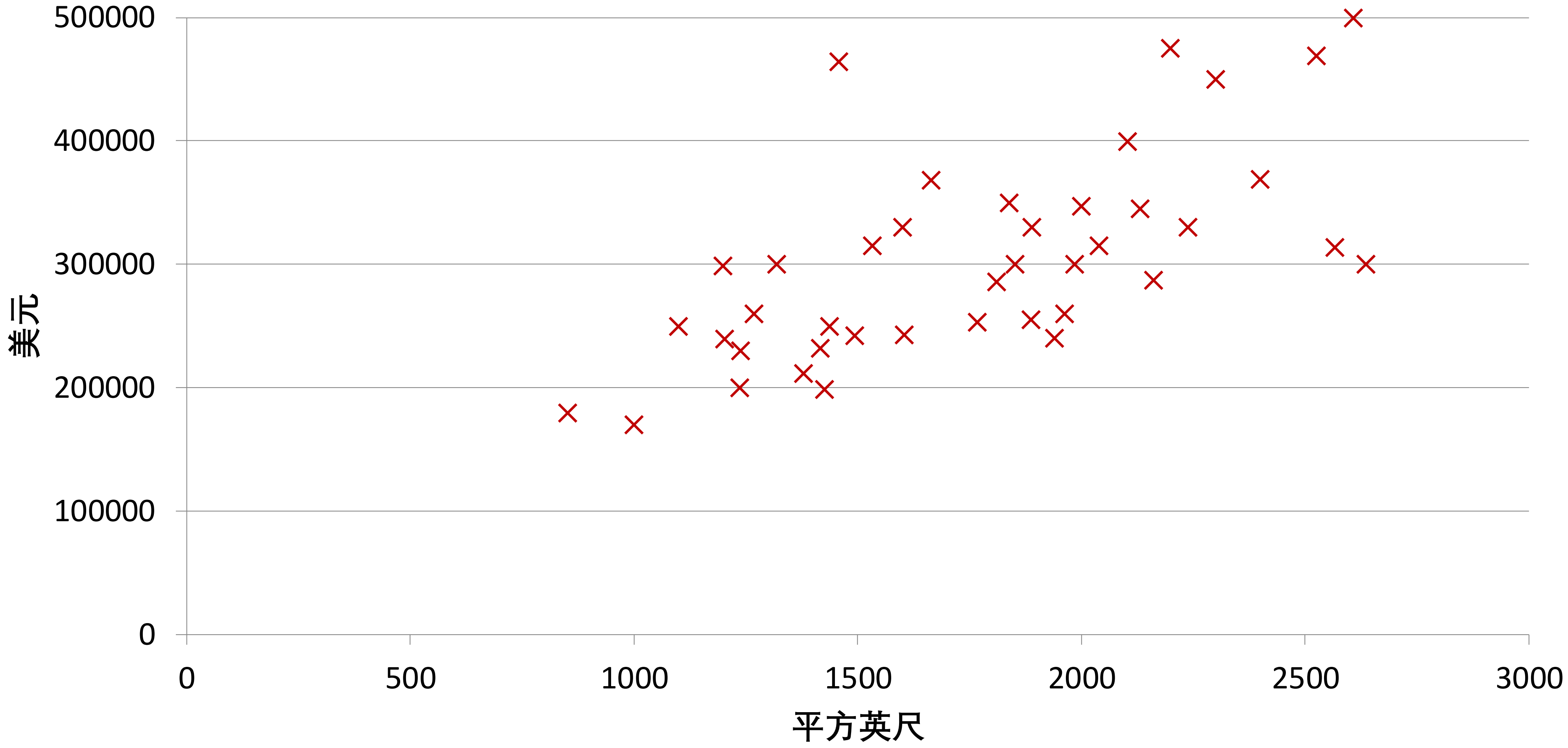
回归分析
Regression

回归(Regression)

- 回归一词最早由英国科学家弗朗西斯·高尔顿（Francis Galton）提出，他还是著名的生物学家、进化论奠基人查尔斯·达尔文（Charles Darwin）的表弟。高尔顿深受进化论思想的影响，并把该思想引入到人类研究，从遗传的角度解释个体差异形成的原因。
- 高尔顿发现，虽然有一个趋势：父母高，儿女也高；父母矮，儿女也矮。但给定父母的身高，儿女辈的平均身高却趋向于或者“回归”到全体人口的平均身高。换句话说，即使父母双方都异常高或者异常矮，儿女的身高还是会趋向于人口总体的平均身高。这也就是所谓的普遍回归规律。
- 高尔顿的这一结论被他的朋友，英国数学家、数理统计学的创立者卡尔·皮尔逊（Karl Pearson）所证实。皮尔逊收集了一些家庭的1000多名成员的身高记录，发现对于一个父亲高的群体，儿辈的平均身高低于他们父辈的身高；而对于一个父亲矮的群体，儿辈的平均身高则高于其父辈的身高。这样就把高的和矮的儿辈一同“回归”到所有男子的平均身高，用高尔顿的话说，这是“回归到中等”。

房价预测

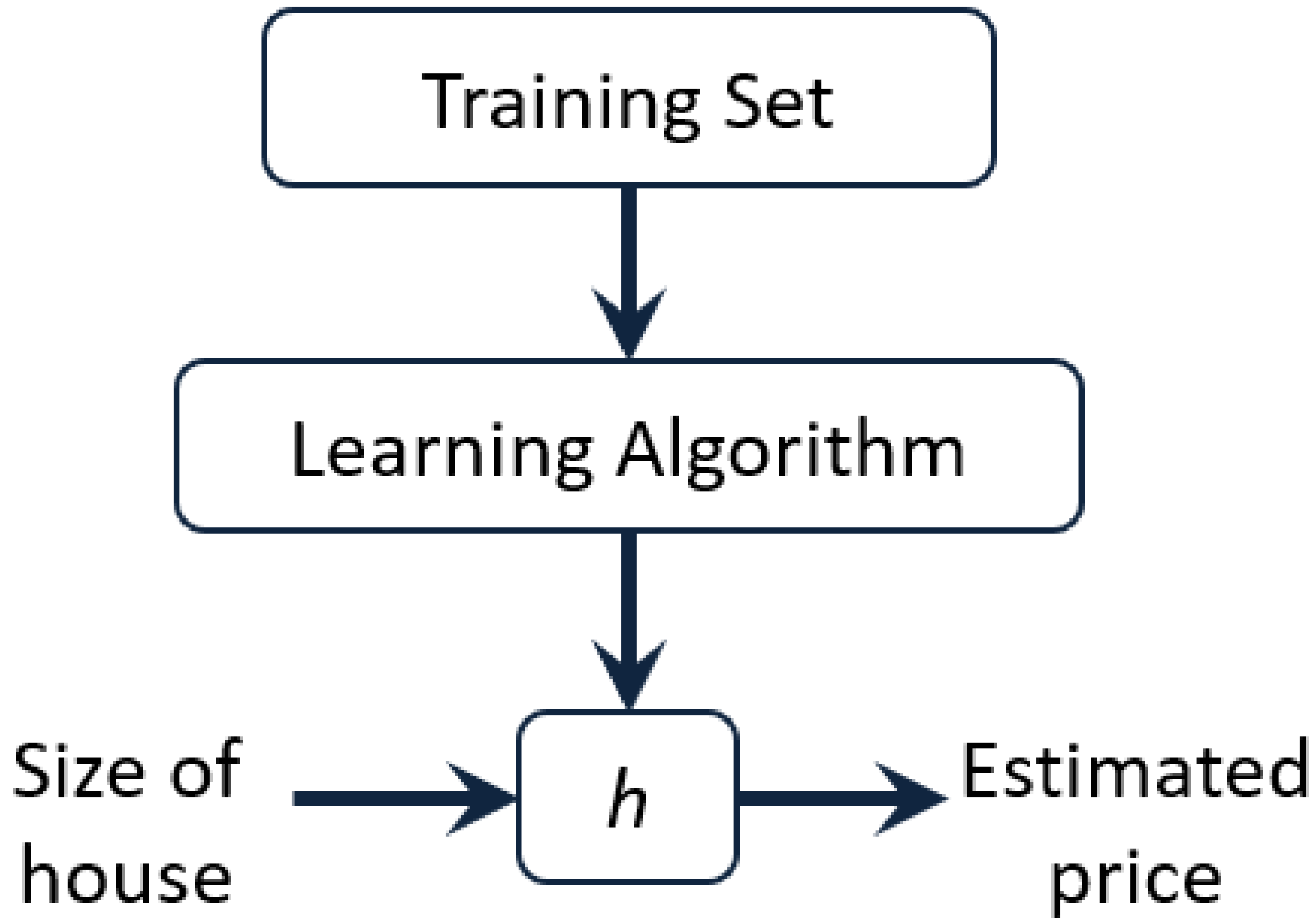

500000
400000
300000
200000
100000
0
美元
0
500
1000
1500
2000
2500
3000
平方英尺

房价预测

Size in feet² (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
…	…

特征值(feature)　　　　标签/结果(target)

房价预测


Training Set
Learning Algorithm
Size of house
h
Estimated price

?

一元线性回归

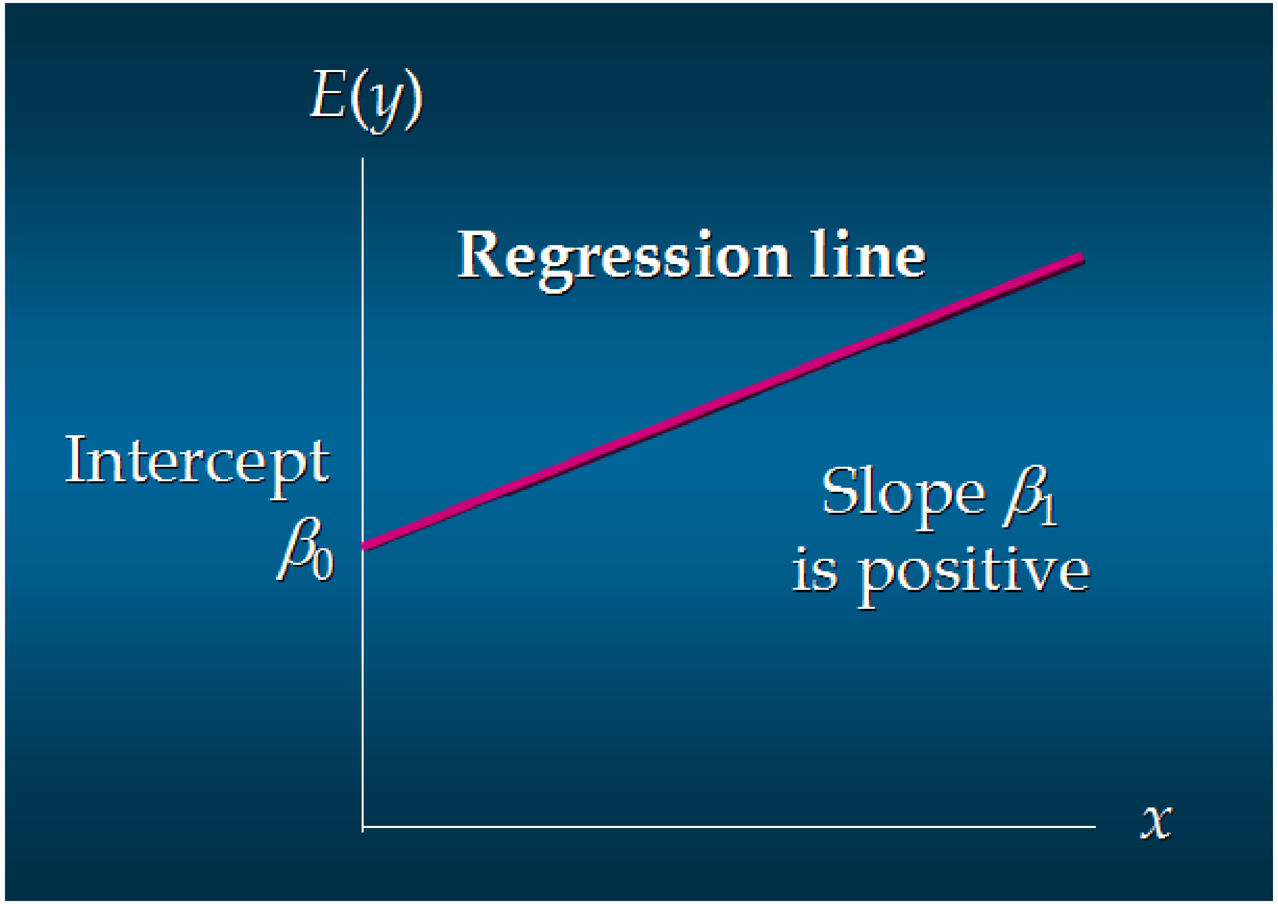
- 回归分析(regression analysis)用来建立方程模拟两个或者多个变量之间如何关联
- 被预测的变量叫做：因变量(dependent variable), 输出(output)
- 被用来进行预测的变量叫做： 自变量(independent variable), 输入(input)
- 一元线性回归包含一个自变量和一个因变量
- 以上两个变量的关系用一条直线来模拟
- 如果包含两个以上的自变量，则称作多元回归分析(multiple regression)

