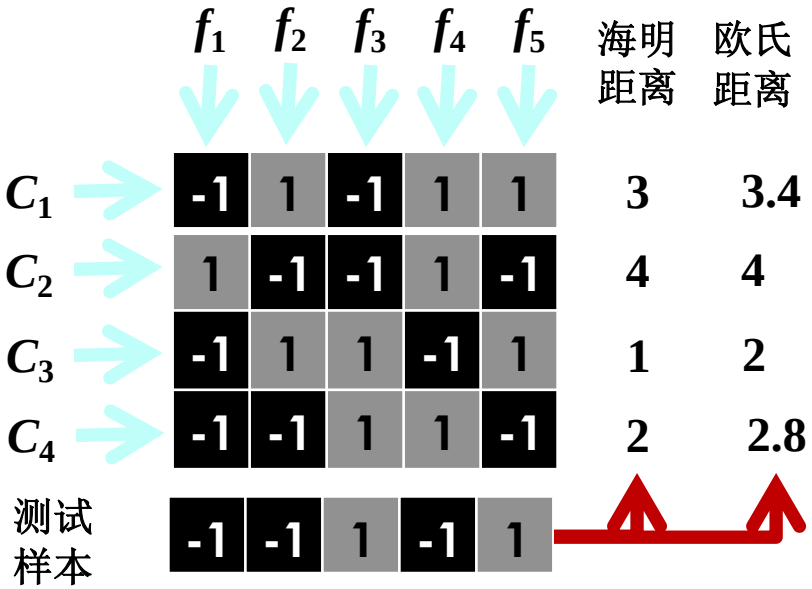
多对多

2、学习：对M个训练集，训练出M个分类器：$f_1, f_2, \ldots, f_M$。

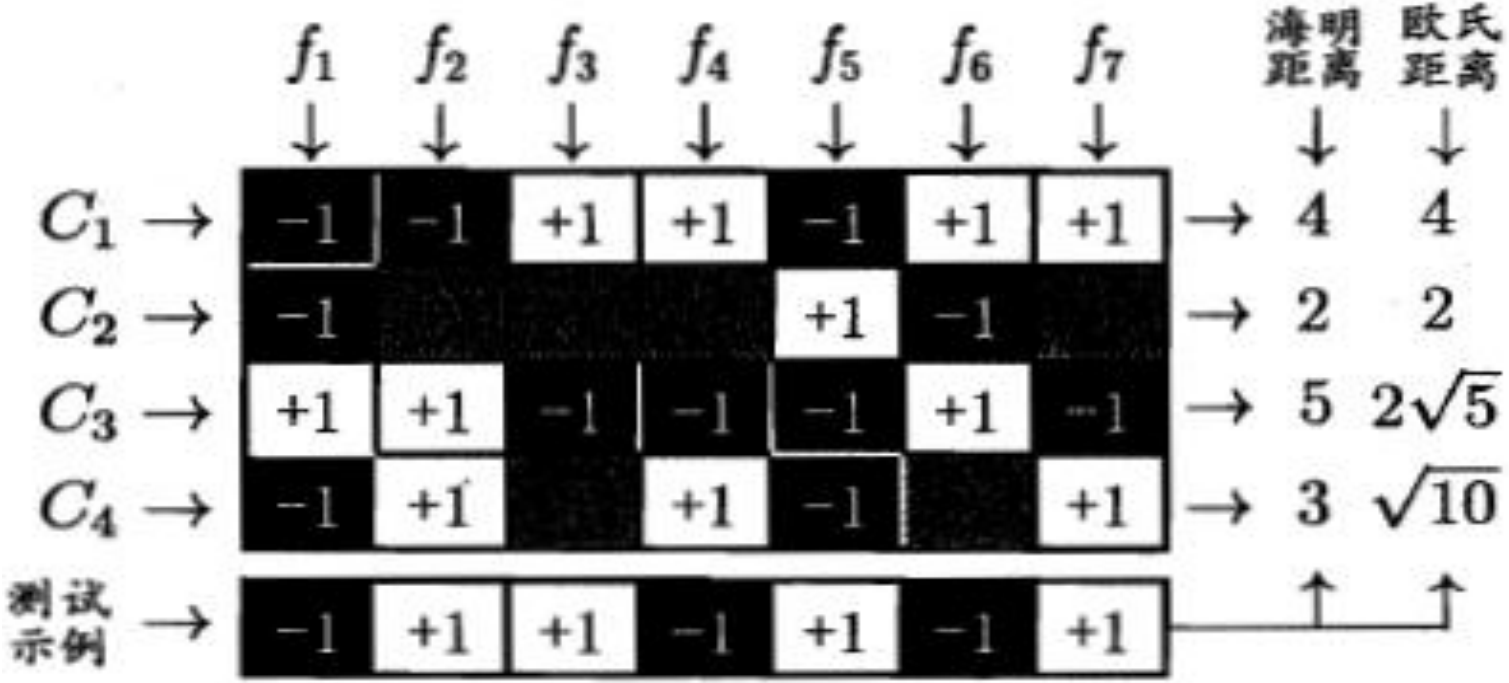
3、解码策略：M个分类器分别对测试样本进行预测，预测结果组成一个预测编码，将预测编码与每个类的编码进行比较，**返回距离最小的类作为最终预测结果。**

海明距离：从二进制方面来看，就是两个等长字符串的二进制对应 bit 不相同的位个数。

欧氏距离



多对多



编码越长，纠错能力越强，但所需分类器越多