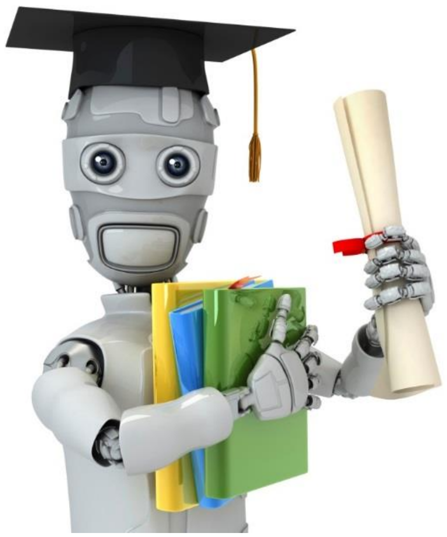


Machine Learning

Regression

梁毅雄

yxliang@csu.edu.cn

Some materials from Andrew Ng, Zico Kolter, Hung-yi Lee, Ryan Tibshirani, Fei-Fei Li and others

单变量线性回归

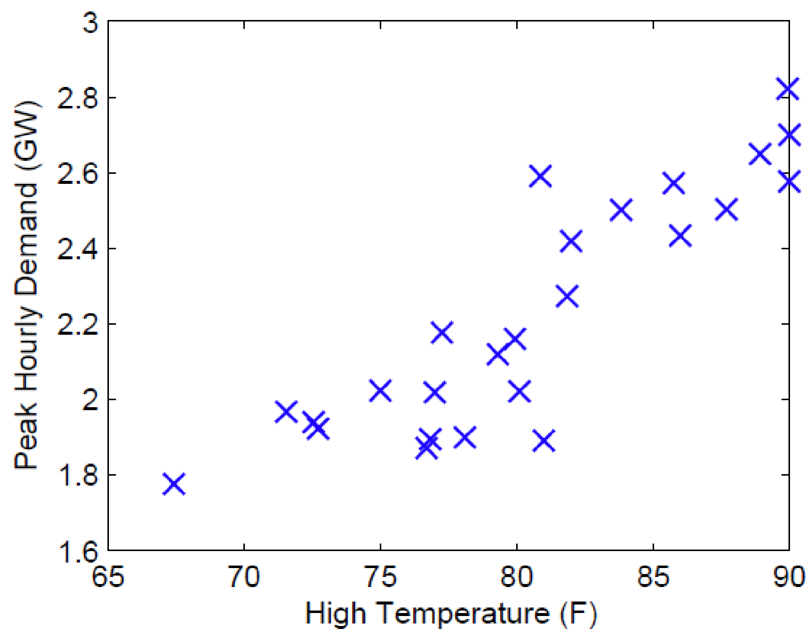
- 任务：预测明天某城市的峰值用电量
- 首先需要收集以往数据

High Temperature (F)	Peak Demand (GW)
76.7	1.87
72.7	1.92
71.5	1.96
86.0	2.43
90.0	2.69
87.7	2.50
⋮	⋮

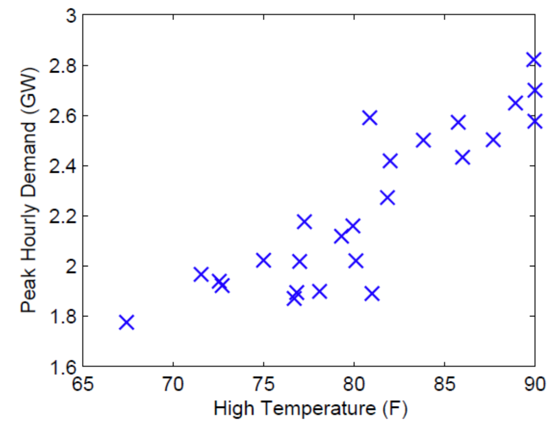
单变量线性回归

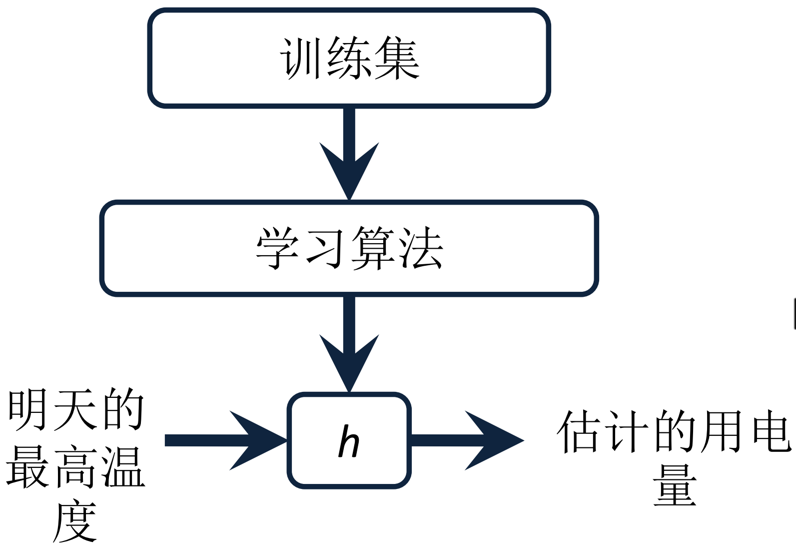
High Temperature (F)	Peak Demand (GW)
76.7	1.87
72.7	1.92
71.5	1.96
86.0	2.43
90.0	2.69
87.7	2.50
⋮	⋮

- 任务： 预测明天某城市的峰值用电量
- 可视化数据



单变量线性回归





- 模型表示：如何表示*h*？

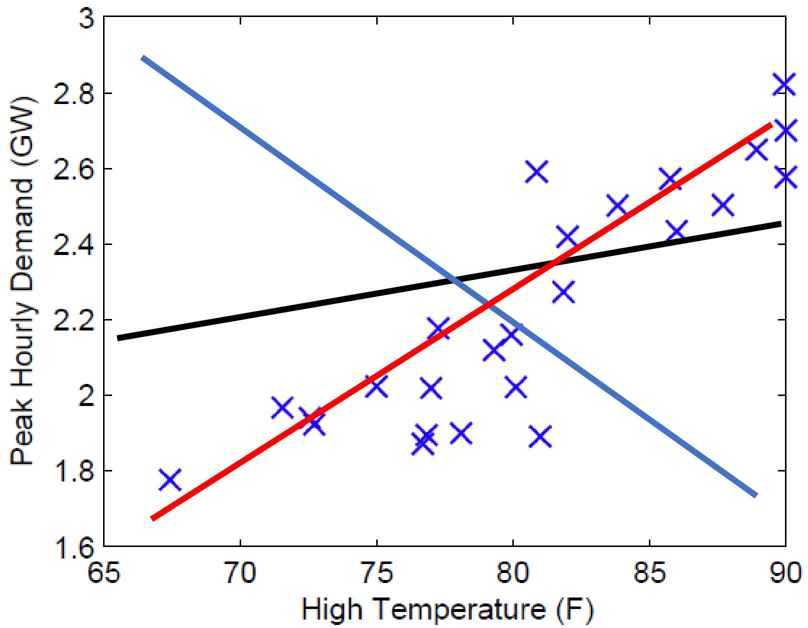
$$\text{Peak demand} \approx \theta_0 + \theta_1 \cdot (\text{High temperature})$$

单变量线性回归
一元线性回归

单变量线性回归

- 模型表示
- 等价于找一条最符合训练数据的直线
- 如何衡量“最符合”?
- 或者说如何选择参数 θ_i

$$\text{Peak demand} \approx \theta_0 + \theta_1 \cdot (\text{High temperature})$$



单变量线性回归

Notation:

- 输入特征：$x^{(i)} \in \mathbb{R}^{n+1}$, $i = 1, \cdots, m$. 如

$$x^{(i)} \in \mathbb{R}^2 = \begin{bmatrix} 1 \\ \text{high temperature for day } i \end{bmatrix}$$

- 输出：$y^{(i)} \in \mathbb{R}$，　　如$y^{(i)} = \{\text{peak demand for day i}\}$
- 参数：$\theta = \mathbb{R}^{n+1}$
- 假设$h_\theta(x) : \mathbb{R}^{n+1} \to \mathbb{R}$, 如$h_\theta(x) = \theta_0 + \theta_1 x$

损失函数

- 衡量“最符合”：通常采用损失函数$\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$
- 直观上，损失函数应该满足：
  - 非负：不存在负损失
  - 如果预测结果$h_\theta(x)$与给定的y差别小，则损失小，反之则损失大
- 如平方损失: $\ell(h_\theta(x), y) = (h_\theta(x) - y)^2$

损失函数

三要素

- 假设：$h_\theta(x) = \theta_0 + \theta_1 x$, 其中参数为：$\theta_0, \theta_1$
- 目标函数:

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

- 优化算法：给定训练集，如何找到最优的参数θ使得

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

损失函数

原始问题

- 假设：$h_\theta(x) = \theta_0 + \theta_1 x$
- 参数：θ_0, θ_1
- 目标函数:
  $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$
- Goal: $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

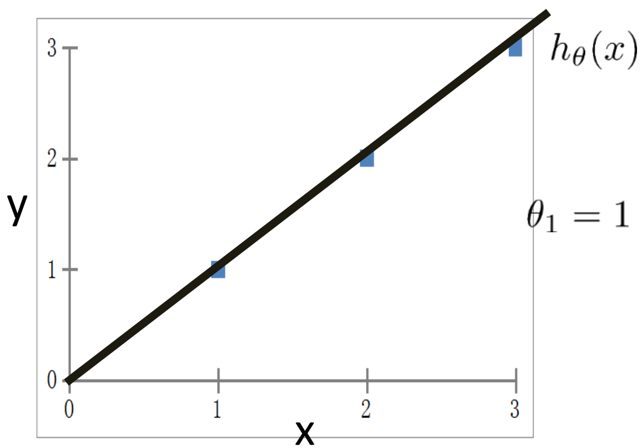
简化后

- 假设：$h_\theta(x) = \theta_1 x$
- 参数：θ_1
- 目标函数:
  $J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$
- Goal: $\min_{\theta_1} J(\theta_1)$

