机器学习

第二章 线性模型

2022年8月

目录

1. 线性回归

1. 1 回归的概念

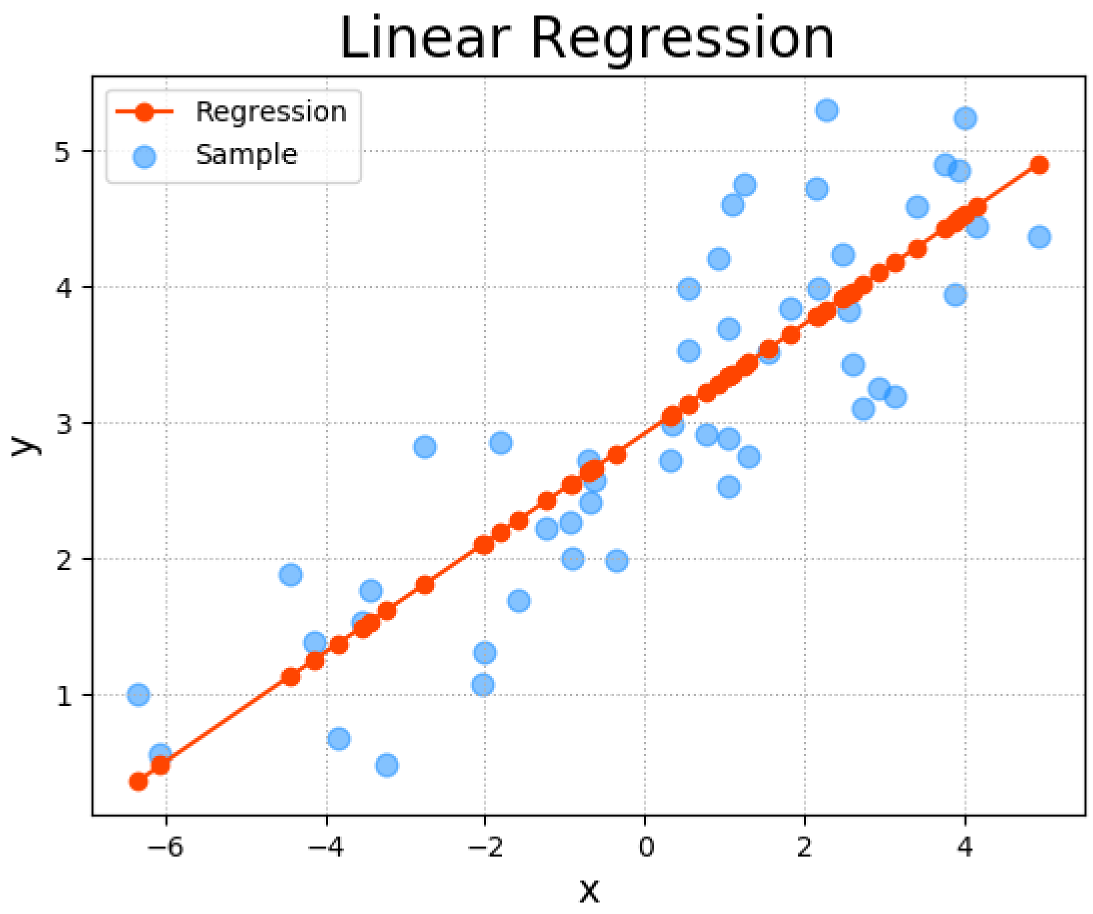
监督学习分为回归和分类

- ✓ 回归（Regression、Prediction） **标签连续**
  - ✓ 如何预测上海浦东的房价？
  - ✓ 未来的股票市场走向？
- ✓ 分类（Classification） **标签离散**
  - ✓ 身高1.85m，体重100kg的男人穿什么尺码的T恤？
  - ✓ 根据肿瘤的体积、患者的年龄来判断良性或恶性？

1.1 线性回归-概念

线性回归（Linear Regression）

是一种通过属性的线性组合来进行预测的线性模型，其目的是找到一条直线或者一个平面或者更高维的超平面，使得预测值与真实值之间的误差最小化。



1.1 线性回归-基本形式

- 线性模型一般形式

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

$\boldsymbol{x} = (x_1; x_2; \ldots; x_d)$是由属性描述的示例，其中$x_i$是$\boldsymbol{x}$在第 i 个属性上的取值

