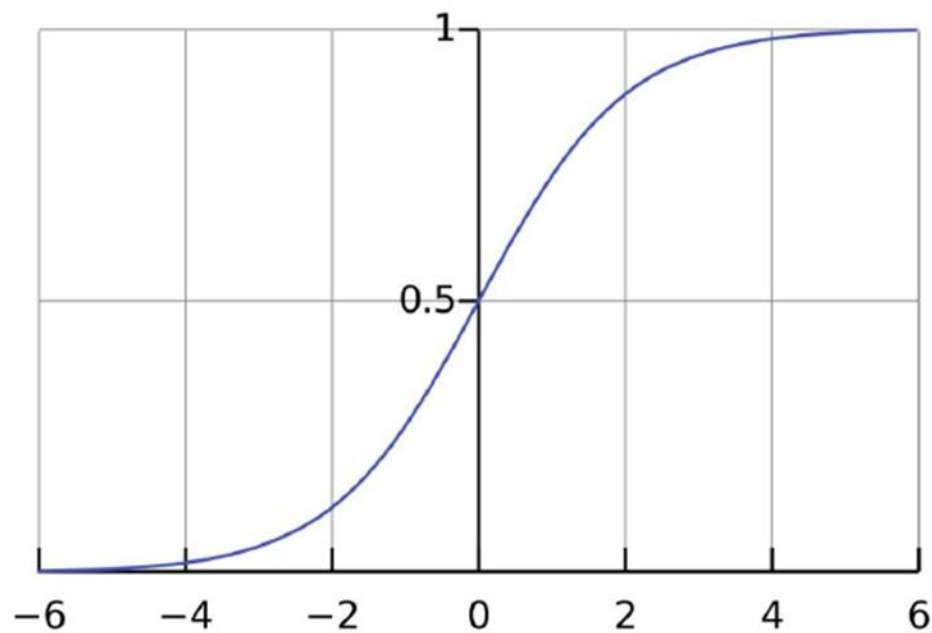
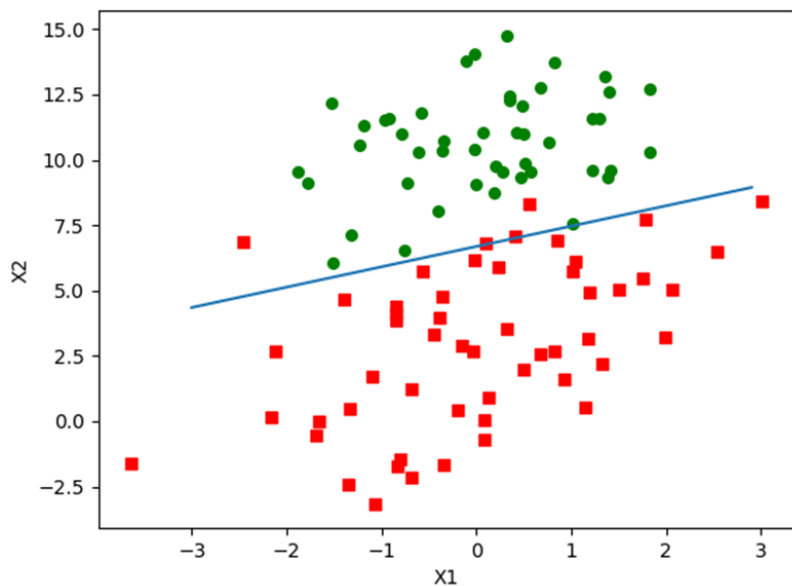


复杂一点的逻辑回归

复习

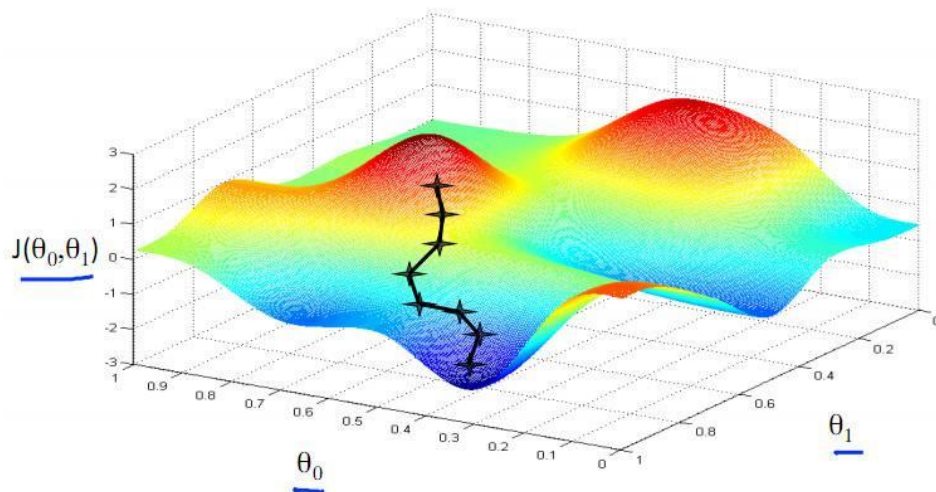


$$h_{\theta}(\mathbf{x}) = g(\boldsymbol{\theta}^T \mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \mathbf{x}}}$$
$$g(z) = \frac{1}{1 + e^{-z}}$$

复习

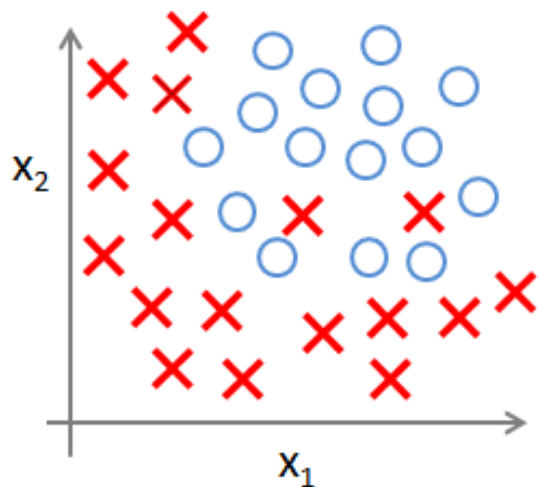
$$J(\boldsymbol{\theta}) = -\frac{1}{m} \sum_{i=1}^m \left(y^{(i)} \log h_{\boldsymbol{\theta}}(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_{\boldsymbol{\theta}}(x^{(i)})) \right)$$

$$\frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta}) = -\frac{1}{m} \sum_{i=1}^m \left(\left(y^{(i)} - h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) \right) \mathbf{x}^{(i)} \right)$$



