# 第三章 线性模型

很熟悉的基本形式

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

以及向量形式

$$f(x) = w^T x + b$$

# 线性回归

数据集  $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ , linear refression 学习得到一个线性模型尽可能准确预测实值的输出标记。

### 均方误差

用于线性回归中的性能度量。

#### 欧几里得距离

欧氏距离 Euclidean distance,就是最熟知的直角三角形求你斜边长。对于二维,  $dis=\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$ ,对于三维,  $dis=\sqrt{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z1)^2}$ 

#### MSE

反映估计量与被估计量之间的差异程度,即估计值与真值之差的平方的期望。MSE可以评价数据的变化程度,值越小精确度越好

和方差

$$SSE = \sum_{i=1}^m w_i (f(x_i) - y_i)^2$$

均方差

$$MSE = rac{SSE}{n} = rac{1}{n} \sum_{i=1}^m w_i (f(x_i) - y_i)^2$$

MSE最小化

$$egin{split} (w^*,b^*) &= arg\min_{(w,b)} \sum_{i=1}^m (f(x_i) - y_i)^2 \ &= arg\min_{(w,b)} \sum_{i=1}^m (f(x_i) - wx_i - b)^2 \end{split}$$

# 最小二乘法

least square nethod --基于均方误差最小化进行模型求解。找一条直线,所有样本到直线上的欧氏距离之和最小。

#### 参数估计

即求 w 和 b 使 SSE 最小化。对 w 和 b 分别求偏导:

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

$$\frac{\partial E_{(w,b)}}{\partial b} = 2\left(mb - \sum_{i=1}^{m} (y_i - wx_i)\right) = 0$$

$$\Rightarrow w = \frac{\sum_{i=1}^{m} y_i(x_i - \overline{x})}{\sum_{i=1}^{m} x_i^2 - \frac{1}{m}(\sum_{i=1}^{m} x_i)^2}$$

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

### 多元线性回归

对于  $f(x_i) = w^T x_i + b_i$  数据集 D 为  $m \times (d+1)$  的矩阵 X:

$$oldsymbol{X} = egin{pmatrix} oldsymbol{x}_{11} & oldsymbol{x}_{12} & oldsymbol{x}_{12} & \dots & oldsymbol{x}_{1d} & 1 \ oldsymbol{x}_{21} & oldsymbol{x}_{22} & \dots & oldsymbol{x}_{2d} & 1 \ dots & dots & \ddots & dots & dots \ oldsymbol{x}_{m1} & oldsymbol{x}_{m2} & \dots & oldsymbol{x}_{md} & 1 \end{pmatrix} = egin{pmatrix} oldsymbol{x}_1^T & 1 \ oldsymbol{x}_2^T & 1 \ dots & dots \ oldsymbol{x}_{m1}^T & 1 \end{pmatrix}$$

标记的向量形式  $oldsymbol{y}=(y_1;y_2;\ldots;y_m)$ ,以及 w 与 b 的向量形式

$$\hat{oldsymbol{w}} = (oldsymbol{w}; b) = \left(egin{array}{c} w_1 \ w_2 \ dots \ w_m \ b \end{array}
ight)$$

则大小为  $(d+1) \times 1$ ,有

$$m{X}\hat{m{w}} = egin{pmatrix} x_{11} & x_{12} & \dots & x_{1d} & 1 \ x_{21} & x_{22} & \dots & x_{2d} & 1 \ dots & dots & \ddots & dots & dots \ x_{m1} & x_{m2} & \cdots & x_{md} & 1 \end{pmatrix} \cdot egin{pmatrix} w_1 \ w_2 \ dots \ w_m \ b \end{pmatrix} = egin{pmatrix} w_1 x_{11} + w_2 x_{12} + \dots + w_d x_{1d} + b \ w_1 x_{21} + w_2 x_{22} + \dots + w_d x_{2d} + b \ dots \ v_1 x_{m1} + w_2 x_{m2} + \dots + w_d x_{md} + b \end{pmatrix} = egin{pmatrix} f(x_1) \ f(x_2) \ dots \ f(x_m) \end{pmatrix}$$

