1

机器学习-逻辑回归

目录

1.分类问题

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。

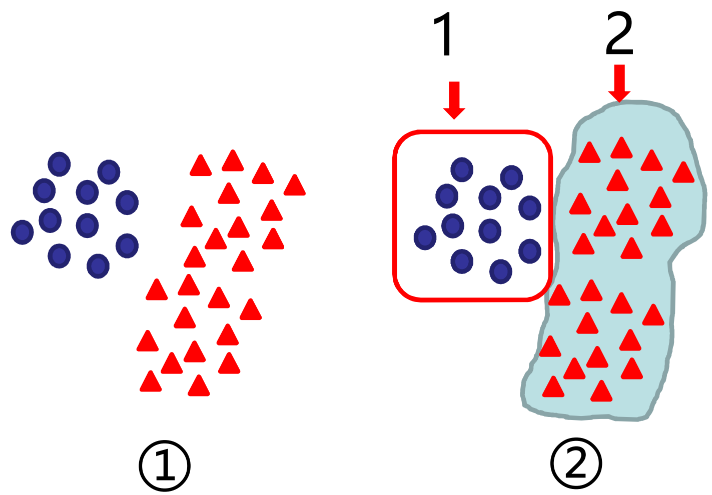
分类问题

二分类

我们先从用蓝色圆形数据定义为类型1，其余数据为类型2；

只需要分类1次

步骤：①->②



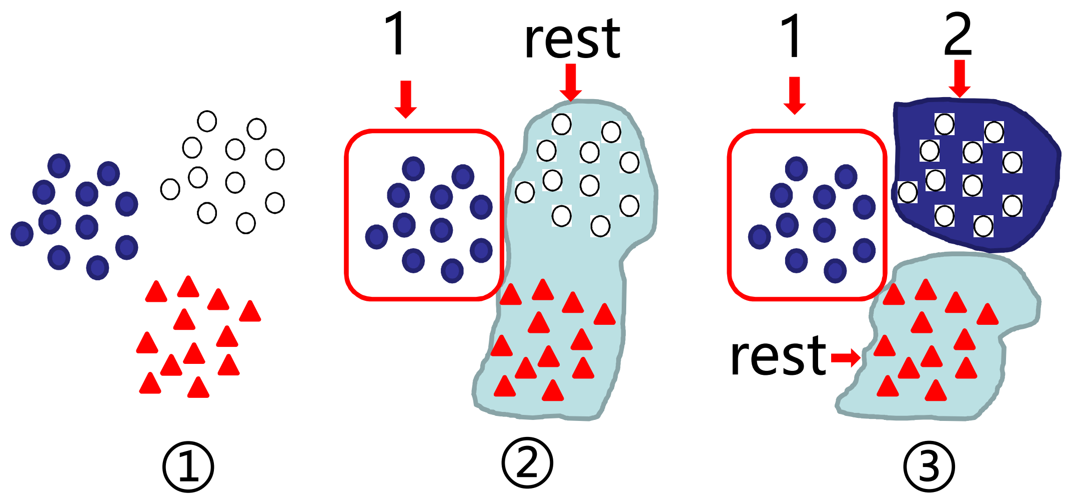
二分类

分类问题

多分类

我们先定义其中一类为类型1（正类），其余数据为负类（rest）；接下来去掉类型1数据，剩余部分再次进行二分类，分成类型2和负类；如果有n类，那就需要分类n-1次

步骤：①->②->③->……



One-vs-All (One-vs-Rest)

一对多 (一对余)