- 向量形式

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

其中 $\boldsymbol{w} = (w_1; w_2; \ldots; w_d)$

1.1 线性回归-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

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

1.1 线性回归-简单的线性回归

m 代表训练集中样本的数量

x 代表特征/输入变量

y 代表目标变量/输出变量

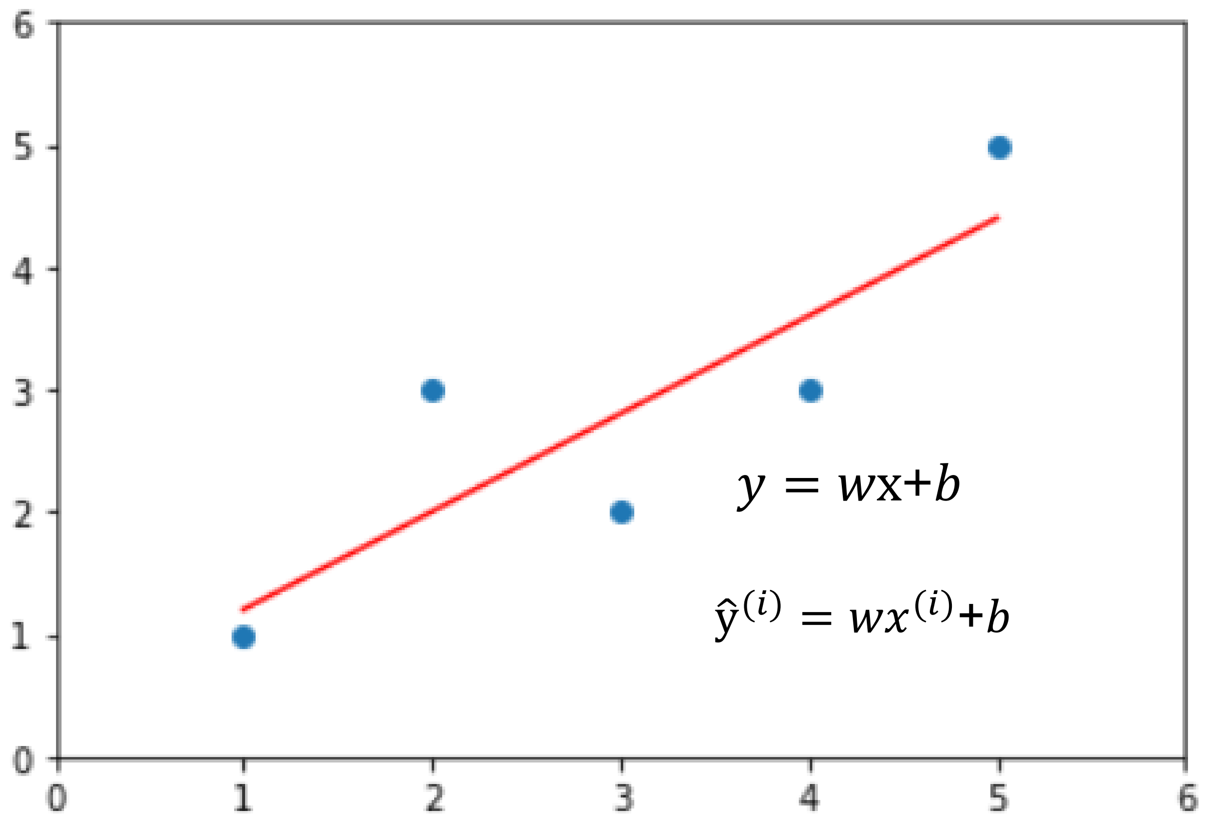
(x, y) 代表训练集中的样本

$(x^{(i)}, y^{(i)})$ 代表第i个观察样本

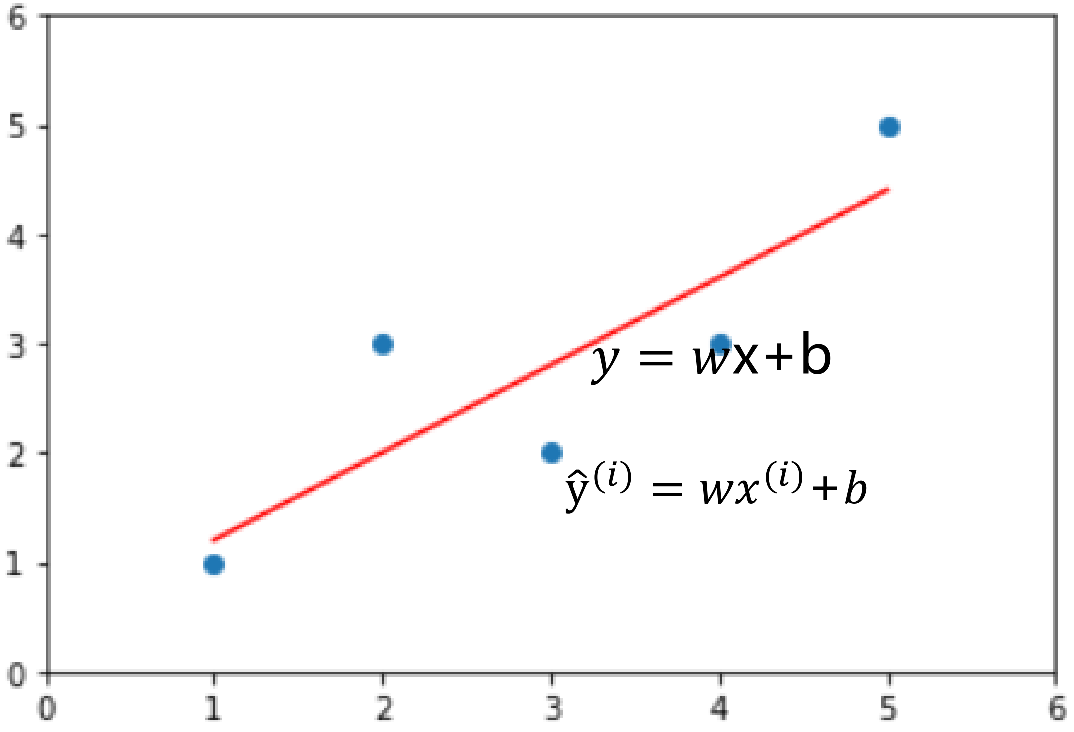
$\hat{y}$ 代表预测的值

w 代表斜率

b 代表截距



1.1 线性回归-简单的线性回归



简单线性回归：样本特征只有一个

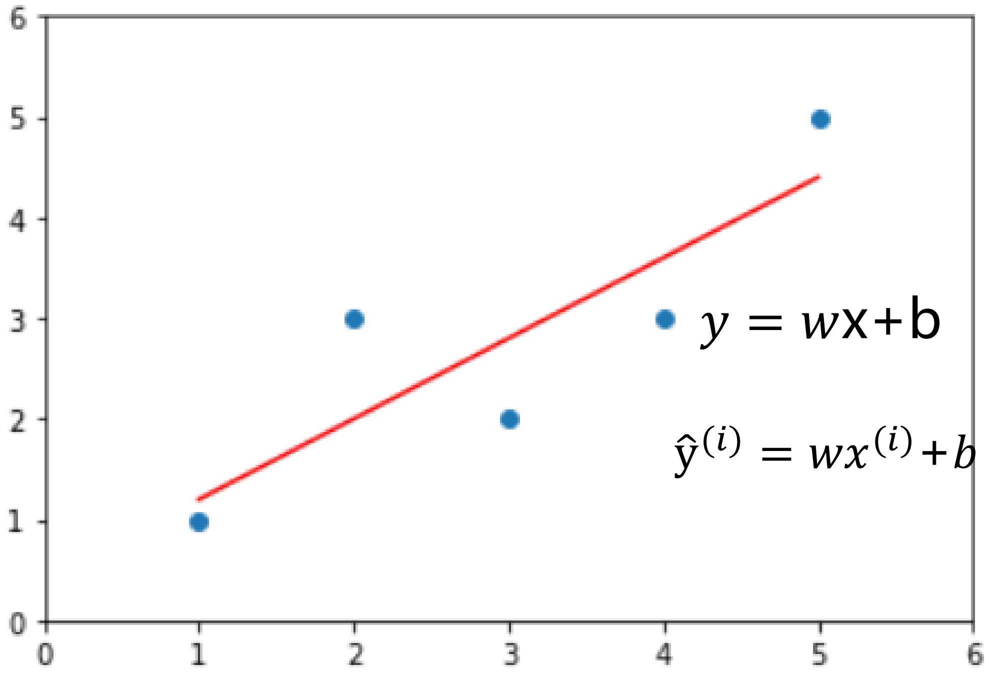
寻找最拟合的曲线: $y = w$x+b

对每一个样本的预测值：$\hat{y}^{(i)} = wx^{(i)} + b$

真值：y^i

想要寻找最佳拟合的直线方程,就应该使得**预测值和真实值要越来越近**(差距要尽量小)。

1.1 线性回归-简单的线性回归



y^i与$\hat{y}^{(i)}$尽可能接近

$|y^i - \hat{y}^{(i)}|$ 尽可能小

考虑所有样本：

$$\sum_{i=1}^{m}(y^{(i)} - \hat{y}^{(i)})^2$$

使$\sum_{i=1}^{m}(y^{(i)} - \hat{y}^{(i)})^2$尽可能小

1.1 线性回归-简单的线性回归

继续转化：

$$J(w,b)=\sum_{i=1}^{m}(y^{(i)}-wx^{(i)}-b)^2 \Longrightarrow \text{尽可能小}$$

典型的**最小二乘法问题**：最小化误差的平方

目标：找到w、b使 $J(w,b)$ 最小 $\Longrightarrow$

$$\begin{cases}\dfrac{\partial J(w,b)}{\partial w}=0\\[2ex]\dfrac{\partial J(w,b)}{\partial b}=0\end{cases}$$

1.1 线性回归-简单的线性回归

$$J(w,b)=\sum_{i=1}^{m}(y^{(i)}-wx^{(i)}-b)^2 \qquad \frac{\partial J(w,b)}{\partial \mathrm{b}}=0$$

$$\frac{\partial J(w,b)}{\partial \mathrm{b}}=\sum_{i=1}^{m}2(y^{(i)}-wx^{(i)}-b)(-1)=0$$

$$\sum_{i=1}^{m}(y^{(i)}-wx^{(i)}-b)=0$$

$$\sum_{i=1}^{m}y^{(i)}-w\sum_{i=1}^{m}x^{(i)}-mb=0$$

$$mb=\sum_{i=1}^{m}y^{(i)}-w\sum_{i=1}^{m}x^{(i)} \Longrightarrow \boldsymbol{b=\bar{y}-w\bar{x}}$$

1.1 线性回归-简单的线性回归

$$J(w,b)=\sum_{i=1}^{m}(y^{(i)}-wx^{(i)}-b)^2$$

$$\frac{\partial J(w,b)}{\partial w}=0$$

$$\frac{\partial J(w,b)}{\partial w}=\sum_{i=1}^{m}2(y^{(i)}-wx^{(i)}-b)(-x^{(i)})=0$$

$$\sum_{i=1}^{m}(y^{(i)}-wx^{(i)}-b)\,x^{(i)}=0$$

$$\sum_{i=1}^{m}(y^{(i)}-wx^{(i)}-\bar{y}+w\bar{x})\,x^{(i)}=0$$

$$\sum_{i=1}^{m}(x^{(i)}y^{(i)}-w(x^{(i)})^2-x^{(i)}\bar{y}+wx^{(i)}\bar{x})=0$$

1.1 线性回归-简单的线性回归

$$\sum_{i=1}^{m}(x^{(i)}y^{(i)} - w(x^{(i)})^2 - x^{(i)}\bar{y} + wx^{(i)}\bar{x}) = 0$$

$$\sum_{i=1}^{m}(x^{(i)}y^{(i)} - x^{(i)}\bar{y} - w(x^{(i)})^2 + wx^{(i)}\bar{x}) = 0$$

$$\sum_{i=1}^{m}(x^{(i)}y^{(i)} - x^{(i)}\bar{y}) - \sum_{i=1}^{m}\left(w\left(x^{(i)}\right)^2 - wx^{(i)}\bar{x}\right) = 0$$

$$\sum_{i=1}^{m}(x^{(i)}y^{(i)} - x^{(i)}\bar{y}) - w\sum_{i=1}^{m}\left(\left(x^{(i)}\right)^2 - x^{(i)}\bar{x}\right) = 0$$

$$w = \frac{\sum_{i=1}^{m}(x^{(i)}y^{(i)} - x^{(i)}\bar{y})}{\sum_{i=1}^{m}\left(\left(x^{(i)}\right)^2 - x^{(i)}\bar{x}\right)}$$

1.1 线性回归-简单的线性回归

$$w=\frac{\sum_{i=1}^{m}(x^{(i)}y^{(i)}-x^{(i)}\bar{y})}{\sum_{i=1}^{m}\left(\left(x^{(i)}\right)^2-x^{(i)}\bar{x}\right)}$$

$$\sum_{i=1}^{m}x^{(i)}\bar{y}=\bar{y}\sum_{i=1}^{m}x^{(i)}=m\bar{y}\,\bar{x}=\bar{x}\sum_{i=1}^{m}y^{(i)}=\sum_{i=1}^{m}\bar{x}y^{(i)}$$

又因为：

$$m\bar{x}\bar{y}=\sum_{i=1}^{m}\bar{x}\bar{y}$$

所以：

$$\boldsymbol{w}=\frac{\sum_{i=1}^{m}(x^{(i)}y^{(i)}-x^{(i)}\bar{y}+\bar{x}\bar{y}-\bar{x}y^{(i)})}{\sum_{i=1}^{m}\left(\left(x^{(i)}\right)^2-x^{(i)}\bar{x}-x^{(i)}\bar{x}+\bar{x}^2\right)}$$

$$=\frac{\sum_{i=1}^{m}\left(\boldsymbol{x}^{(i)}-\bar{\boldsymbol{x}}\right)\left(\boldsymbol{y}^{(i)}-\bar{\boldsymbol{y}}\right)}{\sum_{i=1}^{m}\left(\boldsymbol{x}^{(i)}-\bar{\boldsymbol{x}}\right)^2}$$

1.1 线性回归-简单的线性回归

求解可得：

$$w = \frac{\sum_{i=1}^{m}\left(x^{(i)} - \bar{x}\right)\left(y^{(i)} - \bar{y}\right)}{\sum_{i=1}^{m}\left(x^{(i)} - \bar{x}\right)^2}$$

$$b = \bar{y} - w\bar{x}$$

1. 1 线性回归

m 代表训练集中样本的数量

n 代表特征的数量

x 代表特征/输入变量

y 代表目标变量/输出变量

(x, y) 代表训练集中的样本

$(x^{(i)}, y^{(i)})$ 代表第i个观察样本

h 代表学习算法的解决方案或函数也称为假设（**hypothesis**）

$\hat{y} = h(x)$,代表预测的值

建筑面积	总层数	楼层	实用面积	房价
143.7	31	10	105	**36200**
162.2	31	8	118	**37000**
199.5	10	10	170	**42500**
96.5	31	13	74	**31200**
……	……	……	……	……

$x^{(i)}$是特征矩阵中的第i行，是一个**向量**。

上图的：$x^{(2)} = \begin{bmatrix} 162.2 \\ 31 \\ 8 \\ 118 \end{bmatrix}$ $y^{(2)}$ =37000

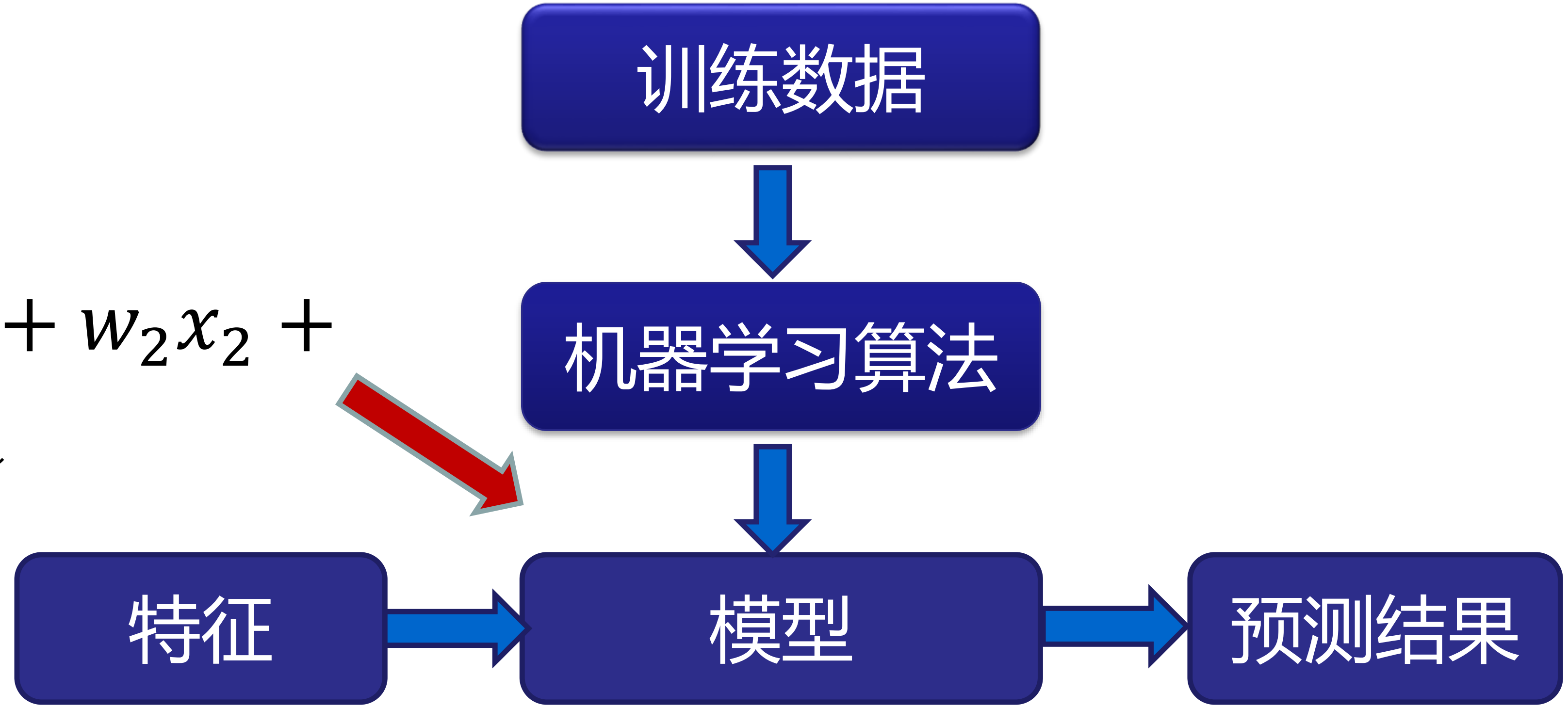
$x_j^{(i)}$代表特征矩阵中第 i 行的第 j 个特征

上图的$x_2^{(2)} = 31, x_3^{(2)} = 8$

1. 1 线性回归-算法流程

x 和y 的关系

$$h(x) = w_0 + w_1x_1 + w_2x_2 + \ldots + w_nx_n$$



可以设$x_0 = 1$

则：$h(x) = w_0x_0 + w_1x_1 + w_2x_2 + \ldots + w_nx_n = w^TX$

注意：若表达式 $h(x) = w_0 + w_1x_1 + w_2x_2 + \ldots + w_nx_n + b$， 则$b$可以融入到$w_0$

1.1 线性回归-算法流程

$$h(x) = w_0 + w_1x_1 + w_2x_2 + \ldots + w_nx_n$$

损失函数采用平方和损失：

$$l(x^{(i)}) = \frac{1}{2}(h(x^{(i)}) - y^{(i)})^2$$

要找到一组 $w(w_0, w_1, w_2, \ldots, w_n)$ ，

使得$J(w) = \frac{1}{2}\sum_{i=1}^{m}(h(x^{(i)}) - y^{(i)})^2$

（残差平方和）最小

损失函数(Loss Function)度量单样本预测的错误程度，损失函数值越小，模型就越好。常用的损失函数包括：0-1损失函数、平方损失函数、绝对损失函数、对数损失函数等。