一元线性回归

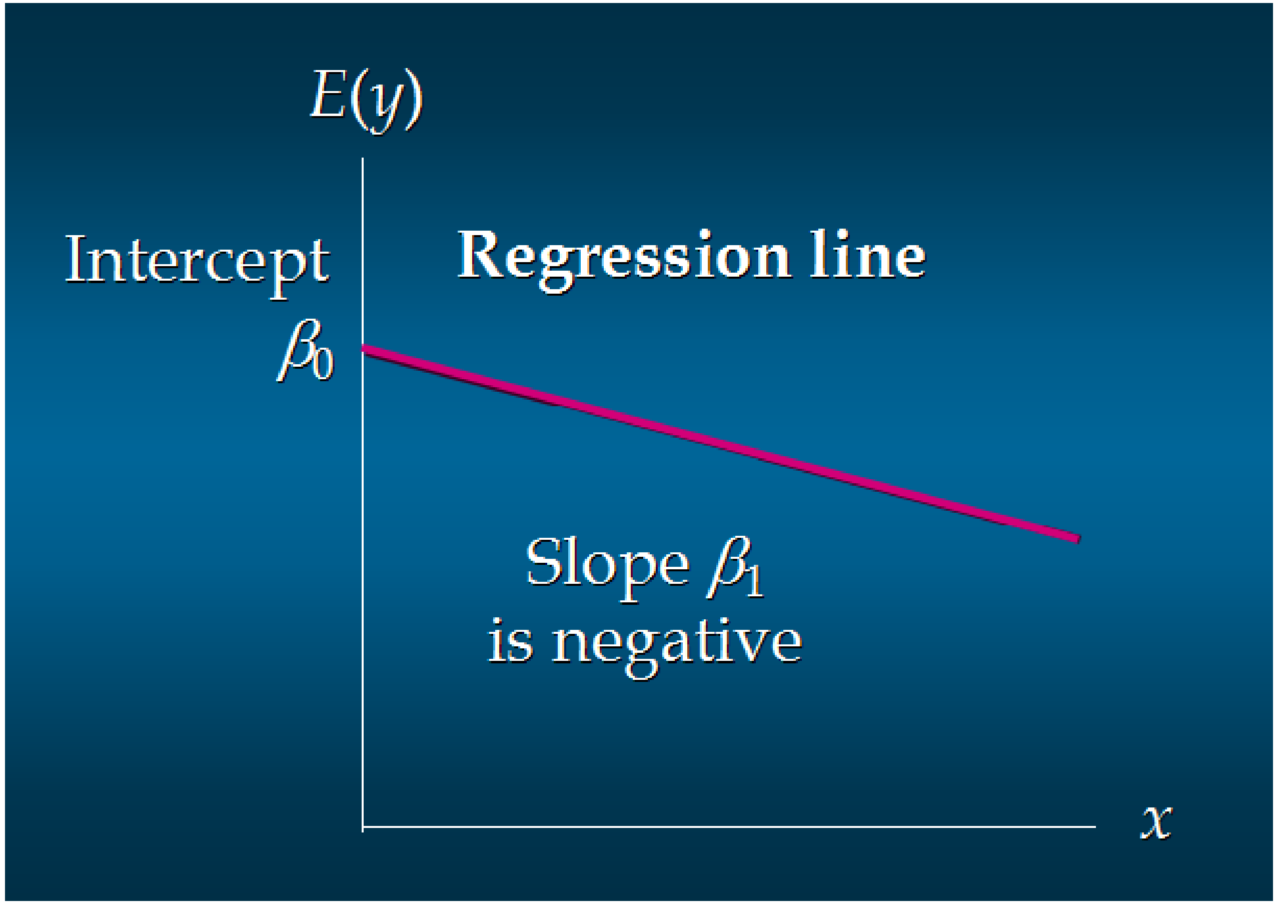
$$h_\theta(x) = \theta_0 + \theta_1 x$$

这个方程对应的图像是一条直线，称作回归线。其中，θ_1为回归线的斜率，θ_0为回归线的截距。

一元线性回归-正相关

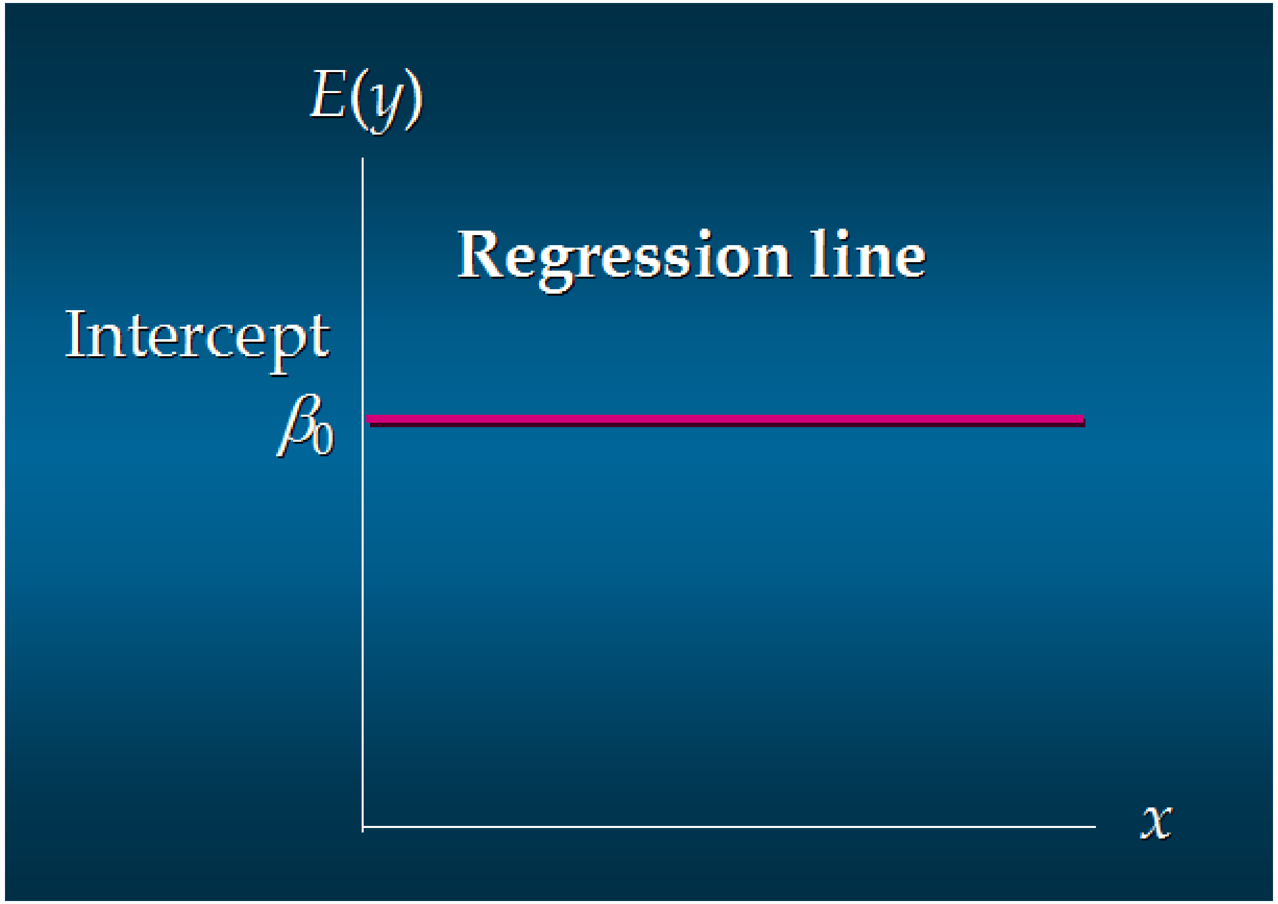

$E(y)$
Regression line
Intercept
β_0
Slope β_1
is positive
x

一元线性回归-负相关



E(y)
Intercept
β_0
Regression line
Slope β_1
is negative
x

一元线性回归-不相关



求解方程系数

训练集：

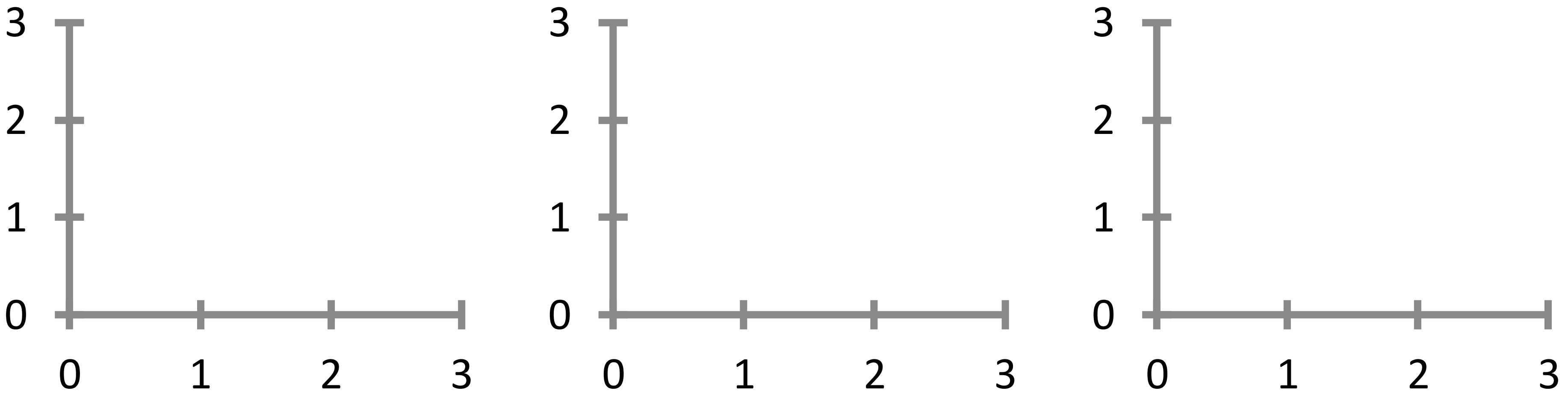
Size in feet² (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
…	…

需要求解方程：

$$h_\theta(x) = \theta_0 + \theta_1 x$$

求解方程系数

$$h_\theta(x) = \theta_0 + \theta_1 x$$

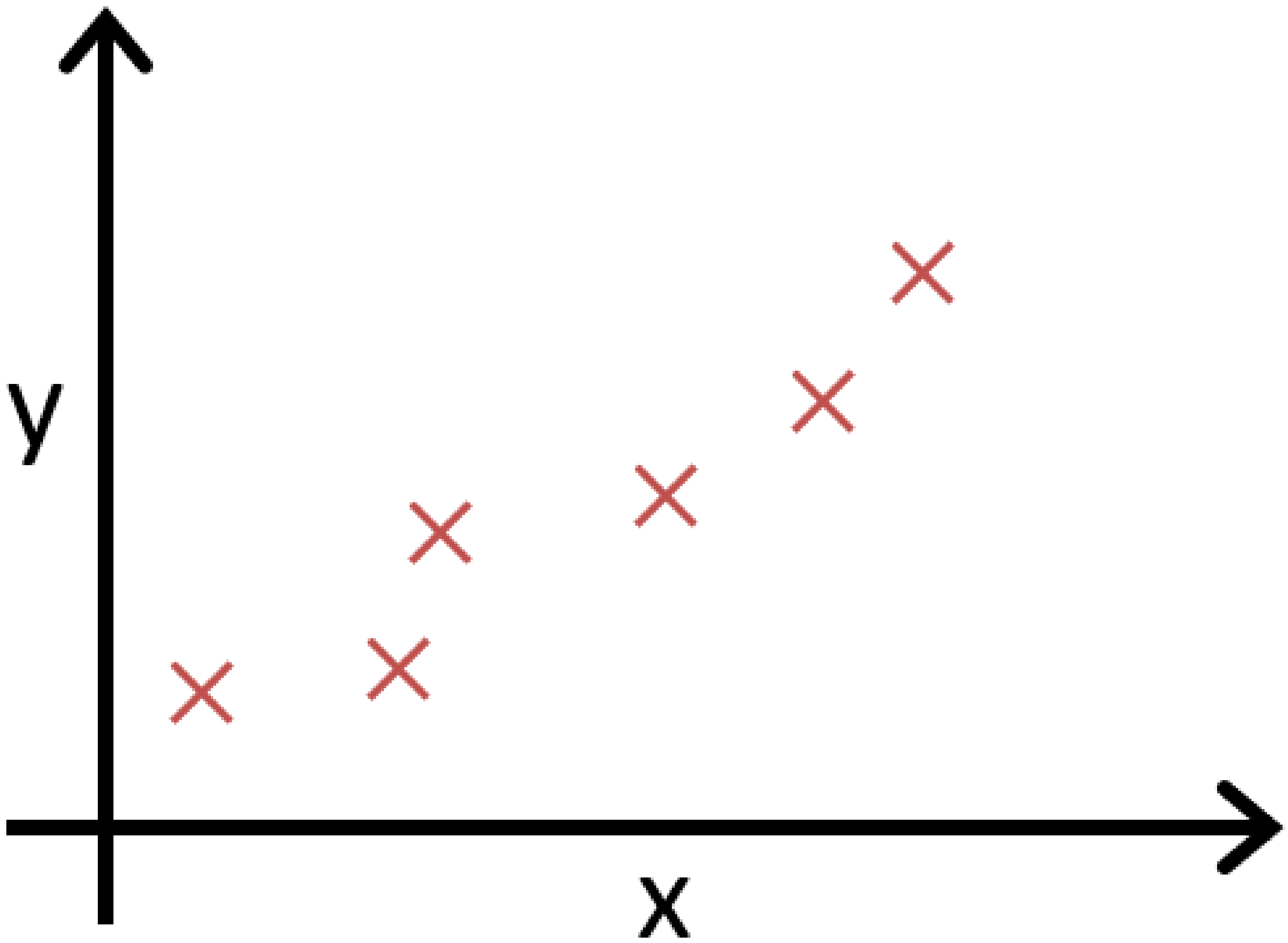


$\theta_0 = 1.5$
$\theta_1 = 0$

$\theta_0 = 0$
$\theta_1 = 0.5$

$\theta_0 = 1$
$\theta_1 = 0.5$

求解方程系数



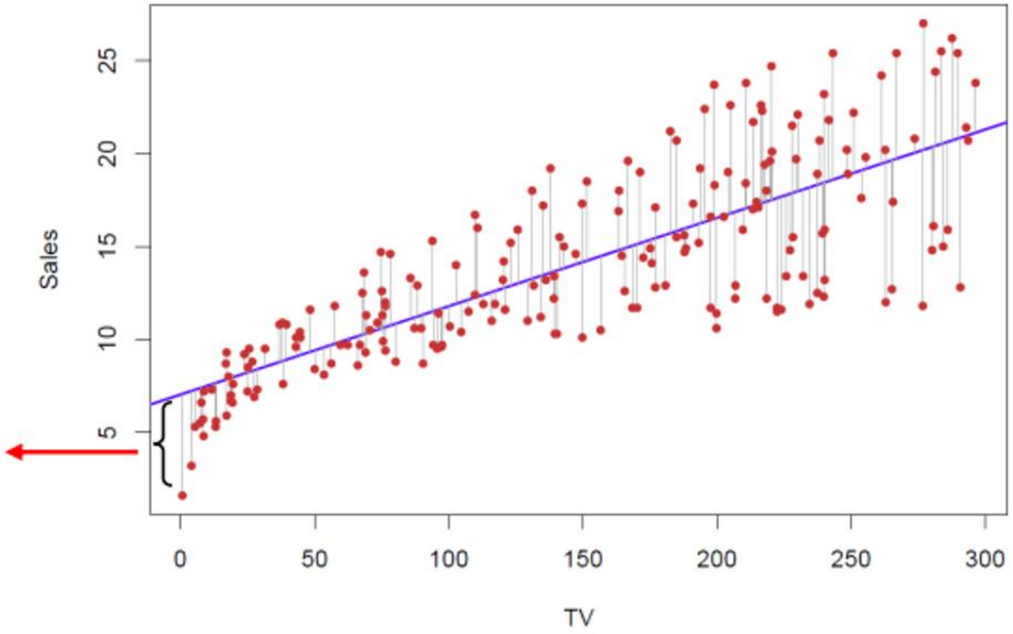
代价函数
Cost Function

代价函数(Cost Function)

- 最小二乘法
- 真实值y，预测值$h_\theta(x)$，则误差平方为$(y - h_\theta(x))^2$
- 找到合适的参数，使得误差平方和：

$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(y^i - h_\theta\left(x^i\right)\right)^2 \text{最小}$$

$y^i - h_\theta\left(x^i\right)$



代价函数(Cost Function)

Hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

Parameters:

$$\theta_0, \theta_1$$

Cost Function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

Simplified

$$h_\theta(x) = \theta_1 x$$

$$\theta_1$$

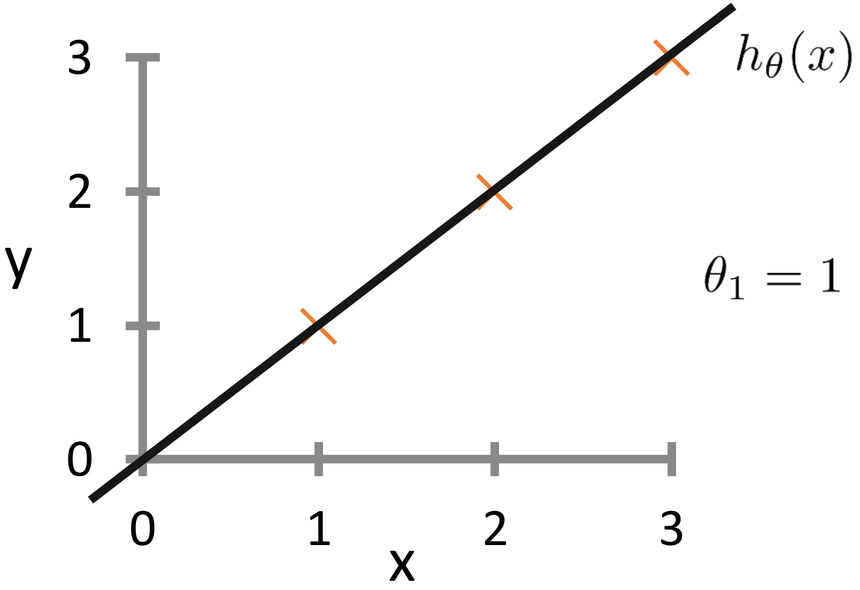
$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$$\underset{\theta_1}{\text{minimize}}\ J(\theta_1)$$

代价函数(Cost Function)

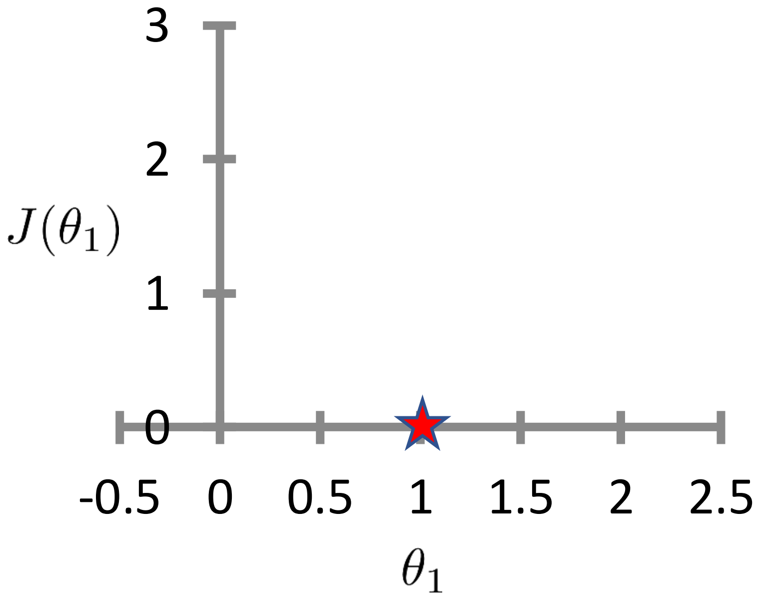
$h_\theta(x)$

(for fixed θ_1, this is a function of x)



$J(\theta_1)$

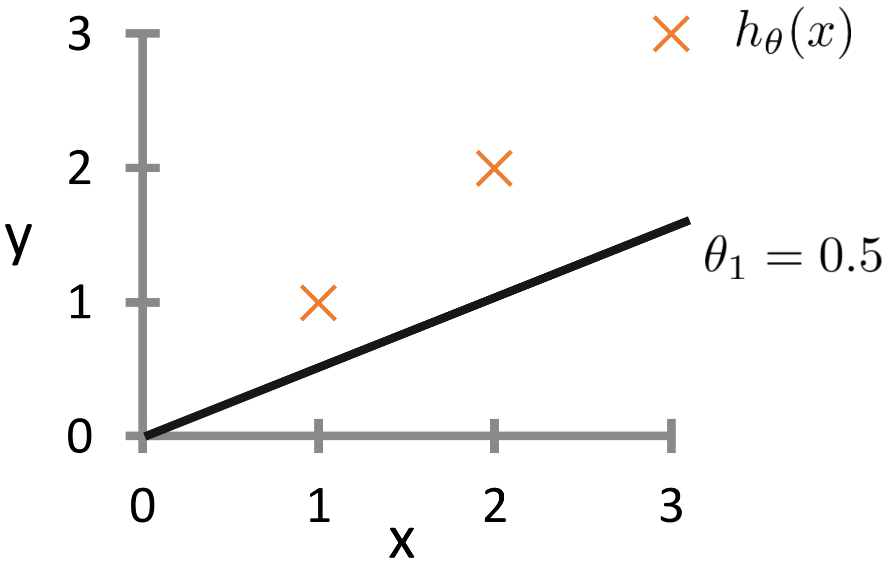
(function of the parameter θ_1)



代价函数(Cost Function)

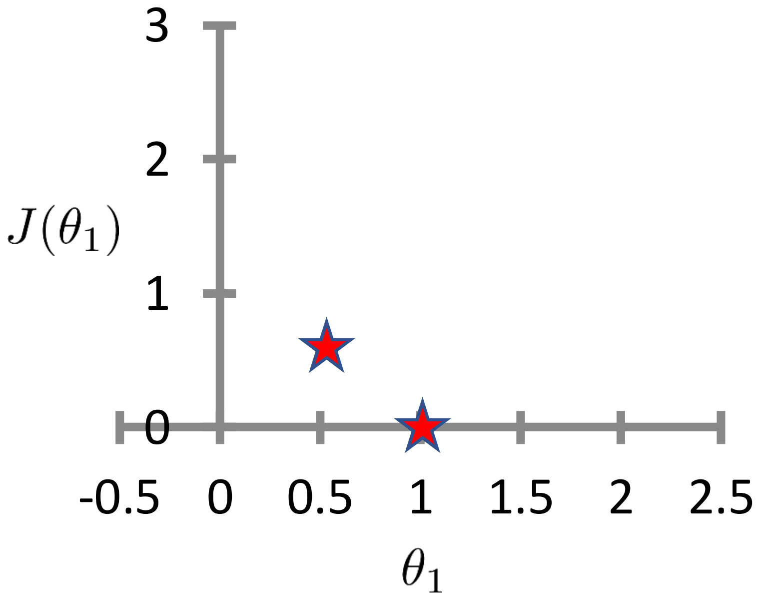
$h_\theta(x)$

(for fixed θ_1, this is a function of x)



$J(\theta_1)$

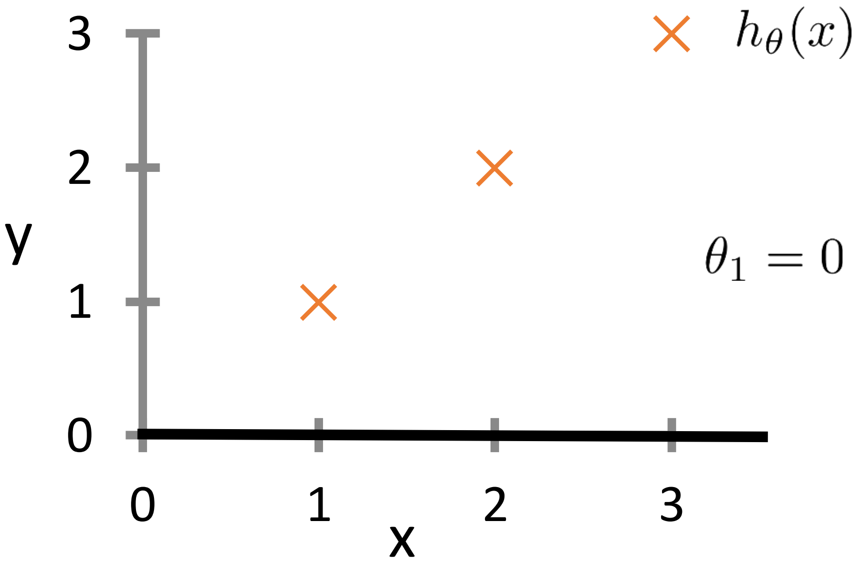
(function of the parameter θ_1)



代价函数(Cost Function)

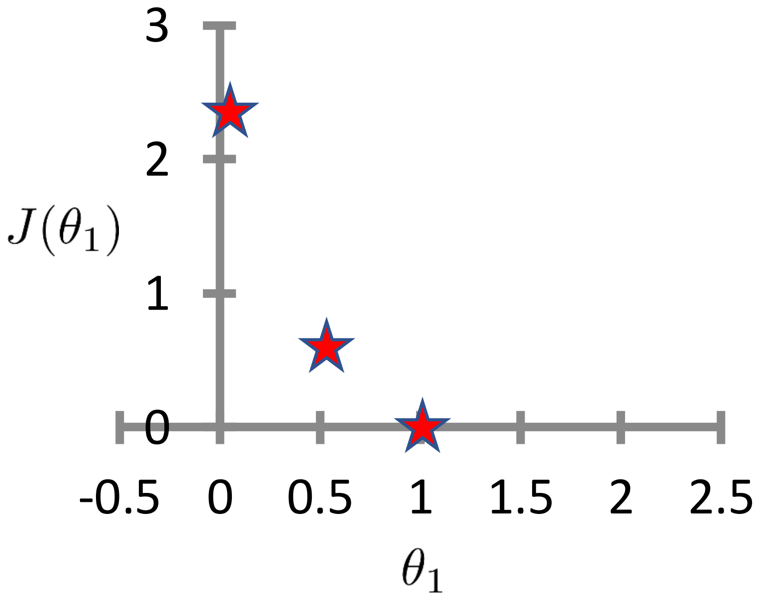
$h_\theta(x)$

(for fixed θ_1, this is a function of x)