损失函数

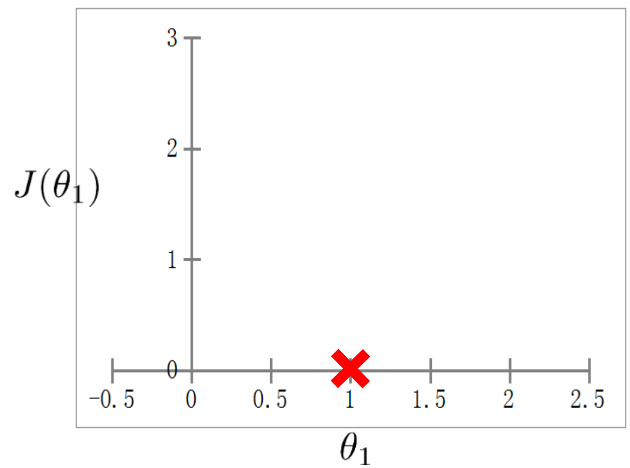
$h_\theta(x)$

(对于固定的θ_1，就有一个函数对应)



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

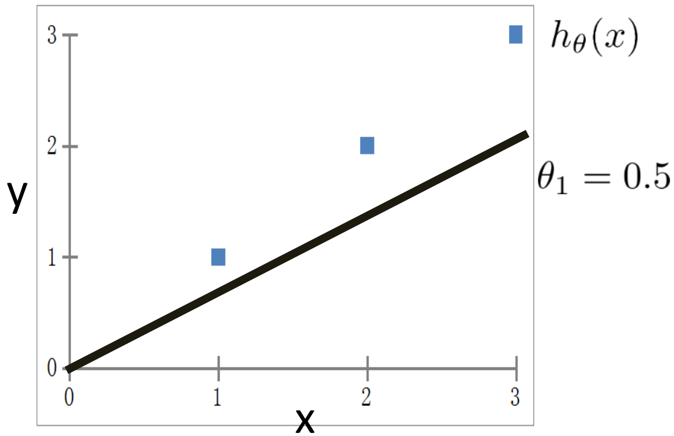
$J(\theta_1)$ (关于θ_1的函数)



损失函数

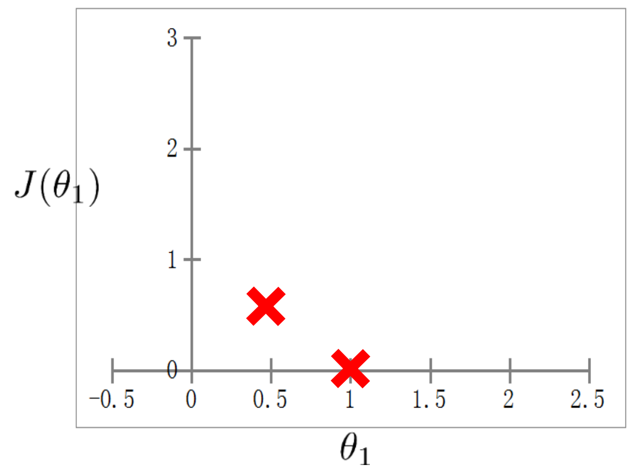
$h_\theta(x)$

(对于固定的θ_1, 就有一个函数对应)



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

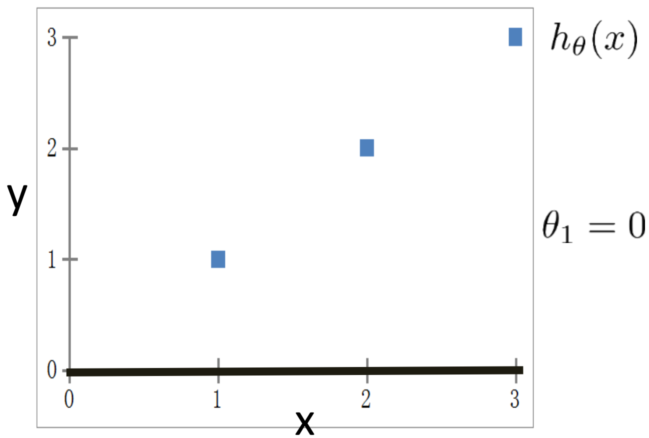
$J(\theta_1)$ (关于θ_1的函数)



损失函数

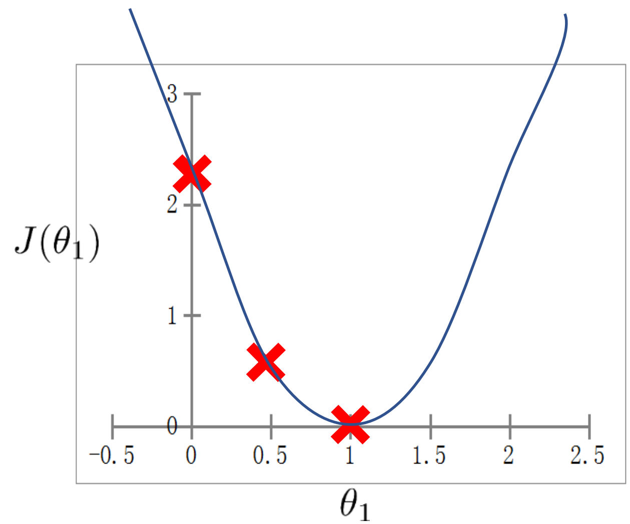
$h_\theta(x)$

(对于固定的θ_1, 就有一个函数对应)



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

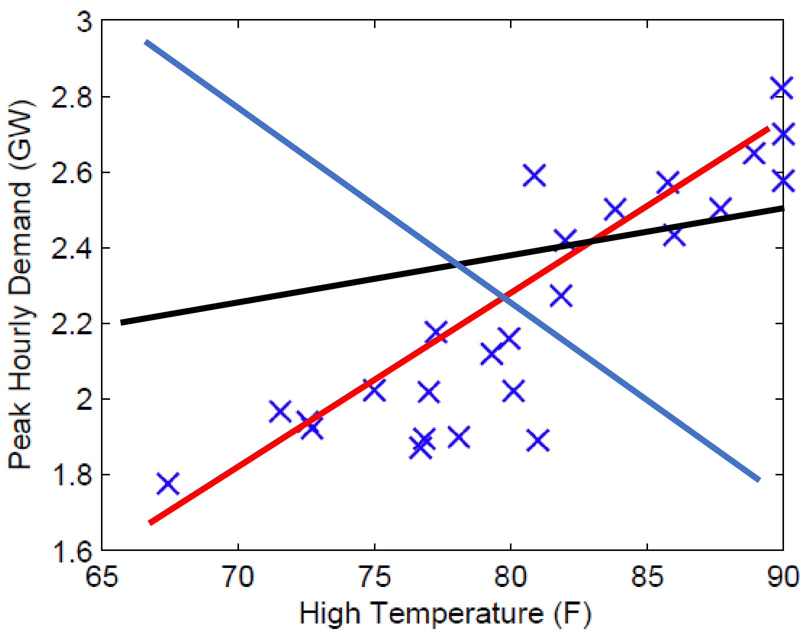
$J(\theta_1)$ (关于θ_1的函数)

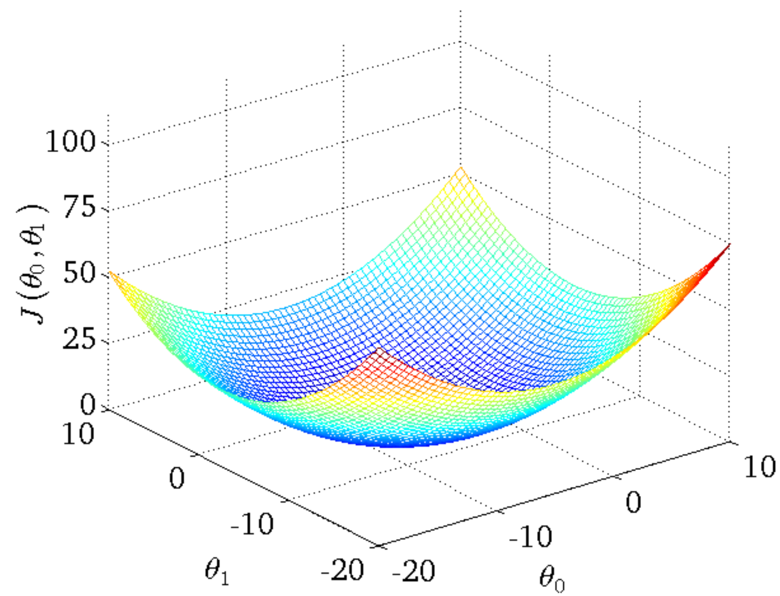


损失函数

$h_\theta(x)$

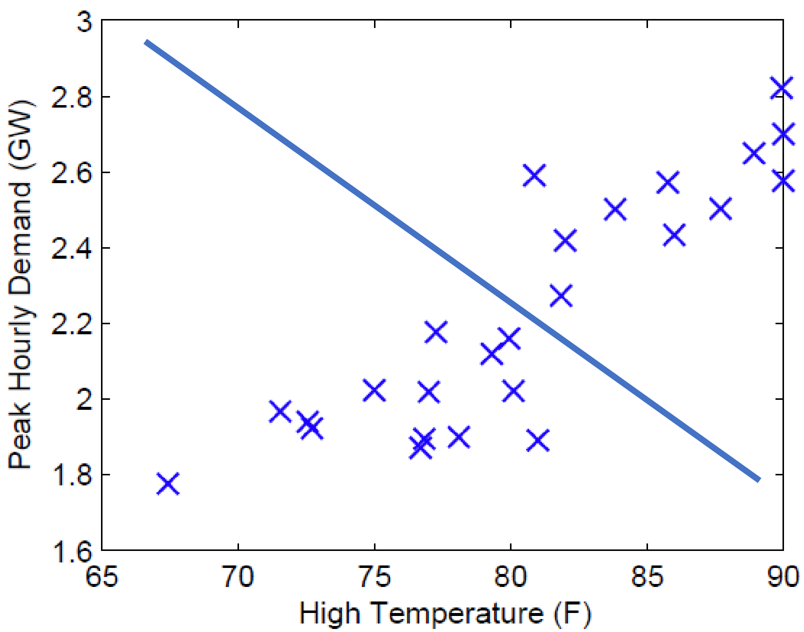
$J(\theta_0, \theta_1)$ (关于θ_0, θ_1的函数)