2.Sigmoid函数

问题

- 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。

- 分类预测值与输出标记

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

- 寻找函数将分类标记与线性回归模型输出联系起来

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

- 最理想的函数——单位阶跃函数

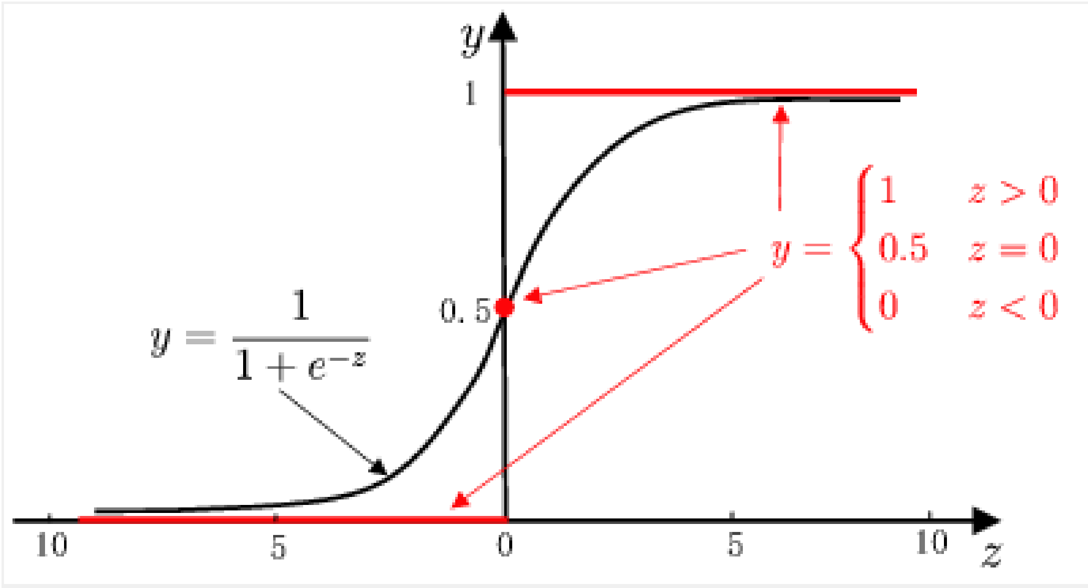
  – 预测值大于零就判为正例，小于零就判为反例，预测值为临界值零则可任意判别

问题

- 单位阶跃函数缺点
  - 不连续
- 替代函数——对数几率函数（logistic function）
  - 单调可微、任意阶可导

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

单位阶跃函数与对数几率函数的比较



Sigmoid函数

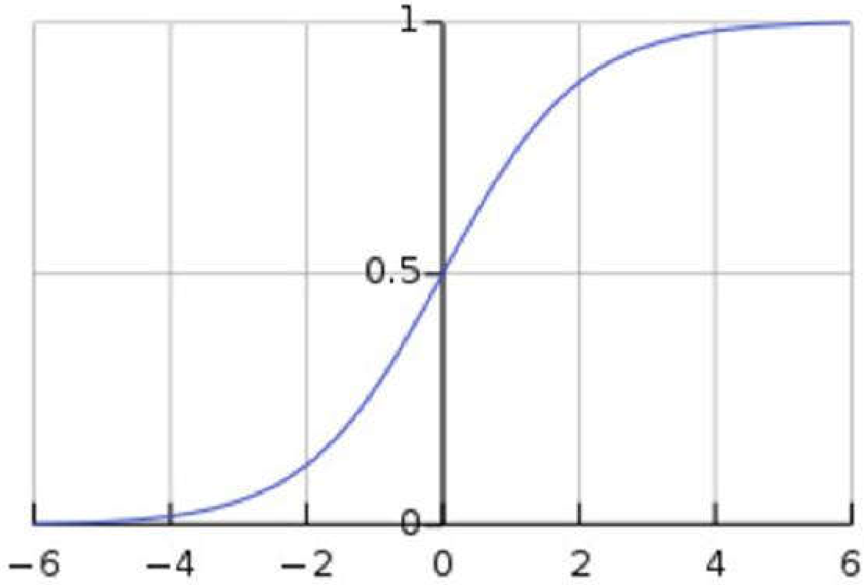
Sigmoid 函数

$\sigma(z)$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：

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

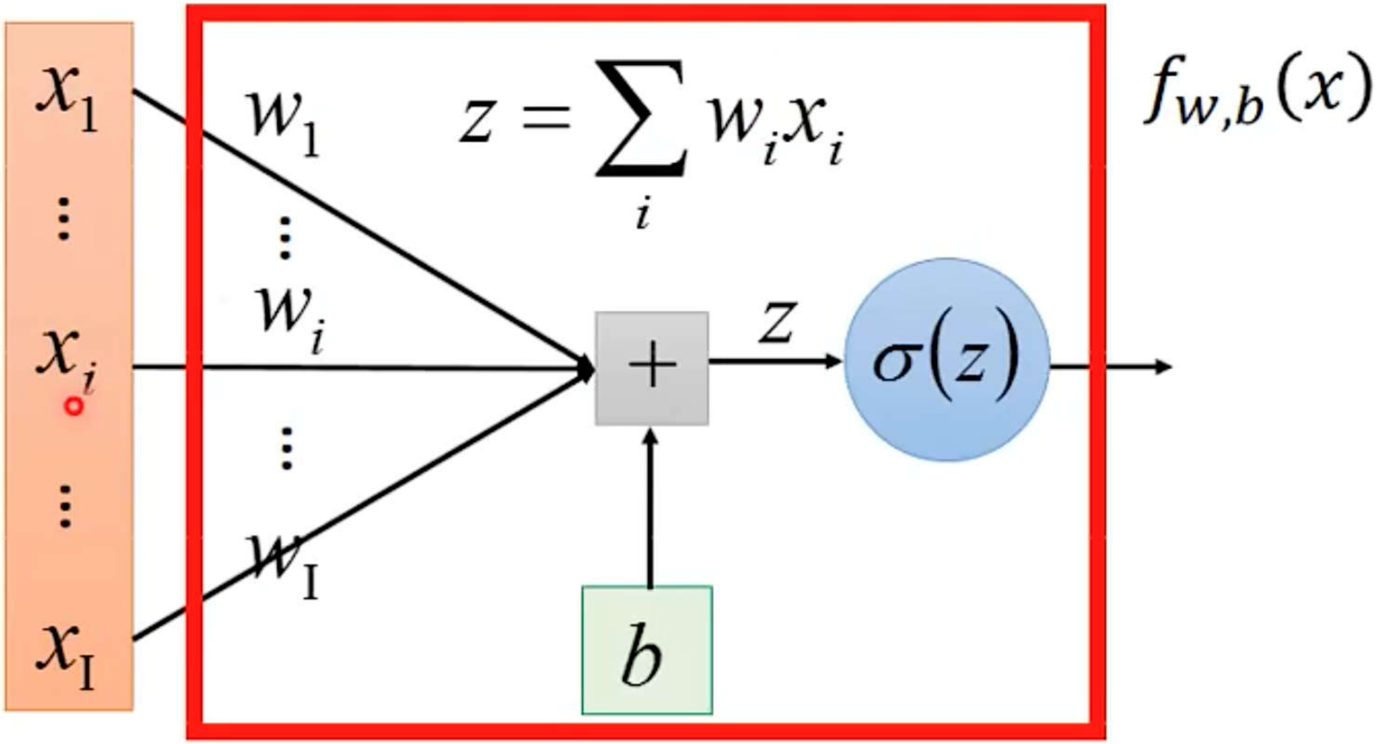