$J(\theta_1)$

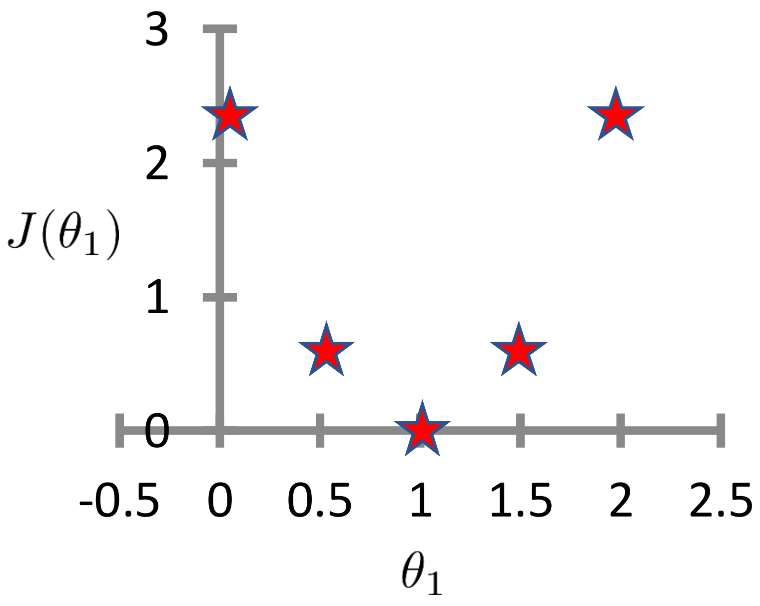
(function of the parameter θ_1)



代价函数(Cost Function)

$J(\theta_1)$

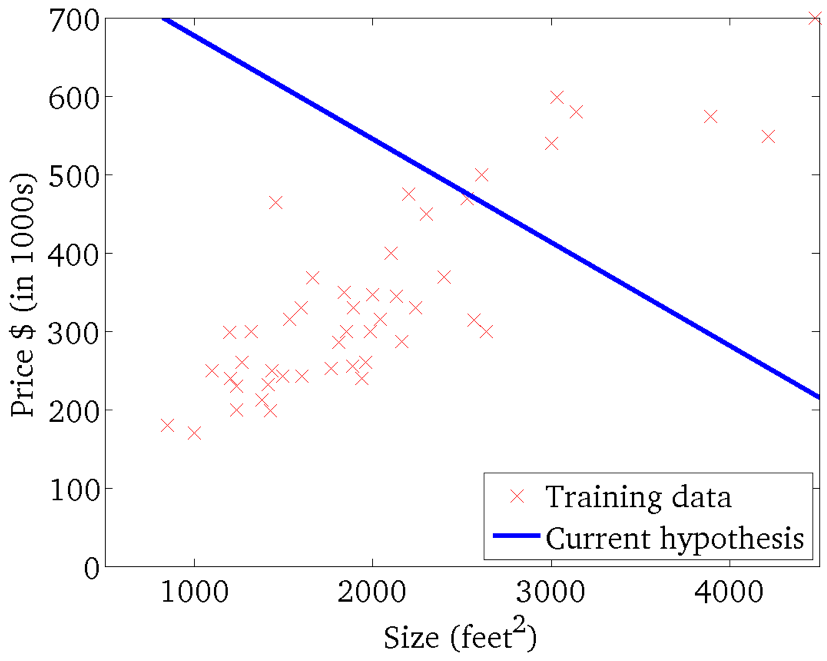
(function of the parameter θ_1)



代价函数(Cost Function)

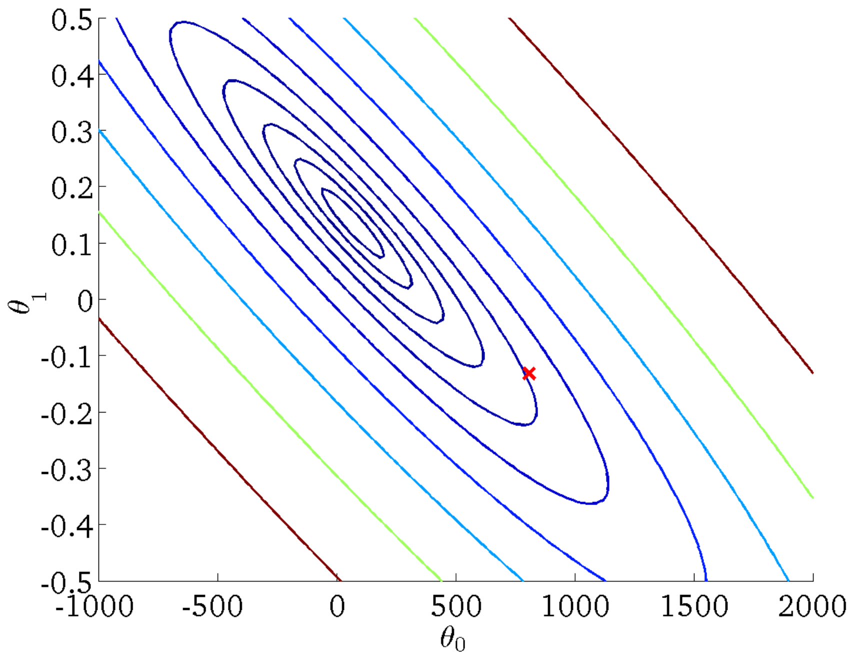
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



$$J(\theta_0, \theta_1)$$

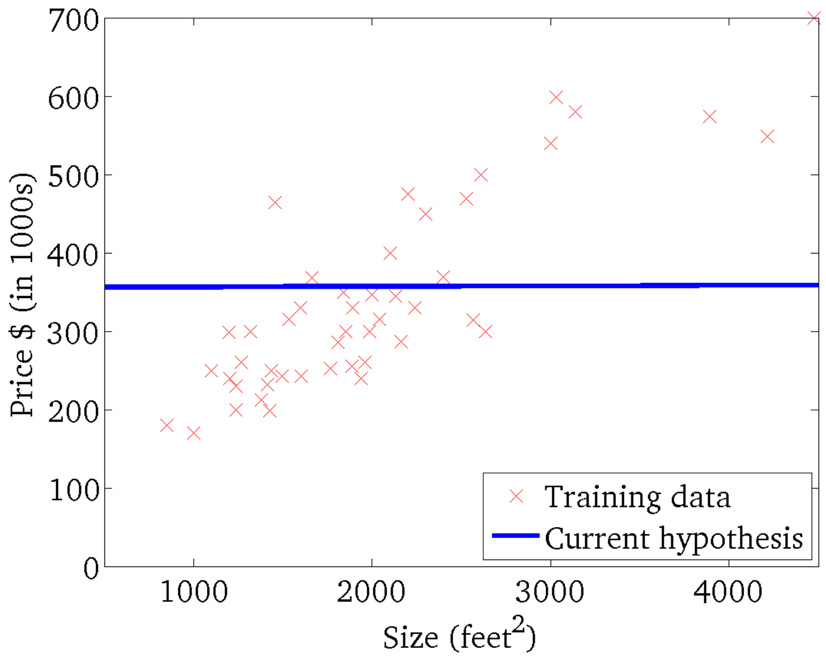
(function of the parameters θ_0, θ_1)



代价函数(Cost Function)

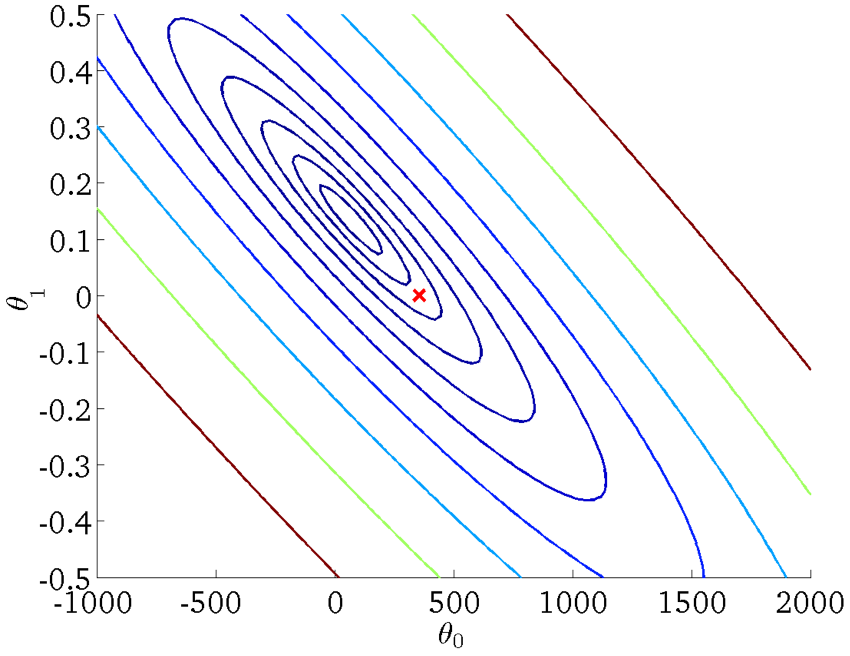
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

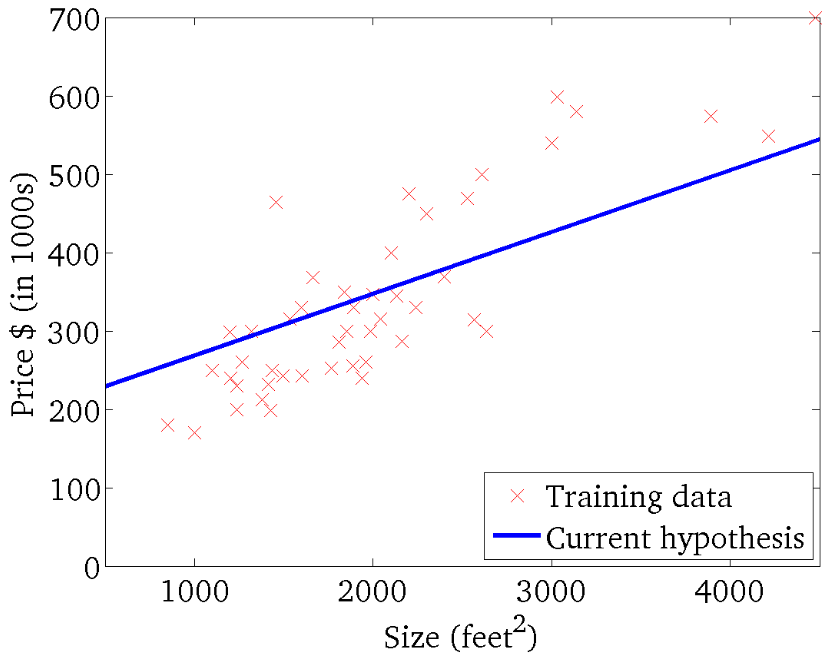
(function of the parameters θ_0, θ_1)



代价函数(Cost Function)

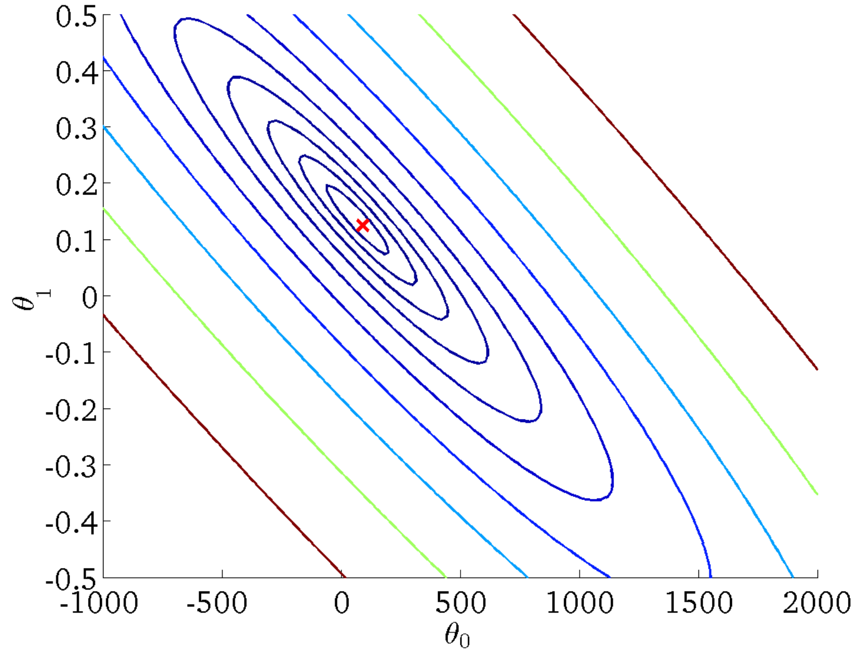
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

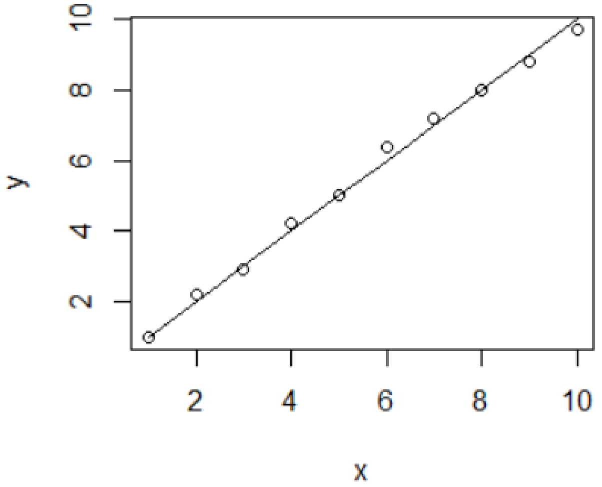
(function of the parameters θ_0, θ_1)

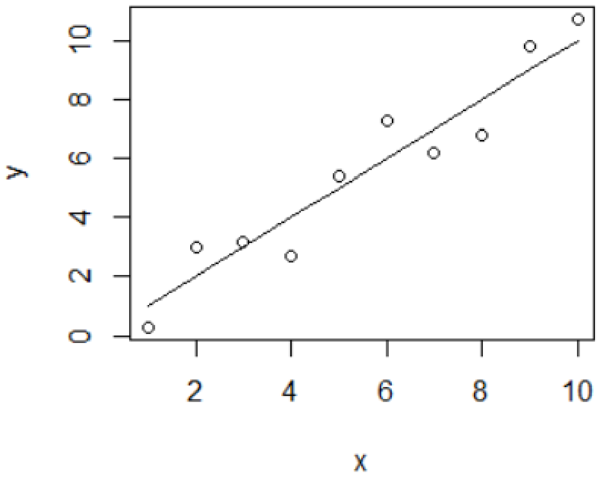


相关系数

我们使用相关系数去衡量线性相关性的强弱

$$r_{XY} = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2 \sum (Y_i - \bar{Y})^2}}$$





通过计算，左图的相关系数为0.993，右图的相关系数为0.957

决定系数

相关系数R^2(coefficient of determination)是用来描述两个变量之间的线性关系的，但决定系数的适用范围更广，可以用于描述非线性或者有两个及两个以上自变量的相关关系。它可以用来评价模型的效果。

总平方和(SST)：$\sum_{i=1}^{n}(y_i-\bar{y})^2$
回归平方和(SSR)：$\sum_{i=1}^{n}(\hat{y}-\bar{y})^2$
残差平方和(SSE)：$\sum_{i=1}^{n}(y_i-\hat{y})^2$
它们三者的关系是：SST = SSR +SSE

