Linear Regression with multiple variables

Multiple features

Multiple features (variables).

Size (feet²) x_1	Number of bedrooms x_2	Number of floors x_3	Age of home (years) x_4	Price ($1000) y
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
...	...	...	...	...

$m = 47$

Notation:

- n = number of features $n = 4$
- $x^{(i)}$ = input (features) of i^{th} training example.
- $x_j^{(i)}$ = value of feature j in i^{th} training example.

$x^{(2)} = \begin{bmatrix} 1416 \\ 3 \\ 2 \\ 40 \end{bmatrix}$

$x_3^{(2)} = 2$

Andrew Ng

Multivariate linear regression:

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

For convenience of notation, define $\boxed{x_0 = 1.}$ $(x_0^{(i)} = 1)$

$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1}$ $\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \in \mathbb{R}^{n+1}$

$\underbrace{[\theta_0 \ \theta_1 \cdots \theta_n]}_{\theta^T}$ $\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$

$(n+1) \times 1$ matrix x

$$h_\theta(x) = \theta_0 \overset{=1}{x_0} + \theta_1 x_1 + \cdots + \theta_n x_n$$

$$= \boxed{\theta^T x.}$$

$\theta^T x$

Gradient descent for multiple variables

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

Parameters: $\cancel{\theta_0, \theta_1, \ldots, \theta_n}$ θ $n+1$ - dimensional vector

Cost function:

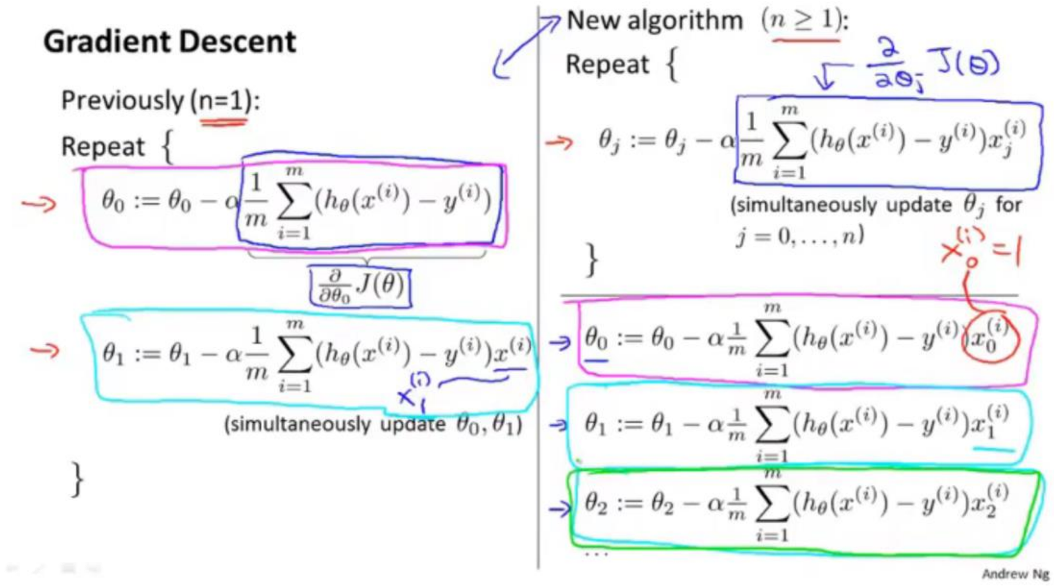
$$\cancel{J(\theta_0, \theta_1, \ldots, \theta_n)} \ J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Gradient descent:

Repeat {

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

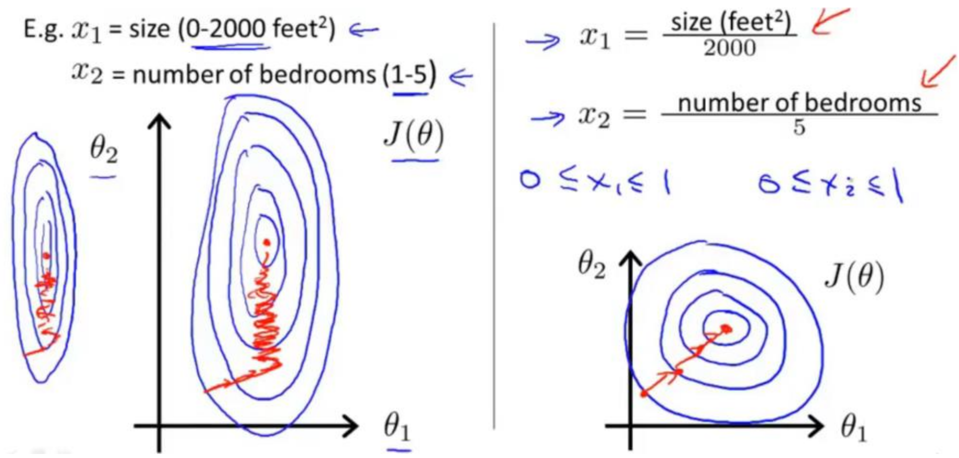
} (simultaneously update for every $j = 0, \ldots, n$)