有和方差

$$SSE = (oldsymbol{y} - oldsymbol{X}\hat{oldsymbol{w}})^T (oldsymbol{y} - oldsymbol{X}\hat{oldsymbol{w}})$$

对 $\hat{w}$ 求导

$$rac{\partial E_{\hat{m{w}}}}{\partial \hat{m{w}}} = 2m{X}^T(m{X}\hat{m{w}} - m{y})$$

#### 要使上式等于0

- 矩阵的秩 用初等行变换将矩阵A化为阶梯形矩阵,则矩阵中非零行的个数就定义为这个矩阵的秩,记为r(A)
- 满秩矩阵 矩阵的行列数都为r(A)

当  $\boldsymbol{X}^T \boldsymbol{X}$  为满秩矩阵或正定矩阵时, $\boldsymbol{X}^T (\boldsymbol{X} \hat{\boldsymbol{w}} - \boldsymbol{y}) = 0$  得到

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

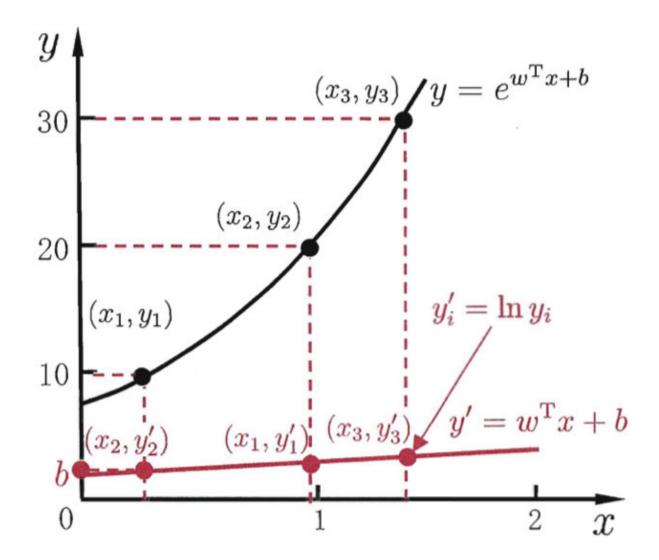
所以

$$f(\hat{oldsymbol{x_i}}) = \hat{oldsymbol{x_i}}^T (oldsymbol{X}^T oldsymbol{X})^{-1} oldsymbol{X}^T oldsymbol{y}$$

但是现实数据集中往往数据很多导致 X 的列数远大于行数,此时可以求解出多个  $\hat{w}$ ,选择其中的一个作为输出则引入 **正则化项**,正则化不展开讨论。

#### 对数线性回归 log-linear regression

简写线性回归模型为  $y'=m{w}^Tm{x}+b$ ,令  $y=e^{m{w}^Tm{x}+b}$ ,则有  $\ln y=m{w}^Tm{x}+b$  对数线性回归示意图



这里看得有点儿懵逼,有几个定义

- 广义线性回归
- 单调可谓函数/联系函数  $g(\cdot)$
- 广义线性回归模型  $y = g^{-1}(\boldsymbol{w}^T\boldsymbol{x} + b)$

对数线性回归是广义线性回归模型在  $g(\cdot) = \ln(\cdot)$  的特例...

# 对数几率回归

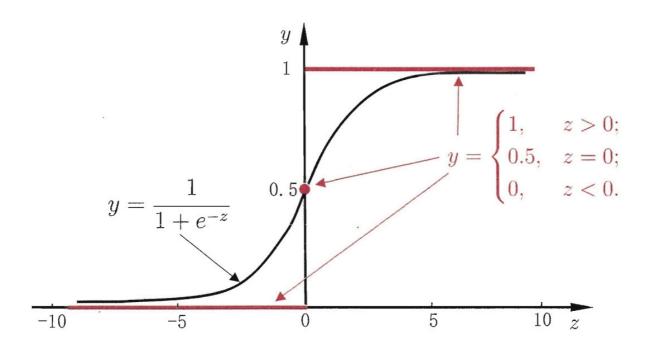
对数几率回归针对分类任务——找一个单调微函数将真实标记 y 和回归模型的预测值进行联系。