代价函数(Cost Function)度量全部样本集的平均误差。常用的代价函数包括均方误差、均方根误差、平均绝对误差等。

目标函数(Object Function)代价函数和正则化函数，最终要优化的函数。

备注：损失函数的系数1/2是为了便于计算，使对平方项求导后的常数系数为1，这样在形式上稍微简单一些。有些教科书把系数设为1/2，有些设置为1，这些都不影响结果。

1.1 线性回归-最小二乘法(LSM)

要找到一组 $w(w_0, w_1, w_2, \ldots, w_n)$，使得$J(w) = \frac{1}{2}\sum_{i=1}^{m}(h(x^{(i)}) - y^{(i)})^2$

(残差平方和) 最小，即最小化：$\frac{\partial J(w)}{\partial w}$

将向量表达形式转为矩阵表达形式，则有$J(w) = \frac{1}{2}(Xw - Y)^2$，其中$X$为$m$行$n+1$列的矩阵（$m$为样本个数，$n$为特征个数），$w$为$n+1$行1列的矩阵(包含了$w_0$)，$Y$为$m$行1列的矩阵，则$J(w) = \frac{1}{2}(Xw - Y)^2 = \frac{1}{2}(Xw - Y)^T(Xw - Y)$

$$X = \begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & \ldots & x_n^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & \ldots & x_n^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_1^{(m)} & x_2^{(m)} & x_3^{(m)} & \ldots & x_n^{(m)} \end{bmatrix} \quad Y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \ldots \\ y^{(m)} \end{bmatrix}$$

需要用到向量平方的性质:

$$\sum_i z_i^2 = z^T z$$

1.1 线性回归-最小二乘法(LSM)

为最小化，接下来对$J(w)$偏导，

$$\frac{\partial J(w)}{\partial w}=\frac{1}{2}\frac{\partial}{\partial w}(Xw-Y)^T(Xw-Y)\ =\frac{1}{2}\frac{\partial}{\partial w}(w^TX^TXw-Y^TXw-w^TX^TY+Y^TY)$$

由于中间两项互为转置：

$$\frac{\partial J(w)}{\partial w}=\frac{1}{2}\frac{\partial}{\partial w}(w^TX^TXw-2w^TX^TY+Y^TY)\ =\frac{1}{2}(2X^TXw-2X^TY+0)$$

$$=X^TXw-X^TY$$

令$\frac{\partial J(w)}{\partial w}=0$,

则有$w=(X^TX)^{-1}X^TY$

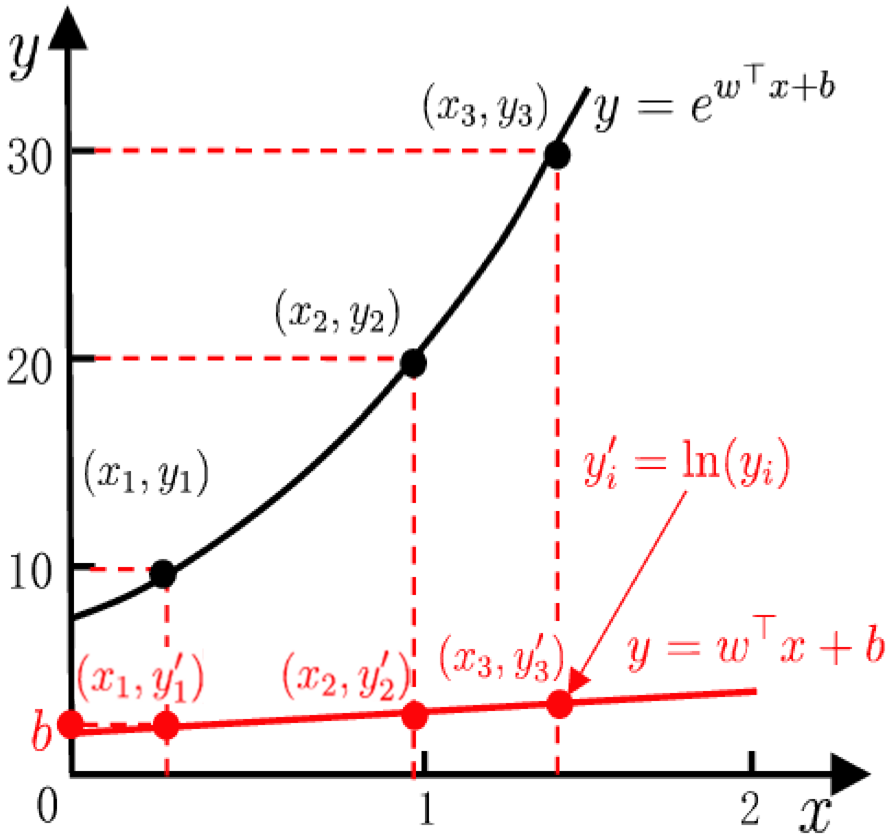
需要用到以下几个矩阵的求导法则:

$$\frac{dX^TX}{dX}=2X \quad \frac{dAX}{dX}=A^T$$

$$\frac{\partial X^TAX}{\partial X}=(A+A^T)X$$,若A为对称阵，$$\frac{\partial X^TAX}{\partial X}=2AX$$

对数线性回归

- 输出标记的对数为线性模型逼近的目标



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

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

线性回归 – 广义线性模型

- 一般形式

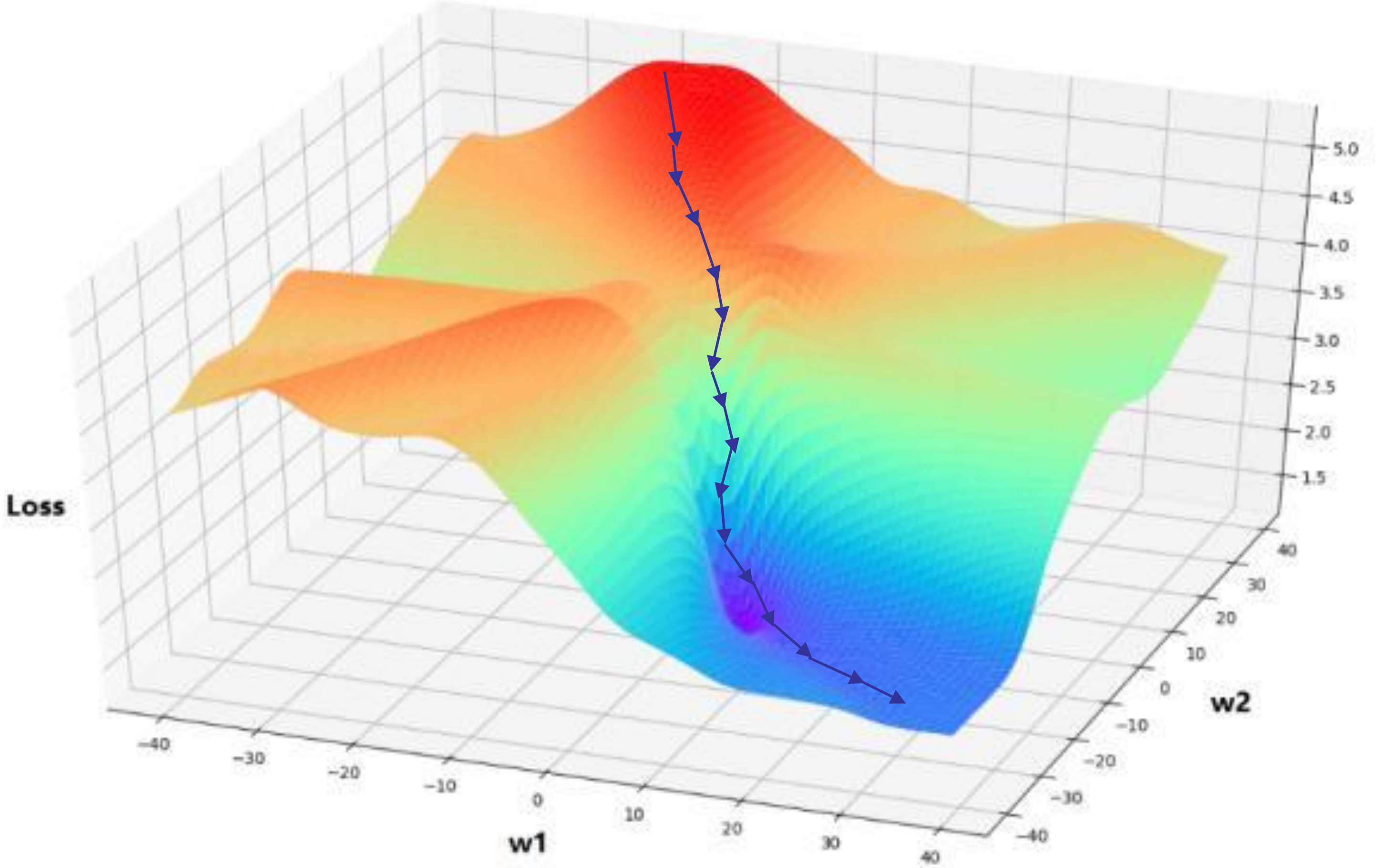
$$y = g^{-1}\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b\right)$$

□ $g(\cdot)$ 称为联系函数（**link function**）

– 单调可微函数

- 对数线性回归是 $g(\cdot) = \ln(\cdot)$时广义线性模型的特例

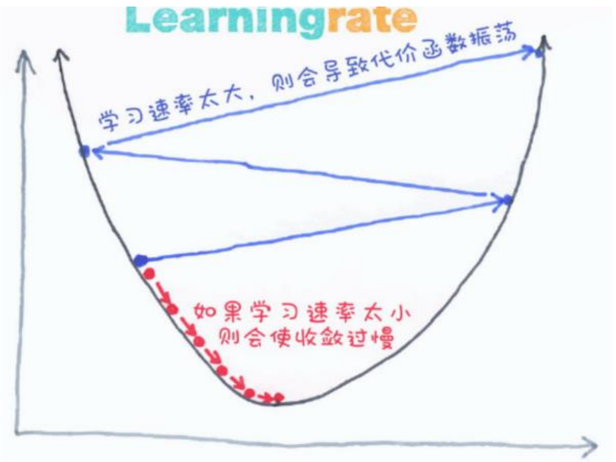
1. 2 梯度下降



学习率

α

步长



1. 2 梯度下降---梯度下降的三种形式

批量梯度下降（Batch Gradient Descent,BGD ）

梯度下降的每一步中，都用到了所有的训练样本

随机梯度下降（Stochastic Gradient Descent,SGD）

梯度下降的每一步中，用到一个样本，在每一次计算之后便更新参数 ，而不需要首先将所有的训练集求和

小批量梯度下降（Mini-Batch Gradient Descent,MBGD）

梯度下降的每一步中，用到了一定批量的训练样本

1. 2梯度下降---批量梯度下降

批量梯度下降（Batch Gradient Descent）

梯度下降的每一步中，都用到了所有的训练样本

参数更新

$$w_j := w_j - \underbrace{\alpha}_{\text{学习率}} \frac{1}{m} \sum_{i=1}^{m} \underbrace{\left(\left(h\left(x^{(i)}\right) - y^{(i)} \right) \cdot x_j^{(i)} \right)}_{\text{梯度}}$$