决定系数：$R^2=\frac{SSR}{SST}=1-\frac{SSE}{SST}$

梯度下降法
Gradient Descent

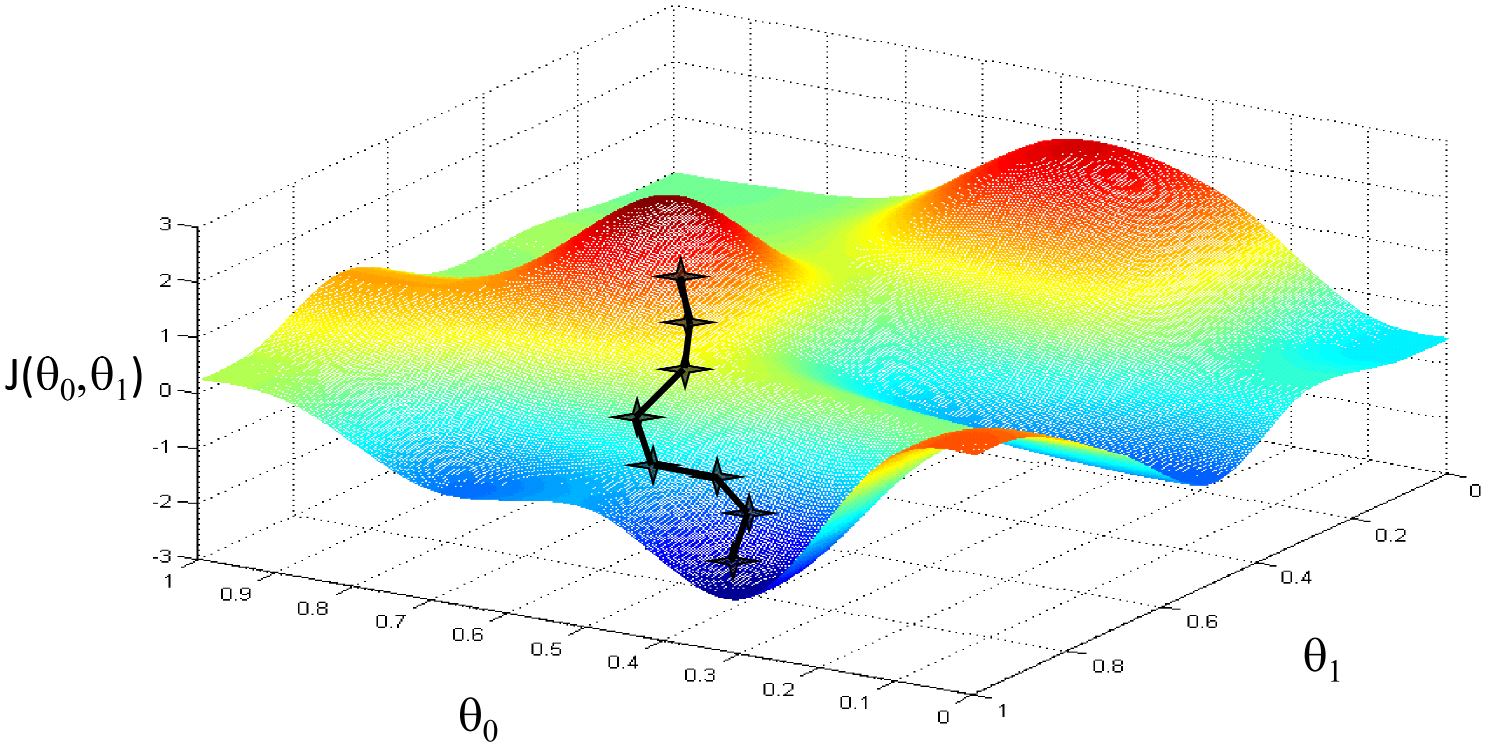
梯度下降法

Have some function $J(\theta_0, \theta_1)$

Want $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

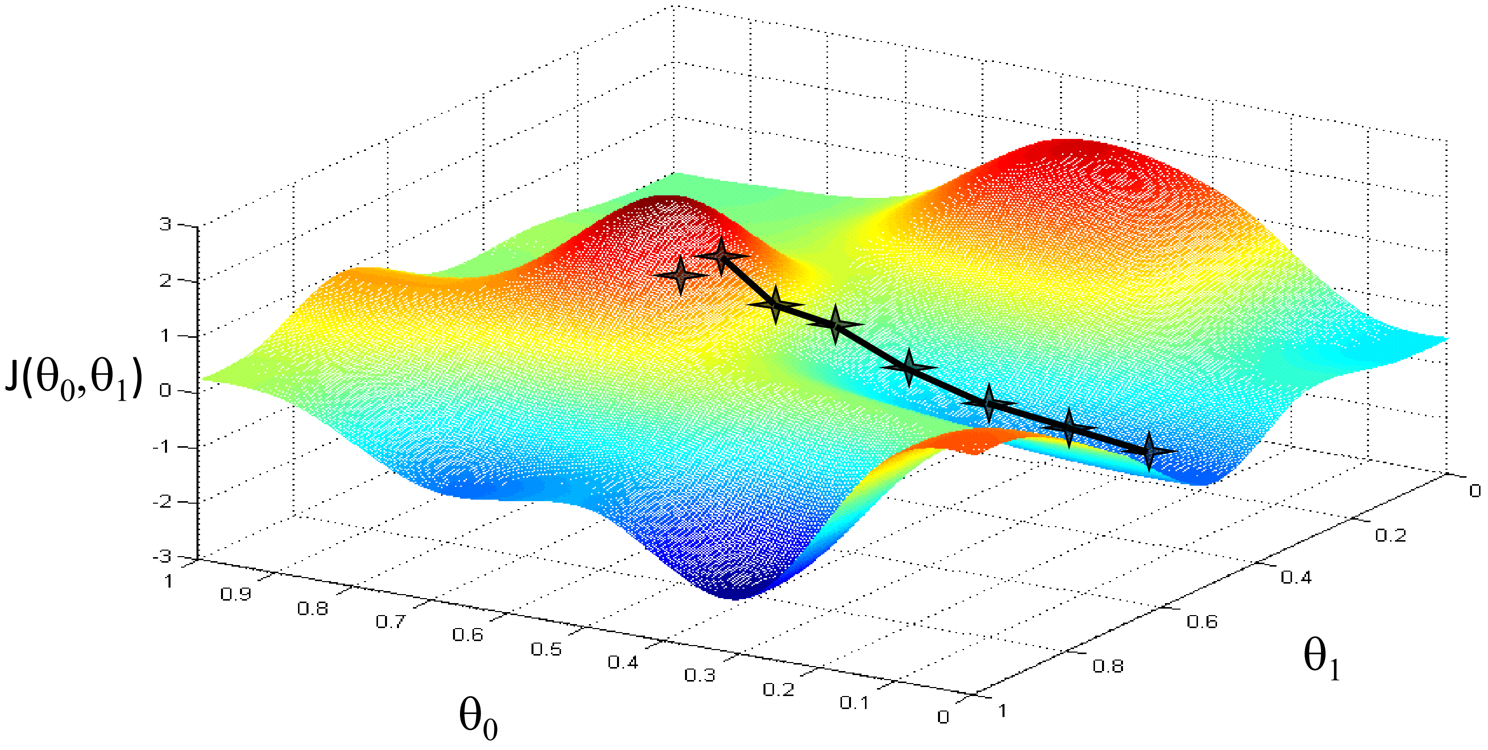
- 初始化θ_0, θ_1
- 不断改变θ_0, θ_1，直到$J(\theta_0, \theta_1)$到达一个全局最小值，或局部极小值。

梯度下降法



$J(\theta_0,\theta_1)$
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1

梯度下降法



$J(\theta_0,\theta_1)$
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1

梯度下降法

$$\text{repeat until convergence } \{$$
$$\quad \underset{\substack{\downarrow \\ \text{学习率}}}{\theta_j := \theta_j - \alpha}\frac{\partial}{\partial\theta_j}J(\theta_0,\theta_1) \quad (\text{for } j=0 \text{ and } j=1)$$
$$\}$$

正确做法：同步更新

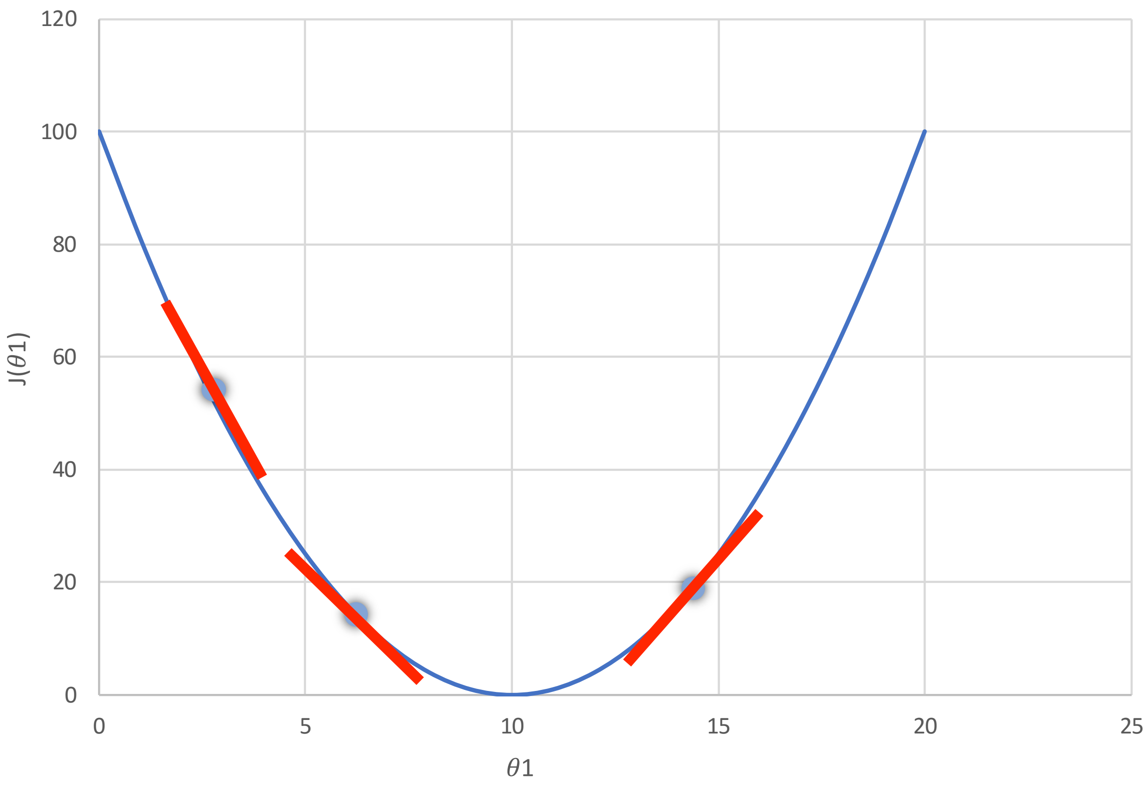
$$\text{temp0} := \theta_0 - \alpha\frac{\partial}{\partial\theta_0}J(\theta_0,\theta_1)$$
$$\text{temp1} := \theta_1 - \alpha\frac{\partial}{\partial\theta_1}J(\theta_0,\theta_1)$$
$$\theta_0 := \text{temp0}$$
$$\theta_1 := \text{temp1}$$

不正确做法

$$\text{temp0} := \theta_0 - \alpha\frac{\partial}{\partial\theta_0}J(\theta_0,\theta_1)$$
$$\theta_0 := \text{temp0}$$
$$\text{temp1} := \theta_1 - \alpha\frac{\partial}{\partial\theta_1}J(\theta_0,\theta_1)$$
$$\theta_1 := \text{temp1}$$

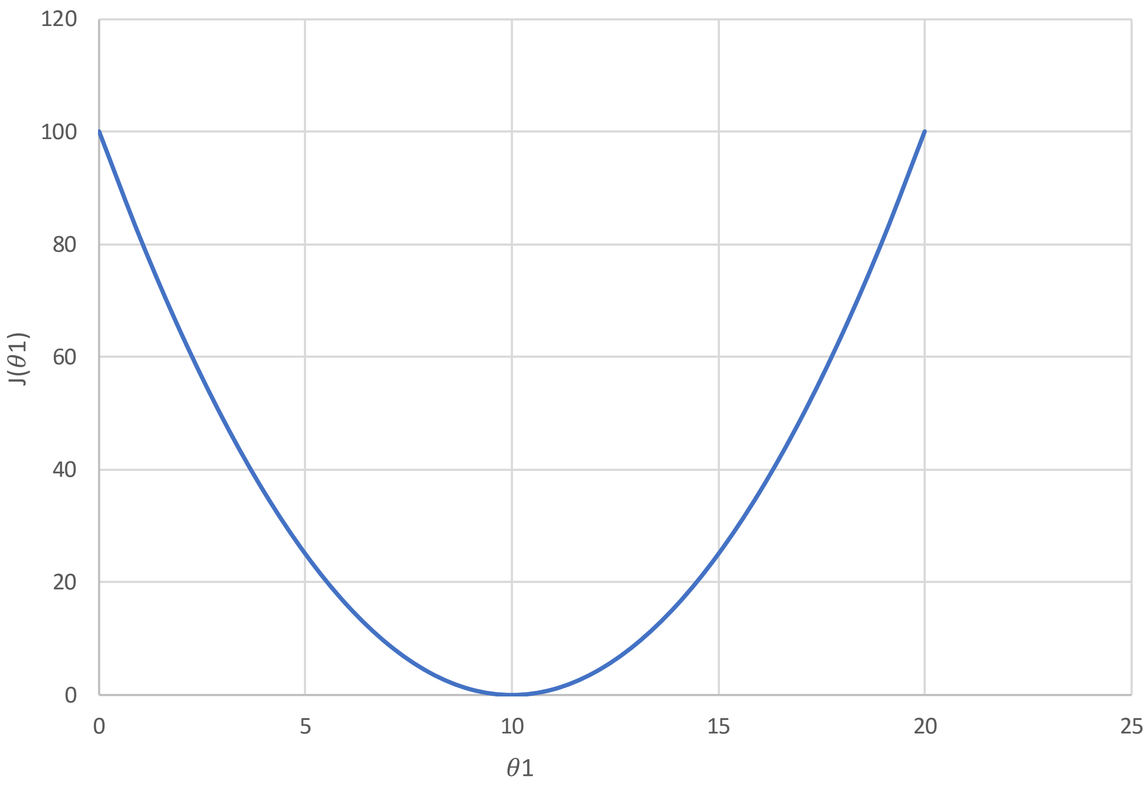
梯度下降法

$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$



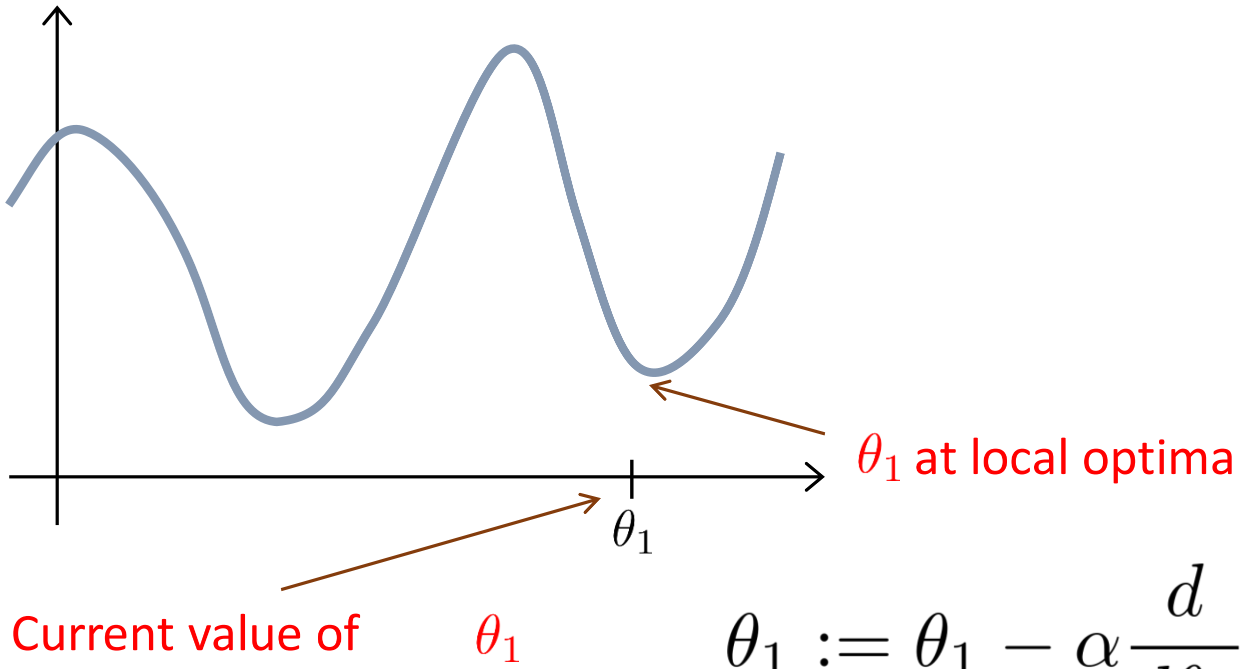
梯度下降法

学习率不能太小，也不能太大，可以多尝试一些值
0.1,0.03,0.01,0.003,0.001,0.0003,0.0001…



梯度下降法

有可能会陷入局部极小值



$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$

用梯度下降法来求解线性回归

梯度下降法

$$
\begin{aligned}
&\text{repeat until convergence } \{ \\
&\quad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \\
&\qquad (\text{for } j = 1 \text{ and } j = 0) \\
&\}
\end{aligned}
$$

线性回归的模型和代价函数

$$h_\theta(x) = \theta_0 + \theta_1 x$$

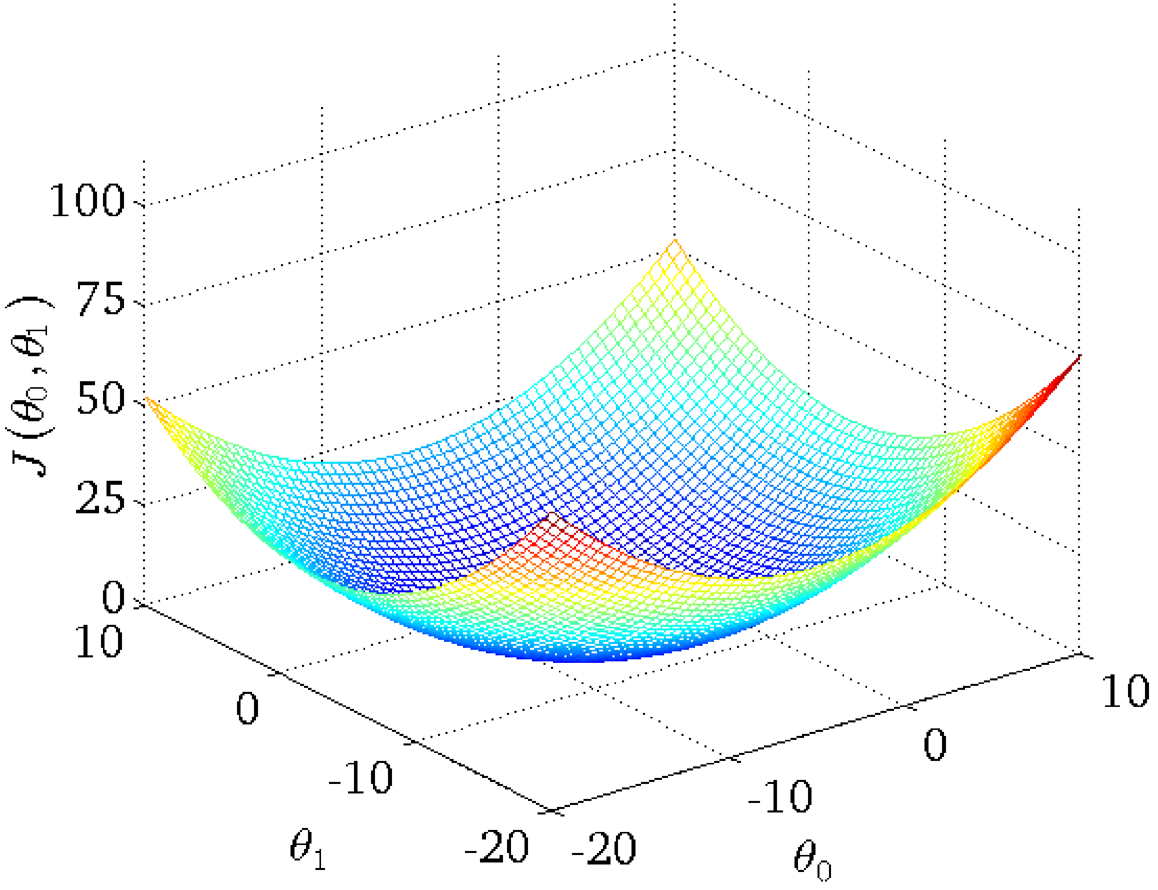
$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

用梯度下降法来求解线性回归

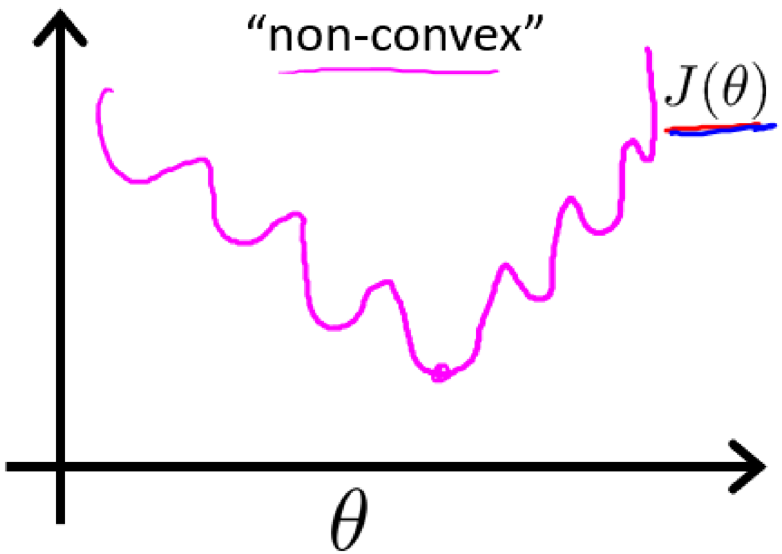
$$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) = \begin{cases} j = 0 : \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \\ j = 1 : \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \cdot x^{(i)} \end{cases}$$

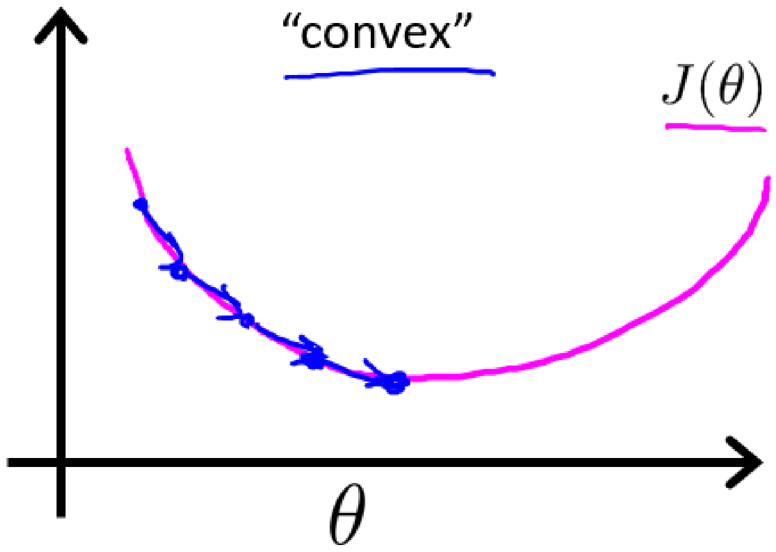
$$\begin{aligned} &\text{repeat until convergence } \{ \\ &\quad \theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \\ &\quad \theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \cdot x^{(i)} \\ &\} \end{aligned}$$

