线性模型

Group Study, ML: 3

2020年6月26日 范嘉楠

本章脉络

- 3.1 基本形式 → 为什么要使用线性模型？
- 3.2 线性回归 → 在回归任务中如何使用、使用的根据
- 3.3 对数几率回归 → 将回归任务拓展到分类任务，先考虑二分类
- 3.4 线性判别分析 → 另一种二分类方法，并可拓展到多分类问题
- 3.5 多分类学习 → 多分类：转化为二分类并解决之
- 3.6 类别不平衡问题 → 此外在分类问题中的常用处理技巧

3.1 基本形式

引入

“三分天注定，七分靠打拼”

“天才就是1%的灵感加上99%的汗水”

$$f(\boldsymbol{x}) = 0.3 \times x_{luck} + 0.7 \times x_{hardwork} + b$$

成功的几率　　　天注定　　　　　打拼

$$f(\boldsymbol{x}) = 0.01 \times x_{inspiration} + 0.99 \times x_{hardwork} + b$$

天才的程度　　　　灵感　　　　　　　汗水

线性模型的一般形式

给定由d个属性描述的示例 $\boldsymbol{x} = (x_1; x_2; \ldots; x_d)$

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：

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

或用向量形式写成 $f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x} + b$

其中 $\boldsymbol{w} = (w_1; w_2; \ldots; w_d)$ ，$\boldsymbol{\omega}$和b学得之后，模型就得以确定。

3.2 线性回归

先捏个软柿子

考虑输入仅有一个属性。此时线性回归(linear regression)试图学得:

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

其中x_i表示第i个数据的属性值，y_i表示x_i对应的标记。

即希望我们的预测值更加逼近标记值

引入

举个例子，考虑数据集$\mathcal{D}$：输入学生的学习时长，输出学生的考试分数

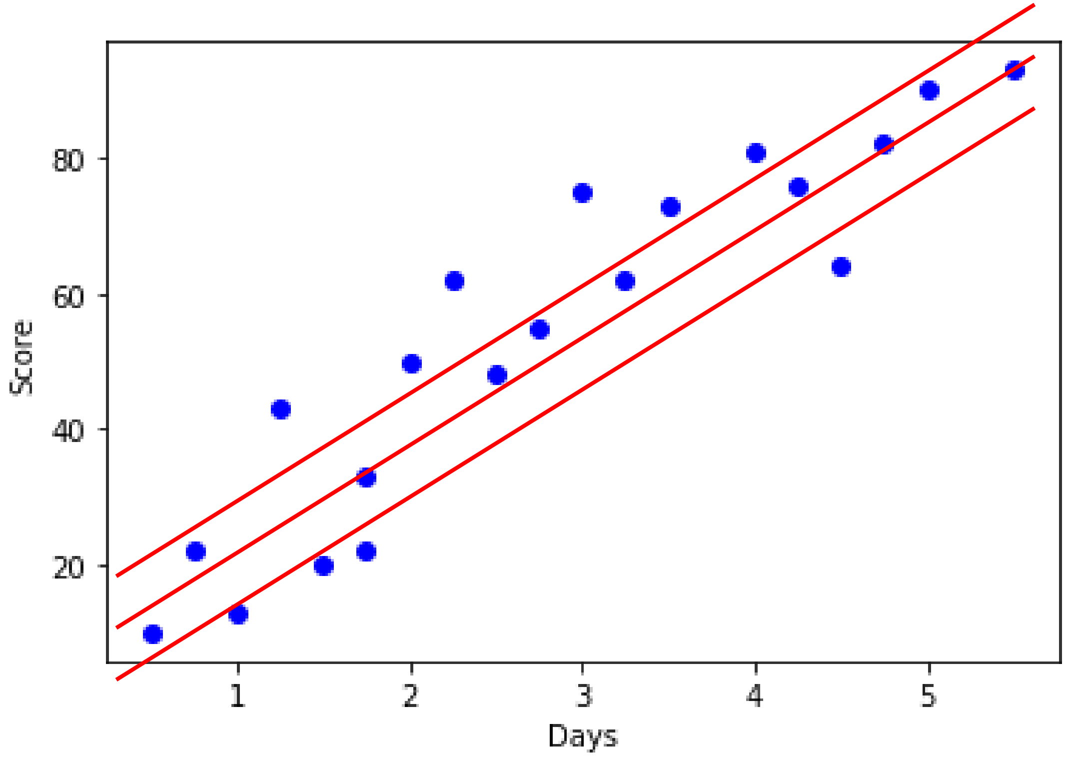
学习时长	0.5	0.75	1	1.25	1.5	1.75	1.75	2	2.25	2.5	2.75
考试分数	10	22	13	43	20	22	33	50	62	48	55

3	3.25	3.5	4	4.25	4.5	4.75	5	5
75	62	73	81	76	64	82	90	93

现在我们尝试对这个数据集建立一个线性模型。

引入

数据集D：输入学生的学习时长，输出学生的考试分数



$$f(x_i) = wx_i + b$$

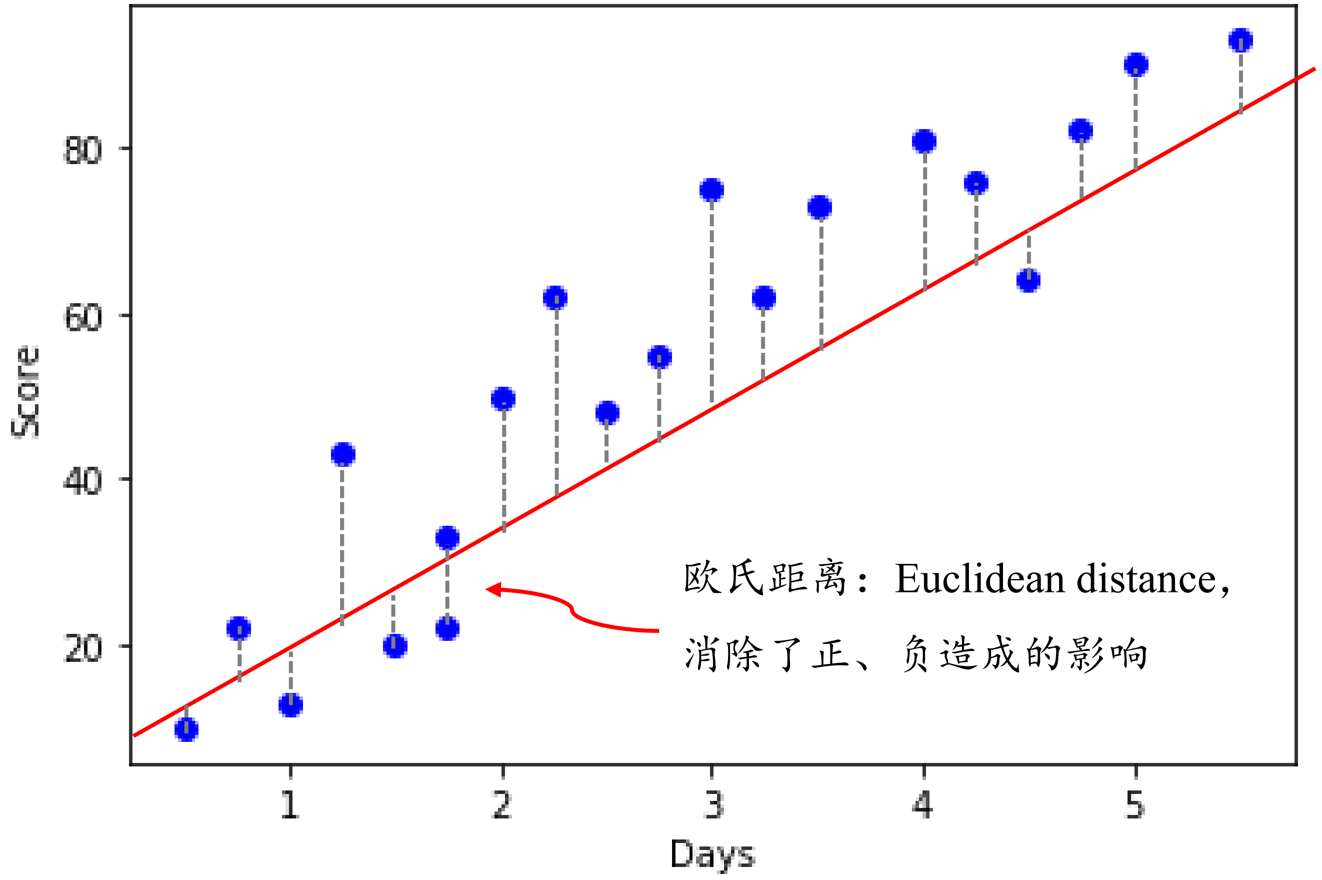
如何确定ω和b?