(同步更新w_j， (j=0,1,...,n))

1. 2 梯度下降---批量梯度下降

批量梯度下降（Batch Gradient Descent）

优点:

由全体训练集确定的方向能够更好的代表样本总体，从而更准确的朝向极值所在的方向，收敛到全局最小值。

缺点:

当样本数m很大时，每次迭代一步都需要对所有样本进行计算，训练过程会很慢。

1. 2 梯度下降---随机梯度下降

随机梯度下降（Stochastic Gradient Descent）

推导 $w = w - \alpha \cdot \frac{\partial J(w)}{\partial w}$ $\quad h(x) = w^T X = w_0 x_0 + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n$

$$J(w) = \frac{1}{2}\left(h\left(x^{(i)}\right) - y^{(i)}\right)^2$$

$$\frac{\partial}{\partial w_j} J(w) = \frac{\partial}{\partial w_j}\frac{1}{2}\left(h\left(x^{(i)}\right) - y^{(i)}\right)^2 = 2 \cdot \frac{1}{2}\left(h\left(x^{(i)}\right) - y^{(i)}\right) \cdot \frac{\partial}{\partial w_j}\left(h\left(x^{(i)}\right) - y^{(i)}\right)$$

$$= \left(h\left(x^{(i)}\right) - y^{(i)}\right) \cdot \frac{\partial}{\partial w_j}\left(\sum_{i=0}^{n} w_i x_i^{(i)} - y^{(i)}\right)$$

$$= \left(h\left(x^{(i)}\right) - y^{(i)}\right) {x_j}^{(i)}$$

1. 2 梯度下降---随机梯度下降

随机梯度下降（Stochastic Gradient Descent）

梯度下降的每一步中，用到一个样本，在每一次计算之后便更新参数，而不需要首先将所有的训练集求和

参数更新

$$w_j := w_j - \alpha\left(h\left(x^{(i)}\right) - y^{(i)}\right)x_j{}^{(i)}$$

(同步更新w_j， (j=0,1,...,n))

1. 2 梯度下降---随机梯度下降

随机梯度下降（Stochastic Gradient Descent）

优点：

即使是大规模数据集，随机梯度下降法也会很快收敛。

缺点：

- 不稳定，因为每一次的方向是不确定的，甚至有可能向反方向前进，准确度下降。
- 可能收敛到局部最优。

1. 2 梯度下降---小批量梯度下降

小批量梯度下降（Mini-Batch Gradient Descent）

梯度下降的每一步中，用到了一定批量的训练样本

每计算常数b次训练实例，便更新一次参数 w

参数更新

$$w_j := w_j - \alpha \frac{1}{b} \sum_{k=i}^{i+b-1} \left(h\left(x^{(k)}\right) - y^{(k)}\right) x_j^{(k)}$$

(同步更新w_j，(j=0,1,...,n))

b=1（随机梯度下降,SGD）
b=m（批量梯度下降,BGD）
b=*batch_size*，通常是2的指数倍，常见有32,64,128等。（小批量梯度下降,MBGD）

优点：小批量梯度下降法使用一部份样本数据参与计算，既降低了计算复杂度，又保证了解的收敛性。

1. 2 梯度下降

	批量梯度下降（BGD）	随机梯度下降（SGD）	小批量梯度下降
优点	非凸函数可保证收敛至全局最优解	计算速度快	计算速度快。收敛稳定
缺点	计算较慢，新样本不能中途加人	计算结果不易收敛，可能会陷入局部最优解	----

1. 2 梯度下降与最小二乘法比较

梯度下降：需要选择学习率α，需要多次迭代，当特征数量n大时也能较好适用，适用于各种类型的模型。

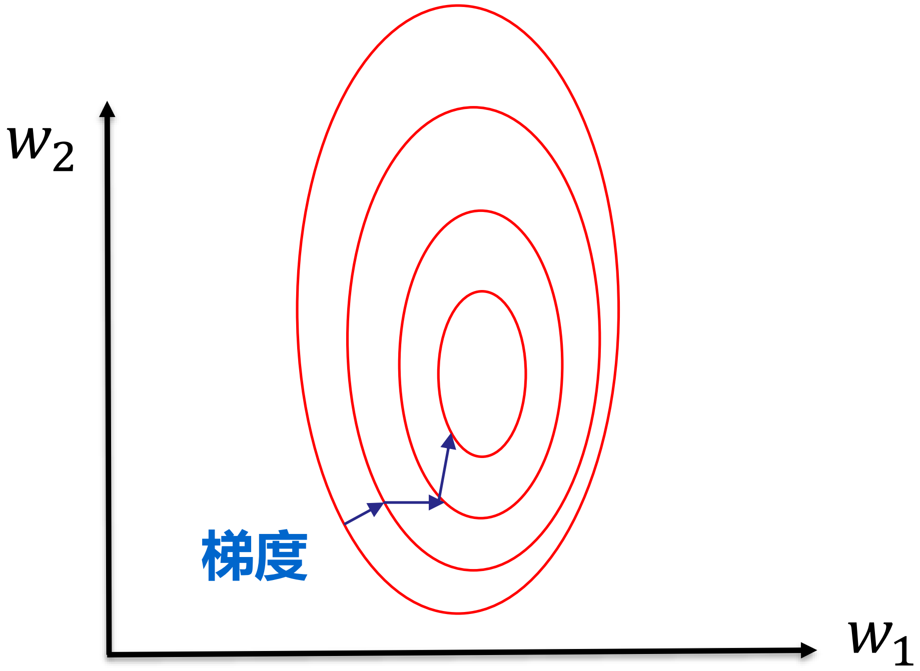
最小二乘法：不需要选择学习率α，一次计算得出，需要计算$(X^T X)^{-1}$，如果特征数量n较大则运算代价大，因为矩阵逆的计算时间复杂度为$O(n^3)$，通常来说当n小于10000 时还是可以接受的，只适用于线性模型，不适合逻辑回归模型等其他模型。

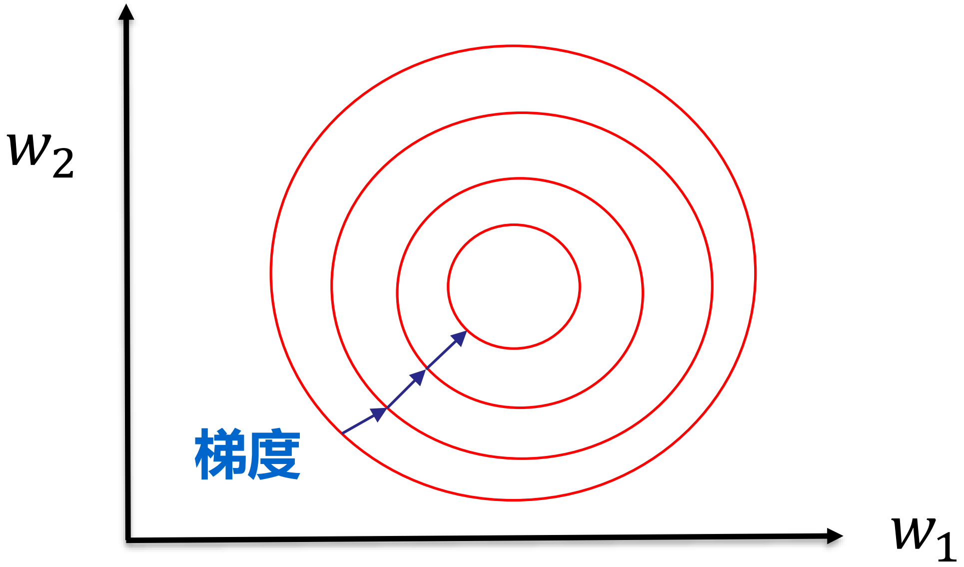
数据归一化/标准化

为什么要标准化/归一化？

提升模型精度：不同维度之间的特征在数值上有一定比较性，可以大大提高分类器的准确性。

加速模型收敛：最优解的寻优过程明显会变得平缓，更容易正确的收敛到最优解。





数据归一化/标准化

归一化（最大 - 最小规范化）

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

将数据映射到[0,1]区间

数据归一化的目的是使得各特征对目标变量的影响一致，会将特征数据进行伸缩变化，所以数据归一化是会改变特征数据分布的。

Z-Score标准化

$$x^* = \frac{x - \mu}{\sigma}$$

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)} - \mu)^2$$

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$$

处理后的数据均值为0，方差为1

数据标准化为了不同特征之间具备可比性，经过标准化变换之后的特征数据分布没有发生改变。

就是当数据特征取值范围或单位差异较大时，最好是做一下标准化处理。

数据归一化/标准化

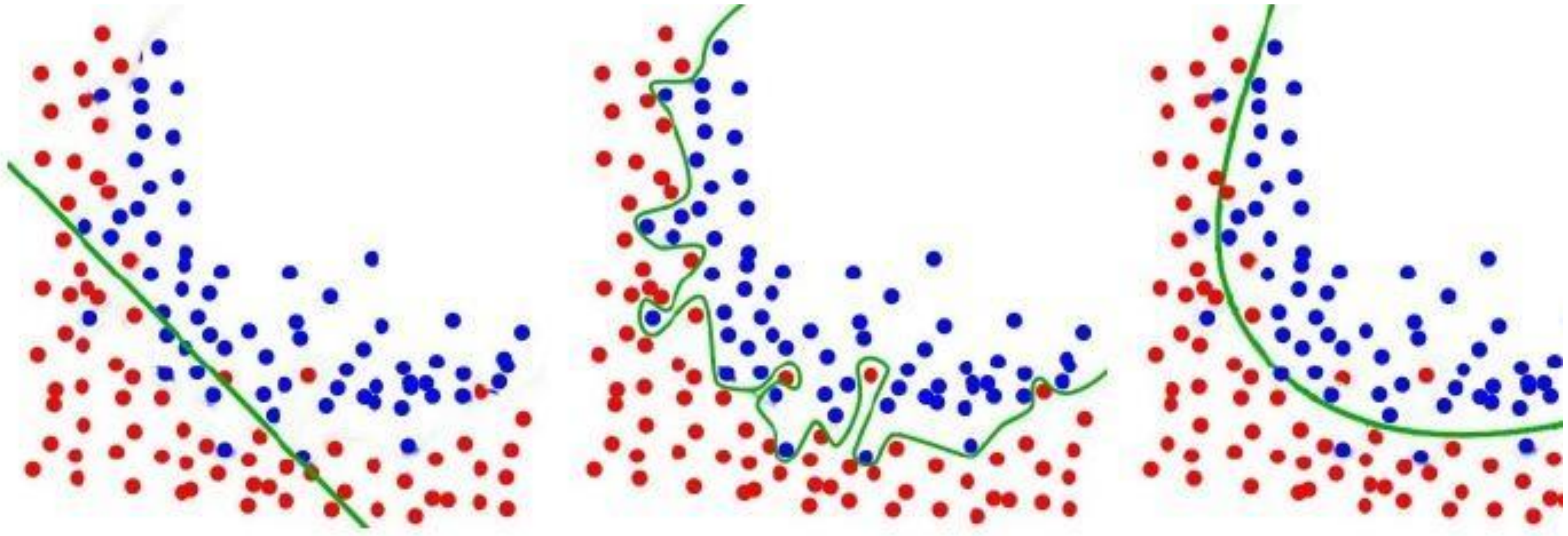
需要做数据归一化/标准化

线性模型，如基于距离度量的模型包括KNN(K近邻)、K-means聚类、感知机和SVM。另外，线性回归类的几个模型一般情况下也是需要做数据归一化/标准化处理的。

不需要做数据归一化/标准化

决策树、基于决策树的Boosting和Bagging等集成学习模型对于特征取值大小并不敏感，如随机森林、XGBoost、LightGBM等树模型，以及朴素贝叶斯，以上这些模型一般不需要做数据归一化/标准化处理。