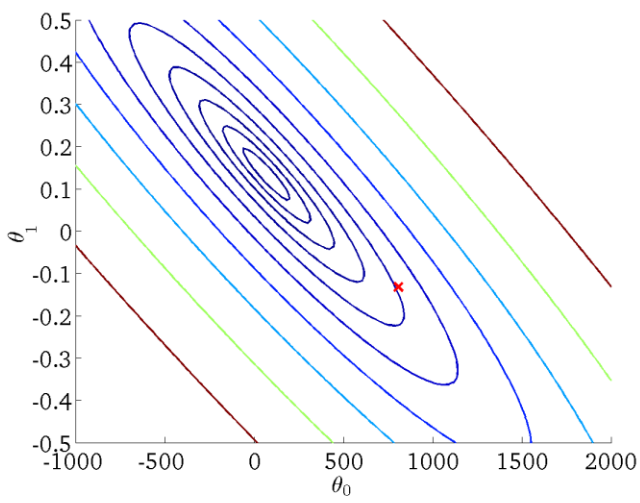


损失函数

$h_\theta(x)$



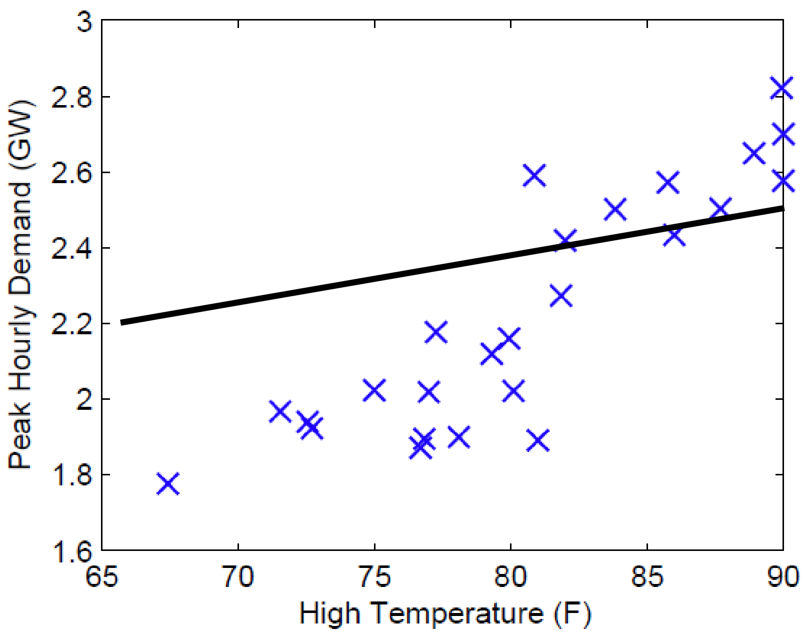
$J(\theta_0, \theta_1)$ (关于θ_0, θ_1的函数)

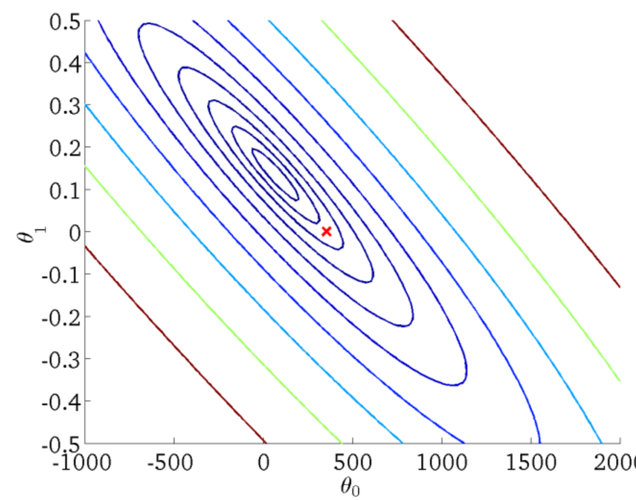


损失函数

$h_\theta(x)$

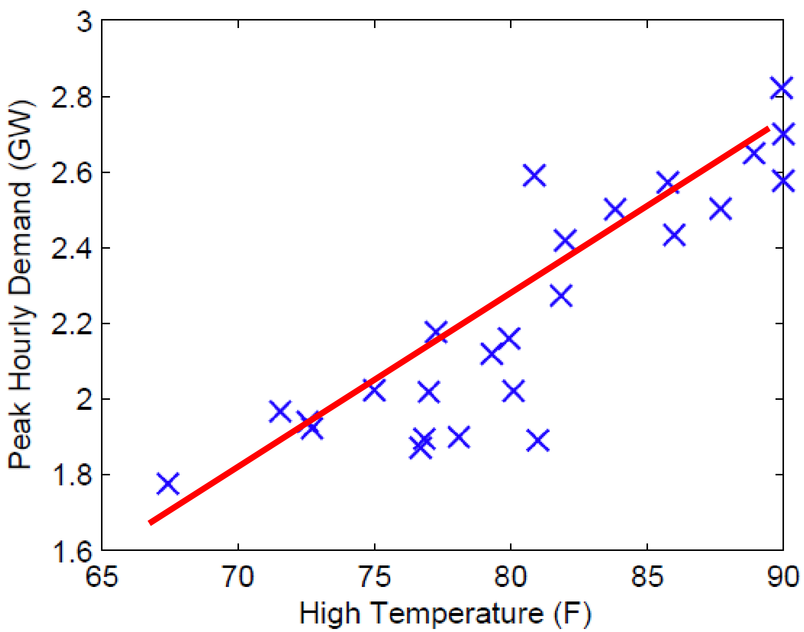
$J(\theta_0, \theta_1)$ (关于θ_0, θ_1的函数)



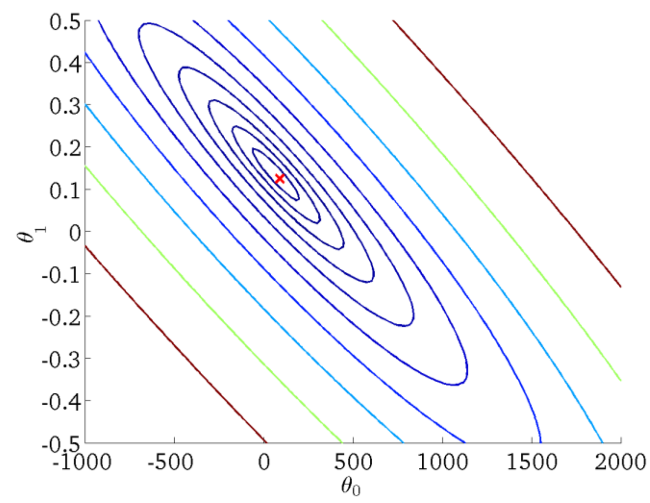


损失函数

$h_\theta(x)$



$J(\theta_0, \theta_1)$ (关于θ_0, θ_1的函数)



参数优化

- 如何找到最优的参数$\theta^* = arg\min_{\theta} J(\theta)$?
  - 策略1：穷举所有的θ STUPID
  - 策略2：随机搜索 瞎猫碰死耗子!
  - 策略3：梯度下降

梯度下降

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

正确: 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:

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

梯度下降

$$\theta^* = arg \min_{\theta} J(\theta)$$

- Consider loss function $J(\theta)$ with one parameter θ :

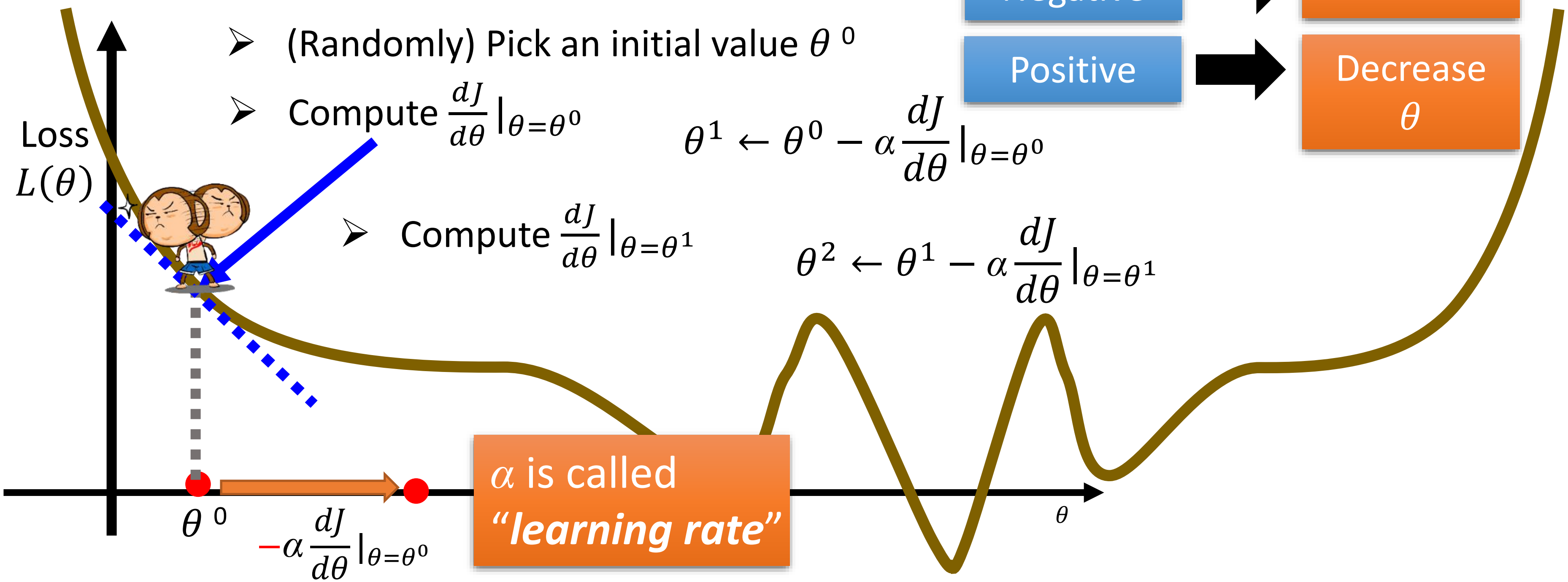
Negative	→	Increase θ
Positive	→	Decrease θ