线性回归的代价函数是凸函数



非凸函数和凸函数

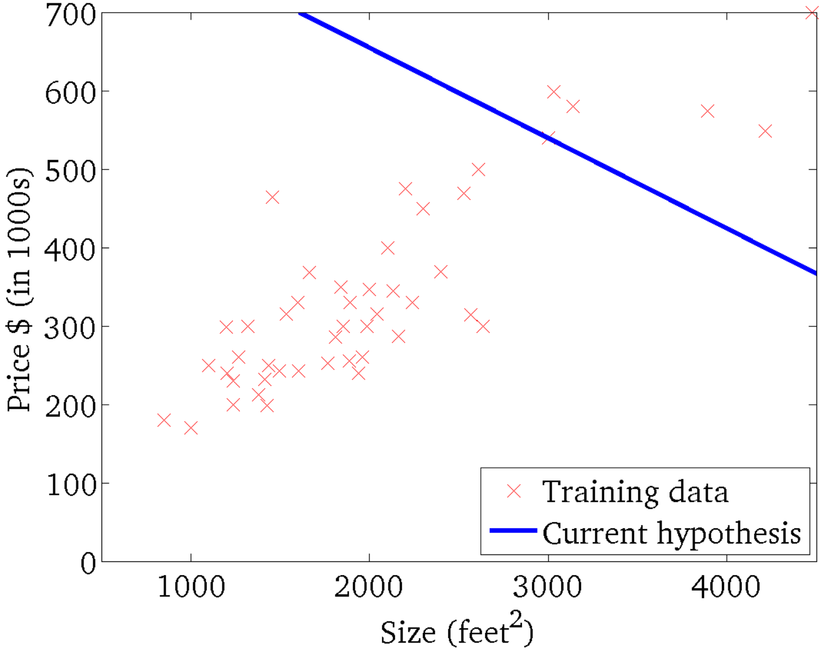




梯度下降法优化过程

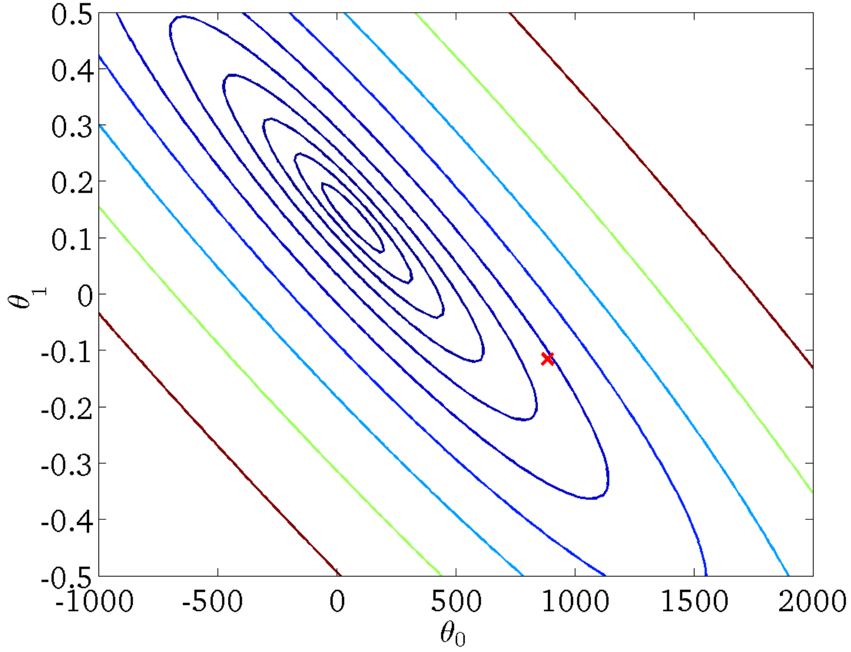
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

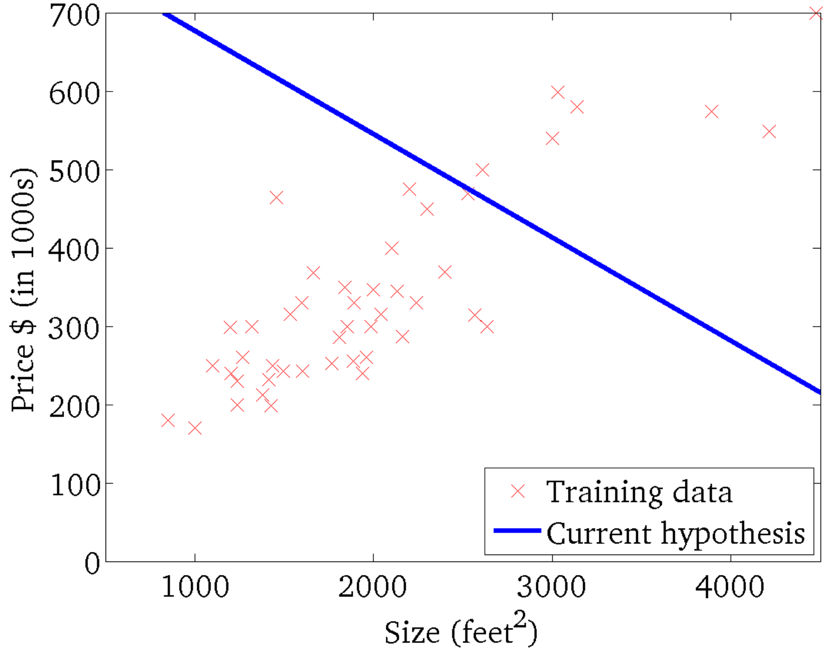
(function of the parameters θ_0, θ_1)



梯度下降法优化过程

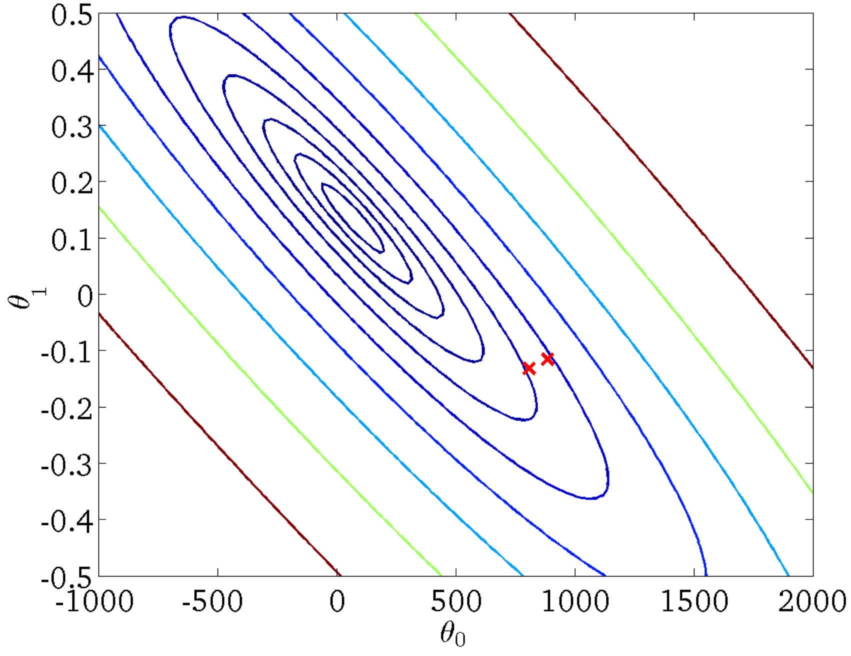
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



$$J(\theta_0, \theta_1)$$

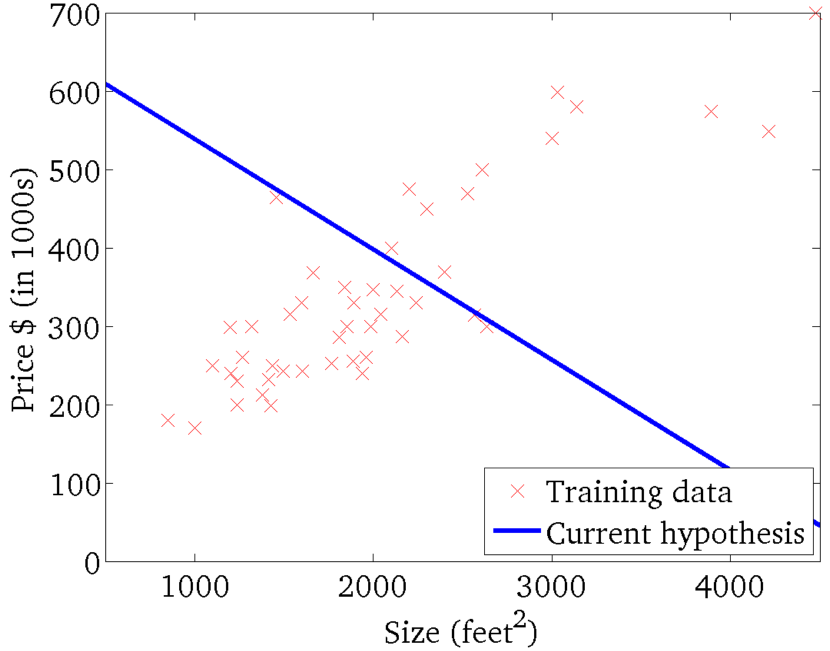
(function of the parameters θ_0, θ_1)



梯度下降法优化过程

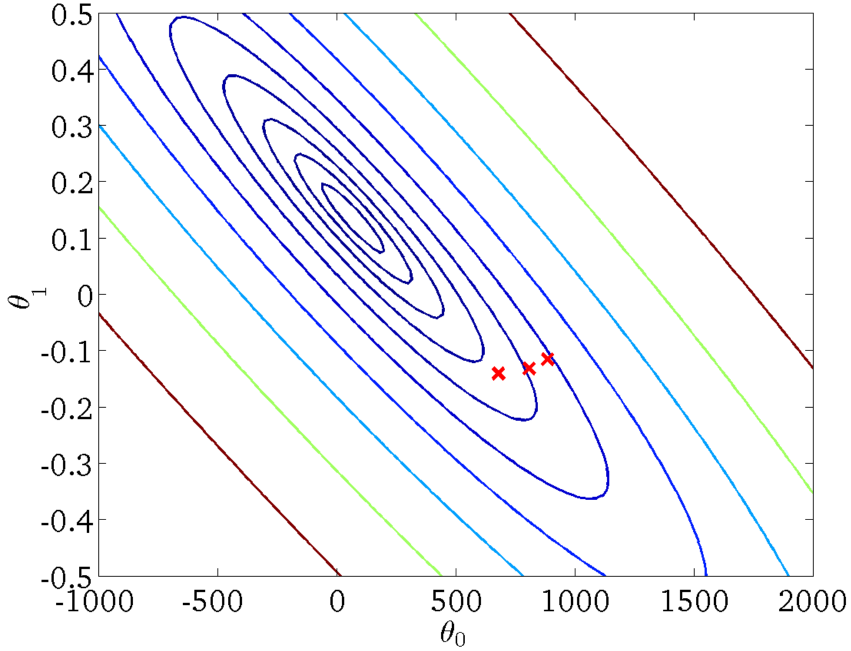
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

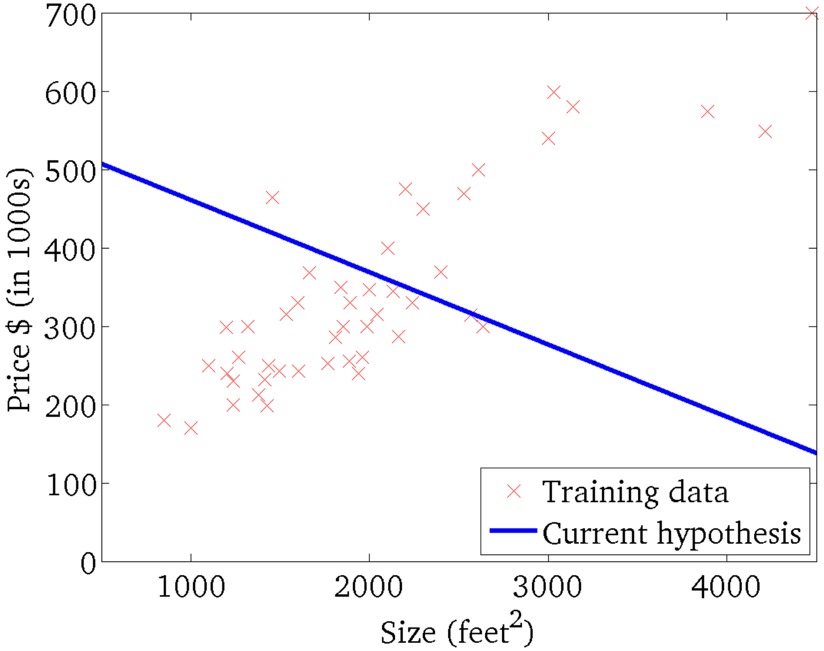
(function of the parameters θ_0, θ_1)



梯度下降法优化过程

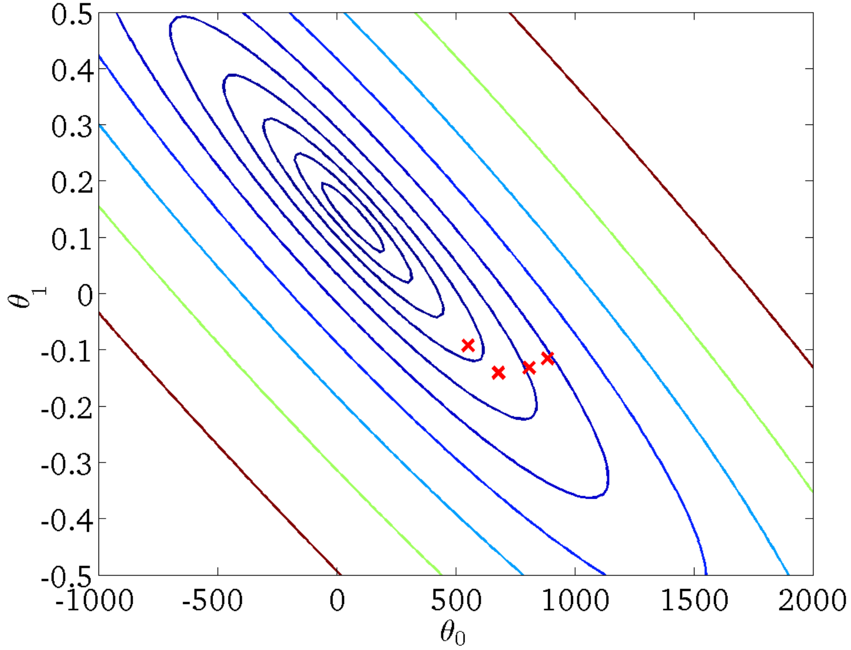
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

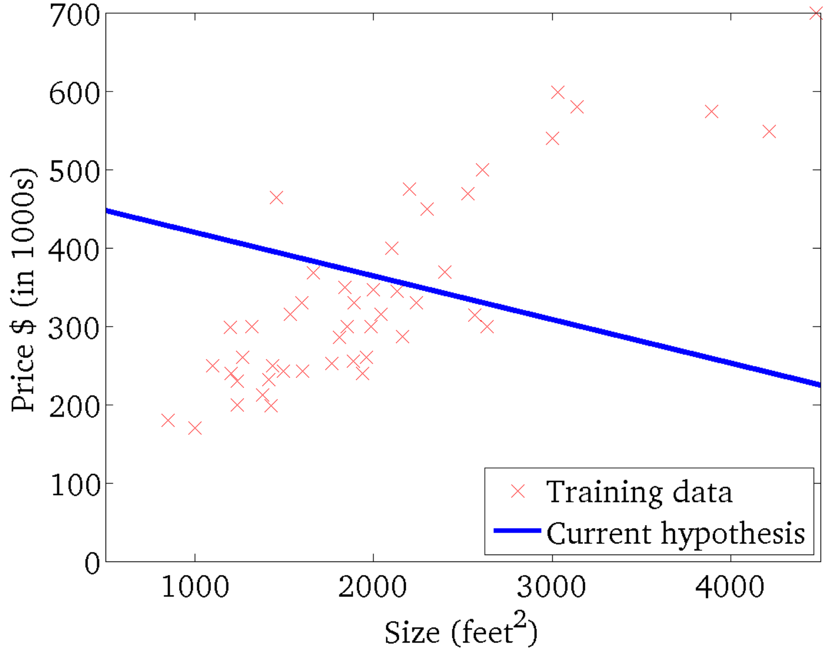
(function of the parameters θ_0, θ_1)



梯度下降法优化过程

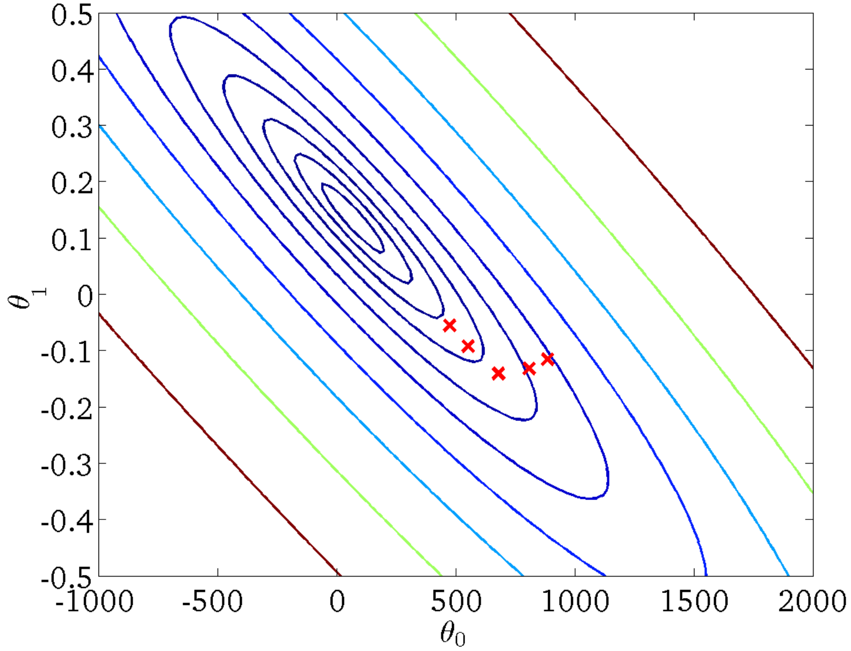
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