过拟合和欠拟合



欠拟合　　　　过拟合　　　　正合适

过拟合的处理

1.获得更多的训练数据

使用更多的训练数据是解决过拟合问题最有效的手段，因为更多的样本能够让模型学习到更多更有效的特征，减小噪声的影响。

2.**降维**

即丢弃一些不能帮助我们正确预测的特征。可以是手工选择保留哪些特征，或者使用一些模型选择的算法来帮忙（例如PCA）。

3.**正则化**

正则化(regularization)的技术，保留所有的特征，但是减少参数的大小（magnitude），它可以改善或者减少过拟合问题。

4.**集成学习方法**

集成学习是把多个模型集成在一起，来降低单一模型的过拟合风险。

欠拟合的处理

1.**添加新特征**

当特征不足或者现有特征与样本标签的相关性不强时，模型容易出现欠拟合。通过挖掘组合特征等新的特征，往往能够取得更好的效果。

2.**增加模型复杂度**

简单模型的学习能力较差，通过增加模型的复杂度可以使模型拥有更强的拟合能力。例如，在线性模型中添加高次项，在神经网络模型中增加网络层数或神经元个数等。

3.**减小正则化系数**

正则化是用来防止过拟合的，但当模型出现欠拟合现象时，则需要有针对性地减小正则化系数。

1. 3 正则化

L_1**正则化**：$J(w)=\frac{1}{2}\sum_{i=1}^{m}(h(x^{(i)})-y^{(i)})^2+\lambda\sum_{j=1}^{n}|w_j|$，Lasso Regression（ Lasso回归）

L_2**正则化**：$J(w)=\frac{1}{2}\sum_{i=1}^{m}(h(x^{(i)})-y^{(i)})^2+\lambda\sum_{j=1}^{n}w_j^2$，Ridge Regression （岭回归）

Elastic Net：$J(w)=\frac{1}{2}\sum_{i=1}^{m}(h(x^{(i)})-y^{(i)})^2+\lambda(\rho\cdot\sum_{j=1}^{n}|w_j|+(1-\rho)\cdot\sum_{j=1}^{n}w_j^2)$

（弹性网络）

其中：

- λ为正则化系数，调整正则化项与训练误差的比例，λ>0。
- 1≥ρ≥0为比例系数，调整L_1正则化与L_2正则化的比例。

1. 4 回归的评价指标

均方误差（Mean Square Error,MSE） 指预测值和真实值之差的平方的均值

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - \hat{y}^{(i)})^2$$

MSE越小越好，说明该模型描述实验数据具有更好的精度。

缺点：
MSE里面带着平方，会改变量纲；

其中，$y^{(i)}$ 和$\hat{y}^{(i)}$分别表示第i个样本的真实值和预测值，m 为样本个数。

1. 4 回归的评价指标

均方根误差 (Root Mean Square Error,RMSE)

$$RMSE(y,\hat{y}) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - \hat{y}^{(i)})^2}$$

RMSE越小越好

优点：

RMSE的存在是开完根号之后，误差的结果就和数据是一个单位级别的，可以更好的描述数据。

缺点：

RMSE/MSE对一组测量中对特大/特小误差反映特别敏感，这种局限性常常发生在短时间内变化比较大的数据上，如风电预测，访问量预测等。

1. 4 回归的评价指标

平均绝对误差（Mean Absolute Error,MAE）

$$MAE(y,\hat{y}) = \frac{1}{m}\sum_{i=1}^{n}\left|y^{(i)} - \hat{y}^{(i)}\right|$$

预测值和真实值之差的绝对值求平均。

MAE越小越好，但是不常用，因为它不能求导。

1. 4 回归的评价指标

R方 [$RSquared(r2score)$]

$$R^2(y,\hat{y}) = 1 - \frac{\sum_{i=0}^{m}(y^{(i)} - \hat{y}^{(i)})^2}{\sum_{i=0}^{m}(y^{(i)} - \overline{y})^2} = \frac{\mathrm{SSR}}{\mathrm{SST}} = 1 - \frac{\mathrm{SSE}}{\mathrm{SST}}$$

$$R^2(y,\hat{y}) = 1 - \frac{\sum_{i=0}^{m}(y^{(i)} - \hat{y}^{(i)})^2/m}{\sum_{i=0}^{m}(y^{(i)} - \overline{y})^2/m} = 1 - \frac{\mathrm{MSE}}{\mathrm{Var}}$$

越接近于1,说明模型拟合得越好

$$SSR = \sum_{i=0}^{m}(\hat{y}^{(i)} - \overline{y})^2$$

$$SSE = \sum_{i=0}^{m}(y^{(i)} - \hat{y}^{(i)})^2$$

$$SST = \sum_{i=0}^{m}(y^{(i)} - \overline{y})^2$$

其中，$y^{(i)}$ 和$\hat{y}^{(i)}$分别表示第i个样本的真实值和预测值，m 为样本个数。

2. 逻辑回归

2.1 分类问题

2.2 逻辑回归概述

2.3 逻辑回归求解

2. 1 分类问题

监督学习的最主要类型

标签离散

- ✓ 分类（Classification）
  - ✓ 身高1.85m，体重100kg的男人穿什么尺码的T恤？
  - ✓ 根据肿瘤的体积、患者的年龄来判断良性或恶性？
  - ✓ 根据用户的年龄、职业、存款数量来判断信用卡是否会违约？

输入变量可以是离散的，也可以是连续的。

2. 1 分类问题---二分类

二分类:

表示分类任务中有两个类别，比如我们想识别一幅图片是不是狗。即训练一个分类器，特征向量x表示输入的图片，输出为y，若y=0表示输出不是狗，y=1表示输出是狗。

二分类是假设每个样本都被设置了一个且仅有一个标签0或者1。

2. 1 分类问题---多分类

多类分类：

表示分类任务中有多个类别，比如对一堆水果图片分类，它们可能是橘子、苹果、梨等。多类分类是假设每个样本都被设置了一个且仅有一个标签：一个水果可以是苹果或者梨，但是同时不可能是两者。

2. 2 逻辑回归概述

逻辑回归的应用场景：

- 广告点击率
- 是否为垃圾邮件
- 是否患病
- 金融诈骗
- 虚假账号

逻辑回归一般用于**解决分类（二分类）问题**

2. 2 逻辑回归概述

- 预测值与输出标记

$$z = w^T x + b \qquad y \in \{0,1\}$$

- 寻找函数将分类标记与线性回归模型输出联系起来

- 最理想的函数：单位阶跃函数$y = \begin{cases} 0, z < 0 \\ 0.5, z = 0 \\ 1, z > 0 \end{cases}$

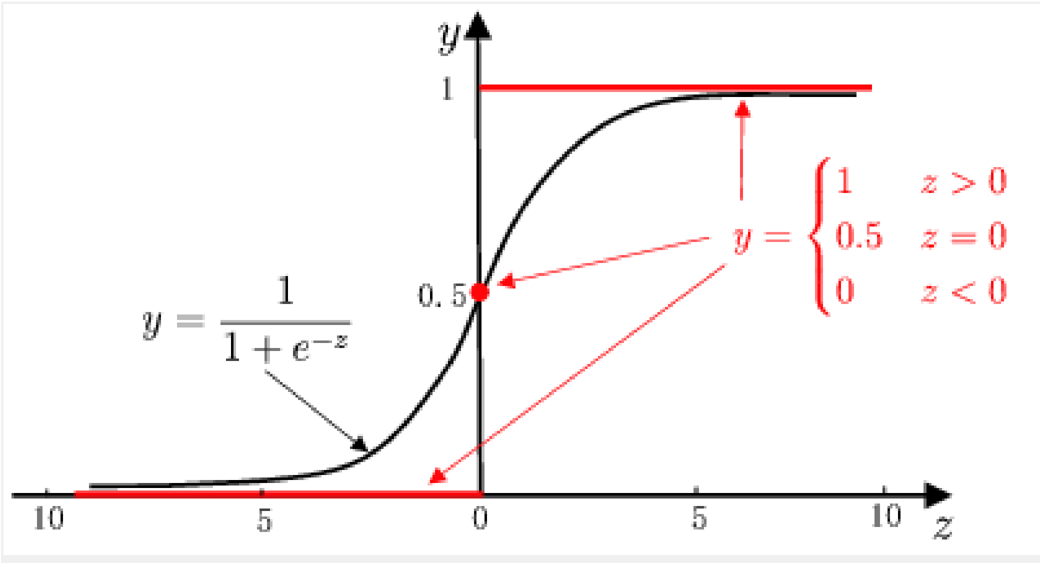
  - **预测值大于零就判为正例，小于零则判为反例，预测值为临界值零则可任意判别**

2. 2 逻辑回归概述

- 单位阶跃函数缺点
  - **不连续**

- 替代函数：对数几率函数（是一种Sigmoid函数）
  - **单调可微、任意阶可导**

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



单位阶跃函数与对数几率函数的比较

2. 2 逻辑回归概述

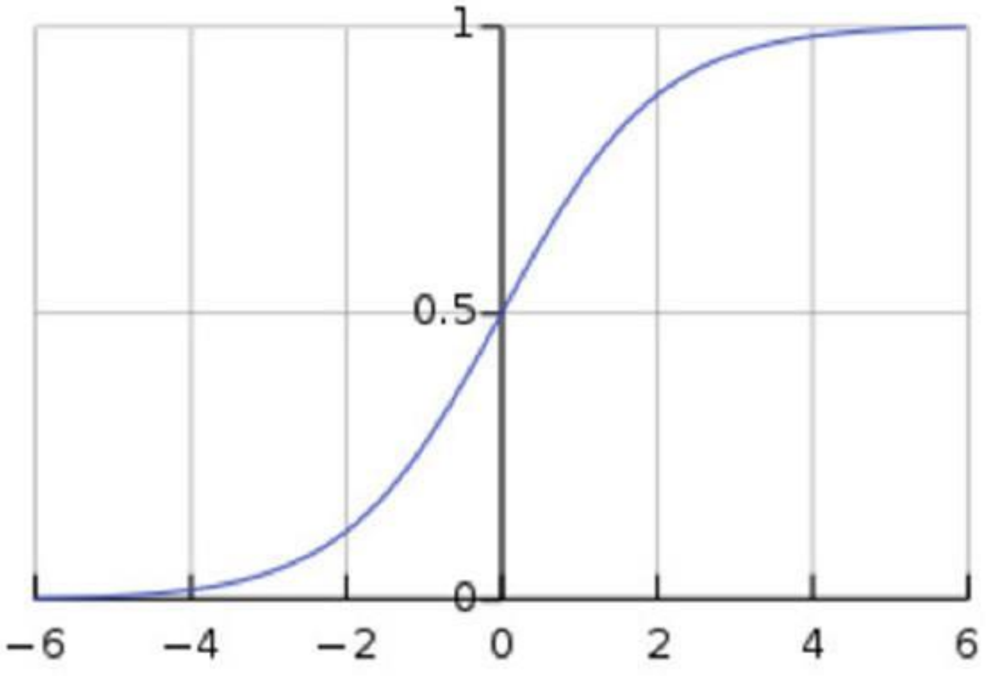
Sigmoid 函数

$\sigma(z)$代表一个常用的逻辑函数（logistic function）为S形函数（Sigmoid function）

则：$\sigma(z) = g(z) = \frac{1}{1+e^{-z}}$ $z{=}w^T x + b$

合起来，我们得到逻辑回归模型的假设函数：

$$L(\hat{y}, y) = -y\log(\hat{y}) - (1 - y)\log(1 - \hat{y})$$



当$\sigma(z)$大于等于0.5时，预测 y=1

当$\sigma(z)$小于0.5时，预测 y=0

注意：若表达式 $h(x) = z = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + b = w^T x + b$，则$b$可以融入到$w_0$，即：$z{=}w^T x$