  - ➢ (Randomly) Pick an initial value $\theta^{\,0}$
  - ➢ Compute $\frac{dJ}{d\theta}|_{\theta=\theta^0}$

$$\theta^1 \leftarrow \theta^0 - \alpha \frac{dJ}{d\theta}|_{\theta=\theta^0}$$

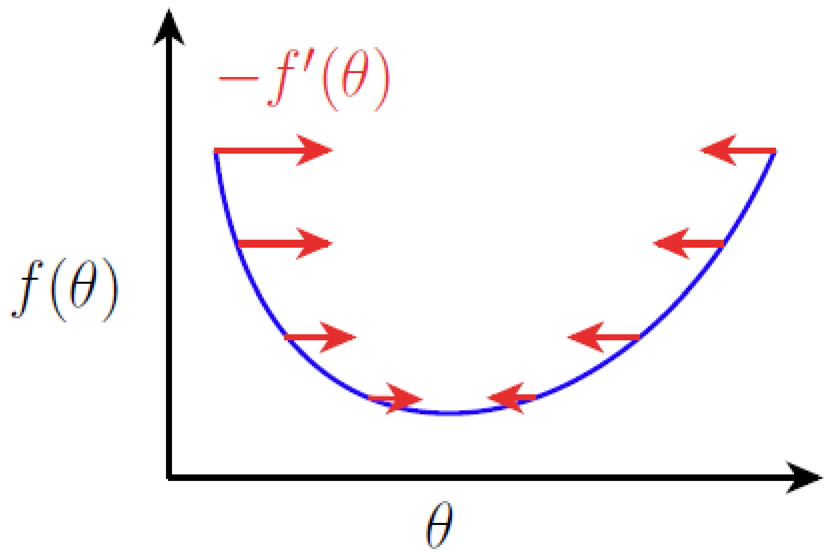
  - ➢ Compute $\frac{dJ}{d\theta}|_{\theta=\theta^1}$

$$\theta^2 \leftarrow \theta^1 - \alpha \frac{dJ}{d\theta}|_{\theta=\theta^1}$$



梯度下降

The (negative) derivative has another useful property: it points in a "downhill" direction



Repeat: $\quad \theta := \theta - \alpha f'(\theta)$

梯度下降

For vector $\theta \in \mathbb{R}^n$, the analog of the derivative is called the *gradient*

$$\nabla_\theta f(\theta) \in \mathbb{R}^n = \begin{bmatrix} \frac{\partial f(\theta)}{\partial \theta_1} \\ \frac{\partial f(\theta)}{\partial \theta_2} \\ \vdots \\ \frac{\partial f(\theta)}{\partial \theta_n} \end{bmatrix}$$

The general gradient descent algorithm is the same as before, just using the gradient

$$\text{Repeat:} \quad \theta := \theta - \alpha \nabla_\theta f(\theta)$$

梯度下降

Gradient descent interpretation

At each iteration, consider the expansion

$$f(y) \approx f(x) + \nabla f(x)^T (y - x) + \frac{1}{2\alpha} \|y - x\|_2^2$$

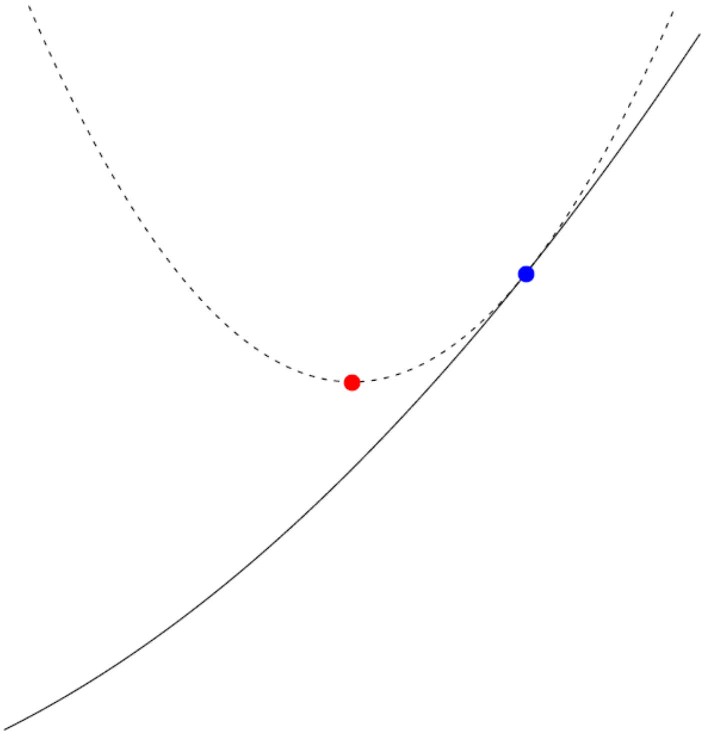
Quadratic approximation, replacing usual $\nabla^2 f(x)$ by $\frac{1}{\alpha} I$

$f(x) + \nabla f(x)^T (y - x)$	linear approximation to f
$\frac{1}{2\alpha} \|y - x\|_2^2$	proximity term to x, with weight $1/(2\alpha)$

Choose next point $y = x^+$ to minimize quadratic approximation:

$$x^+ = x - \alpha \nabla f(x)$$

梯度下降



Blue point is x, red point is

$$x^+ = \arg\min_y f(x) + \nabla f(x)^T (y - x) + \frac{1}{2\alpha}\|y - x\|_2^2$$

单变量线性回归模型的梯度下降

梯度下降法

repeat until convergence {

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

$$\text{(for } j = 1 \text{ and } j = 0\text{)}$$

}

$$j = 0 : \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) =$$

$$j = 1 : \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) =$$

线性回归模型

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

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

单变量线性回归模型的梯度下降

梯度下降法

repeat until convergence {

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

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

}

同步更新
θ_0 和 θ_1

线性回归模型

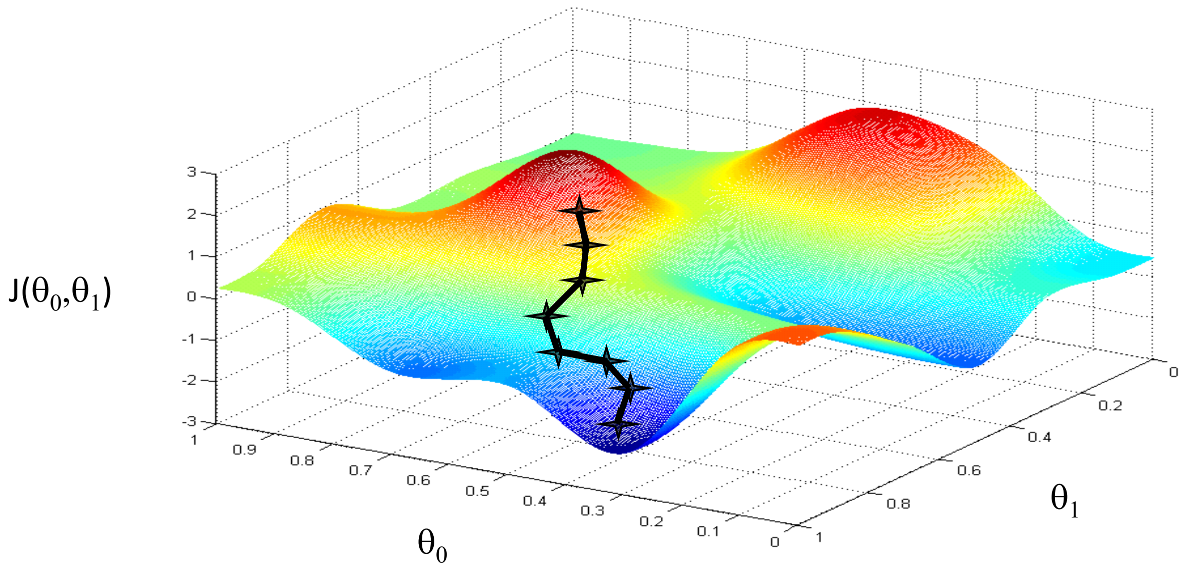
$$h_\theta(x) = \theta_0 + \theta_1 x$$

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

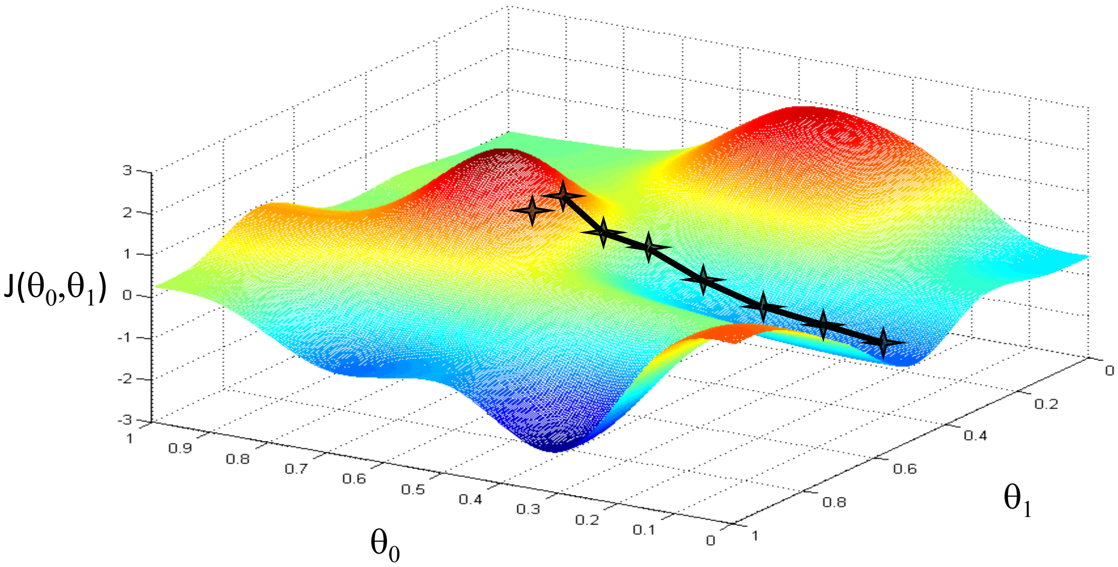
"批处理" 梯度下降

"批处理": 梯度下降的每一步都使用所有的训练样本.