问题

- 花费这么大的力气研究出来的模型只能画条直线，似乎有点不值。如果碰到一个复杂一点的数据集怎么办呢？

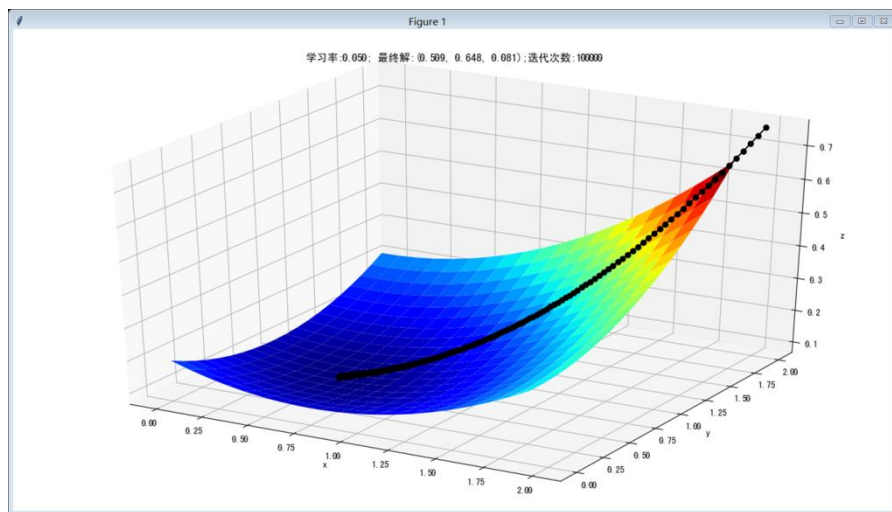


人工智能的两个基本问题

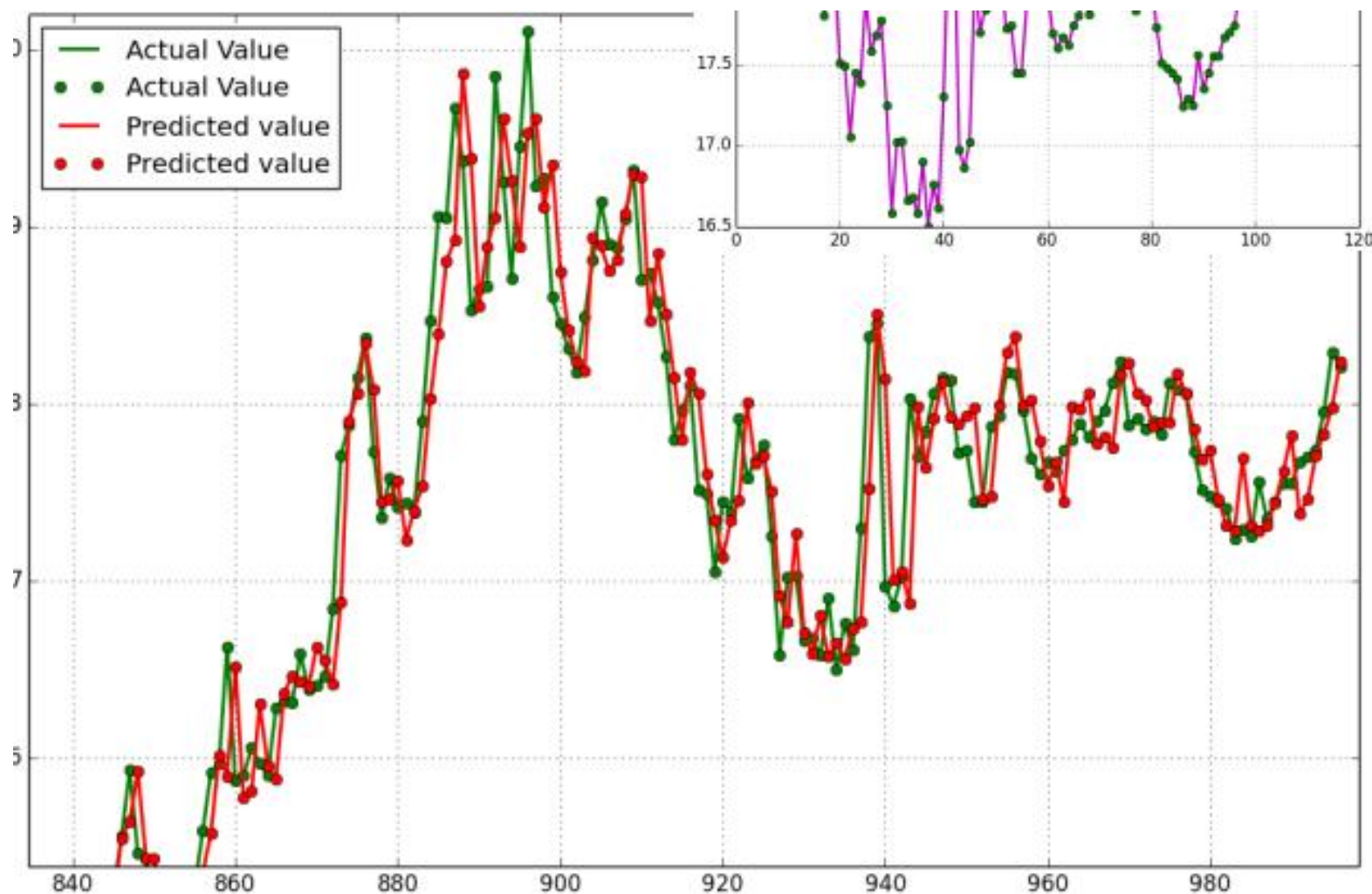
- 构建模型

$$h_{\theta}(\mathbf{x}) = \sum_{i=0}^n \theta_i x_i = \boldsymbol{\theta}^T \mathbf{x} \quad h_{\theta}(\mathbf{x}) = g(\boldsymbol{\theta}^T \mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \mathbf{x}}}$$

- 选择参数



更一般的数据集



以前的
线性回
归和逻辑回归
例子都是画直线，但这并不符合实际。

模型是否要改变

- 现实中的数据既然不总是呈现直线形式的分布，那么线性回归和逻辑回归的模型

$$h_{\theta}(\mathbf{x}) = \sum_{i=0}^n \theta_i x_i = \boldsymbol{\theta}^T \mathbf{x} \quad h_{\theta}(\mathbf{x}) = g(\boldsymbol{\theta}^T \mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \mathbf{x}}}$$

是否要重新构建？

X到底是什么（线性回归模型构建）

- 对于下面的房价数据，如果要用线性回归，模型应该为

$$h_{\theta}(\mathbf{x}) = \sum_{i=0}^n \theta_i x_i = \boldsymbol{\theta}^T \mathbf{x},$$

- 那么， $\boldsymbol{\theta}$ 和 \mathbf{x} 分别都是多少维的，每一个维度都代表什么？

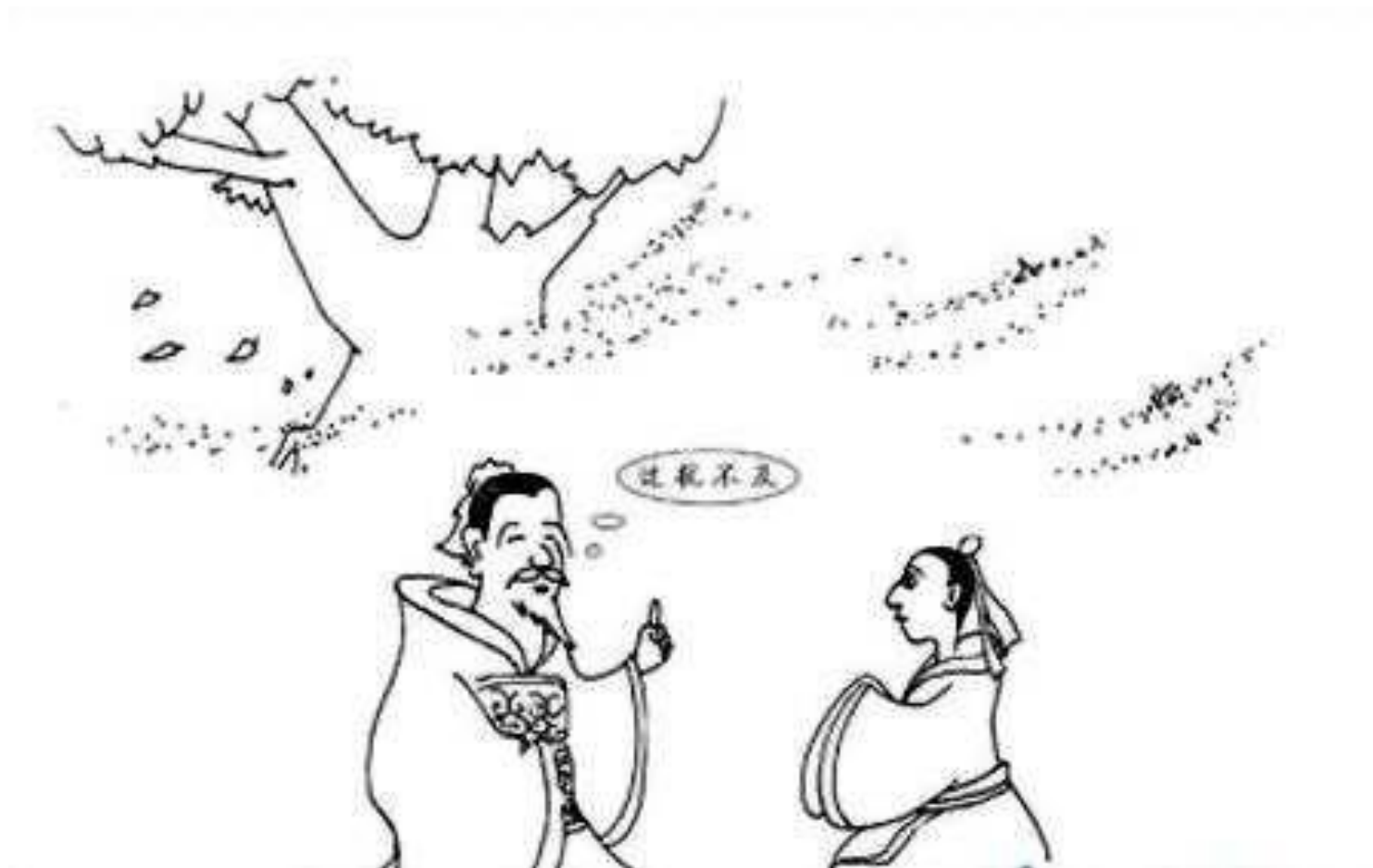
	Size (feet ²)	Number of bedrooms	Number of floors	Age of home (years)	Price (\$1000)
	2104	5	1	45	460
	1416	3	2	40	232
	1534	3	2	30	315
	852	2	1	36	178