当$\sigma(z)$大于等于0.5时，预测 y=1

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：$z$=$w^T x$

Sigmoid函数



Sigmoid函数

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$[0,1]$。

或者将p视为样本x作为正例的可能性，$1-x$是其反例的可能性，则几率$\frac{p}{1-p}$反映了x作为正例的相对可能性。

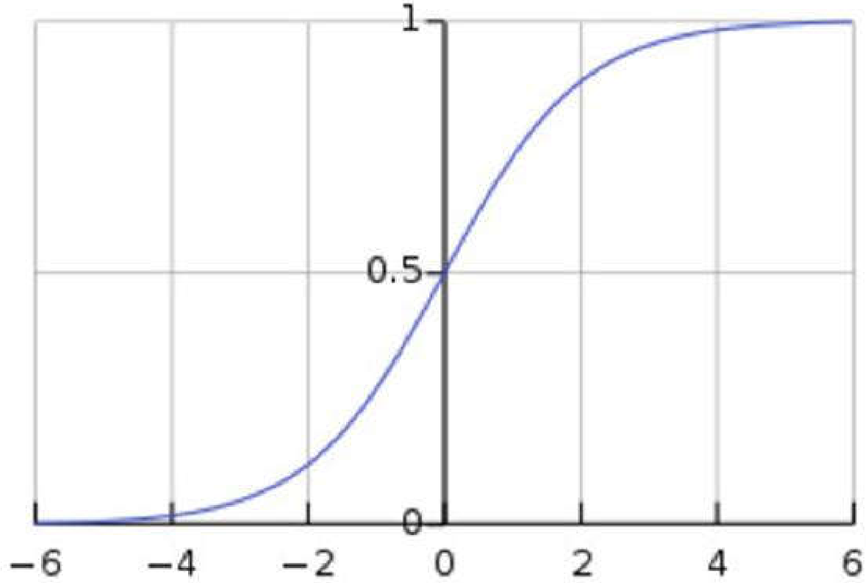
取对数得到：$\log\frac{p}{1-p}$，而$\log\frac{p}{1-p}=w^Tx=z$,即用线性回归模型的预测结果去逼近真实标记的对数几率。

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

2.Sigmoid函数

将z进行逻辑变换：$g(z) = \frac{1}{1+e^{-z}}$

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



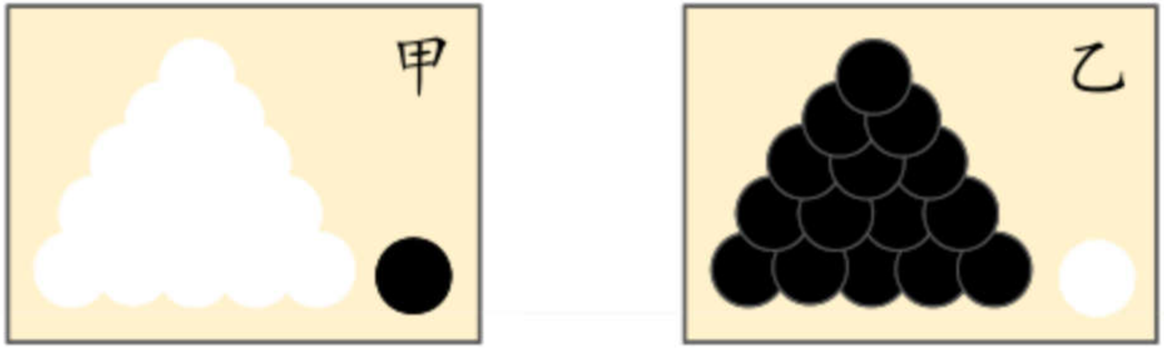
$g(z)$

3.逻辑回归求解

极大似然法

- 在线性回归中，为了估计直线的斜率和截距这两个参数，使用的方法是最小二乘法，即最小化残差平方和，