### 二分类

对于二分类,将预测出的实值  $z=oldsymbol{w}^Toldsymbol{x}+b$  用 **越阶函数** 映射到区间  $\{0,1\}$ ,即熟悉的 Sigmoid函数

$$y=rac{1}{1+e^{-z}}=egin{cases} 0$$
 ,  $z<0 \ 0.5$  ,  $z=0 \ 1, z>0 \end{cases}$ 

图像



对两边取对数,y 样本的正例概率,则 1-y 为反例概率,则  $\frac{y}{1-y}$  为几率,反应了正例 x 的可能性,对几率取对数,就叫做对数几率。

$$lnrac{y}{1-y} = oldsymbol{w}^Toldsymbol{x} + b$$

使用 极大似然估计 来估计两个参数 w 和 b 的值

$$egin{align} lnrac{p(y=1|oldsymbol{x})}{p(y=0|oldsymbol{x})} &= oldsymbol{w}^Toldsymbol{x} + b \ p(y=1|oldsymbol{x}) &= rac{e^{oldsymbol{w}^Toldsymbol{x} + b}}{1 + e^{oldsymbol{w}^Toldsymbol{x} + b}} \ p(y=0|oldsymbol{x}) &= rac{1}{1 + e^{oldsymbol{w}^Toldsymbol{x} + b}} \ l(oldsymbol{w},b) &= \sum_{i=1}^m \ln p(y_i|oldsymbol{x}_i;oldsymbol{w},b) \end{aligned}$$

使每个样本接近真实值的概率越大越好。令 $({m w},b)={m eta}$ , $\hat x=({m x};1)$ ,则 ${m w}^T{m x}+b={m eta}^T\hat x$ 令

$$egin{aligned} p_1(oldsymbol{\hat{x}};eta) &= p(y=1|oldsymbol{\hat{x}};oldsymbol{eta}) \ p_0(oldsymbol{\hat{x}};eta) &= p(y=0|oldsymbol{\hat{x}};oldsymbol{eta}) = 1 - p_1(oldsymbol{\hat{x}};eta) \end{aligned}$$

重写似然项

$$p(y_i|oldsymbol{x_i; w, b}) = y_i p_1(oldsymbol{\hat{x}};eta) + (1-y_i)p_0(oldsymbol{\hat{x}};eta)$$

则把式子代入到最大化式得到

$$l(eta) = \sum_{i=1}^m \ln(y_i p_1(\hat{oldsymbol{x}}_i;eta) + (1-y_i) p_0(\hat{oldsymbol{x}}_i;eta))$$
 
$$\sharp \, \stackrel{\cdot}{=} \, p_1(\hat{oldsymbol{x}}_i;eta) = \frac{e^{eta^T\hat{oldsymbol{x}}_i}}{1+e^{eta^T\hat{oldsymbol{x}}_i}}, p_0(\hat{oldsymbol{x}}_i;eta) = \frac{1}{1+e^{eta^T\hat{oldsymbol{x}}_i}}, \stackrel{\cdot}{\sim} \lambda$$
 
$$l(eta) = \sum_{i=1}^m \ln\left(\frac{y_i e^{eta^T\hat{oldsymbol{x}}_i} + 1 - y_i}{1+e^{eta^T\hat{oldsymbol{x}}_i}}\right) = \sum_{i=1}^m \left(\ln(y_i e^{eta^T\hat{oldsymbol{x}}_i} + 1 - y_i) - \ln(1+e^{eta^T\hat{oldsymbol{x}}_i})\right)$$

因为  $y_i = 0$ or1

$$l(eta) = egin{cases} \sum_{i=1}^m (-\ln(1+e^{eta^T\hat{m{x}}i})), & y_i = 0 \ \sum_{i=1}^m (eta^T\hat{m{x}}_i - \ln(1+e^{eta^T\hat{m{x}}i})), & y_i = 1 \end{cases}$$