最小二乘法

least square method，又称最小平方方法

勒让德、高斯等人提出：让均方误差(MSE, Mean Square Error)最小的那条直线，就是最优解。

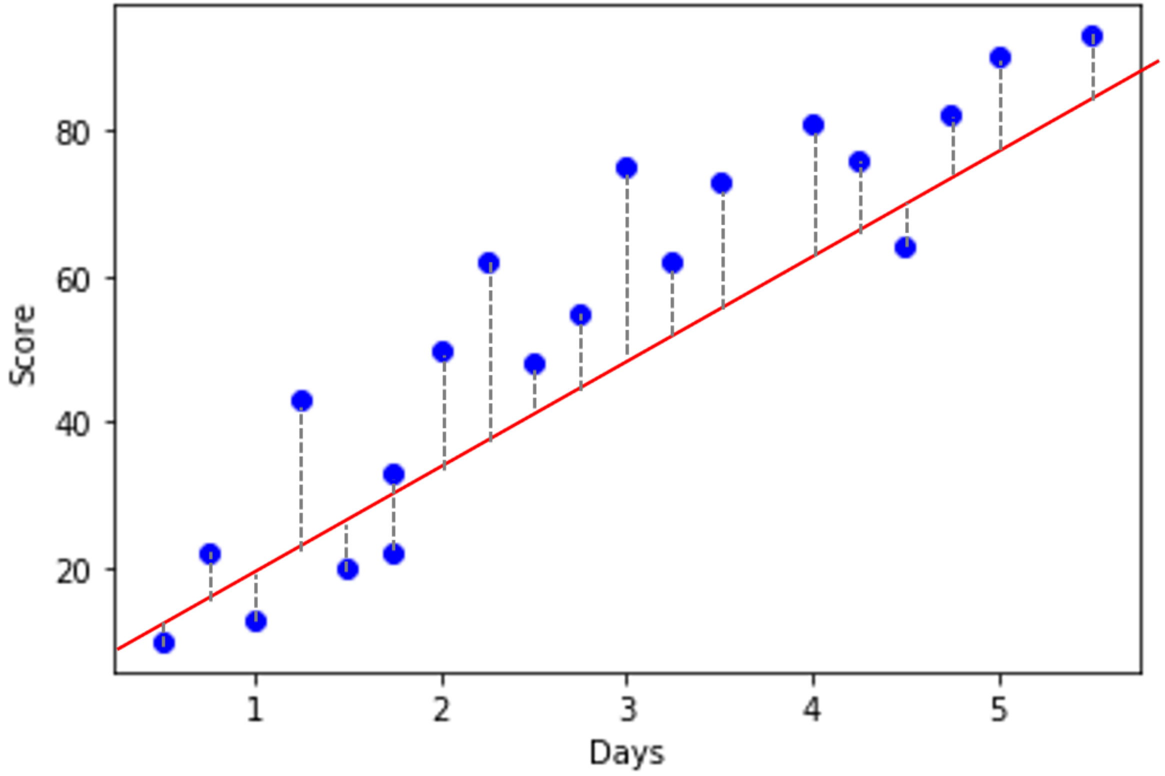
$$(w^*, b^*) = \underset{(w,b)}{\arg\min} \sum_{i=1}^{m} (f(x_i) - y_i)^2$$
$$= \underset{(w,b)}{\arg\min} \sum_{i=1}^{m} (y_i - wx_i - b)^2$$



凭什么使用欧氏距离？

显然，最小二乘法不永远是最优的方法，对应地还有最小绝对值法使用平均绝对误差(MAE, Mean Absolute Error)来进行参数估计。

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}$$



Huber损失函数

此外还有Huber Loss等使用SMAE（平滑平均绝对误差，Smoothed MAE）的方式结合了最小二乘法与最小绝对值法的方式，通过限制误差的阈值来决定使用二者之一。

$$\mathrm{SMAE}_i = \begin{cases} \frac{1}{2}(y_i - \hat{y}_i)^2, & \text{when } |y_i - \hat{y}_i| < \delta, \\ \delta|y_i - \hat{y}_i| - \frac{1}{2}\delta^2, & \text{otherwise} \end{cases}, \mathrm{SMAE} = \frac{\sum \mathrm{SMAE}_i}{n}$$

但是这种方法的一大问题就是我们引入了**另一个未知参数delta**。为了确定最优的delta，我们还需要尝试不同的参数来确定delta的选取。

孰优孰劣

以上讨论的都是线性回归问题中的**损失函数**。最小二乘法亦称L2损失，最小绝对值法亦称L1损失。

1. 最小二乘法**不永远是最优的方法。**对于不同数据形式和建模需求，需要能自行选择合适的建模方式。
2. 相比于最小绝对值法，最小二乘法的优点在于**最优解唯一、求解方便和有好的解析性质**，但缺点在于**受异常值扰动影响大**。
3. Huber Loss结合了最小二乘法和最小绝对值法的优点，但**引入了另一个未知参数delta**。
4. 线性回归还有许多问题不能被最小二乘法或最小绝对值法解决。线性回归里没有一个永远最优的方法。

最小二乘法-结论

$$
\begin{aligned}
(w^*, b^*) &= \underset{(w,b)}{\arg\min} \sum_{i=1}^{m} \left(f(x_i) - y_i\right)^2 \\
&= \underset{(w,b)}{\arg\min} \sum_{i=1}^{m} \left(y_i - wx_i - b\right)^2
\end{aligned}
$$

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

关于求导、化简的过程可以参见南瓜书。

由特殊到一般：多元线性回归

更一般地，输入数据集D中的样本往往由d个属性描述：

$$D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\ldots,(\boldsymbol{x}_m,y_m)\}$$

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

此时我们试图学得：

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

由特殊到一般：多元线性回归

更进一步：我们将b吸收进$\boldsymbol{\omega}$

$$\hat{\boldsymbol{w}}=(\boldsymbol{w};b)$$

对每一个样本$\boldsymbol{x_i}$也增加一个维度，值恒置为1（这样就相当于构造出了$+b$）

$$f(\boldsymbol{x}_i)=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b$$

$$\Downarrow$$

$$f\left(\boldsymbol{x}_i^{'}\right)=\hat{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{x}_i^{'}$$

由特殊到一般：多元线性回归

再进一步：我们希望将标记也写成向量形式、预测值也写成向量形式

$$\boldsymbol{y}=(y_1;y_2;\ldots;y_m)$$

假如我们用\hat表示预测值 $\hat{\boldsymbol{y}}=(\hat{y_1};\hat{y_2};\ldots;\hat{y_m})$

那么进一步有：

$$f\left(\boldsymbol{x}_i^{'}\right)=\hat{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{x}_i^{'}$$

$$\Downarrow$$

$$\hat{\boldsymbol{y}}=\mathbf{X}\hat{\boldsymbol{w}} \qquad \mathbf{X}=\begin{pmatrix} x_{11} & x_{12} & \ldots & x_{1d} & 1 \\ x_{21} & x_{22} & \ldots & x_{2d} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{m1} & x_{m2} & \ldots & x_{md} & 1 \end{pmatrix}=\begin{pmatrix} \boldsymbol{x}_1^{\mathrm{T}} & 1 \\ \boldsymbol{x}_2^{\mathrm{T}} & 1 \\ \vdots & \vdots \\ \boldsymbol{x}_m^{\mathrm{T}} & 1 \end{pmatrix}$$

由特殊到一般：多元线性回归

于是我们得到多元线性回归中，使用最小二乘思想所确定的$\boldsymbol{\omega}$和b（也就是说吸收后的$\hat{\boldsymbol{\omega}}$）

$$\text{式(3.9)}\quad \hat{\boldsymbol{w}}^* = \underset{\hat{\boldsymbol{w}}}{\arg\min}(\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}})^{\mathrm{T}}(\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}})$$

$$\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}} \Longrightarrow \boldsymbol{y} - \hat{\boldsymbol{y}}$$

$$(\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}})^{\mathrm{T}}(\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}}) \Longrightarrow ||\boldsymbol{y} - \mathbf{X}\hat{\boldsymbol{w}}||_2^2$$

$$\Longrightarrow ||\boldsymbol{y} - \hat{\boldsymbol{y}}||_2^2$$

$$\Longrightarrow \boldsymbol{MSE}$$

讨论

对式3.9进行求导得到：

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

零其为零即可解得极值时的$\hat{\boldsymbol{\omega}}$

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

$\Rightarrow$ 也叫**正规方程**
(normal equation)