X到底是什么

$$h_{\theta}(\mathbf{x}) = \sum_{i=0}^n \theta_i x_i = \boldsymbol{\theta}^T \mathbf{x}$$

x确实应该与训练集中的数据有关，
但不一定是训练集中的原始数据

降维VS升维



数据的维度并不是越低越好，在线性回归中加入二次项等高次项相当于增加了维度，在升维。

降维VS升维

- 由于在数据集中可能会存在与所求解问题无关的维度(数据集不一定是为了你要解决的问题专门统计的)，或者某些维度之间存在关联(比如人口统计数据中，男性占人口比例和女性占人口比例)等原因，在用分类、回归等算法求解问题前经常需要对数据进行降维。

降维VS升维

- 但是同时数据集中的原始数据可能并不能满足求解问题的需要，问题的最终解除了与原始数据的各个维度有关之外，可能还与各个维度的组合有关(比如房价除了与面积和卧室数量有关外，还有可能与面积乘以卧室数量有关)，所以还需要升维。
- 升维并不一定要重新统计数据，而是将原始数据重新进行组合，比如两两相乘。

降维VS升维

- 数据集中的每一个数据用两个特征，分别为 x_1 和 x_2 ，用逻辑回归对此数据集进行处理， $\theta^T \mathbf{x}$ 的形式不一定是 $\theta_0 + \theta_1 x_1 + \theta_2 x_2$ ，也可以是

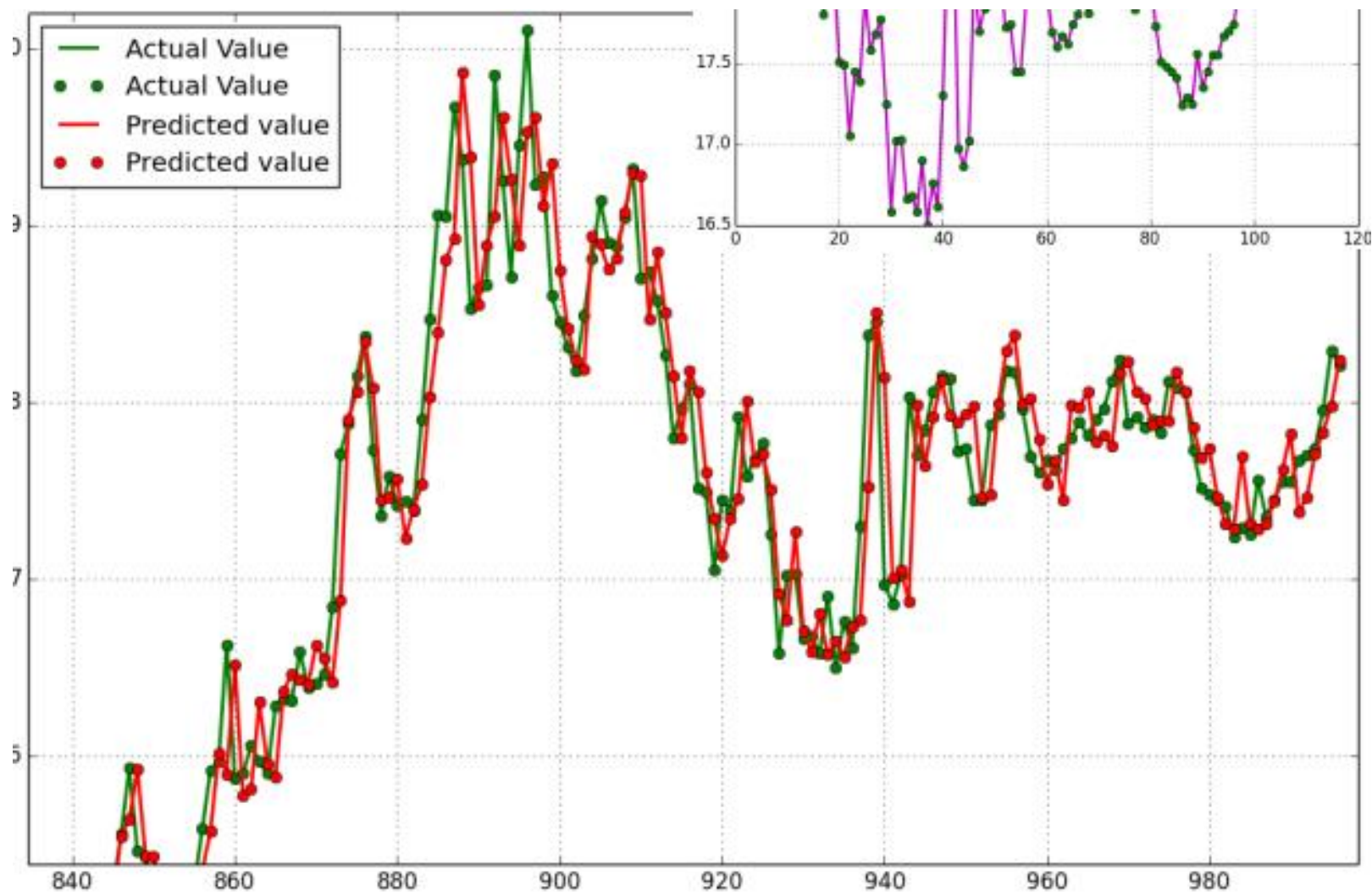
$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_2^2 + \theta_4 x_1 x_2$$

- 或者

$$\theta_0 + \theta_1 x_1 + \theta_3 x_2^2 + \theta_4 x_1 x_2$$

- 或者.....

线性回归



结论

- 所以，只需要选择合适的特征，并进行适当的组合，线性回归的模型仍然是

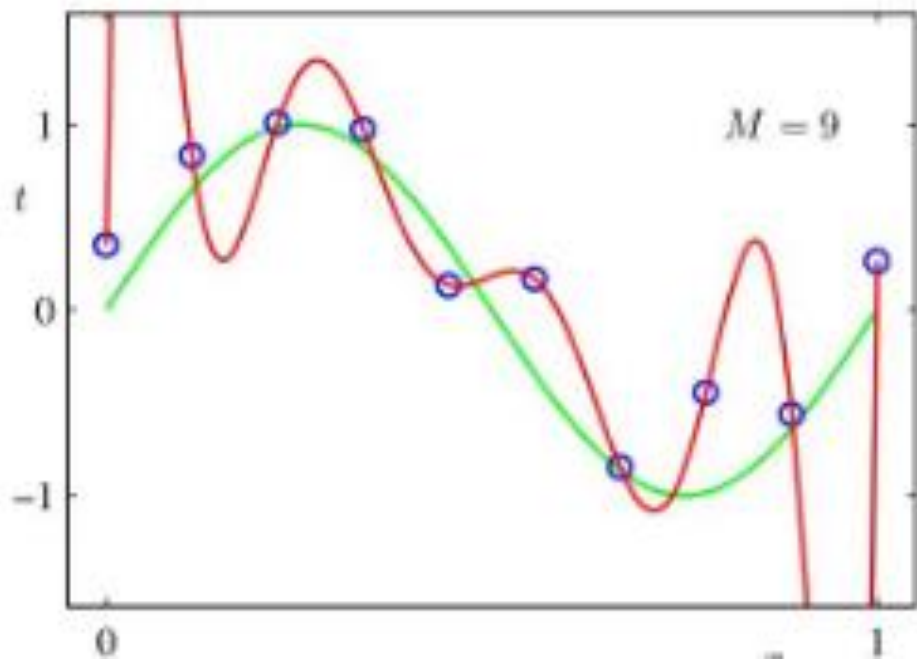
$$h_{\theta}(\mathbf{x}) = \sum_{i=0}^n \theta_i x_i = \boldsymbol{\theta}^T \mathbf{x}$$