Gradient descent in practice I: Feature Scaling

Feature Scaling

Idea: Make sure features are on a similar scale.(Then gradient descents can converge more quickly)



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

$x_0 = 1$

$0 \leq x_1 \leq 3$ ✓

$-2 \leq x_2 \leq 0.5$ ✓

$-100 \leq x_3 \boxed{100}$ ✗

$-0.0001 \leq x_4 \leq \boxed{0.0001}$ ✗

-3 to 3 ✓

$-\frac{1}{3}$ to $\frac{1}{3}$ ✓

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. $\boxed{x_1 = \frac{size - 1000}{2000}}$ Average size = 100

$\boxed{x_2 = \frac{\#bedrooms - 2}{5}}$ 1-5 bedrooms

$\boxed{-0.5 \le x_1 \le 0.5}$ $\boxed{-0.5 \le x_2 \le 0.5}$

$x_1 \leftarrow \frac{x_1 - \mu_1}{s_1}$ avg value of x_1 in training set

s_1: range (max - min) (or standard deviation)

$x_2 \leftarrow \frac{x_2 - \mu_1}{s_2}$

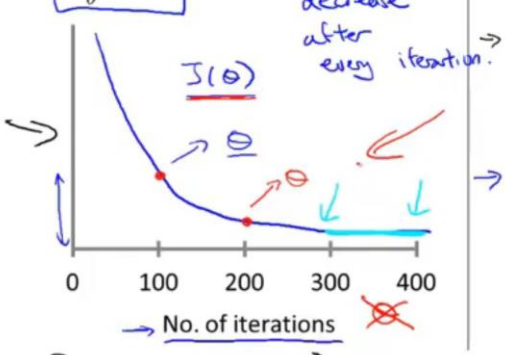
Andrew Ng

Gradient descent in practice II: Learning rate

Making sure gradient descent is working correctly.

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

$J(\theta)$ should decrease after every iteration.



30, 3000, 3,000,000

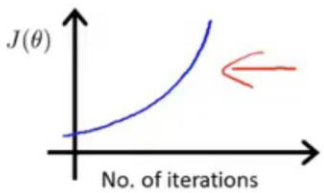
Example automatic convergence test:

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

ε

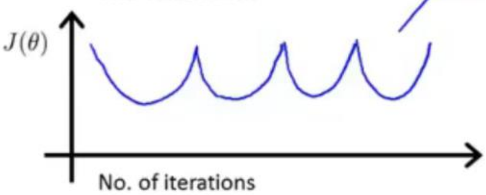
Andrew Ng

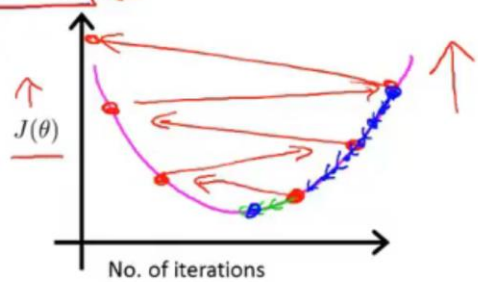
Making sure gradient descent is working correctly.



