

多变量线性回归

Hypothesis: $h_{\theta}(x) = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$ ↗ $x_0 = 1$

Parameters: $\theta_0, \theta_1, \dots, \theta_n$ ○ n+1-dimensional vector

Cost function:

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

$J(\theta)$

Gradient descent:

Repeat {

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

}

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

Gradient Descent

Previously ($n=1$):

Repeat {

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

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

(simultaneously update θ_0, θ_1)

}

New algorithm ($n \geq 1$):

Repeat {

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

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

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

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

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

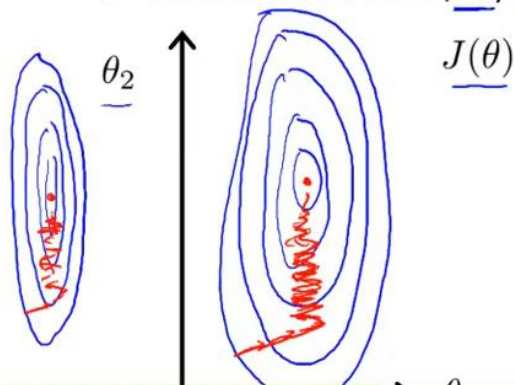
特征收缩 feature scaling

Feature Scaling

Idea: Make sure features are on a similar scale.

E.g. $x_1 = \text{size (0-2000 feet}^2)$ ←

$x_2 = \text{number of bedrooms (1-5)}$ ←



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

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

$$0 \leq x_1 \leq 1 \quad 0 \leq x_2 \leq 1$$



Feature Scaling

Get every feature into approximately a $-1 \leq x_i \leq 1$ range.

$$x_0 = 1$$

$$0 \leq x_1 \leq 3 \quad \checkmark$$

$$-2 \leq x_2 \leq 0.5 \quad \checkmark$$

$$-100 \leq x_3 \leq 100 \quad \times$$

$$-0.0001 \leq x_4 \leq 0.0001 \quad \times$$

$$-3 \text{ to } 3 \quad \checkmark$$

$$-\frac{1}{3} \text{ to } \frac{1}{3} \quad \checkmark$$

Mean normalization 均值归一化

μ 为均值, S 为范围-即最大值减最小值

Mean normalization

Replace x_i with $x_i - \mu_i$ to make features have approximately zero mean
(Do not apply to $x_0 = 1$).

E.g. $\rightarrow x_1 = \frac{\text{size} - 1000}{2000}$

Average size = 1000

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

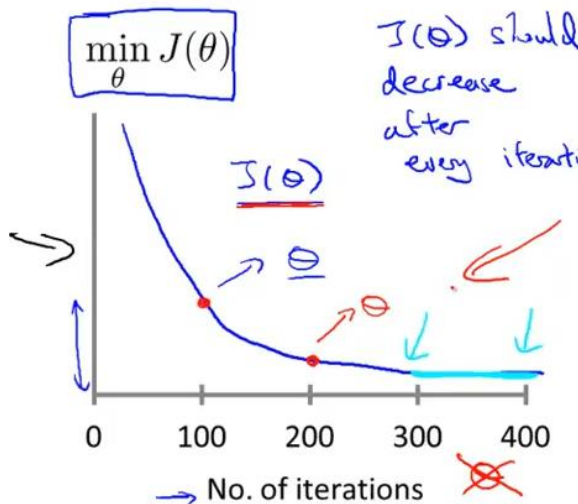
1-5 bedrooms

$$-0.5 \leq x_1 \leq 0.5 \quad -0.5 \leq x_2 \leq 0.5$$

$$x_1 \leftarrow \frac{x_1 - \mu_1}{s_1} \quad x_2 \leftarrow \frac{x_2 - \mu_2}{s_2}$$

Handwritten notes:
- μ_1 is circled in red, with an arrow pointing to it from the text "avg value of x_1 in training set".
- s_1 is circled in red.
- μ_2 is circled in red.

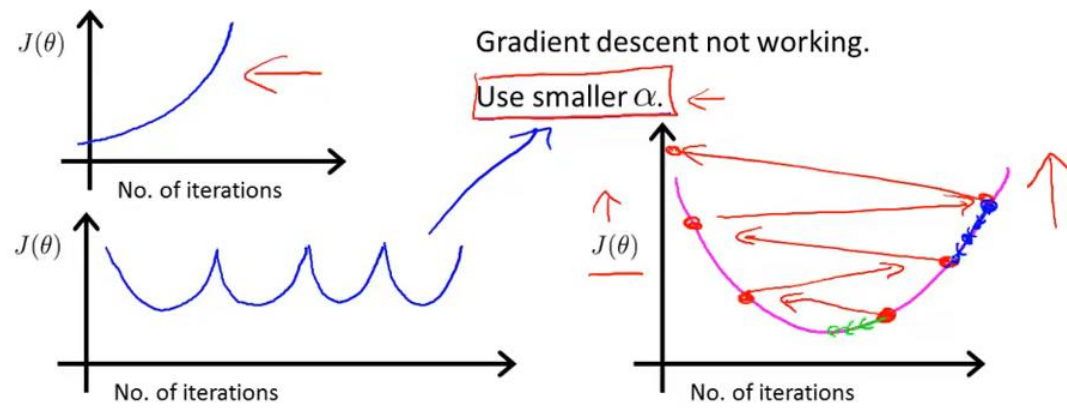
Making sure gradient descent is working correctly.