- 而在逻辑回归中，为了估计参数，使用的方法是**极大似然估计，即最大化一个似然函数，概念“似然”对应的是“可能性likelihood”**，所以也可以理解为最大化一个可能性函数。
- 极大似然估计是一种参数估计的方法：已知某个随机样本满足某种概率分布，但是其中具体的参数不清楚，参数估计就是通过若干次试验，观察其结果，利用结果推出参数的大概值。
- 最大似然估计核心：对于一些情况，样本太多，无法得出分布的参数值，可以采样小样本后，利用极大似然估计获取假设中分布的参数值。

极大似然法



- 例：有两个外形完全相同的箱子，甲箱中有99只白球，1只黑球；乙箱中有99只黑球，1只白球。一次试验取出一球，结果取出的是黑球。
- 问：黑球从哪个箱子中取出？
- 人们的第一印象就是："此黑球最像是从乙箱中取出的"，这个推断符合人们的经验事实。"最像"就是"最大似然"之意，这种想法常称为"最大似然原理"（maximum-likelihood）。

极大似然法

- 极大似然法（maximum likelihood）
  - 给定数据集

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

  - **最大化**样本属于其真实标记的概率

似然函数（linkehood function）：联合概率密度函数 $p(D|\theta)$ 称为相对于 $\{x_1, x_2, \cdots, x_N\}$ 的θ的似然函数。

$$l(\theta) = p(D|\theta) = p(x_1, x_2, \cdots, x_N|\theta) = \prod_{i=1} p(x_i|\theta)$$