并不需要重新构建。

- 逻辑回归同样如此。

思考

- 在线性回归中，给定二维训练集，是否一定可以通过选择合适的特征，使得代价在训练集上为0(不考虑在测试集上的表现)。



证明

- 假设有数据集 $D = \left\{ \left(x^{(1)}, y^{(1)} \right), \left(x^{(2)}, y^{(2)} \right), \dots, \left(x^{(m)}, y^{(m)} \right) \right\}$,
每个数据x都只有一个属性, 用此属性来预测y值,
x和y都是连续的(比如拿身高预测体重)。
- 假设模型 $h_{\theta}(\mathbf{x}) = \sum_{i=0}^n \theta_i x_i = \boldsymbol{\theta}^T \mathbf{x}$ 是m-1次的, 即

$$h_{\theta}(\mathbf{x}) = \boldsymbol{\theta}^T \mathbf{x} = \theta_0 + \theta_1 x + \theta_2 x^2 + \dots + \theta_{m-1} x^{m-1}$$

证明

- 把所有的数据带入刚刚假设的n-1阶的模型可以得到

$$\left\{ \begin{array}{l} \theta_0 + \theta_1 x^{(1)} + \theta_2 \left(x^{(1)} \right)^2 + \cdots + \theta_{m-1} \left(x^{(1)} \right)^{m-1} = h_{\theta} \left(x^{(1)} \right) \\ \theta_0 + \theta_1 x^{(2)} + \theta_2 \left(x^{(2)} \right)^2 + \cdots + \theta_{m-1} \left(x^{(2)} \right)^{m-1} = h_{\theta} \left(x^{(2)} \right) \\ \vdots \\ \theta_0 + \theta_1 x^{(m)} + \theta_2 \left(x^{(m)} \right)^2 + \cdots + \theta_{m-1} \left(x^{(m)} \right)^{m-1} = h_{\theta} \left(x^{(m)} \right) \end{array} \right.$$

证明

- 如果对于 $i = 1, 2, \dots, m$, 满足 $h_{\theta}(x^{(i)}) = y^{(i)}$, 即如果关于 θ 的方程组

$$\begin{cases} \theta_0 + \theta_1 x^{(1)} + \theta_2 \left(x^{(1)}\right)^2 + \dots + \theta_{m-1} \left(x^{(1)}\right)^{m-1} = y^{(1)} \\ \theta_0 + \theta_1 x^{(2)} + \theta_2 \left(x^{(2)}\right)^2 + \dots + \theta_{m-1} \left(x^{(2)}\right)^{m-1} = y^{(2)} \\ \vdots \\ \theta_0 + \theta_1 x^{(m)} + \theta_2 \left(x^{(m)}\right)^2 + \dots + \theta_{m-1} \left(x^{(m)}\right)^{m-1} = y^{(m)} \end{cases}$$

有解, 那么就说明可以找到一条曲线, 使得代价完全为0。

证明

- 上面的方程组可以写为

$$(\theta_0, \theta_1, \dots, \theta_{m-1}) \begin{pmatrix} 1 & 1 & \dots & 1 \\ x^{(1)} & x^{(2)} & \dots & x^{(m)} \\ \vdots & \vdots & \ddots & \vdots \\ \left(x^{(1)}\right)^{m-1} & \left(x^{(2)}\right)^{m-1} & \dots & \left(x^{(m)}\right)^{m-1} \end{pmatrix} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{pmatrix}^T$$


证明

• 即

$$\begin{pmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_{m-1} \end{pmatrix}^T \begin{pmatrix} 1 & 1 & \cdots & 1 \\ x^{(1)} & x^{(2)} & \cdots & x^{(m)} \\ \vdots & \vdots & \ddots & \vdots \\ \left(x^{(1)}\right)^{m-1} & \left(x^{(2)}\right)^{m-1} & \cdots & \left(x^{(m)}\right)^{m-1} \end{pmatrix} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{pmatrix}^T$$

证明

- 即

$$\begin{pmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_{m-1} \end{pmatrix}^T = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{pmatrix}^T \begin{pmatrix} 1 & 1 & \cdots & 1 \\ x^{(1)} & x^{(2)} & \cdots & x^{(m)} \\ \vdots & \vdots & \ddots & \vdots \\ \left(x^{(1)}\right)^{m-1} & \left(x^{(2)}\right)^{m-1} & \cdots & \left(x^{(m)}\right)^{m-1} \end{pmatrix}^{-1}$$


- 如果上面右边矩阵的逆矩阵存在，那么 θ 有解。

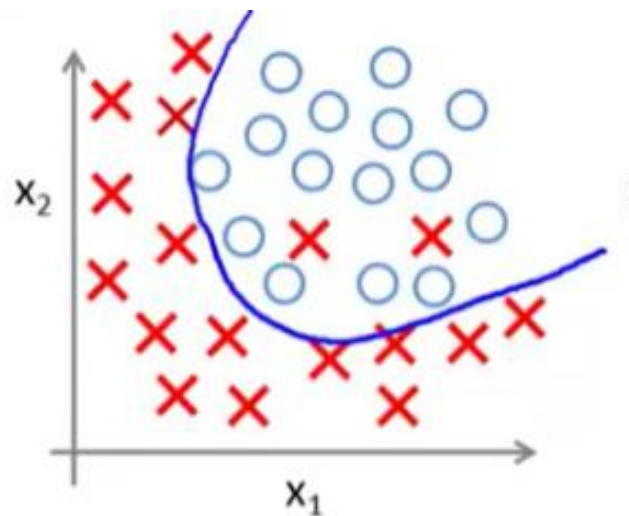
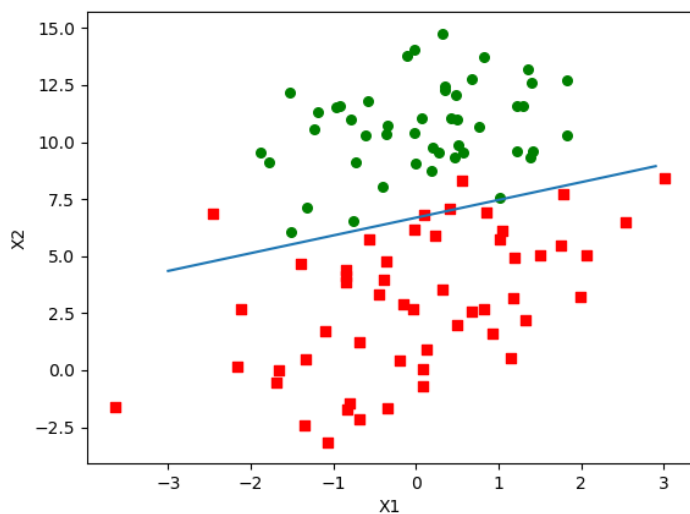
范德蒙行列式

$$D_n = \begin{vmatrix} 1 & 1 & 1 & \cdots & 1 \\ x_1 & x_2 & x_3 & \cdots & x_n \\ x_1^2 & x_2^2 & x_3^2 & \cdots & x_n^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^{n-1} & x_2^{n-1} & x_3^{n-1} & \cdots & x_n^{n-1} \end{vmatrix} = \prod_{i,j(n \geq i > j \geq 1)} (x_i - x_j)$$

证明

- 由范德蒙行列式可以知道如果训练样本中每一个样本的 x 都不相同，那么逆矩阵就一定存在，即 θ 一定有解。
- 所以 m 个训练数据组成的训练集一定可以找到一个 $m-1$ 次的函数使得代价为0。
- 低于 $m-1$ 次的函数可能存在也可能不存在。

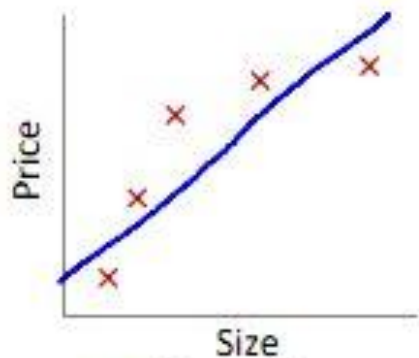
逻辑回归



逻辑回归与线性回归相同，只要次数够高，一定可以找到一条曲线将两类数据完全分开

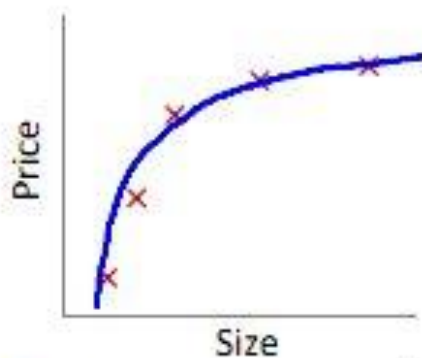
过拟合问题

欠拟合和过拟合

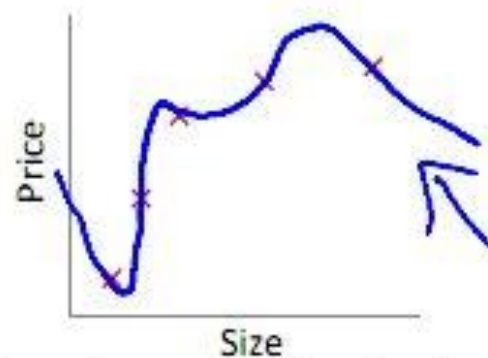


$\rightarrow \theta_0 + \theta_1 x$
"Underfit" "High bias"