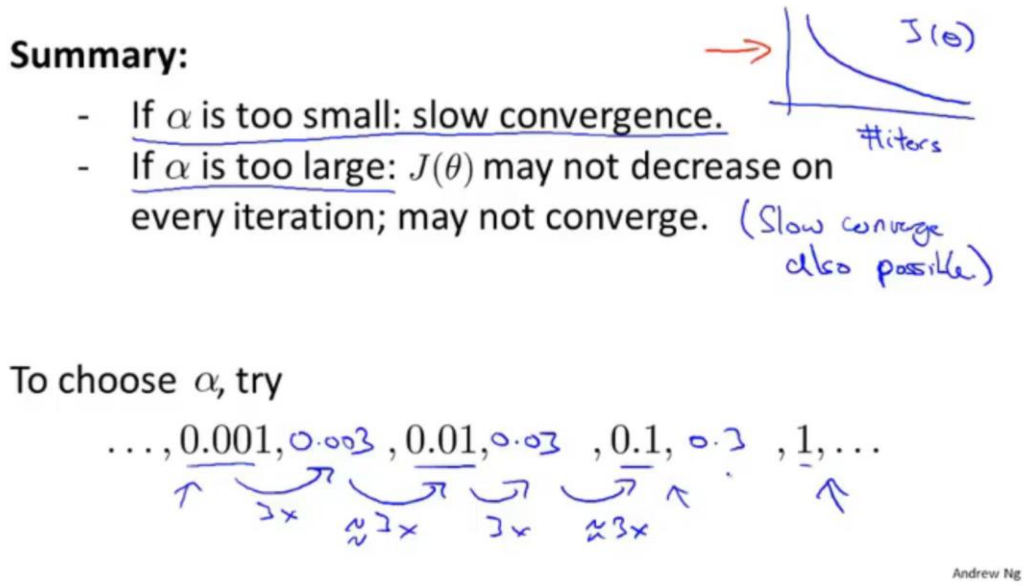
Gradient descent not working.

$\boxed{\text{Use smaller } \alpha.}$



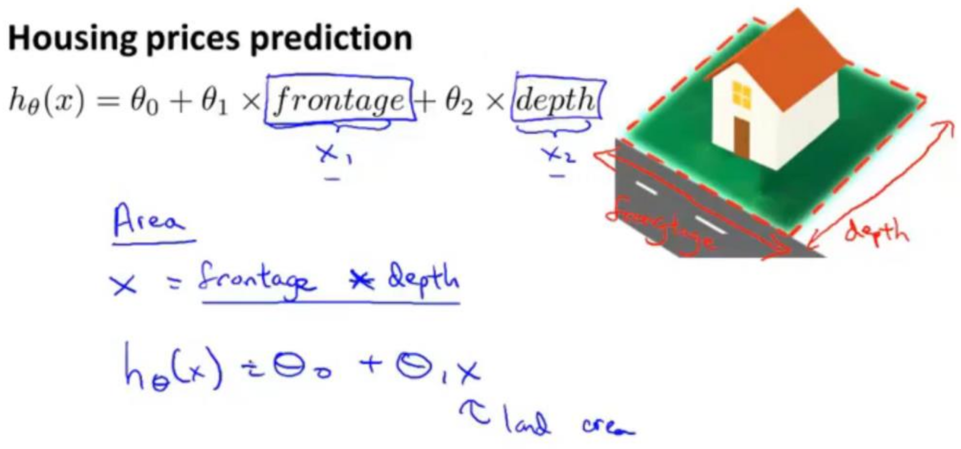


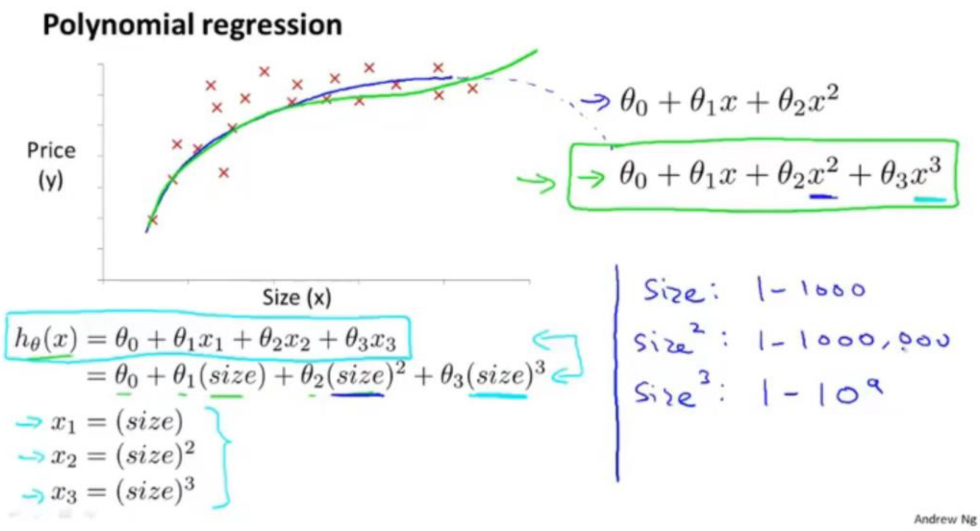
- For sufficiently small α, $J(\theta)$ should decrease on every iteration.
- But if α is too small, gradient descent can be slow to converge.