- 最大化对数似然函数

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

问题求解

假设一个二分类模型：

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

这个式子可以理解为，我们将数据 x 输入到参数为 θ 的模型中，我们期望模型的输出是 y 的概率越大越好。当 $y=1$ 时，上方等号右侧的式子只剩下第一项 $(h_\theta(x))^1$ ，即当 $y=1$ 时， $P(y|x,\theta)$ 越大，也就是 $h_\theta(x)$ 越大。而当 $y=0$ 时，上方等号右侧的式子只剩下第二项 $(1-h_\theta(x))^1$ ，要使这个式子越大，也就是 $h_\theta(x)$ 越小。

则：

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

不断更新 θ 参数，输入标签为 1 的数据 x 时模型的输出越大，而输入标签为 0 的数据 x 的时模型的输出越小。

逻辑回归模型的假设是： $h(x)=g(w^Tx)=g(z)$

其中$z=w^Tx$，逻辑函数（**logistic function**)公式为：

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

问题求解

极大似然估计，即利用已知的样本结果信息，反推最具有可能（最大概率）导致这些样本结果出现的模型参数值。

求解过程：

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

作为目标函数时，有乘法常常会使用 log 函数将其转化成加法。目标是使得最大似然函数越来越大，而作为目标函数时，常把这看作为一个损失，期望不断优化模型参数，使得损失越来越小。也就是使的目标函数越来越小，所以逻辑回归的目标函数是最大对数似然函数的相反数。连乘易下溢。Log单调性不变

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：

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

问题求解

真实为1,希望尽可能判断为1;真实为0,希望尽可能判断为0

损失函数

Numpy观察log值

$\hat{y}$ 表示预测值$h(x)$

y 表示真实值

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

为什么要用这个函数作为逻辑损失函数?

使用损失函数时，希望这个误差尽可能地小，对于这个逻辑回归损失函数，也希望它尽可能地小．为了更好地理解这个损失函数怎么起作用，现举两个例子：

当 $y=1$ 时损失函数 $L=-log(\hat{y})$ ，如果想要损失函数 L 尽可能的小，那么 $\hat{y}$ 就要尽可能的大，因为 `sigmoid` 函数取值 $[0,1]$ ，所以 $\hat{y}$ 会无限接近于1。

当 $y=0$ 时损失函数 $L=-log(1-\hat{y})$ ，如果想要损失函数 L 尽可能的小，那么 $\hat{y}$ 就要尽可能的小，因为 `sigmoid` 函数取值 $[0,1]$ ，所以 $\hat{y}$ 会无限接近于0。

问题求解

交叉熵衡量的是在知道y的真实值时的平均“出乎意料”程度。当输出是期望的值，“出乎意料”程度比较低；当输出不是期望的，“出乎意料”程度就比较高。

为了衡量算法在全部训练样本上的表现如何，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

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

目标函数也可以从交叉熵的角度来理解，常常使用交叉熵函数作为二分类的损失，相比于二次代价函数有一定优势

问题求解

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

问题求解

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：

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

$$\Downarrow$$

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

问题求解

$(\mathrm{e}^x)' = \mathrm{e}^x$

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：

$(\ln x)' = \frac{1}{x}$

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

$$= -\frac{1}{m}\sum_{i=1}^{m}\left(-y^{(i)}\frac{-x_j^{(i)}e^{-w^T x^{(i)}}}{1+e^{-w^T x^{(i)}}} - (1-y^{(i)})\frac{x_j^{(i)}e^{w^T x^{(i)}}}{1+e^{w^T x^{(i)}}}\right)$$

$$-\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}\frac{e^{-w^T x^{(i)}}}{1+e^{-w^T x^{(i)}}} + y^{(i)}\frac{e^{w^T x^{(i)}}}{1+e^{w^T x^{(i)}}} - \frac{e^{w^T x^{(i)}}}{1+e^{w^T x^{(i)}}}\right)x_j^{(i)}$$