Example automatic convergence test:

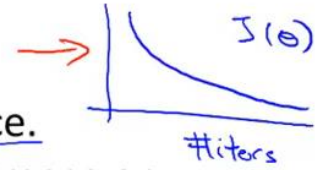
Declare convergence if $J(\theta)$ decreases by less than 10^{-3} in one iteration.

Making sure gradient descent is working correctly.



Summary:

- If α is too small: slow convergence.
- If α is too large: $J(\theta)$ may not decrease on every iteration; may not converge. (Slow converge also possible.)



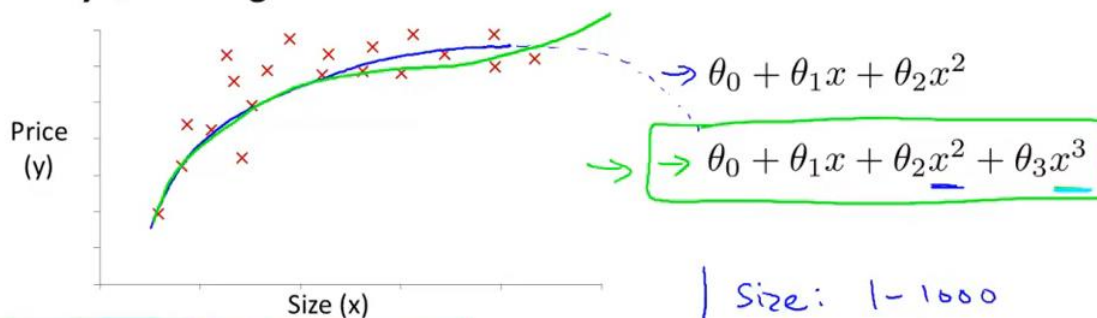
To choose α , try

..., 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, ...

↑ ↗ ↗ ↗ ↗ ↗ ↗ ↗

多项式回归 polynomial regression

Polynomial regression



$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$

$$= \theta_0 + \theta_1(\text{size}) + \theta_2(\text{size})^2 + \theta_3(\text{size})^3$$

→ $x_1 = (\text{size})$

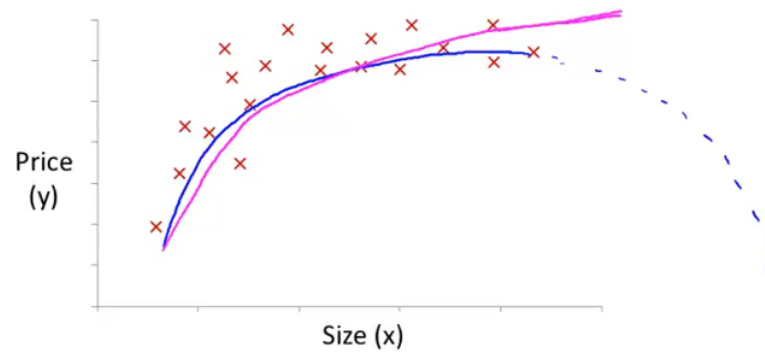
→ $x_2 = (\text{size})^2$

Size: 1-1000

Size²: 1-1000,000

Size³: 1-10⁹

Choice of features



$$\rightarrow h_{\theta}(x) = \theta_0 + \theta_1(\text{size}) + \theta_2(\text{size})^2$$

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



正规方程 Normal equation

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

$(X^T X)^{-1}$ is inverse of matrix $X^T X$.

Set $A = X^T X$

$$(X^T X)^{-1} = A^{-1}$$

Octave: `pinv(X' * X) * X' * y`

$$\text{pinv}(X^T * X) * X^T * y$$

$$\theta = (X^T X)^{-1} X^T y \quad \min J(\theta)$$

X' X^T

~~Feature Scaling~~

$0 \leq x_1 \leq 1$

$0 \leq x_2 \leq 1000$

$0 \leq x_3 \leq -5$ ✓

梯度下降法 与 正规方程法 的比较

m training examples, n features.

Gradient Descent

- • Need to choose α .
- • Needs many iterations.
- Works well even when n is large.

$n = 10^6$

Normal Equation

- • No need to choose α .
- • Don't need to iterate.
- Need to compute $(X^T X)^{-1}$ $n \times n$ $O(n^3)$
- • Slow if n is very large.

$n = 100$
 $n = 1000$
 $n = 10000$

如何提交

复制下面的识别码，运行下载作业内容中的提交脚本。系统提示时，提交您的电子邮箱地址 `xiaoyu1qh1@163.com`。

lWN1ujjsZLfHwe2o

[生成新的识别码](#)

提交识别码可以标识您的身份，因此请不要告诉其他人。您可以无限次数提交。