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

Parameters: $\theta_0, \theta_1, \dots, \theta_n \approx \Theta$ $n+1$ dimensional vector

Cost function

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

Gradient descent:

Repeat {

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

}

$n \geq 1$

Repeat {

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

}

Feature Scaling

Make sure features on a similar scale

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

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

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

$0 \leq x_1 \leq 3$ ✓

$-2 \leq x_2 \leq 0.5$ ✓ $\qquad -3$ to 3 ✓

$-100 \leq x_3 \leq 100$ ✗ $\qquad -1/3$ to $1/3$ ✓

$-.0001 \leq x_4 \leq 0.0001$ ✗

mean normalization

replace x_i with $x_i - \mu_i$ to make features have approximately zero mean

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

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

range (max-min) (or standard deviation)

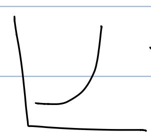
$\frac{x_1 - \mu_1}{s_1}$ $\mu_1 = 81$ $s_1 = 25$

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

making sure GD working correctly

automatic convergence test

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



use smaller α

For sufficiently small α, $J(\theta)$ should decrease on every iteration

But if α is too small, gradient descent takes too long

If α is too small: slow convergence

If α is too large: $J(\theta)$ may not decrease on every iteration; may not converge

---

Normal equation: Method to solve for θ analytically

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

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

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

Solve for θ

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

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

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

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

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

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 $(X^T X)^{-1}$ $n \times n$ $O(n^3)$
	· Slow if n is very large
$n = 10^6$	$n = 100$ ✓
	$n = 1000$ ✓

← $n = 10000$ ~