当且仅当 $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ 是满秩矩阵。（正定阵必满秩，条件更强）

然而往往 $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ 不满秩，因为：

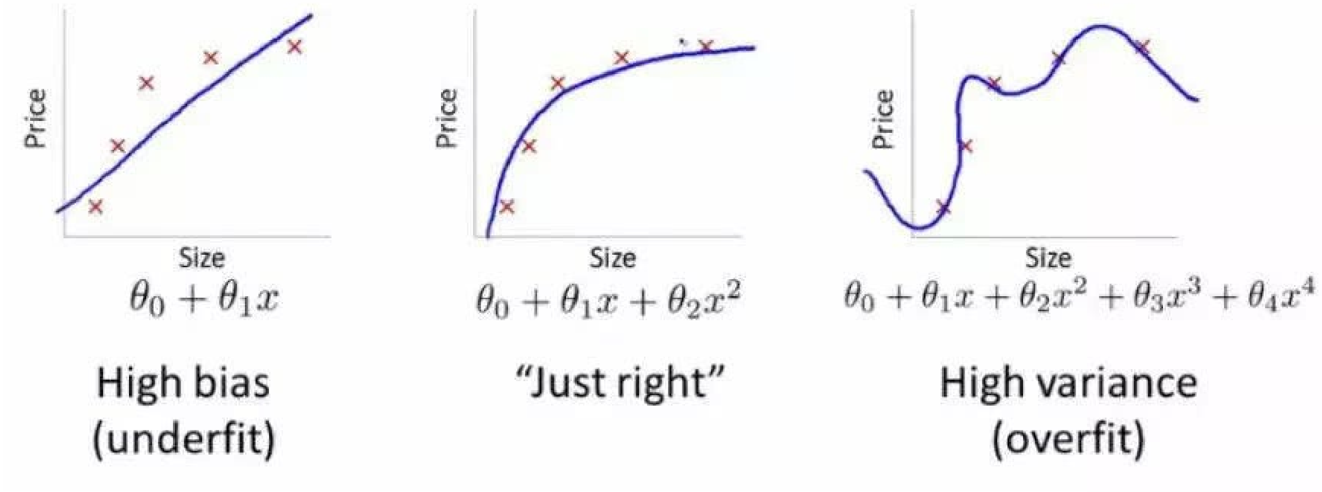
$$r(A) = r(A^T) = r(A^TA) = r(AA^T)$$

讨论

在许多任务中，我们会遇到大量的变量，其数目甚至超过样例数。

在 $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ 不满秩的情况，就遇到了第一章中归纳偏好的问题。

常见的方式是引入正则化（regularization）来解决可能的过拟合问题。



正则化

正则化，只是一种处理过拟合、或者说不满秩的方式

简言之，正则化就是对代价函数增加“惩罚”

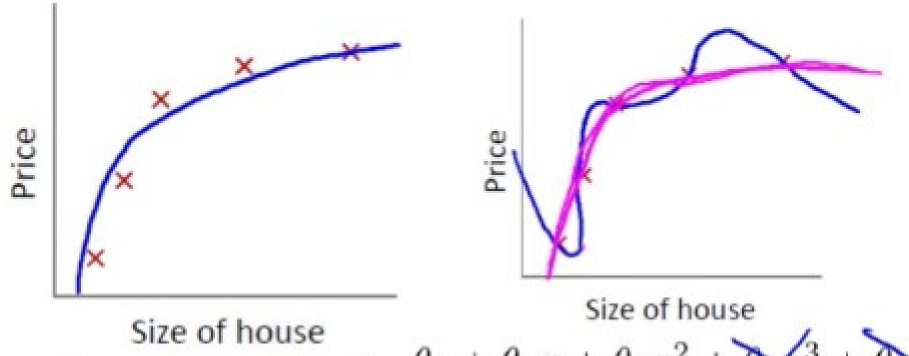
（注意，这是一种优化思想，并不仅针对于线性回归）

$$h_\theta(x)=\theta_0+\theta_1x_1+\theta_2x_2^2+\theta_3x_3^3+\theta_4x_4^4$$

$$\min_\theta \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2+1000\theta_3^2+10000\theta_4^2\right]$$

$$J(\theta)=\frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2+\boxed{\lambda\sum_{j=1}^{n}\theta_j^2}\right]$$

正则化项



$\theta_0+\theta_1x+\theta_2x^2$ $\qquad$ $\underline{\theta_0+\theta_1x+\theta_2x^2}+\cancel{\theta_3x^3}+\cancel{\theta_4x^4}$

其中λ又被称为正则化参数

由特殊到一般：对数线性回归

更进一步，我们从线性模型转变到非线性模型，即希望令预测值逼近y的衍生物

也就是说，如果输出标记在指数尺度上变化，我们仍可将其转化为线性回归问题

这是一种Reduction的思想，也即：

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

这就是“对数线性回归”(log-linear regression)

或者我们也可以说，我们是在让$e^{\omega^{\mathrm{T}}x+b}$逼近$y$

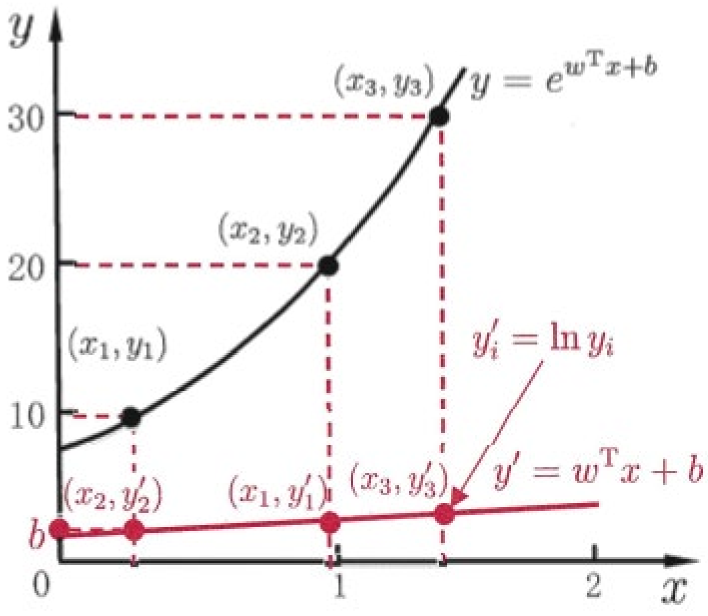


图 3.1 对数线性回归示意图

广义线性模型

更一般地，考虑单调可微函数$g(\cdot)$

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

这样得到的模型就叫做“广义线性模型”(generalized linear model)，其中$g(\cdot)$称为“联系函数”(link function)。显然对数线性回归是一种特例。

这样我们就完成了“用已知的方式解决未知的、复杂的问题”。

3.3 对数几率回归

引入

考虑一个二分类问题。

比如探讨胃癌发生的危险因素。输入一组人群，其中有一些是胃癌患者，一些不是胃癌患者，每个人肯定有不同的体征和生活方式，所以我们可调查的危险因素就可以包括很多，例如年龄、性别、饮食习惯、幽门螺杆菌感染等。

那么我们如何让机器建立一个模型来解决这个问题呢？

对数几率回归：从回归到分类

这仍然是一种Reduction：即使用回归的方式来解决分类问题。虽然对数几率回归叫“回归”，但它实际上在解决分类问题。并且，是二分类问题。

这种转化的关键在于：需要找到一个单调可微函数，将分类任务的真实标记y与线性回归模型预测值联系起来。最简单的比如，单位阶跃函数(unit-step function)：

$$y = \begin{cases} 0, & z < 0 \\ 0,5, & z = 0 \\ 1, & z > 0 \end{cases} \quad \text{其中 } z = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b$$