2. 2 逻辑回归概述

线性回归的函数 $h(x) = z = w^T x$，范围是$(-\infty, +\infty)$。

而分类预测结果需要得到$[0,1]$的概率值。

在二分类模型中，事件的几率odds：事件发生与事件不发生的概率之比为$\frac{p}{1-p}$，

称为事件的发生比（the odds of experiencing an event）

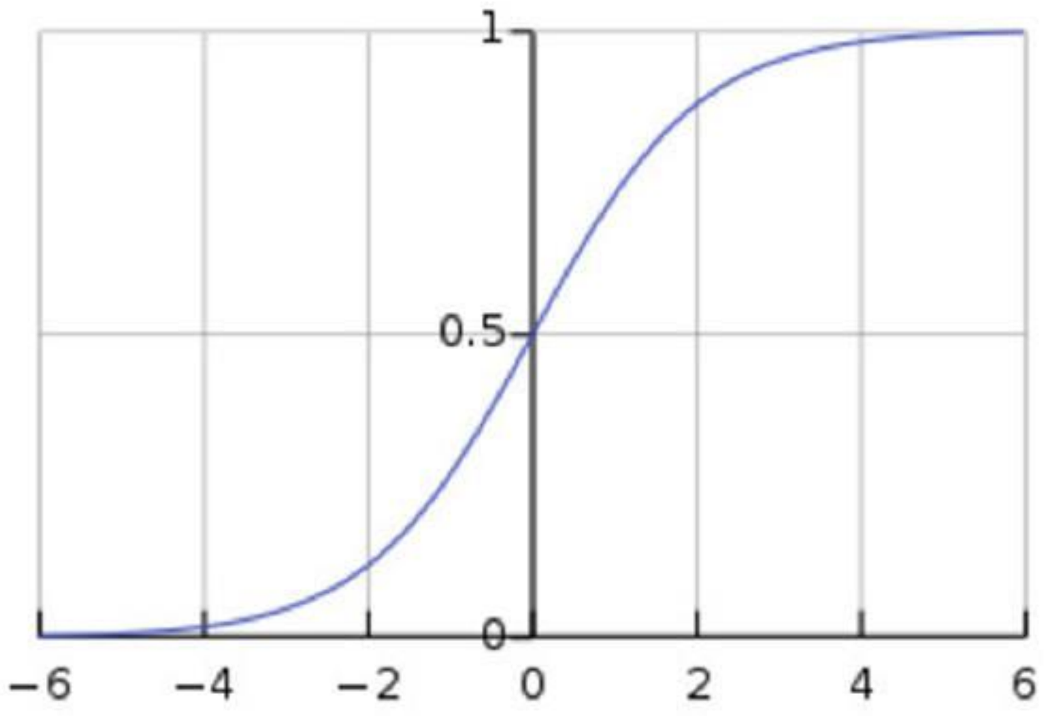
其中p为随机事件发生的概率，p的范围为$[0,1]$。

取对数得到：$\log\frac{p}{1-p}$，而$\log\frac{p}{1-p} = w^T x = z$

求解得到：$p = \frac{1}{1+e^{-w^T x}} = \frac{1}{1+e^{-z}}$

2.2 逻辑回归概述

将z进行逻辑变换：$g(z)=\frac{1}{1+e^{-z}}$



$g(z)$

$$
\begin{aligned}
g'(z) &= \left(\frac{1}{1+e^{-z}}\right)' \\
&= \frac{e^{-z}}{(1+e^{-z})^2} \\
&= \frac{1+e^{-z}-1}{(1+e^{-z})^2} \\
&= \frac{1}{(1+e^{-z})}\left(1-\frac{1}{(1+e^{-z})}\right) \\
&= g(z)(1-g(z))
\end{aligned}
$$

2. 3 逻辑回归求解

假设一个二分类模型：

$$p(y=1|x;w)=h(x)$$

$$p(y=0|x;w)=1-h(x)$$

则：

$$p(y|x;w)=(h(x))^{y}(1-h(x))^{1-y}$$

逻辑回归模型的假设是： $h(x)=g\left(w^{T}x\right)=g(z)$

其中$z=w^{T}x$ ，逻辑函数（logistic function)公式为：

$$g(z)=\frac{1}{1+e^{-z}}\text{，}g'(z)=g(z)(1-g(z))$$

2. 3 逻辑回归求解

损失函数

$$L(\hat{y}, y) = -y\log(\hat{y}) - (1-y)\log(1-\hat{y})$$

$\hat{y}$　表示预测值$h(x)$

y　表示真实值

为了衡量算法在全部训练样本上的表现如何，我们需要定义一个算法的代价函数，算法的代价函数是对m个样本的损失函数求和然后除以m:

代价函数

$$J(w) = \frac{1}{m}\sum_{i=1}^{m} L\left(\hat{y}^{(i)}, y^{(i)}\right) = \frac{1}{m}\sum_{i=1}^{m}\left(-y^{(i)}\log\hat{y}^{(i)} - (1-y^{(i)})\log(1-\hat{y}^{(i)})\right)$$

2. 3 逻辑回归求解

求解过程：

似然函数为： $L(w)=\prod_{i=1}^{m}P\left(y^{(i)}|x^{(i)};w\right)=\prod_{i=1}^{m}\left(h(x^{(i)})\right)^{y^{(i)}}\left(1-h(x^{(i)})\right)^{1-y^{(i)}}$

似然函数两边取对数，则连乘号变成了连加号：

$$l(w)=\log L(w)=\sum_{i=1}^{m}\left(y^{(i)}\log(h(x^{(i)}))+(1-y^{(i)})\log(1-h(x^{(i)}))\right)$$

代价函数为：

$$J(w)=-\frac{1}{m}l(w)=-\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\log(h(x^{(i)}))+(1-y^{(i)})\log(1-h(x^{(i)}))\right)$$

2. 3 逻辑回归求解

梯度下降求解过程：

$$w_j := w_j - \alpha \frac{\partial J(w)}{\partial w}$$

$$J(w) = -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\log(h(x^{(i)})) + (1 - y^{(i)})\log(1 - h(x^{(i)}))\right)$$

$$\frac{\partial}{\partial w_j}J(w) = \frac{1}{m}\sum_{i=1}^{m}\left(h(x^{(i)}) - y^{(i)}\right)x_j^{(i)}$$

则：$w_j := w_j - \alpha \frac{1}{m}\sum_{i=1}^{m}\left(h(x^{(i)}) - y^{(i)}\right)x_j^{(i)}$

2. 3 逻辑回归求解

求解过程： $\frac{\partial}{\partial w_j}J(w)=\frac{1}{m}\sum_{i=1}^{m}(h(x^{(i)})-y^{(i)})x_j^{(i)}$的推导过程：

$$J(w)=-\frac{1}{m}\sum_{i=1}^{m}(\underline{y^{(i)}\log(h(x^{(i)}))+(1-y^{(i)})\log(1-h(x^{(i)}))})$$

$$\Downarrow$$

$$\begin{aligned}
&y^{(i)}\log(h(x^{(i)}))+(1-y^{(i)})\log(1-h(x^{(i)}))\\
&=y^{(i)}\log(\frac{1}{1+e^{-w^Tx^{(i)}}})+(1-y^{(i)})\log(1-\frac{1}{1+e^{-w^Tx^{(i)}}})\\
&=-y^{(i)}\log(1+e^{-w^Tx^{(i)}})-(1-y^{(i)})\log(1+e^{w^Tx^{(i)}})
\end{aligned}$$

2. 3 逻辑回归求解

求解过程： $\frac{\partial}{\partial w_j}J(w)=\frac{1}{m}\sum_{i=1}^{m}(h(x^{(i)})-y^{(i)})x_j^{(i)}$的推导过程：

$$\frac{\partial}{\partial w_j}J(w)=\frac{\partial}{\partial w_j}(-\frac{1}{m}\sum_{i=1}^{m}(-y^{(i)}\log\left(1+e^{-w^Tx^{(i)}}\right)-(1-y^{(i)})\log\left(1+e^{w^Tx^{(i)}}\right)))$$

$$=-\frac{1}{m}\sum_{i=1}^{m}(-y^{(i)}\frac{-x_j^{(i)}e^{-w^Tx^{(i)}}}{1+e^{-w^Tx^{(i)}}}-(1-y^{(i)})\frac{x_j^{(i)}e^{w^Tx^{(i)}}}{1+e^{w^Tx^{(i)}}})$$

$$=-\frac{1}{m}\sum_{i=1}^{m}(y^{(i)}-h(x^{(i)}))x_j^{(i)}$$

$$=\frac{1}{m}\sum_{i=1}^{m}(h(x^{(i)})-y^{(i)})x_j^{(i)}$$

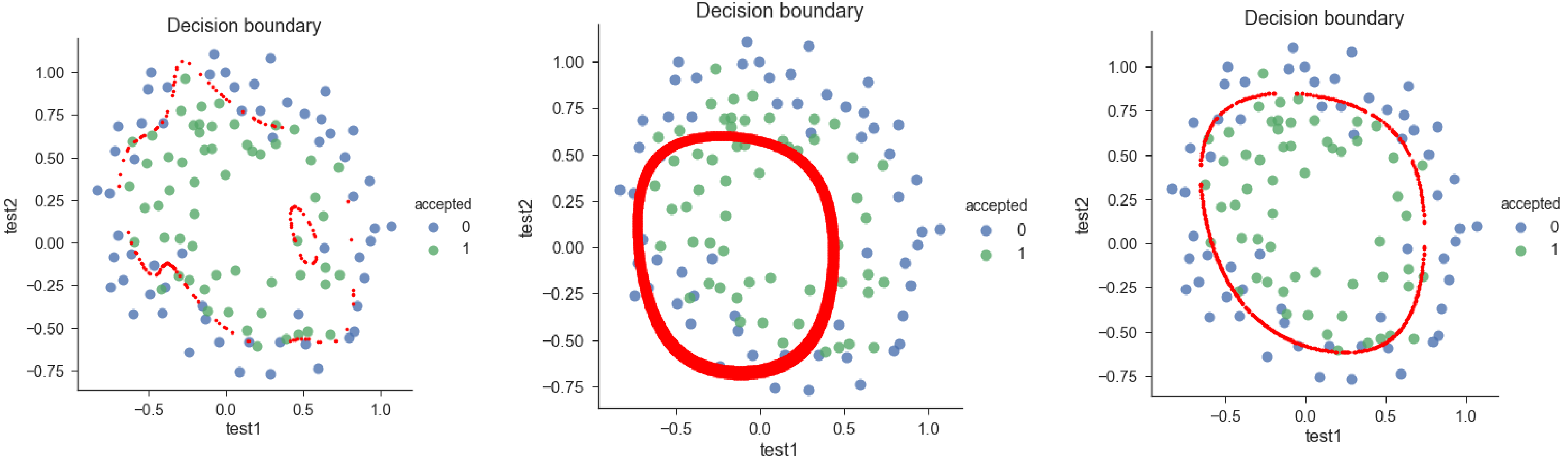
2. 3 逻辑回归求解

正则化：目的是**为了防止过拟合**

正则化项

$$J(w)=\frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log\left(h\left(x^{(i)}\right)\right)-\left(1-y^{(i)}\right)\log\left(1-h\left(x^{(i)}\right)\right)\right]+\boxed{\frac{\lambda}{2m}\sum_{j=1}^{n}w_j^2}$$