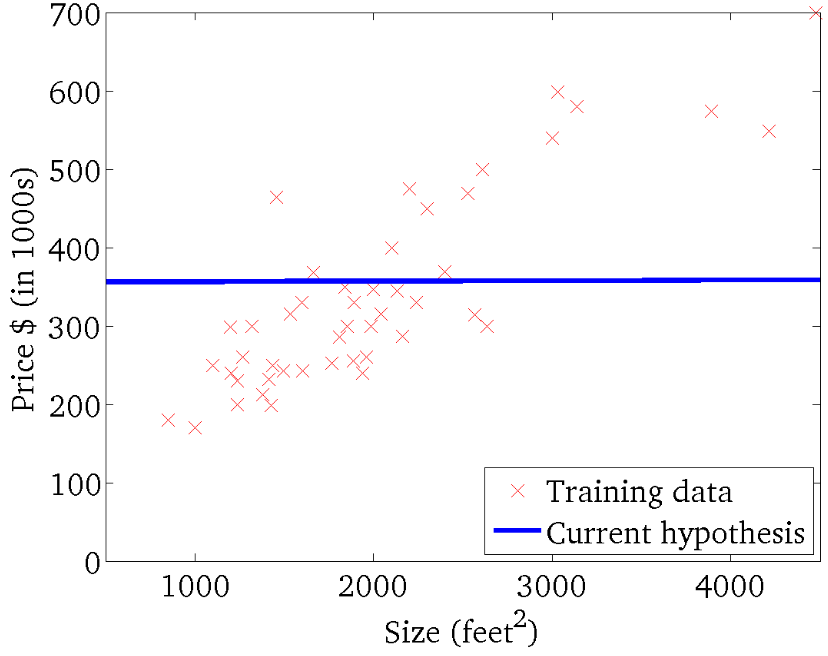
(function of the parameters θ_0, θ_1)



梯度下降法优化过程

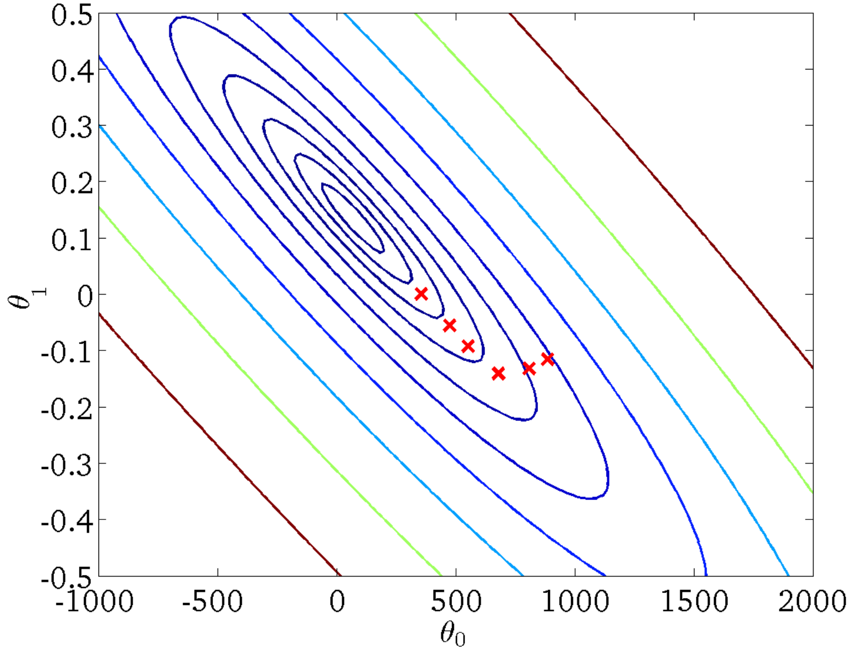
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



$$J(\theta_0, \theta_1)$$

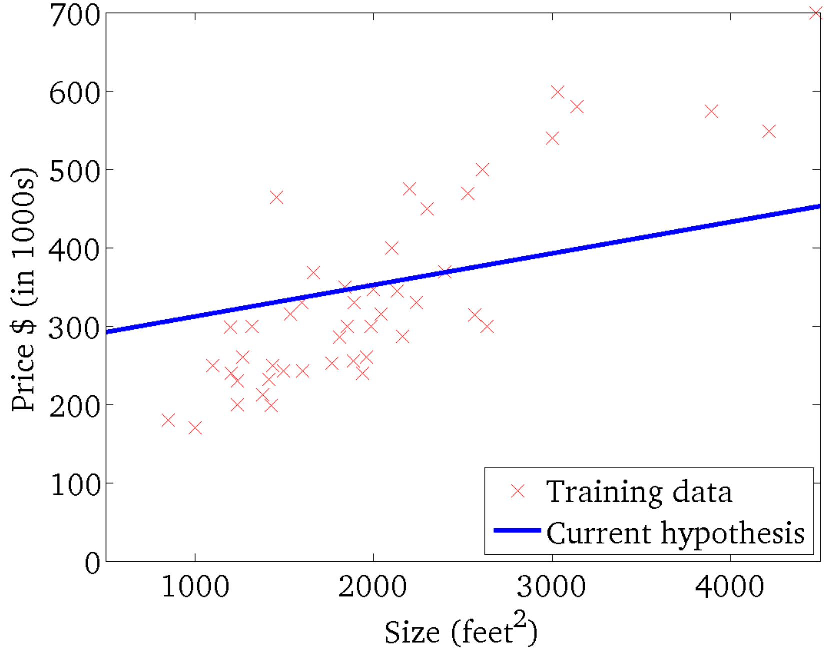
(function of the parameters θ_0, θ_1)



梯度下降法优化过程

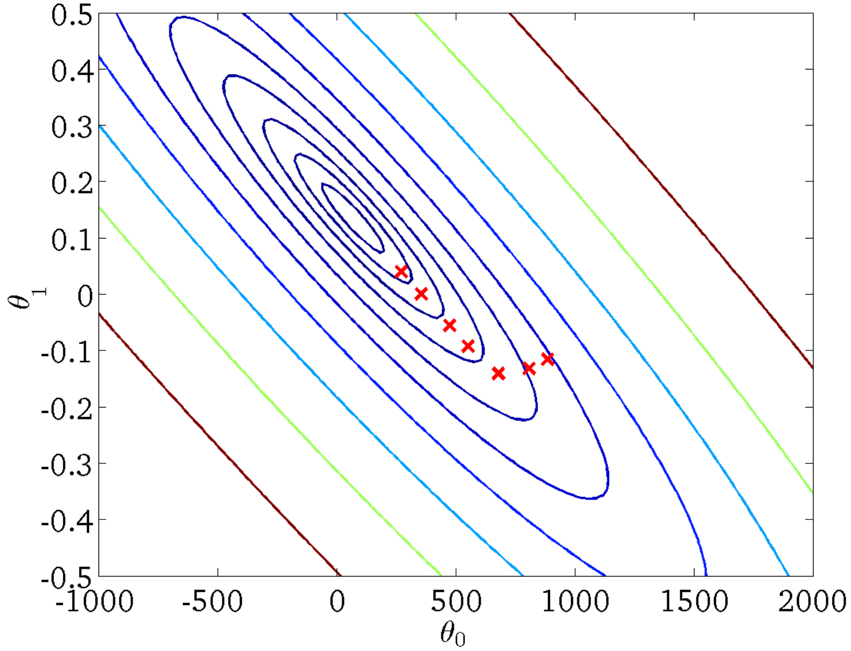
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

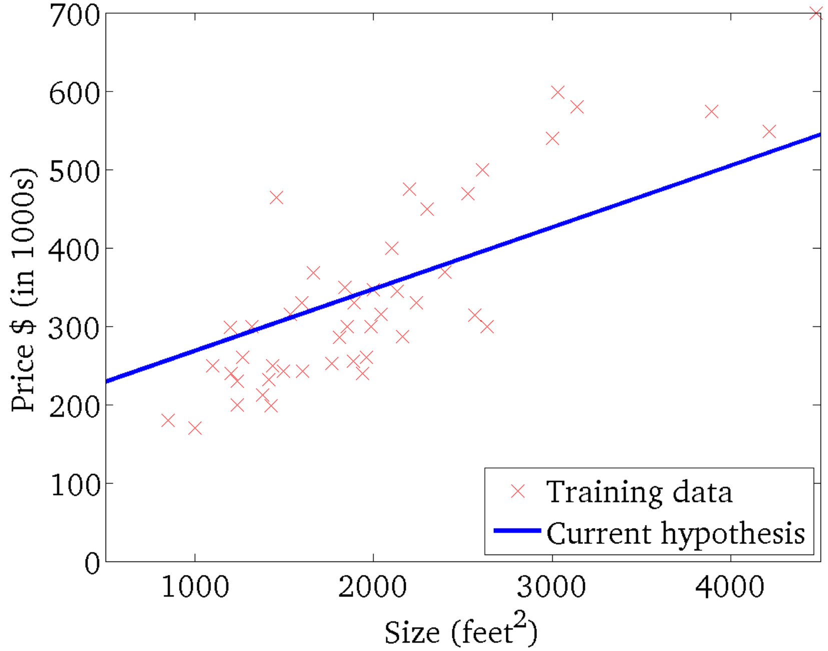
(function of the parameters θ_0, θ_1)



梯度下降法优化过程

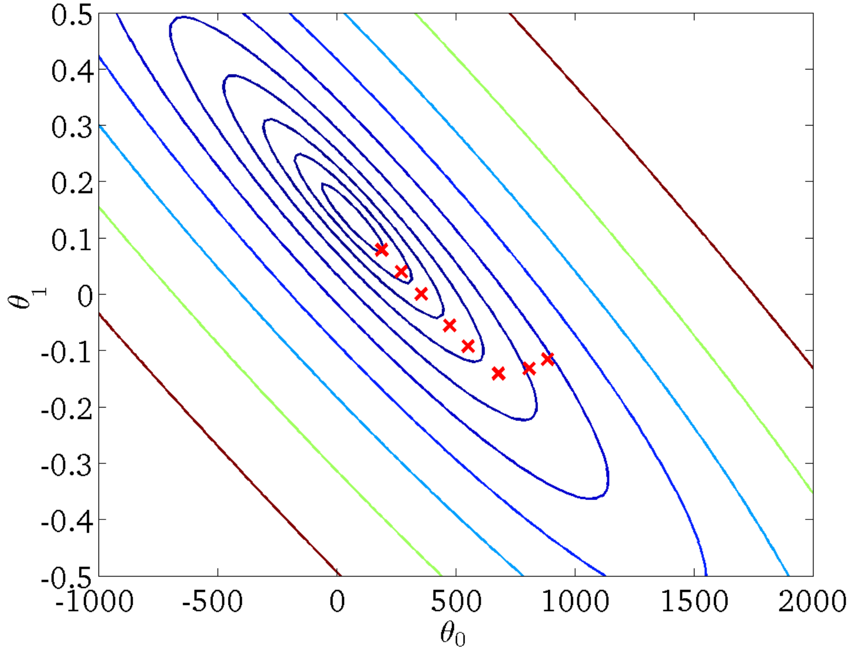
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

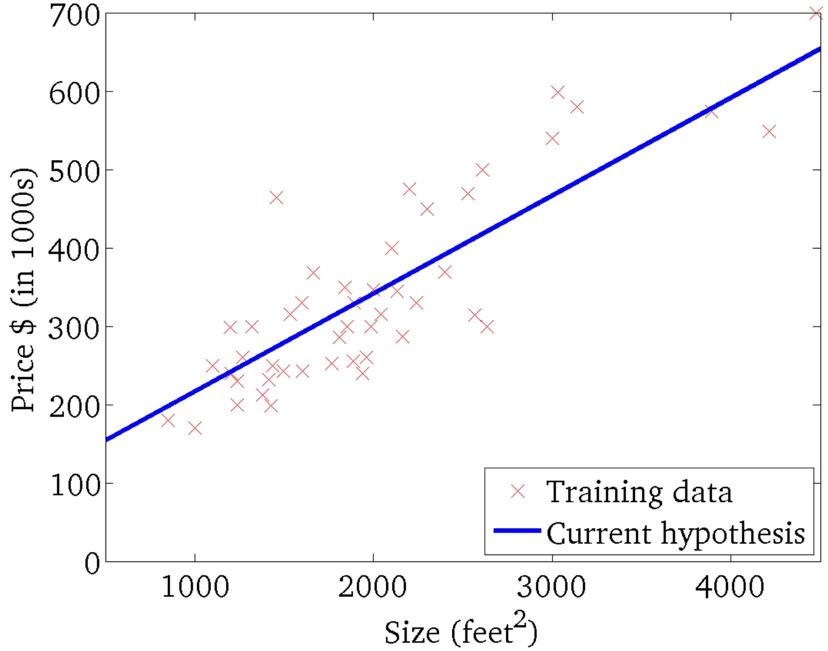
(function of the parameters θ_0, θ_1)



梯度下降法优化过程

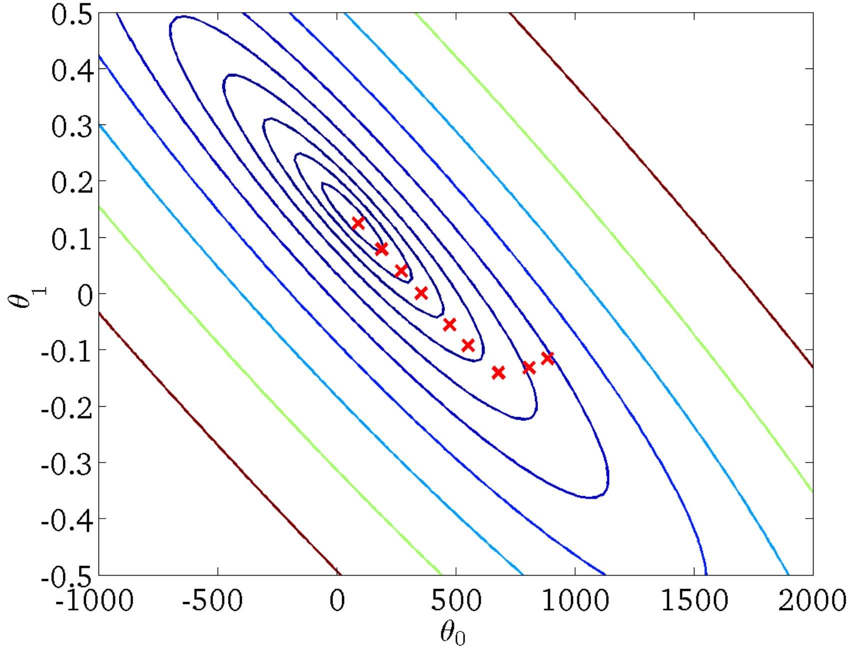
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



梯度下降法-一元线性回归

sklearn-一元线性回归

矩阵运算

矩阵

3行2列的矩阵：

$$A = \begin{bmatrix} 2 & 3 \\ -1 & 6 \\ 4 & 0 \end{bmatrix}$$

$A_{11}=6$,A[1,1]=6
$A_{20}=4$,A[2,0]=4
A[0]=[2,3]

2行3列的矩阵：

$$B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

$B_{02}=3$,B[0,2]=3
$B_{11}=5$,B[1,1]=5
B[0]=[1,2,3]

矩阵运算

正确的按位加减乘除，
两个矩阵的形状要一致：

$$\begin{bmatrix} 2 & 3 \\ -1 & 6 \\ 4 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} = \begin{bmatrix} 3 & 5 \\ 2 & 10 \\ 9 & 6 \end{bmatrix}$$

形状不一致的两个矩阵
不能按位进行加减乘
除：

$$\begin{bmatrix} 2 & 3 \\ -1 & 6 \\ 4 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

矩阵运算

$$3 * \begin{bmatrix} 2 & 3 \\ -1 & 6 \\ 4 & 0 \end{bmatrix} = \begin{bmatrix} 6 & 9 \\ -3 & 18 \\ 12 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 3 \\ -1 & 6 \\ 4 & 0 \end{bmatrix} + 3 = \begin{bmatrix} 5 & 6 \\ 2 & 9 \\ 7 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 3 \\ -1 & 6 \\ 4 & 0 \end{bmatrix} / 3 = \begin{bmatrix} 2/3 & 1 \\ -1/3 & 2 \\ 4/3 & 0 \end{bmatrix}$$

矩阵乘法

n行m列的矩阵乘以m行n列的矩阵得到n行n列的矩阵：

$$\begin{bmatrix} 2 & 3 \\ -1 & 6 \\ 4 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} = \begin{bmatrix} 2\times 1+3\times 4 & 2\times 2+3\times 5 & 2\times 3+3\times 6 \\ -1\times 1+6\times 4 & -1\times 2+6\times 5 & -1\times 3+6\times 6 \\ 4\times 1+0\times 4 & 4\times 2+0\times 5 & 4\times 3+0\times 6 \end{bmatrix}$$

$$= \begin{bmatrix} 14 & 19 & 24 \\ 23 & 28 & 33 \\ 4 & 8 & 12 \end{bmatrix}$$

单位矩阵

单位矩阵：

$$I_{3\times 3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$A \cdot I = I \cdot A = A$$

转置矩阵

$$A = \begin{bmatrix} 2 & 3 \\ -1 & 6 \\ 4 & 0 \end{bmatrix}$$

$$A^T = \begin{bmatrix} 2 & -1 & 4 \\ 3 & 6 & 0 \end{bmatrix}$$

$$A_{ij} = A^T_{ji}$$

逆矩阵

逆矩阵特点：

$$AA^{-1} = A^{-1}A = I$$

举例：

$$\begin{bmatrix} 3 & 4 \\ 2 & 16 \end{bmatrix} \begin{bmatrix} 0.4 & -0.1 \\ -0.05 & 0.075 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

多元线性回归

单特征

Size ($feet^2$)	Price ($1000)
2104	460
1416	232
1534	315
852	178
…	…

$$h_\theta(x) = \theta_0 + \theta_1 x$$

多特征

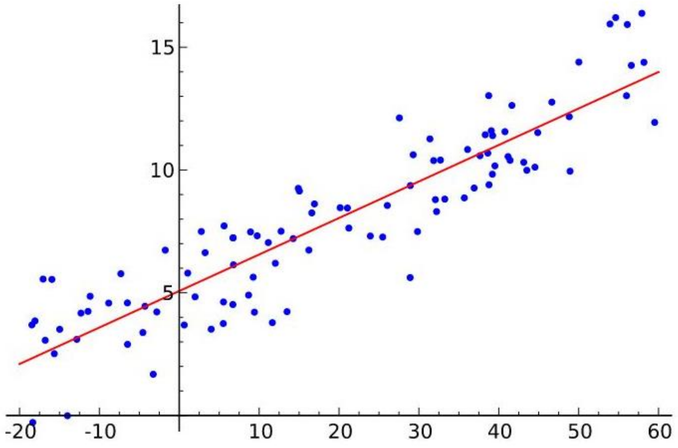
Size (feet^2)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

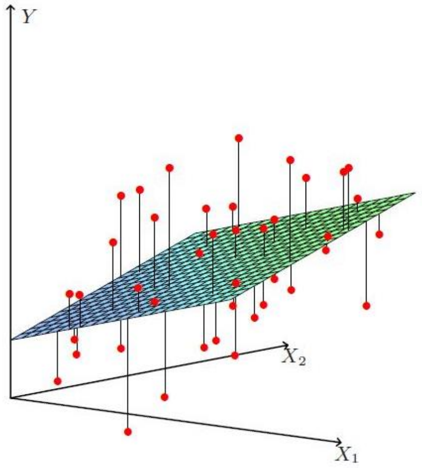
多元线性回归模型

当Y值的影响因素不是唯一时，采用多元线性回归模型

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$



一元线性回归



二元线性回归

多元线性回归

Hypothesis: $h_\theta(x) = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

Parameters: $\theta_0, \theta_1, \ldots, \theta_n$

Cost function:

$$J(\theta_0, \theta_1, \ldots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Gradient descent:

Repeat $\{$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \ldots, \theta_n)$$

$\}$ (simultaneously update for every $j = 0, \ldots, n$)

多元线性回归

Gradient Descent

Previously (n=1):

Repeat $\{$

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

$$\theta_1 := \theta_1 - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x^{(i)}$$

(simultaneously update θ_0, θ_1)

$\}$

New algorithm $(n \geq 1)$

Repeat $\{$

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

(simultaneously update θ_j for $j = 0, \ldots, n$)

$\}$

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

$$\theta_1 := \theta_1 - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_1^{(i)}$$

$$\theta_2 := \theta_2 - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_2^{(i)}$$