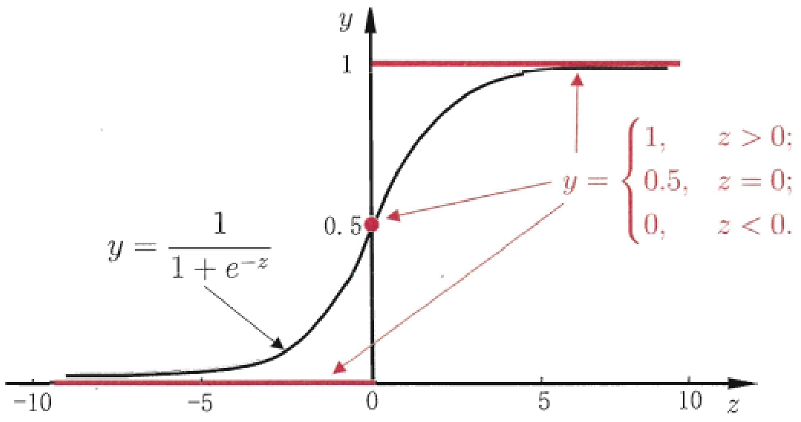


图 3.2　单位阶跃函数与对数几率函数

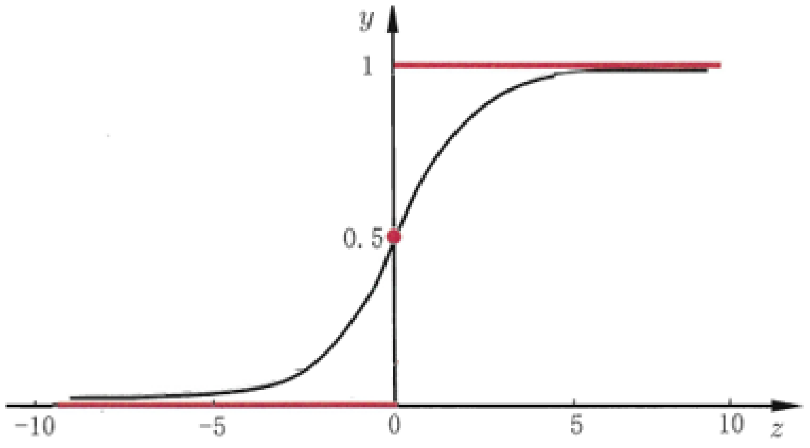
对数几率函数

单位阶跃的弊端：不连续，无法求导。

虽然这种“线性截断”很爽，但是我们没法做线性拟合，这就好像螃蟹很好吃，但是首先要有一个吃螃蟹的人。

这时我们就引入了对数几率函数(logistic function)：

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



也叫对率函数、逻辑斯蒂函数，是Sigmoid函数的一种。

题外话：Sigmoid函数常被用于神经网络中的激活函数。

对数几率函数

显然，这里只是对数几率函数处理后是这样，选择其他函数也会得到其他的线性模型

接下来只需要做数学处理：

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

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

$$\Longrightarrow \quad \ln \frac{y}{1-y} = w^{\mathrm{T}}x + b$$

讨论：$\frac{y}{1-y}$ 样本***x***作为正例的可能性/作为反例的可能性，称为“几率”(odds)

对几率取对数，就称为“对数几率”(log odds, 也称logit) $\ln \frac{y}{1-y}$

如何理解对数几率回归模型

类似于贝叶斯定理，这是一种类后验概率：更科学的不是直接做判断是或否为胃癌患者，而是有多大的几率是胃癌患者，或者有多大的几率患胃癌。

$$
\begin{aligned}
y &\rightarrow \quad p(y=1 \mid \boldsymbol{x}) \\
1-y &\rightarrow \quad p(y=0 \mid \boldsymbol{x})
\end{aligned}
$$

于是我们就可以将之前的式子重写为：

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

对数几率回归模型的参数估计

于是根据隐含条件 $p(y=0 \mid \boldsymbol{x})+p(y=1 \mid \boldsymbol{x})=1$

得到

$$p(y=1 \mid \boldsymbol{x})=\frac{e^{\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}+b}}{1+e^{\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}+b}}$$
$$p(y=0 \mid \boldsymbol{x})=\frac{1}{1+e^{\boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}+b}}$$

现在回到我们一开始的问题：$\boldsymbol{\omega}$和b学得之后，模型就得以确定。那么，如何确定？

类似最小二乘法中使MSE最小，由于我们现在处理的是概率问题，于是可以用概率论中的最大似然估计——尝试确定似然函数，并最大化似然估计。

极大似然法

首先，什么叫似然(likelihood)?

通俗地讲，就是通过样本的数据，反过来估计真实模型最可能的情况。

比如抛一枚硬币，十次全是正面朝上，那么likelihood is这个硬币有问题。

再比如，一个盒子里有黑球和白球，我们需要考察黑球和白球的比例，这时我们对盒中的球进行100次有放回的抽取，有70次抽到黑球，30次抽到白球，那么likelihood is黑球和白球的比例是7:3。

极大似然法

于是，我们的核心要素在于：令每个样本属于其真实标记的概率越大越好。

建立似然函数：

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

连乘操作易造成下溢，通常取对数似然(log-likelihood)：

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

极大似然法

类似地，将$\boldsymbol{\omega}$和b吸收到$\beta=(\boldsymbol{\omega};b)$中，以及令$\widehat{\boldsymbol{x}}=(\boldsymbol{x};1)$，我们就可以将上式中的似然项重写为

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

$$p(y_i \mid \boldsymbol{x}_i;\boldsymbol{w},b)=p(y_i \mid \hat{\boldsymbol{x}}_i;\boldsymbol{\beta})=\begin{cases} p_1(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta}) & y_i=1 \\ p_0(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta}) & y_i=0 \end{cases} = y_i p_1(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta})+(1-y_i)p_0(\hat{\boldsymbol{x}}_i;\boldsymbol{\beta})$$

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

极大似然法

于是，对数似然函数就变成了

此步推导仍然参见南瓜书

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{m} ln[y_i p_1(\hat{\boldsymbol{x}}_i; \boldsymbol{\beta}) + (1 - y_i) p_0(\hat{\boldsymbol{x}}_i; \boldsymbol{\beta})] = \sum_{i=1}^{m}\left(y_i \boldsymbol{\beta}^{\mathrm{T}} \hat{\boldsymbol{x}}_i - \ln\left(1 + e^{\boldsymbol{\beta}^{\mathrm{T}} \hat{\boldsymbol{x}}_i}\right)\right)$$

于是最大化 $\sum_{i=1}^{m}\left(y_i \boldsymbol{\beta}^{\mathrm{T}} \hat{\boldsymbol{x}}_i - \ln\left(1 + e^{\boldsymbol{\beta}^{\mathrm{T}} \hat{\boldsymbol{x}}_i}\right)\right)$ 等价于最小化 $\sum_{i=1}^{m}\left(-y_i \boldsymbol{\beta}^{\mathrm{T}} \hat{\boldsymbol{x}}_i + \ln\left(1 + e^{\boldsymbol{\beta}^{\mathrm{T}} \hat{\boldsymbol{x}}_i}\right)\right)$

即 $\boldsymbol{\beta}^* = \underset{\boldsymbol{\beta}}{\arg\min} \ell'(\boldsymbol{\beta})$

根据凸优化理论，即可使用一些经典的数值优化算法如梯度下降法、牛顿法等求得最优解。

简单理解梯度下降法、牛顿法

在这里做一个抛砖引玉。梯度下降法、牛顿法等数值优化解法，可以简单地理解为“下山”的过程，也常用于有监督学习，一般考虑如下优化问题：

$$\min_{w} J(W)$$

其中，J为目标函数，也叫做损失函数，W为要学习的权重，是一个多维向量。

目标函数$J(W)$可以按泰勒展开，如果只取泰勒展开的二阶项，则$J(W)$可以近似表达为W的二次函数，即：

$$J(W) \approx J(W_0) + g^T(W - W_0) + \frac{1}{2}(W - W_0)^T H (W - W_0)$$

梯度向量

Hessian矩阵

梯度下降法

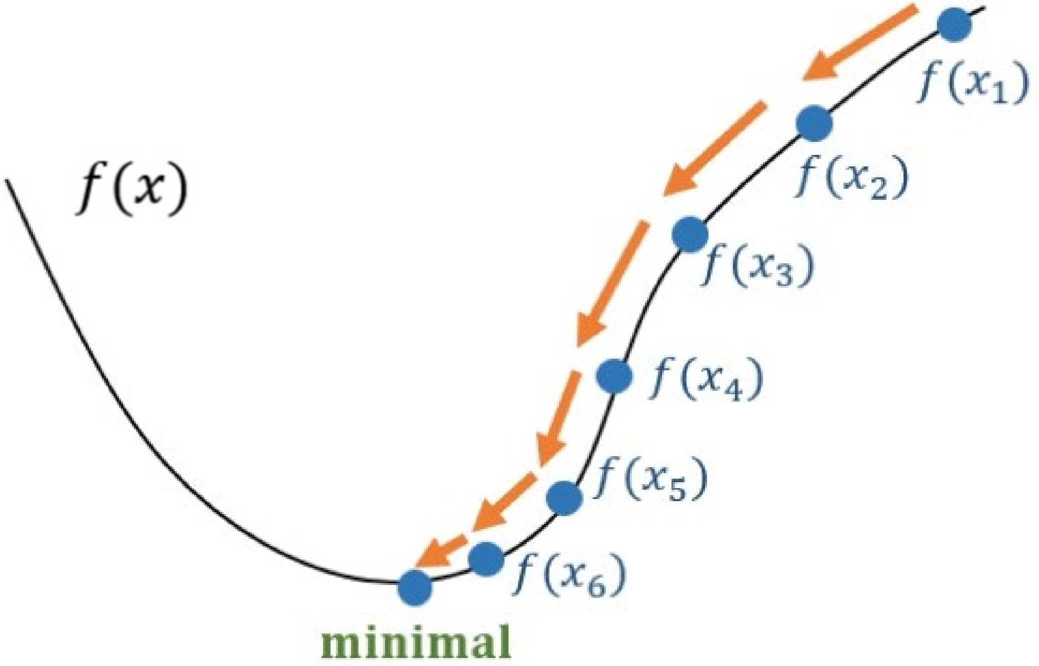
考虑下山问题：怎么样下山最快？沿着坡最大的方向往下走。

梯度是方向导数取得最大值的方向，直观理解就是“最陡峭的地方”，所以梯度下降法也叫“最速下降法”。

所以其思想是在梯度负方向上，前进一小步

$$W_{t+1} = W_t - l_r * g$$

其中 l_r 是学习率，相当于每步走“$l_r * \|g\|$”