没有高次项，代
价高



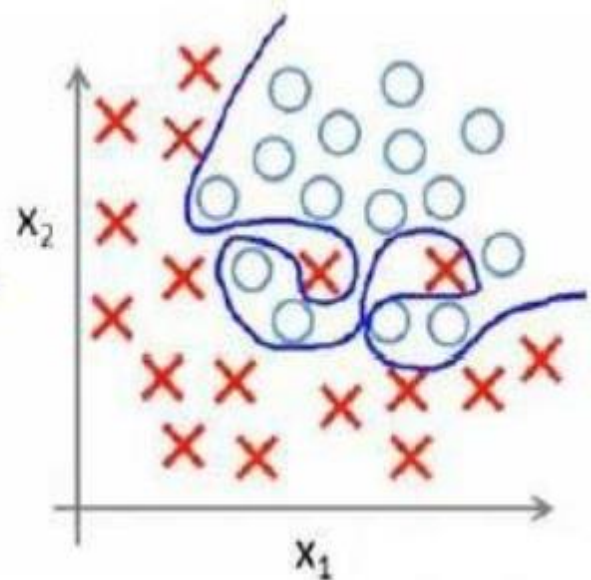
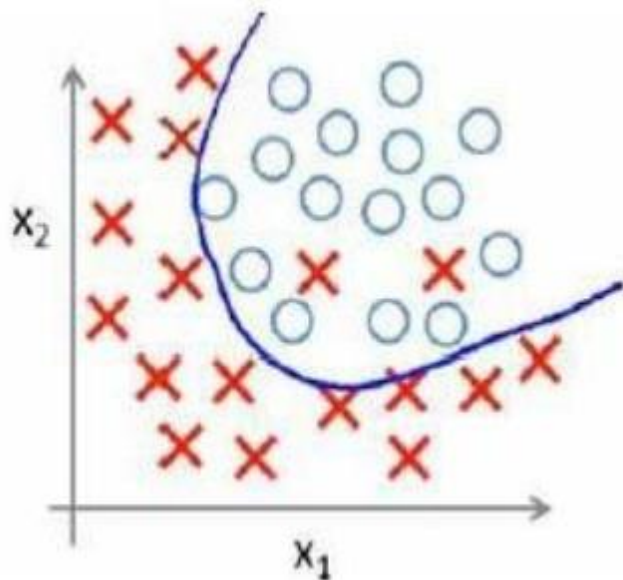
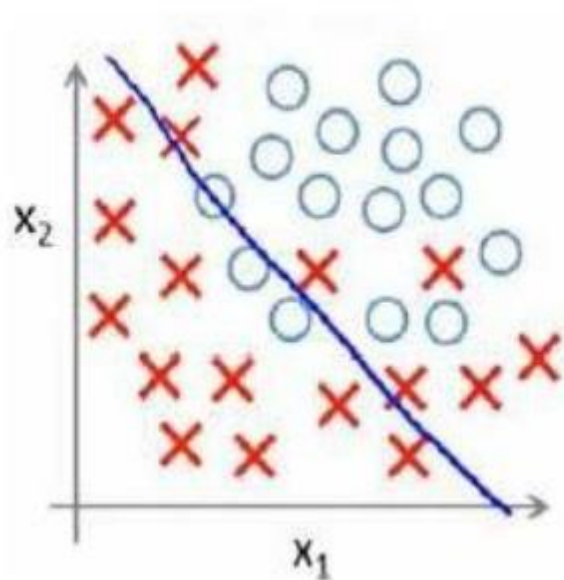
$\rightarrow \theta_0 + \theta_1 x + \theta_2 x^2$
"Just right"



$\rightarrow \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$
"Overfit" "High variance"