梯度下降

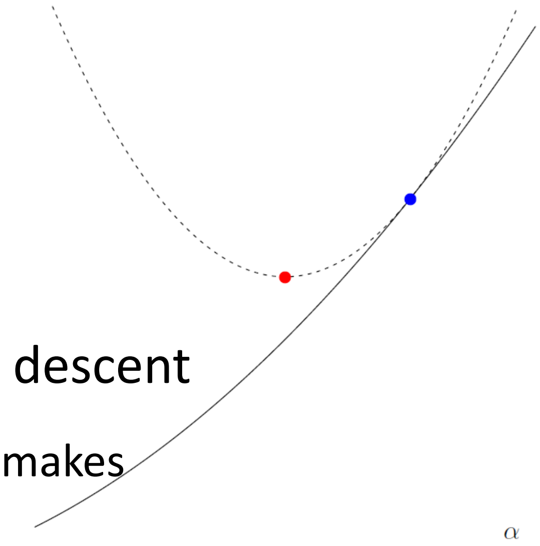


梯度下降



3
2
1
0
-1
-2
-3
$J(\theta_0,\theta_1)$
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

思考:



When solving: $\theta^* = \arg\max_{\theta} J(\theta)$ by gradient descent

Each time we update the parameters, we obtain θ that makes $J(\theta)$ smaller.

Blue point is x, red point is

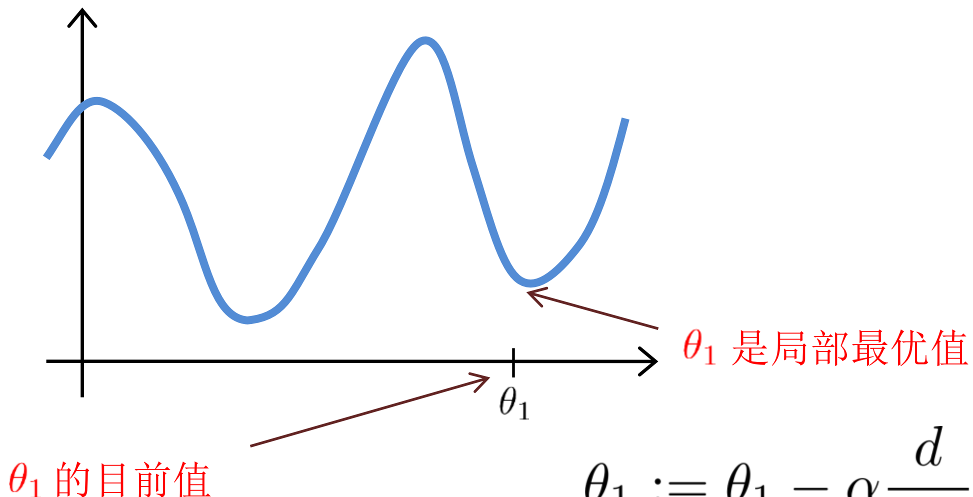
$$x^+ = \underset{y}{\operatorname{argmin}} \; f(x) + \nabla f(x)^T (y - x) + \frac{1}{2t} \|y - x\|_2^2$$

1. 能否保证找到最优的参数?

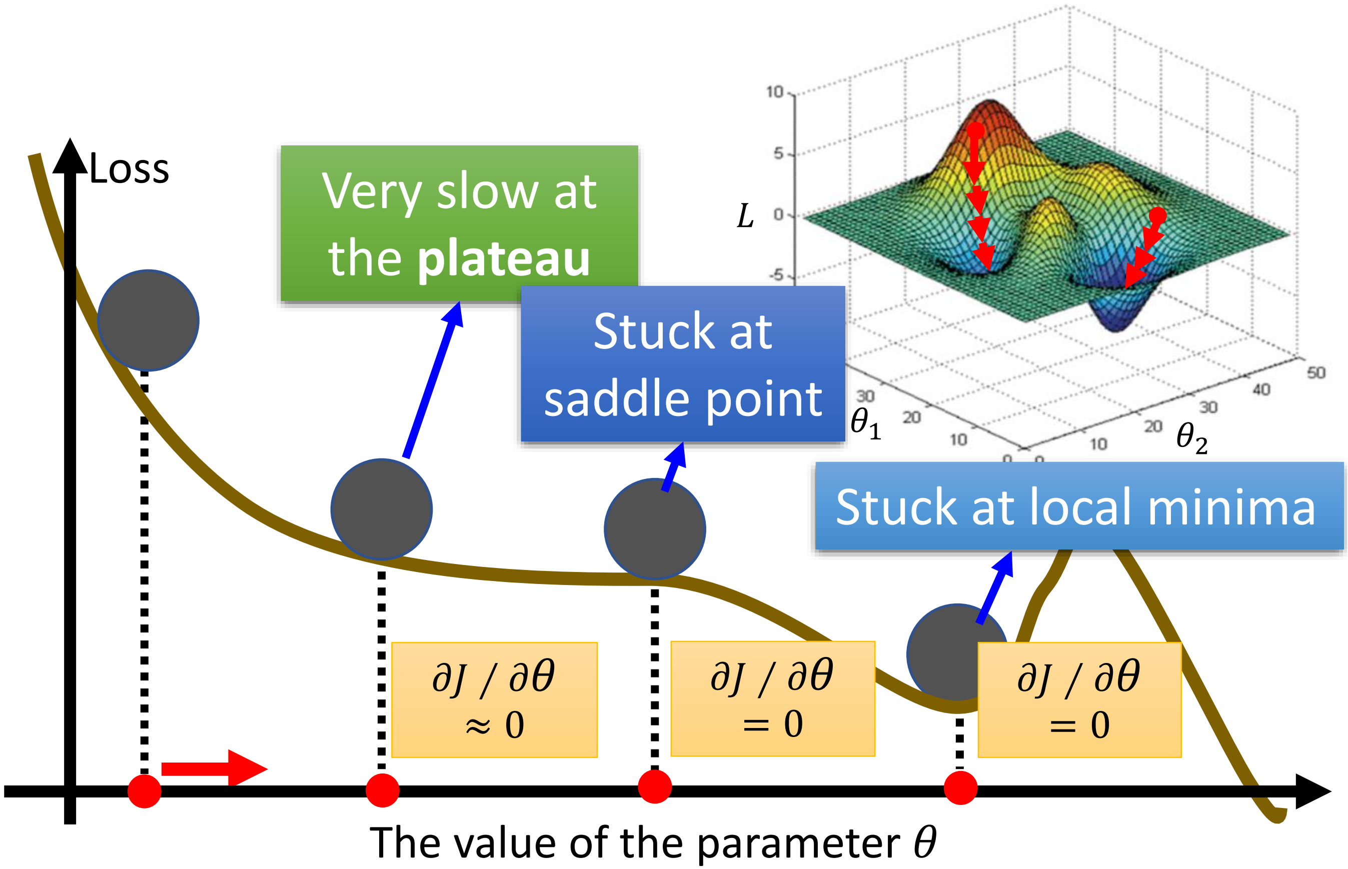
2. 能否保证 $J(\theta^0) > J(\theta^1) > J(\theta^2) > \cdots$

即: 梯度下降法参数更新能是否保证目标函数的值下降?

3. 如何选择参数α(学习率)?



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



Loss
Very slow at the **plateau**
Stuck at saddle point
Stuck at local minima
$\partial J / \partial \theta \approx 0$
$\partial J / \partial \theta = 0$
$\partial J / \partial \theta = 0$
The value of the parameter θ
L
θ_1
θ_2

多特征 (变量)

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

尺寸 (feet²) x	价格 ($1000) y
2104	460
1416	232
1534	315
852	178
…	…

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

尺寸 (feet²)	卧室个数	楼层数	房龄（年）	价格($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

注释：

n: 特征个数

$x^{(i)}$: 第i个训练样本的输入(特征)

$x_j^{(i)}$: 第i个训练样本的第j个特征

多特征 (变量)

- 假设：$h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n,\ x_0 = 1$
- 参数：$\theta_0, \theta_1, \cdots, \theta_n$
- 目标函数:

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

- 梯度下降：

Repeated until converge $\{$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1, \cdots, \theta_n)$$

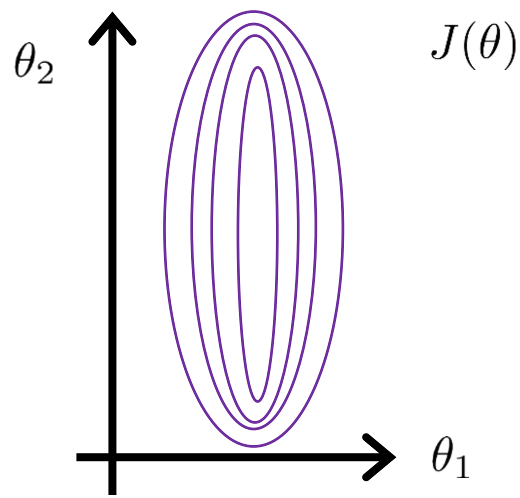
$\}$ 同步更新，对所有$j = 0, 1, \cdots, n$

特征尺度归一化

- 确保特征在相同的尺度.