牛顿法

而牛顿法是一个二阶过程，更“精确”。对二阶展开式求导得：

$$J(W) \approx J(W_0) + g^T(W - W_0) + \frac{1}{2}(W - W_0)^T H(W - W_0)$$

$$J(W) = J(W_0) + g^{\mathrm{T}}W - g^{\mathrm{T}}W_0 + \frac{1}{2}\left(W^{\mathrm{T}}H - W_0^{\mathrm{T}}H\right)(W - W_0)$$

$$J(W) = J(W_0) + g^{\mathrm{T}}W - g^{\mathrm{T}}W_0 + \frac{1}{2}(W^{\mathrm{T}}HW - W^{\mathrm{T}}HW_0 - W_0^{\mathrm{T}}HW + W_0^{\mathrm{T}}HW_0)$$

$$J'(W) = g + \frac{1}{2}\left(HW + H^{\mathrm{T}}W - HW_0 - H^{\mathrm{T}}W_0\right)$$

$$J'(W) = g + (HW - HW_0) = g + H(W - W_0)$$

令 $J'(W) = 0$，得 $W - W_0 = -H^{-1}g$

$\Longrightarrow$ $W = W_0 - H^{-1}g$ 二阶展开下的“极值点”

牛顿法

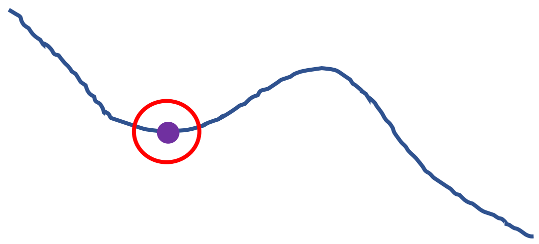
$$W = W_0 - H^{-1}g$$

$d = -H^{-1}g$ 叫做“牛顿方向”。故仍取 l_r 为学习率

则迭代公式为：

$$W_{t+1} = W_t + l_r * d$$

牛顿法迭代速度更快（比如对于特定的二次函数，甚至可以一步到位），但更容易陷入局部最小值：



当然了，梯度下降、牛顿法等也并不是唯一的优化方式，在此只是抛砖引玉。

极大似然法–结论

以牛顿法为例，其第$t+1$轮迭代解的更新公式为

$$\boldsymbol{\beta}^{t+1}=\boldsymbol{\beta}^{t}-\left(\frac{\partial^2 \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^{\mathrm{T}}}\right)^{-1} \frac{\partial \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}$$

其中关于β的一阶、二阶导数分别为

$$\begin{aligned}
\frac{\partial \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} &= -\sum_{i=1}^{m} \hat{\boldsymbol{x}}_i\left(y_i-p_1(\hat{\boldsymbol{x}}_i ; \boldsymbol{\beta})\right) \\
\frac{\partial^2 \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^{\mathrm{T}}} &= \sum_{i=1}^{m} \hat{\boldsymbol{x}}_i \hat{\boldsymbol{x}}_i^{\mathrm{T}} p_1(\hat{\boldsymbol{x}}_i ; \boldsymbol{\beta})\left(1-p_1(\hat{\boldsymbol{x}}_i ; \boldsymbol{\beta})\right)
\end{aligned}$$

3.4 线性判别分析

引入

仍然考虑一个二分类问题。

在一个三维空间中，有一张桌子，其上放着一个苹果和一个橙子。

已知苹果和橙子的点集（样本集），输入一个测试点（要么是苹果要么是橙子），如何判断其为苹果还是橙子？

——这就是一个三维向二维投影的过程

线性判别分析

线性判别分析(LDA, Linear Discriminant Analysis)是一种经典的线性学习方法，其思想非常朴素，即设法将样例投影到一条直线上，使得同类样例的投影点尽可能近、异类样例的投影点尽可能远离。

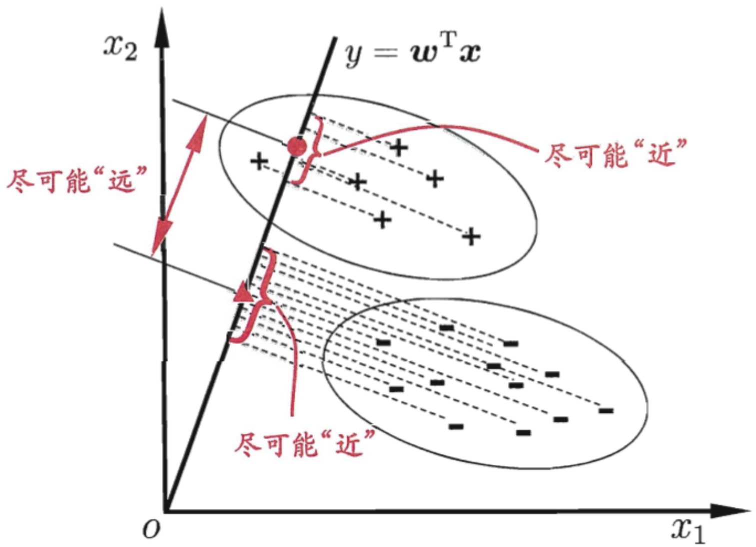
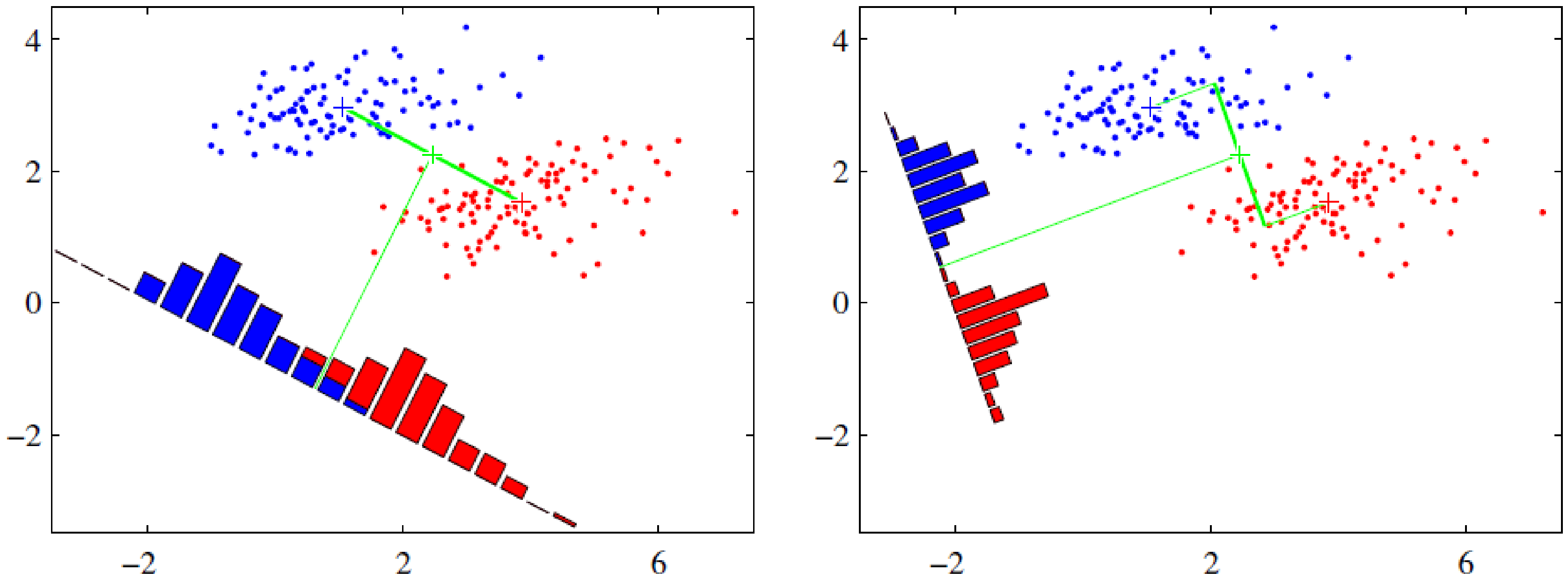


图 3.3 LDA 的二维示意图. “+”、“−”分别代表正例和反例, 椭圆表示数据簇的外轮廓, 虚线表示投影, 红色实心圆和实心三角形分别表示两类样本投影后的中心点.