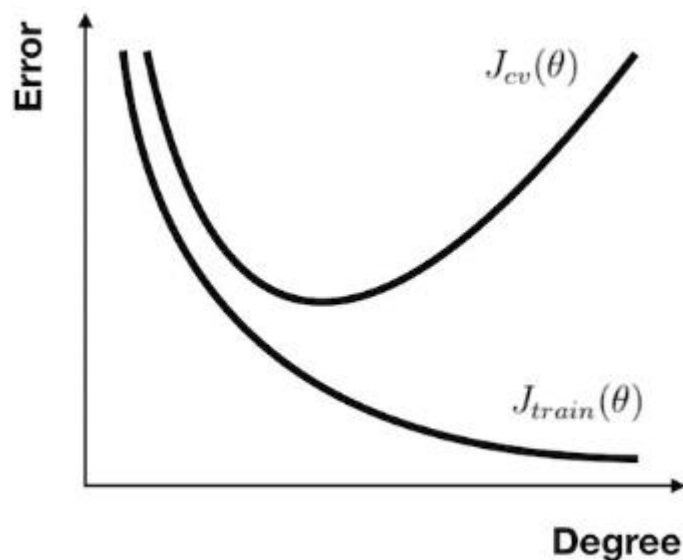
高次项太多，代价
低，但是不适合预
测

逻辑回归的过拟合



过拟合

- 在线性回归中，随着模型最高次项次数的提高，会有如图所示的规律。其中 J_{cv} 表示在交叉验证集上的代价， J_{train} 表示在训练集上的代价。



过拟合

一個非洲酋長到倫敦訪問，一群記者在機場截住了他。

早上好，酋長先生”，其中一人問道：你的路途舒適嗎？

酋長發出了一連串刺耳的聲音哄、哼、啊、吱、嘶嘶，

然后用純正的英語說道：是的，非常地舒適。

那麼！您準備在這裡待多久？

他發出了同樣的一連串噪音，

然後答：大約三星期，我想。

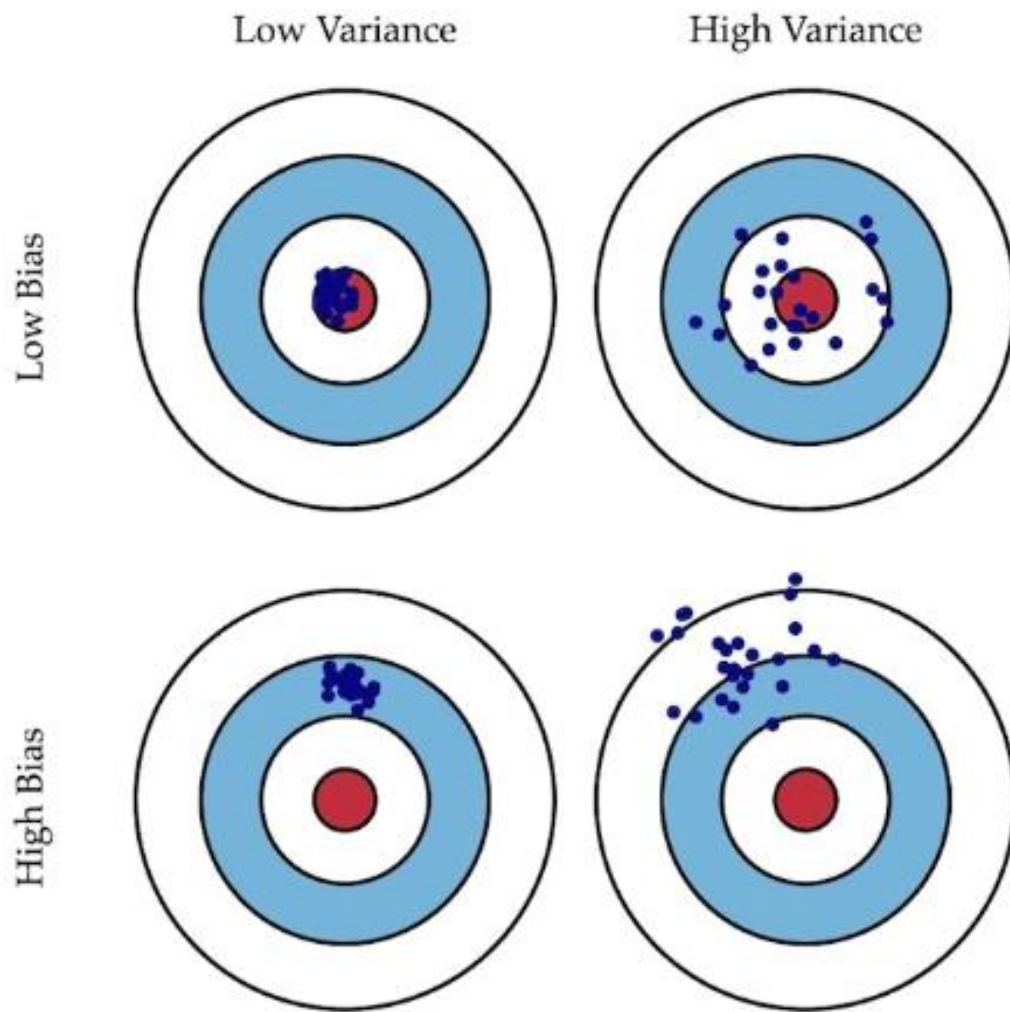
酋長，告訴我，你是在哪學的這樣流利的英語？迷惑不解的記者問。

又是一陣哄、吭、啊、吱、嘶嘶聲，

酋長說：從短波收音機裡。

训练数据中会含有各种误差(称为噪声)，所以一味地追求模型与训练数据完全吻合是不行的。

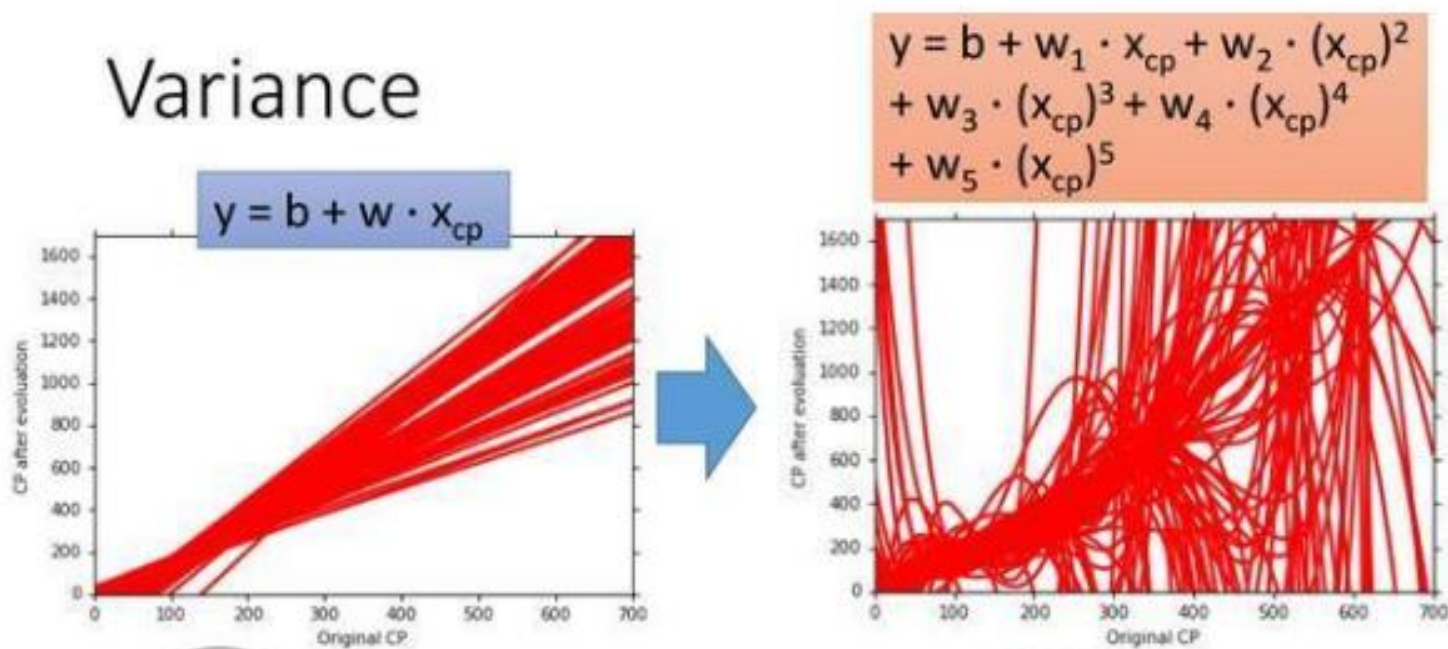
偏差和方差



- 偏差表示模型的预测值与真实值之间的差距。如果模型的偏差比较高，说明没有很好的拟合，属于欠拟合
- 方差表示的是模型的稳定程度。如果方差比较高说明模型在训练集表现的很好，但在非训练集上表现的很差。

例

- 假设有5000个样本，每10个用来训练一个模型，那么就会有500个模型，分别选用一次模型和五次模型时，结果如下图。



例

- 从图中可以看出，使用一次模型时，不同的数据训练出的模型差别不大，模型的稳定性较好(虽然效果可能是稳定的差)，在训练自己所使用的10个数据和其他数据上的表现差不多。
- 而使用五次模型时，不同数据训练处的模型差别很大，那么如果使用10个数据训练出一个五次模型，用此模型来预测其他4990个数据很可能得不到好的结果。

防止过拟合的方法



牛客网
NOWCODER

首页

题库

面试^N

学习

求职

讨论区

发现

首页 > 试题广场 > 在模型训练过程中，下列哪些方法可以防止模型过拟合（overf

? [不定项选择题]

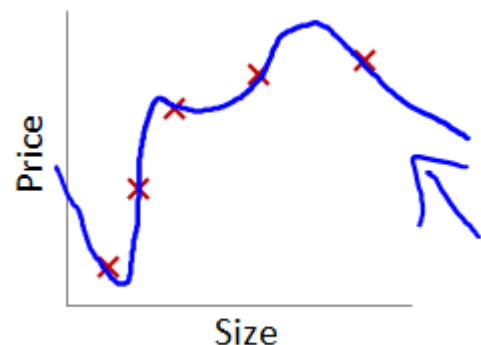
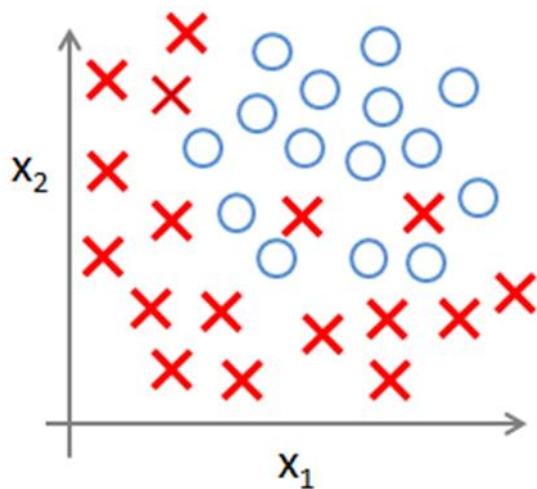
在模型训练过程中，下列哪些方法可以防止模型过拟合（overfitting）：

- A. 增大数据量
- B. 减少feature个数
- C. 正则化
- D. 交叉验证

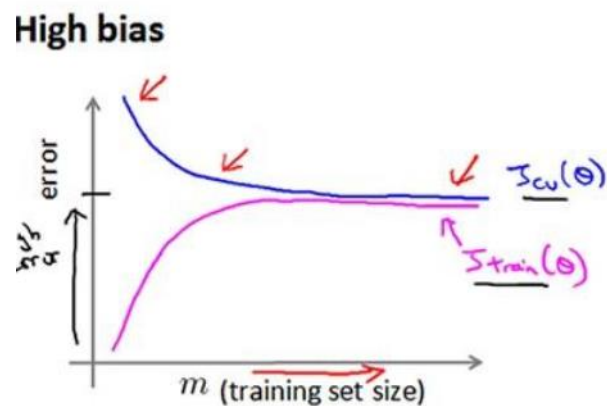
正确答案：A B C D

增大数据量

- 增大训练集的数据量一般可以减轻过拟合，如果数据量增大后训练出来的模型在测试集上的效果没有提升，很有可能是欠拟合。



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$



交叉验证

1. $h_{\theta}(x) = \theta_0 + \theta_1 x$
2. $h_{\theta}(x) = \theta_0 + \theta_1 x + \theta_2 x^2$
3. $h_{\theta}(x) = \theta_0 + \theta_1 x + \cdots + \theta_3 x^3$
- \vdots
10. $h_{\theta}(x) = \theta_0 + \theta_1 x + \cdots + \theta_{10} x^{10}$