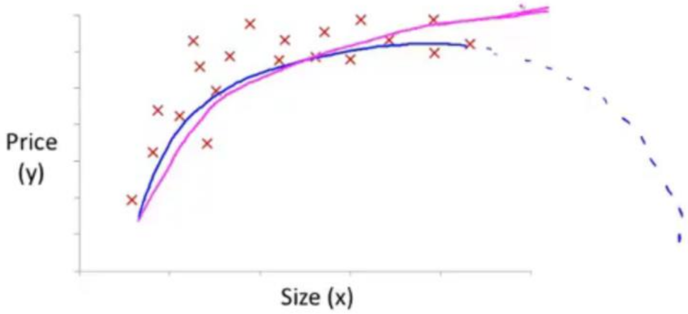
Feature and polynomial regression

define new features: x = frontage * depth.





Choice of features



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

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

Andrew Ng

Normal equation

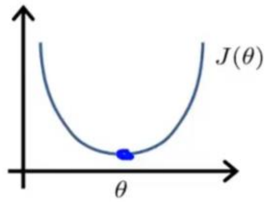
Normal equation: Method to solve for theta analytically.

Intuition: If 1D $(\theta \in \mathbb{R})$

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

$$\frac{\partial}{\partial \theta} J(\theta) = \dots \overset{\text{set}}{=} 0$$

Solve for θ



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

$\frac{\partial}{\partial \theta_j} J(\theta) = \cdots \overset{\text{set}}{=} 0$ $\qquad$ (for every j)

Solve for $\theta_0, \theta_1, \dots, \theta_n$

Examples: $m = 4$.

	Size (feet²)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
x_0	x_1	x_2	x_3	x_4	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

$$X = \underbrace{\begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix}}_{m \times (n+1)} \qquad y = \underbrace{\begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}}_{m\text{-dimensional vector}}$$

$$\boxed{\theta = (X^TX)^{-1}X^Ty} \leftarrow$$

m **examples** $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$; n **features.**

$$x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$\underset{\text{(design matrix)}}{X} = \underbrace{\begin{bmatrix} —\, (x^{(1)})^T \,— \\ —\, (x^{(2)})^T \,— \\ \vdots \\ —\, (x^{(m)})^T \,— \end{bmatrix}}_{m \times (n+1)}$$

E.g. If $x^{(i)} = \begin{bmatrix} 1 \\ x_1^{(i)} \end{bmatrix}$

$$X = \underbrace{\begin{bmatrix} 1 & x_1^{(1)} \\ 1 & x_1^{(2)} \\ \vdots & \vdots \\ 1 & x_1^{(m)} \end{bmatrix}}_{m \times 2} \qquad y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

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

Andrew Ng

Feature scaling isn’t actually necessary in normal equation.

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

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

Set $A = X^T X$

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

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

X' X^T

pinv$(X^T * X) * X^T * y$

$\Theta = (X^T X)^{-1} X^T y$ $\quad \min_{\Theta} J(\Theta)$

~~Feature Scaling~~

$0 \le x_1 \le 1$

$0 \le x_2 \le 1000$

$0 \le x_3 \le 10^{-5}$ ✓

m **training examples,** n **features.**

Gradient Descent	Normal Equation
• Need to choose α.	• No need to choose α.
• Needs many iterations.	• Don’t need to iterate.
• Works well even when n is large.	• Need to compute $(X^T X)^{-1}$ $\quad n \times n \quad O(n^3)$
$n = 10^6$	• Slow if n is very large.
	$n = 100$
	$n = 1000$
	$n = 10000$

Normal equation and non-invertibility

Normal equation

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

$X^T X$

- What if $\boxed{X^T X}$ is non-invertible? (singular/ degenerate)
- Octave: `pinv(X'*X)*X'*y`

θ

$\boxed{\text{pinv}}$

inv

What if $\boxed{X^T X}$ is non-invertible?

- Redundant features (linearly dependent).
  E.g. x_1 =size in feet^2
  ~~x_2 =size in m^2~~
  $x_1 = (3.28)^2 x_2$

$1m = 3.28$ feet

$m = 10$

$n = 100$

$\theta \in \mathbb{R}^{101}$

- Too many features (e.g. $m \leq n$).
  - Delete some features, or use regularization.

later