线性判别分析

线性判别分析(LDA, Linear Discriminant Analysis)是一种经典的线性学习方法，其思想非常朴素，即设法将样例投影到一条直线上，使得同类样例的投影点尽可能近、异类样例的投影点尽可能远离。

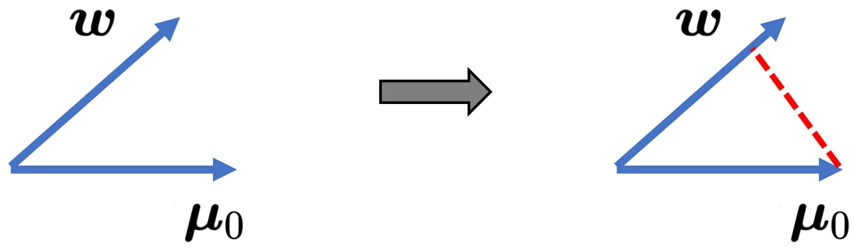


线性判别分析

更一般地，对于一个二分类问题，给定数据集 $\mathcal{D}=\{(\boldsymbol{x}_i,y_i)\}_{i=1}^{m}$，$y_i\in\{0,1\}$（注意，这里的输入就并不一定只是二维了）。

令X_i、$\boldsymbol{\mu}_i$、$\boldsymbol{\Sigma}_i$ 分别表示第$i\in\{0,1\}$ 类示例的集合、均值向量、协方差矩阵。

那么，样本中心（均值）向量在直线$\boldsymbol{\omega}$上的投影分别为$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_0$ 和 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_1$：



$$\boldsymbol{w}\cdot\boldsymbol{\mu}_0=\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_0$$

实际上投影值应该是 $\dfrac{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_0}{|\boldsymbol{w}|}$

但$|\boldsymbol{w}|$显然并不重要

线性判别分析

那么，两类样本的协方差则分别为 $\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{\Sigma}_0\boldsymbol{w}$ 和 $\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{\Sigma}_1\boldsymbol{w}$ ，简单推导如下：

$$\boldsymbol{\Sigma}_{\boldsymbol{i}} = \sum_{\boldsymbol{x}\in X_i}(\boldsymbol{x}-\boldsymbol{\mu}_i)(\boldsymbol{x}-\boldsymbol{\mu}_i)^{\mathrm{T}}$$

$$\Downarrow$$

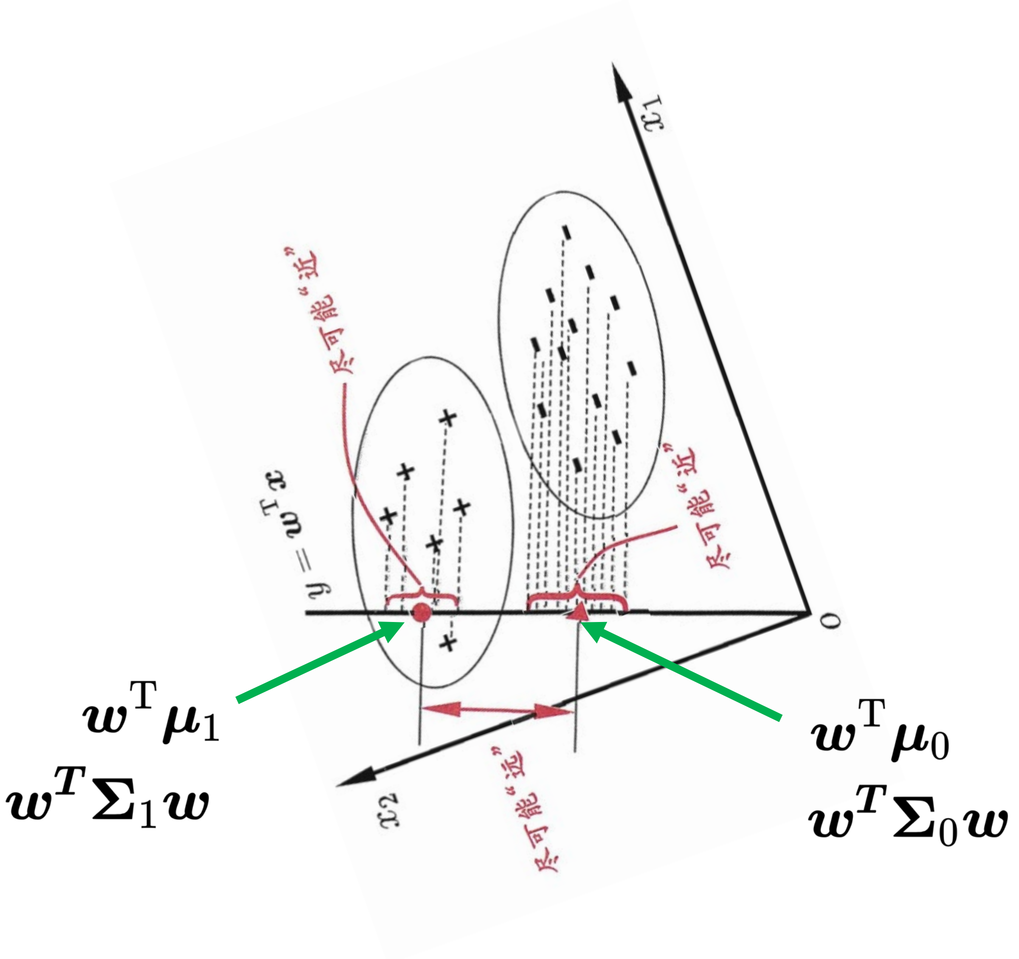
$$\boldsymbol{\Sigma}_i' = \sum_{\boldsymbol{x}\in X_i}\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_i\right)\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_i\right)^{\mathrm{T}}$$

$$\Rightarrow \boldsymbol{\Sigma}_{\boldsymbol{i}}' = \sum_{\boldsymbol{x}\in X_i}\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{\mu}_i)(\boldsymbol{x}-\boldsymbol{\mu}_i)^{\mathrm{T}}\boldsymbol{w}$$

$$\Rightarrow \boldsymbol{\Sigma}_i' = \boldsymbol{w}^{\mathrm{T}}\left(\sum_{\boldsymbol{x}\in X_i}(\boldsymbol{x}-\boldsymbol{\mu}_i)(\boldsymbol{x}-\boldsymbol{\mu}_i)^{\mathrm{T}}\right)\boldsymbol{w} = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_i\boldsymbol{w}$$

线性判别分析

由于直线是一维的，所以$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_0$、$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_1$、$\boldsymbol{w}^{T}\boldsymbol{\Sigma}_0\boldsymbol{w}$ 和 $\boldsymbol{w}^{T}\boldsymbol{\Sigma}_1\boldsymbol{w}$ 均为实数



线性判别分析

同样地，这里作除法也只是一种常用的方式

欲最大化的目标函数：

$$\uparrow J=\frac{\left\|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_0-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\mu}_1\right\|_2^2 \uparrow}{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_0\boldsymbol{w}+\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Sigma}_1\boldsymbol{w} \downarrow}$$

$$=\frac{\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)^{\mathrm{T}}\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{\Sigma}_0+\boldsymbol{\Sigma}_1)\boldsymbol{w}} \qquad =\frac{\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_b\boldsymbol{w}}{\boldsymbol{w}^{\mathrm{T}}\mathbf{S}_w\boldsymbol{w}}$$

定义“类间散度矩阵”(between-class scatter matrix)：

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

定义“类内散度矩阵”(within-class scatter matrix)：

$$\mathbf{S}_w=\boldsymbol{\Sigma}_0+\boldsymbol{\Sigma}_1=\sum_{\boldsymbol{x}\in X_0}(\boldsymbol{x}-\boldsymbol{\mu}_0)(\boldsymbol{x}-\boldsymbol{\mu}_0)^{\mathbf{T}}+\sum_{\boldsymbol{x}\in X_1}(\boldsymbol{x}-\boldsymbol{\mu}_1)(\boldsymbol{x}-\boldsymbol{\mu}_1)^{\mathrm{T}}$$