当 λ 的值开始上升时，降低了方差。



没有正则化，过拟合　　正则化过度，欠拟合　　适当的正则化

线性回归与逻辑回归的区别与联系

- 线性回归用来预测，逻辑回归用来分类。
  - 在线性回归模型中，输出一般是连续的，对于每一个输入的x，都有一个对应的输出y。因此模型的定义域和值域都可以是无穷。
  - 逻辑回归，输入可以是连续的[-∞,+ ∞]，但输出一般是离散的，通常只有两个值{0,1}。
- 线性回归是拟合函数，逻辑回归是预测函数。
- 线性回归和逻辑回归的损失函数:
  - 线性回归中使用的是最小化平方误差损失函数。
  - 逻辑回归使用对数似然函数进行参数估计，使用交叉熵作为损失函数，对预测错误的惩罚是随着输出的增大，逐渐逼近一个常数。

3. 1 线性判别分析概述

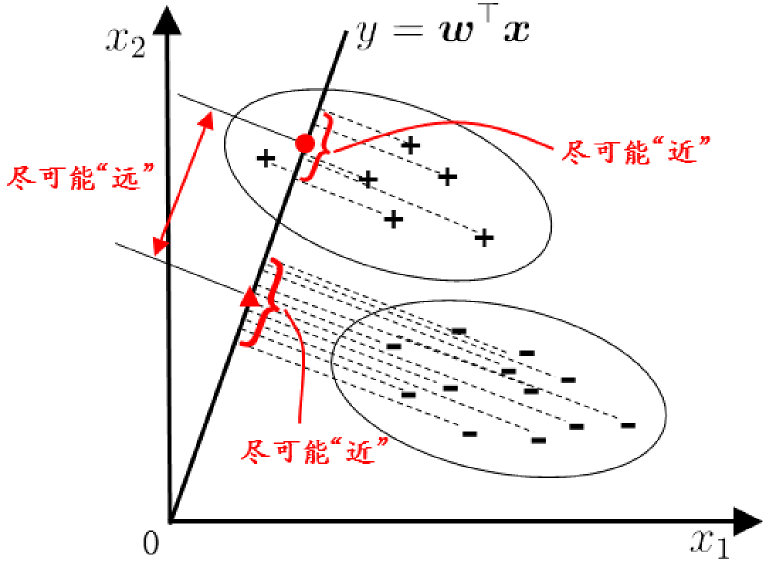
- 线性判别分析（Linear Discriminant Analysis，简称为LDA）也叫Fisher线性判别分析，是特征抽取中最为经典和广泛使用的方法之一。LDA是由R.AFisher于1936年提出来的方法,主要是用来解决生物问题的分类问题。它是在1996年由Belhumeur引入模式识别和人工智能领域的.

3. 1 线性判别分析---二分类任务

■LDA的核心思想

●欲使同类样例的投影点尽可能接近，可以让同类样例投影点的协方差尽可能小

●欲使异类样例的投影点尽可能远离，可以让不同类中心之间的距离尽可能大



LDA也可被视为一种监督降维技术

3. 1 线性判别分析---二分类任务

■ 一些变量

- 第i类示例的集合X_I
- 第i类示例的均值向量μ_i
- 第i类示例的协方差矩阵 $\sum_{x \in X_i}(x-\mu_i)(x-\mu_i)^T$
- 两类样本的中心在直线上的投影：$w^T\mu_0$和$w^T\mu_1$
- 两类样本投影点的协方差：$w^T\sum_0 w$ 和$w^T\sum_1 w$

3. 1 线性判别分析——二分类任务

- 最大化目标

$$J=\frac{|w^T\mu_0-w^T\mu_1|_2^2}{w^T\sum_0 w+w^T\sum_1 w}$$
$$=\frac{w^T(\mu_0-\mu_1)(\mu_0-\mu_1)^T w}{w^T(\sum_0+\sum_1)w}$$

- 类内散度矩阵

$$S_w=\left(\sum\nolimits_0+\sum\nolimits_1\right)$$
$$=\sum_{x\in X_0}(x-\mu_0)(x-\mu_0)^T+\sum_{x\in X_1}(x-\mu_1)(x-\mu_1)^T$$

- 类间散度矩阵

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

3. 1 线性判别分析---二分类任务

- 广义瑞利商（generalized Rayleigh quotient）

$$J=\frac{w^T S_b w}{w^T S_w w}$$

- 令$w^T S_w \mathrm{w}=1$，最大化广义瑞利商等价形式为

$$\min -w^T S_b \mathrm{w}$$
$$s.t. w^T S_w \mathrm{w}=1$$

- 运用拉格朗日乘子法

$$S_b w = \lambda S_w w$$

3. 1 线性判别分析---二分类任务

$S_b w = \lambda S_w w$

- 同向向量

$$S_b w = \lambda S_w(\mu_0 - \mu_1)$$

同向向量

- 结果

$$w = S_w^{-1}(\mu_0 - \mu_1)$$

- 求解
  - **奇异值分解**

$$S_w = U \sum V^T$$

- LDA的贝叶斯决策论解释
  - 两类数据同先验、满足高斯分布且协方差相等时，LDA达到最优分类

LDA推广– 多分类任务

- 全局散度矩阵 $S_t = S_b + S_w$

$$= \sum_{x \in X_0} (x_i - \mu)(x_i - \mu)^T$$

- 类内散度矩阵 $S_w = \sum_{i=1}^{N} S_{w_i}$

  其中

$$S_{w_i} = \sum_{x \in X_i} (x - \mu_1)(x - \mu_1)^T$$

- 求解得

$$S_{\mathrm{b}} = S_t - S_w = \sum_{x \in X_0}^{N} m_i (u_i - \mu)(u_i - \mu)^T$$

LDA推广– 多分类任务

- 类内散度矩阵 $S_w = \sum_{i=1}^{N} S_{w_i}$

  其中 $\sum_{x \in X_i} (x - \mu_1)(x - \mu_1)^T$

- 类间散度矩阵 $S_b = \sum_{x \in X_0}^{N} m_i (u_i - \mu)(u_i - \mu)^T$

- p个投影方向 $W = [w_1, w_2 \ldots w_p]$ $\sum_{j \in p} w_j^T \mathbf{S}_w w_j = tr(\boldsymbol{W}^T \mathbf{S}_w \boldsymbol{W})$

  – 最小化类内投影协方差之和 $\sum_{j \in p} w_j^T S_w w_j = tr(W^T S_w W)$

  – 最大化类间投影距离之和 $\sum_{j \in p} w_j^T S_b w_j = tr(W^T S_b W)$

LDA推广– 多分类任务

- 优化目标

$$\max_{\mathrm{W}} \frac{tr(W^T S_b W)}{tr(W^T S_w W)}$$

其中$W \in \mathbb{R}^{d\times(N-1)}$

$$S_b W = \lambda S_w W$$

W的闭式解则是 $S_w^{-1}S_\mathrm{b}$的前d' $(d' \leq N-1)$个最大非零广义特征值所对应的特征向量组成的矩阵

- 多分类LDA将样本投影到d'维空间，d'通常远小于数据原有的属性数d，因此LDA也被视为一种监督降维技术

3.1 线性判别分析----算法流程

输入：数据集D $\longrightarrow \{(x_1, y_1) \ldots .(x_m, y_m)\}$，任意样本$x_i$为n维向量，$y_1 \longrightarrow \{C_1, C_2, \ldots .\}$，共k个类别。现在要将其降维到d维;
输出：降维后的数据集D′ 。

（1）计算类内散度矩阵S_B；

(2) 计算类间散度矩阵S_w；

（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；

（5）对样本集中的每一个样本特征x_1，转化为新的样本$y_i = W^T x_i$,；

（6）输出样本集。

3. 1 线性判别分析---优缺点

LDA算法既可以用来降维，也可以用来分类，但是目前来说，主要还是用于降维。在我们进行图像识别相关的数据分析时，LDA是一个有力的工具。

LDA算法的主要优点:

- 在降维过程中可以使用类别的先验知识经验。
- LDA在样本分类信息依赖均值而不是方差的时候，比PCA之类的算法较优

3. 1 线性判别分析---优缺点

LDA算法的主要缺点:

- LDA不适合对非高斯分布样本进行降维。
- LDA降维最多降到类别数k-1的维数，如果我们降维的维度大于k-1，则不能使用LDA。当然目前有一些LDA的进化版算法可以绕过这个问题。
- LDA在样本分类信息依赖方差而不是均值的时候，降维效果不好。
- LDA可能过度拟合数据。

4. 多分类学习

4.1 一对一

4.2 一对其余

4.3 多对多

4. 多分类学习

在逻辑回归中提及，逻辑回归只能解决二分类问题

多分类任务怎么解决呢？

4. 多分类学习

- 多分类学习方法
  - **二分类学习方法推广到多类**
  - **利用二分类学习器解决多分类问题（常用）**
    - 对问题进行拆分，为拆出的每个二分类任务训练一个分类器
    - 对于每个分类器的预测结果进行集成以获得最终的多分类结果

- 拆分策略
  - 一对一（One vs. One, OvO）
  - 一对其余（One vs. Rest, OvR）
  - 多对多（Many vs. Many, MvM）

4. 1 一对一

- **拆分阶段**
  - **N个类别两两配对**
    - N(N-1)/2 个二类任务
  - **各个二类任务学习分类器**
    - N(N-1)/2 个二类分类器

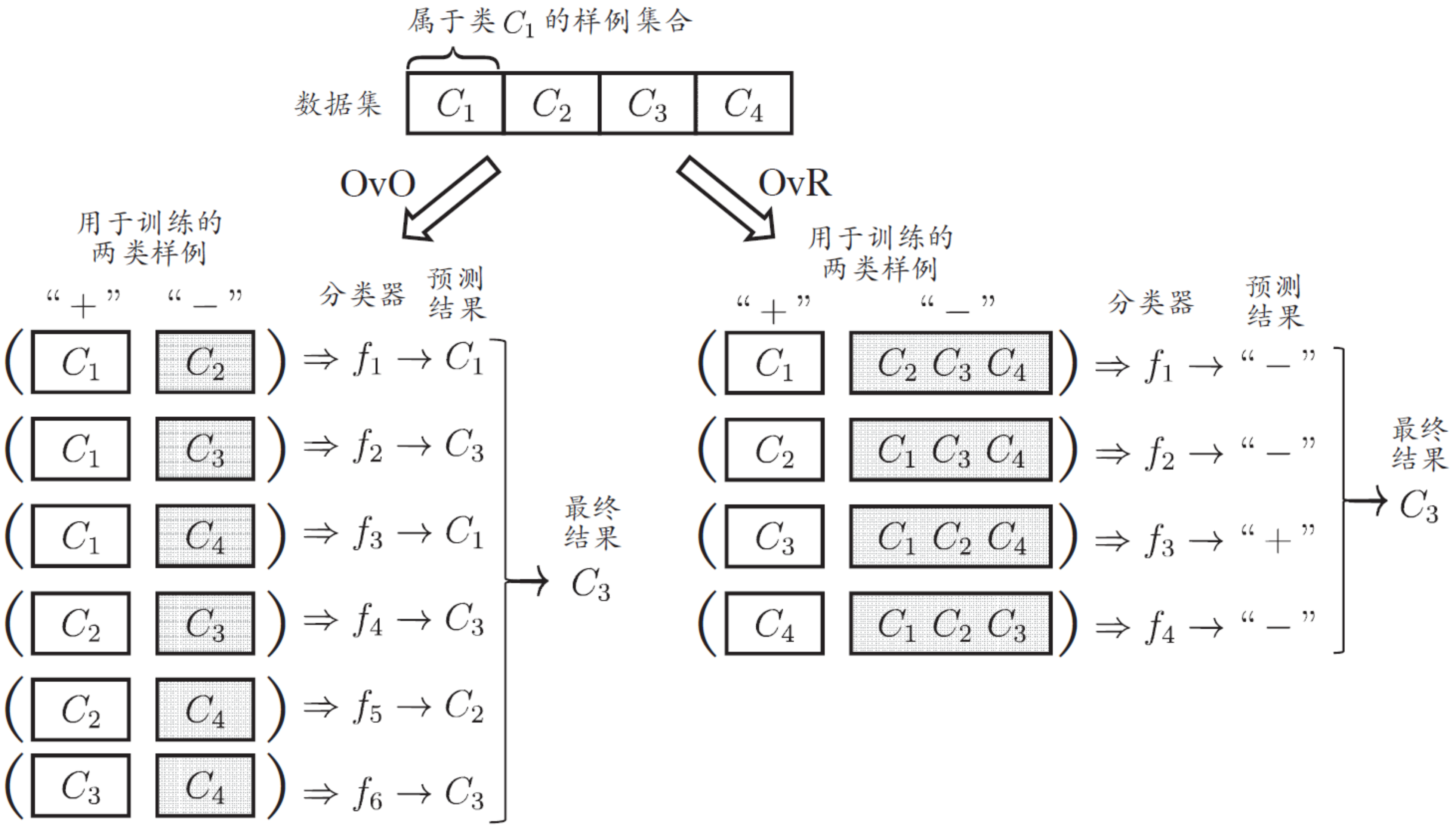
- **测试阶段**
  - **新样本提交给所有分类器预测**
    - N(N-1)/2 个分类结果
  - **投票产生最终分类结果**
    - 被预测最多的类别为最终类别