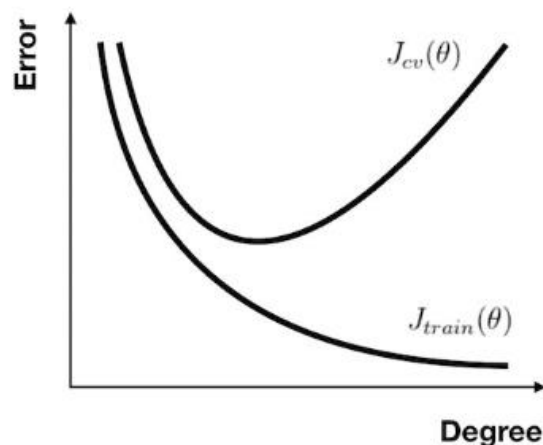
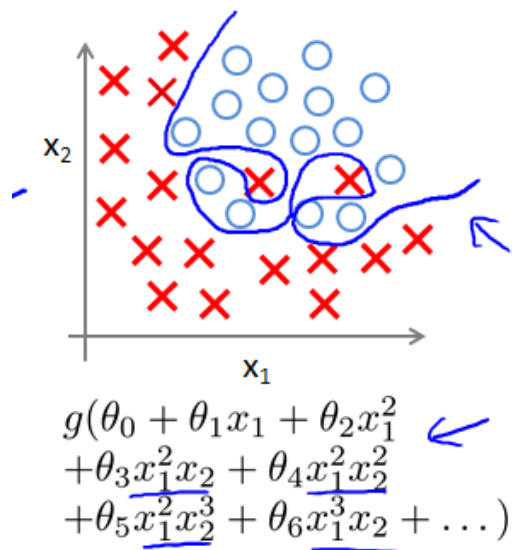
通过交叉验证的方法来确定哪个模型最好。
逻辑回归的模型选择与线性回归类似。

三个数据集

- 训练集、验证集、测试集

正则化

- 从前面的分析可以看出，过拟合一般是由于高次项的次数过高引起的。
- 通过在代价函数中加入惩罚项来使得高次项的 θ 减小，达到防止过拟合的目的。



鸡肋问题

- 人工智能经常面临的问题就是有些高次项或者某些特征等可能并不会对模型起到什么好的作用，甚至有可能起反作用，但是由于人工智能是在预测，并不能确定这些东西没用，所以…



怎么使得高次项的系数缩小

- 我们可以直接缩小高次项的系数，比如降低到原来的十分之一，但是：第一，所有的系数都是通过梯度下降计算得来的，如果直接缩小高阶项的系数，就破坏了梯度下降的规则，缩小后的结果是否还是一个比较好的结果很难说；第二，这不符合人工智能的“规矩”。

L2正则化

- 线性回归的L2正则化:

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

- 逻辑回归的L2正则化:

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

j从1开始，常数项不惩罚

L2正则化后 θ 的更新方法

- 由于 θ_0 为参与惩罚，所以求偏导时需要对其区别对待

Repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

$$\theta_j := \theta_j (1 - \alpha \frac{\lambda}{m}) - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

$$(j = \text{✗}, 1, 2, 3, \dots, n)$$

}

正则化为什么可以解决过拟合

- 从直观的角度来讲，我们的目的是要使得代价函数最小化，而代价函数中加入了求和项，自然是项数越少，求和项的值越小，所以梯度下降会降低每个 θ 的值，同时尽量减少 θ 的数量。由于高次项的变化会使得代价函数变得更快，而梯度下降是沿着函数变化最快的方向下降，所以可能更倾向去降低高次项的 θ （其实未必）。

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

L1正则化

- 除了L2正则化外，常用的正则化方法还有L1正则化。线性回归的L1正则化方法如下：

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

- 逻辑回归的L1正则化方法为：

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

例

线性回归：5阶，系数为： [21.59733285 -54.12232017 38.43116219 -12.68651476 1.98134176 -0.11572371]

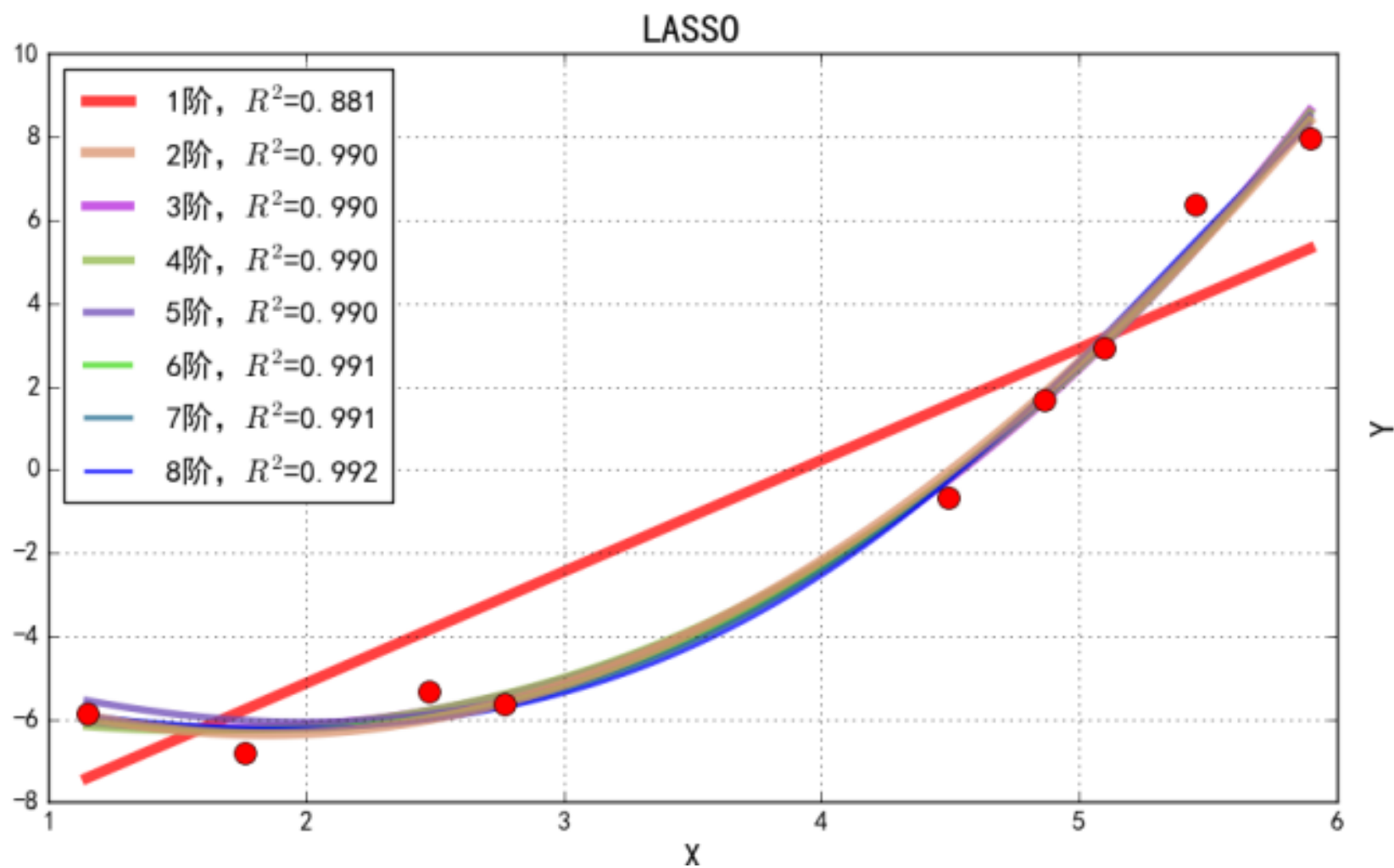
线性回归：6阶，系数为： [14.73304784 -37.87317493 23.67462341 -6.07037979 0.42536833 0.06803132 -0.00859246]