瑞利商

瑞利商(Rayleigh quotient)是指这样一个函数：

$$R(A, x) = \frac{x^H A x}{x^H x}$$

其中$\boldsymbol{x}$为非零向量，A为Hermitian矩阵：共轭转置矩阵和自己相等（在实数域内就退化为转置矩阵），共轭转置用上标H表示，也即是，$A^H = A$。

而瑞利商有一个重要的性质，即它的最大值等于矩阵A最大的特征值，而最小值等于矩阵A的最小的特征值：

$$\lambda_{\min} \leq \frac{x^H A x}{x^H x} \leq \lambda_{\max}$$

瑞利商

当$\boldsymbol{x}$是标准正交基时，瑞利商退化为：

$$R(A,x)=x^H Ax$$

这个形式在谱聚类和PCA中也都有出现，并且经常出现在降维和聚类任务中，先混个眼熟。

（实际上正交分解也是一种特殊的瑞利商）

广义瑞利商

广义瑞利商(generalized Rayleigh quotient)是指这样一个函数：

$$R(A,B,x)=\frac{x^HAx}{x^HBx}$$

之所以叫广义瑞利商，是因为我们可以通过一些操作将其转化为瑞利商。这个操作叫标准化，只需令:$x=B^{-\frac{1}{2}}x'$

$$\text{分母：}\quad x^HBx=x'^H\left(B^{-\frac{1}{2}}\right)^HBB^{-\frac{1}{2}}x'=x'^HB^{-\frac{1}{2}}BB^{-\frac{1}{2}}x'=x'^Hx'$$

$$\text{分子：}\quad x^HAx=x'^HB^{-\frac{1}{2}}AB^{-\frac{1}{2}}x'$$

广义瑞利商

此时我们将广义瑞利商转化为：

$$R(A,B,x)=\frac{x^{H}Ax}{x^{H}Bx} \Longrightarrow R(A,B,x')=\frac{x'^{H}B^{-\frac{1}{2}}AB^{-\frac{1}{2}}x'}{x'^{H}x'}$$

那么广义瑞利商的最小值、最大值对应着$B^{-\frac{1}{2}}AB^{-\frac{1}{2}}$的最小、最大特征值。而$B^{-\frac{1}{2}}AB^{-\frac{1}{2}}$与$B^{-1}A$同特征值。

线性判别分析

故最大化目标J的极大值与$\mathbf{S}_w^{-1}\mathbf{S}_b$ 相关，也即有：

$$\mathbf{S}_w^{-1}\mathbf{S}_b\boldsymbol{w}=\lambda\boldsymbol{w}$$

注意到，$(\mu_0-\mu_1)^{\mathrm{T}}\boldsymbol{w}$ 是标量，故实际上

$$\mathbf{S}_b\boldsymbol{w}=\lambda(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)$$

不妨令两者的λ相等，得到

$$\boldsymbol{w}=\mathbf{S}_w^{-1}(\boldsymbol{\mu}_0-\boldsymbol{\mu}_1)$$

关于式(3.37)的得到，我认为这里和书上是相通的，关于书上的得到过程可以参见南瓜书。

奇异值分解

考虑到数值解的稳定性，实践中通常对$\mathbf{S}_w$进行奇异值分解。

奇异值分解(SVD, sigular value decomposition)解决的是对一个非方阵分解的问题。

（然而$\mathbf{S}_w$不是方阵吗，为啥不用LU、LUP、Cholesky这些分解方式？？）

即有一个$m \times n$的实数矩阵A，我们想要把它分解成如下的形式：

$$A = U\Sigma V^T$$

其中U和V均为单位正交阵，即有$UU^T = I$和$VV^T = I$，U称为左奇异矩阵，V称为右奇异矩阵，Σ仅在主对角线上有值，我们称它为奇异值，其它元素均为0。上面矩阵的维度分别为$U \in R^{m\times m}$，$\Sigma \in R^{m\times n}$, $V \in R^{n\times n}$。

奇异值分解

一般地，Σ有以下形式：

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \end{bmatrix}_{m \times n}$$

显然，特征值分解是一种特殊的SVD。

应用：PCA、图像压缩等

奇异值分解

$$A = U\Sigma V^T$$

想要求解U、V，只需要以下简单的操作：

都是对称阵

$$AA^T = U\Sigma V^T V\Sigma^T U^T = \boxed{U\Sigma\Sigma^T U^T}$$

$$A^TA = V\Sigma^T U^T U\Sigma V^T = \boxed{V\Sigma^T\Sigma V^T}$$

正交分解即可

$$\Sigma\Sigma^T = \begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \ddots \end{bmatrix}_{m\times m} \qquad \Sigma^T\Sigma = \begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \ddots \end{bmatrix}_{n\times n}$$

由特殊到一般：多分类LDA

定义全局散度矩阵：

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

$$\mathbf{S}_w = \sum_{i=1}^{N} \sum_{\boldsymbol{x} \in X_i} (\boldsymbol{x} - \boldsymbol{\mu}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)^{\mathrm{T}}$$

$$\mathbf{S}_b = \mathbf{S}_t - \mathbf{S}_w = \sum_{i=1}^{N} m_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^{\mathrm{T}}$$

类似于一种权重

由特殊到一般：多分类LDA

同样地，这里取迹只是一种常用的方式，也有取$\prod_{\mathrm{diag}}(\cdot)$的

显然，使用 $\mathbf{S}_b$ 、 $\mathbf{S}_t$ 、 $\mathbf{S}_w$ 三者之二就可以完成LDA，常见的是采用优化目标

$$J = \frac{\mathrm{tr}\left(\mathbf{W}^{\mathrm{T}}\mathbf{S}_b\mathbf{W}\right)}{\mathrm{tr}\left(\mathbf{W}^{\mathrm{T}}\mathbf{S}_w\mathbf{W}\right)} \qquad \mathbf{W} = \max_{\mathbf{W}} \frac{\mathrm{tr}\left(\mathbf{W}^{\mathrm{T}}\mathbf{S}_b\mathbf{W}\right)}{\mathrm{tr}\left(\mathbf{W}^{\mathrm{T}}\mathbf{S}_w\mathbf{W}\right)}$$

$\boldsymbol{W}$的闭式解则是 $\mathbf{S}_w^{-1}\mathbf{S}_b$ 的d'个最大非零广义特征值所对应的特征向量组成的矩阵，且有$d' \leq N-1$， $\boldsymbol{W}$又称投影矩阵。

理解：为什么是$N-1$?

①从投影的角度

②从线性相关的角度

再看引入

所以，LDA也会被用于监督降维

即我们不必输入全部三个属性，只需输入部分属性就可以判断其归属的分类。

——比如三维向二维的投影

此外还有一些技术如PCA等也可以用于降维。

3.5 多分类学习

引入

虽然有些二分类学习方法可以直接推广到多分类，比如LDA，但更多地，我们是基于一些基本策略（Reduction），利用二分类学习器解决多分类问题。

现在考虑一个多分类问题：

假设全世界只有四大天王这四个明星，大过年的时候串亲戚，众亲戚对着远房某个亲戚的小孩说道：“诶，你家小孩长得真像（黎明/张学友/郭富城/刘德华）啊！”这时，每一个亲戚就好像是一个分类器(classifier)。

拆解法

不失一般性，考虑N个类别$C_1, C_2, \dots, C_N$，多分类学习的基本思路是“拆解法”，即将多分类任务拆为若干个二分类任务求解。最经典的拆分策略有三种：一对一(OvO)、一对其余(OvR)和多对多(MvM)。