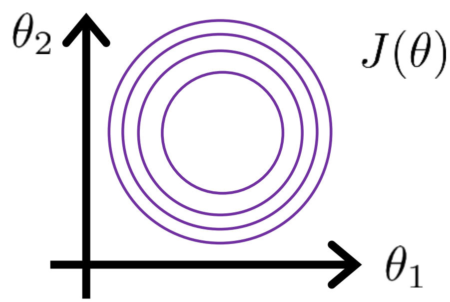
E.g. x_1 = size (0-2000 feet^2)

x_2 = number of bedrooms (1-5)



$$x_1 = \frac{\text{size (feet}^2\text{)}}{2000}$$

$$x_2 = \frac{\text{number of bedrooms}}{5}$$



特征尺度归一化

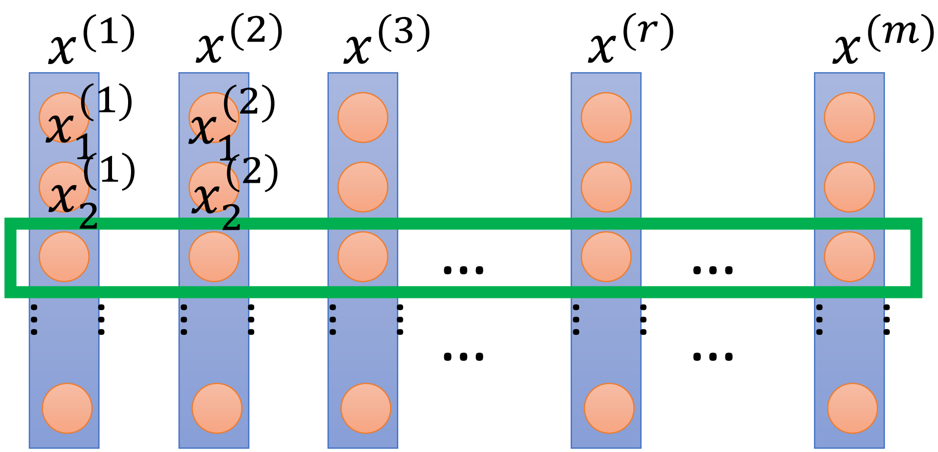
- 范围归一化：使得每个特征尽量接近某个范围，如$0 \le x_i \le 1$,
- 零均值归一化: 用$x_i - \mu_i$ 替代x_i, 即$x_i - \mu_i \rightarrow x_i$, 其中$\mu_i = \frac{1}{m}\sum_{i=1}^{m} x_i$为均值($x_0$除外)
- 零均值+ 范围归一化: 如

  $x_1 = \frac{size-1000}{2000}$,

  $x_2 = \frac{\#bedrooms-2}{5}$

  $-0.5 \le x_1 \le 0.5$, $-0.5 \le x_2 \le 0.5$
- 零均值单位方差归一化:

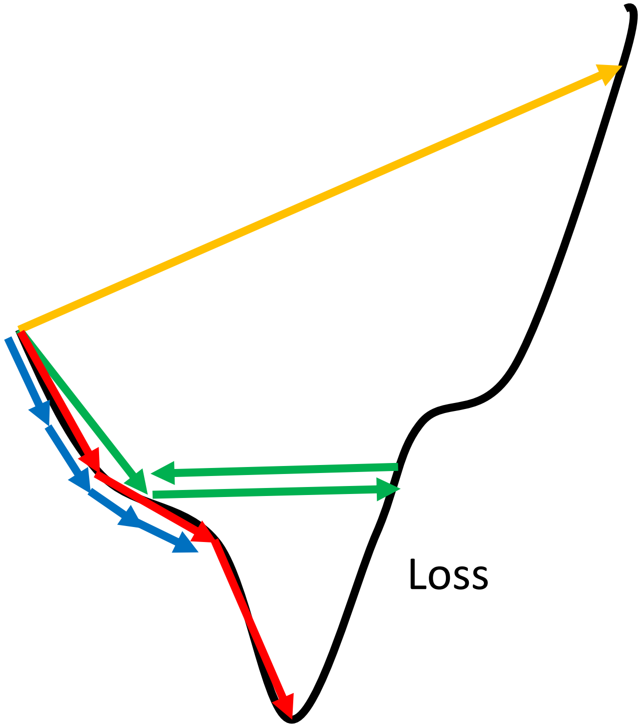
  $\frac{x_i-\mu_i}{\sigma_i} \rightarrow x_i$



学习率

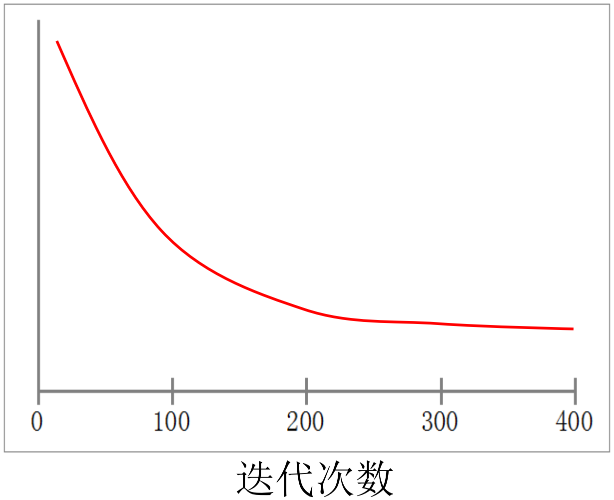
梯度下降: $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$

- ”调试”：怎样确保梯度下降算法正确的执行
- 怎样选择学习速率α



如何确保梯度下降算法正确的执行？

$\min_{\theta} J(\theta)$

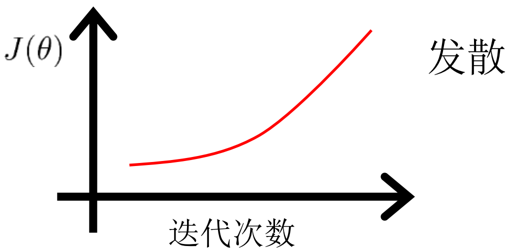


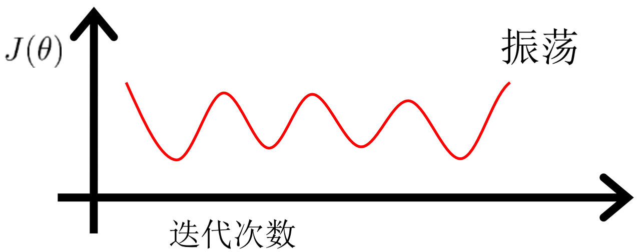
自动收敛测试: 每次迭代损失函数$J(\theta)$是否减少？

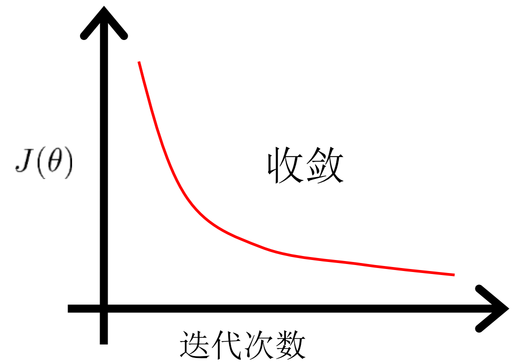
收敛条件：如定义收敛为 如果$J(\theta)$ 在一次迭代中减少不超过10^{-3} .

学习率

确保梯度下降算法正确的执行.

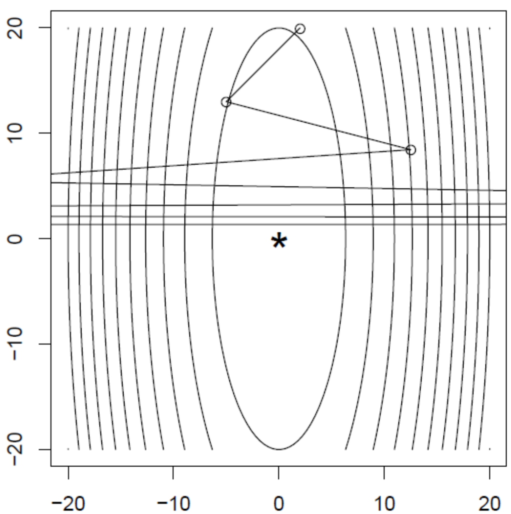




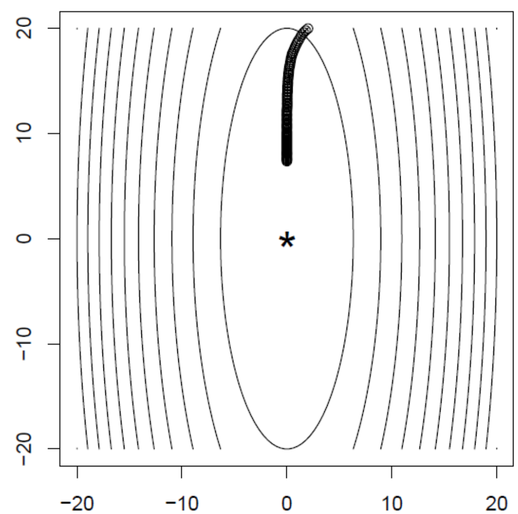


- 对于足够小的α, $J(\theta)$应该在每一次迭代中减小
- 如果α太小, 梯度下降算法则会收敛很慢
- 如果α太大, 梯度下降算法则不会收敛：发散或震荡

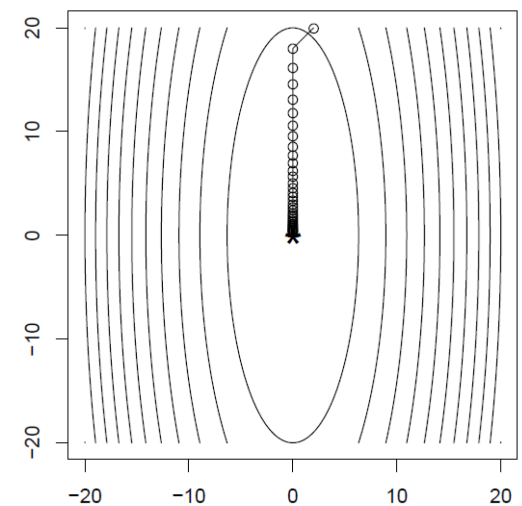
学习率



α过大：不收敛



α过小：收敛慢



α合适：收敛较快

理论：收敛分析

学习率

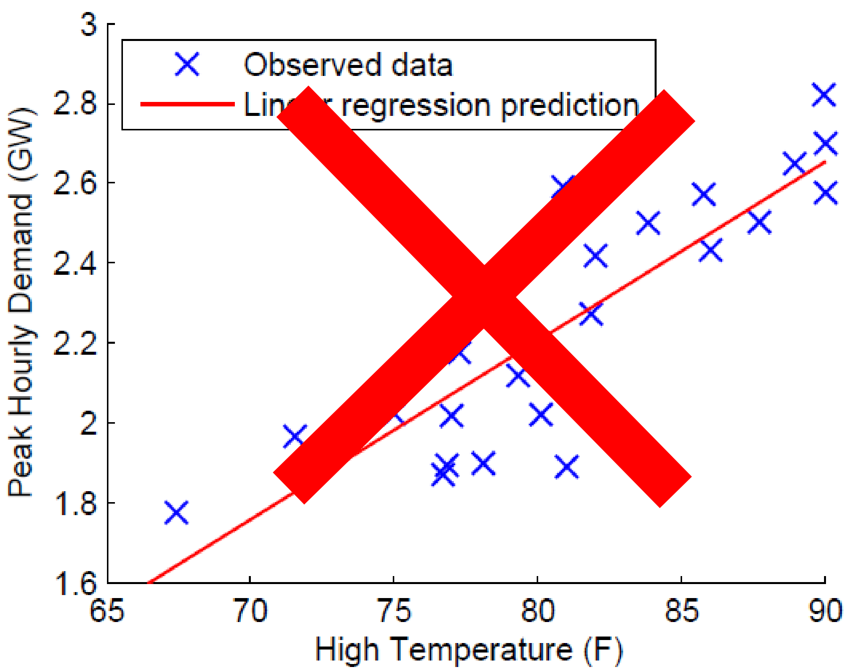
总结:

- 如果 α 太小: 收敛很慢.
- 如果 α 太大: $J(\theta)$ 可能不会在每一次迭代中减小; 并且可能不会达到收敛.