例子

一家快递公司送货：X1： 运输里程 X2：
运输次数 Y：总运输时间

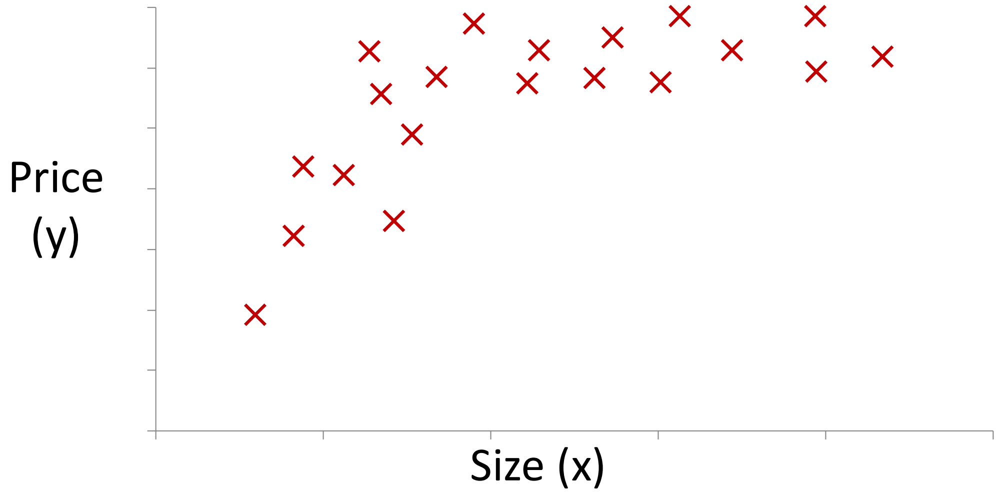
Driving Assignment	X1=Miles Traveled	X2=Number of Deliveries	Y= Travel Time (Hours)
1	100	4	9.3
2	50	3	4.8
3	100	4	8.9
4	100	2	6.5
5	50	2	4.2
6	80	2	6.2
7	75	3	7.4
8	65	4	6.0
9	90	3	7.6
10	90	2	6.1

梯度下降法-多元线性回归

sklearn-多元线性回归

多项式回归

多项式回归

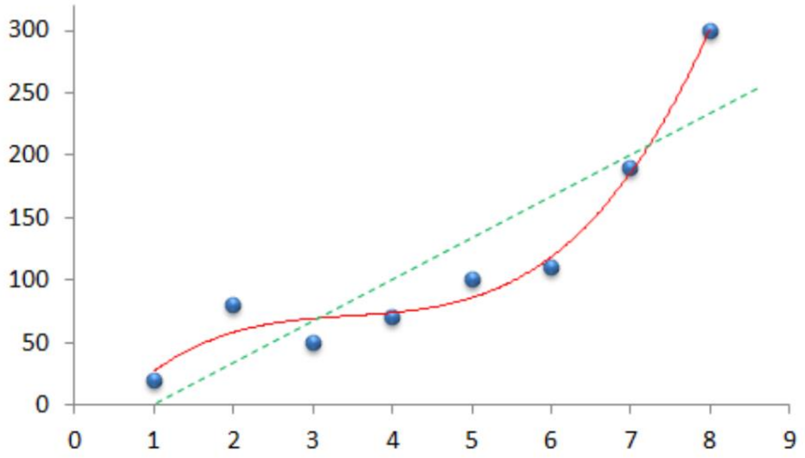


$$\theta_0 + \theta_1 x + \theta_2 x^2$$

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

多项式回归

假如我们不是要找直（或者超平面），而是一个需要找到一个用多项式所表示的曲线（或者超曲面），例如二次曲线：$y=at^2+bt+c$

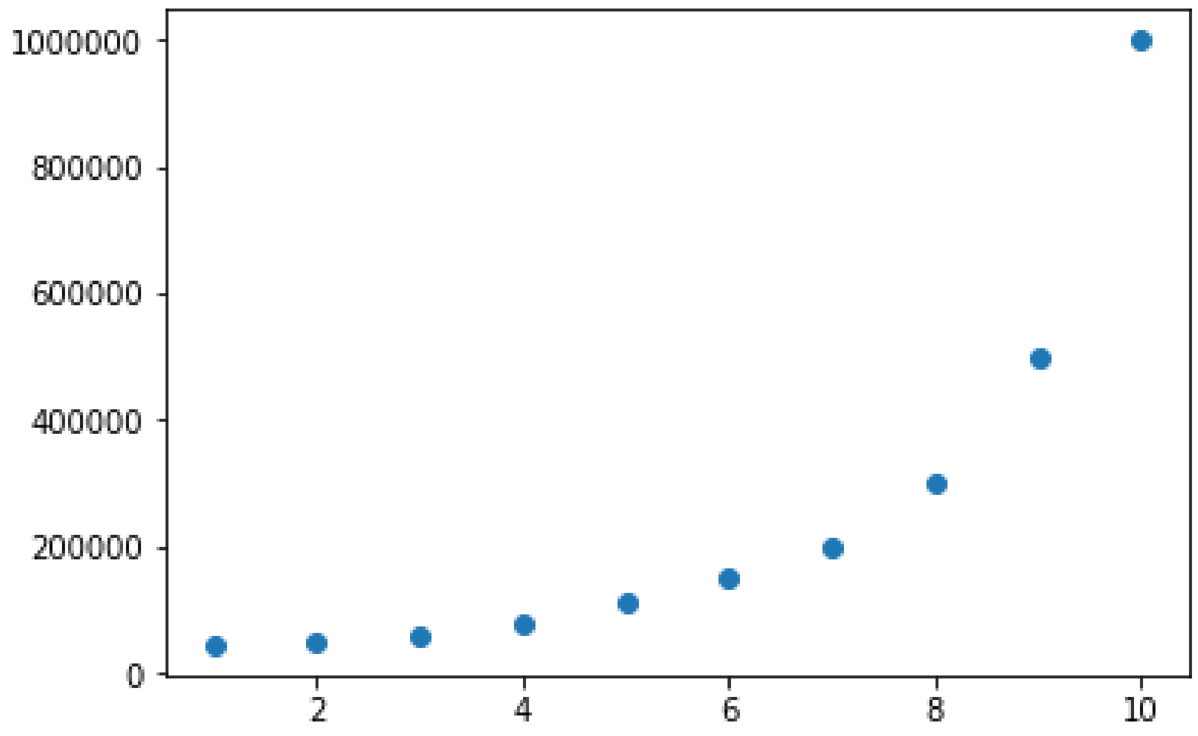


多项式回归可以写成下面这种形式：

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \cdots + \beta_k X_i^k$$

多项式回归

Position	Level	Salary
Business Analyst	1	45000
Junior Consultant	2	50000
Senior Consultant	3	60000
Manager	4	80000
Country Manager	5	110000
Region Manager	6	150000
Partner	7	200000
Senior Partner	8	300000
C-level	9	500000
CEO	10	1000000

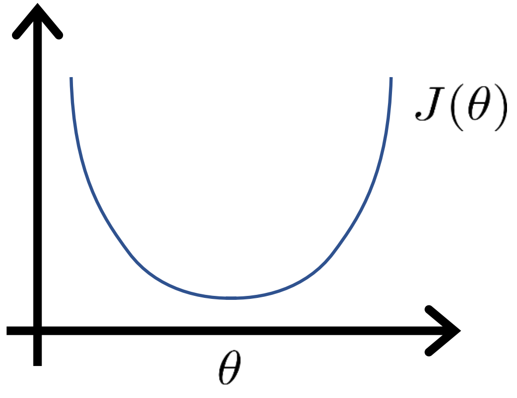


多项式回归-代码实战

标准方程法
Normal Equation

标准方程法

$$J(\theta) = a\theta^2 + b\theta + c$$



$$J(\theta_0, \theta_1, \ldots, \theta_m) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

$$\frac{\partial}{\partial\theta_j}J(\theta) = \cdots = 0$$

求解： $\theta_0, \theta_1, \ldots, \theta_n$

标准方程法

x_0	Size (feet^2) x_1	Number of bedrooms x_2	Number of floors x_3	Age of home (years) x_4	Price ($1000) y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178

$$X = \begin{array}{c} \begin{array}{ccccc} x_0 & x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix} \end{array} \qquad w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} \qquad y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}$$

标准方程法

$$J(\theta_0, \theta_1, \ldots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

$$X = \begin{array}{c} \\ \\ \end{array}\begin{array}{ccccc} x_0 & x_1 & x_2 & x_3 & x_4 \\ \end{array}$$

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix} \qquad w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} \qquad y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}$$

$$\sum_{i=1}^{m} (h_w(x^i) - y^i)^2 = (y - Xw)^T (y - Xw)$$

标准方程法

https://en.wikipedia.org/wiki/Matrix_calculus#Scalar-by-vector_identities

分子布局（Numerator-layout）：分子为列向量或者分母为行向量)
分母布局（Denominator-layout）：分子为行向量或者分母为列向量)

$$\frac{\partial (y-Xw)^T(y-Xw)}{\partial w}$$

$$\frac{\partial (y^Ty-y^TXw-w^TX^Ty+w^TX^TXw)}{\partial w}$$

$$\frac{\partial y^Ty}{\partial w}-\frac{\partial y^TXw}{\partial w}-\frac{\partial w^TX^Ty}{\partial w}+\frac{\partial w^TX^TXw}{\partial w}$$

标准方程法

$$\frac{\partial y^T y}{\partial w} - \frac{\partial y^T X w}{\partial w} - \frac{\partial w^T X^T y}{\partial w} + \frac{\partial w^T X^T X w}{\partial w}$$

① $$\frac{\partial y^T y}{\partial w} = 0$$

② $$\frac{\partial y^T X w}{\partial w} = X^T y$$

③ $$\frac{\partial w^T X^T y}{\partial w} = \frac{\partial (w^T X^T y)^T}{\partial w} = \frac{\partial y^T X w}{\partial w} = X^T y$$

④ $$\frac{\partial w^T X^T X w}{\partial w} = 2X^T X w$$

标准方程法

$$\frac{\partial y^T y}{\partial w} - \frac{\partial y^T Xw}{\partial w} - \frac{\partial w^T X^T y}{\partial w} + \frac{\partial w^T X^T Xw}{\partial w} = 0 - X^T y - X^T y + 2X^T Xw$$

$$-2X^T y + 2X^T Xw = 0$$

$$X^T Xw = X^T y$$

$$(X^T X)^{-1} X^T Xw = (X^T X)^{-1} X^T y$$

$$w = (X^T X)^{-1} X^T y$$

$(X^T X)^{-1}$是$X^T X$的逆矩阵

矩阵不可逆的情况

1.线性相关的特征(多重共线性)。
例如：x1为房子的面积，单位是平方英尺
x2为房子的面积，单位是平方米
预测房价
1平方英尺≈0.0929平方米

2.特征数据太多（样本数m≤特征数量n）

梯度下降法VS标准方程法

梯度下降法

缺点：
需要选择合适的学习率
需要迭代很多个周期
只能得到最优解的近似值

优点：
当特征值非常多的时候也
可以很好的工作

标准方程法

优点：
不需要学习率
不需要迭代
可以得到全局最优解

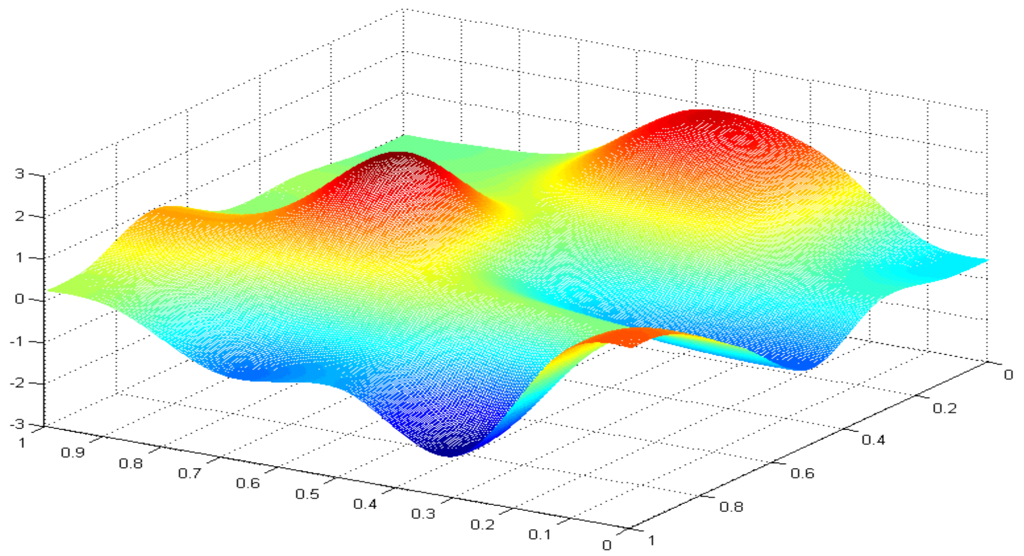
缺点：
需要计算$(X^T X)^{-1}$
时间复杂度大约是O(n^3)
n是特征数量

线性回归标准方程法

特征缩放
交叉验证法

特征缩放

假设有多个特征用于决定房屋的价格
x1 = 房子面积($1000000cm^2$-$2000000cm^2$)
x2 = 房间数量(1-5)



数据归一化

数据归一化就是把数据的取值范围处理为0-1或者-1-1之间。

任意数据转化为0-1之间：
newValue = (oldValue-min)/(max-min)
(1,3,5,7,9)
(1-1)/(9-1)=0
(3-1)/(9-1)=1/4
(5-1)/(9-1)=1/2
(7-1)/(9-1)=3/4
(9-1)/(9-1)=1

任意数据转化为-1-1之间：
newValue = ((oldValue-min)/(max-min)-0.5)*2

均值标准化

x为特征数据，u为数据的平均值，s为数据的方差

newValue = (oldValue-u)/s

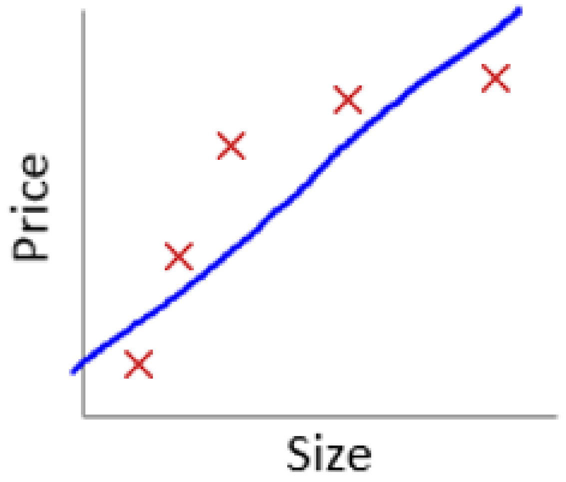
(1,3,5,7,9)
u = (1+3+5+7+9)/5=5
s = $((1-5)^2+(3-5)^2+(5-5)^2+(7-5)^2+(9-5)^2)/5=8$
(1-5)/8=-1/2
(3-5)/8=-1/4
(5-5)/8=0
(7-5)/8=1/4
(9-5)/8=1/2

交叉验证法

Training set
Training folds
Test fold
1st iteration
2nd iteration
3rd iteration
10th iteration
E_1
E_2
E_3
E_{10}
$E = \frac{1}{10}\sum_{i=1}^{10} E_i$

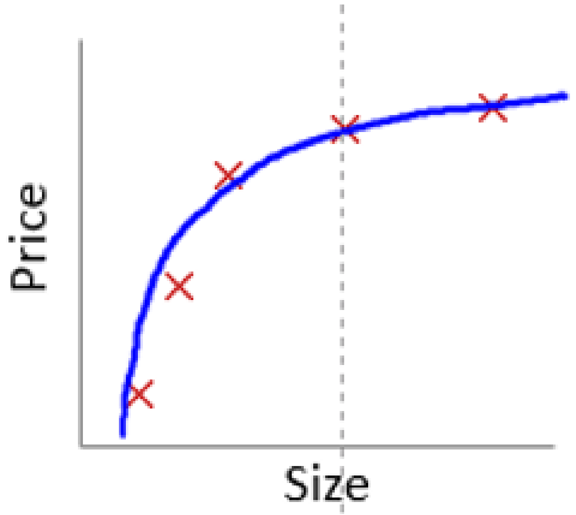
过拟合(Overfitting)
正则化(Regularized)

拟合



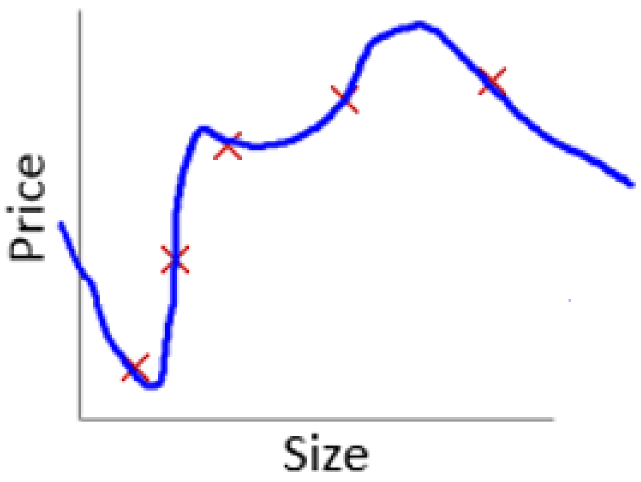
$$\theta_0 + \theta_1 x$$

欠拟合(Underfitting)



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

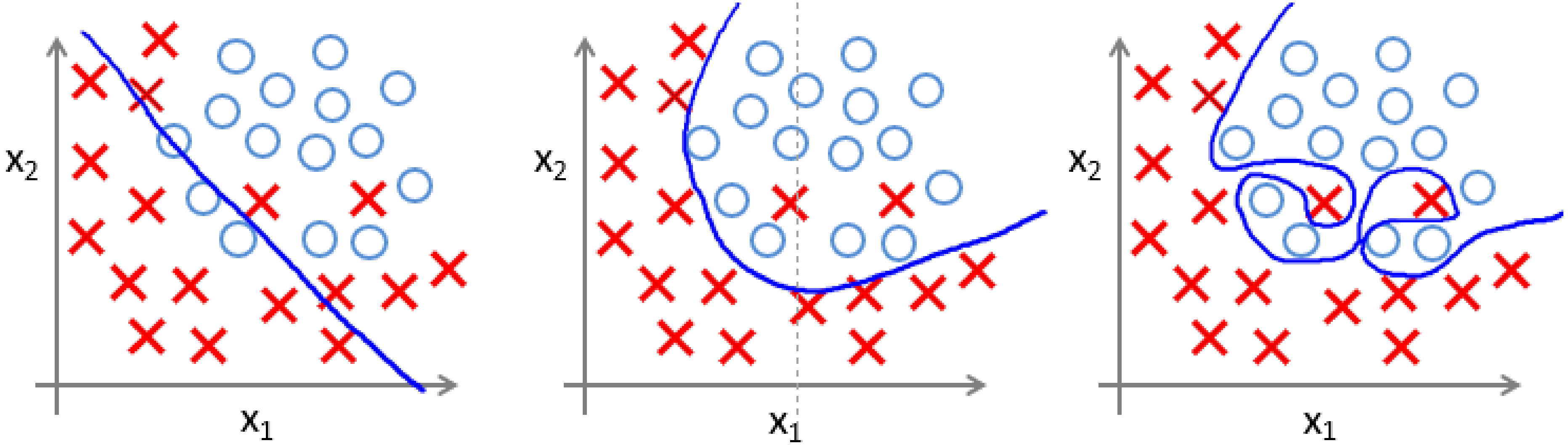
正确拟合(Just right)



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

过拟合(Overfitting)