4. 2 一对其余

- **任务拆分**
  - **某一类作为正例，其他反例**
    - N 个二类任务
  - **各个二类任务学习分类器**
    - N 个二类分类器

- **测试阶段**
  - **新样本提交给所有分类器预测**
    - N 个分类结果
  - **比较各分类器预测置信度**
    - 置信度最大的类别作为最终类别

两种策略比较


属于类 C1 的样例集合
数据集
C1 C2 C3 C4
OvO
OvR
用于训练的两类样例
"+" "−"
分类器
预测结果
f1 → C1
f2 → C3
f3 → C1
f4 → C3
f5 → C2
f6 → C3
最终结果
C3
f1 → "−"
f2 → "−"
f3 → "+"
f4 → "−"
最终结果
C3

两种策略比较

一对一

- 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

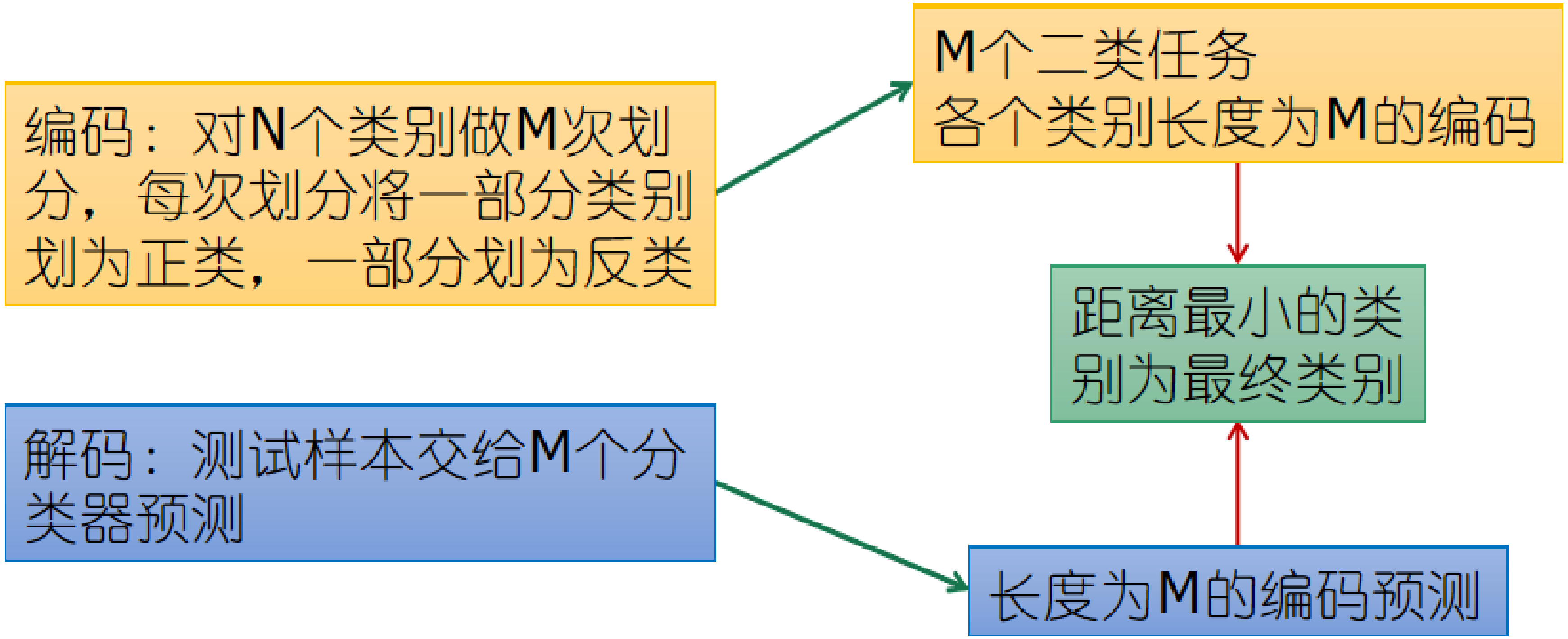
一对其余

- 训练N个分类器，存储开销和测试时间小
- 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

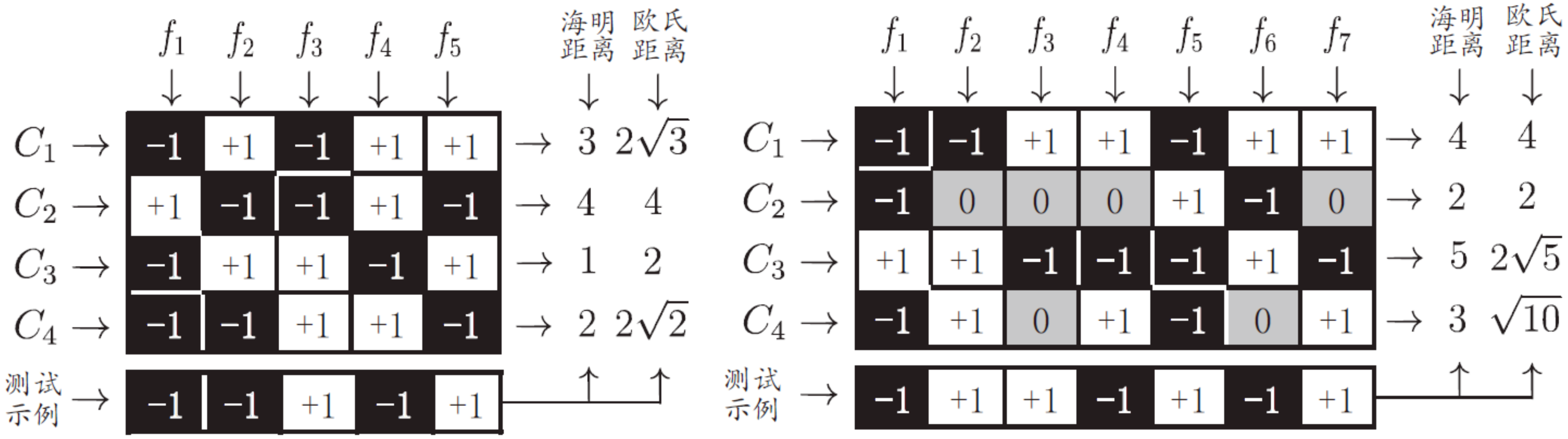
4. 3 多对多

- 多对多（Many vs Many, MvM）
  - **若干类作为正类，若干类作为反类**
- 纠错输出码（Error Correcting Output Codes, ECOC）



4. 3 多对多

- 纠错输出码(Error Correcting Output Codes, ECOC)



(a) 二元 ECOC 码　　(b) 三元 ECOC 码

- 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

5. 类别不平衡

5.1 类别不平衡概述

5.2 类别不平衡导致分类困难的原因

5.3 类别不平衡的解决办法

5. 1 类别不平衡问题概述

- 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。
- 在现实的分类学习任务中，我们经常会遇到类别不平衡，例如在通过拆分法解决多分类问题时，即使原始问题中不同类别的训练样例数目相当，在使用OvR、MvM策略后产生的二分类任务仍可能出现类别不平衡现象，因此有必要了解类别不平衡性处理的基本方法。

5. 2 类别不平衡导致分类困难的原因

- 正负样本特征区别较大，边界较宽;
- 少数类分布的稀疏性（sparsity)以及稀疏性导致的拆分多个子概念(sub-concepts，可理解为子clusters)并且每个子概念仅含有较少的样本数量；
- 离群点过多(即过多的少数类样本出现在多数类样本密集的区域);
- 类别之间的分布严重重叠（即不同类别的样本相对密集地出现在特征空间的同一区域);
- 数据中本身存在的噪声，尤其是少数类的噪声。

5. 3 类别不平衡的解决方法

用$y = w^T x + b$对新样本a进行分类时，事实上是在用预测出的y值与一个阈值进行比较。

例如：

- 通常在g >0.5时判别为正例，否则为反例
- 阈值设置为 0.5 则表示分类器认为真实正、反例可能性相同。
- y实际上表达了正例的可能性
- 几率$\frac{y}{1-y}$则反映了正例可能性与反例可能性之比值

5. 3 类别不平衡的解决方法

- 当正反例个数相同时：

$$若\frac{y}{1-y} > 1 则 预测为正例$$

- 正、反例的个数不同时：

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

其中，m^+表示正例数目, m^-表示反例数目, $\frac{m^+}{m^-}$表示观测几率，由于我们通常假设训练集是真实样本总体的无偏采样,因此观测几率就代表了真实几率.于是,只要分类器的预测几率高于观测几率就应判定为正例。

对预测值进行调整：

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

这就是类别不平衡学习的一个基本策略——“再缩放”(rescaling).

5. 3 类别不平衡的解决方法

- **再缩放**
  - ➢ **欠采样（undersampling）**
    - 去除一些反例使正反例数目接近（EasyEnsemble [Liu et al.,2009]）
  - ➢ **过采样（oversampling）**
    - 增加一些正例使正反例数目接近（SMOTE [Chawla et al.2002]）
  - ➢ **阈值移动（threshold-moving）**

优化提要

- 各任务下（回归、分类）各个模型优化的目标
  - 最小二乘法：最小化均方误差
  - 对数几率回归：最大化样本分布似然
  - 线性判别分析：投影空间内最小（大）化类内（间）散度

- 参数的优化方法
  - 最小二乘法：线性代数
  - 对数几率回归：凸优化梯度下降、牛顿法
  - 线性判别分析：矩阵论、广义瑞利商

参考文献

[1] Andrew Ng. Machine Learning[EB/OL]. Stanford University,2014. https://www.coursera.org/course/ml

[2] 周志华. 机器学习[M]. 清华大学出版社,2016.

[3] 李航. 统计学习方法[M]. 清华大学出版社,2019.

[4] WEINBERGER K. Distance metric learning for large margin nearest neighbor classification[J]. Advances in Neural Information Processing Systems, 2006, 18.

[5] HOERL A E, KENNARD R W. Ridge regression: applications to nonorthogonal problems[J]. Technometrics, 1970, 12(1): 69–82.

[6] TIBSHIRANI R. Regression selection and shrinkage via the lasso[J]. Journal of the Royal Statistical Society Series B, 1996, 58(1): 267–288.

[7] TIBSHIRANI R, BICKEL P, RITOV Y, et al. Least absolute shrinkage and selection operator[J]. Software: http://www.stat.stanford.edu/ tibs/lasso.html, 1996.

谢 谢！