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

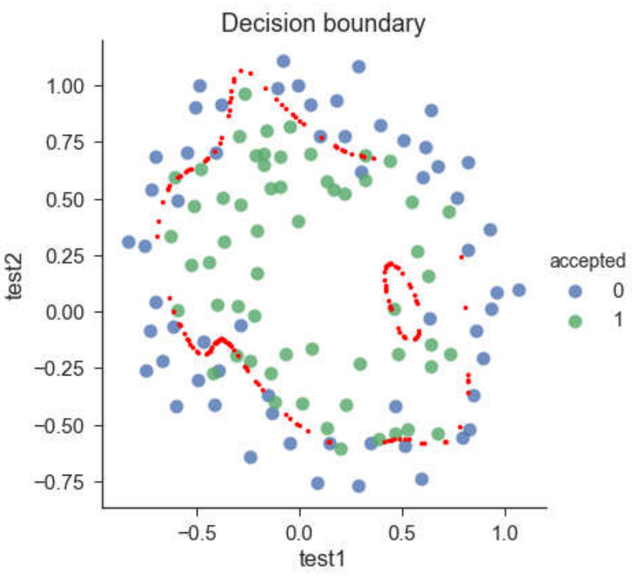
问题求解

正则化：目的是**为了防止过拟合**

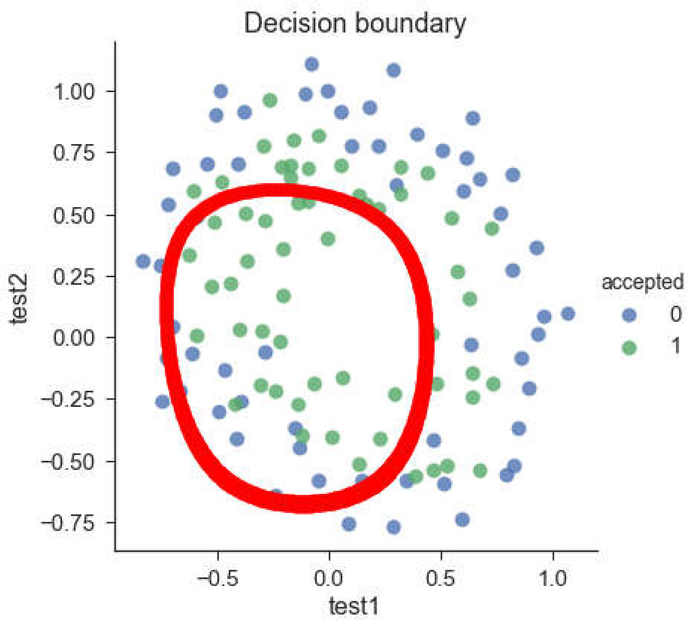
正则化项

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

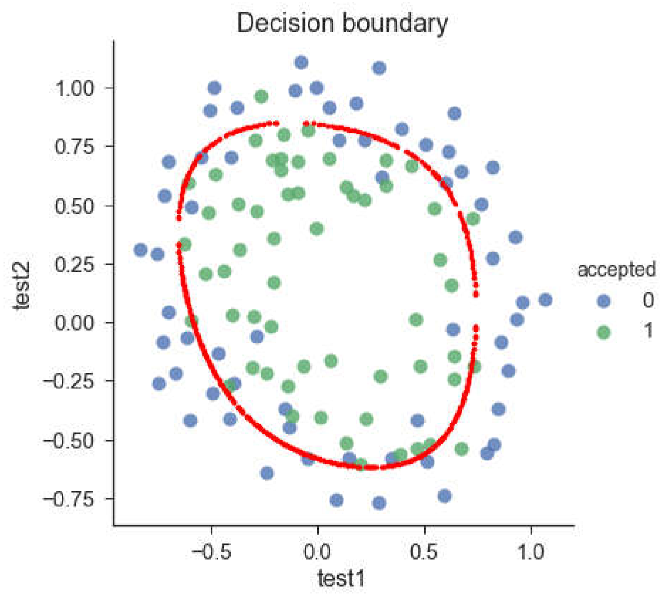
当 λ 的值开始上升时，降低了方差。



没有正则化，过拟合



正则化过度，欠拟合



适当的正则化

4.逻辑回归代码实现