拟合



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

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

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

欠拟合(Underfitting)

正确拟合(Just right)

过拟合(Overfitting)

防止过拟合

1.减少特征

2.增加数据量

3.正则化
(Regularized)

正则化(Regularized)

正则化代价函数：

L2正则化： $$J(\theta)=\frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2+\lambda\sum_{j=1}^{n}\theta_j^2\right]$$

L1正则化： $$J(\theta)=\frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2+\lambda\sum_{j=1}^{n}\left|\theta_{\mathrm{j}}\right|\right]$$

岭回归
Ridge Regression

岭回归(Ridge Regression)

$$w = (X^T X)^{-1} X^T y$$

如果数据的特征比样本点还多，数据特征n，样本个数m，如果n>m，则计算$(X^T X)^{-1}$时会出错。因为$(X^T X)$不是满秩矩阵，所以不可逆。

为了解决这个问题，统计学家引入了岭回归的概念。

$$w = (X^T X + \lambda I)^{-1} X^T y$$

λ为岭系数，I为单位矩阵(对角线上全为1，其他元素全为0)

岭回归(Ridge Regression)

$$J(\theta)=\frac{1}{2}\sum_{i=1}^{n}(h_\theta(x_i)-y_i)^2+\lambda\sum_{i}^{n}\theta_i^2$$

$$J(\theta)=\frac{1}{2}(X\theta-Y)^T(X\theta-Y)+\lambda\theta^T\theta$$

$$=\frac{1}{2}\left(\theta^TX^TX\theta-\theta^TX^TY-Y^TX\theta+Y^TY\right)+\lambda\theta^T\theta$$

$$\frac{\partial J(\theta)}{\partial\theta}=X^TX\theta-X^TY+\lambda\theta$$

$$\theta=\left(X^TX+\lambda I\right)^{-1}X^TY$$

岭回归(Ridge Regression)

岭回归最早是用来处理特征数多于样本的情况，现在也用于在估计中加入偏差，从而得到更好的估计。同时也可以解决多重共线性的问题。岭回归是一种有偏估计。

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

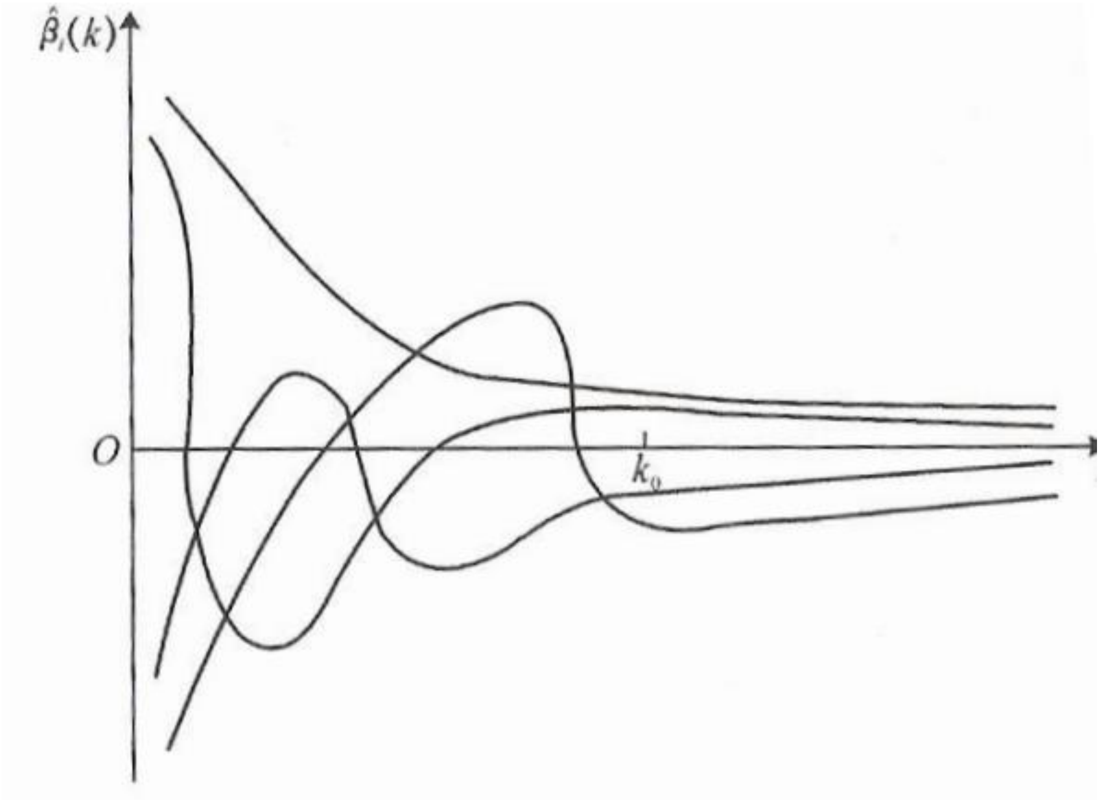
线性回归标准方程法： $\mathrm{w} = (X^TX)^{-1}X^T\mathrm{y}$

岭回归求解： $\mathrm{w} = (X^TX + \lambda I)^{-1}X^T\mathrm{y}$

λ为岭系数。

岭回归(Ridge Regression)

选择λ值，使到：
1.各回归系数的岭估计基本稳定。
2.残差平方和增大不太多。



Longley数据集

Longley数据集来自J．W．Longley（1967）发表在JASA上的一篇论文，是强共线性的宏观经济数据,包含GNP deflator(GNP平减指数)、GNP(国民生产总值)、Unemployed(失业率)、ArmedForces(武装力量)、Population(人口)、year(年份)，Emlpoyed(就业率)。
LongLey数据集因存在严重的多重共线性问题，在早期经常用来检验各种算法或计算机的计算精度。

sklearn-岭回归

标准方程法-岭回归

LASSO

LASSO

- Tibshirani(1996)提出了Lasso(The Least Absolute Shrinkage and Selectionator operator)算法。
- 通过构造一个一阶惩罚函数获得一个精炼的模型；通过最终确定一些指标（变量）的系数为零（岭回归估计系数等于0的机会微乎其微，造成筛选变量困难），解释力很强。
- 擅长处理具有多重共线性的数据，与岭回归一样是有偏估计。

LASSO与岭回归

岭回归代价函数：

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

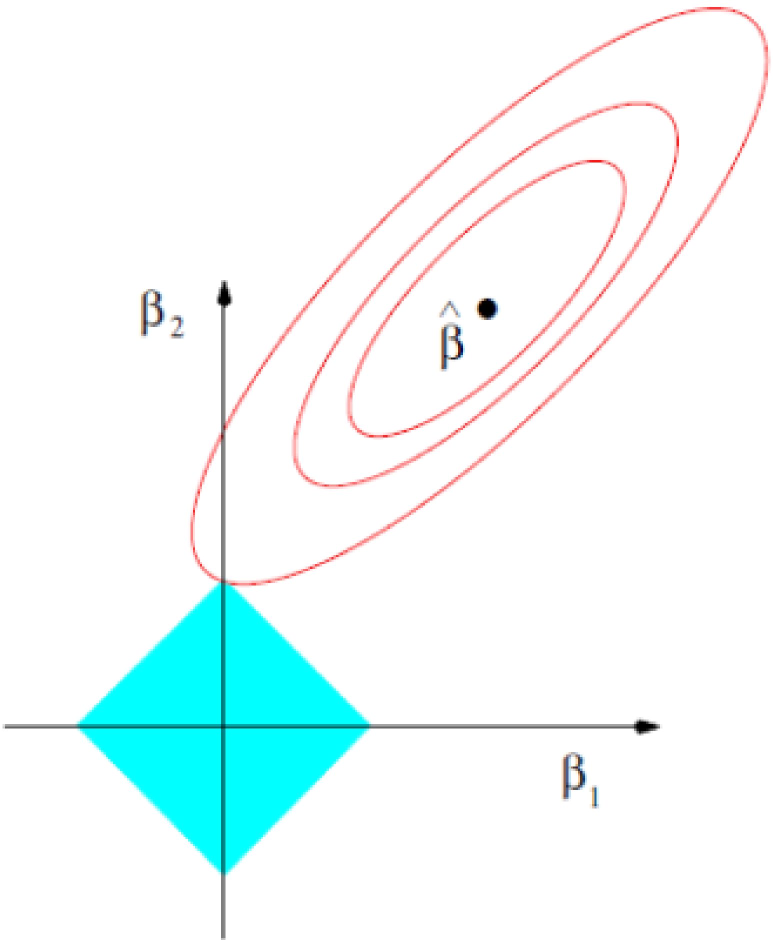
λ的值可以用于限制$\sum_{j=1}^{n}\theta_j^2\le t$

LASSO代价函数：

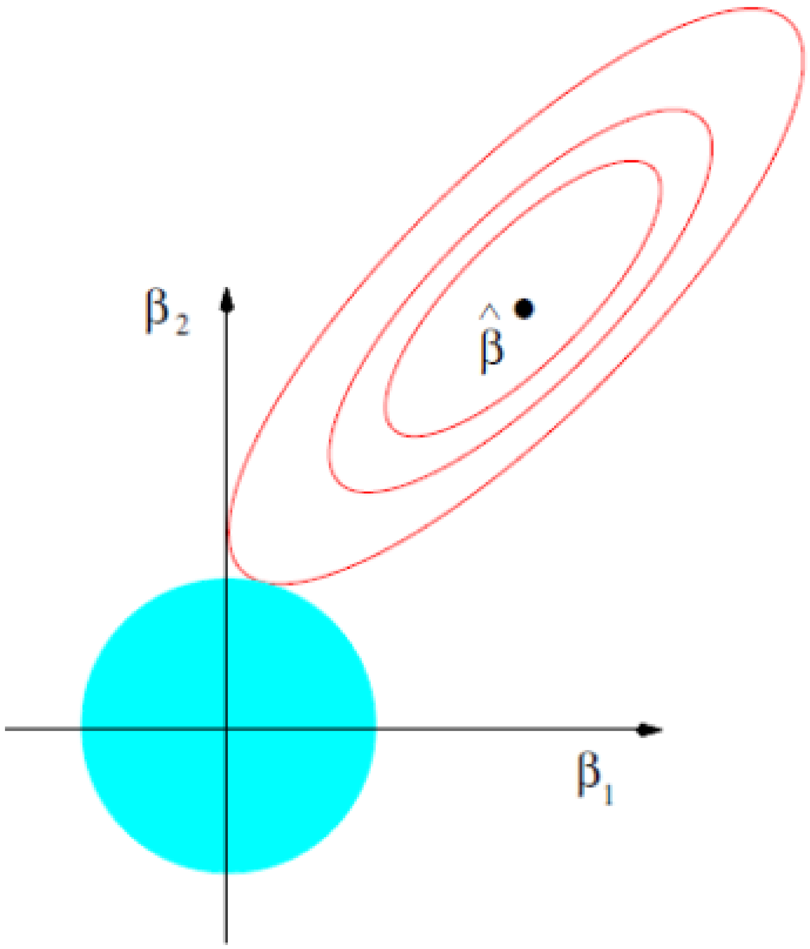
$$J(\theta)=\frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2+\lambda\sum_{j=1}^{n}\left|\theta_{\mathrm{j}}\right|\right]$$

λ的值可以用于限制$\sum_{j=1}^{n}\left|\theta_{\mathrm{j}}\right|\le t$

LASSO与岭回归

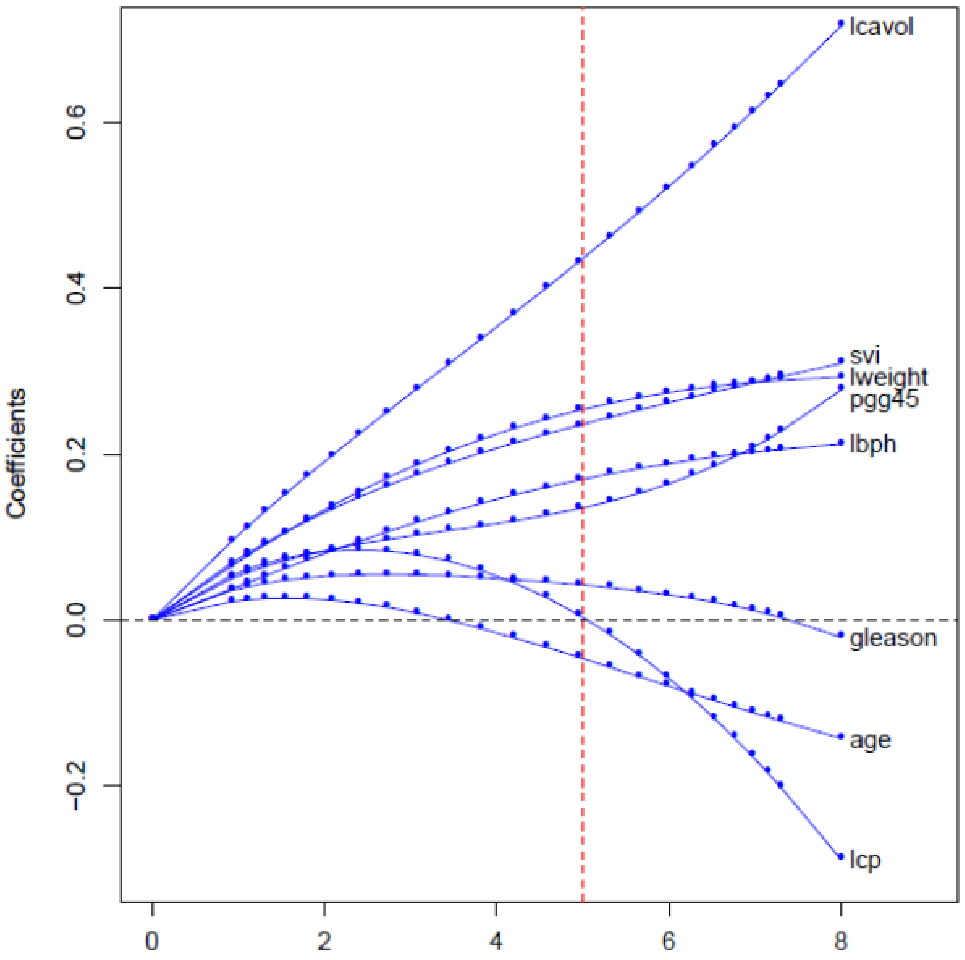


LASSO

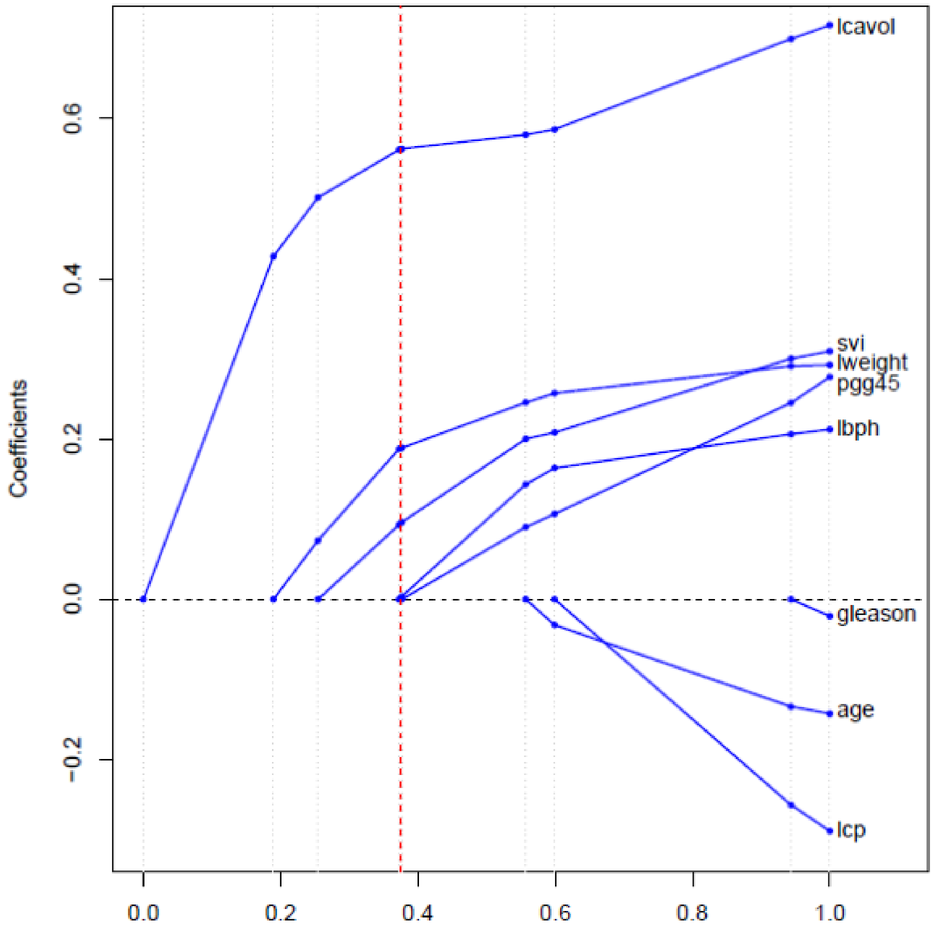


岭回归

LASSO与岭回归



岭回归



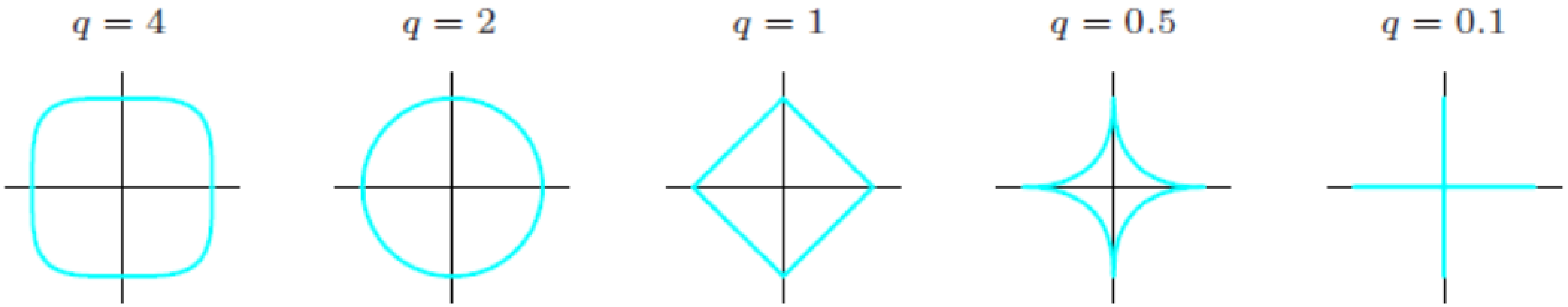
LASSO

LASSO

弹性网
Elastic Net

弹性网(Elastic Net)

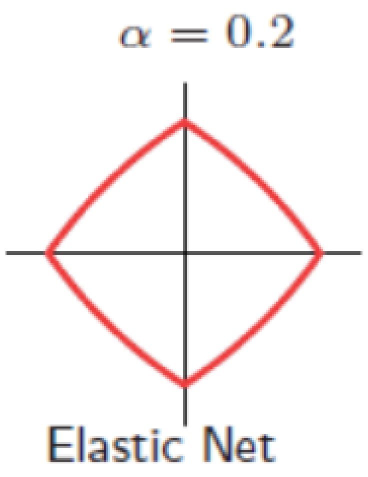
$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\left|\theta_j\right|^q\right]$$



弹性网(Elastic Net)

Zou and Hastie (2005)提出elasticnet

$$\lambda \sum_{j=1}^{n} \left(\alpha \theta_j^2 + (1-\alpha)\left|\theta_j\right|\right)$$



sklearn-弹性网ElasticNet