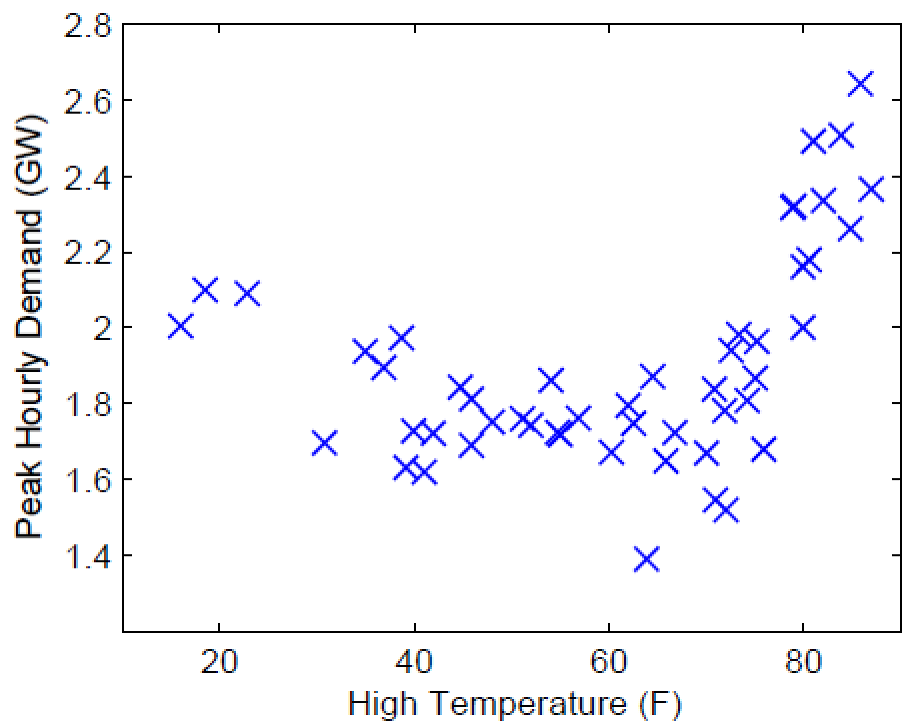
为了找到合适的 α, 可以尝试

$$\ldots, 0.001, \qquad , 0.01, \qquad , 0.1, \qquad , 1, \ldots$$

多项式回归



Observed data
Linear regression prediction
Peak Hourly Demand (GW)
High Temperature (F)
3
2.8
2.6
2.4
2.2
2
1.8
1.6
65
70
75
80
85
90

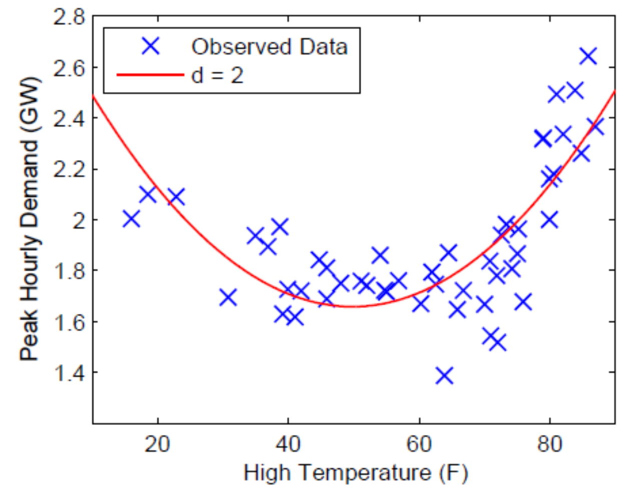


Peak Hourly Demand (GW)
High Temperature (F)
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
20
40
60
80

多项式回归

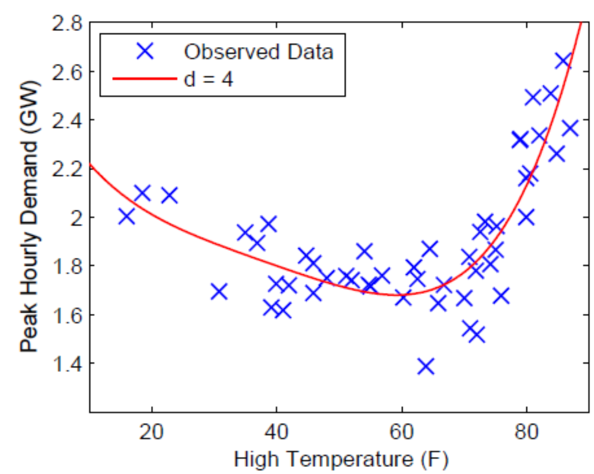
$$x^{(i)} \in \mathbb{R}^3 = \begin{bmatrix} 1 \\ \text{high temperature for day } i \\ (\text{high temperature for day } i)^2 \end{bmatrix}$$

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 = \theta_0 + \theta_1 x + \theta_2 x^2$$

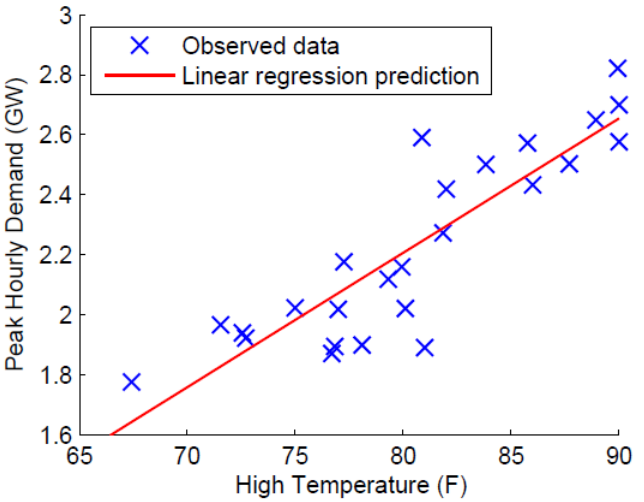


$$x^{(i)} \in \mathbb{R}^{d+1} = \begin{bmatrix} 1 \\ \text{high temperature for day } i \\ (\text{high temperature for day } i)^2 \\ \vdots \\ (\text{high temperature for day } i)^d \end{bmatrix}$$

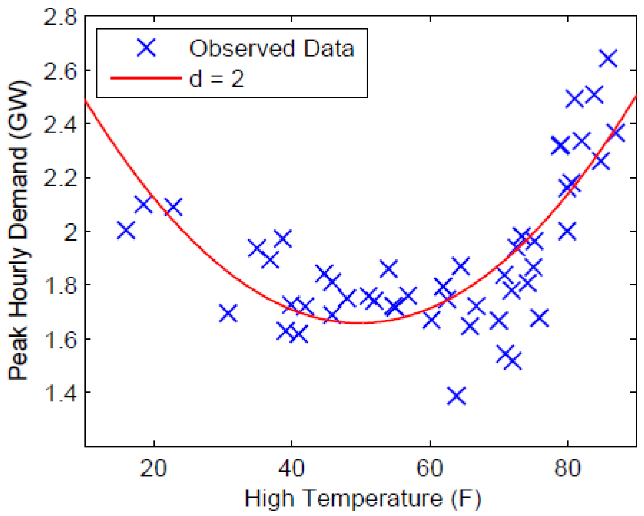
$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_d x_d = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_d x^d$$



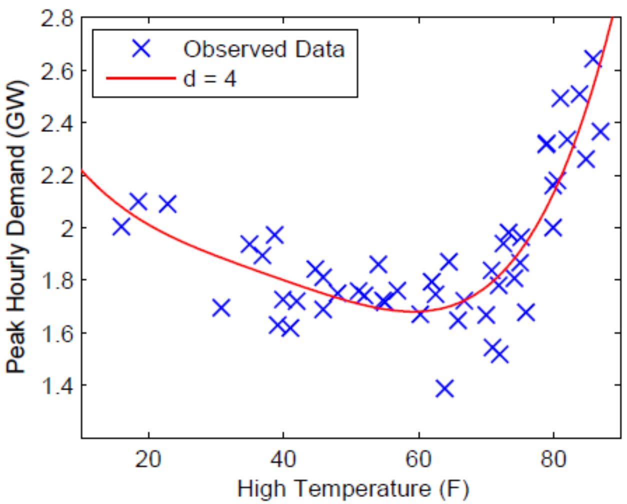
多项式回归



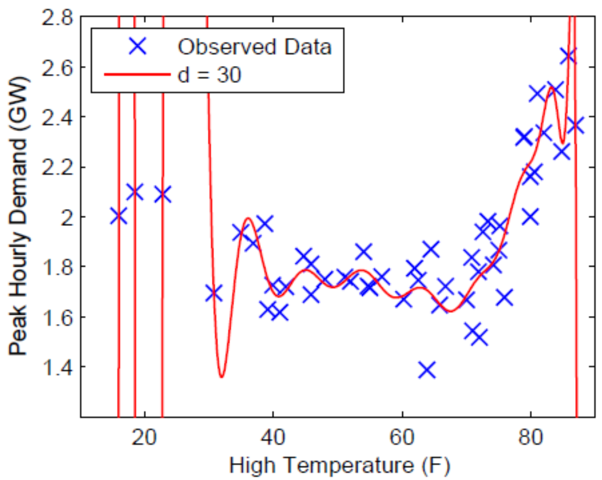
Observed data
Linear regression prediction
Peak Hourly Demand (GW)
High Temperature (F)