OvO：会产生 $C_n^2 = N(N-1)/2$ 个二分类任务，N个类别两两配对：比如四种分类就对应6个任务

OvR：亦称OvA，一个为正类，其余全为反类

MvM：通过某些特殊的设计构造正、反类

显然OvO和OvR都是MvM的特殊情况。

再看引入：OvO

（4个分类对应C_4^2 =6个分类器）

亲戚1：“只认识黎明、张学友和像他们的人” $\Longrightarrow$ 你家小孩长得像黎明

亲戚2：“只认识黎明、郭富城和像他们的人” $\Longrightarrow$ 你家小孩长得像郭富城

亲戚3：“只认识黎明、刘德华和像他们的人” $\Longrightarrow$ 你家小孩长得像黎明

亲戚4：“只认识张学友、郭富城和像他们的人” $\Longrightarrow$ 你家小孩长得像郭富城

亲戚5：“只认识张学友、刘德华和像他们的人” $\Longrightarrow$ 你家小孩长得像张学友

亲戚6：“只认识郭富城、刘德华和像他们的人” $\Longrightarrow$ 你家小孩长得像郭富城

最后，众亲戚投票：你家小孩长得像郭富城

投票的规则是什么？虽然不会有“五五开”，但“三分天下”怎么办？

再看引入：OvR

（4个分类对应4个分类器）

亲戚1：“见过很多人，但特喜欢黎明，对人分类都以他为标准” ⟹ 你家小孩长得不像黎明

亲戚2：“见过很多人，但特喜欢张学友，对人分类都以他为标准” ⟹ 你家小孩长得不像张学友

亲戚3：“见过很多人，但特喜欢郭富城，对人分类都以他为标准” ⟹ 你家小孩长得像郭富城

亲戚4：“见过很多人，但特喜欢刘德华，对人分类都以他为标准” ⟹ 你家小孩长得不像刘德华

最后，只有亲戚3表达了肯定的回答：你家小孩长得像郭富城

若有多个分类器预测为正类，则考虑各分类器置信度，选择置信度最大的类别标记作为分类结果

（比如某个亲戚年纪更大、威望更高），但对应到样本，有没有特定的评判标准？

再看引入：MvM

实际情况更像是MvM：每个亲戚只对个别的一个或几个明星情有独钟、并且不是所有人都能认识所有明星，这就对应了三元ECOC码中的“停用类”。

纠错输出码(ECOC, Error Correcting Output Codes)是一种最常用的MvM技术，其将编码的思想引入类别拆分，并尽可能在解码过程中具有容错性。

其工作过程主要分为两步：编码和解码。

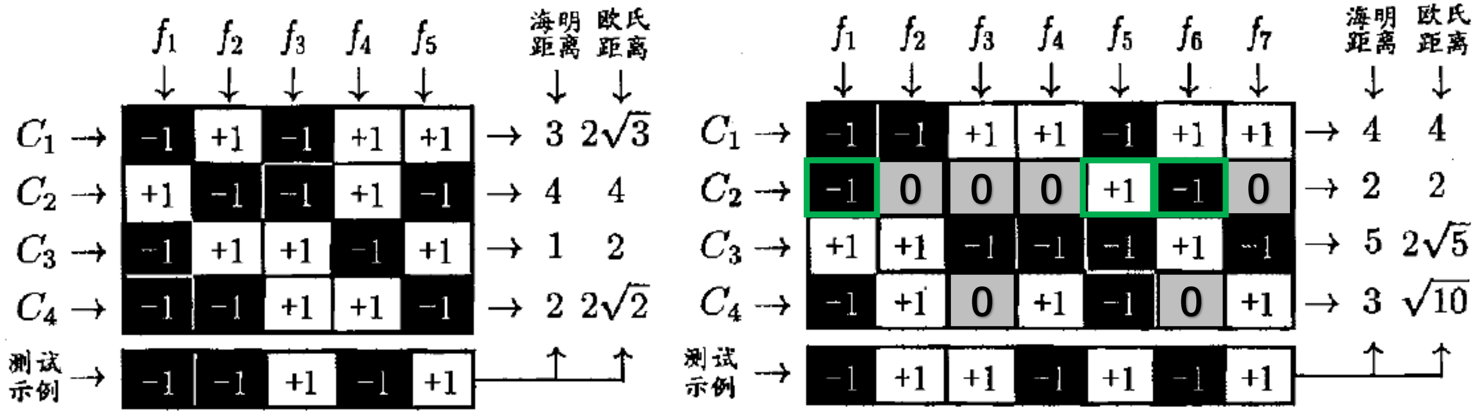
编码：对N个类别做M次划分，每次划分将一部分类别划为正类，一部分划为反类（，或将一部分划为停用类），产生M个训练集，训练出M个分类器。

解码：将预测编码与各自的编码进行比较，返回距离最小值作为最终预测结果。

再看引入：MvM

（可能对应多个分类器，但x元ECOC码的N分类问题，其上限是x^N个）

理解：



(a) 二元 ECOC 码　　(b) 三元 ECOC 码

⟹ 认识张学友的人，都觉得他长得挺像张学友（甚至2/3觉得只像他）

⟹ 亲戚2、3对于像不像黎明的问题形成了鲜明的反差，总之有人觉得像，有人觉得不像

⟹ 亲戚1、3、6和7都认为完全不像郭富城，所以3的海明距离、欧氏距离都最大，很符合直觉

⟹ 对于刘德华，对比张学友来看一样也有三个认同像的，但是有两个完全的反对票

多分类问题–结论

例子只是抛砖引玉，但我们在实际中完全可以参数化脸部模型（比如FIFA等球员脸型建模）

OvO的存储开销和测试时间开销通常比OvR大，但是在类别较多时训练时间开销通常更小；

OvO在训练时，每个分类器仅用到部分样例，OvR则为全部样例；

多数情况下，两者预测性能差不多；

ECOC编码对分类器错误有一定的容忍和修正能力；

ECOC编码越长，纠错能力越强，但也意味着所需训练的分类器越多，计算、存储开销都会增大；

组合数目也是有限的，码长超过一定范围后就失去了意义；

通常，机器学习涉及很多因素，并不是说编码理论性质越好，分类性能越好。

3.6 类别不平衡问题

类别不平衡问题

类别不平衡(class-imbalance)就是指分类任务中不同类别的训练样例数目差别很大的情况。

比如在之前的胃癌问题中：如果对学习器输入了998份胃癌患者样例、2份非胃癌患者样例，那么如果我们不加以“再缩放”(rescaling)的话，学习器在拿到大部分新样本时，很可能都判以患者。

所以rescaling解决的主要是“训练集未必是真实样本总体的无偏采样”的问题。

影响最终训练效果的因素主要是：①采样层面，②学习模型方面

所以，如果采样层面是无偏估计——那么不必再缩放；

如果采样层面就有偏差（或者并不清楚）——那么直接进行再缩放。

类别不平衡问题

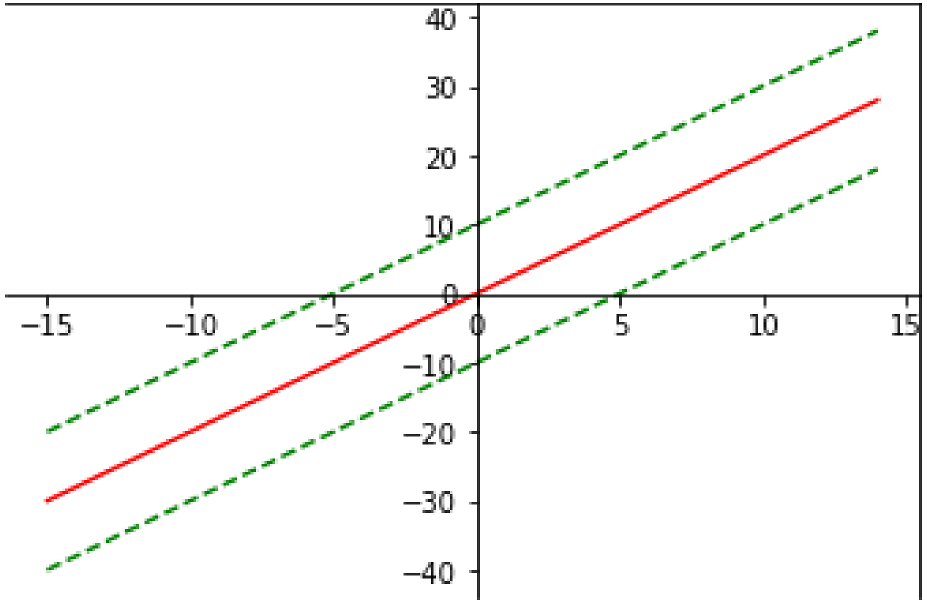
对于对数几率模型，我们有：

$$\frac{p(y=1 \mid \boldsymbol{x})}{p(y=0 \mid \boldsymbol{x})} = e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b}$$

$$\frac{y'}{1-y'} = \frac{p(y=1 \mid \boldsymbol{x})}{p(y=0 \mid \boldsymbol{x})} \times \frac{m^-}{m^+} = e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b} \times \frac{m^-}{m^+}$$

$$= e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b} \times e^{\ln \frac{m^-}{m^+}} = e^{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b+\ln \frac{m^-}{m^+}}$$

所以实质上是在线性模型上增加了一个截距因子。



类别不平衡问题

一般有三类做法：

①欠采样(undersampling，也叫下采样)，去除一些反例使得正、反例数目接近，再学习。

→时间开销较小，“全局来看不会丢失重要信息”，代表如EasyEnsemble；

②过采样(oversampling，也叫上采样)，增加一些正例使得正、反例数目接近，再学习。

→不能简单地对初始样本重复采样，否则过拟合，可以通过插值，代表如SMOTE；

③阈值移动(threshold-moving)，即采用再缩放的基本策略。

→也是代价敏感学习的基础。