线性回归：7阶，系数为： [314.30344773 -827.89447316 857.33293588 -465.46543853 144.21883915 -25.67294689 2.44658613]

线性回归：8阶，系数为： [-1189.50198207 3643.69252986 -4647.93115 3217.22929147 -1325.87429346 334.32879953 -50.571]

LASSO：8阶，alpha=0.001000，系数为： [-4.62623251 -1.37717809 0.17183854 0.04307765 0.00629505 0.00069171 0.0000355 -0.00000000]

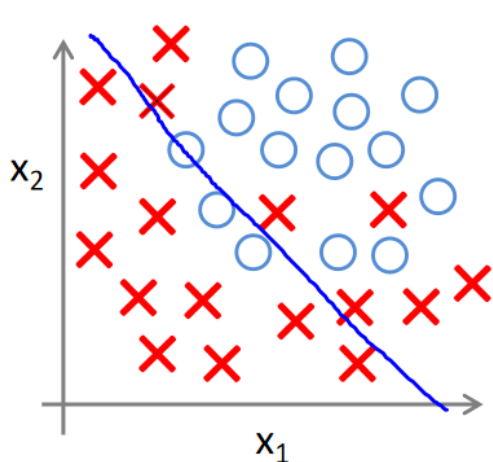
例



λ 怎么选

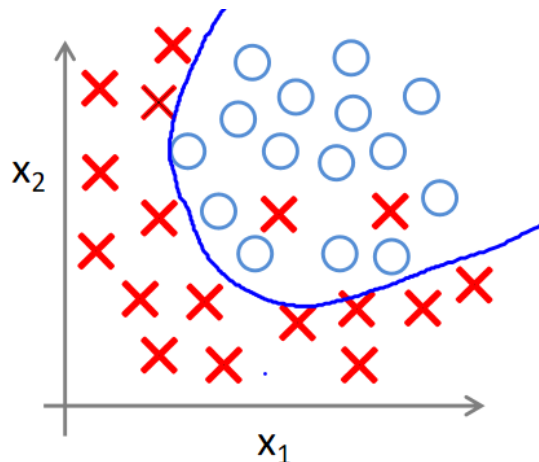
- 参考梯度下降学习率学习率

逻辑回归的正则化

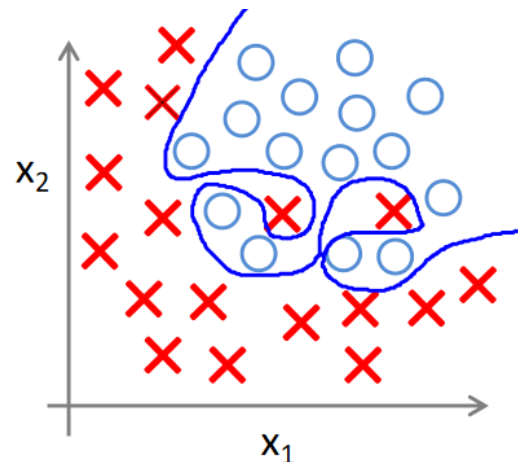


$$h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

(g = sigmoid function)



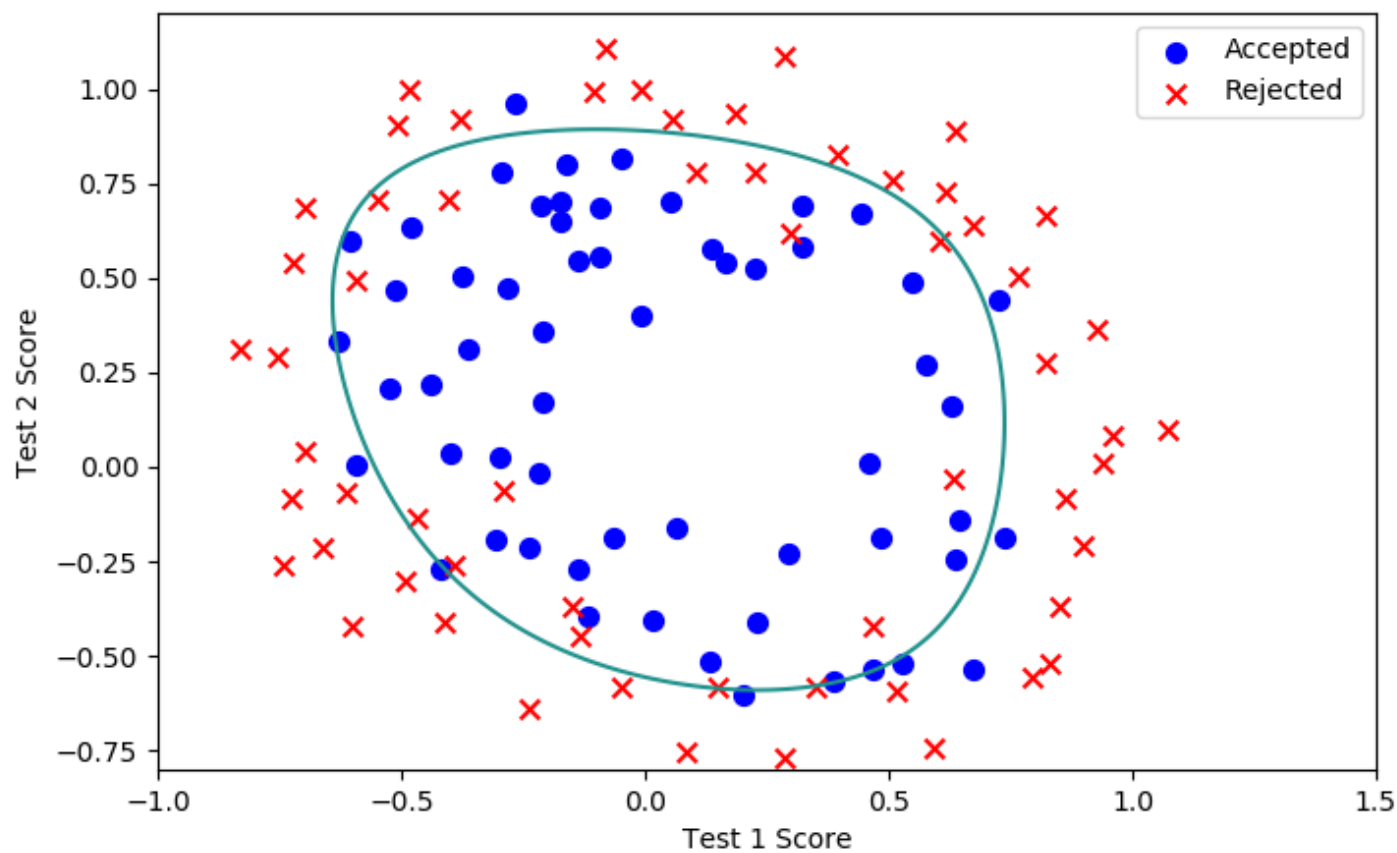
$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$$



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots)$$

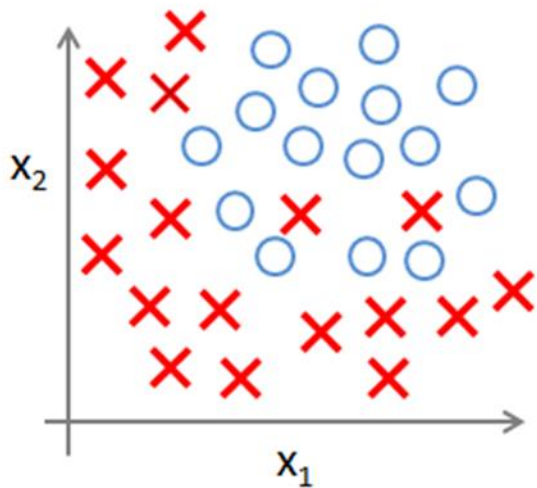
逻辑回归的正则化方法与线性回归相同。

逻辑回归的正则化示例



总结

- 逻辑回归可以解决下图中的问题。



- 逻辑回归解决上图问题的模型仍然是

$$h_{\theta}(\mathbf{x}) = g(\boldsymbol{\theta}^T \mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \mathbf{x}}}$$

总结

- 逻辑回归解决上一页问题的办法是把原始数据的各个属性进行组合，形成高次项。
- 高次项容易造成过拟合。
- 过拟合的解决办法有：增加数据量、交叉验证、正则化等。