Observed Data
d = 2
Peak Hourly Demand (GW)
High Temperature (F)



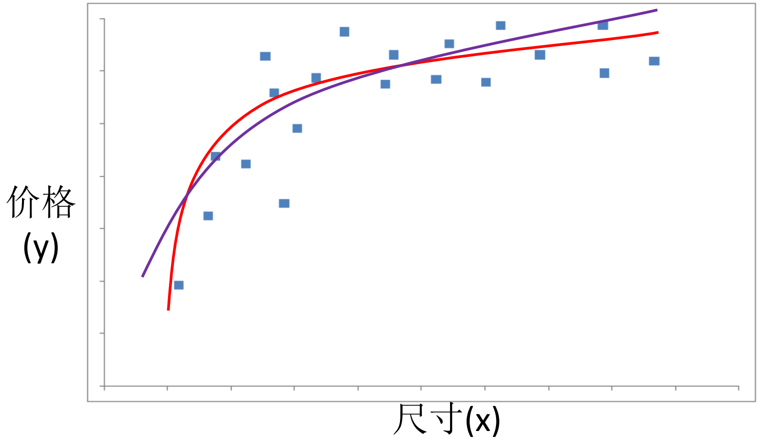
Observed Data
d = 4
Peak Hourly Demand (GW)
High Temperature (F)



Observed Data
d = 30
Peak Hourly Demand (GW)
High Temperature (F)

多项式回归

特征选择



$$h_\theta(x) = \theta_0 + \theta_1(size) + \theta_2(size)^2$$

$$h_\theta(x) = \theta_0 + \theta_1(size) + \theta_2\sqrt{(size)}$$

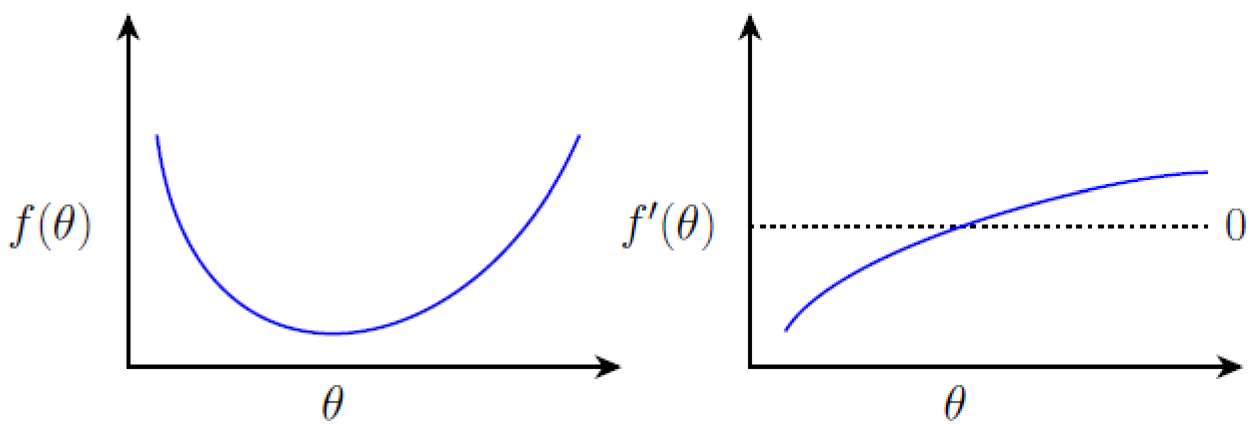
特征尺度归一化

正规方程（The normal equations）

- 对于求函数极小值问题，除了采用迭代的方法外，还有其他方法？

令函数的微分为零，然后求解方程！可得到解析解

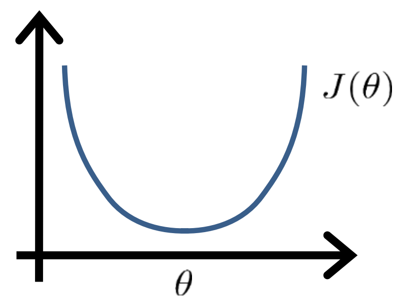
An example for one-dimensional θ



正规方程（The normal equations）

直观解释：如果是1维的$(\theta \in \mathbb{R})$

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



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

$$\frac{\partial}{\partial\theta_j}J(\theta) = \cdots = 0 \quad \text{(每一个}j\text{)}$$

解出 $\quad \theta_0, \theta_1, \ldots, \theta_n$

Reminder: 矩阵的迹

- $n \times n$ 矩阵A的迹$\mathtt{tr}(A)$ 定义为对角线上所有元素的和，即

$$\mathtt{tr}(A) = \sum_{i=1}^{n} A_{ii}$$

- 迹的性质：
  - 若两个矩阵A，B满足其乘积AB为方阵，则有$\mathtt{tr}(AB) = \mathtt{tr}(BA)$
  - 可以推出

$$\mathtt{tr}(ABC) = \mathtt{tr}(CAB) = \mathtt{tr}(BCA),$$
$$\mathtt{tr}(ABCD) = \mathtt{tr}(DABC) = \mathtt{tr}(CDAB) = \mathtt{tr}(BCDA).$$

  - 若A，B为方阵，a为标量

$$\mathtt{tr}(A) = \mathtt{tr}(A^T)$$
$$\mathtt{tr}(A+B) = \mathtt{tr}(A) + \mathtt{tr}(B)$$
$$\mathtt{tr}(aA) = a\mathtt{tr}(A)$$

Reminder: Matrix Derivatives

- 给定函数$f(A): \mathbb{R}^{m\times n} \mapsto \mathbb{R}$, 其对矩阵$A \in \mathbb{R}^{m\times n}$的微分可定义为

$$\nabla_A f(A) = \begin{bmatrix} \frac{\partial f}{\partial A_{11}} & \cdots & \frac{\partial f}{\partial A_{1n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial A_{m1}} & \cdots & \frac{\partial f}{\partial A_{mn}} \end{bmatrix}$$

因此该梯度$\nabla_A f(A)$可表示为一个$m\times n$矩阵, 其中(i,j)-th 元素为$\partial f/\partial A_{ij}$.

- 例如，给定2×2矩阵$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$以及对应的函数$f$为

$$f(A) = \frac{3}{2}A_{11} + 5A_{12}^2 + A_{21}A_{22}.$$

对应的梯度为

$$\nabla_A f(A) = \begin{bmatrix} \frac{3}{2} & 10A_{12} \\ A_{22} & A_{21} \end{bmatrix}.$$

Reminder: Matrix Derivatives

这里仅给出部分特定函数的矩阵微分(其他的可参考《The matrix cookbook》)

$$
\begin{aligned}
\nabla_A \mathtt{tr}(AB) &= B^T \\
\nabla_{A^T} f(A) &= (\nabla_A f(A))^T \\
\nabla_A \mathtt{tr}(ABA^T C) &= CAB + C^T AB^T \\
\nabla_A |A| &= |A|(A^{-1})^T,
\end{aligned}
$$

这里$|A|$ 表示方阵A的行列式. 根据上面的第2、3式可以有

$$\nabla_{A^T} \mathtt{tr}(ABA^T C) = B^T A^T C^T + BA^T C$$

Petersen, Kaare Brandt, and Michael Syskind Pedersen. "The matrix cookbook." *Technical University of Denmark* 7, no. 15 (2008): 510.

正规方程（The normal equations）

例子: $m = 4.$

x_0	尺寸 (feet²) x_1	卧室个数 x_2	楼层 x_3	房龄（年） x_4	价格 ($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**

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

$$X = \begin{bmatrix} \text{—} \; (x^{(1)})^T \; \text{—} \\ \text{—} \; (x^{(2)})^T \; \text{—} \\ \vdots \\ \text{—} \; (x^{(m)})^T \; \text{—} \end{bmatrix} \qquad y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

正规方程（The normal equations）

矢量化表示: $h_\theta(x^{(i)}) = (x^{(i)})^T\theta$

$$
\begin{aligned}
X\theta - y &= \begin{bmatrix} (x^{(1)})^T\theta \\ \vdots \\ (x^{(m)})^T\theta \end{bmatrix} - \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix} \\
&= \begin{bmatrix} h_\theta(x^{(1)}) - y^{(1)} \\ \vdots \\ h_\theta(x^{(m)}) - y^{(m)} \end{bmatrix}
\end{aligned}
$$

$$
\begin{aligned}
\frac{1}{2m}(X\theta - y)^T(X\theta - y) &= \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 \\
&= J(\theta)
\end{aligned}
$$

正规方程（The normal equations）

$$J(\theta) = \tfrac{1}{2m}(X\theta - y)^T(X\theta - y)$$

$$\begin{aligned}
\nabla_\theta J(\theta) &= \nabla_\theta \frac{1}{2m}(X\theta - y)^T(X\theta - y) \\
&= \frac{1}{2m}\nabla_\theta \left(\theta^T X^T X\theta - \theta^T X^T y - y^T X\theta + y^T y\right) \\
&= \frac{1}{2m}\nabla_\theta \,\mathtt{tr}\left(\theta^T X^T X\theta - \theta^T X^T y - y^T X\theta + y^T y\right) \\
&= \frac{1}{2m}\nabla_\theta \left(\mathtt{tr}(\theta^T X^T X\theta) - 2\mathtt{tr}(y^T X\theta)\right) \\
&= \frac{1}{2m}\left(X^T X\theta + X^T X\theta - 2X^T y\right) \\
&= \frac{1}{m}\left(X^T X\theta - X^T y\right) = 0
\end{aligned}$$

$$\boxed{X^T X\theta = X^T y}$$

Thus, the value of θ that minimizes $J(\theta)$ is given in closed form by the equation

$$\theta = (X^T X)^{-1} X^T y.$$

梯度下降 vs. 正规方程

m **训练样本,** n **个特征.**

- 梯度下降
  - 需要选择合适的α.
  - 需要多次迭代.
  - 即使n很大，效果也很好.
- 正规方程
  - 不需要选择α.
  - 不需要迭代.
  - 需要计算$(X^TX)^{-1}$
  - 如果n很大，速度将会很慢.

矩阵不可逆情况下怎么办?

- 太多的特征 (如 $m \leq n$).
  - 删减一些特征, 或者进行正则化.

Thanks!

Any questions?