综合得到

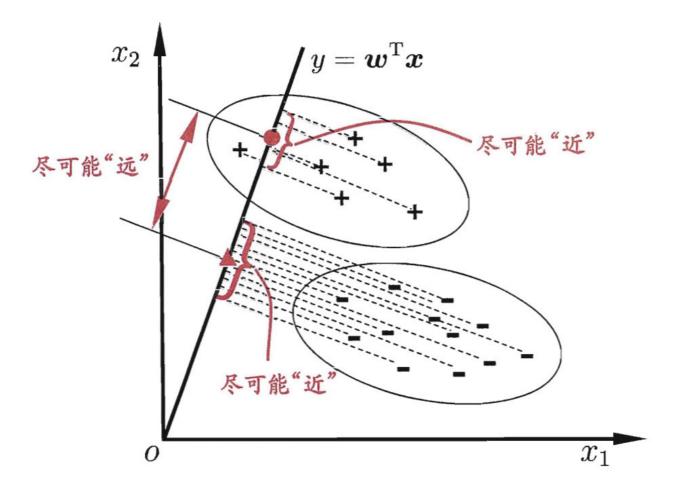
$$l(eta) = \sum_{i=1}^m (-y_i oldsymbol{eta}^T \hat{x_i} + \ln(1 + e^{oldsymbol{eta}^T \hat{x_i}}))$$

对 $\beta$ 求导得

$$egin{aligned} rac{\partial l(eta)}{\partial eta} &= -\sum_{i=1}^m \hat{m{x}}_i(y_i - \hat{y}i) = \sum_{i=1}^m \hat{m{x}}_i(\hat{y}_i - y_i) \ &= m{X}^T(\hat{m{y}} - m{y}) \ &= m{X}^T(p_1(m{X};eta) - m{y}) \end{aligned}$$

# 线性判别分析LDA

Linear Discriminant Analysis,思路:将训练样本投影到一条直线上,使得同类的样例尽可能近,不同类的样例尽可能远



- 数据集  $D = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^m, y_i \in \{0, 1\}$
- $X_i,\ \mu,\ \sum_i,\ i\in\{0,1\}$ 数据集、均值向量、协方差矩阵
- 两类样本在直线 w 上的投影分别为  $w^T \mu_o$ ,  $w^T \mu_1$
- 两类样本的协方差分别为  $\boldsymbol{w}^T \sum_{o} \boldsymbol{w}, \ \boldsymbol{w}^T \sum_{1} \boldsymbol{w}$  尽可能近:  $\boldsymbol{w}^T \sum_{o} \boldsymbol{w} + \boldsymbol{w}^T \sum_{1} \boldsymbol{w}$  尽可能小

   尽可能远:  $||\boldsymbol{w}^T \boldsymbol{\mu}_o \boldsymbol{w}^T \boldsymbol{\mu}_1||_2^2$  尽可能大

#### 这里就涉及到几个距离

• 类间散度矩阵 between-class scaltter matrix, 越大越好

$$S_w = \sum 0 + \sum 1 = \sum_{x \in X_0} (x - \mu_0) (x - \mu_0)^T + \sum_{x \in X_1} (x - \mu_1) (x - \mu_1)^T$$

• 类间散度矩阵 between-class scaltter matrix, 越小越好

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

• LDA 的最大化目标"广义瑞利商" generalized Rayleigh quotient, 越大越好

$$J = rac{ig|oldsymbol{w}^T \mu_0 - oldsymbol{w}^T \mu_1ig|_2^2}{oldsymbol{w}^T (\Sigma_0 + \Sigma_1) oldsymbol{w}} \ = rac{ig|ig|oldsymbol{w}^T (\Sigma_0 + \Sigma_1) oldsymbol{w}}{oldsymbol{w}^T (\Sigma_0 + \Sigma_1) oldsymbol{w}} \ = rac{ig|ig|(\mu_0 - \mu_1)^T oldsymbol{w}ig|_2^2}{oldsymbol{w}^T (\Sigma_0 + \Sigma_1) oldsymbol{w}} \ = rac{ig|oldsymbol{w}^T (\mu_0 - \mu_1)^T oldsymbol{w}}{oldsymbol{w}^T (\Sigma_0 + \Sigma_1) oldsymbol{w}} \ = rac{oldsymbol{w}^T (\mu_0 - \mu_1)(\mu_0 - \mu_1)^T oldsymbol{w}}{oldsymbol{w}^T S_{oldsymbol{w}} oldsymbol{w}}$$

