Multivariate Linear Regression 多元线性回归

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

$\Downarrow$

训练数据不止一个特征。

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

($x_0^{(i)} = 1$: 第i个样本的第0个特征定义为1.) $x_j^{(i)}$: 第i个样本的第j个特征

$\Downarrow$

Hypothesis

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

$$= \theta^T x = [\theta_0 \ \theta_1 \cdots \theta_n]_{1\times n+1} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}_{(n+1)\times 1}$$

intuition : x_1 面积、 x_2 卫生间个数 x_3 卧室

房价 $\Leftarrow$ θ_1 每平方米单价 θ_2 每个卫生间价格 $\theta_3 \cdots$

Parameter: $\theta \in \mathbb{R}^{n+1} = \theta_0, \theta_1, \theta_2, \ldots \theta_n$

Cost Function $J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}[h_\theta(x^{(i)}) - y^{(i)}]^2$

Gradient decent:

Repeat until convergence: { $\theta_j := \theta_j - \alpha \frac{\partial}{\partial\theta_j} J(\theta)$ } for $j := 0, \ldots,$ m

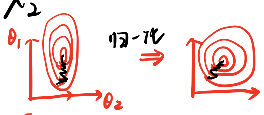
$$\frac{\partial}{\partial\theta_j} J(\theta) = \frac{1}{m}\sum_{i=1}^{n}[h_\theta(x^{(i)}) - y^{(i)}] \cdot x_j^{(i)}$$

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

$\theta_1 = \theta_1 - \cdots\cdots x_1^{(i)}$

$\theta_2 = \theta_2 - \cdots\cdots x_2^{(i)}$

eg.



$\theta_1 \in [0, 2000]$ $\theta_2 \in (0, 5)$

Gradient Descent in practice

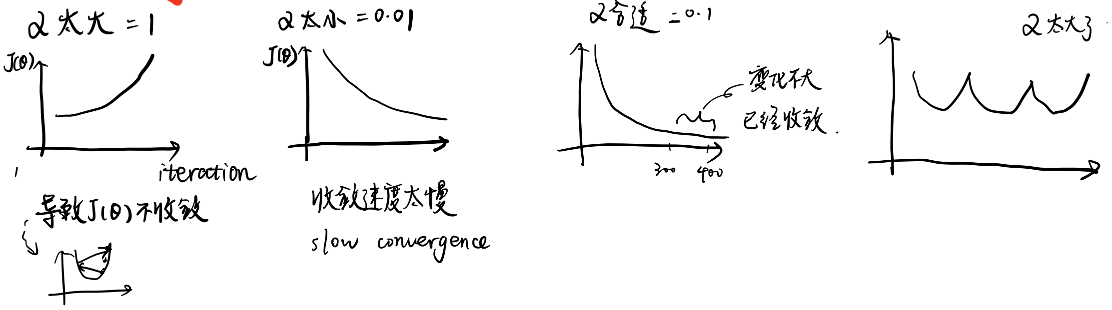
Way to speed up — Feature Scaling 缩放

因为如果θ的范围太大，那么会收敛得很慢，且有可能在最优值附近振荡

- feature scalling
- mean normalization: $\frac{x - \mu}{s}$ (scale of the data / standard deviation)

$\rightarrow x \in (-0.5, 0.5)$ / $(-1, 1)$ / ......

Learning Rate α



Polynominal Regression

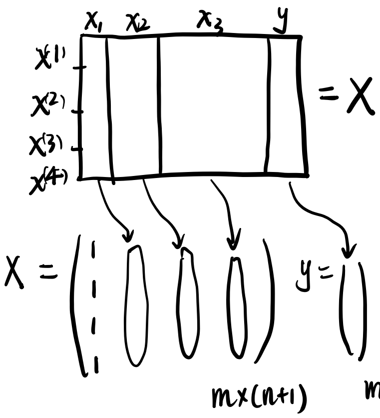
$y=\theta_0+\theta_1 x+\underset{\sim}{\theta_2 x^2}+\theta_3 \underset{\sim}{x^3}$

or: $y=\theta_0+\theta_1 x+\theta_2\sqrt{x}$

eg: size = 1~1000
$size^2 = 1\sim10^6$
$size^3 = 1\sim10^{12}$
Feauture scale 注意!

Nomal Equation

$$y=\theta_0+\theta_1x_1+\theta_2x_2+\theta_3x_3=(\theta_0\ \theta_1\ \theta_2\ \theta_3)\begin{pmatrix}1\\x_1\\x_2\\x_3\end{pmatrix}$$



$$y=\begin{pmatrix}1 & x^{(1)}\\1 & x^{(2)}\\1 & x^{(3)}\\1 & x^{(4)}\end{pmatrix}$$

m×2

$\theta=(X^TX)^{-1}X^Ty$ → in octave: pinv(X'*X)*X'*Y

如果采用了这种形式表示，就不用做Feature Scaling; $x\in[0,10^{-5}]$ ✓ $x\in[0,10^4]$ ✓ OK.

m training samples, n features

Gradient Descent VS Normal Equation

1. need to choose α　　1. 不需要选择α

2. Need many iterations　　2. 不需迭代

---

3. Work well even n is large　　3. 需要计算 $(X^TX)^{-1}$ n×n: $O(n^3)$

✓更常用.　　4. 当n很大时，会很慢.
eg n > 10000 时.

若 X^TX 不可逆 (non-invertible)
① 有重复特征 (2列相同)
② 太多特征 (m ≤ n) (m×n)