我们要在此确立 w,由于解只和方向有关和长度无关,令  $oldsymbol{w}^T oldsymbol{S}_w oldsymbol{w} = 1$ ,则广义瑞利商可以进一步表示为

$$\min_{w} = -\mathbf{w}^T \mathbf{S}_b \mathbf{w}$$
 $s.t. \ \mathbf{w}^T \mathbf{S}_w \mathbf{w} = 1$ 

拉格朗日乘子法 等价于  $S_b w = \lambda S_w w$ ,推导过程见南瓜书[https://datawhalechina.github.io/pumpkin-book/#/chapter3/chapter3?id=\_330]

令 
$$oldsymbol{S}_boldsymbol{w}=\lambda(oldsymbol{\mu}_0-oldsymbol{\mu}_1),$$
得到  $oldsymbol{w}=oldsymbol{S}_w^{-1}(oldsymbol{\mu}_0-oldsymbol{\mu}_1)$ 

# 多分类学习

- N 个类别  $C_1, C_2, \ldots, C_N$
- 数据集  $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}, \ y \in \{C_1, C_2, \dots, C_N\}$

### 拆解法

#### "-对-"

One vs. One , **OvO**,将 N 个类别两两匹配产生 N(N-1)/2 个二分类任务,结果通过投票产生,被预测最多的类别作为最终的分类结果。

#### "一对其余"

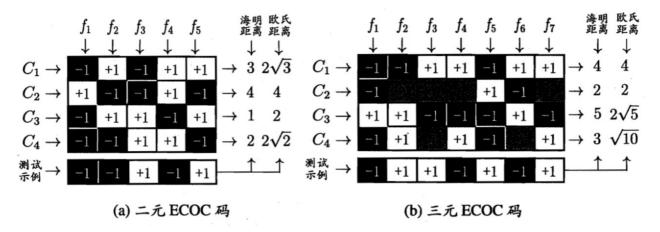
One vs. Rest , **OvR**,每次被一个类的样例作为正例,其余的样例作为反例训练 N 个分类器,最终得到 N 个分类结果,选择分类器的 **预测置信度大** 的。

#### "多对多"

Many vs. Many, MvM, 每次将若干个类作为正例, 其余作为反例。为了能够正确选取正反例, 用到"纠错输出码" Error Correcting Output Codes, ECOC:

- 编码, N 个类做 M 次划分, 得到 M 个分类器
- 解码,M 个分类器妒忌样本进行预测,预测标记组成一个编码,与每个类别自身的编码比较,其中 **距离最小** 的为预测结果

类别划分通过 "编码矩阵" 指定, 二元码制定正反类, 三元码多一个停用类。



上图为编码示意图,+1 为正例,-1 位反例, f 为分类器,C 为类别,对于纠错输出码中的最优编码问题,是个 NP 问题。

#### 方法对比

OvO 时间少, OvR 存储开销少, 两者在大多数情形下的预测性能都取决于数据分布,差别不大。

# 类别不平衡问题

对于一个98%都是正例的样本来说,就预测不出反例,这就是 类别不平衡 问题。

- 预测出的值为 y, 则  $\frac{y}{1-y}$  为几率
- 用之前的阈值 0.5 来决策的规则是, 若几率大于1则为正例
- ullet 令  $m^+,\ m^-$  分别表示正例数与反例数,则观测几率为  $rac{m^+}{m^-}$
- 所以新的判定为当  $\frac{m^+}{m^-} > 1$  时为正例
- 对两者进行综合调整,得到 "再缩放"值

$$rac{y'}{1-y'}=rac{y}{1-y} imesrac{m^-}{m^+}$$

解决方法有以下三种

### 欠采样

在训练样本较多的类别中少取

### 过采样

在训练样本较少的类别中多取

#### 再缩放

基于原数据,对预测值进行再缩放处理。